Premier Reference Source

Integrating the Internet of Things Into Software Engineering Practices

D. Jeya Mala

IGI Global

Integrating the Internet of Things Into Software Engineering Practices

D. Jeya Mala
Thiagarajar College of Engineering, India

A volume in the Advances in Systems Analysis, Software Engineering, and High Performance Computing (ASASEHPC) Book Series

IGI Global
DISSEMINATOR OF KNOWLEDGE

Published in the United States of America by
IGI Global
Engineering Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue
Hershey PA, USA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: http://www.igi-global.com

Library of Congress Cataloging-in-Publication Data

Names: Mala, D. Jeya, 1974- editor.
Title: Integrating the internet of things into software engineering practices / D. Jeya Mala, editor.
Description: Hershey, PA : Engineering Science Reference, [2019] | Includes bibliographical references.
Identifiers: LCCN 2018036342| ISBN 9781522577904 (h/c) | ISBN 9781522577911 (eISBN)
Subjects: LCSH: Internet of things. | Software engineering.
Classification: LCC TK5105.8857 .I526 2019 | DDC 004.67/8--dc23 LC record available at https://lccn.loc.gov/2018036342

This book is published in the IGI Global book series Advances in Systems Analysis, Software Engineering, and High Performance Computing (ASASEHPC) (ISSN: 2327-3453; eISSN: 2327-3461)

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

All work contributed to this book is new, previously-unpublished material.
The views expressed in this book are those of the authors, but not necessarily of the publisher.

For electronic access to this publication, please contact: eresources@igi-global.com.

Advances in Systems Analysis, Software Engineering, and High Performance Computing (ASASEHPC) Book Series

ISSN:2327-3453
EISSN:2327-3461

Editor-in-Chief: Vijayan Sugumaran, Oakland University, USA

MISSION

The theory and practice of computing applications and distributed systems has emerged as one of the key areas of research driving innovations in business, engineering, and science. The fields of software engineering, systems analysis, and high performance computing offer a wide range of applications and solutions in solving computational problems for any modern organization.

The **Advances in Systems Analysis, Software Engineering, and High Performance Computing (ASASEHPC) Book Series** brings together research in the areas of distributed computing, systems and software engineering, high performance computing, and service science. This collection of publications is useful for academics, researchers, and practitioners seeking the latest practices and knowledge in this field.

COVERAGE

- Virtual Data Systems
- Software Engineering
- Storage Systems
- Engineering Environments
- Human-Computer Interaction
- Network Management
- Enterprise Information Systems
- Distributed Cloud Computing
- Metadata and Semantic Web
- Performance Modelling

The Advances in Systems Analysis, Software Engineering, and High Performance Computing (ASASEHPC) Book Series (ISSN 2327-3453) is published by IGI Global, 701 E. Chocolate Avenue, Hershey, PA 17033-1240, USA, www.igi-global.com. This series is composed of titles available for purchase individually; each title is edited to be contextually exclusive from any other title within the series. For pricing and ordering information please visit http://www.igi-global.com/book-series/advances-systems-analysis-software-engineering/73689. Postmaster: Send all address changes to above address.

Titles in this Series

For a list of additional titles in this series, please visit:
https://www.igi-global.com/book-series/advances-systems-analysis-software-engineering/73689

Analyzing the Role of Risk Mitigation and Monitoring in Software Devlopment
Rohit Kumar (Chandigarh University, India) Anjali Tayal (Infosys Technologies, India) and Sargam Kapil (C-DAC, India)
Engineering Science Reference • ©2018 • 308pp • H/C (ISBN: 9781522560296) • US $225.00

Handbook of Research on Pattern Engineering System Development for Big Data Analytics
Vivek Tiwari (International Institute of Information Technology, India) Ramjeevan Singh Thakur (Maulana Azad National Institute of Technology, India) Basant Tiwari (Hawassa University, Ethiopia) and Shailendra Gupta (AISECT University, India)
Engineering Science Reference • ©2018 • 396pp • H/C (ISBN: 9781522538707) • US $320.00

Incorporating Nature-Inspired Paradigms in Computational Applications
Mehdi Khosrow-Pour, D.B.A. (Information Resources Management Association, USA)
Engineering Science Reference • ©2018 • 385pp • H/C (ISBN: 9781522550204) • US $195.00

Innovations in Software-Defined Networking and Network Functions Virtualization
Ankur Dumka (University of Petroleum and Energy Studies, India)
Engineering Science Reference • ©2018 • 364pp • H/C (ISBN: 9781522536406) • US $235.00

Advances in System Dynamics and Control
Ahmad Taher Azar (Benha University, Egypt) and Sundarapandian Vaidyanathan (Vel Tech University, India)
Engineering Science Reference • ©2018 • 680pp • H/C (ISBN: 9781522540779) • US $235.00

Green Computing Strategies for Competitive Advantage and Business Sustainability
Mehdi Khosrow-Pour, D.B.A. (Information Resources Management Association, USA)
Business Science Reference • ©2018 • 324pp • H/C (ISBN: 9781522550174) • US $185.00

For an entire list of titles in this series, please visit:
https://www.igi-global.com/book-series/advances-systems-analysis-software-engineering/73689

IGI Global
DISSEMINATOR OF KNOWLEDGE

701 East Chocolate Avenue, Hershey, PA 17033, USA
Tel: 717-533-8845 x100 • Fax: 717-533-8661
E-Mail: cust@igi-global.com • www.igi-global.com

Editorial Advisory Board

Table of Contents

Detailed Table of Contents

aims at unifying software development (Dev) and software operation (Ops). The main characteristic of DevOps is the automation and monitoring at all steps of software construction, from integration, testing, releasing to deployment and infrastructure management. DevSecOps is a practice of integrating security into every aspect of an application lifecycle from design to development.

Despite the rapid growth in IoT research, a general principled software engineering approach for the systematic development of IoT systems and applications is still missing. Software engineering as a discipline provides the necessary platform to carry on the underlying design, coding, implementation, as well as maintenance of such systems. UML diagrams present a visually comprehensible outlay of the construction of IoT systems. The chapter covers the modelling of IoT systems using UML diagrams. Starting with the architectural design of any IoT system to behavioral aspects is covered in this chapter using a case study of IoT-based remote patient health monitoring system. The diagrams shown in this chapter are the sample diagrams for understanding IoT-based complex systems. The chapter focuses on the work carried out by Franco Zambonelli in context of developing abstract model of an IoT system using software engineering concepts. The chapter also focus on the pioneer work carried by J. F. Peters in intelligent system design patterns for robotic devices using pattern classification.

Wireless sensor network (WSN) is an outdated technology that is used to monitor the physical changes in environment and take necessary actions. The advancement in WSN leads to automation in physical environment by uploading the sensed data to internet or cloud. The internet of things concept deals with the issues of making things connected to the internet as well as in a network of smart devices. IoT application development presents an enormous opportunity to reshape entire industries. According to McKinsey & Co, the merging of the physical and digital worlds via IoT could generate up to $11.1 trillion a year in economic value by 2025. Hence, the development of the web-based IoT applications will take automation research to the next level. Many authors have proposed many solutions to make internet of things possible in day-to-day life. This chapter gives an introduction about the web-based application development based on internet of things. The major objective of this chapter is to discuss and resolve the challenges in IoT to automate the real-time problems.

The internet of things (IoT) is aimed at modifying the life of people by adopting the possible computing techniques to the physical world, and thus transforming the computing environment from centralized form to decentralized form. Most of the smart devices receive the data from other smart devices over the network and perform actions based on their implemented programs. Thus, testing becomes an intensive process in the IoT that will require some normalization too. The composite architecture of IoT systems and their distinctive characteristics require different variants of testing to be done on the components of IoT systems. This chapter will discuss the necessity for IoT testing in terms of various criteria of identifying and fixing the problems in the IoT systems. In addition, this chapter examines the core components to be focused on IoT testing and testing scope based on IoT device classification. It also elaborates the various types of testing applied on healthcare IoT applications, and finally, this chapter summarizes the various challenges faced during IoT testing.

In the IoT applications development process, the consumers expectations are always high. Thus, the development environment should be focusing on virtual provisioning, manipulation, and testing and debugging. This has also raised more challenges in terms of proper testing to be done in both user interface level as well as the functionality level. It will be really challenging to test a connected device within a full IoT environment, which will have more devices with varied functionalities and data processing. These challenges have made a new way of testing to be done so that the test cases will be more efficient in revealing the errors in the software. In this chapter, UML use case diagram-based test cases generation for an IoT environment is explained in detail. Also, a real-time case study IoT application is taken to showcase how this approach helps in generating the test cases to test the embedded software in these IoT devices in terms of data flow, control flow, and functionalities with improved performance.

Internet of things (IoT) describes an emerging trend where a large number of embedded devices (things) are connected to the internet to participate in automating activities that create compounded value for the end consumers as well as for the enterprises. One of the greatest concerns in IoT is security, and how software engineers address it will play a deeper role. As devices interact with each other, businesses need to be able to securely handle the data deluge. With focused approach, it is possible to minimize the vulnerabilities and risks exposed to the devices and networks. Adopting security-induced software development lifecycle (SDL) is one of the major steps in identifying and minimizing the zero-day vulnerabilities and hence to secure the IoT applications and devices. This chapter focuses best practices for adopting security into the software development process with the help of two approaches: cryptographic and machine learning techniques to integrate secure coding and security testing ingrained as part of software development lifecycle.

The idea of an intelligent, independent learning machine has fascinated humans for decades. The philosophy behind machine learning is to automate the creation of analytical models in order to enable algorithms to learn continuously with the help of available data. Since IoT will be among the major sources of new data, data science will make a great contribution to make IoT applications more intelligent. Machine learning can be applied in cases where the desired outcome is known (guided learning) or the data is not known beforehand (unguided learning) or the learning is the result of interaction between a model and the environment (reinforcement learning). This chapter answers the questions: How could machine learning algorithms be applied to IoT smart data? What is the taxonomy of machine learning algorithms that can be adopted in IoT? And what are IoT data characteristics in real-world which requires data analytics?

This chapter addresses the key role of machine learning and artificial intelligence for various applications of the internet of things. The following are the most significant applications of IoT: (1) manufacturing industry: automation of industries is on the rise; there is an urge for analyzing the energy in the process industry; (2) anomaly detection: to detect the existing fault and abnormality in functioning by using ML algorithms thereby avoiding the adverse effect during its operation; (3) smart campus: in-order to efficiently handle the energy in buildings, smart campus systems are developed; (4) improving product decisions: with the help of the predictive analytics system products are designed and developed based on the user's requirements and usability; (5) healthcare industry: IoT with machine learning provides numerous ways for the betterment of the human wellbeing. In this chapter, the most predominant approaches to machine learning that can be useful in the IoT applications to achieve a significant set of outcomes will be discussed.

Internet of things has gathered significance within the latest technology domain and trends. As a result, it offers greater ways of accessing data and utilizing intelligent systems. IoT applications are developed for specific scenarios (i.e., smart home, smart transportation, smart agriculture, e-health, etc.). Such IoT applications are inefficient for sharing data and knowledge through services. This results in an inefficient exploitation of different IoT service applications. Social internet of things (SIoT) has efficient and effective ways to support these kinds of services. A concept of social internet of things has been proposed in this chapter in order to support efficient data sharing. This chapter explores related work and literature study on social internet of things, concentrates on mapping IoT with SIoT, and describes a possible architecture for SIoT, components, layers and processes of SIoT. It also illustrates applications, where SIoT can be used, and at the end, the authors provide a few challenges related to SIoT.

Preface

Generally, Software Engineering principles and practices make the Software development industries to follow Software Development Life Cycle (SDLC) process models for product development. Nowadays, the application of IoT in developing mission critical systems and real-world applications have become mandatory which thus makes the software development process to be revisited.

As IoT is a combination of different types of devices connected via internet with cloud as its storage media and other supporting software components are being the part of the entire application, the SDLC process needs to be now provided with more precise development aspects of IoT. In addition, as management of huge amount of data using big data analytics and security as a key issue for successful IoT implementation, the researchers and industries are more concerned about how to bring them as part of the software development process. Also, apart from traditional testing methods, some of the testing methods such as penetration testing and vulnerability testing have to be now analyzed to identify some of the security breaches in the software of IoT.

This book is a complete novel idea on how to bring forth the application of Software Engineering principles and practices in IoT based software development. Taking into consideration of all these insights, this book brings the content apart from general Software Engineering principles, the phases of SDLC such as requirements elicitation, design documents preparation and testing equipped with IoT based inclusions.

Hence, this book elaborates the Software development life cycle framework, methodology, test driven development, testing and quality assurance models to achieve secured software development. It will also provide in depth details on security engineering activities, security assurance activities with organizational and project management activities.

The objective of this book is to provide complete SDLC best practices and guidelines for IoT environment and provides security and quality assurance activities in software development.

Impact:

- Process framework for IoT based software development
- Application of UML in IoT design document generation
- Test cases from requirements and use case-based analysis
- Testing best practices for IoT
- Secure software development framework along with methodology

The book is also going to provide details of vulnerability, complexity and robustness management and risk-driven security metrics methodologies and tools that help developers to achieve adequate security, trust, dependability and privacy goals in various development environments coming under IoT.

As security, trust, dependability and privacy are issues of IoT, they must be given higher level of importance over the whole life-cycle of the system and software development. In the case of Agile based software development, the activities starting from gathering requirements to deploying the system and service in practice for IoT have a lot number of challenges and issues. This book is going to present several ideas related to these issues and challenges and best practices that can be adopted for IoT.

Further, this book is organized as follows:

Chapter 1 provides a new process model for IoT based software development that has a greater relevance in order to reduce the associated risk. To exploit the capability of IoT driven innovations which enable organizations to enhance their revenue streams, reduce time to market while increasing business agility, organizations need to determine how best to employ IoT enabled business models that promote sustainable competitive advantage.

Chapter 2 summarizes the concepts and challenges of DevOps in IoT, DevSecOps in IoT, Integrating security into IoT, Machine Learning and AI in IoT. DevOps is a software engineering culture and practice that aims at unifying software development (Dev) and software operation (Ops). The main characteristic of DevOps is the automation and monitoring at all steps of software construction, from integration, testing, releasing to deployment and infrastructure management. DevSecOps is a practice of integrating security into every aspect of an application lifecycle from design to development.

Chapter 3 covers the modelling of IoT systems using UML diagrams. Starting with the architectural design of any IoT system to behavioural aspects is covered in this chapter using a case study of IoT based Remote Patient Health Monitoring System .The diagrams shown in this chapter are the sample diagrams for understanding IoT based complex systems. The chapter focuses on the work carried out by Franco Zambonelli in the context of developing an abstract model of an IoT system using software engineering concepts. The chapter also focuses on the pioneer work carried by J.F. Peters in Intelligent system design patterns for robotic devices using pattern classification.

Chapter 4 gives the introduction about the web based application development based on Internet of Things (IoT). The major objective of this chapter is to discuss and resolve the challenges in IoT to automate the real time problems.

Chapter 5 will discuss about the necessity for IoT testing in terms of various criteria of identifying and fixing the problems in the IoT systems. In addition, this chapter examines the core components to be focused on IoT testing and testing scope based on IoT device classification. It also elaborates the various types of testing applied on healthcare IoT applications and finally, this chapter summarizes the various challenges faced during IoT testing.

Chapter 6 provides challenges in testing a connected device within a full IoT environment which will have more devices with varied functionalities and data processing. These challenges have made a new way of testing to be done so that, the test cases will be more efficient in revealing the errors in the software. In this chapter, UML Use Case diagram-based test cases generation for an IoT environment is explained in detail. Also, a real time case study IoT application is taken to showcase how this approach helps in generating the test cases to test the embedded software in these IoT devices in terms of data flow, control flow and functionalities with improved performance.

Chapter 7 focuses on the best practices for adopting security into the software development process with the help of two approaches: Cryptographic and Machine Learning techniques to integrate secure coding and security testing ingrained as part of Software Development Lifecycle.

Chapter 8 is to answer how could machine learning algorithms be applied to IoT smart data? What is the taxonomy of machine learning algorithms that can be adopted in IoT? And what are IoT data characteristics in real-world which requires data analytics?

In Chapter 9, the most predominant approaches to machine learning that can be useful in the IoT applications to achieve a significant set of outcomes are discussed.

In Chapter 10, the author proposes a concept of Social Internet of Things (SIoT) in order to support efficient data sharing. This chapter explores related work, literature review on Social Internet of Things, mapping IoT with SIoT and a possible architecture for SIoT, components and layers of SIoT and processes of SIoT. It also illustrates applications where SIoT can be used and at the end the author provides few challenges related to this concept.

Chapter 11 provides an IoT application being developed as per the discussions done. A novel mobile application which could be used by the user to know about the present and future pollution level is developed using a progressive web application development environment. This mobile application uses the Google Map to get the location information which is integrated with this model and helps the user to predict the hazardous level of the pollutants in a particular location.

Acknowledgment

At the foremost, I heart fully thank the Lord Almighty who lifts up the poor from the dust and the needy from their misery for providing me the strength, knowledge and wisdom in completing this book project.

I sincerely acknowledge my host Institute Thiagarajar College of Engineering, Madurai, Tamil Nadu, India for providing me all the support in making this book project a successful one.

I thank IGI Global Publishers, USA for providing an opportunity to edit a book on "Integrating the Internet of Things in Software Engineering best practices" which is a collection of ideas from various authors working in this area.

I thank the authors for their timeless efforts and prompt submissions to make this book a usefil one.

I solemnly thank all the Editorial Board Members, Reviewers and institute colleagues for providing all the support for the successful completion and publishing of this book.

This acknowledgement will not be fulfilled without my family; I thank my husband Mr. J. Abraham and my son A. Pradeep Reynold for rendering their support and patience in the success of this project.

Chapter 1
A New Process Model for IoT-Based Software Engineering

K. S. Jasmine
R. V. College of Engineering, India

ABSTRACT

Internet of things (IoT) is a new trending paradigm for advanced technological development which has drawn significant research attention in the recent years. IoT comprises intelligent communicating "things," putting a big challenge on ensuring security, reliability, efficiency, and safety in their interaction. Staying connected always, constant evolution, and grappling with multiple life cycles are the major factors of concern. In this context, a new process model for IoT-based software development has a greater relevance in order to reduce the associated risk. To exploit the capability of IoT-driven innovations which enable organizations to enhance their revenue streams, reduce time to market while increasing business agility, organizations need to determine how best to employ IoT-enabled business models that promote sustainable competitive advantage.

DOI: 10.4018/978-1-5225-7790-4.ch001

INTRODUCTION

In today's rapidly changing business environment, it is predicted that 'Internet of Things' will become the backbone of future customer value and dedicated IoT platforms will have a significant impact and security will remain a key concern. Organizations are grabbing opportunities around the intelligent IoT products to create novel services and diverse set of IoT- enabled business models which have thecapabilities for remote product management, monitoring, and control by creating operational efficiencies and engaging customers through innovative path.

In the IoT based Business context, the following are the identified challenges:

- Market need to follow new strategic ways of interacting with customers through new interfaces like smart home speakers, smart watches, wearable devices etc
- Need of integration softwares that combines diverse set of IoT-enabled business assets into cohesive business process
- Increasing need of IoT platform to have support services for data processing and analysis at both edge and core of the network with a sustainable model
- Accelerated IoT platform adoption from public cloud providers due to developer requirements for low adoption costs, quick deployment, global reach, easy integration with minimal maintenance burden
- Key challenging Process areas to focus are remote machine setup, material supply, product pricing, information reporting and Quality control through corrective and predictive maintenance

The forthcoming sections investigates the feasibility of IoT based Software engineering solutions on how organizations can deliver high business value through technology and operations strategy engagements at the same time can generate return on Investment (ROI) by effectively utilizing the possibilities of IoT in business.

BACKGROUND

IoT is defined as a new paradigm which can make adifference to organizations' businessvalue bybuilding theright infrastructure, using existing devices and services in new ways, andincorporating the right technology ("13. How the Internet of Things Is Improving Transportation and Logistics",2015).Social networks can play a major role in experiencesharingand personalized insights with great possibility of integration for business-centric applications. The integration and interoperability can be enhanced by realtime analytics, business intelligence and agent-based autonomous services (Uckelmann et al.,2011).

It is essential to transform the considerable volumes of data into useful knowledge in order achieve the required gain through IoT applications (.Zancul et al., 2016).. In this context, Semantic modeling for the IoTusing ontology based knowledge representation to resolve the associated problems in the distributed environment, thereby supporting service discovery, testing and dynamic composition of the "Things" is apromising solution. (Wang et al.,2012). With the advent of cloud computing, Sensing as a service on cloud can be a favorable service solution which can be additionally facilitated with applications like Augmented Reality, Virtual Reality and Environment monitoring (Rao et al.,2012).

Even though IoT is the technology going to rule the future, It is necessary to a have a planned strategy for organizations before launching an IoT initiative, along with adaptation to new change in culture. Otherwise it will lead to many risks such as overspending, data security and privacy threats, limiting ROI from IoT technologies. The issue of data ownership right is also a challenge of the future. Security aspects related to IoT and Cloud technologies have to be addressed while enjoying the integration benefits of Internet of Things and Cloud Computing technologies such as lack of equipment standardization, possibility of data leakage, need to protect data from corruption/interference, potential risk associated with unsecured devices connected to the Internet, Trustworthiness of cloud services etc.

CAPTURE VALUE FROM THE INTERNET OF THINGS

Internet of Things (IoT)

Internet of Things is the concept of empowering the computers to interact with its environment and manage intelligently. It is sometimes referred to as the Internet of Everything (IoE). The 'Things' in IoT can be any device with built-in-sensors which has the capability of gathering and exchanging data over a network through which it is connected without human intervention. The embedded technology facilitates these 'things' to interact with surrounding environment in an efficient manner based on their change in states.

Some of the common sensors used are Proximity sensors, Pressure sensors, temperature sensors, gas sensors, image sensors, acoustic sensors, IR sensors, optical sensors, humidity sensors, Smoke sensors, Level sensors, motion detection sensors etc.

IoT embraces technologies such as smart grids, smart homes, intelligent transportation and smart cities etc

Layers of IoT

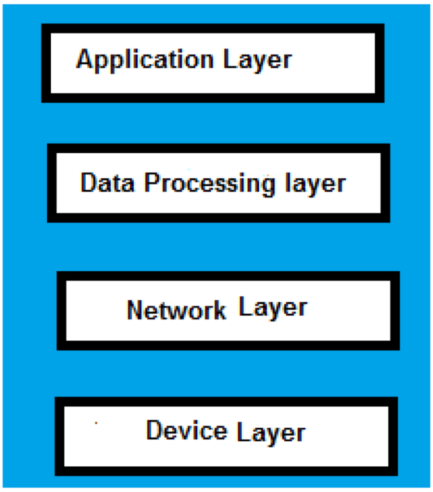
The Internet of Things (IoT) is a collection of devices, sensors, protocols and network connectivity to collect and process datain order to effectively convert data into information.

There exists mainly four layers.

- **Device Layer:** Consists of various devices like wearables, smart meters, smart phones etc
- **Data Processing Layer:** Data received from various devices are processed to convert into information and insights
- **Network Layer:** Responsible for transmitting data to the application layer through various network technologies
- **Application Layer:** This layer constitutes the front end of the whole IoT architecture.

Figure 1. Smart 'things' in IoT- a representation

Figure 2. Layers of IoT applications



IOT BASED SOFTWARE DEVELOPMENT LIFE CYCLE(SDLC)

In this section, the critical phases of process model which are having crucial impact in the overall performance of IoT applications is discussed. Since the common programming languages like C, java, Python etc are suitable for the development of any IoT applications, coding phase is not discussed below.

Phase I: Self Adaptive Requirement Phase

'Things' within IoT are operating in a rapidly changing environment. In such environment, autonomous self-adaptation driven by requirements satisfaction is necessary among these interactive objects. Similar to the various phases in traditional SDLC, in the IoT based Software Engineering also one can follow the same with dynamism incorporated into these phases because movement, location and communication are the driving factors in the self-adaptation among these 'things'. The major challenges are i) rapidly changing operating environment at run time ii) heterogeneity of devices iii) lack of human interference in managing these devices. In this scenario, sensing in the environment, detecting changes and adjust or react accordingly is very much essential (Sawyeret al., 2010). The decision is based on their own requirements satisfaction to meet a common goal by 'learning' from each other to best adapt. But identifying who is 'intelligent thing' who can do the decision making and who are 'reactive things', just follow the order by the intelligent things are challenging.

Phase II: Model Based Design Phase

Due to the heterogeneity of interacting 'things ', the design of software systems for Internet of Things (IoT) is challenging. It is difficult to build a complete static design solution for the application's requirements in prior. A run time design model based on the application scenario which ensures synchronization among model transitions and transformations of change in states of objects/ 'things' will be a promising solution. This approach can help to tackle the number of failure modes and inadvertent consequences and issues because if we correct errors at the design, it will be cost effective than if we do the same in later stages. Requirements coverage and continuous customer validation should be the focus in this stage with the help of small scale simulation models.

The change in state of the 'things' can be modeled based on monitoring environmental/ quality control parameters and initiate the corrective action, like generating an event to raise the alarm if it exceeds the threshold value set for a specific parameter. For Example, in the case of an automatic car, raising an alarm or display of red light when the distance between an object and the car is less than the minimum value set while taking a deviation or reversing.

Phase III: Model Based Testing Phase

Model-based testing is a testing approach which can be used in IoT applications. But deciding the scalable infrastructure, architecture for IoT based applications is crucial. The adoption of Model based testing requires a different culture and mindset because it is crucial to make modifications to front end application code to improve the testability while creating models for the system under test. Here model itself will act as a test plan.

Testing Approaches in IoT

In IoT applications, testing non-functional requirements are more important. The following list shows the important ones.

- **Usability:** In terms of displaying data, processing data, pushing notifications, error messages, warnings etc
- **Security:** Since IoT is data centric it should be ensured that data transferred among devices is protected or encrypted
- **Availability:** maintaining robustness in the device connectivity ensures availability of the communicating devices

- **Performance:** Scalability and system performance in the case of increasing users, devices in the connectivity network is challenging.

IoT testing approach is user centric and is challenging due to the complexity of heterogeneous devices, protocols, hardware, operation systems, firmware etc

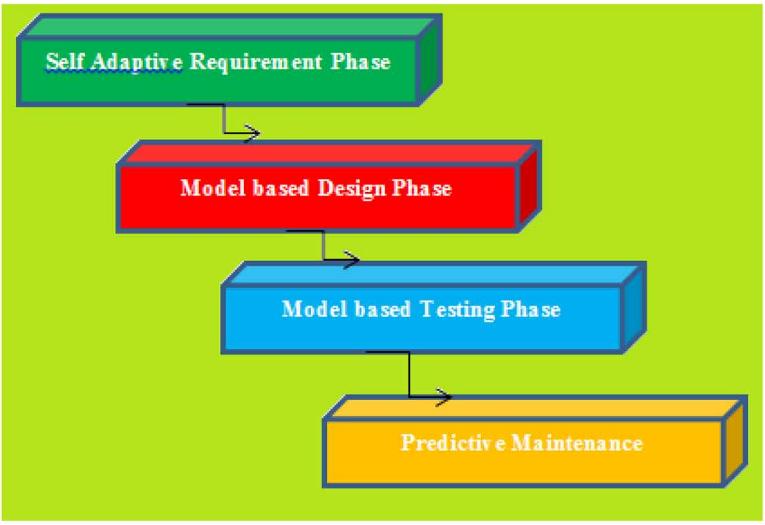
Phase IV: Predictive Maintenance

Predictive maintenance is very essential in reducing costs by predicting equipment faults before they occur. With this, there will be high product quality, reduced downtime, increased reliability and improved customer satisfaction, and optimized resource management in real –time. Collecting usage data from various connected devices and identifying their pitfalls dynamically and updating is the challenging task in this phase.

Theoretical Evidence: Guiding the Process of Translating Proposed Model Into Practice

Process models help to implement the theoretical approach into practice in the context of real world Scenario. There are many studies which outline the various phases of the research outcomes into practice process. The knowledge gained during the study can be directly transferred from the producers to users. The SDLC Model

Figure 3. Critical phases of IoT SDLC



proposed here support process flow and emphasizes on various organizational context, thereby focusing on dissemination of the information and the knowledge gained to various implementation aspects. It is the practitioners discretion to plan the strategies for implementation from the guidelines provided by or the insights gained from the process model .

Implementation Determinants

The implementation plan can be based on the following characteristics:

1. **Characteristics of the Evidence:** Related to innovative ideas, nature of study conducted, intervention characteristics
2. **Characteristics of the Strategy Adopters:** Related to personal characteristics, organization factors
3. **Characteristics of the End Users:** Factors related to resources available (Human resources, system resources, infrastructure resources etc
4. **Characteristics of the Environment/Context:** Factors related to economic, administrative and organizational context

To conclude, organizational theories concerning organizational culture, leadership abilities, market advocacy address the influence of the context on implementation outcomes.

IDENTIFYING BUSINESS NEEDS THROUGH IOT

It is predicted that future business opportunities mainly based on IoT . Value propositions of Business will be depends on the factors such as cost reduction, low environmental impacts, improved functionality, better safety, and quality.

Gartner predicts that by 2020 a staggering 26 billion objects will be IoT connected or IP-enabled, interactive, and ‘smart’. In fact, this growing global market could hit a worth of $7.1 trillion by that date.

Two Approaches to IoT Based Business

There are two approaches to IoT based Business. One is aimed at customers which enable the them to be smart to communicate with smart objects or ‘things’ around them such as Television, refrigerators, fans, toasters, kettles etc.This approach will concern manufactures of these objects to take decision on measuring the right data at right time and its efficient usage for decision making at the same time, making these sensors virtually invisible and the second focuses on business side by how efficiently IoT data can be understood and optimized for business processes, communication and analyzing the customer behavior.

In this scenario, protecting our personal data will be a big threat by preserving its privacy. In the competitive edge, and using IoT to gain its advantage at the fullest, there is a need to figure out the quantity and meaningful uses of real-time quality data collated from various sources and how best it can be analyzed for optimum results will lead the business success. IoT capacity, network traffic, and maximum secure coverage are the features that should be carefully considered and planned. System scalability can be improved with the cloud adoption. But keeping the detailed track of source of data and its security while maintaining operational scalability will be a huge challenge to ripe the benefits of IoT based business.

IoT impacts on business process can be categorized into three:

1. Business to Consumer (B2C): e.g. connected people, connectedhome, connected car
2. Business to Business (B2B): e.g. connected buildings,connected industry
3. Business to Business to Consumer (B2B2C): e.g. smart cities, smart grid

IOT IN VARIOUS DOMAINS

It is a known fact that IoT can provide powerful solutions in various domains of our day to day work. A few of them are listed below:

Energy Efficiency

Energy efficient devices are very important to protect our environment. IoT plays a crucial role in energy efficient innovations. IoT facilitates it in monitoring level, usage level and the distribution level like Smart Meters; Smart Grids are used to monitor energy consumption which helps to have system stability and performance. It is predicted that, the number of connected objects by 2020 will be 26 billion units(Middleton, P. et al.,2014).

Healthcare

The impact of Internet of things in healthcare is very demanding, with applications facilitates the patients to monitor their health in regular intervals and take medication accordingly.

The usages of Biometric sensors, fitness wearables, Smart watches etc have changed the frequency of health monitoring.

While there are many benefits of IoT in healthcare, data security and IoT device management are challenging.

Education

In streamlining the education sector, IoT plays a vital role. It can facilitate personalized teaching-learning process through smart communication and collaboration. Devices like wristbands, smart pens which can scan and send the text into your smart phones etc makes the education field more tech-savvy. So IoT acts a mediator to fill the gap in the education field by not only improving the quality of education but also optimizes the cost and efficiency.

Transportation and Logistics

With the advent of IoT, end to end visibility made possible in transportation and logistics businesses facilitating effective, timely decisions and to reduce delays in decision making.With barcode scanners and mobile computers and with Radio frequency identification(RFID), many transportation and logistics companies could reach nearly 100 percent shipping 99.5 percent inventory accuracy, 30 percent faster order processing and 30 percent reduction in labor costs (Middleton, P. et al.,2014). IoT has brought rapid change in the transportation sector with the provision of self-driving cars, automated traffic signals that can switch automatically based on the density of traffic, Intelligent parking assistance, vehicle status indicators, temperature, pressure indicators etc.

Pollution Control

In preventing considerable contamination by detecting pollution in the air and water IoT minimizes the human intervention in environment monitoring with the help of pH sensors, Smoke sensors etc. There are systems which can automatically detect changes in weather, crops, soil etc.

Marketing

Data-driven marketing is a new terminology in digital marketing with the advent of IoT. Focusing on customer experience and customer engagement by making them ‘connected’ is the new business strategy in marketing. It requiresbetter analysis of the ‘Big data’ throws by the Web& respond to customer preferences.

With this, large majority of marketers (90%) move beyond segmentation towards one-to-one personalization in a real-time context, for 38% of respondents the major challenge is improving both customer acquisition and customer retention and 78% of marketers use data systematically (Rifkin,2014). Also integrated marketing cloud approach seems still better platform for future marketing.

Government Sector

IoT supports the development of smart nations and smart cities with the enhancement of sectors like defense, healthcare, energy, transportation, education etc. It requires detailed and accurate analysis of real-time data collected in these areas and produce accurate information.

CONCLUSION AND FUTURE WORK

The chapter addresses the business needs in IoT and the associated challenges and proposes the guidelines and recommendations which can be incorporated into the SDLC of IoT based process model which emphasizes on the dynamism incorporated phases. The study lacks in providing evidence in the practical aspects which can be easily done with a case study in the industrial context. Future work can be focused in this direction.

REFERENCES

Ali, R., Dalpiaz, F., & Giorgini, P. (2010). A goal-based framework for contextual requirements modeling and analysis. *Requirements Engineering*, *15*(4), 439–458. doi:10.100700766-010-0110-z

Atzori, L., Iera, A., & Morabito, G. (2017). Understanding the Internet of Things: Definition, potentials, and societal role of a fast evolving paradigm. *Ad Hoc Networks*, *56*, 122–140. doi:10.1016/j.adhoc.2016.12.004

Casagras, R. (2011). *RFID and the inclusive model for the Internet of Things report*. Academic Press.

Hackbarth, R., Mockus, A., Palframan, J., & Sethi, R. (2016). Improving Software Quality as Customers Perceive It. *IEEE Software*, *33*(4), 40–45. doi:10.1109/MS.2015.76

He & Li. (2014). Internet of Things in Industries: A Survey. *IEEE Transactions on Industrial Informatics, 10*(4), 2232-2243.

Henze, M., Hermerschmidt, L., Kerpen, D., Häußling, R., Rumpe, B., & Wehrle, K. (2016). A Comprehensive approach to privacy in the cloud-based Internet of Things. *Future Generation Computer Systems*, *56*, 701–718. doi:10.1016/j.future.2015.09.016

How the Internet of Things Is Improving Transportation and Logistics. (2015). *Transportation News*.

Internet of Things Research Study Report. (2015). Hewlett Packard Enterprise. Retrieved from www8.hp.com/h20195,/V2/GetPDF.aspx/4AA5-4759 ENW.pdf

Key findings of the Teradata 2015 Global Data-Driven Marketing Survey. (2015). Retrieved from https://www.i-scoop.eu/data-driven-marketing-the-state-benefits-and-drivers-of-data-marketing/

Middleton, P. (2014). *Forecast: Internet of Things, Endpoints and Associated Services, Worldwide, 2014 (G00270264)*. Gartner Database.

Rao, Saluia, Sharma, Mittal, & Sharma. (2012). Cloud computing for Internet of Things & sensing based applications. *Proceedings of Sixth International Conference on Sensing Technology,* 374-380.

Rifkin, J. (2014). *The Zero Marginal Cost Society: The Internet of Things, the Collaborative Commons, and the Eclipse of Capitalism*. New York, NY: Macmillan.

Sanger, D., & Perlroth, N. (2016). A New Era of Internet Attacks Powered by Everyday Devices, *New York Times*.

Sawyer, P., Bencomo, N., Whittle, J., Letier, E., & Finkelstein, A. (2010). Requirements -Aware Systems: A Research Agenda for RE for Self-adaptive Systems. In Requirements Engineering Conference (RE), 18th IEEE International. Sydney: IEEE Computer Society.

Stankovic, J. A. (2014). Research Directions for the Internet of Things. *IEEE Internet of Things J.*, *1*(1), 3–9. doi:10.1109/JIOT.2014.2312291

Stergiou. Psannis, Kim, & Gupta. (2018). Secure integration of IoT and Cloud Computing. Future Generation Computer Systems, 78(3), 964-975.

Tecnalia, Inspearit, Favaro, & Taneja. (2017). Software Engineering For Internet of Things. IEEE Software, 24-28.

T¨onjes, R., Reetz, E. S., Moessner, K., & Barnaghi, P. M. (2012). A test-driven approach for life cycle management of internet of things enabled services. Proceedings of Future Network and Mobile Summit, 1–8.

Uckelmann, D., Harrison, M., & Michahelles, F. (2011). *An Architectural Approach Towards the Future Internet of Things*. Architecting the Internet of Things. doi:10.1007/978-3-642-19157-2

Wang, De, Toenjes, Reetz, & Moessner. (2012). A comprehensive ontology for knowledge representation in the internet of things. In *Proceedings of the 11th International Conference on Trust, Security and Privacy in Computing and Communications*, (pp. 1793–1798). IEEE.

Watson Internet of Things. (2016). Retrieved from IBMwww.ibm.com/internet-ofthings

Zancul, Takey, Barquet, Kuwabara, Miguel, & Rozenfield. (2016). Business process support for IoT based product-service systems (PSS). *Business Process Management Journal*, *22*(2), 305–323. doi:10.1108/BPMJ-05-2015-0078

Zancul, E., Takey, S. M., Barquet, A. P. B., Kuwabara, L. H., Cauchick Miguel, P. A., & Rozenfeld, H. (2016). Business process support for IoT based product-service systems (PSS). *Business Process Management Journal*, *22*(2), 305–323. doi:10.1108/BPMJ-05-2015-0078

Chapter 2
Current Trends in Integrating the Internet of Things Into Software Engineering Practices

S. Kavitha
Velammal College of Engineering and Technology, India

J. V. Anchitaalagammai
Velammal College of Engineering and Technology, India

S. Nirmala
Velammal College of Engineering and Technology, India

S. Murali
Velammal College of Engineering and Technology, India

ABSTRACT

The chapter summarizes the concepts and challenges of DevOps in IoT, DevSecOps in IoT, integrating security into IoT, machine learning and AI in IoT of software engineering practices. DevOps is a software engineering culture and practice that aims at unifying software development (Dev) and software operation (Ops). The main characteristic of DevOps is the automation and monitoring at all steps of software construction, from integration, testing, releasing to deployment and infrastructure management. DevSecOps is a practice of integrating security into every aspect of an application lifecycle from design to development.

DOI: 10.4018/978-1-5225-7790-4.ch002

INTRODUCTION

IoT is on the edge of a huge rise in the market, where the number of machines that are connected to the internet has increased by three times and present day over 12 billion of devices are connected to the internet according to Cisco . Due to a sudden change in the using, the IoT technologies in both home and office leads to the impact in the global market, suppose if the price of the device is reduced and performance is increased then the IoT adoption increases. According to the survey conducted by the Economist Intelligence unit [1], for conducting the business based on the IoT. They found that 46% of respondents told the existing business model will change due to IoT, 30% respondents told IoT will unlock new revenue opportunities from the existing product or services and 29% told that IoT will inspire new business process. Now the concepts and challenges of DevOps in IoT, DevSecOps in IoT, Integrating security into IoT, Machine Learning and AI in IoT of software engineering practices will discussed in detail.

- **DevOps** (a clipped compound of “development” and “operations”) is a culture, movement or practice that emphasizes the collaboration and communication of both software developers and other IT professionals while automating the process of software delivery and infrastructure changes. It aims to establish a culture and environment where building, testing and releasing software can happen rapidly, automatically and more reliably.
- **DevSecOps** movement builds on the idea that everyone is responsible for security and inherently accepts that retrofitting current solutions is no longer sufficient as hackers have changed the rules and also enjoy the advantage of being on the offensive.
- **Machine learning** is a field of computer science that often uses statistical techniques to give computers the ability to “learn” (i.e., progressively improve performance on a specific task) with data, without being explicitly programmed.[1]
- **Artificial intelligence** (**AI**, also **machine intelligence**, **MI**) is intelligence demonstrated by machines, in contrast to the **natural intelligence** (**NI**) displayed by humans and other animals. In computer science AI research is defined as the study of “intelligent agents”: any device that perceives its environment and takes actions that maximize its chance of successfully achieving its goals.[2]

DevOps

DevOps is a set of practices intended to reduce the time between committing a change to a system and the change being placed into normal production, while ensuring high quality.[3]

The history of DevOps,at 2008 Agile Toronto conference, Andrew Shafer and Patrick Debois introduced the term in their talk on “Agile Infrastructure”.[4] From 2009, the DevOps term has been steadily promoted and brought into more mainstream usage through a series of “devopsdays”,[5] which started in Belgium and has now spread to other countries

- **DevOps** (a clipped compound of “development” and “operations”) is a software engineering culture and practice that aims at unifying software development (Dev) and software operation (Ops). The main characteristic of the DevOps movement is to strongly advocate automation and monitoring at all steps of software construction, from integration, testing, releasing to deployment and infrastructure management. DevOps aims at shorter development cycles, increased deployment frequency, and more dependable releases, in close alignment with business objectives.

DevOps is not a new concept, but the calls for such a cultural shift and approach to application development have grown louder. Particularly in the security industry, experts now recognize the need to shift toward an approach where automation and orchestration are at the foundation of the development and deployment processes. This new approach is called DevSecOps.

DevOps Toolchain

As DevOps is intended to be a cross-functional mode of working, rather than a single DevOps tool there are sets (or “toolchains”) of multiple tools.[6] Such DevOps tools are expected to fit into one or more of these categories, reflective of key aspects of the development and delivery process:

1. **Code:** Code development and review, source code management tools, code merging
2. **Build:** Continuous Integration tools, build status
3. **Test:** Continuous Testing tools that provide feedback on business risks
4. **Package:** Artifact Repository, application pre-deployment staging
5. **Release:** Change management, release approvals, release automation

6. **Configure:** Infrastructure configuration and management, Infrastructure as Code tools
7. **Monitor:** Applications Performance Monitoring, end–user experience

Note that there exist different interpretations of the DevOps toolchain (e.g. Plan, Create, Verify, Package, Release, Configure, and Monitor).

Relationship to Other Approaches

Agile

The need for DevOps arose from the increasing success of agile software development, as that led to organizations wanting to release their software faster and more frequently. As they sought to overcome the strain this put on their release management processes, they had to adopt patterns such as application release automation, continuous integration tools, and continuous delivery.[7][8]

Continuous Delivery

Continuous delivery and DevOps have common goals and are often used in conjunction, but there are subtle differences.[9][10]

While continuous delivery is focused on automating the processes in software delivery, DevOps also focuses on the organization change to support great collaboration between the many functions involved.[9]

DevOps and continuous delivery share a common background in agile methods and lean thinking: small and frequent changes with focused value to the end customer.[11] They are well communicated and collaborated internally, thus helping achieve faster time to market, with reduced risks.[7][12]

DataOps

The application of continuous delivery and DevOps to data analytics has been termed DataOps. DataOps seeks to integrate data engineering, data integration, data quality, data security, and data privacy with operations.[13] It applies principles from DevOps, Agile Development and the statistical process control, used in lean manufacturing, to improve the cycle time of extracting value from data analytics.[14]

SciOps (Scientific DevOps)

Scientific DevOps refers to DevOps practices applied in the context of scientific computing.[15] While the tools and methodologies are the same, the goals are different: DevOps delivers a software product, while SciOps delivers scientific insights. An alternative interpretation of the term is as a specialization of DevOps.

ResOps (Research DevOps)

Research DevOps groups together all the tools and techniques used to deliver and support research operations in cloud environments (i.e., data transfer or data storage) [16]. In addition, ResOps also focuses on the optimisation of research workloads for clouds, defining two main approaches: legacy, where on-prem infrastructure is replicated in the cloud environment, and cloud-first, where cloud computing paradigms are fully adopted when designing the workloads. Both approaches have their own advantages and disadvantages, and impact the efficiency of the designed solution.

Goals

The goals of DevOps span the entire delivery pipeline. They include:

- Improved deployment frequency;
- Faster time to market;
- Lower failure rate of new releases;
- Shortened lead time between fixes;
- Faster mean time to recovery (in the event of a new release crashing or otherwise disabling the current system).

Simple processes become increasingly programmable and dynamic, using a DevOps approach.[17] DevOps aims to maximize the predictability, efficiency, security, and maintainability of operational processes. Very often, automation supports this objective.

DevOps integration targets product delivery, continuous testing, quality testing, feature development, and maintenance releases in order to improve reliability and security and provide faster development and deployment cycles. Many of the ideas (and people) involved in DevOps came from the enterprise systems management and agile software development movements.[18]

Views on the Benefits Claimed for DevOps

Companies that practice DevOps have reported significant benefits, including: significantly shorter time to market, improved customer satisfaction, better product quality, more reliable releases, improved productivity and efficiency, and the increased ability to build the right product by fast experimentation.[7]

However, a study released in January 2017 by F5 of almost 2,200 IT executives and industry professionals found that only one in five surveyed think DevOps had a strategic impact on their organization despite rise in usage. The same study found that only 17% identified DevOps as key, well below software as a service (42%), big data (41%) and public cloud infrastructure as a service (39%).[19]

Cultural Change

DevOps is more than just a tool or a process change; it inherently requires an organizational culture shift. This cultural change is especially difficult, because of the conflicting nature of departmental roles:

- **Operations:** Seeks organizational stability
- **Developers:** Seek change
- **Testers:** Seek risk reduction[20]

Getting these groups to work cohesively is a critical challenge in enterprise DevOps adoption.[21][22]

DevOps as a Job Title

While DevOps describes an approach to work rather than a distinct role (like system administrator), job advertisements are increasingly using terms like "*DevOps Engineer*".[23][24]

While DevOps reflects complex topics, the DevOps community uses analogies to communicate important concepts, much like "The Cathedral and the Bazaar" from the open source community.[25]

- **Cattle not pets:** The paradigm of disposable server infrastructure.
- **10 Deployments per day:** The story of Flickr adopting DevOps.

Building a DevOps Culture

DevOps principles demand strong interdepartmental communication—team-building and other employee engagement activities are often used—to create an environment that fosters this communication and cultural change, within an organization. Team–building activities can include board games, trust activities, and employee engagement seminars.

Deployment

Companies with very frequent releases may require a DevOps awareness or orientation program. For example, the company that operates the image hosting website Flickr developed a DevOps approach, to support a business requirement of ten deployments per day; this daily deployment cycle would be much higher at organizations producing multi-focus or multi-function applications. This is referred to as continuous deployment or continuous delivery and has been associated with the lean startup methodology

Architecturally Significant Requirements

To practice DevOps effectively, software applications have to meet a set of architecturally significant requirements (ASRs), such as: deployability, modifiability, testability, and monitorability. These ASRs require a high priority and cannot be traded off lightly.

Although in principle it is possible to practice DevOps with any architectural style, the microservices architectural style is becoming the standard for building continuously deployed systems. Because the size of each service is small, it allows the architecture of an individual service to emerge through continuous refactoring, hence reducing the need for a big upfront design and allows for releasing the software early and continuously.

Scope of Adoption

Some articles in the DevOps literature assume, or recommend, significant participation in DevOps initiatives from outside an organization's IT department, e.g.: "DevOps is just the agile principle, taken to the full enterprise.

A survey published in January 2016 by the SaaS cloud-computing company RightScale, DevOps adoption increased from 66 percent in 2015 to 74 percent in 2016. And among larger enterprise organizations, DevOps adoption is even higher — 81 percent.

Adoption of DevOps is being driven by many factors — including:

1. Use of agile and other development processes and methods;
2. Demand for an increased rate of production releases — from application and business unit stakeholders;
3. Wide availability of virtualized[and cloud infrastructure — from internal and external providers;
4. Increased usage of data center automation and configuration management tools;
5. Increased focus on test automation and continuous integration methods;
6. A critical mass of publicly–available best practices.

DevOps Transformation

DevOps transformation is the process of transforming and adapting a software development methodology in accordance with agile development methods and extending this across the full organisation value stream

DevSecOps

The DevSecOps movement builds on the idea that everyone is responsible for security and inherently accepts that retrofitting current solutions is no longer sufficient as hackers have changed the rules and also enjoy the advantage of being on the offensive.

This DevSecOps movement is due to a massive shift toward the cloud and the rise of both virtualization and containerization. The combination of new technologies with a cultural shift toward embracing these advances will eventually make DevSecOps the common thread in every security approach in the near future.

Automation is a key component of the movement; attempts to keep up with security manually no longer work. Every advance in technology and products opens up more possibilities for hackers to exploit.

How DevSecOps Improves Security

There are many reasons to shift to a DevSecOps approach but the most obvious is to slow the efforts of hackers. In daily security battles hackers have three key advantages:

1. They take a continuous approach in their efforts
2. They only have to get in once
3. They can be as aggressive as they wish in their attacks

All of these realities require fighting fire with fire; acting exactly like a hacker is the best way to stop their onslaught.

To start adopting a DevSecOps approach means implementing automated sources to scan source code and all libraries up and down the stack in your organization, not just using point solutions. It also means integrating security tools into a common platform via APIs. Everyone in the organization should be empowered to recognize that security is part of their responsibility.

From a macro level, adopting a DevSecOps approach flips security from a defensive to an offensive posture that is both automated and constant -- mimicking the tactics of hackers.

Hackers have long enjoyed the advantages of speed, automation, aggression and relentlessness. Adopting these characteristics in organizational security is the only way to combat their attacks. DevSecOps as a movement will take time and begins with communication to the entire organization; everyone is responsible for security.

Integrating Security Into IoT

Internet of Things (IoT) systems have rapidly assumed important roles in daily life by providing new capabilities to streamline diverse tasks. IoT catalyzes new capabilities and creates opportunities for increased productivity and societal benefits.

As the IoT ecosystem unfolds, this wide range of functions and components will result in significant privacy and security challenges that should be addressed. Large-scale, pervasive networks that collect data about the world around us and new predictive technologies and algorithms enable innovative IoT uses. Yet, the data collected from IoT devices and algorithms that use this data for decision-making can result in unintended consequences. The ubiquity of IoT also presents new security and privacy concerns, making it important to incorporate privacy and security controls into the life cycle of IoT devices.

Certain IoT components have minimal functionality, limited computational power and storage, and low energy resources, which can sometimes make conventional security and privacy protections difficult to deploy. Such deployments may need to take advantage of new protocols and system designs that are better equipped to operate in resource-limited environments.

Policy and technical approaches to emerging IoT privacy and security challenges should continue to encourage innovation while ensuring that consumer confidence in these devices and systems is bolstered by strong privacy and security practices. The principles outlined in this statement provide an approach for addressing privacy and security challenges in the IoT ecosystem:

Support Privacy and Security Throughout the IoT Device Life Cycle

- **Address Privacy and Security Risks Throughout the IoT Device Life Cycle:** Addressing IoT privacy and security challenges from requirements specification to end-of-life, including across changes in maintenance ownership, can help prevent systemic vulnerabilities and avoid difficult retrofitting. Manufacturers and solution providers should conduct scheduled privacy and security assessments and ensure the efficacy of privacy and security measures prior to deployment and regularly over the course of a device's life span.
- **Ensure Continuous, Reliable Device Operation:** The failure of IoT systems can cause irreparable damage and have serious implications for physical safety. The pervasiveness of IoT systems in the everyday lives of consumers raises the stakes. IoT device manufacturers should aim to deploy reliable and dependable systems.
- **Provide Regular Patches, Upgrades, and Software Updates:** IoT components may ship with vulnerabilities, and new vulnerabilities may be discovered over time; manufacturers are encouraged to provide mechanisms to maintain the privacy and security of IoT components throughout their life cycle. It is similarly important to build consumer understanding and awareness on the importance of software upgrades and to encourage manufacturers to deploy patching and software updates whenever possible.
- **Consider Issues With Abandoned, Orphaned, and Legacy Components:** Manufacturers should monitor concerns related to abandoned or legacy technology that can pose privacy and security threats to existing or new components as new vulnerabilities arise.

Develop New Technologies to Support IoT Privacy and Security

- **Support Flexible Access Control:** IoT components, especially those without user interfaces, should support secure and private interactions and updates. The IoT ecosystem needs flexible approaches to access controls that foster privacy and security.
- **Leverage Advances in Cryptography and Cybersecurity:** Technically-limited IoT components may benefit from advances in "lightweight" cryptography and new encryption implementations. These options are less resource intensive and therefore more usable within the constraints imposed by many IoT components.

Protect Consumer Data

- **Address Data Ownership:** The ubiquity of IoT components means an increase in the scale of data captured, shared, collected, and analyzed. Stakeholders should plan for future challenges of data ownership as they relate to privacy, security, and intellectual property.
- **Build Consumer Awareness About Privacy and Data Sources:** As IoT permeates consumers' lives, it will be important to educate consumers on the privacy and security issues that IoT presents and on how to best protect themselves from attacks. Organizations should be transparent to consumers about how data about them is collected, used, retained, and shared.
- **Protect data integrity**: IoT components should receive, process, and create data that is accurate, consistent, and relevant for the purposes for which it was collected or produced.

Foster Cooperation among Stakeholders

- **Promote an Interdisciplinary Approach to Trust:** Hardware and software engineering, cryptography, human factors, and social science can all contribute to fostering a safe, secure, and trustworthy IoT ecosystem.
- **Encourage Coordinated Efforts Among Stakeholders:** Improved coordination between the public and private sectors can foster IoT innovation and protect privacy and security. Cooperation among governments and other stakeholders, including businesses, academia, professional societies, consumer advocates, nonprofits, and other civil society organizations will help drive and realize IoT innovation. Similarly, technical issues cross borders and require international coordination and cooperation.

Machine Learning and AI in IoT

Machine learning is closely related to (and often overlaps with) computational statistics, which also focuses on prediction-making through the use of computers. It has strong ties to mathematical optimization, which delivers methods, theory and application domains to the field. Machine learning is sometimes conflated with data mining, where the latter subfield focuses more on exploratory data analysis and is known as unsupervised learning. Machine learning can also be unsupervised and be used to learn and establish baseline behavioral profiles for various entities and then used to find meaningful anomalies.

Within the field of data analytics, machine learning is a method used to devise complex models and algorithms that lend themselves to prediction; in commercial use, this is known as predictive analytics. These analytical models allow researchers, data scientists, engineers, and analysts to "produce reliable, repeatable decisions and results" and uncover "hidden insights" through learning from historical relationships and trends in the data.[26]

Machine Learning Tasks

Machine learning tasks are typically classified into two broad categories, depending on whether there is a learning "signal" or "feedback" available to a learning system:

- **Supervised Learning:** The computer is presented with example inputs and their desired outputs, given by a "teacher", and the goal is to learn a general rule that maps inputs to outputs. As special cases, the input signal can be only partially available, or restricted to special feedback:
- **Semi-Supervised Learning:** The computer is given only an incomplete training signal: a training set with some (often many) of the target outputs missing.
- **Active Learning:** The computer can only obtain training labels for a limited set of instances (based on a budget), and also has to optimize its choice of objects to acquire labels for. When used interactively, these can be presented to the user for labeling.
- **Reinforcement Learning:** Training data (in form of rewards and punishments) is given only as feedback to the program's actions in a dynamic environment, such as driving a vehicle or playing a game against an opponent.
- **Unsupervised Learning:** No labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal in itself (discovering hidden patterns in data) or a means towards an end (feature learning).

Machine Learning Applications

Another categorization of machine learning tasks arises when one considers the desired *output* of a machine-learned system:

- In classification, inputs are divided into two or more classes, and the learner must produce a model that assigns unseen inputs to one or more (multi-label classification) of these classes. This is typically tackled in a supervised way. Spam filtering is an example of classification, where the inputs are email (or other) messages and the classes are "spam" and "not spam".

- In regression, also a supervised problem, the outputs are continuous rather than discrete.
- In clustering, a set of inputs is to be divided into groups. Unlike in classification, the groups are not known beforehand, making this typically an unsupervised task.
- Density estimation finds the distribution of inputs in some space.
- Dimensionality reduction simplifies inputs by mapping them into a lower-dimensional space. Topic modeling is a related problem, where a program is given a list of human language documents and is tasked to find out which documents cover similar topics.

Among other categories of machine learning problems, learning to learn learns its own inductive bias based on previous experience. Developmental learning, elaborated for robot learning, generates its own sequences (also called curriculum) of learning situations to cumulatively acquire repertoires of novel skills through autonomous self-exploration and social interaction with human teachers and using guidance mechanisms such as active learning, maturation, motor synergies, and imitation.

Approaches

Decision Tree Learning

Decision tree learning uses a decision tree as a predictive model, which maps observations about an item to conclusions about the item's target value.

Association Rule Learning

Association rule learning is a method for discovering interesting relations between variables in large databases.

Artificial Neural Networks

An artificial neural network (ANN) learning algorithm, usually called "neural network" (NN), is a learning algorithm that is vaguely inspired by biological neural networks. Computations are structured in terms of an interconnected group of artificial neurons, processing information using a connectionist approach to computation. Modern neural networks are non-linearstatistical data modeling tools. They are usually used to model complex relationships between inputs and outputs, to find patterns in data, or to capture the statistical structure in an unknown joint probability distribution between observed variables.

Deep Learning

Falling hardware prices and the development of GPUs for personal use in the last few years have contributed to the development of the concept of deep learning which consists of multiple hidden layers in an artificial neural network. This approach tries to model the way the human brain processes light and sound into vision and hearing. Some successful applications of deep learning are computer vision and speech recognition.[27]

Inductive Logic Programming

Inductive logic programming (ILP) is an approach to rule learning using logic programming as a uniform representation for input examples, background knowledge, and hypotheses. Given an encoding of the known background knowledge and a set of examples represented as a logical database of facts, an ILP system will derive a hypothesized logic program that entails all positive and no negative examples. Inductive programming is a related field that considers any kind of programming languages for representing hypotheses (and not only logic programming), such as functional programs.

Support Vector Machines

Support vector machines (SVMs) are a set of related supervised learning methods used for classification and regression. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that predicts whether a new example falls into one category or the other.

Clustering

Cluster analysis is the assignment of a set of observations into subsets (called *clusters*) so that observations within the same cluster are similar according to some predesignated criterion or criteria, while observations drawn from different clusters are dissimilar. Different clustering techniques make different assumptions on the structure of the data, often defined by some *similarity metric* and evaluated for example by *internal compactness* (similarity between members of the same cluster) and *separation* between different clusters. Other methods are based on *estimated density* and *graph connectivity*. Clustering is a method of unsupervised learning, and a common technique for statistical data analysis.

Bayesian Networks

A Bayesian network, belief network or directed acyclic graphical model is a probabilistic graphical model that represents a set of random variables and their conditional independencies via a directed acyclic graph (DAG). For example, a Bayesian network could represent the probabilistic relationships between diseases and symptoms. Given symptoms, the network can be used to compute the probabilities of the presence of various diseases. Efficient algorithms exist that perform inference and learning.

Reinforcement Learning

Reinforcement learning is concerned with how an *agent* ought to take *actions* in an *environment* so as to maximize some notion of long-term *reward*. Reinforcement learning algorithms attempt to find a *policy* that maps *states* of the world to the actions the agent ought to take in those states. Reinforcement learning differs from the supervised learningproblem in that correct input/output pairs are never presented, nor sub-optimal actions explicitly corrected.

Representation Learning

Several learning algorithms, mostly unsupervised learning algorithms, aim at discovering better representations of the inputs provided during training. Classical examples include principal components analysis and cluster analysis. Representation learning algorithms often attempt to preserve the information in their input but transform it in a way that makes it useful, often as a pre-processing step before performing classification or predictions, allowing reconstruction of the inputs coming from the unknown data generating distribution, while not being necessarily faithful for configurations that are implausible under that distribution.

Manifold learning algorithms attempt to do so under the constraint that the learned representation is low-dimensional. Sparse coding algorithms attempt to do so under the constraint that the learned representation is sparse (has many zeros). Multilinear subspace learning algorithms aim to learn low-dimensional representations directly from tensorrepresentations for multidimensional data, without reshaping them into (high-dimensional) vectors.[28] Deep learning algorithms discover multiple levels of representation, or a hierarchy of features, with higher-level, more abstract features defined in terms of (or generating) lower-level features. It has been argued that an intelligent machine is one that learns a representation that disentangles the underlying factors of variation that explain the observed data.[29]

Similarity and Metric Learning

In this problem, the learning machine is given pairs of examples that are considered similar and pairs of less similar objects. It then needs to learn a similarity function (or a distance metric function) that can predict if new objects are similar. It is sometimes used in Recommendation systems.

A **recommender system** or a **recommendation system** is a subclass of information filtering system that seeks to predict the "rating" or "preference" a user would give to an item.

Sparse Dictionary Learning

In this method, a datum is represented as a linear combination of basis functions, and the coefficients are assumed to be sparse. Let *x* be a *d*-dimensional datum, *D* be a *d* by *n*matrix, where each column of *D* represents a basis function. *r* is the coefficient to represent *x* using *D*. Mathematically, sparse dictionary learning means solving $x \approx D_r$ where *r* is sparse. Generally speaking, *n* is assumed to be larger than *d* to allow the freedom for a sparse representation.

Learning a dictionary along with sparse representations is strongly NP-hard and also difficult to solve approximately.[30] A popular heuristic method for sparse dictionary learning is K-SVD.

Sparse dictionary learning has been applied in several contexts. In classification, the problem is to determine which classes a previously unseen datum belongs to. Suppose a dictionary for each class has already been built. Then a new datum is associated with the class such that it's best sparsely represented by the corresponding dictionary. Sparse dictionary learning has also been applied in image de-noising. The key idea is that a clean image patch can be sparsely represented by an image dictionary, but the noise cannot.[31]

Genetic Algorithms

A genetic algorithm (GA) is a search heuristic that mimics the process of natural selection, and uses methods such as mutation and crossover to generate new genotype in the hope of finding good solutions to a given problem. In machine learning, genetic algorithms found some uses in the 1980s and 1990s.[32][33] Conversely, machine learning techniques have been used to improve the performance of genetic and evolutionary algorithms.[34]

Rule-Based Machine Learning

Rule-based machine learning is a general term for any machine learning method that identifies, learns, or evolves “rules” to store, manipulate or apply, knowledge. The defining characteristic of a rule-based machine learner is the identification and utilization of a set of relational rules that collectively represent the knowledge captured by the system. This is in contrast to other machine learners that commonly identify a singular model that can be universally applied to any instance in order to make a prediction.[33] Rule-based machine learning approaches include learning classifier systems, association rule learning, and artificial immune systems.

Learning Classifier Systems

Learning classifier systems (LCS) are a family of rule-based machine learning algorithms that combine a discovery component (e.g. typically a genetic algorithm) with a learning component (performing either supervised learning, reinforcement learning, or unsupervised learning). They seek to identify a set of context-dependent rules that collectively store and apply knowledge in a piecewise manner in order to make predictions.[34]

Applications of Machine Learning

Applications for machine learning include:

- Agriculture
- Automated theorem proving
- Adaptive websites
- Affective computing
- Bioinformatics
- Brain–machine interfaces
- Cheminformatics
- Classifying DNA sequences
- Computational anatomy
- Computer Networks
- Telecommunication
- Computer vision, including object recognition
- Detecting credit-card fraud

- General game playing
- Information retrieval
- Internet fraud detection
- Linguistics
- Marketing
- Machine learning control
- Machine perception
- Medical diagnosis
- Economics
- Insurance
- Natural language processing
- Natural language understanding
- Optimization and metaheuristic
- Online advertising
- Recommender systems
- Robot locomotion
- Search engines
- Sentiment analysis (or opinion mining)
- Sequence mining
- Software engineering
- Speech and handwriting recognition
- Financial market analysis
- Structural health monitoring
- Syntactic pattern recognition
- Time series forecasting
- User behavior analytics
- Translation[40]

In 2006, the online movie company Netflix held the first "Netflix Prize" competition to find a program to better predict user preferences and improve the accuracy on its existing Cinematch movie recommendation algorithm by at least 10%. A joint team made up of researchers from AT&T Labs-Research in collaboration with the teams Big Chaos and Pragmatic Theory built an ensemble model to win the Grand Prize in 2009 for $1 million. Shortly after the prize was awarded, Netflix realized that viewers' ratings were not the best indicators of their viewing patterns ("everything is a recommendation") and they changed their recommendation engine accordingly.

In 2010 The Wall Street Journal wrote about the firm Rebellion Research and their use of Machine Learning to predict the financial crisis.

In 2012, co-founder of Sun Microsystems Vinod Khosla predicted that 80% of medical doctors jobs would be lost in the next two decades to automated machine learning medical diagnostic software.

In 2014, it has been reported that a machine learning algorithm has been applied in Art History to study fine art paintings, and that it may have revealed previously unrecognized influences between artists.

REFERENCES

Aharon, M., Elad, M., & Bruckstein, A. (2006). K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. *Signal Processing, IEEE Transactions on*, *54*(11), 4311–4322. doi:10.1109/TSP.2006.881199

Ambler, S. W. (2014). We need more Agile IT Now! In *Dr. Dobb's The world of software Development*. San Francisco: UBM.

Bassel, G. W., Glaab, E., Marquez, J., Holdsworth, M. J., & Bacardit, J. (2011). Functional Network Construction in Arabidopsis Using Rule-Based Machine Learning on Large-Scale Data Sets. *The Plant Cell, 23*(9), 3101–3116. doi:10.1105/tpc.111.088153

Bengio, Y. (2009). *Learning Deep Architectures for AI*. Now Publishers Inc.

Best Practices in change, Configuration and Release Management (Report). (2010, July 14). Gartner.

Bourne, J. (2017). *New research questions strategic importance of DevOps despite rise in usage*. CloudTech.

Chen, L. (2015). Continuous Delivery: Huge Benefits, but Challenges Too. *IEEE Software*, *32*(2), 50–54. doi:10.1109/MS.2015.27

DataKitchen. (2017). *How to Become a Rising Star with Data Analytics*. Author.

Debois, P. (2009). *DevOpsDays Ghent*. DevopsDays.

DevOps: A Job Title or a School of Thought? (2017). Monster Career Advice.

Gartner IT Glossary – devops. (2015). Gartner.

Hammond, J. (2011). *The Relationship between DevOps and Continuous Delivery*. Forrester Research.

Humble, J., & Farley, D. (2011). *Continuous Delivery: reliable software releases through build, test, and deployment automation*. Pearson Education Inc.

Is DevOps a Title? (2014). DevOps.com.

Jones, S., Noppen, J., & Lettice, F. (2016). *Management challenges for DevOps adoption within UK SMEs*. Academic Press.

Koza, J. R., Bennett, F. H., Andre, D., & Keane, M. A. (1996). Automated Design of Both the Topology and Sizing of Analog Electrical Circuits Using Genetic Programming. In *Artificial Intelligence in Design '96*. Springer; doi:10.1007/978-94-009-0279-4_9

Lee, H., Grosse, R., Ranganath, R., & Ng, A. Y. (2009). Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations. *Proceedings of the 26th Annual International Conference on Machine Learning*. 10.1145/1553374.1553453

Loukides, M. (2012). *What is Devops?* O'Reilly Media.

Lu, H., Plataniotis, K. N., & Venetsanopoulos, A. N. (2011). A Survey of Multilinear Subspace Learning for Tensor Data. *Pattern Recognition*, *44*(7), 1540–1551. doi:10.1016/j.patcog.2011.01.004

Machine Learning: What it is and why it matters. Retrieved from www.sas.com

Nasrat, P. (2011). *Agile Infrastructure*. InfoQ.

Palmer, A. (2015). *From DevOps to DataOps*. Tamr Inc.

ResOps, daily adventures of DevOps in Research - EMBL-EBI Technical Services Cluster blog. (2018, February 8). EMBL-EBI Technical Services Cluster blog.

Samuel, A. (1959). Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development*, *3*(3), 210–229. doi:10.1147/rd.33.0210

Tillmann, A. M. (2015). On the Computational Intractability of Exact and Approximate Dictionary Learning. *IEEE Signal Processing Letters*, *22*(1), 45–49. doi:10.1109/LSP.2014.2345761

Urbanowicz, R. J., & Moore, J. H. (2009). Learning Classifier Systems: A Complete Introduction, Review, and Roadmap. *Journal of Artificial Evolution and Applications*, 1–25. doi:10.1155/2009/736398

What are known useful and misleading memes in the DevOps culture? (2017). Retrieved from devops.stackexchange.com

What is DevOps? (2014). NewRelic.com.

Zhang, Zhan, Lin, Chen, Gong, Zhong, … Shi. (2011). Evolutionary Computation Meets Machine Learning: A Survey. *IEEE Computational Intelligence Magazine, 6*(4), 68–75. doi:. doi:10.1109/mci.2011.942584

Chapter 3
Modelling and Designing of IoT Systems Using UML Diagrams:
An Introduction

K. Sridhar Patnaik
Birla Institute of Technology – Mesra, India

Itu Snigdh
Birla Institute of Technology – Mesra, India

ABSTRACT

Despite the rapid growth in IoT research, a general principled software engineering approach for the systematic development of IoT systems and applications is still missing. Software engineering as a discipline provides the necessary platform to carry on the underlying design, coding, implementation, as well as maintenance of such systems. UML diagrams present a visually comprehensible outlay of the construction of IoT systems. The chapter covers the modelling of IoT systems using UML diagrams. Starting with the architectural design of any IoT system to behavioral aspects is covered in this chapter using a case study of IoT-based remote patient health monitoring system. The diagrams shown in this chapter are the sample diagrams for understanding IoT-based complex systems. The chapter focuses on the work carried out by Franco Zambonelli in context of developing abstract model of an IoT system using software engineering concepts. The chapter also focus on the pioneer work carried by J. F. Peters in intelligent system design patterns for robotic devices using pattern classification.

DOI: 10.4018/978-1-5225-7790-4.ch003

INTRODUCTION

The term “Internet of Things” (IoT) was first used in 1999 by British technology pioneer Kevin Ashton to describe a system in which objects in the physical world could be connected to the Internet by sensors. Ashton coined the term to illustrate the power of connecting Radio-Frequency Identification (RFID) tags used in corporate supply chains to the Internet in order to count and track goods without the need for human intervention. Today, the Internet of Things has become a popular term for describing scenarios in which Internet connectivity and computing capability extend to a variety of objects, devices, sensors, and everyday items. Internet of Things is a platform where every day devices become smarter, every day processing becomes intelligent, and every day communication becomes informative. While the Internet of Things is still seeking its own shape, its effects have already stared in making incredible strides as a universal solution media for the connected scenario. (Giusto.D et al., 2010).

Though IoT and software engineering paradigm are greatly disconnected, any system that has a wide spread applicability needs to be built on concrete concepts. Software engineering as a discipline provides the necessary platform to carry on the underlying design, coding, implementation as well as maintenance of such systems. With the abundance of applications that have emerged due to the IoT concept, there are a number of underlying processes that now need to be generalized as for example the data gathering, service discovery and the interfaces design. The contributors to successful software for IoT are mainly the designers, testers and the developers. However their level of association to an IoT project is different. An application designer works with the design of the application while the tester and the developer are more connected to the simulation, programming framework and execution platform backend. UML diagrams aim to guide the IoT design to a more standardised methodology of development and deployment. It presents a visually comprehensible outlay of the construction of IoT systems.

In this chapter, the authors attempt at framing the key concepts and abstractions that revolve around the design and development of IoT systems and applications, and that could represent the ground on which to start shaping the guidelines of a new IoT-oriented software engineering discipline. Architecture specific study does always pave the conformation of related field. The lack of overall architectural knowledge is presently resisting the researchers to get through the scope of Internet of Things centric approaches.

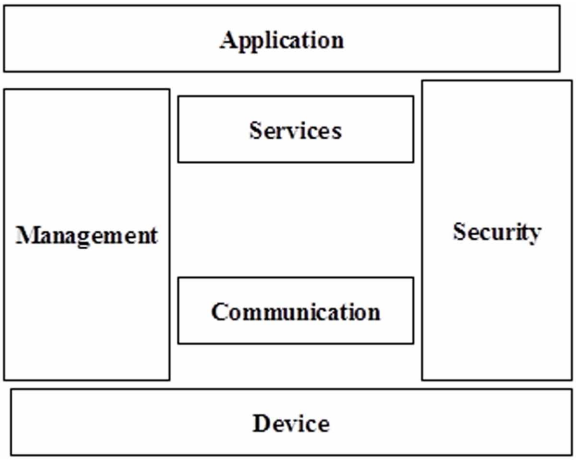
So as to capture the different views of an IoT system, UML plays an important role. Starting with the architectural design of any IoT system to Behavioural aspects is covered in this chapter .The Chapter also tries to cover some basic aspects of Intelligent system design patterns for robotic devices using pattern classification.

IOT ARCHITECTURE

The architecture of IoT system consists of physical layer, virtual layer, or a hybrid of the two, with a collection of numerous active physical things, sensors, actuators, cloud services, specific IoT protocols, communication layers, users, developers, and enterprise layer. Figure 1 shows the Layered architectural framework of an IoT system (Bagga and Meddisetti, 2015). Various domain specific architectures based on the broad areas, such as: RFID, service oriented architecture, wireless sensor network, supply chain management, industry, healthcare, smart city, logistics, connected living, big data, cloud computing, social computing, and security are described in (Ray,2016). The selection of these domains depends upon current scenario of IoT applicability. An IoT system comprises of a number of functional blocks (Bagga and Meddisetti, 2015) that provide the system the capabilities for identification, sensing, actuation, communication, and management. These functional blocks are described as:

- **Device:** Devices that provide sensing, actuation, monitoring and control functions.
- **Communication:** The communication block handles the communication for the IoT system through various IoT protocols.
- **Services:** An IoT system uses various types of IoT services such as services for device monitoring, device control services, data publishing services and services for device discovery.
- **Management:** Management functional block provides various functions to govern the IoT system.
- **Security:** Security functional block secures the IoT system and by providing functions such as authentication, authorization, message and content integrity, and data security.
- **Application:** IoT applications compromise an interface that the users can use to control and monitor various features of the IoT system. Applications also permit users to view the system status and view or evaluate the processed data.

Figure 1. Layered architectural framework of an IoT system



Depending on the mode of development the architecture of IoT systems may be classified as:

- Service oriented architecture
- API oriented architecturea

The SOA architecture comprises of four distinct layers, namely, Sensing, network, service and interface layer. The sensing layer deals with integrating hardware objects o sense the state of things while the network layer is the infrastructure to aid wireless or wired communication among the objects. The service layer is responsible for creation and management of services required by users or application and the interface layer consists of interaction method with the users or applications (R. Buyya, A. V. Dastjerdi, 2016)

The advantage of SOA based architectures are:

1. Abstraction and modular design
2. Augments interoperability and scalability among objects
3. Ease of building a more diverse and complex service die to modular composability.

Web- APIs and representational state transfer (REST) based methods are types of API oriented architecture. They incorporate light weight data exchange formats and use communication channel and power of devices more efficiently. Developers and business managers share APIs from an early stage of their application development lifecycle in order to provide an open data environment. This exposes the data to other developers and end users and hence facilitates collaborative information gathering, sharing and updating.

THE NEED

The software engineering principles (Mall.R, 2014)(Pressman. R.S 2009) requires identifying the general features and issues that characterize most current approaches to IoT systems design and development (Zambonelli, 2016).

Software engineering principles aim at better understanding of the problem, capability to design a workable solution, implementing in a solid way, and testing the solution thoroughly (R.S Pressman, 1998) It also tries to control changes to it as the work progresses and have some mechanism for ensuring the end result's quality. However, applying solid software engineering principles to IoT applications may often prove quite difficult. One main cause is that every IoT application requires much shorter development times and product life cycles. For this we need faster tools and we need to plan and design only as the specific project's purpose dictates.

This doesn't mean that classical methods don't apply at all, but a necessary adaptation is required. An argument in favor of engineering any IoT application or using a model is that like any other software critical qualitative requirements should be specified and a suitable architecture derived to be able to estimate and be prepared for the incurring cost and other constraints. Failing to do so will substantially increase the applications' risk of failure, because of misunderstanding what the application is really all about.

Another major concern in developing an IoT application is that it requires a very light process. We are aware that the heart of a good and light development process is a discipline of design: careful decomposition into cohesive pieces with thin interfaces between them. Thus, adoption of a modeling strategy allows specifying the critical requirements, especially the qualitative ones, in a measurable and testable manner.

Therefore there is a need to train the new generation of IoT software developers. The apriori knowledge of software development and deployment becomes imperative to avoid developing IoT applications in an uncontrolled fashion that could possibly endanger lives. The IEEE Software community, with its deep collective knowledge of both past and present practices, is uniquely positioned to lead the way and avoid costly reinvention of the wheel (R. Buyya, A. V. Dastjerdi, 2016)

ROLE OF SOFTWARE MIDDLEWARE

Internet of Things is a combined part of future Internet and ubiquitous computing. It demands interactions with the heterogeneous raw sensors, aggregators, actuators and diverse domain of context aware applications, preserving the security and privacy. It comprises two definite components; Internet and things.

"Things" are heterogeneous in nature and seamlessly integrated into the information network. To make "things" (Zambonelli, 2016) usable and capable of serving purposes, there is need of software infrastructures (IoT middleware) (Razzaque et al. 2016) capable of supporting the different things and of providing some means for stakeholders and users to access the IoT system and take advantage of its functionalities. This involves a variety of technical issues like *Interoperability, Semantics, Discovery, Group Formation, Coordination, Context-awareness and self-adaptation*. These support the "Things" for working in unison as well as provide different functionalities and possible accesses to the system units. However these software infrastructures' design faces the above mentioned challenges that may be enumerated as follows (Zambonelli, 2016):

1. **Interoperability:** As "Things" are usually diverse in nature or heterogeneous. Also, they are usually coupled to each other for the desired functionality.
2. **Common Semantics:** To be able to identify, refer to and enable integration and cooperation among the things. This becomes most important when we need to integrate and operate devices seamlessly.
3. **Group Discovery and Coordination:** To be able to associate humans with devices for achieving the desired functionalities. The devices require accepting information and process data accordingly as per need.
4. **Self-Adaptation:** To be able to recover from anomalous situations for example unavailability of resources due to inherent mobility of the things. These also include changes in the environment of operation etc.
5. **Context Awareness:** To be able to tune in data suitable to the requirements of the functionalities.

In order to meet the above said demands IoT will require a software platform defined as middleware, fundamentally providing abstraction to applications from the things, and offering multiple services. Development of middleware in the domain of IoT is an active area of research. There have been a lot of researches towards building up this middleware addressing interoperability across heterogeneous devices serving diverse domains of applications, adaptation, context awareness, device discovery and management, scalability, managing a large data volumes and, privacy, security aspects of the said IoT environment. Therefore there is a strong need to understand how the existing IoT-middleware systems work and address the different requirements of ubiquity before designing of an IoT application (Valérie Issarny etal., 2011; M. Weyrich and C. Ebert,2016).

IOT: SOFTWARE ENGINEERING

The "things" (Zambonelli, 2016] in the IoT vision may encompass a large number of physical objects, and also include places and persons. Physical objects and places can be tracked and controlled by connecting them to low-cost wireless electronic devices. At the lower end of the spectrum, RFID tags or Bluetooth beacons, based on low-cost and short-range communication protocols, can be attached to any kind of objects to enable tracking their positions and status, and possibly to associate some digital information with them. Advanced devices with integrating environmental or motion sensors can detect the present and the past activities associated with objects or with some place. These objects can be controlled remotely via proper digitally-controller actuators–and possibly autonomous–delegating themof autonomously direct their activities.

Taking into account the "access" to the functionalities and capabilities of individual things by users, the scene is currently dominated by the so called "Web of Things" (WoT) vision(Heuer et al.2015).

The term "IoT System" is generally referred to the overall set of IoT devices and to the associated middleware infrastructure devoted to manage their networking and their context-aware interactions. Over the logical layer of an IoT system, specific software can be deployed to handle the activities of the system so as to provide: A number of *specific services*(Zambonelli, 2016). This services enablesthe stakeholders and users to access and exploit individual things not only to direct/activate their sensing/actuating capabilities, but also to coordinate services that access groups of things and coordinate their sensing/actuating capabilities. For instance, if we consider a smart conference room of smart hotel (Zambonelli, 2016) other than the basic services to access and control individual appliances, one can think of providing extra options. Some of them may be given as manipulating the overall ambience of the room by providing a coordinated service that, through accessing and directing the lighting system by light sensors and the windows obscuring system, can

As far as the general *applications or suites* are concerned, the software systems has to both regulate the overall functionality of an IoT system (or of some of its parts), so as to ensure specific overall behaviour of the system, as well as to provide a matched set of services to access the system and (possibly) its configuration. In context of smart hotel scenario, one can think at applications to control the overall heating systems and lightening systems, and giving to hotel clerksthe access to services to change the configuration of the associated parameter.Clearly, depending on the specific scenario, one can think at IoT systems in which services may exists only confined within the context of some general application, but also at scenarios in which there are services that can be deployed as stand-alone software(Ray,2016 ;Minerva,2016;Zambonelli, 2016).

The development of IoT systems is based on the concepts and abstractions involved in the life cycle (spanning analysis, design, and implementation). Figure 2 graphically frames such concepts.(Zambonelli, 2016)

The analysis layer concerns the actors who are the stakeholders and other users like global managers, local managers and end users. Global Managers (Zambonelli, 2016) are the owners of an overall IoT system and infrastructure, or delegates empowered to exert control and establishing policies over the configuration, structure, and overall functioning of its applications and services. Considering the previously mentioned smart hotel scenario, the global manager corresponds with the system manager who is devoted to control the overall IoT system of the hotel according to the directives of the hotel management for e.g., for deciding the heating levels or for surveillance strategies. Local Managers are owners/delegates (whether permanently

Figure 2. Key concepts and abstractions for IoT software engineering. (Zambonelli, 2016)

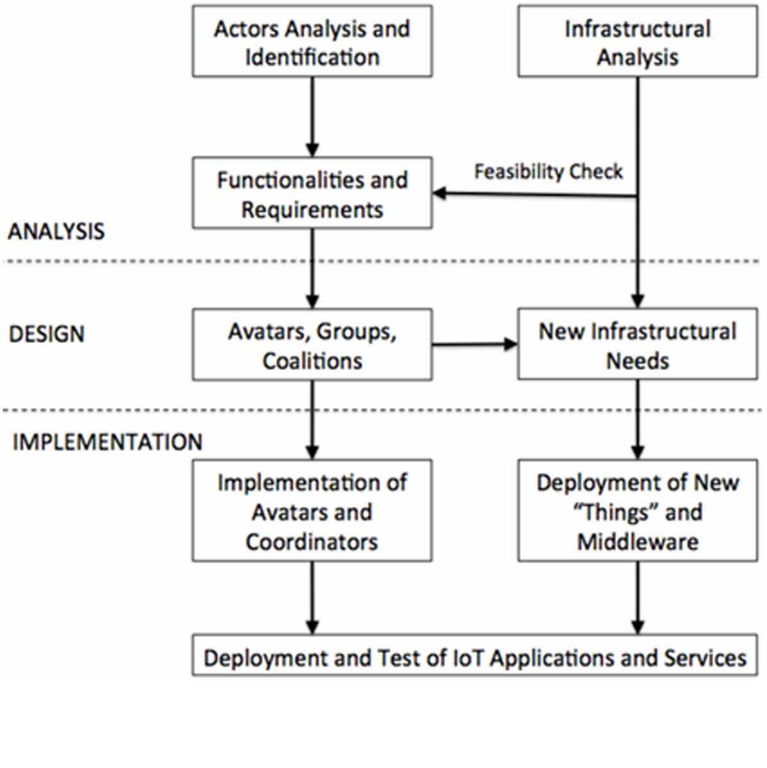
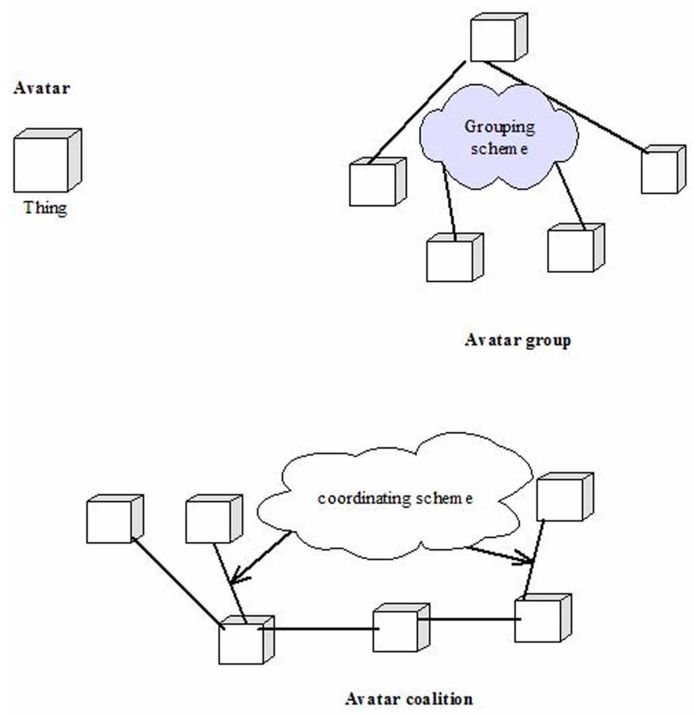

Actors Analysis and Identification
Infrastructural Analysis
Feasibility Check
Functionalities and Requirements
ANALYSIS
DESIGN
Avatars, Groups, Coalitions
New Infrastructural Needs
IMPLEMENTATION
Implementation of Avatars and Coordinators
Deployment of New "Things" and Middleware
Deployment and Test of IoT Applications and Services

Figure 3. Avatars, groups, and coalitions



or on a temporary basis) of a limited portion of the IoT system, empowered to enforce local control and policies for that portion of the

system. In the hotel scenario (Zambonelli, 2016), these could correspond to hotel guests, empowered to control the IoT system in their room, and tune the local parameters and exploit its services according to own specific needs. Or they can be the conference organizers in charge of managing and configuring the services of the rented conference rooms.

Users are the persons or groups that have limited access to the overall configuration of the IoT applications and services, i.e., cannot impose policies on them, but are nevertheless entitled to exploit its services. In the hotel scenario, these include conference delegates authorized to access the conference facilities (e.g., uploading presentations in the projector), but are not entitled to modify the configuration of the conference room.

The three identified classes of actors are of a very general nature, beside the hotel scenario. For example, in a scenario of energy management in a smart city, they could correspond to, respectively: city managers, house/shop owners, private citizens and tourists. In the area of urban mobility, they could correspond to, respectively: mobility managers, parking owners or car sharing companies, private drivers.

Functionalities

Beside things provided with basic sensing/actuating functionalities, one should consider the presence of smarter things that can be activated to perform in autonomy some long-term activities associated with their nature and with their role in the socio/physical environment in which they situates. These can range from simply cleaning robots to more sophisticated autonomous personal assistants (Ray,2016),(Zambonelli, 2016).

IoT applications are not simply concerned with providing a suite of coordinated functionalities, but they should also globally regulate the activities of the IoT systems on a continuous basis, according to the policies established by its stakeholders and to their objectives.

As a consequence, other than analyzing the specific functionalities to deliver, one also has to identify the policies and goals to be associated with services and applications, i.e., the desirable "state of the affairs" to strive for in the context of the socio-cyber-physical system where IoT applications and services operate. In this perspective, the general classes of functionalities to be identified for the development of IoT applications and services include: Policies(Local and Global),Goals and Functions.

Avatars and Coalitions

An avatar is the general abstraction for individual things and also for group of things (and possibly other avatars) that contribute to define a unique functionality/service. Avatars abstract away form the specific physical/social/technological characteristics of the things they represent, and are defined by means of: Identity, Services, Goals and Events(Fig 3).

Clearly, for group of avatars, an internal orchestration scheme must be defined for coordinating the activities/functionalities of the things (or of the other avatars) it includes. In general terms, an orchestration scheme defines the internal workflow of activities among the composing things and avatars, and the constrains/conditions they are subjected to. Orchestration scheme may also account for contextual information, to make the activities of the group of context-aware. The need of defining orchestrations schemes and constraints to rules the access and usage of (group of) things is generally attributed – with specific characteristics and terminologies – in most middleware and programming approaches for IoT.

More in general, the avatar abstraction is in line, and account for all the typical characteristics, of most existing IoT approaches. Although the idea is not fully in line with that of RESTfulWoT approaches, because of the stateful concepts of goals and events, most of them recognize the need to somehow incorporate similar concepts within RESTful architectures (Heuer .J, Hund.J, and Pfaff,2015) to suit the dynamic and contextual nature of IoT systems and applications.

- **Coalitions:** A coalition as a group of avatars that coordinate each other's activities in order to reach specific goals, or enact specific policies. Accordingly, coalitions may be of a temporary or permanent nature. Unlike avatar groups, coalitions do not necessarily have an identity, and does not necessarily provide services.
- **Rules for membership**, to specify the conditions upon which an avatar should/could enter coalitions. From the viewpoint of individual avatars, the act of entering a coalition can be represented by the activation of a specific goal based on pre-conditions that correspond to the rules for membership. Coordination pattern, to define the pattern (interaction protocol and shared strategy) by which the members of the coalition have to interact. The coordination pattern may include an explicit representation of the goal by which the coalition has been activated. However, such goal can also be implicit in the definition of the protocol and of the strategy. Coordination law, to express constraints that must be enforced in the way the avatars involved in the coalition should act and interact.

UML REPRESENTATIONS OF IoT SYSTEMS

The Unified Modelling Language (UML) (Miles and Hamilton, 2006), (Gamma et al.2003) (Flower.M)is a standard language for specifying, visualizing,constructing, and documenting the artifacts of software systems, as well as for businessmodeling and other non-software systems. The UML represents a collection of bestengineering practices that have proven successful in the modeling of large and complexsystems. The UML is a very important part of developing object oriented software and thesoftware development process. The UML uses mostly graphical notations to express thedesign of software projects. Using the UML helps project teams communicate, explorepotential designs, and validate the architectural design of the software.

Each UML diagram is designed to let developers and customers view a software system froma different perspective and in varying degrees of abstraction. UML diagrams commonlycreated in visual modeling tools include:

- *Use Case Diagram* displays the relationship among actors and use cases.
- *Class Diagram* models class structure and contents using design elements such as classes, packages and objects. It also displays relationships such as containment, inheritance, associations and others.
- **Interaction Diagrams:**
  - *Sequence Diagram* displays the time sequence of the objects participating in theinteraction. This consists of the vertical dimension (time) and horizontal dimension (different objects).
  - *Collaboration Diagram* displays an interaction organized around the objects and theirlinks to one another. Numbers are used to show the sequence of messages.
  - *State Diagram* displays the sequences of states that an object of an interaction goes throughduring its life in response to received stimuli, together with its responses and actions.
  - *Activity Diagram* displays a special state diagram where most of the states are action statesand most of the transitions are triggered by completion of the actions in the source states.This diagram focuses on flows driven by internal processing.
- **Physical Diagrams:**
  - *Component Diagram* displays the high level packaged structure of the code itself.Dependencies among components are shown, including source code components,binary code components, and executable components. Some components exist atcompile time, at link time, at run times well as at more than one time.
  - *Deployment Diagram* displays the configuration of run-time processing elementsand the software components, processes, and objects that live on them. Softwarecomponent instances represent run-time manifestations of code units.

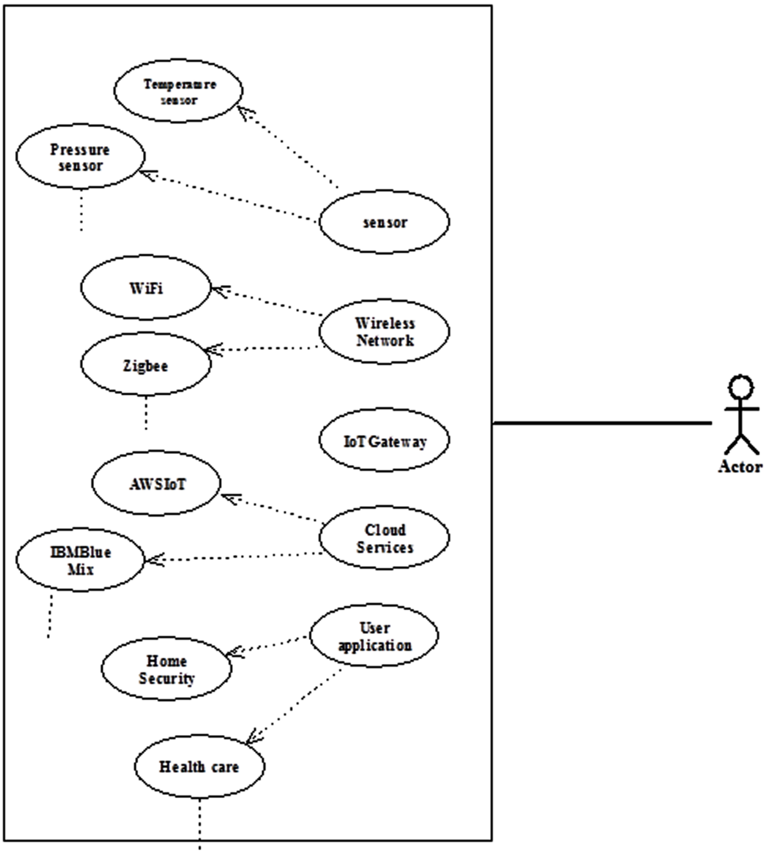
Use Case Diagram of an IoT System

Figure4 shows the generalized UML use case diagram of an IoT system consisting of various functional requirements like Sensing, Communication to Cloud through IoT Gateway and the user applications.

- **Sample Case Study**: IoT based Remote Patient Health Monitoring System

Medical care and healthcare represent one of the most attractive application areas of the IoT(Budida DAM and Mangrulkar R.S,2017). The Internet of Things (IoT) has the potential to give rise to many medical applications such as remote health

Figure 4. Use case diagram of an IoT system



monitoring, fitness programs, incurable diseases, and elderly care. Thus, various medical devices, sensors, diagnostic and imaging devices can be viewed as smart devices or smart objects constituting an interior component of the IoT. IoT-based healthcare services are foreseen to minimize costs, increase and provide a better quality of life, and enrich the user's experience.

As far as the healthcare industry is concerned diagnosis and monitoring of health is a very important task. Due to time constraint, people are not visiting hospitals, which might and possibly lead to a lot of health issues in one instant of time (Budida DAM and Mangrulkar R.S,2017). Most of the healthcare systems have been developed to predict and diagnose the health of the patients by which people who are busy in their schedule can also monitor their health at regular intervals. Many studies show that early prediction is the best way to cure health because early diagnosis will help and alert the patients to know the health status. Healthcare being a global issue more particularly India being a most populated nation where majority of which live in villages deprived of healthcare facilities on real time basis continuously and regularly. With the increasing use of technology, there is an urgent need to have such a smart remote health monitoring system that can communicate between network devices and application which will help the patients and doctors to monitor, track and record the patient's sensitive data containing medical information (Islam. Set al.2015), Hassanalieragh.M et al.2015)(Sotiriadis.S et al.2013)

This case study depicts the idea of solving health issues using the latest technology, Internet of Things (IoT) with help of UML diagrams. The high level architectural representation (see Figure 5)(Mora.H et al., 2017) of smart health care system using Internet of Things(IoT), which is aimed to provide a Better HealthCare to everyone. Using this system architecture, patient's body parameters can be measured in real time. Sensors collect patient's body parameters and transfer that data to Microcontroller (with Arduino/Raspberry pi) which further transfers that data to the cloud database server (MySQL) through IoT gateway. This MySQL database server manages the data and provides accessibility to the patients and doctors through Android App. If data is abnormal then patient gets notification, also care takers will get emergency messages. With the help of different decision making algorithms, decisions can be made easily and fast and can beaccessed by the patients. The system provides a better HealthCare to everyone and error free and smooth communication to patients.

- **Use Case Diagram:** A Use Case Diagram (Flower.M;Matha M.P, 2008;Rumbaugh.J;Blaha, 2007)consists of set of elements and the relationships between them. It depicts all the scenarios, regarding how our application interacts with users and other external systems to achieve the goals of application. The main components of a use case diagram include actors, use cases and their relationships. The use case is an external view of the system that represents some actions that the user performs to get a job done. Actors are the users who interact with the application.Figure 6 shows the use case diagram of IoT based Remote Patient Health Monitoring System.

Actors

The Actors of the system are Patient, Guardian and Doctor

Use Cases

A set of use cases based on the functionalities and goals of the application.

Figure 5. Architectural view of IoT based Remote Patient Health Monitoring System

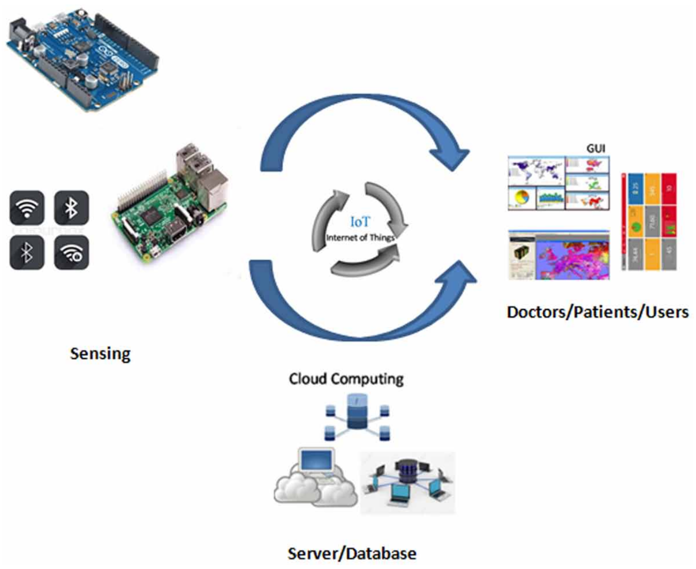
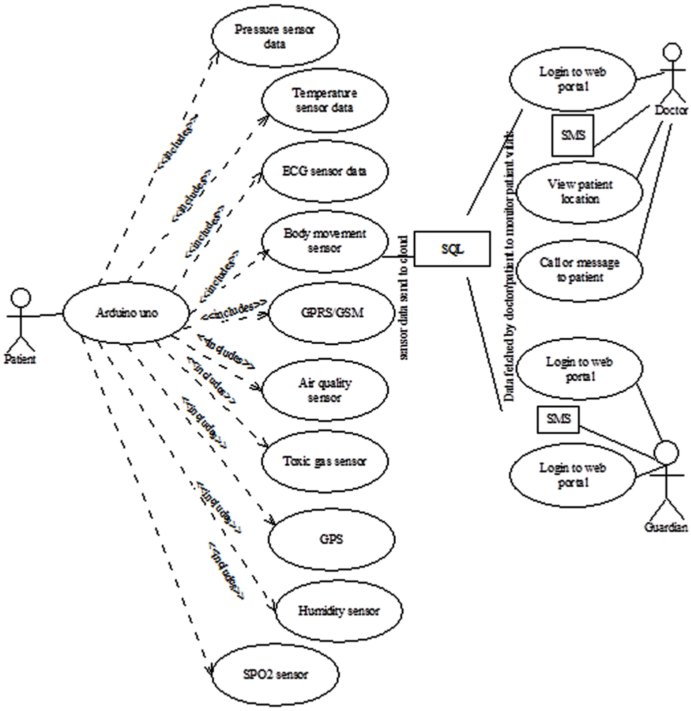


Figure 6. IoT based remote patient health monitoring system (Rao, 2017)



- **Login:** This use case denotes a set of actions required for Subject to login into the application.
- **Call Service:** This use case denotes a set of actions required by doctor to call a guardian or patient in case medical emergencies.
- **View Location:** This use case denotes a set of actions required by Guardian or Doctor to locate subject on map after receiving his location details.
- **Messaging Service:** This use case denotes a set of actions required by Doctor to send a message to subject's guardian in case of emergencies.
- **Arduino/Raspberry pi:-**This use case denotes the functionality of gathering sensing data from sensors connected to the device (here patient) and the board.
- **Class Diagram:** Class diagrams (Flower.M;Matha M.P,2008;Rumbaugh.J and Blaha,2007) are a type of structure diagram because they describe what must be present in the system being modelled. Since classes are the building block of objects, class diagrams are the building blocks of UML. A System is a collection of objects which are instances of classes. Every class is associated or related with other one or two classes. The overall static structural representation is shown by class diagrams. Figure 7 shows the sample class diagram of the health monitoring system. The important classes and their relationships are shown in the Figure 7.

Interaction Diagram

Figure 8 shows a sample collaboration diagram (Peters, 2004, 2003) of sensor with three instances: Stimulus, Converter and Filter. Stimulus invokes convert() in the converter. After conversion of an input signal represented by values in set X. converter invokes modulate()in filter. Many other sensor models are possible (Peters, 2004, 2003) .Various design patterns(Peters, 2004, 2003) can also be used to represent an IoT system.

- **Sequence Diagram:** Sequence diagram (Flower.M;Matha M.P,2008;Rumbaugh.J; Blaha,2007)is an interaction diagram that illustrates the interaction between objects within a system. It is structured in such a way that it represents a timeline which begins at the top and descends gradually to mark the sequence of interactions. Each object has a column and the messages exchanged between them are represented by arrows. Figure 9 shows the interactions through message passing from one object to another.
- **Collaboration Diagram:** Communication diagrams (Flower.M;Matha M.P,2008;Rumbaugh.J and Blaha,2007) formerly known as collaboration

Figure 7. Shows the sample class diagram of the health monitoring system

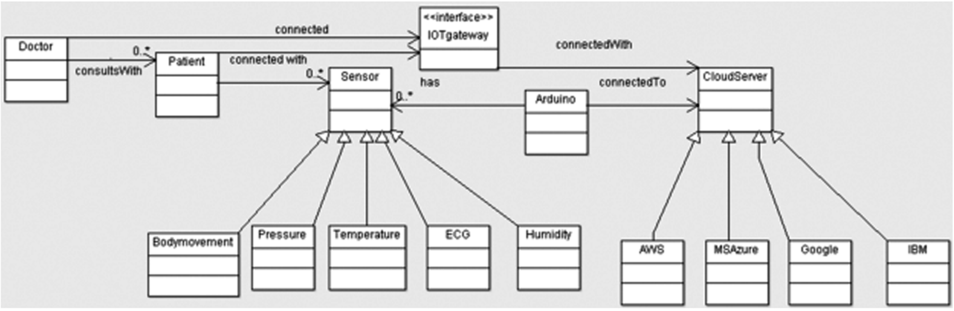
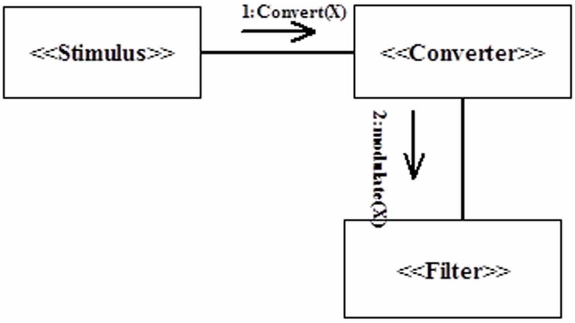


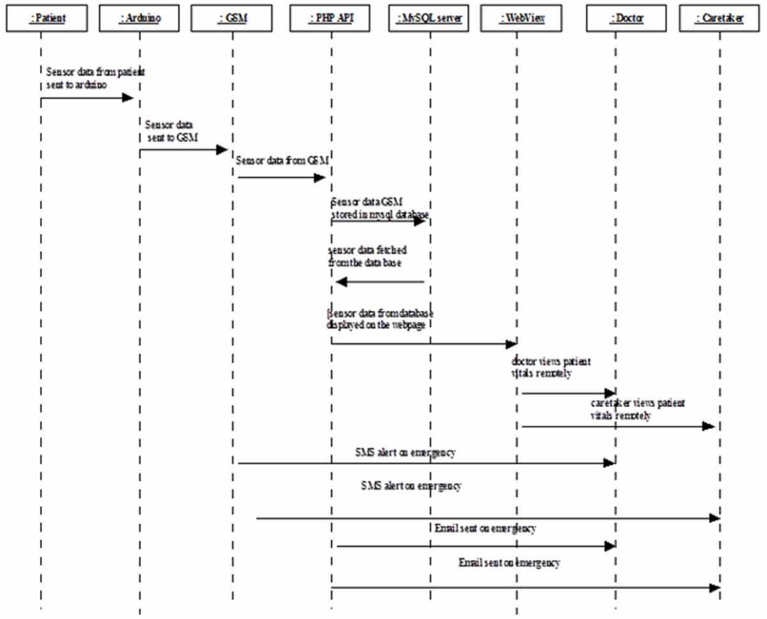
Figure 8. Sensor model (Peters, 2004, 2003)



diagrams, are almost identical to sequence diagrams in UML, but they focus more on the relationships of objects—how they associate and connect through messages in a sequence rather than interactions. Figure 10 shows the collaboration diagram of the given case study.

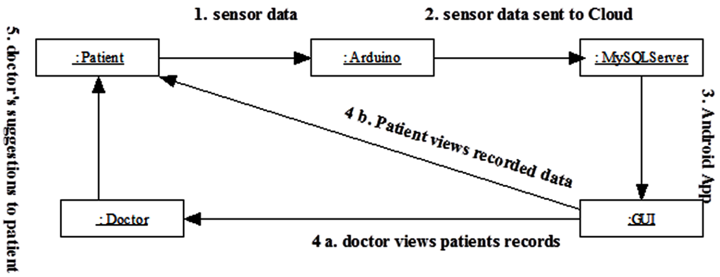
- **Activity Diagrams:** Activity diagram (Flower.M; Matha M.P,2008;Rumbaugh.J and Blaha,2007) represents the dynamics of the system.They are flow charts that are used to show the workflow of a system, the flow of control from activity to activity in the system andwhat activities can be done in parallel alsoalternate paths through the flow.They can show the flow across use cases or within a use case. Figure 11 shows the activity diagram of the health monitoring system.
- **State Chart Diagram**: Statechart diagram (Flower.M; Matha M.P, 2008; Rumbaugh.J and Blaha,2007) is one of the five UML diagrams used to model the dynamic nature of a system. They define different states of an object during its lifetime and these states are changed by events. Statechart diagrams are useful to model the reactive systems. Reactive systems can be

Figure 9. Sequence diagram



: Patient
: Arduino
: GSM
: PHP API
: MySQL server
: WebView
: Doctor
: Caretaker
Sensor data from patient sent to arduino
Sensor data sent to GSM
Sensor data from GSM
Sensor data GSM stored in mysql database
sensor data fetched from the data base
Sensor data from database displayed on the webpage
doctor views patient vitals remotely
caretaker views patient vitals remotely
SMS alert on emergency
SMS alert on emergency
Email sent on emergency
Email sent on emergency

Figure 10. Collaboration diagram



1. sensor data
2. sensor data sent to Cloud
: Patient
: Arduino
: MySQLServer
3. Android App
4 b. Patient views recorded data
: Doctor
:GUI
4 a. doctor views patients records
5. doctor's suggestions to patient

defined as a system that responds to external or internal events. Statechart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. The most important purpose of Statechart diagram is to model lifetime of an object from creation to termination. Statechart diagrams are also used for forward and reverse engineering of a system.Figure 12 shows the sample state chart diagram of the health monitoring system.

- **Component Diagram:** The purpose of a component diagram (Flower.M; Matha M.P, 2008) is to show the relationship between different components

Figure 11. Activity diagram of the health monitoring system

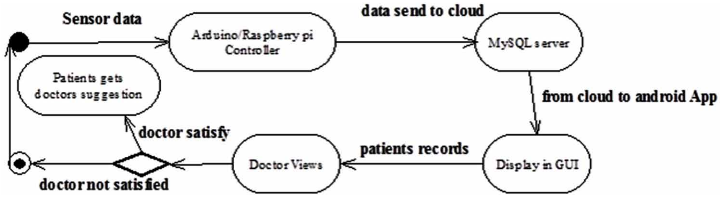
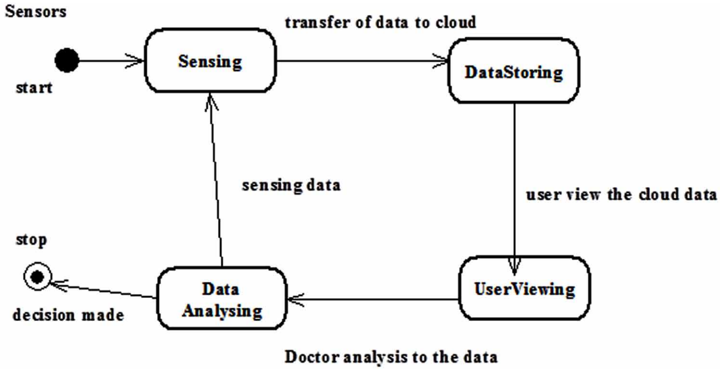


Figure 12. State Chart diagram



in a system. For the purpose of UML 2.0, the term “component” refers to a module of classes that represent independent systems or subsystems with the ability to interface with the rest of the system.Figure 13 shows a sample component diagram with four components and there dependency.The dependency of the four components are shown below.

- **Deployment Diagram:** Deployment diagrams (Flower.M; Matha M.P,2008;Rumbaugh.J and Blaha,2007) are used for describing the hardware components, where software components are deployed. Component diagrams and deployment diagrams are closely related.Component diagrams are used to describe the components and deployment diagrams shows how they are deployed in hardware.Most of the UML diagrams are used to handle logical components but deployment diagrams are made to focus on the hardware topology of a system. Deployment diagrams are used by the system engineers. Figure 14 shows the sample deployment diagram of the case study.

Note: The above diagrams are the sample diagrams for understanding the modelling and designing of the complex IoT systems.

Figure 13. A sample component diagram

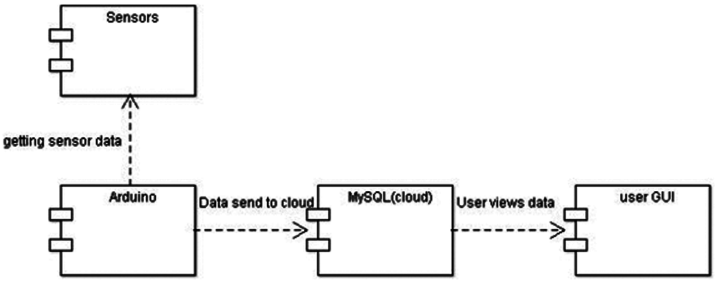
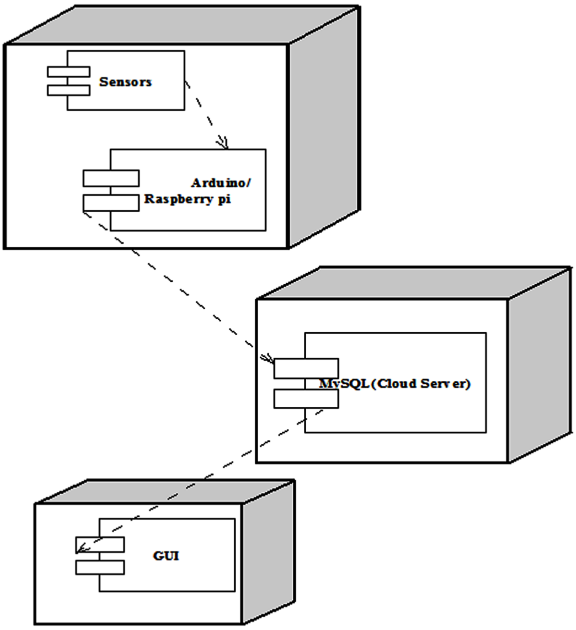


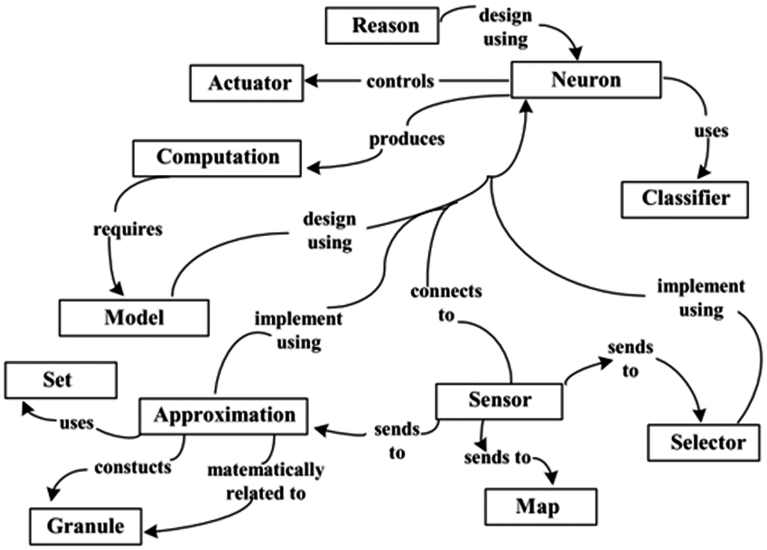
Figure 14. Shows the sample deployment diagram of the case study



INTELLIGENT SYSTEM DESIGN PATTERNS FOR IoT BASED ROBOTIC DEVICES

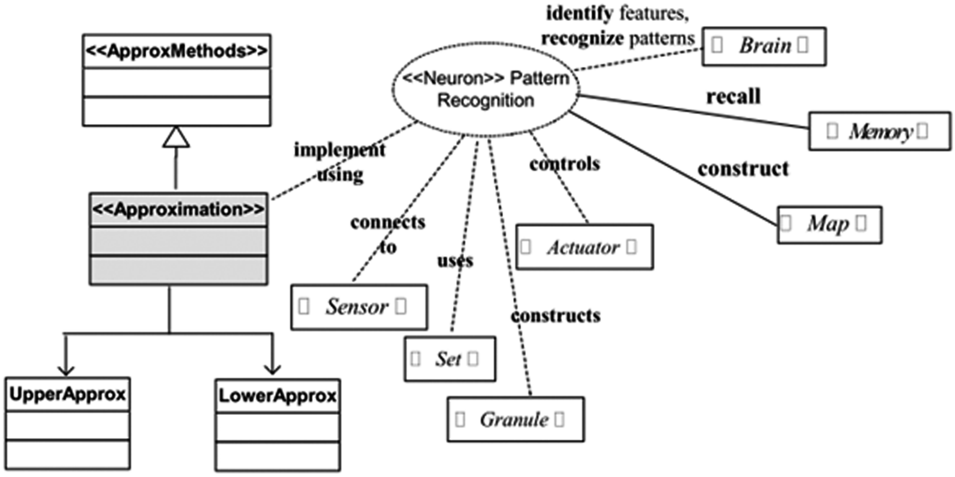
It has been suggested by J.F.Peters in (Peters, 2004, 2003) that the basic object of design is form(Alexander, 1964). A basictenet in intelligent systems engineering is that it is possible to achieve somedegree of match between a pattern (well-understood form of the function andstructure of an IS component) and its context (particular application requiringsome form of sapient-like behavior). In general, a pattern provides aparadigm or typical example that is a model for a collection of individualobjects. In

Figure 15. A partial intelligent system design pattern map (Peters, 2014, 2013)



this work, every IS pattern is a collection of classifiers that can beconsidered either structurally (diagram of the pattern) and functionally (actions defined by methods belonging to each class in the pattern). In somesense, an IS pattern provides a means of satisfying a requirement for thedesign of a component in a sapient-like machine. The context for an IS system defines an IS design problem to be solved. Itcan be seen that IS patterns have a hierarchical organization, where a patternsuch as Set and Granule are super-patterns relative to the Aggregation ($\sum$, $\int$), Approximation and Change (Δ, d, ∂) patterns. A partial IS pattern catalogueand corresponding map is given in Figure 15. The map in Figure 15 is organizedrelative to how each pattern (represented by a named box) relates to otherpatterns in an IS system. The sensors in an intelligent system provide a basisfor a Sensor pattern (this includes Filter and Noise subclasses) that is a superpatternof (has a link to) what are known as Approximation, Measurement, Selector, and Map patterns. The Set pattern is a basic pattern in the design ofintelligent systems. This pattern has a Methods interface which can beimplemented in different ways (e.g., Zermelo-Fraenkel axiomatic set theory,fuzzy sets (Zadeh, 1965) rough sets(Pawlak, 1991) depending on the interpretation of the inclusionoperation. In this paper, the Set and Granule patterns are modelled withrespect to rough sets in the context of a calculus of granules and measuresuseful in approximate reasoning (Skowron andStepaniuk, 1998),(Peters et al.2002) The Approximation design pattern isfundamental in providing a basis for reasoning and decision-making by an intelligent system.

Figure 16. Neuron pattern (Peters, 2004, 2013)



The sensors in an intelligent system provide a basis for a Sensor pattern (this includes Filter and Noise subclasses) that is a super-pattern of (has a link to) what are known as Approximation, Measurement, Selector, and Map patterns. The «Neuron» pattern is a basic pattern in the design of intelligent systems (see Figure 16). This pattern is modeled as a collection of classifiers, each with its own role in a collaboration that makes a particular form of pattern recognition possible. The «Approximation» classifier in Figure 16 has an «ApproxMethods» interface which can be implemented in different ways (e.g., rough sets(Zadeh, 1965)(Pawlak, 1991) depending on the interpretation of the inclusion operation. In this paper, the Set and Granule patterns are modeled with respect to rough sets in the context of a calculus of granules and measures useful in approximate reasoning (Skowron and Stepaniuk, 1998). Notice that a model of the «Approximation» pattern (modeled statically as a class diagram) is also included in Figure 8 This pattern provides a basis for approximate reasoning and decision-making (see, e.g., (Skowron and Stepaniuk, 1998)).

Some Definitions

- **Rough Sets:** Rough set theory(Pawlak.Z, 1991) initially introduced by ZdzisławPawlak during the early 1980s and further developed over the last 25 years provides an approach to approximation of sets that leads to useful forms of granular computing. The basic idea is to discover to what extent a given set of objects (e.g., pixel windows in an image) approximates another of set of objects of interest. Objects are compared by considering their descriptions].

Rough set theory offers a novel approach to manage uncertainty that has been used for the discovery of data dependencies, importance of features, patterns in sample data, feature space dimensionality reduction, and the classification of objects.

- **Set Approximation:** An equivalence relation induces a partitioning of the universe. These partitions can be used to build new subsets of the universe. Subsets that are most often of interest have the same value of the outcome attribute. (see Skowron. A and Stepaniuk J,1998)

CONCLUSION AND FUTURE WORK

Despite the large number of research works that attack specific problems related to the design and development of IoT applications and services, a general software engineering approach is still missing. This chapter, with reference to Franco Zambonelli and his research in IoT - Software Engineering framed some key conceptual abstractions revolving about the IoT universe, can represent a first small step towards a general discipline for engineering IoT systems and applications. As IoT technologies mature, and real-world experiences accumulate, more research in the area of software engineering for IoT systems will be needed, possibly exploiting contaminations with the related areas of agent-oriented software engineering and software engineering for self-adaptive and self-organizing systems, and eventually leading to the identification of a widely accepted general methodology – and associated tools – for the IoT-oriented software engineering. A sample UML diagrams have been used to justify the title and in future more diagrams will be introduced in a broader perspective.It's not necessary to incorporate all the diagrams for all the applications .For simple system only use case diagram, class diagram and only one interaction diagram is sufficient .When classes has significant state than use state chart diagram. If several hardware components are present than deployment diagram is preferable.

ACKNOWLEDGMENT

The authors acknowledge the pioneer work carried by Franco Zambonelli in IoT and software engineering. The authors also acknowledge J.F Peters for his contributions in the area of designing intelligent systems design patterns using UML modelling.

REFERENCES

Alexander, C. (1964). *Notes on the Synthesis of Form*. Cambridge, MA: Harvard University Press.

Bahga, A., & Madisetti, V. (2015). *Internet of Things-A Hands on approach*. University Press.

Budida, D. A. M., & Mangrulkar, R. S. (2017). Design and Implementation of Smart HealthCare System Using IoT. *International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*. 10.1109/ICIIECS.2017.8275903

Buyya, R., & Dastjerdi, A. V. (2016). *Internet of Things: Principles and Paradigms*. Retrieved from http://www.buyya.com/papers/IoT-Book2016-C1.pdf

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (2003). *Design Patterns: Elements of Reusable Object-Oriented Software with Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process by 2003*. Academic Press.

Giusto, D., Iera, A., Morabito, G., & Atzori, L. (Eds.). (2010). The Internet of Things. Springer.

Hassanalieragh, M., Page, A., Soyata, T., Sharma, G., Aktas, M., Mateos, G., … Andreescu, S. (2015). Health Monitoring and Management UsingInternet-of-Things (IoT) Sensing with Cloud-based Processing: Opportunities and Challenges. *IEEE International Conference on Services Computing*, 285-292.

Heuer, J., Hund, J., & Pfaff. (2015). Toward the web of things: Applying web technologies to the physical world. *Computer, 48*(5), 34–42.

Islam, S. M., Raizul, Kwak, D., Kabir, M. D., Humaun, Hossain, M., & Kwag, K-S. (2015). The Internet of Things for Health Care: A Comprehensive Survey. *IEEE Access, 3*, 678-708.

Mall, R. (2014). *Fundamentals of Software Engineering (4th ed.)*. PHI-Delhi.

Matha, M. P. (2008). *Object-Oriented Analysis and Design using UML*. Delhi: PHI-N.

Miles, R., & Hamilton, K. (2006). *Learning UML 2.0*. O'Reilly.

Minerva, R. (2016). *IoT and its Challenges*. Retrieved from https://iot.ieee.org/images/files/pdf/iot_and_its_challenges_roberto_minerva.pdf

Mora, H., David, G., Terol, R. M., Azorin, J., & Szymanski, J. (2017). An IoT-Based Computational Framework forHealthcare Monitoring in Mobile Environments. *Sensors (Basel)*, *17*(10), 2302. doi:10.339017102302

Pawlak, Z. (1991). *Rough Sets: Theoretical Aspects of Reasoning About Data*. Boston, MA: Kluwer Academic Publishers. doi:10.1007/978-94-011-3534-4

Peters, J. F. (2003). Design Patterns in Intelligent Systems. In N. Zhong, Z.W. Ras, S. Tsumoto, & E. Suzuki (Eds.), Foundations of Intelligent Systems, Lecture Notes in Artificial Intelligence 2871 (pp. 262-269). Springer.

Peters, J. F. (2004). Approximation space for intelligent system design patterns. *Engineering Applications of Artificial Intelligence*, *17*(4), 393–400. doi:10.1016/j.engappai.2004.04.012

Peters, J. F., Skowron, A., Stepaniuk, J., & Ramanna, S. (2002). Towards an ontology of approximate reason. *Fundamenta Informaticae*, *51*(1), 2, 157–173.

Prehofer, C., & Chiarabini, L. (2017). *From IoT Mashups to Model-based IoT*. Retrieved from https://www.w3.org/2014/02/wot/papers/prehofer.pdf

Pressman, R. S. (1998). Can Internet-Based Applications Be Engineered? Issue No. 05 - September/October: Vol. 15. DOI Bookmark. http://doi.ieeecomputersociety.org/10.1109/MS.1998.714869

Pressman, R.S. (2009). *Software Engineering: A Practitioner's Approach*. McGraw Hill, Intl edition.

Rao, N. J. M. (2017). *IoT based Remote Patient Health Monitoring System (Master's thesis)*. Kansas State University.

Ray, P. P. (2016). *A Survey on Internet of Things Architecture" Journal of King Saud University –Computer and Information Sciences, 1319-1578*. Elsevier.

Razzaque, M. A., Milojevic-Jevric, M., Palade, A., & Clarke, S. (2016). Middleware for internet of things: A survey. *IEEE Internet of Things Journal*, *3*(1), 70–95. doi:10.1109/JIOT.2015.2498900

Rumbaugh, J., & Blaha. (2007). *Object Oriented Modelling and design with UML-2* (2nd ed.). Pearson.

Skowron, A. (2001). Toward intelligent systems: Calculi of information granules. Bulletin of the International Rough Set Society, 5(1/2), 9-30.

Skowron, A., & Stepaniuk, J. (1998). Information granules and approximation spaces. *Proc. of the 7th Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-based Systems (IPMU'98)*, 1354-1361.

Skowron, A., Stepaniuk, J., & Peters, J. F. (2001). Hierarchy of information granules. In H.D. Burkhard, L. Czaja, H.S. Nguyen, P. Starke (Eds.), *Proc. of the Workshop on Concurrency, Specification and Programming* (pp. 254-268). Academic Press.

Sommerville, I. (n.d.). *Software Engineering* (7th ed.). Pearson Education Publication.

Sotiriadis, S., Petrakis Euripides, G.M., Covaci, S., Zampognaro, P., Georga, E., & Thuemmler, C. (2013). *An architecture for designing Future Internet (FI) applications insensitive domains: Expressing the Software to data paradigm by utilizing hybrid cloud technology*. DOI:. doi:10.1109/BIBE.2013.6701578

Weyrich & Ebert. (2016). *Reference Architectures for the Internet of Things. IEEE Software, 33(1).*

Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, *8*(3), 338–353. doi:10.1016/S0019-9958(65)90241-X

Zambonelli, F. (2006). *Towards a General Software Engineering Methodology for the Internet of Things*. Academic Press.

Zambonelli, F. (n.d.). *Towards a discipline of IoT-Oriented Software Engineering*. Retrieved from https://zdoc.site/towards-a-discipline-of-iot-oriented-software-engineering.html

Chapter 4
Web-Based IoT Application Development

S. Gopikrishnan
Karpagam College of Engineering, India

P. Priakanth
Kongu Engineering College, India

ABSTRACT

Wireless sensor network (WSN) is an outdated technology that is used to monitor the physical changes in environment and take necessary actions. The advancement in WSN leads to automation in physical environment by uploading the sensed data to internet or cloud. The internet of things concept deals with the issues of making things connected to the internet as well as in a network of smart devices. IoT application development presents an enormous opportunity to reshape entire industries. According to McKinsey & Co, the merging of the physical and digital worlds via IoT could generate up to $11.1 trillion a year in economic value by 2025. Hence, the development of the web-based IoT applications will take automation research to the next level. Many authors have proposed many solutions to make internet of things possible in day-to-day life. This chapter gives an introduction about the web-based application development based on internet of things. The major objective of this chapter is to discuss and resolve the challenges in IoT to automate the real-time problems.

DOI: 10.4018/978-1-5225-7790-4.ch004

INTRODUCTION

Wireless Sensor Network (WSN) is an outdated technology which is used to monitor the physical changes in environment and take necessary actions. The advancement in WSN leads to automation in physical environment by uploading the sensed data to internet or cloud. Internet of Things is not actually new it's been around for quite some time back in 2010. For example, printer that was connected to the computer over a Wi-Fi, media player connected to the headset over Bluetooth and there were other gadgets that were essentially connected to either mobile phones or to other devices. Today from smart lighting to smart thermostats, connected cameras to environmental sensors, personal voice assistants to variables there are Internet of Things and connected devices everywhere. A different end centre of this revolution is the web which essentially enabled us to connect these devices to each other through a centralized location and with the ability to acquire data store it, process it, query it and analyze it in the cloud (Taivalsaari et al, 2017;l Yao et al, 2015). So these rails of web and the need for connected devices brought us into the era of Internet of Things.

The concept internet of things deals with the issues of making things connected to the internet as well as in a network of smart devices. IoT application development presents an enormous opportunity to reshape entire industries. According to McKinsey & Co, the merging of the physical and digital worlds via IoT could generate up to $11.1 trillion a year in economic value by 2025. Hence, the development of the web based IoT applications will take automation research into next level. Many authors have proposed many solutions to make Internet of Things possible in day today life. This chapter gives the introduction about the web based application development based on internet of things. The major objective of this chapter is to discuss and resolve the challenges in IoT to automate the real time problems.

IOT Architecture

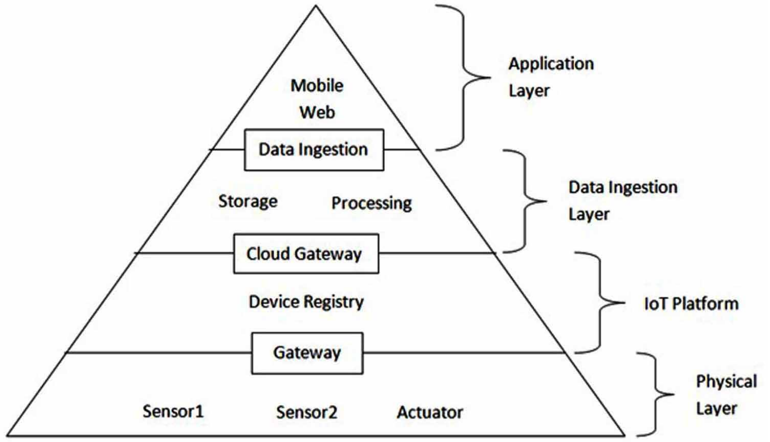
This section takes a closer look at the architecture of IoT. As depicted at the big picture of IOT shown in Figure 1, it is a four-tier architecture. The first tier is the devices and second tier is the IOT platform which is a middleware for connecting the devices. The third tier is the Data Ingestion layer for processing the data at the web and then the top tier is the applications tier. In physical layer, irrespective of how the devices are connected, they establish a connection with the IOT platform. Their multiple building blocks handle the workflow and take care of the connectivity and processing of various data sets and data streams.

The outcome of the IOT platform is consumed and visible through the application tier (Gyrard et al, 2016; Patel et al, 2015). For example the data that is streamed by the devices to the IOT platform is available as a dashboard across your television, web application, mobile or variable. At the same time we can use any of these front-end applications to control one of the devices. For example it is possible to pull out our mobile phone and control a device that is connected to the IOT platform. The detailed explanation about these layers has been described further.

Physical Layer (Devices)

The devices tier consists of various devices both intelligent devices and legacy devices. Intelligent devices can make a connection to the outside world over the Internet. Non intelligent and legacy devices take help of a proxy which is called as the local gateway. These hardware devices are of three types. The first is the IP capable device which can establish connectivity to the IOT platform over Internet. It uses higher end protocols like TCP, HTTP or Web Sockets. Then there are non IP capable devices which use low power memory constrained devices and they cannot establish a connection over TCP or HTTP. Hence they rely on very lightweight protocols like ZigBee, Bluetooth, Low Energy z-wave or power on Power over Ethernet. The third type is legacy devices that have been a part of manufacturing equipment and automobile industry for a long time and they rely on legacy technologies like SCADA-RTU, PA and PLC. These are the devices that are essentially connected to the physical equipment or physical switches that wants to operate and they're also capable of acquiring data through the sensors and sending them to the next stage.

Figure 1. IoT layered architecture



IoT Platform (Gateways)

The legacy devices that are not capable of talking to the platform directly take help of the Gateway. Gateways are of two types. The first is called the field gateway which is essentially representing a collection of devices or an array of devices that are not able to establish connectivity to the central cloud or an IOT platform. The field gateway that is also called as IOT edge gateway is responsible for representing those devices with the cloud platform. The gateway registers each device with unique identifier and maintains a device registry for all the connected devices metadata like the serial number, model number, firmware version etc.

The Gateway also enables machine to machine communication and it is responsible for orchestrating the messages flowing back and forth between devices. The Gateway also handles security and it authorizes devices when they connect, when they publish and subscribe to messages. Devices can be white listed or blacklisted in case one of the devices gets stolen or gets corrupted. Gateways typically expose a set of disparate protocols including XMTP, MQTT, CoAP and web sockets.

IoT Platform (IoT Protocols)

Three major real-time protocols are used by IoT devices are XMPP, CoAP, and MQTT. Interestingly enough, XMPP started life as Jabber, an open instant messenger protocol.

- **XMPP:** The eXtensible Messaging and Presence Protocol (XMP P) is a TCP communications protocol based on XML that enables near-real-time exchange of structured data between two or more connected entities. Out-of-the- box features of XMPP include presence information and contact list maintenance. While both features were originally designed for instant messaging, they have obvious applications for IoT. Due in part to its open nature and XML foundation, XMPP has been extended for use in publish-subscribe systems -- again, perfect for IoT applications.
- **CoAP:** The Constrained Application Protocol (CoAP) was specifically developed to allow resource-constrained devices to communicate over the Internet using UDP instead of TCP. Developers can interact with any CoAP-enabled device the same way they would with a device using a traditional REST-based API. CoAP is particularly useful for communicating with low-power sensors and devices that need to be controlled via the Internet. CoAP is a simple request/response protocol (again, very similar to REST) that follows a traditional client/server mode l. Clients can make GET, PUT, POST, and DELETE requests to resources. It uses bit fields as packets to maximize

memory efficiency, and they make extensive usage of mappings from strings to integers to keep the data packets small enough to transport and interpret on-device. The Constrained of Application Protocol (COAP) is used which employs based on the User Datagram Protocol (UDP).

- **MQTT:** Message Queue Telemetry Transport (MQTT) is publish-subscribe messaging protocol. Similar to CoAP, it was built with resource-constrained devices in mind. MQTT has a lightweight packet structure designed to conserve both memory usage and power. A connected device subscribes to a topic hosted on an MQTT broker. Every time another device or service publishes data to a topic, all of the devices subscribed to it will automatically get the updated information. The major advantages of MQTT are publish subscribe message queue and the many-to-many broadcast capabilities. Using a long-lived outgoing TCP connection to the MQTT broker, sending messages of limited bandwidth back and forth is simple.

Data Ingestion Component

The third building block of an IOT stack is the data ingestion component. The data ingestion layer is responsible for acquiring the data from tens of thousands and sometimes when millions of devices and passing that stream of data to multiple parties that are subscribed to the data injection endpoint. It can be done in few seconds with technologies like Apache Kafka which is one of the popular open-source data ingestion engine. It enables one-way communication between devices and platform and need to connect only those devices that are generating data and that needs to be acquired in real time. So those devices typically get connected to the IOT platform via the data ingestion layer.

Storage

Once the data passes through the ingestion layer, it is going to be stored for further processing. It is highly recommended to use a combination of object storage, NoSQL databases, the time series databases and relational databases. The choice of database depends on the use case, but essentially the idea is to store the raw data as well as store the process data which is going to be useful in deriving insights. There is also a concept called Data Lake which is going to store the data in its raw form. It is primarily used for tracking and performing audit trails on the incoming data stream. Both the raw data and the data from real-time analysis are also stored in the same storage which is exposed as a REST API for applications to consume. This REST API will be used to generate visualizations and dashboards that that is useful for decision makers to get insights into the platform or the connected devices.

Hot Path Analytics

From storage data typically passes through one of the channels called hot path analytics where the data can be transform, process, query and analyze. An example is a connected car that is sending the current latitude and longitude. The process is to acquire the data in real time and call another web service that will transform the latitude/longitude get a point into a zip code. If in case there is a problem form with the car it uses that data to guide the driver to the nearest service centre and all this happens in real time. Therefore, the data points that go through the real time stream analysis is also called as complex event processing engine or hot path analytics. It process data points as they come using concepts of sliding window and tumbling window.

Cold Path Analytics

Data can also be stored and processed over a period of time and this typically takes place as a batch process and that is used for deriving historical trends and capturing insights from data that has been processed and maintained over a period of time. For example, the same connected car can send the fuel consumption details every hour but you can't really make out the fuel efficiency of a car by processing it as it comes. Instead if this data is accumulated over a period of time and compare it with the number of kilometres travels, a correlation between the number of kilometres is covered and the fuel consumption and then arrive at the fuel efficiency index for that car. The companies like uber could actually use that to find out the most efficient fuel efficient car in their fleet and the least fuel efficient car in the in the fleet. The data that is captured for long term processing and historical trends goes through this path which is called the cold path.

Application Layer

The role of an application layer is of twofold. The first is to get the user experience in terms of the visualizations, dashboards and real-time graphs that indicate the current state of the devices. More operational teams would watch these very busy graphs and slides and dashboards to basically make sense of the current state of the devices.

The second role of an application layer is to send commands to the devices. For example when a specific switch or equipment is to be controlled or an actuator, it has to be done from application. So the application tier acts as the front-end for controlling the devices as well as consuming the insights and presenting that in the most intuitive form. All these are the layers of a typical IOT platform and this is how an enterprise IOT stack is built from the ground up.

THE CHALLENGE OF IOT APPLICATION DEVELOPMENT

IoT web application development requires many, disparate technologies (Alessi et al, 2016; Botta et al, 2016). IoT solutions are typically composed of a complex, heterogeneous mix of IoT endpoints, platforms, back-end systems and data (e.g. sensors, actuators, processors, embedded software, local and long-range connectivity, middleware, apps, analytics, machine learning, etc.).

In addition, building IoT applications requires scarce, hard-to-find specialist skills (Belli et al, 2015; Kao et al, 2017). Developers must not only be versed in their organization's IoT platform and its underlying services but the big data and machine learning technologies required to make sense of real-time data streams. At the same time, because IoT represents unchartered territory, they must collaborate with the business to experiment with new ideas and bring new solutions to market through rapid iteration. These all result in a lot of time and effort to build IoT applications. On top of that, the accelerating pace of change is making it hard for enterprise IT teams to keep up with new capabilities and advancements.

EXISTING WEB APPLICATIONS FOR IOT

In order to address the above mentioned challenges, various approaches have been proposed (for a detailed discussion of various systems available for application development, refer (Patel et al, 2015)). One of the approaches is node-centric programming (Whitehouse et al, 2004; Roman et al, 2002; Costa et al, 2007) allows for the development of extremely efficient systems based on complete control over individual devices. However, it is not easy to use for IoT applications due to the large size and heterogeneity of systems. In order to address node-centric programming limitation, various macro programming systems (Pathak et al, 2007; Bischoff et al, 2007) have been proposed. However, most of macro programming systems largely focus on development phase while ignoring the fact that it represents a tiny fraction of the application development life-cycle. The lack of a software engineering methodology to support the entire application development life-cycle commonly results in highly difficult to maintain, reuse, and platform-dependent design, which can be tackled by the model-driven approach. To address the limitations of macro programming systems, approaches based on model-driven design (MDD) have been proposed (France et al, 2007; Mellor et al, 2003; Kulkarnie et al, 2003). Major benefits came from the basic idea that by separating different concerns of a system at a certain level of abstraction, and by providing transformation engines to convert these abstractions to a target code, productivity (e.g., reusability, maintainability) in the application development process can be improved.

DEVELOPING AN IOT DEVICE OR DISTRIBUTED IOT SERVICE

From a development standpoint, creating IoT devices hinges on embedded programming (Taivalsaari et al, 2017; Vujovic et al, 2015). There are both software and hardware angles to consider when creating an IoT prototype—the small computer embedded in the object or device, and the software that makes it run (Komaki et al, 2017). As mentioned above, this includes things like wearable, connected home devices, circuit design, GPS programming, 3D design, and more. Fortunately, many of these software systems and software development kits (SDKs) now use programming languages and operating systems that engineers already use for mobile and web development, which opens the field up to many more developers.

When a fully fledged distributed IoT service is created, there are many angles to consider: development of the embedded device itself, the IT and networking services that power it, data and analytics, and design and development of an integrated UI (e.g., a mobile app to control your home's thermostat).

The following things have to be followed:

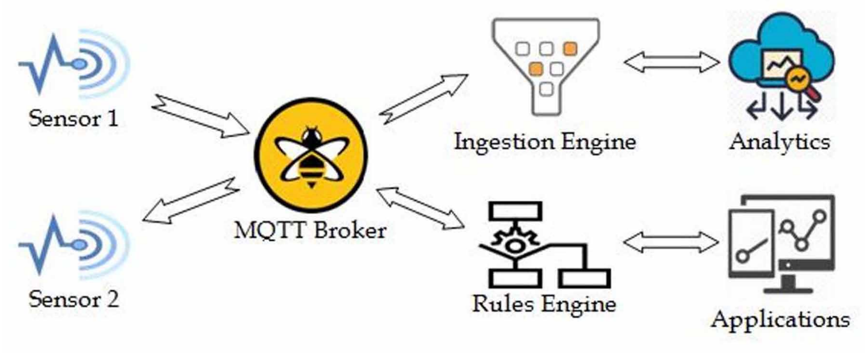
- Choose your hardware platform (i.e., your processing board)
- Develop the application software, including any back-end and networking support
- Create the integrated UI
- Develop the APIs, beacons, web sockets, and procedure calls that enable the high-level communications that occur between devices
- Establish security, data storage, and analytics measures

WEB PROTOTYPE FOR AN IOT APPLICATION

This section gives a sense of how to build a simple end-to-end complete IOT solution. When the most essential components of an IOT solution is analyzed, devices are at one end of the spectrum and analytics and applications are at the other end of the spectrum. Figure 2 shows atypical web prototype for an IoT Application.

The devices generate a lot of data and are used from the analytics or the application tier. The application can control any of the devices that are connected to the MQTT broker or to the platform. The devices that contact to the platform are actually talking to a centralized orchestration engine called the MQTT brokers. MQTT is the gateway and m2m orchestration engine which will facilitate the messaging across multiple devices. It uses a pub/sub model where a device will publish its data to a topic and any interested party subscribe to the same topic to receive that message. On the other side of the intuitive, two more subscribers are there. The first is the ingestion

Figure 2. Typical web prototype of an IoT application



engine which is responsible for acquiring the data that is entering the MQTT broker. It can be configured that to get all the messages or filter on a specific topic and get selective messages.

The second is the rules engine which is also acting as a subscriber to the impurity broker. Its job is to get the data and check for certain rules or business logic which will either control the device or will invoke an application based on an alert or a notification. For example device1 could be a smart thermostat device2 could be an air conditioner. when the smart thermostat starts publishing the current temperature it is going to be received by the rules engine and there could be a simple rule that says if the temperature falls below 20 degrees Celsius switch on the a/c. so that rule is defined in the rules engine and when the threshold is met or the rule condition evaluates to true, it talks back to the impurity broker and intuitive broker sense and message to the air conditioner to turn itself on. This workflow is managed by the rules engine as well as ingestion engine. The ingestion engine is a layer that decouples data from rest of the consumers. Finally, the analytics layer is responsible for helping us visualize the aggregated data or the outcome of the processing. These are the most essential but complete set of building blocks of an IOT.

OPEN SOURCE TECHNOLOGIES FOR WEB BASED IOT

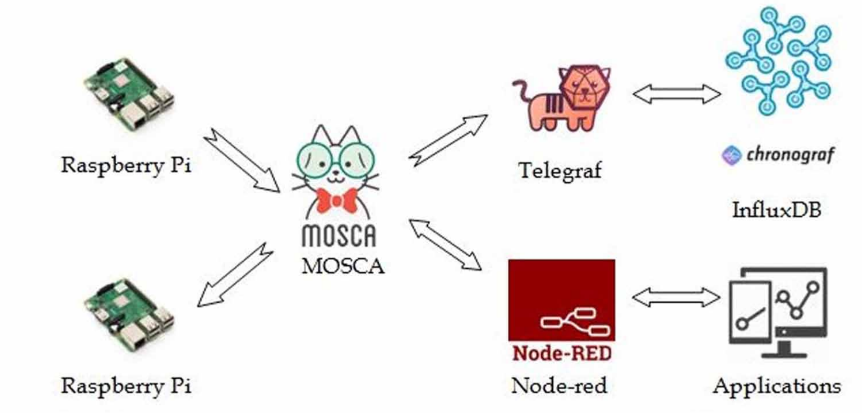
This section discusses the real time demo of the web based IOT application. Figure 3 represents the Open source tools for web application development for IoT (Soldatos et al, 2015). Take two Raspberry Pi devices that are connected to a bunch of sensors and actuators and also LEDs and these are completely independent. First Raspberry Pi which is responsible for receiving the data so it is called as a subscriber. This has two LEDs one is green and the other one is the red. Depending on a rule or a condition it will turn on and turn off these LEDs. It is completely independent and connected to a power bank.

The Raspberry Pi which is the publisher and it has two components. One is a sensor which is called the dht11. Dht11 is responsible for streaming the temperature and humidity via Raspberry Pi. The other one is a switch which is also like a sensor but it is more analogue. So it either senses zero or one based on the state.

With these hardware devices, this demo will show how to develop a web based IOT application (Grgic et al, 2016; Guinard et al, 2016). Assume that publisher raspberry Pi generates the data or sends the signal. Another Raspberry Pi acts as a subscriber. Now those two devices are connected to an open source MQTT broker called Mosca. Mosca is an open source written in node.js and it is available on github. It's also packaged as a docker image. Thus Mosca is used as the MQTT broker for connecting our raspberry PI's acts as publisher and subscriber.

The messages that flow into Mosca are consumed by another open source ingestion engine called Telegraph. As the data enters Telegraph, it can flow into multiple channels. In this case, it is stored in open source software called influxDB which is a time series database. The output of influx DB is going to a visualization front end called Cronograf. Cronograf is a very powerful dashboarding tool that can grab the data from influxDB and helps us to visualize. The data processing analytics ETL has been done with the rules engine in the form of node-red which is very powerful and JavaScript can be used to basically create simple rules. This can route the data as it enters and can invoke third-party applications. With this experimental setup, to send messages Twitter to be tested. When the temperature goes up, send a tweet will be sent and it will be monitored account to find how the thermostat is functioning.

Figure 3. Open source web prototype for IoT application



THE STEPS INVOLVED IN WEB BASED IOT APPLICATION

1. Launching a droplet
2. Install and configure Docker.
3. Run Docker images for
    a. Mosca which is the MQTT Broker
    b. Telegraph's used as Ingestion Engine
    c. InfluxDB used as Time Serious DB
    d. Cronograf which is a Dashboard
    e. Node-Red used to configure rules.
4. Connect devices to Mosca.
5. Configure the rules in Node-Red
6. Start visualizing the data in Cronograf

Configure an Open Source IOT Stack

The first step is to launch a droplet in disclosure. The Figure 4 shows the initial configuration of the droplet which uses the thing server which is a VM server to host the IOT stacks. The first step is to grab the IP address and SSH into the specified address part. The second step is to configure the Docker. The publisher Raspberry Pi has to be logged on which is going to act as the publisher. This is running Linux on the Raspberry Pi it's running on ARM architecture arm v7. Screenshot has been given in Figure 5.

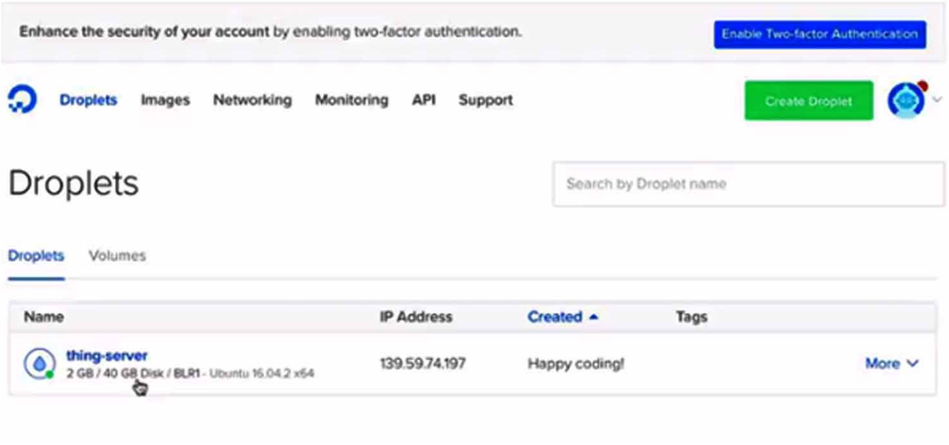
The other subscriber Rasberry Pi has to be logged in another terminal. It is also running a flavour of Linux. The Figure 6 shows the screenshot of the subscriber Pi.

Raspberry Pi acts as a publisher another one acts as a subscriber. Centralized server is available. With this, the infrastructure setup has been over. The stack has to be built. Some directories have to be created called as Mosca, influxDB, Telegraf which is used to store some data files as containers. A special directory is no need for Node-Red since no storage is required. Docker network is created. This is a very important step that all the containers that is to be launched to the same overlay network in a shared network for all the containers. Figure 7 shows the screenshot of creating a docker network. Here the overlay network is created as IOT.

Launching the Broker

The next step is launching the broker which is the MQTT broker. So the docker run command is very familiar. Both 1883 and port 8080 are expected. The port 1883 is basically the MQTT standard port and 8080 is the HTTP port. So data can be sent to this broker by either port 80 which is HTTP or via 1883 which is MQTT. Then

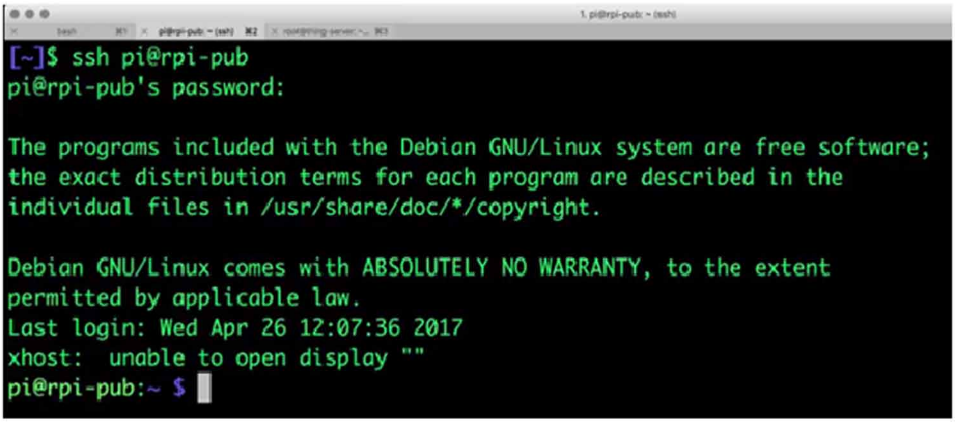
Figure 4. Initial configuration of Droplet



Enhance the security of your account by enabling two-factor authentication.

Name	IP Address	Created ▲	Tags	
thing-server 2 GB / 40 GB Disk / BLR1 - Ubuntu 16.04.2 x64	139.59.74.197	Happy coding!		More ⌄

Figure 5. Publisher Raspberry Pi initiation

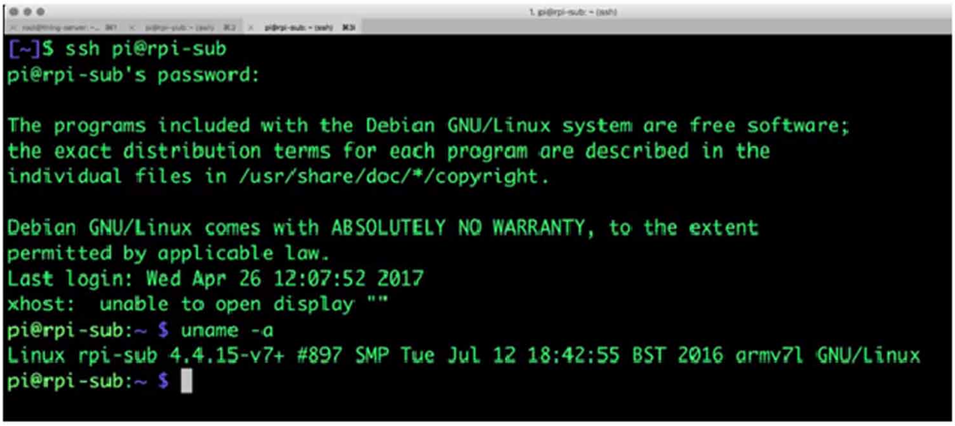


```
[~]$ ssh pi@rpi-pub
pi@rpi-pub's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Apr 26 12:07:36 2017
xhost:  unable to open display ""
pi@rpi-pub:~ $
```

mount the DB directory inside the container as /db and pull this image and run it as shown in Figure 8.

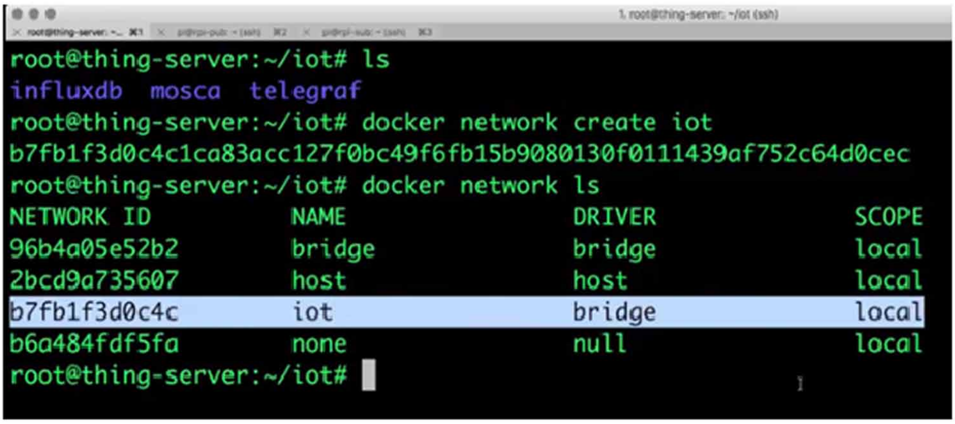
The Mosca container up and running is considered. It is now available on 1883 port. Now connect the MQTT broker to our devices. Go to publisher Raspberry Pi which is connected to the internet via the built-in Wi-Fi. Wi-Fi is used to connect to disclosure droplet. A couple of scripts are created. The first script is the button.py.

Figure 6. Subscriber Raspberry Pi initiation



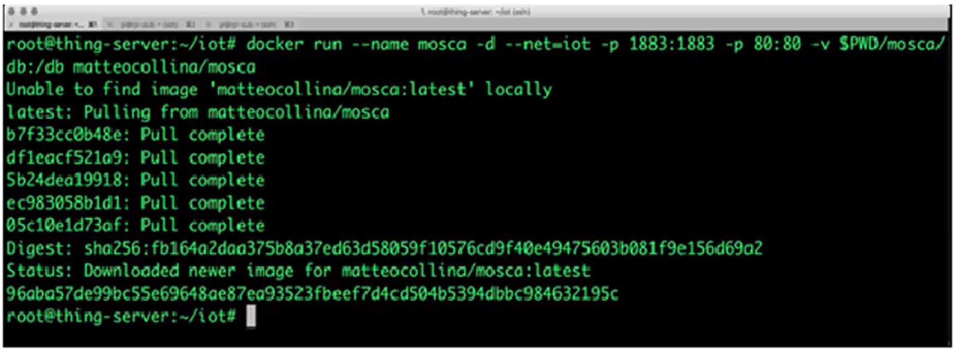
```
[~]$ ssh pi@rpi-sub
pi@rpi-sub's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Apr 26 12:07:52 2017
xhost:  unable to open display ""
pi@rpi-sub:~ $ uname -a
Linux rpi-sub 4.4.15-v7+ #897 SMP Tue Jul 12 18:42:55 BST 2016 armv7l GNU/Linux
pi@rpi-sub:~ $ 
```

Figure 7. Creation of Docker Network



```
root@thing-server:~/iot# ls
influxdb  mosca  telegraf
root@thing-server:~/iot# docker network create iot
b7fb1f3d0c4c1ca83acc127f0bc49f6fb15b9080130f0111439af752c64d0cec
root@thing-server:~/iot# docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
96b4a05e52b2        bridge              bridge              local
2bcd9a735607        host                host                local
b7fb1f3d0c4c        iot                 bridge              local
b6a484fdf5fa        none                null                local
root@thing-server:~/iot# 
```

Figure 8. Setup of MQTT Broker and Mount the DB directory



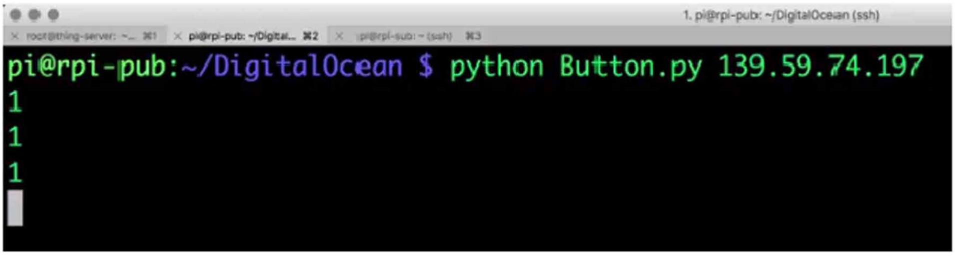
```
root@thing-server:~/iot# docker run --name mosca -d --net=iot -p 1883:1883 -p 80:80 -v $PWD/mosca/
db:/db matteocollina/mosca
Unable to find image 'matteocollina/mosca:latest' locally
latest: Pulling from matteocollina/mosca
b7f33cc0b48e: Pull complete
df1eacf521a9: Pull complete
5b24dea19918: Pull complete
ec983058b1d1: Pull complete
05c10e1d73af: Pull complete
Digest: sha256:fb164a2daa375b8a37ed63d58059f10576cd9f40e49475603b081f9e156d69a2
Status: Downloaded newer image for matteocollina/mosca:latest
96aba57de99bc55e69648ae87ea93523fbeef7d4cd504b5394dbbc984632195c
root@thing-server:~/iot# 
```

Button.py

```
import sys
import time
import grovepi
import paho.mqtt.client as mqtt
button = 3 mqtt_host=sys.argv[1]
grovepi.pinMode(button,”INPUT”) mqttc = mqtt.Client()
mqttc.connect(mqtt_host, 1883)
while True:
try:
button_state=grovepi.digitalRead(button)
if button_state == 1: mqttc.publish(“Buttonl”,’{”value”:
“on”}’) else:
mqttc.publish(“Buttonl”,’{”value”: “off”}’) time.sleep(.5)
print(button_state)
except I0Error:
print (“Error”)
```

It uses paho MQTT library which is the standard way of connecting to an MQTT broker. So, the MQTT is grabbed host via the command line arguments and pass the public IP address of the droplet. This code is tracking the state of the button in a loop. Every time the button state becomes one it sends the value on and every time the button becomes 0 it sends another JSON payload with value off. Then launch this python code with the IP address of our droplet. So that it can register with the MQTT broker. Now it starts sending one. Because the button which is attached with the publisher Pi device is in on position. When turn it off it becomes 0. The Figure 9 shows the subscriber Pi’s output.

Figure 9. Output of publisher Raspberry Pi


1. pi@rpi-pub: ~/DigitalOcean (ssh)
root@thing-server: ~... ⌘1
pi@rpi-pub: ~/Digital... ⌘2
pi@rpi-sub: ~ (ssh) ⌘3

```
pi@rpi-pub:~/DigitalOcean $ python Button.py 139.59.74.197
1
1
1
```

Now the subscriber Pi device is launched which is connected with LEDs using the second python script Bulb.py. The idea is to control the LED remotely from this button.

Bulb.py

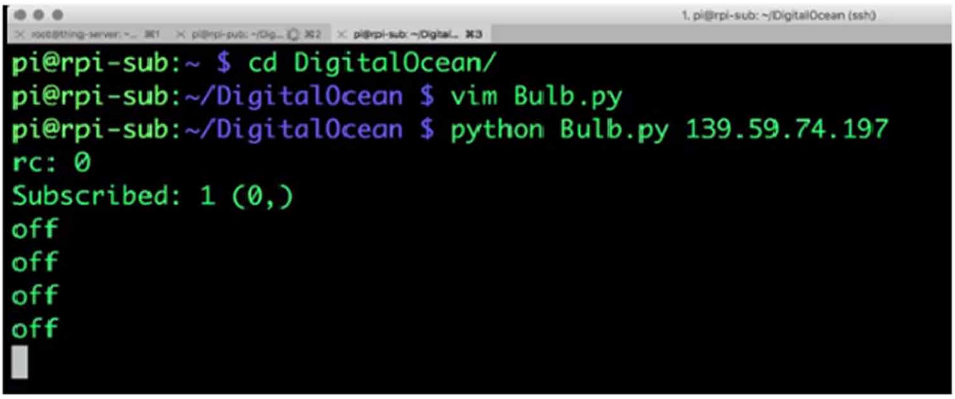
```
import sys
import paho.mqtt.client as mqtt import json
import grovepi
led=4
button=”off”;
mott_host=sys.argv[1]
def on_message(client, userdata, msg): button=json.loads(msg.
payload)[“value”] if button == “on”:
grovepi.digitalWrite(led,l)
else:
grovepi.digitalWrite(led,0) print(button);
def on_connect(mosq, obj, rc):
print(“rc: “+str(rc))
mottc.subscribe(“Buttonl”)
def on_subscribe(mosq, obj, mid, granted_qos):
print(“Subscribed: “+str(mid)+” “+str(granted_qos))
```

So it subscribes to our topic which is called the “Button1” and it grabs the payload. Based on the value it will turn the LED either on or off. The LED is connected to pin4. When launch the bulb.py with the IP address of our droplet, the bulb is going to be controlled from the button. The Figure 10 shows the subscriber Pi’s output.

Figure 11 shows a hardware output of the above two codes. There is a bulb at subscriber Pi node and there is a button at the publisher Pi. When the switch is turned on the red LED it goes off when it is turned off.

The next piece of code which is the sensor with publisher raspberry PI’s which is streaming the humidity and temperature. The code T-Sesnoris considered.py which is basically grabs the data from the sensor and publishes that to Mosca using a specific format.

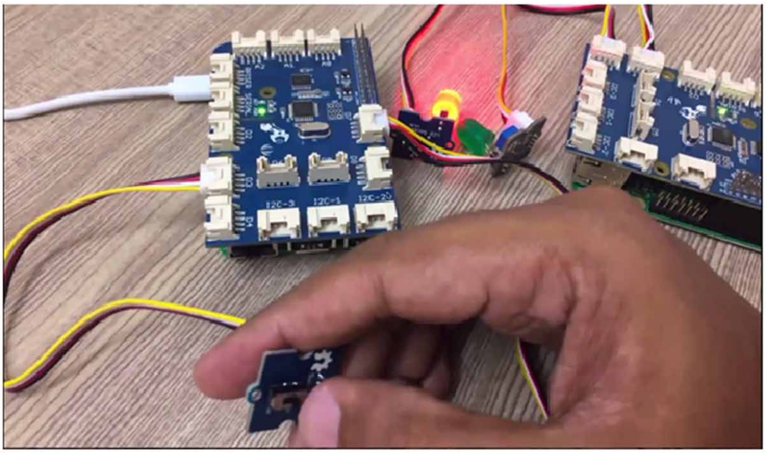
Figure 10. Output of subscriber Raspberry Pi



1. pi@rpi-sub: ~/DigitalOcean (ssh)
root@thing-server: ~... ⌘1
pi@rpi-pub: ~/Dig... ⌘2
pi@rpi-sub: ~/Digital... ⌘3

```
pi@rpi-sub:~ $ cd DigitalOcean/
pi@rpi-sub:~/DigitalOcean $ vim Bulb.py
pi@rpi-sub:~/DigitalOcean $ python Bulb.py 139.59.74.197
rc: 0
Subscribed: 1 (0,)
off
off
off
off
```

Figure 11. Hardware implementation of Raspberry Pi



T-Sensor.py

```
Wusr/bin/env python
import sys
import time
import grovepi
import json
import paho.mqtt.client as mqtt
dht_sensor_port 7 mqtt_host.sys.argv[1]
```

```
mqttc - mqtt.Client() mqttc.connect(mqtt_host, 1883)
while True:
try:
[ t,h ]grovepi.dht(dht_sensor_port,0)
if isinstance(t,float) and isinstance(h,float): mqttc.publish
("environment","temperature,building-1 value."+str(t)) time.
sleep(.5)
print(str(t) + "\t" + str(h) + "\t" + "1.282"); except IOError:
print ("Error")
```

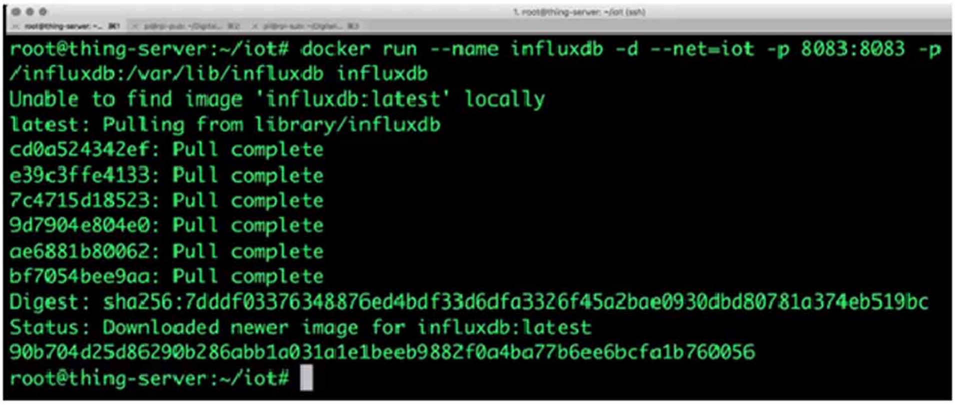
It is used to publish the live temperature to a topic called environment. So temperature is grabbed from the sensor and converting that into a string by adding some metadata like sensor id as a tag.

Integrating the Sensor Data With Web Application

Before it is published, it has to be made sure that storage and time series database are properly configured. The time series database has to be run. So it's going to pull the influx DB image from the docker hub and to have the time series database up and running. The Figure 12 shows the configuration of influxDB.

It has to be confirmed that the influx DB container is also up and running. So before ingesting is started the data to influxDB, it needs to be initialized. To initialize the influxDB, launch another container to access CLI and talk to the influx DB database. After initializing a database called environment is created. This database is going to be the container for our time series database. Now, couple of sample

Figure 12. Configuring influxDB



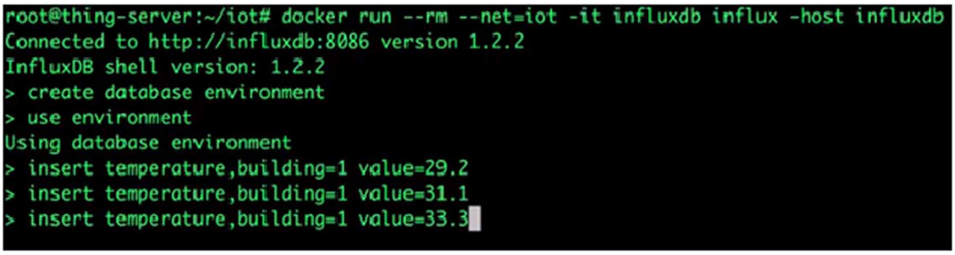
```
root@thing-server:~/iot# docker run --name influxdb -d --net=iot -p 8083:8083 -p
/influxdb:/var/lib/influxdb influxdb
Unable to find image 'influxdb:latest' locally
latest: Pulling from library/influxdb
cd0a524342ef: Pull complete
e39c3ffe4133: Pull complete
7c4715d18523: Pull complete
9d7904e804e0: Pull complete
ae6881b80062: Pull complete
bf7054bee9aa: Pull complete
Digest: sha256:7dddf03376348876ed4bdf33d6dfa3326f45a2bae0930dbd80781a374eb519bc
Status: Downloaded newer image for influxdb:latest
90b704d25d86290b286abb1a031a1e1beeb9882f0a4ba77b6ee6bcfa1b760056
root@thing-server:~/iot#
```

records are inserted as depicted in Figure 13. The first one that is inserted is the collection called temperature and second is building which is a metadata.

The database has been initialized and populated it with some sample data within the container. Now query the rest endpoint of influxDB. InfluxDB is listening on port 8086. If endpoint is hit, a JSON payload is seen as shown in Figure 14.

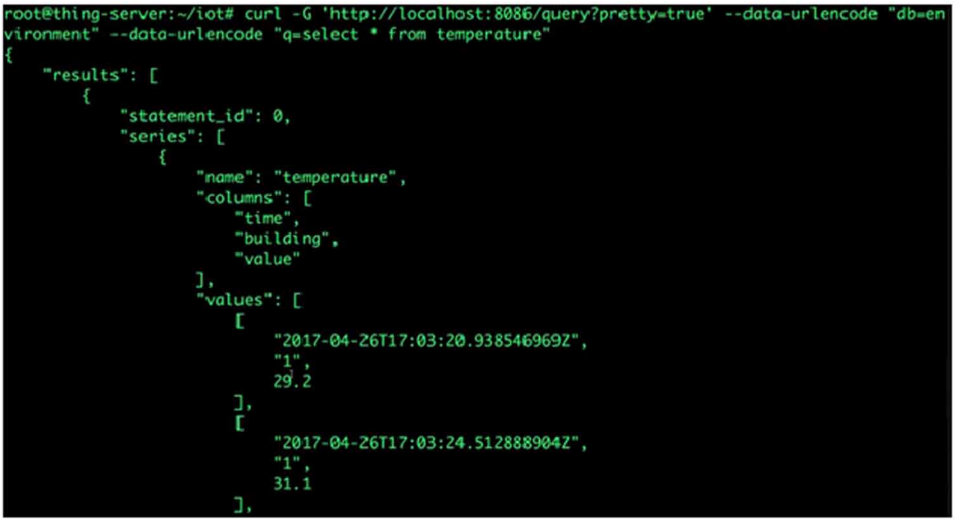
Time series data base is configured, Infiniti broker. Telegraph's is to be configured to connect the entity broker with the time series data base. telegraf.cons needs to be created for configuration file called which acts as the glue between the MQTT broker and in Flex DB. In the configuration file, the output of Mosca is piped as an input to influxDB. The following code shows the telegraf.cons file.

Figure 13. Inserting sample data to influxDB



```
root@thing-server:~/iot# docker run --rm --net=iot -it influxdb influx -host influxdb
Connected to http://influxdb:8086 version 1.2.2
InfluxDB shell version: 1.2.2
> create database environment
> use environment
Using database environment
> insert temperature,building=1 value=29.2
> insert temperature,building=1 value=31.1
> insert temperature,building=1 value=33.3
```

Figure 14. JSON Payload



```
root@thing-server:~/iot# curl -G 'http://localhost:8086/query?pretty=true' --data-urlencode "db=en
vironment" --data-urlencode "q=select * from temperature"
{
    "results": [
        {
            "statement_id": 0,
            "series": [
                {
                    "name": "temperature",
                    "columns": [
                        "time",
                        "building",
                        "value"
                    ],
                    "values": [
                        [
                            "2017-04-26T17:03:20.938546969Z",
                            "1",
                            29.2
                        ],
                        [
                            "2017-04-26T17:03:24.512888904Z",
                            "1",
                            31.1
                        ],
```

telegraf.cons File for Outputs

```
[[outputs.influxdb]]
urls = [“http://influxdb:8086”]
database = “environment”
 retention_policy = “autogen”
precision = “s”
timeout = “Ss”
```

So this is the configuration and there is an output which is influxDB and is basically it is discussed that Telegraph’s to use this URL. InfluxDB and Moscow share the same Network overlay network. Telegraph’s can very easily discover these containers by name. InfluxDB will be resolved because they contact the same network segment. “environment” is the database name and precision is seconds which is a timeout that happens at five seconds.

The input is from MQTT. We are configuring Mosca which is the container name and saying subscribe to environment topic which is where our raspberry pi is going to publish the data. so let’s grab this and create a configuration file as shown below.

```
telegraf.cons file for inputs
# Read metrics from MQTT topic(s)
[[inputs.mg-tt_consumer]]
servers = [“mosca:1883”]
qos = 0
topics = [
“telegraf/host01/cpu”,
“telegraf/+/mem”,
“environment/#”
]
client id = “”
data format = “influx”
```

Telegraph container is to be launched and map the con file to the con file inside the container in a read-only mode. This will initialize Telegraph as it comes up and it instructs telegraphs to take the input from MQTT and feed the output to influxDB. Docker hub is pulled. Three containers the Mosca activity broker, influxDB for time series database and Telegraph as the ingestion engine are available. Our next step is to launch the dashboarding tool which is called the chronograph. Chronograph is listening on port 10000. Launch this standalone utility which is going to help us visualize the data. Within another minute it pulls the containers and visualizes the streaming data.

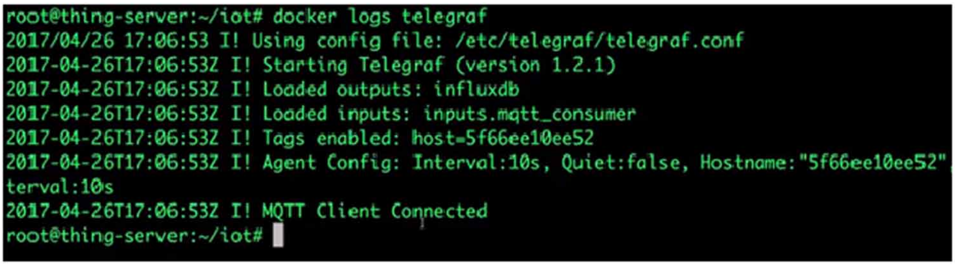
Execution

Start streaming the temperature data by calling the sensor dot pipe with the IP address of our droplet. This will start sending the temperature, humidity and the electricity consumption. The logs the darker logs are to be checked followed by Telegraph's and this tells us that the agent is configured and the MQTT client is connected. Figure 15 is an indication that Telegraph is able to mediate the data flow between Mosca and influxDB.

Another browser has to be chosen and hit for 10000 on our droplet and that immediately shows us the influx DB configuration of chronographs. A new server is to be added with the host IP address of the droplet and visualization is added. this temperature is to be checked and pull the live data coming from our sensor into our dashboard coming via influx DB. Figure 16 shows the visualization of temperature at the dashboard tool.

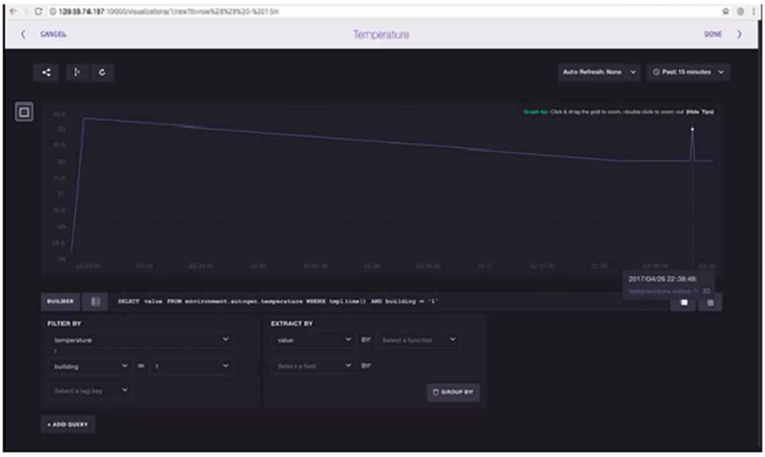
Data pipeline is created it's not a very sophisticated pipeline but nevertheless injection engine time series data base and front-end to visualize data are available. Rules engine has to be configured which is Node-Red. So, the node-red container is launched and that's going to listen on port 1880. It is connected to the same overlay network called IOT. Node read from the browser is accessed and a couple of rules are created. The first rule is to control the colour of the LED based on the temperature. The second rule that will configure is to tweet when the temperature goes beyond the specific threshold. First step is to connect node red as a subscriber to our MQTT broker which is Mosca. Configuration is given in Figure 27 and it has to be followed.

Figure 15. Verification of all configurations of Docker, Mosca and influxDB



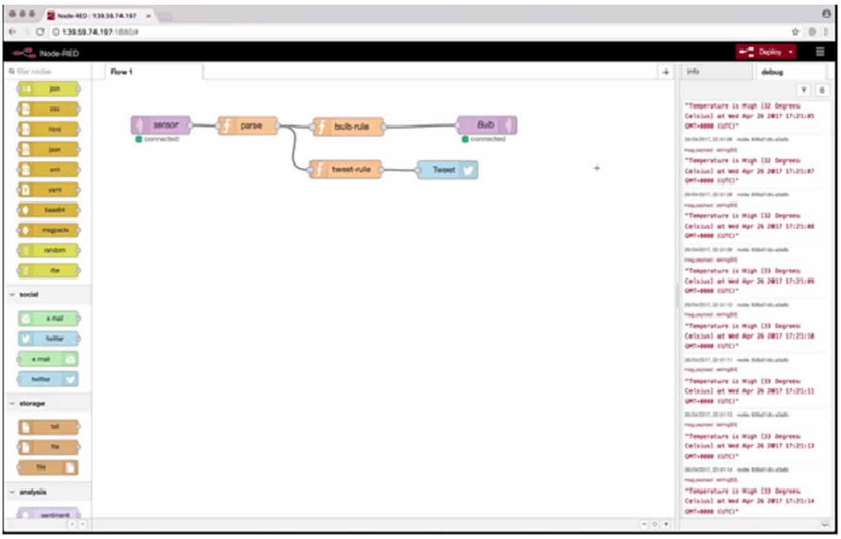
```
root@thing-server:~/iot# docker logs telegraf
2017/04/26 17:06:53 I! Using config file: /etc/telegraf/telegraf.conf
2017-04-26T17:06:53Z I! Starting Telegraf (version 1.2.1)
2017-04-26T17:06:53Z I! Loaded outputs: influxdb
2017-04-26T17:06:53Z I! Loaded inputs: inputs.mqtt_consumer
2017-04-26T17:06:53Z I! Tags enabled: host=5f66ee10ee52
2017-04-26T17:06:53Z I! Agent Config: Interval:10s, Quiet:false, Hostname:"5f66ee10ee52",
terval:10s
2017-04-26T17:06:53Z I! MQTT Client Connected
root@thing-server:~/iot#
```

Figure 16. visualization of temperature at the dashboard tool



```
SELECT value FROM environment.autogen.temperature WHERE tmpl.time() AND building = '1'
```

Figure 17. Rule setting using Node-Red rules engine



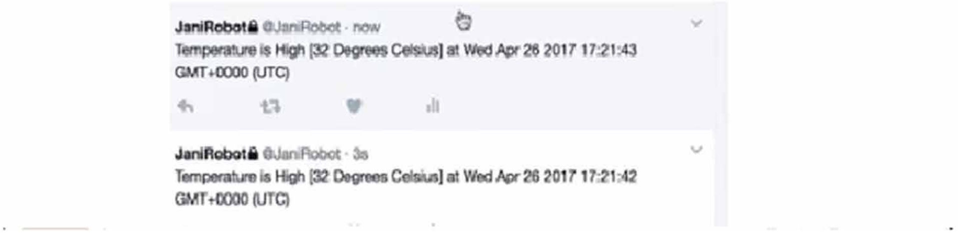
```
"Temperature is High (32 Degrees Celsius) at Wed Apr 26 2017 17:21:05 GMT+0000 (UTC)"
"Temperature is High (32 Degrees Celsius) at Wed Apr 26 2017 17:21:07 GMT+0000 (UTC)"
"Temperature is High (32 Degrees Celsius) at Wed Apr 26 2017 17:21:08 GMT+0000 (UTC)"
"Temperature is High (33 Degrees Celsius) at Wed Apr 26 2017 17:21:06 GMT+0000 (UTC)"
"Temperature is High (33 Degrees Celsius) at Wed Apr 26 2017 17:21:10 GMT+0000 (UTC)"
"Temperature is High (33 Degrees Celsius) at Wed Apr 25 2017 17:21:11 GMT+0000 (UTC)"
"Temperature is High (33 Degrees Celsius) at Wed Apr 26 2017 17:21:13 GMT+0000 (UTC)"
"Temperature is High (33 Degrees Celsius) at Wed Apr 26 2017 17:21:14 GMT+0000 (UTC)"
```

After successful configuration, the first rule can be created which is to turn on and turn off the bulb or change the colour of the LED. The bulb rule is to be called and write a tiny code snippet to change the colour of the bulb. The live stream is tweet, certain threshold can be defined and tweeting begins Create a new rule as tweet rule and the idea is to actually send the payload to Twitter. JavaScript code snippet is to be written if the temperature is more than 45. Fortunately node-red gives us a Twitter output right so that the tweet can be sent directly from node-red. Connect this and configure Twitter account. The live streaming is verified of tweets when temperature reaches the threshold level as shown in Figure 18.

SUMMARY

This is an end-to-end demo. The complete workflow is here. This can be used to either control the device that is basically turning on and turning off or even talking to a third-party application like Twitter. So this is a very powerful way of building IOT applications. Dashboard is very useful and the data is not hovering between multiple values and it gives us a high-quality visualization of the temperature data. At the summary of this chapter, it is basically two devices are connected to the MQTT broker in the cloud configured as Mosca. The output is captured or rather subscribed by Node-Red and Telegraf. The data is passing to influxDB from where the data is visualized to a Cronograf dashboard. Node-Red is used as the rules engine to configure powerful rules to control the devices or to talk to third-party applications.

Figure 18. Output at Twitter account based on the rules



JaniRobot @JaniRobot · now
Temperature is High [32 Degrees Celsius] at Wed Apr 26 2017 17:21:43 GMT+0000 (UTC)

JaniRobot @JaniRobot · 3s
Temperature is High [32 Degrees Celsius] at Wed Apr 26 2017 17:21:42 GMT+0000 (UTC)

REFERENCES

Alessi, M., Giangreco, E., Pinnella, M., Pino, S., Storelli, D., Mainetti, L., ... Patrono, L. (2016). A web based virtual environment as a connection platform between people and IoT. In *Computer and energy science (SpliTech), international multidisciplinary conference on* (pp. 1–6). IEEE. doi:10.1109/SpliTech.2016.7555925

Belli, L., Cirani, S., Davoli, L., Gorrieri, A., Mancin, M., Picone, M., & Ferrari, G. (2015). Design and deployment of an IoT application-oriented testbed. *Computer*, *48*(9), 32–40. doi:10.1109/MC.2015.253

Bischoff, U., & Kortuem, G. (2007). Life cycle support for sensor network applications. In *Proceedings of the 2nd international workshop on Middleware for sensor networks* (pp. 1-6). ACM. 10.1145/1376860.1376861

Botta, A., De Donato, W., Persico, V., & Pescapé, A. (2016). Integration of cloud computing and internet of things: A survey. *Future Generation Computer Systems*, *56*, 684–700. doi:10.1016/j.future.2015.09.021

Costa, P., Mottola, L., Murphy, A. L., & Picco, G. P. (2007). Programming wireless sensor networks with the teeny lime middleware. In *ACM/IFIP/USENIX International Conference on Distributed Systems Platforms and Open Distributed Processing* (pp. 429-449). Springer.

France, R., & Rumpe, B. (2007). Model-driven development of complex software: A research roadmap. In *2007 Future of Software Engineering* (pp. 37–54). IEEE Computer Society. doi:10.1109/FOSE.2007.14

Grgić, K., Špeh, I., & Heđi, I. (2016). A web-based IoT solution for monitoring data using MQTT protocol. In *Smart Systems and Technologies (SST), International Conference on* (pp. 249-253). IEEE. 10.1109/SST.2016.7765668

Guinard, D., & Trifa, V. (2016). *Building the web of things: with examples in node.js and raspberry pi*. Manning Publications Co.

Gyrard, A., Patel, P., Sheth, A. P., & Serrano, M. (2016). Building the web of knowledge with smart iot applications. *IEEE Intelligent Systems*, *31*(5), 83–88. doi:10.1109/MIS.2016.81

Kao, K.-C., Chieng, W.-H., & Jeng, S.-L. (2018). Design and development of an IoT-based web application for an intelligent remote SCADA system. In *IOP Conference Series: Materials Science and Engineering* (*vol. 323*, no. 1, pp. 12-25). IOP Publishing. 10.1088/1757-899X/323/1/012025

Komaki, D., Yamaguchi, S., Shinohara, M., Horio, K., Murakami, M., & Matsui, K. (2017). Design and Implementation of a Multimedia Control and Processing Framework for IoT Application Development. *International Journal of Informatics Society*, *9*(2), 73–84.

Kulkarni, V., & Reddy, S. (2003). Separation of concerns in model-driven development. *IEEE Software*, *20*(5), 64–69. doi:10.1109/MS.2003.1231154

Mellor, S. J., Clark, T., & Futagami, T. (2003). Model-driven development: Guest editors' introduction. *IEEE Software*, *20*(5), 14–18. doi:10.1109/MS.2003.1231145

Patel, P., & Cassou, D. (2015). Enabling high-level application development for the Internet of Things. *Journal of Systems and Software*, *103*, 62–84. doi:10.1016/j.jss.2015.01.027

Pathak, A., Mottola, L., Bakshi, A., Prasanna, V. K., & Picco, G. P. (2007). A compilation framework for macroprogramming networked sensors. In *International Conference on Distributed Computing in Sensor Systems* (pp. 189-204). Springer. 10.1007/978-3-540-73090-3_13

Román, M., Hess, C., Cerqueira, R., Ranganathan, A., Campbell, R. H., & Nahrstedt, K. (2002). Gaia: A middleware platform for active spaces. *Mobile Computing and Communications Review*, *6*(4), 65–67. doi:10.1145/643550.643558

Soldatos, J., Kefalakis, N., Hauswirth, M., Serrano, M., Calbimonte, J.-P., Riahi, M., & .... (2015). Openiot: Open source internet-of-things in the cloud. In *Interoperability and open-source solutions for the internet of things* (pp. 13–25). Cham: Springer.

Taivalsaari, A., & Mikkonen, T. (2017). A roadmap to the programmable world: Software challenges in the IoT era. *IEEE Software*, *1*(1), 72–80. doi:10.1109/MS.2017.26

Vujović, V., & Maksimović, M. (2015). Raspberry Pi as a Sensor Web node for home automation. *Computers & Electrical Engineering*, *44*, 153–171. doi:10.1016/j.compeleceng.2015.01.019

Whitehouse, K., Sharp, C., Brewer, E., & Culler, D. (2004). Hood: a neighborhood abstraction for sensor networks. In *Proceedings of the 2nd international conference on Mobile systems, applications, and services* (pp. 99-110). ACM. 10.1145/990064.990079

Yao, L., Sheng, Q. Z., & Dustdar, S. (2015). Web-based management of the internet of things. *IEEE Internet Computing*, *19*(4), 60–67. doi:10.1109/MIC.2015.77

Chapter 5
Internet of Things Testing Framework, Automation, Challenges, Solutions and Practices:
A Connected Approach for IoT Applications

Karthick G. S.
Bharathiar University, India

Pankajavalli P. B.
Bharathiar University, India

ABSTRACT

The internet of things (IoT) is aimed at modifying the life of people by adopting the possible computing techniques to the physical world, and thus transforming the computing environment from centralized form to decentralized form. Most of the smart devices receive the data from other smart devices over the network and perform actions based on their implemented programs. Thus, testing becomes an intensive process in the IoT that will require some normalization too. The composite architecture of IoT systems and their distinctive characteristics require different variants of testing to be done on the components of IoT systems. This chapter will discuss the necessity for IoT testing in terms of various criteria of identifying and fixing the problems in the IoT systems. In addition, this chapter examines the core components to be focused on IoT testing and testing scope based on IoT device classification. It also elaborates the various types of testing applied on healthcare IoT applications, and finally, this chapter summarizes the various challenges faced during IoT testing.

DOI: 10.4018/978-1-5225-7790-4.ch005

INTRODUCTION: OVERVIEW OF INTERNET OF THINGS TESTING

The IoT is an outcome of technological revolution which interrelates the unified computing devices, mechanical instruments, hi-tech electronic machines and humans that are equipped with capacity to exchange data over a network. The IoT was first formulated with the back support of Radio Frequency Identification (RFID) that can be applied to track the location of objects (Luigi Atzori et al, 2010). For example, products in the shopping malls are interconnected to their own network, which enables tracking the location of products and increases the billing process flexible at the point of sales depots. Every individual product is exclusively identified and categorized based on its RFID. This uses machine-to-machine networks and these resembles the IoT through network connected systems and data/information. The likelihood of connecting objects to the network allows tagging, tracing and reading of data from objects with greater technical efforts, technology of this era established called as IoT.

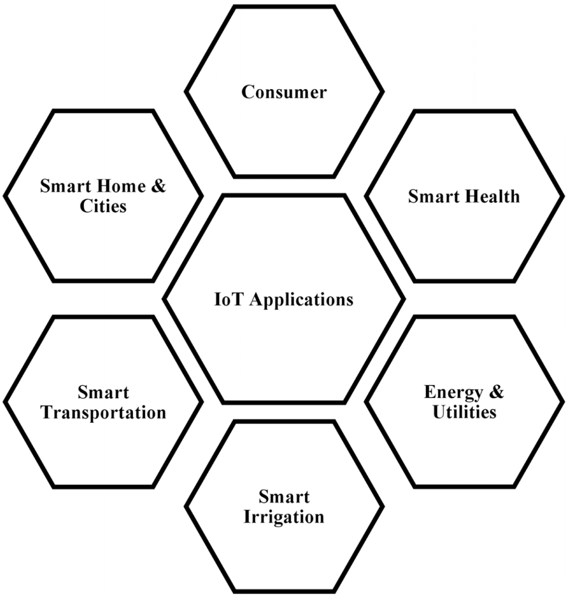
The essentials that emerged the IoT in current and future applications have been elaborated comprehensively and have been characterized by many authors. (Gubbi et al., 2013) and (Li et al., 2015) has discussed about the major components and architectural elements in IoT. The millions of sensing elements, actuators and other devices are exist at the lowest level of the IoT. Each of which requires a unique identification and addressing schemes because of their deployment are at large scale and also have high degree of constraints such as energy and computational resources. Communication is another important element which interconnects 'n' number of heterogeneous devices for providing smart services. Some of the short and long range technologies used for communications in IoT applications which may include Wireless Sensor Networks (WSNs), Radio Frequency Identification (RFID), IETF Low power Wireless Personal Area Networks (6LoWPAN) (G. Montenegro et al., 2007) and protocols like IEEE 802.11 (Wi-Fi), IEEE 802.15 (Bluetooth). As IoT devices generates a vast amount of raw data, thus increases the need of data storage and analytics. The data analytics, processing and machine learning in most of the IoT applications are deployed via cloud services. The IoT services are classified as Identity related, information aggregation, collaboration aware and ubiquitous services (Al-Fuqaha et al., 2015). Identity related services provide the unique identification for every deployed thing. Information aggregation services are responsible for collecting and storing the data received from sensors. Collaborative aware services make use of the data provided by information aggregation services to take decisions and to provide smartness to the system. Ubiquitous services enables the users to access the services without geographical restrictions (Al-Fuqaha et al., 2015). (Li et al., 2015) categorized the generic service-oriented architecture as sensing layer, network layer, service layer and interface layer.

The IoT is the interconnection of uniquely identifiable embedded physical objects with the prevailing network infrastructure (Gubbi et al., 2013). The occurrence of the IoT rapidly increasing the amount of connected devices, which simultaneously ignited the companies to provide IoT testing services. While considering IoT systems ranges from home appliances, security systems and other devices are communicated to the mobile applications. Those IoT enabled devices may unsuccessful in the connected environment and at the same time users also become more unaccustomed to the connected IoT devices. Usually the users will expect the technologies to work perfectly from the beginning. Thus the importance of focusing on developing the quality IoT products will be the game changer.

The various vertical areas which are impacted by the evolving IoT technology can be classified based on type of network availability, coverage, scalability, heterogeneity, user participation and repeatability (Sebastien Ziegler et al., 2013) (Ericsson, 2011). The following Figure 1 depicts the categorization of IoT applications.

The development of IoT applications and data transferring introduces the ease of accessibility and convenience for enterprises and end users. For example, the merging of Smart Home and Utility-Oriented IoT Applications generates electricity consuming data at the smart home environment and transfer the data to the utility

Figure 1. Categorization of IoT applications



(electricity) provider which enables them to optimize the power demand and supply at Utility IoT environment. The sharing of data between service providers and the user is being enabled by internet technologies. This kind of applications emerging business opportunities and such business applications must be highly quality assured.

LITERATURE REVIEW

IoT system testing have more technical hitches which are not exists in traditional systems, which includes web services, due to assorted and highly distributed nature of its constituents (Reetz et al., 2013). In order to assure the accurate functioning of such multifaceted systems, various evaluations must be performed before the IoT system deployment in a real-time environment. It requires collaboration with the physical world which are to be observed, using common software testing methods (Reetz et al., 2013). The assorted nature of IoT components stresses upon strong testing capabilities to ensure the user needs as well as service level agreements. Performance testing has done experimentally using the middleware for evaluating end-to-end delivery and bandwidth consumption of MQTT and CoAP protocols (Thangavel et al., 2014). This performance testing is focused on the transportation of sensor data from gateway to storage or back-end server. From the experimental results, it has been identified that different network conditions may directly create an impact on the performance of various protocols. Further the selection of protocol can be done based on the current network conditions for improving the performance of network.

From the literature survey (Kiruthika, J. and Khaddaj, S., 2015) (Marinissen et al., 2016) (Xu et al., 2014) (Bertino et al., 2016) (Sicari et al., 2015) (Lin et al., 2016) (Sajid and Abbas, 2016) (Worthy, 2016) various discussions have been identified about the IoT issues but there is no methodological analysis regarding the influence of software testing methods and techniques. In Table 1, the relationship between the IoT issues and testing has been drawn.

NECESSITY OF IoT TESTING

According to Gartner, more than twenty five billions of IoT systems are existing in today's world, which requires a better automated testing protocol (Mark Hung, 2017). IoT can be considered as a combination of software, hardware and architectures that facilitates the uniquely identifiable physical objects to sense and act together with the environment via internet. By nature, IoT devices are subjected to error-prone. When IoT systems are scaled up with respect to complexity, features and number of

Table 1. Relationship between the IoT issues and testing process

References	Issues	Inference
(Kiruthika and Khaddaj 2015) (Sicari et al., 2015)	There is no flexible infrastructure is required to deal with security threats in a dynamic environment.	An appropriate testing solution is needed to guarantee the access control, confidentiality and user privacy among IoT devices.
(Marinissen et al., 2016) (Xu et al., 2014) (Bertino et al., 2016) (Lin et al., 2016) (Sajid and Abbas, 2016)	Increased risk of semantic attacks and physical access to various IoT devices like sensors and actuators.	Thus increases the demand for security testing with respect to privacy aspects.
(Xu et al., 2014) (Sicari et al., 2015) (Pering, 2018)	The process of selecting the devices for IoT system will be quite easier task but the integration into the IoT architecture could consume more time. At that time, few selected devices may undergo firmware and hardware specification changes.	Thus increases the requirement of effective integration testing and testing automation.
(Saksoft, 2018)	The amount of interoperability among IoT devices is still lack due to: ➢ Mismatch during integration of IoT devices because of diversified manufacturers. ➢ Operating System dependency ➢ Firmware compliance issue ➢ Protocols standard issues	Testing of interoperability issues between IoT devices under different environmental and technical conditions for providing compatible IoT systems.
(TestingWhiz, 2018)	The demand for IoT systems are increasing rapidly which requires efficient and full-bodied IoT applications quickly.	Testing automation helps the IoT system development process using agile methodologies.

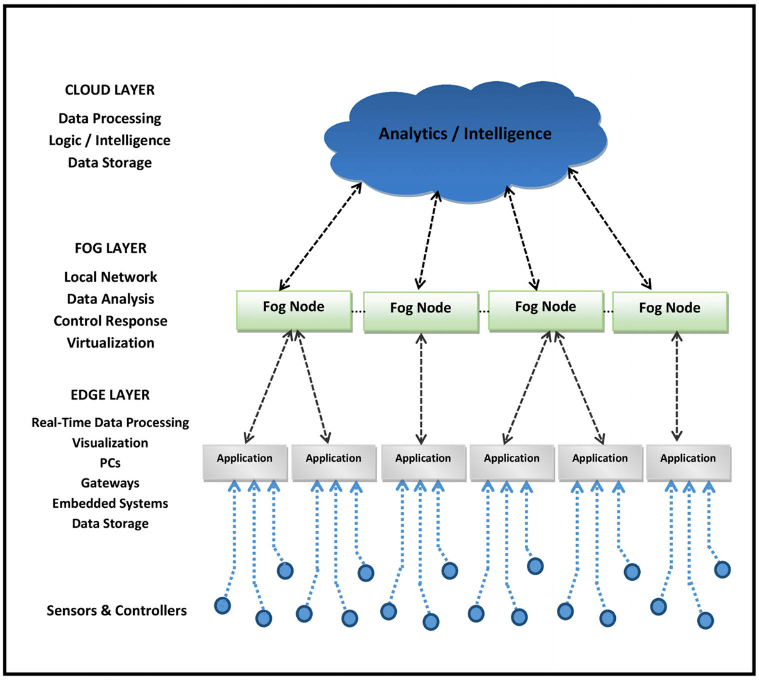
devices simultaneously, the amount of errors will be increased with its scale. Apart from the scaling issues in IoT systems, the following factors which make the IoT systems unusually complex to test and validate:

- Dynamic nature of Topologies
- Unreliable Connectivity
- Heterogeneity in Devices and Protocols.

Hence, defining the test cases for IoT system is considered as a tedious task. The real life testing scenarios does not satisfy the IoT testing needs and there are additional test scenarios like performance, scalability, reliability, and security factors to be considered for testing the IoT systems. Focusing on testing the IoT systems at various layers and elements becomes a phase of a system development. The IoT system testing can range from low-level elements to high-level elements (Cigniti, 2018). Generally, IoT system development can be categorized into three layers: Edge, Fog and Cloud as shown in Figure 2. Each layer of IoT system has its own functionalities, thus requires different testing needs.

- **Edge Layer:** Edge layer is the bottom most layer which comprises of loosely coupled systems and human interactive resources such as sensors, microcontrollers, micro data storage, gateways and computing devices. These resources have computational capacities ranging from highly capable devices like nano data centers to less capable devices like smart mobiles. It enables performing of possible computations at the edge of the network, in-order to reduce the data bottleneck issues and network traffic issues. The data control and security is more important at edge layer as it deals with bidirectional data processing. Most of the embedded systems organizes the real time data at an edge and performs low-level processing before transferring the data for high-level computing environments. The edge layer must maintain an end-to-end and reliable connectivity with the Fog layer.

Figure 2. IoT system development layered architecture



- **Fog Layer:** The fog layer is an upper layer to the edge layer, which associates all the high-end computing networking devices such as switches and routers. This association enables the IoT systems to run cloud applications on its native architecture. The fog layer is responsible for handling the critical tasks assigned by edge resources. Additionally it does the jobs such as data analysis, data reduction and control responses.
- **Cloud Layer:** This layer is the top most layers which provide the data processing capabilities along with it serves as repository for storing all data in the cloud. A centralized storage of data can be accessed by both edge and fog layers with high degree of reliability. Based on the functionalities of each layer requires various testing strategies like performance testing, security testing, usability testing and connectivity testing must be adopted to ensure quality. These testing strategies must work closely with IoT system development to overcome the key hurdles such as:
- Lack of interoperability is a major issue which affects the application of emerging networking standards (D.Bandyopadhyay et al., 2011) (J.Song et al., 2017) and protocols for connecting and collaborating smart objects.
- The proper implementation of IoT testing identifies the errors and reduces the failure of IoT systems after deploying at user end.
- IoT systems generates vast amount of data which must be captured, routed and analyzed effectively, which is a time critical task to do with existing traditional architectures.
- Most of the IoT systems likely to have very less human interference thus increase the risk of security breaches which may cause intensive failures of systems. It requires continuous testing of IoT systems to reduce security breaches and protect the systems from intensive failures.

The future world is interlinked with the application of IoT and the quality factors play a vital role in assisting the IoT systems to be succeeded in markets. Most of the IoT testing addresses the data management, security, performance and privacy issues, which promotes the development of trusted IoT products.

BASIC LAYERED ARCHITECTURE OF IOT TESTING

An all-inclusive Quality Assurance (QA) is necessary to fulfill the complexity and extensiveness of IoT testing. QA strategy may include various types of testing, tools and simulators to be deployed. Generally, architecture of IoT testing can be classified into two layers: 1. Device Interface Layer and 2. User Interface Layer (AFour Technologies, 2017).

Device Interface Layer

Device Interface Layer is composed of software's and hardware's through which a real-time IoT devices can interact with each other's. For example, when a Bluetooth device transmitting a real-time data to a mobile device, requires many interaction testing to satisfy the functional portion of QA and also other types of testing to be done to satisfy the non-functional portion of QA. The wide classifications of typical software testing requirements are standards, interoperability and security.

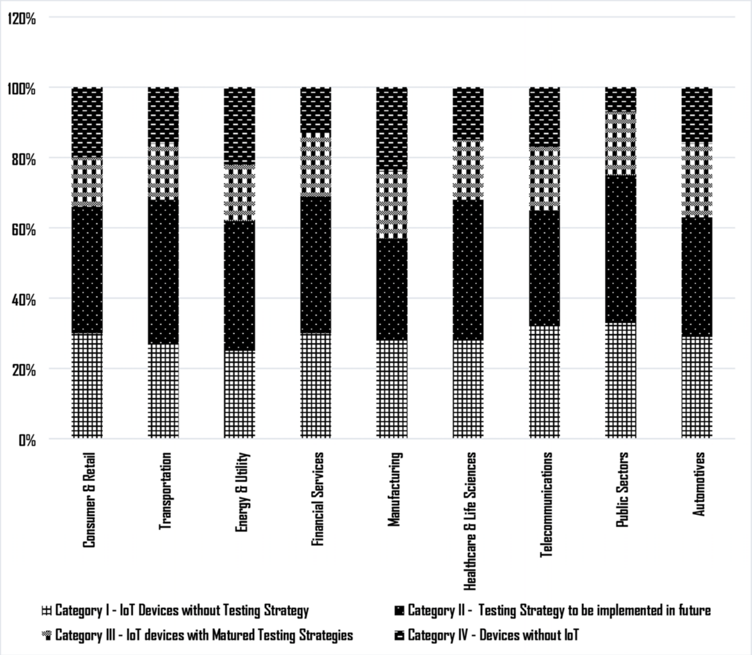
User Interface Layer

The user interface layer acts as a communication point between the physical things and the users. This layer is responsible for improving the quality of IoT systems by performing the following testing:

- **Device Level Testing:** This testing approach is indeed to validate the device related elements which includes system connectivity, serial protocol, properties and abilities of the device, scheduling, power supply modes and Over-the-test (OTA).
- **Cloud Level or Network Capability Testing:** This testing is equipped with integration tools which are capable of automating the functional testing, integration testing and Application Programming Interface (API) testing. The important features of Cloud Level or Network Capability Testing includes:
    - Performance, Reliability and Scalability Testing
    - Security Testing
    - Data Management and Data Privacy Testing.
- **Mobile Level Control and end-to-end Testing:** This testing procedure focuses at mobile level components which includes API, back-end and their interactions. The end-to-end testing of IoT devices plays a vital role of QA which comprises of automated testing of cloud, physical devices and applications.

The World Quality Report (WQR) stated that 65% of total IoT devices have not followed any testing strategies to ensure quality (Capgemini Sogeti, 2016). The classification of IoT device development organizations with respect to applications and testing strategy adoption are depicted in the following Figure 3.

Figure 3. Classification of IoT system development organizations



IoT TESTING FOCUS AREA AND SCOPE

IoT testing procedures were reported by various industries (Cognizant, 2016) (RCR Wireless, 2016), but academic-oriented reports about IoT system testing has been found to be low. Most of the industries and academic reports states that they have focused on performance evaluation (Lunardi et al., 2015) (Vandikas and Tsiatsis, 2014), IoT emulation (Sanchez et al., 2014) and Testbed deployments (Adjih et al., 2016). A logic-based technique has been used to perform conformance and performance of XMPP protocol and this method have evaluated against prototype experiments (X. Che and S. Maag, 2013). The discharging ratio of IoT systems is growing simultaneously to the consumer expectations in terms of quality IoT products. Due to the increasing expectations of customer on IoT systems, it is necessary to develop a non-intrusive, end-to-end functional system by identifying the major areas to be focused on testing. Every IoT systems have their own set of functionality issues and challenges.

For example a specific IoT system may not able to communicate with the mobile application; a sensor may fail to send the data; installation of application on IoT device may fail and frequent crashing of IoT systems. Most of these issues may arise due to i) introducing the IoT products into the market without proper testing, ii) compromising IoT testing platforms, and iii) lack of functionality and usability behavior. Thus permits the IoT systems unlikely to work as per the expectations. To ensure the working of any IoT systems as expected, the development organizations must focus on end-to-end quality testing. There are assorted concerns to be tested which are mission critical in nature as it requires high coding efforts. The testing areas to be focused are initiated from sensors, devices with back-end data analysis, and real-time IoT environment validations to overcome the application complexities and performance issues (Rajesh Shanmugasundaram, 2015). The following are the identified major IoT Testing areas:

Connectivity

The connectivity is a key factor which establishes the communication link between the IoT system components. The success ratio of an IoT system highly depends on the enabled reliable communication links between the devices and the hubs. The unreliable communication may cause interrupt the data transmission from one end to another end which affects the accuracy of data and the system becomes impracticable (W3C-Group, 2016). Every IoT system must be connected around the clock even under in power saving mode. At the time of power saving mode, the IoT devices frequently floods the ping messages on the path to ensure the connection is not lost. To ensure seamless connectivity among the IoT system components, connectivity test cases should be evolved.

Sample Connectivity Test Cases

- Validate that all the devices grouped in the IoT are able to register to the network without any interruptions.
- Validate the application that ensures the consistency and durability of data storage so that whenever the connection is restored, data is back to shape as it was before.

Security

The implementation of adequate security validations is highly important and correlated with other testing strategies. This ensures the IoT devices are appropriately authenticated before the establishment of communication (Desnitsky and Kotenko, 2016). For example, in terms of connection established via Bluetooth technology, only the synchronized and paired devices can able to communicate with each other. To ensure the security among the components of IoT systems: i) Communication must be established only after the successful authentication, ii) The data transmission should take place in an encrypted form and iii) If a device fails the maximum number of connection establishment attempts, the IoT system should prohibit the particular communicating device to connect again for a pre-defined time period.

If a maximum number of connections have been attempted, the device should not try to connect again for a pre-defined time period. There are other possibilities for cyber-attacks and breaches for security breaks. IoT environment has huge amount of entry point for security attacks.

Sample Security Test Cases

- Validating and ensuring no unauthorized access to the IoT system or data.
- Validate the remote wiping of data on compromised IoT systems.

Performance

Another crucial aspect of testing is analyzing the performance of the IoT system. Every authenticated IoT system can able to transmit any amount of data to other system as per the requirements (Esquiagola et al., 2017). In case an IoT system wants to transmit the data more than a predefined amount, such transmission must be initiated only after a pre-set delay time or after receiving an acknowledgment from the receiver. The IoT system can able to transfer the data even under low power status. The low power status information of every device should be flooded over the network. Therefore, the performance of an IoT system is analyzed in three aspects of communication such as Device-to-Device, Device-to-Server and Server-to-Server communication. These communications are evaluated based on the factors like bandwidth efficiency, latency and packet loss.

Sample Performance Testing Cases

- Validate the system response time against benchmarked time with specified connectivity conditions.

Functionality

Functionality of an IoT system is not about adding more functions and making them to work. Every IoT system is fully operational only when all the functions of the system are properly tested under multiple environments, platforms and simulations (Hillah et al, 2017). The functional testing of an IoT system can be beaten by an exclusive performance testing for ensuring quality and uninterrupted user experience. This testing verifies the various scenarios of IoT systems which includes web or mobile applications, functional requirements, access control, data storage and identity management.

Sample Functionality Testing Cases

- Applying a set of appropriate inputs to the module functions for verifying the returning outputs.

Compatibility

In the scenario of IoT systems, it is expected that many devices can establish a communication with IoT system. Each of them has differed in terms of device version, protocol version, operating system, software and hardware configurations (Gyory et al., 2017). At such scenario, examining the various compatibilities in an IoT system acquired more importance. The compatibility is assumed to be a challenging task for QA team in future because the scope of coverage may be expanded in terms of amount of devices. Therefore, validating the possible working combination of differed hardware, protocol versions, software versions and operating systems is done via compatibility testing.

Sample Compatibility Testing Cases

- Verify the compatibility ratio of an IoT system by allowing IoT software to work with the set of devices.
- Determine the compatibility issues remains within the different network layers, by tracking the data movement at various networks and devices.

Usability

As discussed above, there is more number of devices which are of different configurations and factors used in IoT systems. Therefore, the perception of users while working on IoT system may differ from user to user. This requires the checking of IoT system real-time usability and the verification of human-machine interactions are crucially important. Usability verification determines the IoT system in five different arenas which includes usefulness, findable, accessible, usable and desirable.

Sample Usability Testing Cases

- Verifying whether the IoT system is beneficial and adds value to the target audience.

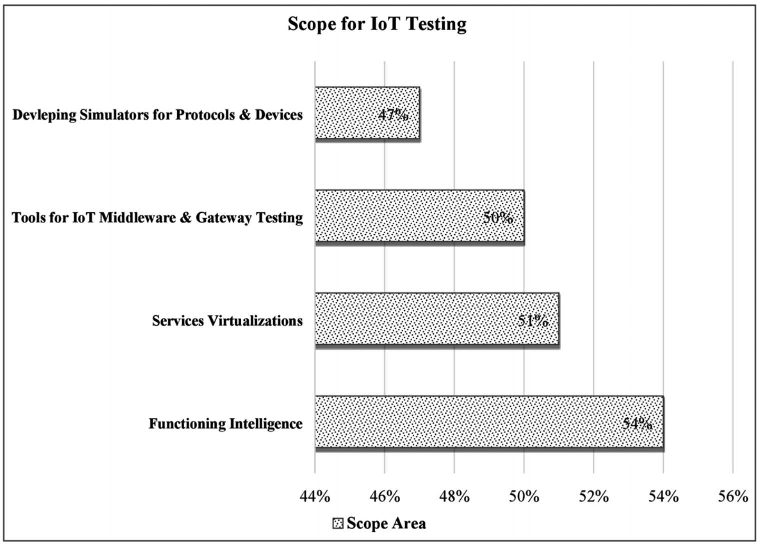
The key components of IoT platform can be classified into four types which include sensors, applications, communication and data storage. The finding of IoT system testing scope and coverage is a vital element for designing the comprehensive testing strategies. The identified key testing scopes for each of the components of IoT platform is depicted in the Table 2.

According to the WQR report (Capgemini Sogeti, 2016), the scope for IoT testing can broadly classified and analyzed based on: (i) functioning intelligence (ii) services virtualizations (iii) tools for IoT middleware and gateway testing and (iv) developing simulators for protocols and devices as shown in Figure 4.

Table 2. Testing scopes for IoT system components

IoT System Components	Testing Scopes
Sensors	Device hardware
	Embedded software
	Sensor Response Time and Performance
Application	Application functionality
	Error handling mechanism
	User friendliness
	User roles and Access Levels
Communication	Multiple request handling
	Network connectivity
	Interaction among devices
	Frequency of Data Transmission
	Data packet loss
	Data Security – Data Encryption and Decryption
Data Storage	Data Consistency and Integrity Validation
	Verification of Data Values

Figure 4. Analysis on scopes of IoT testing



TESTING FRAMEWORK FOR IoT APPLICATIONS

IoT have thriving its path towards the development of testing framework which provides the endowment for carrying out the functional validation of IoT system. The development of an IoT system is expected to overcome heterogeneous mixture of challenges and so designing a strong testing framework is in place for validating the functional and non-functional requirements of IoT systems. IoT development process follows the traditional systematic QA practices to validate the IoT applications.

Consider an IoT-based Medical Healthcare Tracking System as shown in Figure 5 which continuously monitors the physical vital signs of the patients and transfer the reports to the concerned physicians. The reports generated can be reviewed in future whenever required. In order to test certain IoT-based Medical Healthcare Tracking System, it is mandatory to design an IoT testing framework (Cigniti, Aug, 2018) as shown in the Figure 6 by emphasizing the entire core IoT elements (devices, communications, processing).

Connectivity Testing

As the network connectivity can be considered as a backbone for any IoT applications (W3C-Group, 2016). Especially IoT-based healthcare solution, the connectivity plays a predominant role, because such systems must be available at any time and it must possess an uninterrupted connectivity with the users. The connectivity testing must

Figure 5. IoT-based medical healthcare tracking system

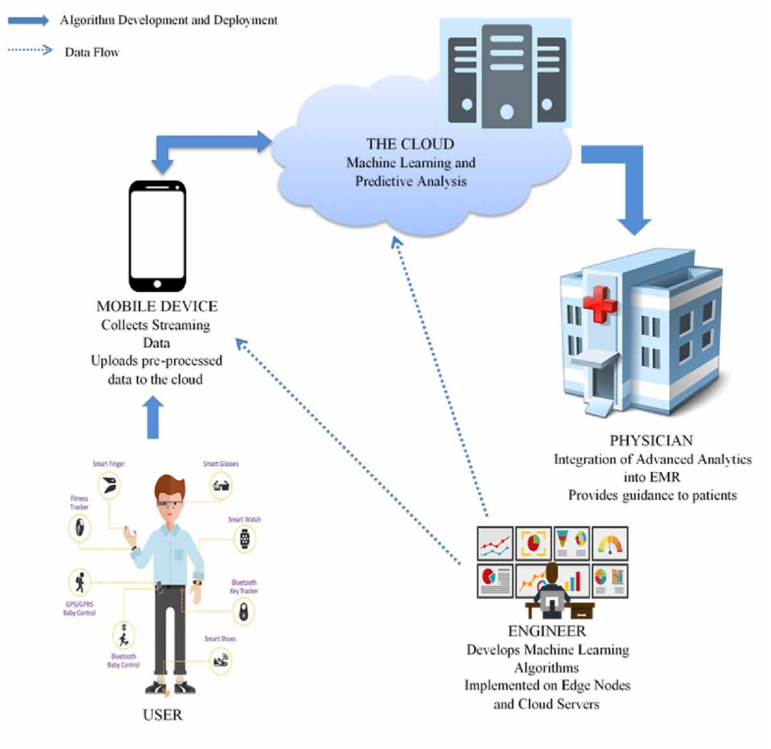
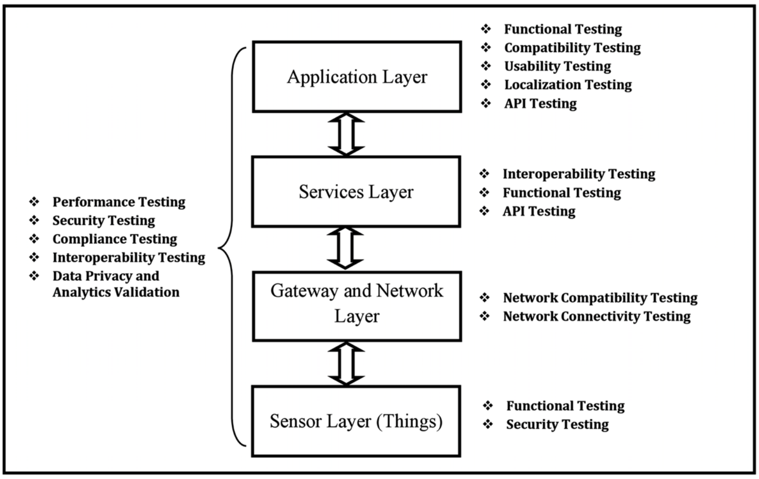


Figure 6. IoT testing framework

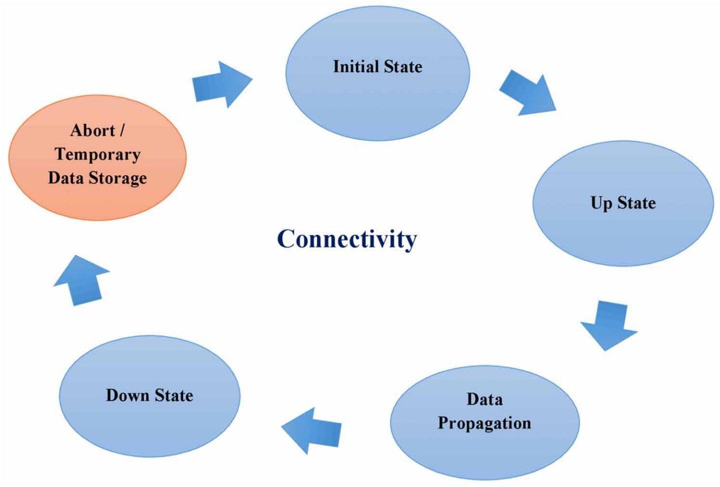


ensure the availability of seamless connection while transmitting and receiving the data from participating sensor nodes when the connection is in UP state. Then the connection DOWN state condition must be considered. Scaling the robustness of IoT systems is not suitable for all situations; there is a possibility for the system to go offline. Being a good IoT tester, testing the system in offline mode is binding. When an IoT-based medical system goes to DOWN state, an alert message to be issued which enables the physicians to monitor the health status of their patients manually until system turn into UP state as shown in Figure 7. Optionally, a mechanism can be deployed in the system which can collect and stores the data generated during DOWN state. When the system turns into UP state, the stored data must be transmitted to ensure the data integrity.

Security Testing

Every IoT system is said to be a data centric in which all the functions are done based on the available data. There must be a data flow between the devices and at the same time, there may be a chance that unauthorized users can access or read the data on transmission. Some of the susceptible variables are identified as injectable points for new security threats are wireless sensor networks, online streaming data collection applications, middleware, M2M protocols and application programming interfaces (APIs). IoT testing framework includes a security testing as a core area for securing both the devices and cloud services or networks (Fernández-Caramés et al., 2016). From the tester perspective, it is necessary to verify the data with respect to

Figure 7. IoT connectivity testing states



the various security aspects like data protection, encryption/decryption and device authentication (Visoottiviseth et al, 2017). The security testing at the device and protocol level is primarily carried out for detecting the issues at the source node. The testers have to incorporate the mechanism of suspending the compromised IoT devices from the network as shown in Figure 8. Such scenario can be verified by attempting access to the data at any IoT device on the network with invalid credentials and during that situation the mechanism incorporated must remotely wipe-out the device from the network.

Performance Testing

When considering an IoT-based system for healthcare domain, it is a crucial factor to ensure whether the system is capable to tolerate the scalability to larger extent. For example, during testing process the data propagation among 10-20 devices will be analyzed in terms of performance as shown in Figure 9. But in real-time scenario, it may require to connect the whole hospital with 'n' number of devices. This increases the data propagation rate among those devices and the system is expected to perform evenly under added data propagation. In addition, performance testing includes the network model along with the internal computation proficiencies of the system. This has to be done at various levels like Application Level, System Level (Processing, Storage, and Analytics), Network and Gateway Level (MQTT (Brian Raymor, 2014), CoAP (Z. Shelby et al., 2014) (C. Bormann, 2014), and HTTP (R. Fielding, 1999)). IoT performance testing is slightly differed from the traditional performance testing as illustrated in the Table 3.

Figure 8. Mechanism of security testing using invalid credential injection

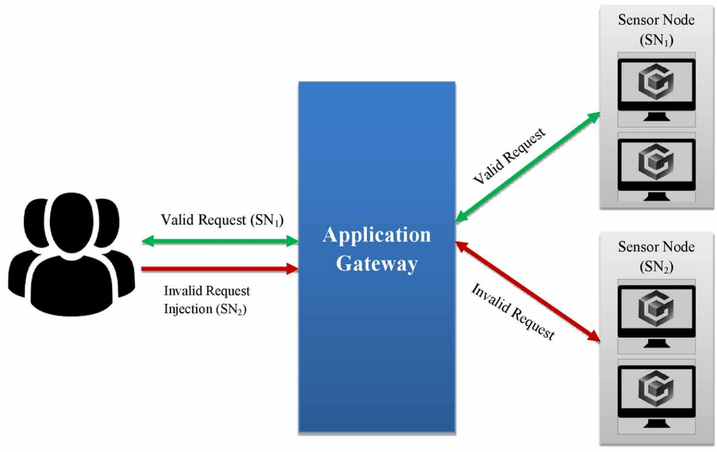


Figure 9. Performance testing scenario under load

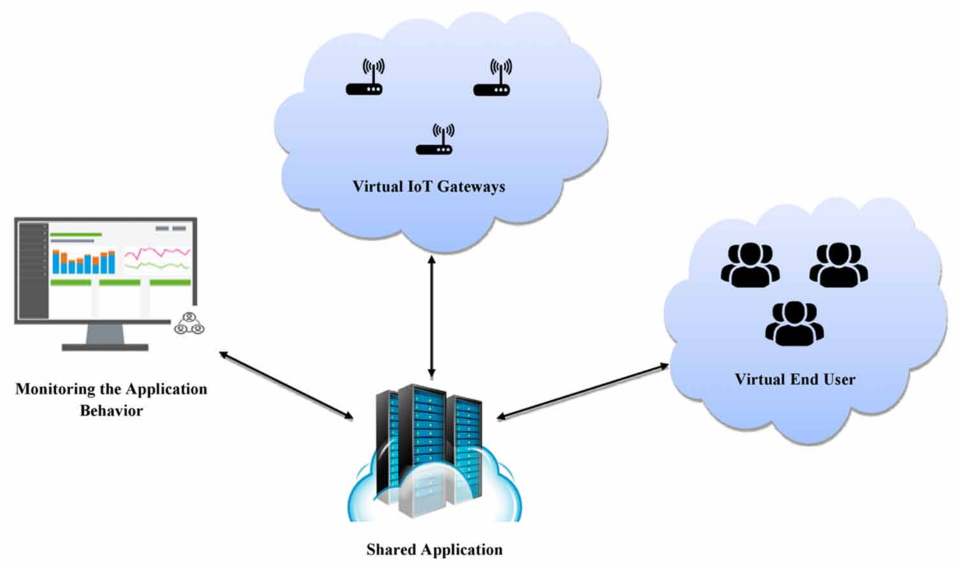


Table 3. Traditional performance testing vs. IoT performance testing

Key Factors	Traditional Performance Testing	IoT Performance Testing
Simulation	Simulation of System Users	Simulation of Sensors & Devices
Scalability	Hundred users to Thousand Users	Thousand Devices to Millions of Devices
Amount of data propagation	Large amount of data propagated for a single request	Minimal amount of data propagated for a single request. Additionally, data propagation uses time interval per request
Protocols	Make use of standard protocols for communication	Make use of non-standard and new protocols for communication
Requests & Responses	User create a request and receive a response	IoT devices may issue a request and receive response as well as receive a request and issue a response
Business Intelligence	Only a small number of application have business intelligence as a part of testing	Business Intelligence is a part of IoT system, it desires to measure the performance by applying huge load on devices

Challenges in IoT Performance Testing

The topics discussed below are the major challenging issues identified in IoT performance testing:

- **Protocols and Performance Testing Tool:** IoT does not have any standard protocols to establish the communication between IoT devices and applications. The widely used IoT protocols are HTTP (R. Fielding, 1999), IoTivity, MQTT, AMQP, CoAP (Brian Raymor, 2014) (Z. Shelby et al., 2014) (C. Bormann, 2014) and more. These protocols are in the early stage of development and evolving continuously with frequent updation done on IoT applications. Hence, the traditional testing tools may not support them efficiently.
- **Geographically Distributed Network Conditions:** IoT systems are composed of huge number of sensor nodes which are geographically distributed and maintain the seamless communication with the IoT servers for exchanging data. According to performance testing, there is a necessity to simulate the dispersed sensors nodes with suitable short and long range communication technologies.
- **Load Conditions:** It is obligatory to test the IoT application with various load patterns and real-world conditions. This may be dynamically complex and it will be difficult to gather the data.
- **Real-Time Decision Making:** Most of the IoT implementation requires the processing of streaming data and based on the processed data, the corresponding decision will be taken. The decisions may include notifications, alerts, and requests to different devices which does certain action. As a part of performance testing, the entire decision making factors are to be analyzed to measure the time taken to generate a notification or alert or request. On rectifying these discussed challenges of IoT performance testing, the key features and benefits are achieved as depicted in the Table 4.

Table 4. Key features and benefits of refined IoT perfromance testing

Features of IoT Performance Testing	Benefits of IoT Performance Testing
Support for various communication protocols	Resolves the interoperability issue
Supports different network simulations	Acquires market popularity easily
Supports cloud based load generation	It doesn’t require to perform a device simulation at cloud
Testing framework supports committed new protocols	Adopting new protocols is easy

Functionality and Compatibility Testing

The architecture of an IoT-based Medical Healthcare Tracking System is much complex than other IoT architectures. Since healthcare system architecture is a combination of various device versions, protocol versions, operating system versions and communication modes (Hillah et al, 2017). The compatibility testing is an important phase to be done at application and network layer of the IoT testing framework. It is performed to validate the possible working of combined heterogeneous elements. Compatibility testing is said to be non-functional testing that ensures the end-users satisfaction. Compatibility testing for an IoT-based Medical Healthcare Tracking System is performed as follows:

- **Hardware:** Tester executes the IoT application on different hardware configurations and identifies the incompatible hardware device. Then the applications can be modified based on the test report to provide hardware compatibility.
- **Network:** Tester propagates the sample set of data to all the sensor nodes or devices using different communication protocols.
- **Software Version:** Software used in IoT application may have multiple versions and it is necessary to test the IoT devices with all the available software versions. This can be carried out using two types of version inspections such as,
- **Backward Compatibility Testing:** IoT application is tested against the old version of software and it is called as downward compatible.
- **Forward Compatibility Testing:** IoT application is tested against the new version of software and it is called as forward compatible.

Usability Testing

Usability testing is a phenomenon which allows verifying the IoT application has met all the features and specifications as provided by the end-users (Wittstock et al, 2012). It scale the user experience on IoT application and hence this testing is also called as user experience testing. Usability testing ensures user comfort while operating the IoT system in the anticipated manner. As per IoT-based Medical Healthcare Tracking Systems concern, it must be portable enough to be moved in hospitals from one place to another and the system not only generate notifications but also triggers error messages and warnings. These notifications and error / warning messages must be displayed properly in the handheld end devices (mobile devices).

Usability testing primarily focus on:

- Ease of use
- Ease of familiarizing the IoT system
- Providing high satisfaction for the users with the experience.

This has many dimensionalities to do it but a structured usability testing is a major way which translates the user experience into a validation process (Wittstock et al, 2012). In general, there are some methods identified to perform structured usability testing.

1. Evaluate the prototype of an IoT system during design phase and verify whether it is feasible to carry out or not.
2. Develop the system and offer set of real-time users to work on the IoT system and prepare the experience reports.
3. Use of tools which are able to provide a statistical report based on the inputs given and outputs expected.
4. An external evaluator will be hired to validate the strengths and weakness of the system.

Structured Usability Testing Process

The phases of structured usability testing process is depicted in Figure 10 and discussed below:

Phase 1: Identification of Users

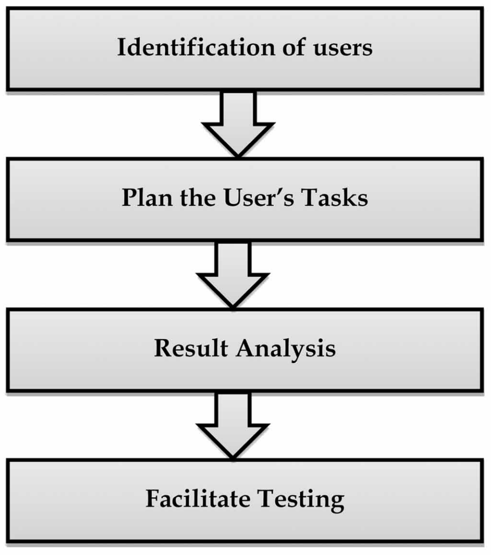
At this phase, a set of users will be selected based on emphasizing how the real-time users are going to be (Wittstock et al, 2012). The selection of users can be of two categories: experts or beginners. The experts can easily evaluate the entire process of an IoT system whereas beginners require a lot of training even to start the testing process.

Phase 2: Plan the User's Tasks

This phase acts as a platform for planning the tasks that the selected users are expected to perform on an IoT system. A set of circumstances are made prior to starting the test in which the IoT systems are expected to deploy.

Phase 3: Facilitating the Testing

Figure 10. Structured usability testing process



During this phase, the users perform the pre-planned tasks on the IoT systems and record the test progress and results. Additionally, facilitating the development teams to participate in usability testing which provides a better knowledge on how the system was used.

Phase 4: Result Analysis

At the end of this test, overall system performance for the specified tasks is presented to all the actual users and the potential problematic areas are explored.

Other than the IoT-based Medical Healthcare Tracking System, the following Table 5 illustrates the applicability of different testing types for several IoT components.

Pilot Testing

As far as the IoT is concerned, pilot testing can be defined as the process of validating the IoT system by exposing the system into real-time environment. Testing the IoT system in lab environment may function well but it may go wrong when exposing the system into real-time conditions. To avoid such backfire, the system is exposed to the small number of users in the real-time. Then the system is made robust with the help of feedback provided by real-time users.

Table 5. Applicability of testing on IoT components

Components Testing Types	Sensor	Application	Network	Data Storage
Connectivity Testing	No	No	Yes	No
Functional Testing	Yes	Yes	No	No
Usability Testing	Yes	Yes	No	No
Security Testing	Yes	Yes	Yes	Yes
Performance Testing	No	Yes	Yes	Yes
Compatibility Testing	Yes	Yes	No	No
Services Testing	No	Yes	Yes	Yes
Operational Testing	Yes	Yes	No	No

Some important IoT applications areas and its testing types are shown in Table 6 based upon the MarketsandMarkets report (MarketsandMarkets, 2017).

AUTOMATED TESTING SERVICE

In the fast growing era, the IoT systems are said to be smart enough which are made of interconnected sensors and devices, generating a large amount of diversified types of data with larger complexity. The series of connected devices must work together without any flaw from beginning to end. End-users are not fascinated to know the reason for system failure, service crash, sensor hanging and computational data missing. The end-users need the system to work but the development organizations have to focus on the quality of the product. In IoT, Testing as a Service (TaaS) ensures the system to remain top in the quality and in market (H. Kim et al., 2018). TaaS offers the entire automated support starting from the planning for the IoT-based systems to make them perform as expected (Osama Abu Oun, 2015). Every IoT systems require the meticulous automation testing services has five phases as shown in the Figure 11.

Plan

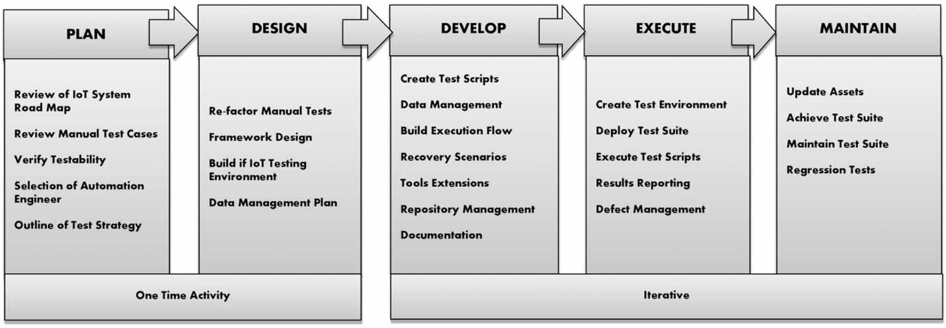
- **Reviewing of IoT System Road Map:** In the first phase of testing automation, QA team review the IoT system and analyze the system documentation to learn the functionalities of the system.

Table 6. IoT applications areas and its testing types

Application Type	Application Inclusion and Testing
Smart Grid	Smart grid provides efficient utilities for energy depletion, metering services, connecting and automating the unified energy supply. It must incorporate functional testing, performance testing, security testing, network testing compatibility testing and usability testing.
Smart Healthcare	Smart healthcare offers the remote monitoring of patients health via mobile applications, bio-sensors, smart devices and wearbales. Smart healthcare applications are time critical, so it must incorporate connectivity testing, functional testing, performance testing, security testing, network testing compatibility testing and usability testing.
Smart Home	Smart home focuses on interconnected homes, automating home operations, climate control, parking management and security. It has to incorporate functional testing, performance testing, security testing, network testing compatibility testing and usability testing.
Smart Industries	Smart indutries is responsible for automating industrial process, monitoring of production process and supply-chain management systems. It require the incorporation of pilot testing, operational testing, functional testing, performance testing, security testing, network testing compatibility testing and usability testing.
Smart Transportation	Smart transportation offers remote vehicle monitoring, vehicle controlling, logistics optimization, fleet management, monitoring drivers dizziness, vehicle tracking and traffic management. It require the incorporation of operational testing, functional testing, performance testing, security testing, network testing compatibility testing and usability testing.

- **Reviewing of Manual Test Cases:** When QA team identifies the system functionalities to be clear; then manual test cases of the system will be reviewed. The authors (Aho et al., 2013) used Model-Based Testing (MBT) technique which generates the test cases from behavioral models of the system. It is also an effective platform independent approach for mining Finite State Machine (FSM) model for Graphical User Interface (GUI) application testing.
- **Verify Testability:** QA team verifies the manual test cases and choose the most appropriate cases from the manual cases for automation.
- **Selection of Automation Engineer:** Once the system review process is completed, the suitable automation engineer as per the system requirement is chosen from QA for automating the tests.
- **Testing Strategy Outline:** After finalizing the automation team, the strategies for automation testing of the system will be defined.

Figure 11. Phases of testing automation



Framework Designing

- **Re-Organizing Test Cases:** In this stage, automation team verifies the manual test cases and screen the cases based on the automation priorities.
- **Building of IoT Testing Environment:** At this stage, a laboratory based setup is built for testing the hardware devices, networks, servers and software's.
- **Framework Design:** The system framework provided by the customer is converted into a framework according to the system requirement which includes testing and simulators.

Development of Automation

- **Test Script Creation:** At this stage, the manual tests are automated into test scripts.
- **Implement the Automation Framework:** The framework developed in the previous phase will be implemented for testing and simulation.
- **Data and Load Generation:** It tests the hardware, software and computational capabilities with huge amount of data generated from the physical things.
- **Building Execution Flow:** This stage enables the automated test environment to run using pre-defined and scheduled tasks.
- **Service Virtualization:** It is a process of revealing the nature of specific components in heterogeneous IoT-based systems.
- **Documentation:** This stage deals with the development of user guide to setup automation test environment with future integration.

Execution of Automation

- **New Test Environment Creation and Deployment:** In this stage, for test execution an environment will be virtually created and deployed.
- **Test Script Execution:** Based on the successful test script deployed, QA executes the following tasks in environment: i) functional validation on various devices, various networks, database consistency checks, connectivity tests, front-end and back-end tests, ii) device and communication protocols standard conformance, iii) Performance tests, iv) load tests, v) system level exploratory tests from user's perspective and vi) Finally security testing (Visoottiviseth et al, 2017) which covers the following features shown in the Table 7.
- **Result Reporting:** In this stage, a test report is generated after the execution of every test scripts.
- **Fault Management:** As discussed previously, the generated report will provide a brief description of faults and failures faced during the automation execution. Such bugs are registered in a client's log file and used by bug reporting tool.

Maintenance

- **Updation and Maintenance:** According to the changing functionalities, QA team will regularly update the test scripts, testing framework and so on for the effective execution of the automated test suite.
- **Regression Test:** This is a process which verifies the functionalities of the IoT system after every new build operation.

Table 7. Features of IoT security testing

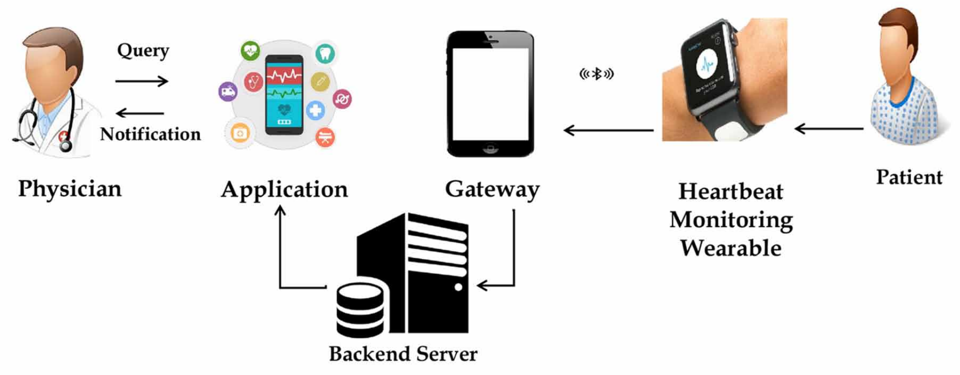
Insecure Web Interface	Data protection and encryption.
Insufficient Authentication /Authorization	Automated Hack attack scenario
Insecure Network Services	Lack of Transport Encryption
Privacy Concerns	Insecure Cloud Interface
Insecure Mobile Interface	Insufficient Security Configurability
Insecure Software /Firmware	Poor Physical Security

CASE STUDY: IoT TESTING USING BLE PROTOCOL

Bluetooth Low Energy (BLE) is one of the widely used protocols for IoT devices which consume limited power supply (Babusiak and Borik, 2015). Though, testing such IoT devices at large scale is tedious. Consider a BLE device which can operate for more than two years with a small battery. The sensor enabled by BLE can able to sense any vital sign from any human being, transmits the sensed data directly to the mobile device. In addition, the mobile device analyzes the data and recommends the various actions that need to be taken based on the analyzed data. This kind of scenario can be applied on various types of healthcare systems ranging from heartbeat monitoring of cardiac patients to the glucose level monitoring in diabetes patients. Such kind of IoT solutions improves the human life quality. Figure 12 depicts the heartbeat monitoring system of a person, in which the heartbeat rate is collected by the IoT device (sensor) and sent to the mobile application via BLE. The mobile applications sum-up the data and forward it to the server on cloud. The server can able to control the BLE device by using BLE requests. The physicians can request the data remotely in real-time and also receives the alert messages on emergency situations. The following figure shows the data flows between an IoT devices connected to a person via a mobile app to the backend server.

The Generic Attributes (GATT) of BLE protocol defines the two ways of data transferring using the services and characteristics (Leonardi et al, 2018). The data is stored in a table with a unique id and the protocol defines three kinds of actions for data entries namely READ, WRITE and NOTIFY. The default generic characteristics are defined by the BLE protocol but also the private characteristics can be defined using specific code in the monitoring application and BLE device (Leonardi et al, 2018).

Figure 12. Heartbeat monitoring system



For pairing, a BLE device broadcasts its services requests and the device which receives the request will initiate the connection. Generic device accept the connection but the other devices accept the connection based on the private characteristics which has been defined earlier. Once the devices paired, they can exchange data. The data transmission takes place based on the characteristics values and actions. The data transmission flow from IoT-based heartbeat device to mobile device after pairing is shown in the Figure 13.

The challenging factors with regard to the developing and testing BLE based IoT systems are:

- The location of IoT system and end-system (mobile device) must in be in same location.
- The time taken to develop IoT hardware device is high, so the parallel process of developing and testing the IoT software and hardware is impossible.
- Tester doesn't have any control over the data transmitted from IoT device to end device.
- In case of healthcare applications, the system requires more number of proven tests on a various devices.
- In offshore testing, communicating data to remote applications is a major challenge.

The illustration considered heartbeat monitoring system, consists of IoT device and a mobile application. The primary requirements for testing this application are: i) Tester, ii) Mobile Device and iii) IoT device. The testing process must be automated to acquire the best report and reorganize the system. It cannot be scaled to 'n' number of devices and tested. So, it is mandatory to automate the testing and it can be done only via software virtualization of BLE device in the cloud. Then the user can able to define or update the characteristics and the services, which are controlled by automation scripts.

Testing Scenario

A BLE server with 20 BLE dongles is added to the lab environment, in which each dongle can virtualize a single BLE device as shown in the Figure 14. The automation server with full script has the entire control over the IoT device and mobile devices. BLE services are configured using JSON, which defines the services. The mobile device receives the data from RF channel but it assumes the data is from BLE devices. Finally the commands are added to the automation system for carrying operations and to provide a command interface for testing the BLE applications.

Figure 13. Pairing process of IoT-based heartbeat device and mobile device

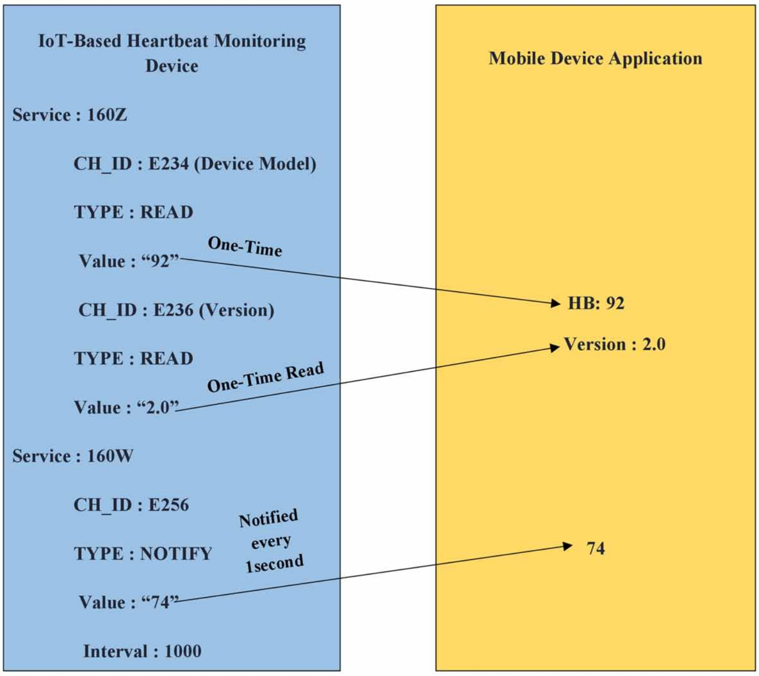
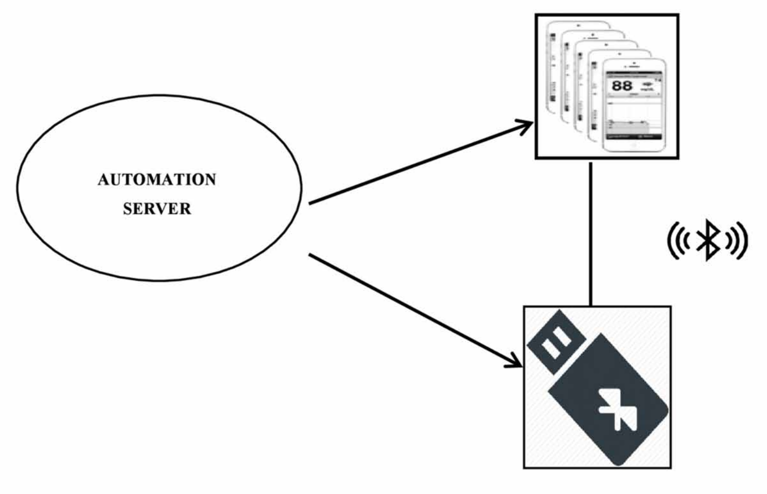


Figure 14. BLE device virtualization



SOLUTIONS AND BEST PRACTICES: IoT TESTING

IoT systems require robust and rigorous testing competences to ensure the quality and performance of the services must satisfy the requirements (IEEE Internet Technology Policy Community, 2017). IoT testing will be successful on adopting the operational practices and some of the important best practices are illustrated below:

- Readiness of properly defined system requirements along with a focused test plan, unit testing and integration testing brings the great significance.
- An efficient platform can provide good communication and processing capability which draws the valid information from the large amount of data, thus saves the time in terms of transmission and execution.
- For the accurate execution of the IoT application the QA testing team can make use of consoles, tools, simulators and viewers.
- Requires deep understanding of the software, hardware, architecture and protocols used for designing the IoT system.

IoT TESTING: BIG CHALLENGES

The vision of QA teams is intelligent towards devices and sensors testing. An IoT-based smart system generates the large volume of data that introduces the more technical complexities, requires an inclusive approach to handle. IoT systems have various unique factors:

- A single system which incorporates hardware, software, communication, gateway and sensors.
- Real-time streaming analytics and event processing.
- Provides provision for large amount of data, rapidity, diversity and reliability.
- Large-scale data visualization.

The above characteristics present the set of challenges on performing testing on IoT systems. The challenges of IoT testing are beyond software implementation and hardware devices because IoT adds new complexity parameters to the classic test models (Al-Fuqaha et al., 2015). The challenges can be discussed into two types: 1) Primary Challenges and 2) Operational Challenges.

Primary Challenges

- **Dynamic Environment:** The traditional testing environment is a static environment, whereas the environment of IoT systems is drastically dynamic with millions of sensors, devices and intelligent software.
- **Real-Time Complexity:** As IoT systems are entirely based on multiple, real-time scenarios and hence its use cases are tremendously complex.
- **System Scalability:** Developing a test environment for assessing the functionalities of the IoT system are much complex because it is highly scalable in nature.
- **Expensive Replication:** During IoT testing, replicating the same environment is much expensive and demands too much of efforts.

Operational Challenges

- **Security:** The architecture of IoT is made of multiple layers, each if which have their own security vulnerabilities.
- **Automation Risk:** The main intention of IoT system is to automate the task, but innumerous dependency issues make the automation job highly challenging.
- **Quality Sensors and Accuracy:** The devices incorporated in IoT system may or may not be of high quality or it may not provide accurate data.
- **Complex Use Cases:** IoT systems have hard and complex set of use cases, thus increases the risks in test case and test data generation.

Each and every types of testing has its own challenges, which are specified in the Table 8.

Table 8. Challenges vs. testing types

Types of Testing	Challenges
Functional Testing	Difficult to re-create and end-to-end setup
Performance Testing	Scale: Devices, communication and computation
Security Testing	Data security: Sensor Accessibility, Hazards
Interoperability Testing	Heterogeneous Devices and Connectivity
Exploratory Testing	New Field, Plenty of Unknowns
Usability Testing	Test for ease of use, Accuracy, Expectations

CONCLUSION

The influence of IoT technology in multiple domains increased the chance for software development and testing the IoT applications over the time. In near future the number of hardware devices and software devices in IoT will grow rapidly. As an outcome of such growth there will be a need for more testing solutions and experienced testers for delivering high quality IoT components to end users. Moreover is purposeful to define the set of testing requirements and strategies are essential to the effectual performance of the specific IoT device. Based on the background study it has been found that industrial and academic reports in the field of IoT testing are limited. Hence, this chapter broadly describes the needs for IoT testing in terms of various criteria of identifying and fixing the problems in the IoT systems. The IoT testing focusing area and the classification of IoT testing scope enables the testers to accurately identify and apply the testing strategies for any IoT applications. For deeper insight into the IoT testing, a testing framework is defined for IoT-based Medical Healthcare Tracking System with respect to various testing process. The discussion on the testing automation based on TaaS approach is expected to support the development of efficient IoT system and make them to perform as expected. This chapter summarized the IoT solutions and best practices to be followed on testing and explores the challenges in IoT testing with respect to various aspects. Entirely, this chapter covers the major constituents of IoT testing and is expected to be helpful for academicians, researchers and system developers for their future enhancements.

REFERENCES

W3C-Group. (2016). *Direct to device connectivity in the internet of things*. Retrieved from https://www.w3.org/WoT/

Abu Oun. (2015). *Designing multiscale hybrid platform for testing and evaluating IoT systems*. Université de Franche-Comté.

Adjih, C., Baccelli, E., Fleury, E., Harter, G., Mitton, N., Noel, T., . . . Watteyne, T. (2016). FIT IoT-LAB: A large scale open experimental IoT testbed. *IEEE World Forum on Internet of Things, WF-IoT 2015 - Proceedings*, 459–464.

AFour Technologies. (2017). *IoT Testing Services*. Retrieved from https://afourtech.com/iot-testing-services/

Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys and Tutorials*, *17*(4), 2347–2376. doi:10.1109/COMST.2015.2444095

Atzori, Iera, & Morobito. (2010). The Internet of Things: A survey. *Journal of Computer Network, 54*(15), 2787-2805.

Babusiak, B., & Borik, S. (2015). Low energy wireless communication for medical devices. *38th International Conference on Telecommunications and Signal Processing (TSP)*, 444-447. 10.1109/TSP.2015.7296301

Bandyopadhyay & Sen. (2011). Internet of Things: Applications and challenges in technology and standardization. *Springer International Journal of Wireless Personal Communications*, 49-69.

Bertino, E., Choo, K. K. R., Georgakopolous, D., & Nepal, S. (2016). Internet of Things (IoT): Smart and secure service delivery. *ACM Transactions on Internet Technology*, *16*(4), 22. doi:10.1145/3013520

Bormann, C. (2014). *Test descriptions for ETSI plug tests coap 4*. Eur. Telecommunication Standards Institute London, U.K. Tech. Rep. 7-9.

Che, X., & Maag, S. (2013). A passive testing approach for protocols in Internet of Things. *IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*, 678–684. 10.1109/GreenCom-iThings-CPSCom.2013.124

Cigniti. (2018). *The Need for Testing the Internet of Things*. Retrieved from https://www.cigniti.com/blog/the-need-for-testing-the-internet-of-things/

Cigniti. (2018). *Why the Healthcare Sector Needs QA & Testing*. Retrieved from https://www.cigniti.com/blog/top-6-reasons-healthcare-sector-needs-qa-testing/

Cognizant. (2016). *The internet of things: QA unleashed*. Retrieved from https://www.cognizant.com/InsightsWhitepapers/theinternet-of-things-qa-unleashed-codex1233.pdf

Desnitsky, V., & Kotenko, I. (2016). Automated design, verification and testing of secure systems with embedded devices based on elicitation of expert knowledge. *Journal of Ambient Intelligence and Humanized Computing*, *7*(5), 705–719. doi:10.100712652-016-0371-6

Ericsson. (2011). *More than 50 billion connected devices*. White Paper 284 23-3149 Uen.

Esquiagola, J., Costa, L., Calcina, P., Fedrecheski, G., & Zuffo, M. (2017). Performance Testing of an Internet of Things Platform. *Proceedings of the 2nd International Conference on Internet of Things, Big Data and Security*, 309-314. 10.5220/0006304503090314

Fernández-Caramés, T. M., Fraga-Lamas, P., Suárez-Albela, M., & Castedo, L. (2016). Reverse Engineering and Security Evaluation of Commercial Tags for RFID-Based IoT Applications. *Sensors (Basel)*, *17*(1), 28. doi:10.339017010028 PMID:28029119

Fielding, R. (1999). *Hypertext Transfer Protocol-HTTP/1.1, document RFC 2616*. Network Working Group. doi:10.17487/rfc2616

Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of things (iot): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, *29*(7), 1645–1660. doi:10.1016/j.future.2013.01.010

Gyory, N., & Chuah, M. (2017). IoTOne: Integrated platform for heterogeneous IoT devices, *International Conference on Computing, Networking and Communications (ICNC)*, 783-787. 10.1109/ICCNC.2017.7876230

Hillah, L. M., Maesano, A. P., De Rosa, F., Kordon, F., Wuillemin, P. H., Fontanelli, R., & Maesano, L. (2017). Automation and intelligent scheduling of distributed system functional testing. *International Journal of Software Tools for Technology Transfer*, *19*(3), 281–308. doi:10.100710009-016-0440-3

Hung, M. (2017). Leading the IoT. *Gartner Insights on How to lead in a Connected World*. Retrieved from https://www.gartner.com/imagesrv/books/iot/iotEbook_digital.pdf

IEEE Internet Technology Policy Community. (2017). *Internet of Things (IoT) Security Best Practices*. Retrieved from https://internetinitiative.ieee.org/images/files/resources/white_papers/internet_of_things_feb2017.pdf

Kim, H., Ahmad, A., Hwang, J., Baqa, H., Le Gall, F., Reina Ortega, M. A., & Song, J. S. (2018). IoT-TaaS: Towards a Prospective IoT Testing Framework. *IEEE Access: Practical Innovations, Open Solutions*, *6*, 15480–1549. doi:10.1109/ACCESS.2018.2802489

Kiruthika, J., & Khaddaj, S. (2015). Software Quality Issues and Challenges of Internet of Things. *14th International Symposium on Distributed Computing and Applications for Business Engineering and Science*, 176-179. 10.1109/DCABES.2015.51

Leonardi, L., Patti, G., & Lo Bello, L. (2018). Multi-Hop Real-Time Communications Over Bluetooth Low Energy Industrial Wireless Mesh Networks. *IEEE Access: Practical Innovations, Open Solutions*, *6*, 26505–26519. doi:10.1109/ACCESS.2018.2834479

Li, S., Da Xu, L., & Zhao, S. (2015). The internet of things: A survey. *Information Systems Frontiers*, *17*(2), 243–259. doi:10.100710796-014-9492-7

Lin, H., & Bergmann, N. W. (2016). IoT privacy and security challenges for smart home environments. *Information*, *7*(3), 44. doi:10.3390/info7030044

Lunardi, W. T., de Matos, E., Tiburski, R., Amaral, L. A., Marczak, S., & Hessel, F. (2015). Context-based search engine for industrial iot: Discovery, search, selection, and usage of devices. *IEEE 20th Conference on Emerging Technologies Factory Automation*, 1–8.

Marinissen, E. J., Zorian, Y., Konijnenburg, M., Huang, C. T., Hsieh, P. H., Cockburn, P., & Verbauwhede, I. (2016). Iot: Source of test challenges. *21th IEEE European Test Symposium*, 1-10.

MarketsandMartkets. (2017). *Internet of Things (IoT) Testing Market by Testing Type (Functional, Performance, Network, Security, Compatibility, and Usability), Service Type (Professional and Managed), Application Type, and Region - Global Forecast to 2021*. Retrieved from https://www.marketsandmarkets.com/Market-Reports/iot-testing-market-51412648.html

Montenegro, Kushalnagar, Hui, & Culler. (2007). *Transmission of IPV6 Packets Over IEEE 802.15.4 Networks*. document RFC 4944, 2007.

One M2M Testing Framework, document oneM2M TS-0015 v2.0.0, Aug. 2016. (n.d.). Retrieved from http://www.onem2m.org/images/files/deliverables/Release2/TS-0015-Testing_Framework-V2.0.0.pdf

Pering, T., Farrington, K., & Dahm, T. (2018). Taming the IoT: Operationalized Testing to Secure Connected Devices. *IEEE Computer*, *51*(6), 90–94. doi:10.1109/MC.2018.2701633

Raymor, B., & Coppen, R. (2014). *OASIS Message Queuing Telemetry Transport (MQTT) TC*. Retrieved from https://www.oasis-open.org/committees/mqtt/

RCR-Wireless. (2016). *Testing the internet of things: Making the IoT work*. Author.

Reetz, E. S., Kuemper, D., Moessner, K., & Toenjes, R. (2013). How to test IoT-based services before deploying them into real world. *Wireless Conference (EW), Proceedings of 19th European Wireless Conference*, 1–6.

Sajid, A., & Abbas, H. (2016). Data privacy in cloud-assisted healthcare systems: State of the art and future challenges. *Journal of Medical Systems*, *40*(6), 155. doi:10.100710916-016-0509-2 PMID:27155893

Saksoft. (2018). *IoT Interoperability Testing*. Retrieved from https://www.360logica.com/blog/iot-interoperability-testing

Sanchez, L., Muñoz, L., Galache, J. A., Sotres, P., Santana, J. R., Gutierrez, V., ... Pfisterer, D. (2014). SmartSantander: IoT experimentation over a smart city testbed. *Computer Networks*, *61*, 217–238. doi:10.1016/j.bjp.2013.12.020

Shanmugasundaram. (2015). *IoT Basics and Testing Focus.* Retrieved from https://theinternetofthings.report/Resources/Whitepapers/793406e1-2095-40a5-9369-70d3df83e844_iot_basics_and_testing_focus.pdf

Shelby, Z., Hartke, K., & Bormann, C. (2014). *The Constrained Application Protocol (COAP), document RFC 7252, Internet Engineering Task Force*. IETF.

Sicari, S., Rizzardi, A., Grieco, L. A., & Coen-Porisini, A. (2015). Security, privacy and trust in Internet of Things: The road ahead. *Computer Networks*, *76*, 146–164. doi:10.1016/j.comnet.2014.11.008

Sogeti, C. (2016). *HPE World Quality Report*. Retrieved from https://www.capgemini.com/resources/world-quality-report-2016-17/

Song, J., Kunz, A., Schmidt, M., & Szczytowski, P. (2014). Connecting and managing M2M devices in the future Internet. *Mobile Networks and Applications*, *19*(1), 4–17. doi:10.100711036-013-0480-9

TestingWhiz. (2018). *How Test Automation can be Helpful for IoT Applications*. Retrieved from https://www.testing-whiz.com/blog/how-test-automation-can-be-helpful-for-iot-applications

Thangavel, D., Ma, X., Valera, A., Tan, H., & Tan, C. K. (2014). Performance evaluation of MQTT and CoAP via a common middleware. *IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, 1-6. 10.1109/ISSNIP.2014.6827678

Vandikas, K., & Tsiatsis, V. (2014). Performance evaluation of an IoT platform. *Proceedings - International Conference on Next Generation Mobile Applications, Services and Technologies*, 141–146.

Visoottiviseth, V., Akarasiriwong, P., Chaiyasart, S., & Chotivatunyu, S. (2017). PENTOS: Penetration testing tool for Internet of Thing devices. TENCON 2017 - 2017 IEEE Region 10 Conference, 2279-2284.

Wittstock, V., Lorenz, M., Wittstock, E., & Pürzel, F. (2012). A Framework for User Tests in a Virtual Environment. Advances in Visual Computing, 358-367.

Worthy, P., Matthews, B., & Viller, S. (2016). Trust me: doubts and concerns living with the Internet of Things. *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*, 427-434. 10.1145/2901790.2901890

Xu, T., Wendt, J. B., & Potkonjak, M. (2014). Security of IoT systems: Design challenges and opportunities. *Proceedings of the 2014 IEEE/ACM International Conference on Computer-Aided Design, IEEE*, 417-423.

Ziegler, S., Crettaz, C., Ladid, L., Krco, S., Pokric, B., Skarmeta, A. F., & Jung, M. (2013). Lecture Notes in Computer Science: Vol. 161-172. *IoT6 Moving to an IPv6-based future IoT. The Future Internet Assembly.*

ADDITIONAL READING

Bull, C., Euteneuer, S., & Gawlik, K.-U. (2016). *Testing the Internet of Things-Intelligence is Required*. SQS Group Limited UK.

KEY TERMS AND DEFINITIONS

API: Application programming interface (API) is a set of definitions that specifies how the IoT devices interact with each other. It also provides tools and protocols for designing IoT software applications.

BLE Protocol: Bluetooth low energy (BLE) protocol is a protocol specifically designed for wireless personal area network (WPAN), which contains a stack to executing connection establishment, device discovery, and connection termination between peer devices.

CoAP: Constrained application protocol (CoAP) is a protocol which allows low-powered computing devices can operate in the internet of things (IoT) and this protocol is specifically designed for machine-to-machine (M2M) applications.

M2M: Machine to machine (M2M) refers to the inter-connected machine via wireless communication technology that transfers data between them and perform operations without any manual intervention.

MQTT: Message queuing telemetry transport (MQTT) is an IoT connectivity and lightweight protocol specifically designed for sensors and mobile devices to manage information distribution.

Chapter 6
IoT Functional Testing Using UML Use Case Diagrams:
IoT in Testing

D.Jeya Mala
Thiagarajar College of Engineering, India

ABSTRACT

In the IoT applications development process, the consumers expectations are always high. Thus, the development environment should be focusing on virtual provisioning, manipulation, and testing and debugging. This has also raised more challenges in terms of proper testing to be done in both user interface level as well as the functionality level. It will be really challenging to test a connected device within a full IoT environment, which will have more devices with varied functionalities and data processing. These challenges have made a new way of testing to be done so that the test cases will be more efficient in revealing the errors in the software. In this chapter, UML use case diagram-based test cases generation for an IoT environment is explained in detail. Also, a real-time case study IoT application is taken to showcase how this approach helps in generating the test cases to test the embedded software in these IoT devices in terms of data flow, control flow, and functionalities with improved performance.

DOI: 10.4018/978-1-5225-7790-4.ch006

INTRODUCTION

The Internet of Things (IoT) is the current evolving technology that makes the connection of hardware devices and software components to be connected seamlessly in order to facilitate information exchange in a collaborative working environment. As per Gartner in his survey indicated that the number of things connected in 2017 is 8.4 billion and it may reach approximately 20.4 billion in the year of 2020 (CapeGemini's white paper 2018).

Generally, Internet of Things or IoT is used to connect things to the internet by means of variety of connectivity options with more devices are employed to capture different types of data. Any kind of things that are used in IoT applications ranging from Air Conditioners, Security Cameras, Cable Set-Top boxes, vehicles to industrial systems such as conveyer belts, manufacturing machines and traffic signals, smart phones and any kind of devices that can be powered.

Thus smart homes with smarter appliances, smart automobiles, wearable technology and robotic applications have made IoT as everyday reality. During past software development process, if software is written in a chip so that it can be embedded within a physical device has termed the software as Embedded Software. Now, this has been expanded to connected devices or products so that IoT is achieved. This has been termed as IoT revolution which has enforced changes to be done in the entire development process.

At the Consumer Electronics Show (CES) in the year 2015, Samsung CEO Boo-Keun Yoon stated that the IoT is "not science fiction anymore. It's science fact". Now, a smooth paradigm shift is happening in industries in which they are thinking on how to increase the performance of the overall system when the devices or things are integrated together.

Nowadays, the technological advancements such as mobile, internet, cloud and virtual development and operating environments makes a strong need for IoT and is being considered as one of the most important and crucial technology. This technology rapidly advances in almost all the industries which thus increases the numbers of firms adopt the technology.

While all these benefits are really more appreciable to be used in the IT industry, there are certain critical aspects or components in IoT need to be addressed with higher level of importance. This has raised certain important considerations during the application of this technology in the field use such as interoperability, personalizing services to exchange information, performance improvement and operations control and information processing. In fact, a survey taken during 2010, among nearly 500 embedded software engineers they responded that, Software is the most crucial component in the embedded systems and gives rise to more than 50% value for its proper working.

For physical connectivity between devices in a computing environment, generally computer networks are used. But in the case of IoT applications, device-to-device and human-to-device interactions must be done in a reliable robust manner. This has also raised some challenges in terms of data dependency, control dependency, information interchange and timely executed functionalities.

Here, in the case of pure device to device communication, there is no need for data visualization whereas nowadays, as the IoT applications are more human centered, the data needs to be visualized to present the required processed information to the end users in an easy to understand manner. Also, many of the IoT devices have intelligence in order to identify the current status, exchange information between them, monitoring the working environment and also took some intelligent decisions to act on any problems identified without any human intervention.

In the IoT applications development process, the consumers expectations are always high which thus makes the development environment should be focusing on virtual provisioning, manipulation and testing and debugging.

This has provided an insight that, the devices are more software specific with intelligence being a part of it. This has also raised more challenges in terms of proper testing to be done in both user interface level as well as the functionality level. It will be really challenging to test a connected device within a full IoT environment which will have more devices with varied functionalities and data processing.

These challenges have made a new way of testing to be done so that, the test cases will be more efficient in revealing the errors in the software. In this chapter, UML Use Case diagram based test cases generation for an IoT environment is explained in detail. Also, a real time case study IoT application is taken to showcase how this approach helps in generating the test cases to test the embedded software in these IoT devices in terms of data flow, control flow and functionalities with improved performance.

BACKGROUND

Kim et.al (2018) have provided an IoT testing framework to resolve constraints with respect to coordination between devices, costs and scalability issues in the context of IoT devices development. The framework they have proposed was called as IoT-TaaS that is composed of scalability of distributed and remote testing, automated conformance testing and validation testing in IoT environment.

Greg Sypolt (2016) who is a Senior Engineer at Gannett – USA Today Network and co-founder of Quality Element has posted in an white paper at Sauce Labs, elaborated the importance of functional testing in IoT devices. He has also explained the challenges related to functional testing of IoT and the need for new innovative techniques to perform it. In this post, the need for creating virtual devices to simulate the real-time environments and their connectivity is also discussed. The core components of the product should be focused by applying the existing functional testing principles to IoT products and websites. Then these components will be tested using the test cases derived from the functional testing principles.

In a white paper of QualiTest (2018), the challenges related to IoT testing have been discussed with their industrial experience. This paper indicated the need for organizing the Quality Assurance (QA) and testing functions with a combination of both centralized and decentralized approaches. Hence, a testing team should be tightly integrated into the development process of IoT application development. It suggested a DevOps and Agile based models to achieve this continuous testing and output.

In HCL's white paper (2015), Rajesh Shanmugasundaram has discussed four testing scopes for testing any IoT-enabled product. Those components are: Components validation, Function validation, performance validation and Security and Data validation. The suggestion is to make the testing team to concentrate on these core components to treat functional and connectivity testing as the most important elements in IoT testing. Also, based on the dependency level of the IoT product, the testing should be prioritized. The various types of testing discussed in this white paper are given below:

- **Functional Testing:** To check the way the consumer wants the output, based on specific inputs given to the IoT app
- **Compatibility Testing:** Categories verify and validate the possible combination of device versions, communication protocol versions, mobile devices, resolutions, and mobile OS version
- **Usability Testing:** Verify user experience of IoT app, with respect to its usage, visibility of text, appeal, and usefulness of the content to the end user
- **Network Testing:** Verify the IoT app with different network connections and ensure app to sync with all different backend combination protocol
- **Security Testing:** Validate privacy of data, reliability of IoT app, verification, availability, and authorization are the factors that need to be considered. Also verifies if the IoT app uses any weak password or missing data encryption. Verify apps followed network security standards and authentication mechanism to authenticate of the required app.
- **Performance Testing:** To check the overall performance of the IoT app and validate the response time based on different user loads, optimize the code to improve the performance, verify with different scenarios like low battery, less memory, switch between different networks.

- **Services Testing:** Check the IoT app enabled with different service requests with backend system with online and offline mode. Also verifies request and response of system.
- **Operational Testing:** To verify behavior of IoT app when battery fully discharged when updated version app get installed or for any interruption of message or call received

In the Infosys white paper (2017), the differences between traditional performance testing and IoT performance testing have been provided. In the case of IoT performance testing, the simulation should be done for devices and sensors, the request and response sequences, measurement of performance based on various loads have been the key issues. This white paper has also projected the tool they have developed for IoT performance testing.

Rosborough (2018) in his blog post, explained key aspects of dynamic testing for IoT security. As the IoT applications are ranging from home, industry and even to military based applications, security becomes the most important concern in these applications. Some of the security threats includes, data collected and shared between various parties involved in the IoT environment, anonymity in sharing data, cyber attacks, hackers attacks, layered security, failure rate etc. needs to be considered in providing dynamic testing techniques for IoT security. The Black box testing method of functional testing should also focus on vulnerabilities assessment.

In a white paper of CTS (2015), the entire QA of IoT has been divided into two layers. The first layer is device interaction layer and the other is user interaction layer. The device interaction layer includes the testing of conformance of standards such as data protection, encryption and storage in local and cloud. The user interaction layer includes network capability and device level tests, usability and user experience and back-end IoT environment. Here, functionality testing should include Web/UI, embedded and back-end computing.

The Test and Verification Solutions in their white paper () have exhibited the V&V of autonomous software in IoT devices. The V&V process includes safety, reliability, availability, resilience and security testing. Here, testing should not only include traditional testing approaches but also dynamic analysis and testing.

Lee and Lee (2015) in their paper have presented five IoT technologies to be used in the deployment of successful IoT based products and have also provided some techniques to enhance the customer value.Also, they discussed the technical and management challenges in the development of IoT based applications.

In a white paper published by Zephyr (2017) have provided the future of IoT testing. In their finding, they mentioned the need for testing to cover data storage and restoration when a connection is unexpectedly stopped and resumed respectively. Also, it exposes the testing to be done around switching between connections. As the testing of IoT devices in a controlled environment is different from testing the same devices under the actual deployment environment, the testing procedures for IoT should include interfaces testing in real time usage of IoT devices to provide continuous data transfer.

Software Testing for IoT

Any IoT application will have software embedded in the devices and this software helps to provide the required functionality from the application. This software contains a set of functionalities and/or user interface as it is required from one application to another.

The analysis has revealed that, this software must be tested for functionality and interface errors as otherwise it will lead to field failures on deployment.

The different types of testing in IoT application are:

- Unit testing
- Functional Testing
- Load Testing
- Security Testing
- Exception Testing
- Workflow Testing

Unit Testing

In this type of testing, each and every unit/module will be tested for its logical errors. Generally, this testing will be done by the developer/tester in an organization. The test cases generated will be purely based on the code generated. Here, data flow, control flow and exceptions are tested for the proper functioning of the software. This testing is normally coming under "White Box testing" technique in which the logic is tested for its correctness.

For IoT based applications, generally the code part will not be too complex rather they will be simple enough so that they can be executed faster to provide the required functionality. Hence, for any IoT based applications, the developer may use any of the development platforms such as Adriano Studio to develop and debug their software written for the specific IoT device/component.

Functional Testing

This testing is coming under “Black Box Testing” technique and is used to examine the functionalities of the system. Here, the functional elements will be tested based on the customers’ requirements. The requirements are collected from the customers and are refined and formulated as a collection of well defined requirements. These requirements are then analyzed to generate test cases to test the final product.

In this type of testing, the focus in not on the internal logical errors in the software rather, it will be on the behavioral aspects of the system. The test cases are generated to test the functionalities present in the system using various techniques such as Equivalence Class Partitioning, Boundary Value Analysis and Cause Effect Graphing etc for applications such as web based, desktop or some interface based applications.

In the case of IoT applications, we require models as the functionalities cannot be easily tested with the device as there won’t be any visual interface as in the case of other applications. Hence, the static models such as Use Case Diagrams and dynamic models such as Finite State Machines could be used to generate test cases for functional testing.

Load Testing

Here, the system is validated for its performance under varying loads. In the case of web based systems, number of requests/sec to a web server, time to open a web page when too many images are loaded at any point of time are typical load testing scenarios.

In the case of IoT based applications, the situation is different, the load testing helps to check whether the device is capable of delivering its required functionality even under heavy loads such as Sensor data is too higher/second to process.

Security Testing

Every application needs to be tested for its security aspects in terms of data and access privileges. This can be done by means of Penetration Testing, User access control mechanisms etc.

In the case of IoT testing, the security aspects are highly vulnerable which will affect the entire systems’ field use. Hence, for IoT applications, security must be assessed with various threat models, fault sequences, fault models and so on.

Exception Testing

Monitoring an application for problems like runtime errors, buffer overflow, memory leak, resource leak, etc. is termed as exception testing.

For IoT applications this exception testing is highly essential as they are testing the run time behavior of the system.

Workflow Testing

This is done to ensure if the interface engine handles the specified workflow as expected or not. It helps in monitoring the communication between various components in an IoT application.

UML Diagrams in Software Development

The Unified Modeling Language (UML) has become a sizzling focus in the software industry. UML happens to be the creole pidgin language in the software development industry facilitating the stakeholders to exchange their designs without any difficulty.

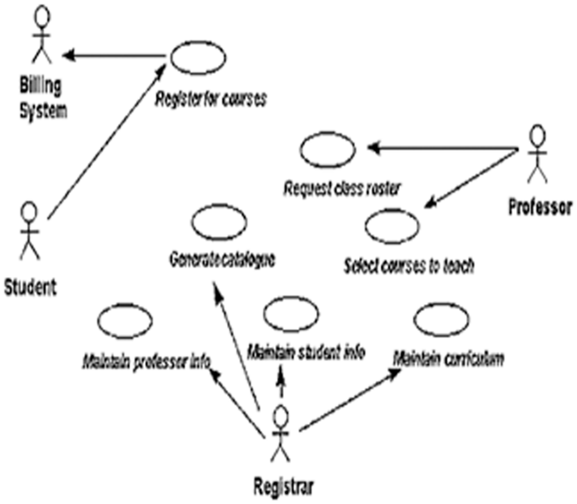
This object-oriented system of notation has evolved from the collaborative work of Grady Booch, James Rumbaugh, Ivar Jacobson, and many others. These renowned computer scientists fused their respective technologies into a single, standardized model. Today, UML has been accepted by the Object Management Group (OMG) as the standard for modeling object oriented programs.

UML symbols closely resemble those of the Booch and OMT notations, and also include elements from other notations. UML defines thirteen types of diagrams: Class diagram, Object diagram, Package diagram, Component diagram, Composite structure diagram, Deployment diagram, Use case diagram, Activity diagram, State-machine diagram, Sequence diagram, Communication diagram, Interaction overview diagram and Timing diagram. Figure 1 shows a sample of UML notation.

The Unified Modeling Language (UML) is a diagramming language or notation representing unified best engineering practices for specifying, visualizing, constructing and documenting the components of business modeling, software and even non-software systems.

- **Specification**: The UML can be used for specifying "what" is required from a system and "how" a system may be implemented. It captures all the important requirements, analysis, design and implementation decisions that need to be established during a system development lifecycle.

Figure 1. Sample UML diagram



- **Visualization:** The graphical support provided by UML allows the visualization of the entire systems before their implementation. The use of shapes with well defined semantics, for communication with a cross-section of audience proves to be more concise and precise than a descriptive narrative and more comprehensive form of communication represented by a programming language.
- **Construction:** The UML can be employed to direct and craft the implementation of a complex system. Furthermore, with the aid of a variety of case tools on the market, it's quite likely to generate object oriented source code from UML models, while it's also possible to reverse engineer the source code in to UML models.
- **Documenting:** The UML offers an effective means of capturing knowledge and documenting the deliverables such as requirements, documents, functional specifications and test plans. All these are critical in controlling, measuring, and communicating a system through out its life cycle.

UML is definitely not a development method, which means it does not convey you the steps of implementation like what to do first and what to do next or how to design your system, but it helps you to visualize your system design and communicate with others.

UML may be used to support a number of methodologies, such as rational unified process. Some methodologies are more suitable to mammoth enterprise applications with a huge team of designers and developers, while others are more apt for a single person or small teams working on small embedded systems. UML accommodates the requirement of both of these extreme methodologies. Another feature favouring the use of UML is that there are many UML development tools available such as Rational Rose, Enterprise Architect, Eclipse, Umbrello UML Modeler, Altova, MagicDraw, Describe and even Microsoft Visuo.

Due to the proliferation of the rapid application development (RAD) tools like Delphi or Visual Basic, the process of developing a software system is fairly easy. These tools provide support with a drawing package, which people with a basic programming knowledge, can easily design the layouts and create forms with a simple drag-and-drop or double-clicking operation. But the caveat here, is do these applications possess a professional quality?

Most of the people using RAD tools believe that these tools help them to develop an application quickly. Directly they start with the implementation phase without allotting sufficient time for analysis and design. The RAD tools force the developers to write a prototype, and then keep adding more code until the application is complete. This is done iteratively till the application runs fundamentally well. But this scheme has a major problem. The application developed in this manner lacks a formal, well-defined architecture since the developers would not have thought about it. The basic object oriented principles would have been compromised and thus it leads to an undocumented, inefficient, change-resistive and maintenance-tough code.

This wide of the mark tendency of the RAD developers is shifted from direct leap into development phase to the analysis and design phases through the usage of an appropriate UML development tool and an applicable process or methodology. Therefore ample scope is available for reducing the risks and providing a vehicle for testing the architecture of a system before the coding is started. The analysis and design overhead will eventually pay dividends as the system has been user driven, well-documented and when its time to start developing, the UML tools will generate the skeleton code that will be efficient, object oriented and support re-use.

Furthermore, the use of UML offers the following benefits.

- Software systems are proficiently designed and documented before they are coded and hence all the stakeholders are precisely aware of what they are getting, in advance
- Since system design precedes the coding, UML facilitates re-usable section of the code to be identified easily and coded with the highest efficiency, thereby reducing the software development costs

- UML easily spots the logic 'holes' in design drawings and hence ensures that the software will behave as expected
- The overall system design described in UML will direct the way in which the software is developed and hence right decisions are made early on in the process. Again, this minimizes software development costs by eliminating re-work in all areas of the life cycle.
- UML presents an enterprise level, bird's-eye view of the entire system and, as a result, competent memory and processor efficient systems can be designed
- UML supports easy maintenance of the application, by providing more effective visual representations of the system which are precisely understood by the maintenance team. Consequently, maintenance costs are reduced.
- UML diagrams aid in providing resourceful training to new members of the development team
- UML provides an effective medium of communication with both internal and external stakeholders since it documents the system much more efficiently

The purpose of the UML diagrams is to present the multiple views of a system – the collection of these multiple views is called a model. The following sub-sections briefly describe the most common diagrams of the UML and the intended use of each of them. The appreciable feature with these diagrams is that they can be combined to yield hybrid diagrams and they have provisions to be extended.

The various UML diagrams that are used in each of the OO development phases are given as follows:

At the end of requirements gathering phase, the analysts specify the requirements through

- Use case diagram – Business and Secondary use case diagrams
- Use case description

At the end of the Design phases, the deliverable UML diagrams are

- Design classes, packages and subsystems
- Class diagram
- State chart diagram
- Object diagram

The run time architecture are described from

- Process diagram
- Activity diagram

The architecture is described from

- Deployment diagram

It is easily understood from the above list that UML serves to be a consistent language for specifying, visualizing and documenting OO systems and thus it facilitates a comfortable platform for the OO methodology based software development.

The interesting feature to notice about the UML is that there are a bunch of different diagrams (models) to get used to. This is due to the fact that a system is looked at from many different viewpoints. A software development will have many stakeholders playing a part, as mentioned earlier – for example:

- Analysts
- Designers
- Coders
- Testers
- QA
- The Customer
- Technical Authors

Every category of these people is interested in different aspects of the system, and each of them needs a diverse level of detail. For instance, a programmer has to understand the design of the system and be able to translate the design to a low level code. In contrast, a technical writer is focused towards the behaviour of the system as a whole, and requires an understanding of how the product functions. The tester needs to understand the functionalities of the various modules individually as well as the interactions among the modules so that he generates suitable test cases to fix the bugs present in the code.

Meticulous system design involves all the possible viewpoints, and each UML diagram gives you a way of incorporating a particular view. The objective is to communicate clearly and unambiguously with all the stakeholders. The UML attempts to provide a common language sufficiently expressive so that all these stakeholders can benefit from at least one UML diagram.

UML Use Case Diagram: An Introduction

It is the graphical overview of the functionality and requirement of the system and the interface with outside the system, as well as it shows the actors and the relationship between the actors and the use cases. Primary use case diagram shows the design features. Moreover, the primary use case diagram is the first point when designing new system by using UML and when explaining the requirement for the system in analysis, implementation and documentations stage. The primary use case diagram elicits the overall functions that are expected to be accomplished by the system.

Using scenario-based requirements elicitation, the involved stakeholders are queried for the variety of tasks they want the system to do. They are asked about how they visualize the system in use. Then these system problem statements are mapped to respective system specification; the specification is denoted as a set of actors and use cases, which are discussed below. The business analyst team works with the customer to enumerate a complete set of possible scenarios, which are documented in simple natural language (as opposed to using any formal notation) in customer's terminology itself. The thing is that every single task expected to be carried out by the system should be brought out in the complete set of scenarios. Scenarios are quite useful for eliciting, validating, and documenting the requirements. Scenario-based approaches help to connect and map the user/stakeholder view and the functional view of the future system so that the expected system under development will meet the intended requirements of its users. Hence scenario based approaches find abundant usage within industry

Use Case Flow-of-Events

The use case diagram helps easy visualization of a system. Still, a textual description of the sequence of transactions of a use case is also needed additionally to understand better what really happens in a use case. In this part, the use case *flow-of-events,* a description of what the system should do is presented. The flow-of-events is given in terms of *what the system should do*, not how the system does it.

Several different templates are available for documenting a use case flow of events. The prefect structure of these templates may differ slightly from version to version. The essence is that the Flow-of-events should convey the course of any use case. One such template is shown below:

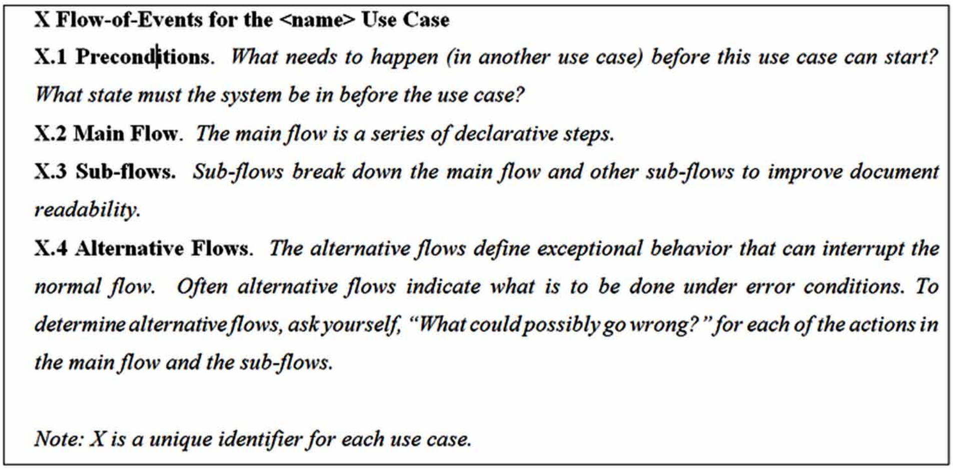
Below is an example flow-of-events for the Application Blocker Use Case. The example uses the template of Figure 3 to structure the flow of events.

NOTATIONS USED IN USE CASE DIAGRAM

The following are the six modeling elements that frame the Use Case diagram: systems, actors, use cases, associations, dependencies, and generalizations.

1. **System:** Defines the boundary of the system with respect to the actors who use it (external to the system) and the features it must execute (inside the system).

Figure 2. Use case flow-of-events template



X Flow-of-Events for the <name> Use Case

X.1 Preconditions. *What needs to happen (in another use case) before this use case can start? What state must the system be in before the use case?*

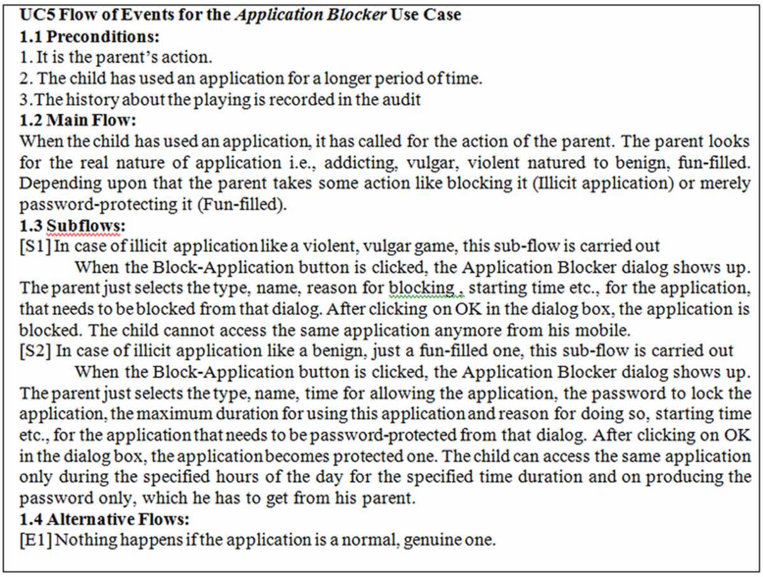
X.2 Main Flow. *The main flow is a series of declarative steps.*

X.3 Sub-flows. *Sub-flows break down the main flow and other sub-flows to improve document readability.*

X.4 Alternative Flows. *The alternative flows define exceptional behavior that can interrupt the normal flow. Often alternative flows indicate what is to be done under error conditions. To determine alternative flows, ask yourself, "What could possibly go wrong?" for each of the actions in the main flow and the sub-flows.*

Note: X is a unique identifier for each use case.

Figure 3. Use case flow-of-events template for application blocker use case



UC5 Flow of Events for the *Application Blocker* Use Case
1.1 Preconditions:
1. It is the parent's action.
2. The child has used an application for a longer period of time.
3.The history about the playing is recorded in the audit
1.2 Main Flow:
When the child has used an application, it has called for the action of the parent. The parent looks for the real nature of application i.e., addicting, vulgar, violent natured to benign, fun-filled. Depending upon that the parent takes some action like blocking it (Illicit application) or merely password-protecting it (Fun-filled).
1.3 Subflows:
[S1] In case of illicit application like a violent, vulgar game, this sub-flow is carried out
When the Block-Application button is clicked, the Application Blocker dialog shows up. The parent just selects the type, name, reason for blocking , starting time etc., for the application, that needs to be blocked from that dialog. After clicking on OK in the dialog box, the application is blocked. The child cannot access the same application anymore from his mobile.
[S2] In case of illicit application like a benign, just a fun-filled one, this sub-flow is carried out
When the Block-Application button is clicked, the Application Blocker dialog shows up. The parent just selects the type, name, time for allowing the application, the password to lock the application, the maximum duration for using this application and reason for doing so, starting time etc., for the application that needs to be password-protected from that dialog. After clicking on OK in the dialog box, the application becomes protected one. The child can access the same application only during the specified hours of the day for the specified time duration and on producing the password only, which he has to get from his parent.
1.4 Alternative Flows:
[E1] Nothing happens if the application is a normal, genuine one.

2. **Actor:** Defines the role played by a person, system, hardware component or device that has a stake in performing the operation of the system successfully.
3. **Use Case:** Defines a key functionality/feature of the system. The system will not meet out the user/actor requirements without these features. Each use case denotes a goal that the system must execute.

4. **Relationships:** Defines an interaction that is possible between the actors and use cases. A set of related scenarios is produced from these associations, which operate as test cases when evaluating the analysis, design, and implementation of the use case.
   a. **Association:** Defines the way be which the actors and the use cases are communicating.
   b. **Dependency:** Defines a communication relationship that is possible between two use cases.
      i. Extends relationship
      ii. Include relationship
   c. **Generalization:** Defines a relationship that is possible between two actors or two use cases in the system where one use case inherits/derives and adds to or overrides the properties of the other use case.
      i. Actor generalization
      ii. Use case generalization

PURPOSE OF USE CASE DIAGRAM

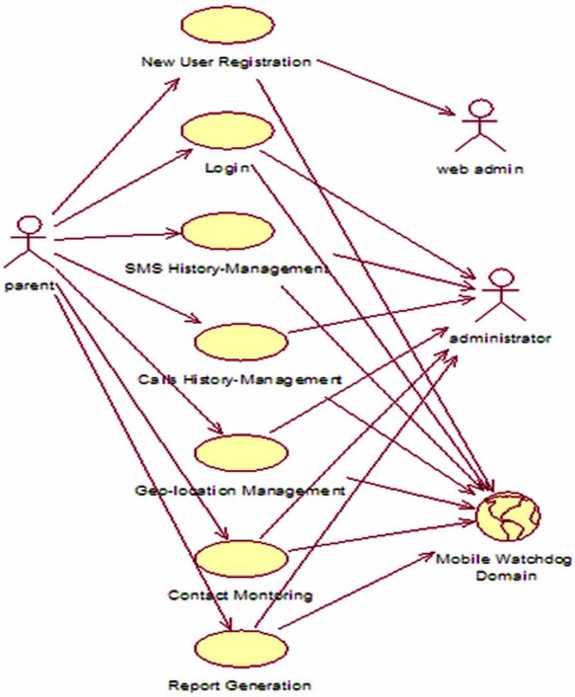
Primarily a *use case* characterizes a goal-oriented set of interactions occurring between external actors and the system under consideration. *Actors* are the parties that are outside the system which interact with the system An actor could be a class of users, the roles users undertakes, or even other systems. A *primary* actor is the one having a goal requiring the assistance of the system. On the other hand a *secondary* actor is one from which the system requires assistance. A use case is triggered by a user with a particular goal in mind, and is completed successfully when that goal is met. It narrates the sequence of interactions between actors and the system in order to deliver the required service that satisfies the goal. It also includes the possible variants of this sequence, e.g., any alternative sequences that could also satisfy the same goal, as well as sequences that may result in failure to complete the service because of exceptional behavior, error handling, etc. The system is treated as a "black box", and the interactions with system, including system responses, are as perceived from outside the system. Thus, use cases confine *who* (actor) does *what* (interaction) with the system, for what *purpose* (goal), without dealing with system internals. A complete set of use cases identifies all the different possible ways to use the system, and therefore defines all behavior required of the system, bounding the scope of the system. Generally, use case steps are written in an easy-to-understand structured narrative using the vocabulary of the domain. This is engaging for users who can easily follow and validate the use cases, and the accessibility encourages users to be actively involved in defining the requirements.

Summarily Use case diagrams are used for

- Documenting the existing process (As-Is)
- Analyzing the new process concepts (To-Be)
- Identifying IT levers if any
- Generating functional Test Cases
- Finding out re-engineering opportunities
- Identifying the boundaries of the system by making a semantic network diagram
- Changing and extending the actor's or user's functionality
- Addressing the non-functional requirements for some scenarios

Use cases bear the project controlling in iterative approach of systems developed or configuration by capturing and documenting visually the features of the system to be developed. Perhaps use cases can't be used to illustrate the inside small processes in modeling.

Figure 4. A sample use case diagram for a real time scenario



Functional Testing using Use Case Diagrams

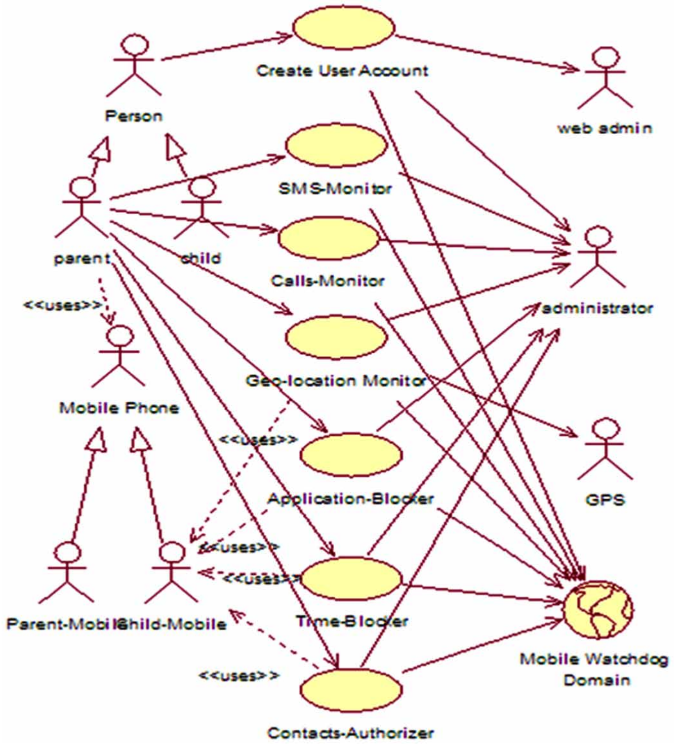
As the use case diagrams are showing the functionality of the software, they have been extensively applied for generating test cases. Generally, a test case is a pair showing the test data and the expected output.

In the case of use case diagrams, the test scenarios are collected to generate the test cases. As an example, consider the Use Case diagram shown in Figure 5.

From figure 5, the sample test scenarios derived are:

1. To access the application, a person has to create an user account through the website which is controlled by the web admin
2. Parent and child are the users of the system
3. Parent can use SMS Monitor function to monitor the functionality of a child.
4. Parent can use Calls Monitor function to monitor the functionality of a child.
5. Parent can use Application Blocker function to monitor the functionality of a child.

Figure 5. Use case diagram for mobile monitoring system from a parent perspective



6. Parent can use Time Blocker function to monitor the functionality of a child.
7. Parent can use Contacts Authorizer function to monitor the functionality of a child.

Test cases are now generated from the above scenarios using the flow of events given in the Use Case description with normal and alternative flow of events as given in Figure 3.

Let us consider the preconditions to generate test cases:

UC5 Flow of Events for the Application Blocker Use Case

Preconditions

1. It is the parent's action.
2. The child has used an application for a longer period of time.
3. The history about the playing is recorded in the audit

Table 1. Test cases generation from flow of events from uml use case diagram

Test Case #	Test Cases	
	Test Data	Expected Output
TC1	Type of user = 'Parent', Login id='Valid Id", Password="Valid Pwd"	"Welcome Message"
TC2	Type of user="Child", Period of usage>=1 hr	"Application should be blocked"
TC3	Type of user="Parent", View History=True	Display Audit Log

Table 2. Test cases generation from main flow of events from UML use case diagram

Test Case #	Test Cases	
	Test Data	Expected Output
TC4	Type of user = 'Parent', Enable-Application Blocker = True	"Application Blocker operation enabled"
TC5	Type of user ='Parent', Application = "Illicit Type", Block-Application=True	"Application blocked"
TC6	Type of user="Parent", Application = "Fun-Filled", Password-Protection=True	"Application Password Protected"

Main Flow

When the child has used an application, it has called for the action of the parent. The parent looks for the real nature of application i.e., addicting, vulgar, violent natured to benign, fun-filled. Depending upon that the parent takes some action like blocking it (Illicit application) or merely password-protecting it (Fun-filled).

Subflows

[S1] In case of illicit application like a violent, vulgar game, this sub-flow is carried out

When the Block-Application button is clicked, the Application Blocker dialog shows up. The parent just selects the type, name, reason for blocking, starting time etc., for the application, that needs to be blocked from that dialog. After clicking on OK in the dialog box, the application is blocked. The child cannot access the same application anymore from his mobile.

[S2] In case of illicit application like a benign, just a fun-filled one, this sub-flow is carried out

When the Block-Application button is clicked, the Application Blocker dialog shows up. The parent just selects the type, name, time for allowing the application, the password to lock the application, the maximum duration for using this application and reason for doing so, starting time etc., for the application that needs to be password-protected from that dialog. After clicking on OK in the dialog box, the application becomes protected one. The child can access the same application only during the specified hours of the day for the specified time duration and on producing the password only, which he has to get from his parent.

Based on the above sub-flows, the test cases are generated as shown in Table 3.

Alternative Flows

[E1] Nothing happens if the application is a normal, genuine one.

For this alternative flow also, a test case is generated as shown in Table 4.

In the above sample, one can understand how a use case diagram and its description helps in generating test cases for a single use case scenario of “Blocking an Application in a child’s mobile by a parent”.

The above sample is a simple IoT solution to monitor a Child’s mobile from any moral misbehavior and from any social threats. In this application, a Mobile device, a GPS sensor in the mobile device and a web application are integrated to produce

Table 3. Test cases generation from sub- flow of events from UML use case diagram

Test Case #	Test Cases	
	Test Data	Expected Output
TC7	Type of user = ‘Parent’, Button-Click=”Block-Application”	Display Application Blocker Dialog window
TC8	Type of user =’Parent’, Dialog Window=”Application Blocker”, Application=”Illicit Type”, Selection=True, Button Click=”OK”	“Application blocked from Child Mobile”
TC9	Type of user=”Parent”, Dialog Window=”Application Blocker”, Application = “Fun-Filled”, Password-Protection=True, Button Click=”OK”	“Allow Application as Password Protected and can be used only for the specified time duration”

Table 4. Test cases generation from alternative flow of events from UML use case diagram

Test Case #	Test Cases	
	Test Data	Expected Output
TC10	Type of user = ‘Parent’, Application Type =”Normal”	“Application is not blocked”

it as a complete IoT solution. In the similar manner, the test cases are generated for any of the IoT applications designed to achieve a particular purpose.

CONCLUSION

The IoT systems have a lot of challenges in testing process. Especially, when it is coming on to the functionality based testing, the way in which the various components such as users, software components and hardware components to interact with each other to produce the required functionality is hard to characterize by means of a textual documentation. In order to provide a complete view of the system in terms of its functionalities, the UML provides diagrammatic representations. One of the most important such diagram is a Use Case diagram which shows the functionalities in the system and the interaction between the components and the functionalities.

Hence, this chapter has provided a complete idea on how to represent any IoT based application using a UML Use Case diagram and how to generate test cases from this diagrammatic representation and the corresponding Use Case specifications. This will leverage the possible test cases to test the system once its development is completed.

REFERENCES

Cognizant IoT Quality Assurance. (2017). CTS, Whitepaper, 2017. Retrieved from https://seeing-things-differently.cognizant.com/pdf/IoT_Testing_Quality_Assurance.pdf

Key Considerations in Testing Internet of Things (IOT) Applications. (2018). CapeGemini, White paper, 2018, Retrieved from https://www.capgemini.com/2016/10/key-considerations-in-testing-internet-of-things-iot-applications/

Kukkuru, M. G. (2017). *Testing IoT Applications: A perspective*. InfoSys whitepaper, 2017. Retrieved from https://www.infosys.com/IT-services/validation-solutions/Documents/testing-iot-applications.pdf

Lee, I., & Lee, K. (2015). The Internet of Things (IoT): Applications, investments, and challenges for enterprises. *Business Horizons*, *58*(4), 431–440. doi:10.1016/j.bushor.2015.03.008

Rosborough, B. (2018). *Security Testing of Internet of Things: Dynamic Testing (Fuzzing) for IoT Security*. Beyond Security, White Paper, 2018. Retrieved from https://www.beyondsecurity.com/blog/security-testing-the-internet-of-things-iot

Sand, B. (2018). *IoT Testing – The Big Challenge Why, What & How*. QUALITEST, White paper, 2018. Retrieved from https://www.qualitestgroup.com/white-papers/iot-testing-the-big-challenge/

Shanmugasundaram, R. (2015). *IoT Basics and Testing Focus*. HCL White paper, 2015. Retrieved from https://theinternetofthings.report/Resources/Whitepapers/793406e1-2095-40a5-9369-70d3df83e844_iot_basics_and_testing_focus.pdf

Sypolt, G. (2016). *Functional Testing And The IoT*. Sauce Labs, White paper, 2016. Retrieved from https://saucelabs.com/blog/functional-testing-and-the-iot

Zalavadiam, S. (2017). *The future of IoT testing*. Zephyr, Whitepaper, 2017. Retrieved from https://www.getzephyr.com/insights/future-iot-testing

Chapter 7

Best Practices:
Adopting Security Into the Software Development Process for IoT Applications

Anchitaalagammai J. V.
Velammal College of Engineering and Technology, India

Murali S.
Velammal College of Engineering and Technology, India

Kavitha Samayadurai
Velammal College of Engineering and Technology, India

Padmadevi S.
Velammal College of Engineering and Technology, India

Shantha Lakshmi Revathy J.
Velammal College of Engineering and Technology, India

ABSTRACT

Internet of things (IoT) describes an emerging trend where a large number of embedded devices (things) are connected to the internet to participate in automating activities that create compounded value for the end consumers as well as for the enterprises. One of the greatest concerns in IoT is security, and how software engineers address it will play a deeper role. As devices interact with each other, businesses need to be able to securely handle the data deluge. With focused approach, it is possible to minimize the vulnerabilities and risks exposed to the devices and networks. Adopting security-induced software development lifecycle (SDL) is one of the major steps in identifying and minimizing the zero-day vulnerabilities and hence to secure the IoT applications and devices. This chapter focuses best practices for adopting security into the software development process with the help of two approaches: cryptographic and machine learning techniques to integrate secure coding and security testing ingrained as part of software development lifecycle.

DOI: 10.4018/978-1-5225-7790-4.ch007

INTRODUCTION

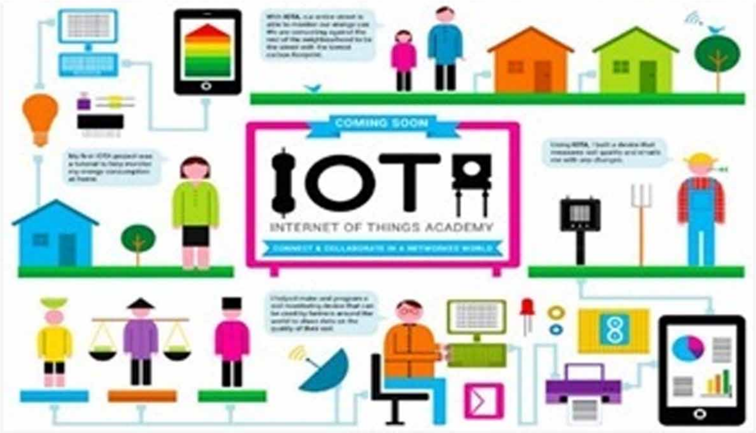
The Internet of Things (IoT) is a complex paradigm where billions of devices are connected over a network. These connected devices form an intelligent system of systems that share the data without human-to-computer or human-to-human interaction. These systems extract meaningful data that can transform human lives, businesses and the world in significant ways. Figure 1 shows the conception of IoT.

However, the reality of IoT is prone to countless cyber-attacks in the extremely hostile environment like internet. To secure an IoT system, the traditional high-end security solutions are not suitable, as IoT devices are of low storage capacity and less processing power. Moreover, the IoT devices are connected for longer time periods without human intervention. This raises a need to develop smart security solutions which are light-weighted, distributed and have high longevity of service. Rather than per-device security for numerous IoT devices, it is more feasible to implement security solutions for network data. In this chapter,we have applied the concept of Cryptographic and Machine learning approach in order to build a light-weight distributed security solution with high durability for IoT network security.

THE IOT ARCHITECTURE

1. **Perception Layer:** the main task of the perception layer is to perceive the physical properties of things around us that are part of the IoT. This process of perception is based on several sensing technologies (e.g. RFID, WSN, GPS,

Figure 1. Conception of IoT


COMING SOON
IOT
INTERNET OF THINGS ACADEMY
CONNECT & COLLABORATE IN A NETWORKED WORLD

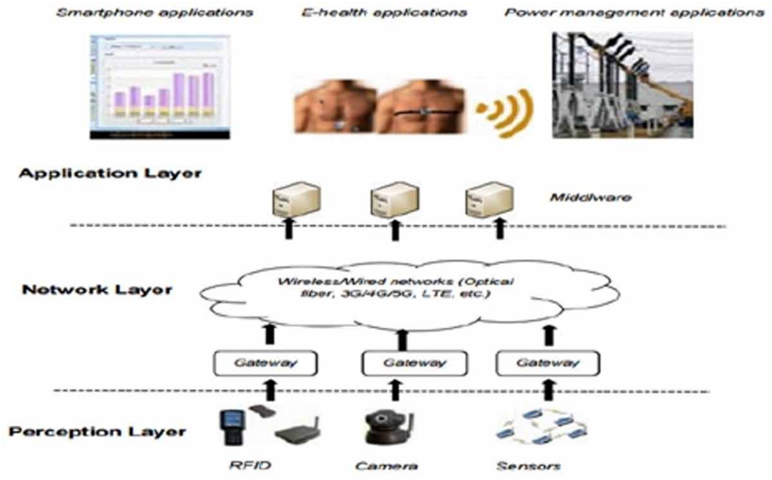
NFC, etc.). In addition, this layer is in charge of converting the information to digital signals, which are more convenient for network transmission. However, some objects might not be perceived directly. Thus, microchips will be appended to these objects to enhance them with sensing and even processing capabilities. Indeed, nanotechnologies and embedded intelligence will play a key role in the perception layer. The first one will make chips small enough to be implanted into the objects used in our everyday life. The second one will enhance them with processing capabilities that are required by any future applications.

2. **Network Layer:** the network layer is responsible for processing the received data from the Perception Layer. In addition, it is in charge of transmitting data to the application layer through various network technologies, such as wireless / wired networks and Local Area Networks (LAN). The main media for transmission include FTTx, 3G / 4G, Wifi, bluetooth, Zigbee, UMB, infrared technology, and so on. Huge quantities of data will be carried by the network. Hence, it is crucial to provide a sound middleware to store and process this massive amount of data. To reach this goal, cloud computing is the primary technology in this layer. This technology offers a reliable and dynamic interface through which data could be stored and processed. Indeed, research and development on the processing part is significant for the future development of IoT.
3. **Application Layer:** the application layer uses the processed data by the previous Layer. In fact, this layer constitutes the front end of the whole IoT architecture through which IoT potential will be exploited. Moreover, this layer provides the required tools (e.g. actuating devices) for developpers to realize the IoT vision. In this vision, the range of possible applications is impressive (e.g. Intelligent transportation, logistics management, identity authentication, location based services, safety, etc.). Figure 2 shows the Layered IoT Architecture.

SECURITY REQUIREMENTS

The basic security issues of IoT require the identity authentication mechanisms and protection of the confidentiality of the data. The three basic areas are data confidentiality, data integrity and data availability. Breaching any one of these three basic security areas may cause security damages to the IoT system. Thus, each of the four layers of the IoT network system should meets these minimum requirements. Figure 3 shows the basic security requirements for the IoT.

Figure 2. Layered IoT architecture



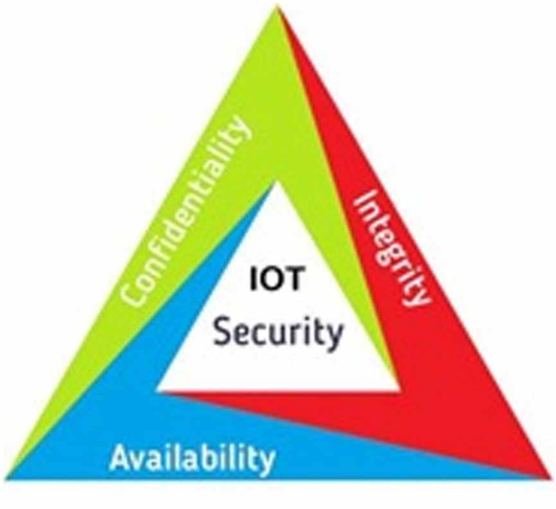
Data Confidentiality

The goal of Data confidentiality is protecting the privacy of sensitive information by using some mechanisms and preventing the unauthorized access (Miorandi et al, 2012). For the IoT devices such as the sensors and nodes, data confidentiality means the data collected by the sensors and nodes should not be transmitted to an unauthorized party. Data encryption is a mechanism to ensure the data confidentiality. The encrypted data convert into cipher text; thus, unauthorized users cannot easily access the data. The two-step verification is another method to ensure the data confidentiality. In this method, the user can only access to the data by passing two dependent authentication tests.

Data Integrity

Data Integrity protects the useful information from tempering by the cybercriminals, during the communication. There are varieties of cases such as crash of servers or power disturbance that may affect the data integrity. One method to ensure the data integrity at the first-level is the cyclic redundancy check(CRC): CRC is a simple error detector mechanism to encode the message by adding a fixed-length check value for the error detection in IoT communication networks, the data integrity can be ensured by checking the check value (Atzori et al, 2010). Other method like Version control can syncing and backup the data to keep the file changes in the system, thus ensure the data integrity by restore the changing data in case of deletion or lost.

Figure 3. Basic Security Requirement for the IoT



Data Availability

Data availability is very important to the security of IoT, data availability ensure the users can access to the information resources in both normal and disastrous situations, and data availability also ensure the consequent flowing of the information. To guarantee the data availability and reliability, IoT system needs the backup and redundant techniques to provide the duplication of the important information and prevent the data lose in system failure or system confliction conditions. Denial-of-service (DoS) and distributed-denial- of-services (DDoS) attacks cause the security issues of data availability, router filtering can countermeasure the issue and ensure the data availability of the IoT system.

RELATED WORKS

The authors of (Groswami et al, 2014) propose an edge router to take the responsibility of being higher in computation power, maintaining the Key database and communicating with the server for the Certificate Authority (CA) over the IPv6 network. But the implementation and performance evaluation showed that through this scheme security was achieved but at the cost of time and packet count. IPSec provides security for the IoT enabled devices, by assuring them authentication and privacy in terms of encryption. Shows an implementation and evaluation of IPSec over 6LoWPAN and provide with critical conclusions that it is possible to secure the end-to end (E2E) communication between a sensor node in WSN and an IPv6 enabled node (Raza et al, 2011). In (Raza et al, 2012) the DTLS is compressed and integrated into the 6LoWPAN. It is found that this has a direct impact on the security bits, as they have found to be reduced by 62%. The Public Key Infrastructure (PKI) supported by conventional WSNs cannot be directly integrated into the IoT and the 6LoWPAN. In (Yue et al, 2015) the author proposes a Symmetric Key Cryptographic scheme, the EAKES6Lo that operates at the 6LoWPAN layer for a sensor node enabled IoT. This scheme was successful in preventing some of the main attacks such as the replay attack, impersonation attack, compromised key attack etc

CRYPTOGRAPHIC APPROACHES

Element layer consists of different nodes and sensors to collect the data from connected network environment. The nodes and sensors are exposed to different threats such as unauthorized access, eavesdropping, spoofing, etc. The service layer is processing the data and providing the links to the storage for the collected data from the element layer. The service layer security should prevent attacks like DoS attack, unauthorized access and malicious insider. The application layer consists of variety applications of IoT. The application layer security concerns such as DDoS attack, malicious code injection attack and phishing attack need to be addressed. Table 1 shows the security services as well as mechanisms at each layer.

Table 1. Security services and mechanisms at each layer

Layers	Threats Attacks	Security Services	Security Mechanisms
Perception	• Unauthorized Access • Eavesdropping • Spoofing	• Authentication • Access Control • Confidentiality	• Digital Signature • Access Control Table • PKI*
Network	• Denial of Services • Man-in-the-middle • Malicious Code Injection	• Availability • Integrity • Anti-virus	• Router Filtering • Data Encryption • Anti-virus
Application	• DDoS • Malicious Code Injection • Phishing	• Availability • Anti-virus • Anti-phishing	• IDS* • Anti-virus • Spam Filtering

*PKI: A framework provides data privacy in communications by using encryption and authentication.

*IDS: A security system to monitor the traffic of the network to detect the DoS attacks.

IEEE 802.15.4 at Perception Layer

IEEE 802.15.4 is a standard that specifies the physical layer (PHY) and the media access control (MAC) communication for the low-speed low-cost communication between devices in wireless personal area networks. IEEE 802.15.4 implements the advanced encryption standard (AES) symmetric cryptography mechanism and support several security modes; (Rahman et al, 2016) these security modes provide security services such as confidentiality, authentication, and integrity. Table 2 shows the security services and the security modes of the IEEE 802.15.4.

MACHINE LEARNING APPROACHES

With the development of machine learning (ML) and smart attacks, IoT devices have to choose the defense policy and determine the key parameters in the security protocols for the tradeoff in the heterogeneous and dynamic networks. This task is challenging as an IoT device with restricted resources usually has difficulty accurately estimating the current network and attack state in time.

For example, the authentication performance of the scheme in (Xiao et al, 2016) is sensitive to the test threshold in the hypothesis test, which depends on both the radio propagation model and the spoofing model. Such information is unavailable for most outdoor sensors, leading to a high false alarm rate or miss detection rate in the spoofing detection.

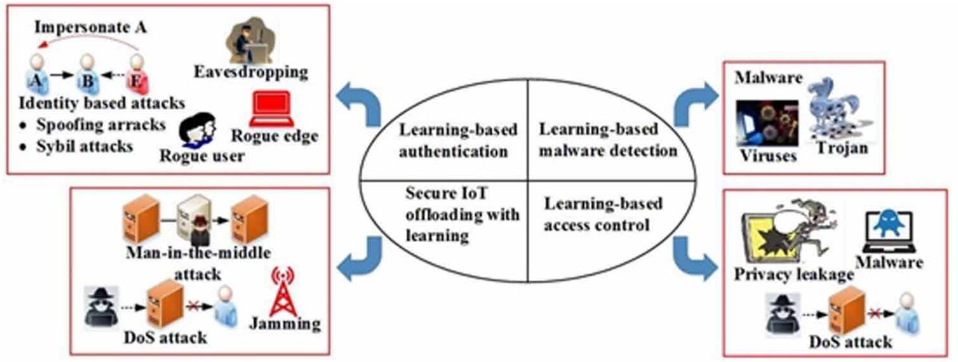
Table 2. Security services and the security modes of the IEEE 802.15.4

Security Service	Security Mode	Security Mechanism
Confidentiality	AES-CTR	Data is encrypted using AES in the counter mode with 128-bit keys
Authentication Integrity	AES-CBC-MAC/MIC-32	Data is encrypted using AES in the cypher block chaining mode with message authentication code and message integrity code in 32/64/128-bit keys
	AES-CBC-MAC/MIC-64	
	AES-CBC-MAC/MIC-128	
Confidentiality Authentication Integrity	AES-CCM-32	Data is encrypted using AES in the counter and cypher block chaining mode with message authentication code and message integrity code in 32/64/128-bit keys
	AES-CCM-64	
	AES-CCM-128	

Machine learning techniques including supervised learning, unsupervised learning, and reinforcement learning (RL) have been widely applied to improve network security, such as authentication, access control, anti-jamming offloading and malware detections.

- **Supervised learning techniques** such as support vector machine (SVM), naive Bayes, K- nearest neighbor (K-NN), neural network, deep neural network (DNN) and random forest can be used to label the network traffic or app traces of IoT devices to build the classification or regression model. For example, IoT devices can use SVM to detect network intrusion and spoofing attacks, apply K-NN in the network intrusion and malware detections, and utilize neural network to detect network intrusion and DoS attacks (Branch et al, 2013; Buczak et al, 2015; Kulkarni et al, 2009). Naive Bayes can be applied by IoT devices in the intrusion detection and random forest classifier can be used to detect malwares (Narudin et al, 2016). IoT devices with sufficient computation and memory resources can utilize DNN to detect spoofing attacks (Alsheik et al, 2014).
- **Unsupervised learning** does not require labeled data in the supervised learning and investigates the similarity between the unlabeled data to cluster them into different groups (Kulkarni et al, 2009). For example, IoT devices can use multivariate correlation analysis to detect DoS attacks (Tan et al, 2013) and apply IGMM in the PHY-layer authentication with privacy protection. The machine learning based IoT authentication, access control, secure offloading techniques, and malware detections. Figure 4 shows the threat model in Internet of Things

Figure 4. Illustration of the threat model in Internet of Things



LEARNING-BASED AUTHENTICATION

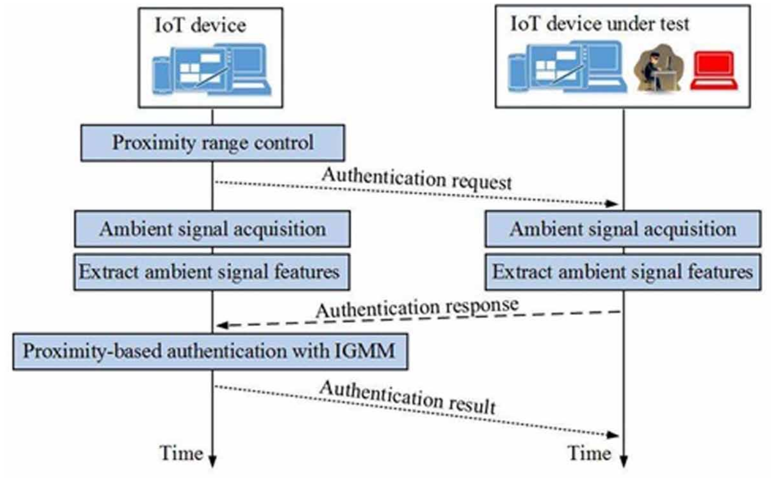
Traditional authentication schemes are not always applicable to IoT devices with limited computation, battery and memory resources to detect identity-based attacks such as spoofing and Sybil attacks. Physical (PHY)-layer authentication techniques that exploit the spatial decorrelation of the PHY-layer features of radio channels and transmitters such as the received signal strength indicators (RSSIs), received signal strength (RSS), the channel impulse responses (CIRs) of the radio channels, the channel state information (CSI), the MAC address can provide light-weight security protection for IoT devices with low computation and communication overhead without leaking user privacy information . PHY-layer authentication methods such as build hypothesis tests to compare the PHY-layer feature of the message under test with the record of the claimed transmitter. Their authentication accuracy depends on the test threshold in the hypothesis test. However, it is challenging for an IoT device to choose an appropriate test threshold of the authentication due to radio environment and the unknown spoofing model. As the IoT authentication game can be viewed as a Markov decision process (MDP), IoT devices can apply RL techniques to determine the key authentication parameters such as the test threshold without being aware of the network model.

As shown in Figure 5. This scheme requests the IoT device under test to send the ambient signals features such as the RSSIs, MAC addresses and packet arrival time internal of the ambient signals received during a specific time duration. The IoT device extracts and sends the ambient signals features to the legal receiver. Upon receiving such authentication messages, the receiver applies IGMM to compare the reported signal features with those of the ambient signals observed by itself in the proximity based test. The receiver provides the IoT device passing the authentication with access to the IoT resources.

LEARNING-BASED ACCESS CONTROL

IoT devices such as sensors outdoor usually have strict resource and computation constraints yielding challenges for anomaly intrusion detection techniques usually have degraded detection performance in IoT system. ML techniques help build light-weight access control protocols to save energy and extend the lifetime of IoT systems. For example, the outlier detection scheme as developed in Branch (2013) applies K-NN to address the problem of unsupervised outlier detection in WSNs and offers flexibility to define outliers with reduced energy consumption. This scheme can save the maximum energy by 61.4% compared with the Centralized scheme with similar average energy consumption.

Figure 5. Illustration of the ML-based authentication in IoT systems


IoT device
IoT device under test
Proximity range control
Authentication request
Ambient signal acquisition
Ambient signal acquisition
Extract ambient signal features
Extract ambient signal features
Authentication response
Proximity-based authentication with IGMM
Authentication result
Time
Time

The multilayer perceptron (MLP) based access control as presented in Kuklarni (2009) utilizes the neural network with two neurons in the hidden layer to train the connection weights of the MLP and to compute the suspicion factor that indicates whether an IoT device is the victim of DoS attacks. This scheme utilizes backpropagation (BP) that applies the forward computation and error backpropagation and particle swarm optimization (PSO) as an evolutionary computation technique that utilizes particles with adjustable velocities to update the connection weights of the MLP. The IoT device under test shuts down the MAC layer and PHY-layer functions to save energy and extend the network life, if the output of the MLP exceeds a threshold.

SECURE IOT OFFLOADING WITH LEARNING

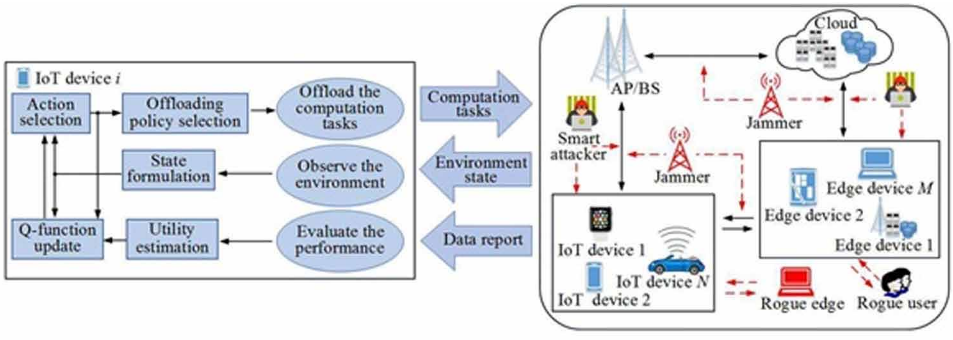
IoT offloading has to address the attacks launched from the PHY-layer or MAC layer attacks, such as jamming, rogue edge devices, rouge IoT devices, eavesdropping, man-in-the-middle attacks and smart attacks (Roman et al, 2018). As the future state observed by a IoT device is independent of the previous states and actions for a given state and offloading strategy in the current time slot, the mobile offloading strategy chosen by the IoT device in the repeated game with jammers and interference sources can be viewed as a MDP with finite states (Xiao et al, 2016). RL techniques can be used to optimize the offloading policy in dynamic radio environments. Q-learning, as a model-free RL technique, is convenient to implement with low computation complexity. For example, IoT devices can utilize the Q-learning based offloading as proposed in Xiao (2016) to choose their offloading data rates against jamming and spoofing attacks. As illustrated in Figure 6, the IoT device observes the task importance, the received jamming power, the radio channel bandwidth, the channel gain to formulate its current state, which is the basis to choose the offloading policy according to the Q-function. The Q-function, which is the expected discounted long-term reward for each action-state pair and represents the knowledge obtained from the previous anti-jamming offloading. The Q-values are updated via the iterative Bellman equation in each time slot according to the current offloading policy, the network state and the utility received by the IoT device against jamming.

The IoT device evaluates the signal-to-interference-plus-noise ratio (SINR) of the received signals, the secrecy capacity, the offloading latency the and energy consumption of the offloading process and estimates the utility in this time slot. The IoT device applies the ϙ-greedy algorithm to choose the offloading policy that maximizes its current Q-function with a high probability and the other policies with a small probability, and thus makes a tradeoff between the exploration and the exploitation. This scheme reduces the spoofing rate by 50%, and decreases the jamming rate by 8%, compared with a benchmark strategy as presented in (Xaio et al, 2016).

LEARNING-BASED IOT MALWARE DETECTION

IoT devices can apply supervised learning techniques to evaluate the runtime behaviors of the apps in the malware detection. In the malware detection scheme as developed in an IoT device uses K-NN and random forest classifiers to build the malware detection model. As illustrated in Figure. 7, the IoT device filters the TCP packets and selects the features among various network features including the frame

Figure 6. Illustration of the ML-based offloading



number and length, labels them and stores these features in the database. The K-NN based malware detection assigns the network traffic to the class with the largest number of objects among its K nearest neighbors. The random forest classifier builds the decision trees with the labeled network traffic to distinguish malwares. According to the experiments in Narudin (2016), the true positive rate of the K-NN based malware detection and random forest based scheme with MalGenome dataset are 99.7% and 99.9%, respectively.

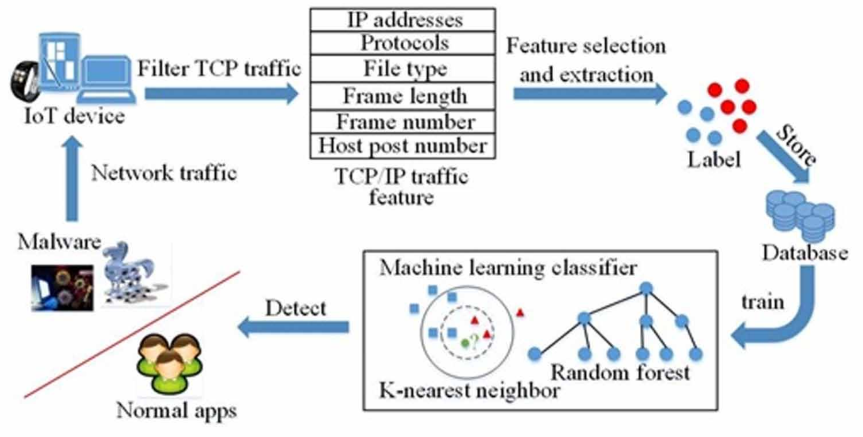
CONCLUSION

In this article, we have identified the IoT attack models and suggested the IoT security based on Cryptographic techniques and Machine Learning techniques for secure software development. In general, a number of services and network providers are involved in implementation of the IoT system. The optimum implementation of the Secured IoT with different network platforms need to be investigated as future work in this area.

ACKNOWLEDGMENT

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Figure 7. Illustration of the ML-based malware detection



REFERENCES

Abu Alsheikh, M., Lin, S., Niyato, D., & Tan, H. P. (2014). Machine learning in wireless sensor networks: Algorithms, strategies, and applications. *IEEE Communications Surveys and Tutorials*, *16*(4), 1996–2018. doi:10.1109/COMST.2014.2320099

Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A Survey. *Computer Networks*, *54*(15), 2787–2805. doi:10.1016/j.comnet.2010.05.010

Branch, J. W., Giannella, C., Szymanski, B., Wolff, R., & Kargupta, H. (2013). In-network outlier detection in wireless sensor networks. *Knowledge and Information Systems*, *34*(1), 23–54. doi:10.100710115-011-0474-5

Buczak, L., & Guven, E. (2015). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys and Tutorials*, *18*(2), 1153–1176. doi:10.1109/COMST.2015.2494502

Goswami, S., Misra, S., Taneja, C., & Mukherjee, A. (2014). Securing intra-communication in 6LoWPAN: A PKI integrated scheme. *2014 IEEE International Conference on Advanced Networks and Telecommuncations Systems (ANTS)*, 1–5. 10.1109/ANTS.2014.7057265

Kulkarni, R. V., & Venayagamoorthy, G. K. (2009). Neural network based secure media access control protocol for wireless sensor networks. *Proc. Int'l Joint Conf. Neural Networks*, 3437–3444. 10.1109/IJCNN.2009.5179075

Miorandi, D., Sicari, S., De Pellegrini, F., & Chlamtac, I. (2012). Internet of Things: Vision, applications and research challenges. *Ad Hoc Networks*, *10*(7), 1497–1516. doi:10.1016/j.adhoc.2012.02.016

Narudin, F. A., Feizollah, A., Anuar, N. B., & Gani, A. (2016). Evaluation of machine learning classifiers for mobile malware detection. *Soft Computing*, *20*(1), 343–357. doi:10.100700500-014-1511-6

Ozay, M., Esnaola, I., Yarman Vural, F. T., Kulkarni, S. R., & Poor, H. V. (2015). Machine learning methods for attack detection in the smart grid. *IEEE Transactions on Neural Networks and Learning Systems*, *27*(8), 1773–1786. doi:10.1109/TNNLS.2015.2404803 PMID:25807571

Rahman & Shah. (2016). Security analysis of IoT protocols: A focus in CoAP. *MEC International Conference on Big Data and Smart City*, 1-7.

Raza, S., Duquennoy, S., Chung, T., Yazar, D., Voigt, T., & Roedig, U. (2011). Securing communication in 6LoWPAN with compressed IPsec. *2011 International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS)*, 1–8. 10.1109/DCOSS.2011.5982177

Raza, S., Trabalza, D., & Voigt, T. (2012). 6LoWPAN Compressed DTLS for CoAP. *2012 IEEE 8th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 287–9.

Roman, R., Lopez, J., & Mambo, M. (2018). A survey and analysis of security threats and challenges. *Future Generation Computer Systems*, *78*(3), 680–698. doi:10.1016/j.future.2016.11.009

Tan, Z., Jamdagni, A., He, X., Nanda, P., & Liu, R. P. (2013). A system for Denial-of-Service attack detection based on multivariate correlation analysis. *IEEE Transactions on Parallel and Distributed Systems*, *25*(2), 447–456.

Xiao, L., Li, Y., Han, G., Liu, G., & Zhuang, W. (2016). PHY-layer spoofing detection with reinforcement learning in wireless networks. *IEEE Transactions on Vehicular Technology*, *65*(12), 10037–10047. doi:10.1109/TVT.2016.2524258

Xiao, L., Xie, C., Chen, T., Dai, H., & Poor, H. V. (2016). A mobile offloading game against smart attacks. *IEEE Access: Practical Innovations, Open Solutions*, *4*, 2281–2291. doi:10.1109/ACCESS.2016.2565198

Yue, Q., & Maode, M. (2015). An authentication and key establishment scheme to enhance security for M2M in 6LoWPANs. *2015 IEEE International Conference on Communication Workshop (ICCW)*, 2671–6

Chapter 8
Machine Learning Techniques for Internet of Things

P. Priakanth
Kongu Engineering College, India

S. Gopikrishnan
Karpagam College of Engineering, India

ABSTRACT

The idea of an intelligent, independent learning machine has fascinated humans for decades. The philosophy behind machine learning is to automate the creation of analytical models in order to enable algorithms to learn continuously with the help of available data. Since IoT will be among the major sources of new data, data science will make a great contribution to make IoT applications more intelligent. Machine learning can be applied in cases where the desired outcome is known (guided learning) or the data is not known beforehand (unguided learning) or the learning is the result of interaction between a model and the environment (reinforcement learning). This chapter answers the questions: How could machine learning algorithms be applied to IoT smart data? What is the taxonomy of machine learning algorithms that can be adopted in IoT? And what are IoT data characteristics in real-world which requires data analytics?

DOI: 10.4018/978-1-5225-7790-4.ch008

INTRODUCTION

The two major trends are going on world today among the computing research community which one is the rapid rise of IoT and the other one is the rapid rise of AI and machine learning. In many ways they are intersecting with each other to create some really smart new scenarios. One such an IoT AI smart scenario has been discussed with demonstration here.

Consider a scenario to learn how a cheap IoT sensor was deployed to measure air quality, how they can push data to the cloud and how we can use that for machine learning. In this scenario the first part is discussing about the design of sensor nodes with IoT capability to measure the air quality. Then it look at how the cloud can be empowering the IOT AI scenario that helps to collect data and analyze it and learning some intelligence with GD plotting models. Then the third one will be discussed about the devices like raspberry PI's and our handheld devices that are handling the device communication and inference with them as well.

According to IHS research they are expected to be 31 billion connected IOT devices by the end of this year. And also this research is showing a 27.6% compound annual growth rate of the data will be pushing to the cloud by those devices. With those lot of data, there is some usual things such as fill, sinks, filtering, sorting and querying all that kind of stuff are done well. But a smart scenario can also use that to start training some models, all of that data and all of those models Gubbi(2013). Hence machine learning models can be used to generate intelligence and to help to start making intelligent decisions based on the data that is being generated by the devices Khan(2012) and Jin(2014).

Why Machine Learning?

The world is filled with lot of data like pictures, music, words, spreadsheets, videos and it does not going to slow down anytime. Soon machine learning brings the promise of deriving meaning from all of that data. The value of machine learning is only just beginning to show itself. There is a lot of data in the world today generated not only by people but also by computers, phones and other devices. This will only continue to grow in the years to come. Traditionally humans have analyzed data and adapted systems to the changes in data patterns. However as the volume of data surpasses the ability for humans to make sense of it and manually write those rules. It turns the world increasingly to automated systems that can learn from the data and importantly the changes in data to adapt, to a shifting landscape.

Machine learning exists all around the world in the daily use products. However it is not always apparent that machine learning is behind it all. While things like tagging objects and people inside of photos are clearly machine learning and recommending the next video to watch is also powered by machine learning. Google search is also one of the biggest examples of machine learning. Every time you use Google search you are using a system that has many machine learning systems at its core. From understanding the text of the search query to adjusting the results based on the personal interests such as knowing which results to show you first. When searching for Java depending on whether it shows coffee or programming language or both. These powerful capabilities can be applied to a wide range of fields from diabetic retinopathy and skin cancer detection to retail, transportation and self-driving vehicles. Now every company is pivoting to use machine learning in their products in some way. It is rapidly becoming well an expected feature just as users expect companies to have a website that works on their mobile device or perhaps an app. From detecting skin cancer to detecting escalators in need of repair machine learning has granted computer systems entirely new abilities.

As machine learning used to make human tasks better faster and easier than before it can also look further into the future. The simple definition of machine learning is “Using data to answer questions”. Using data is referred to as training and answering questions is referred to as making predictions or inference. These two sides refer to using the data to inform the creation and fine-tuning of a predictive model. This predictive model can then be used to serve up predictions on previously unseen data and answer those questions. As more data is gathered, the model can be improved over time and new predictive models deployed. The key component of this entire process is data. Everything hinges on data. Data is the key to unlocking machine learning just as much as machine learning is the key to unlocking that hidden insight in data. In future sections this chapter discusses about what are the popular machine learning algorithms available, what are the processes involved in machine learning and how it can apply to IoT data.

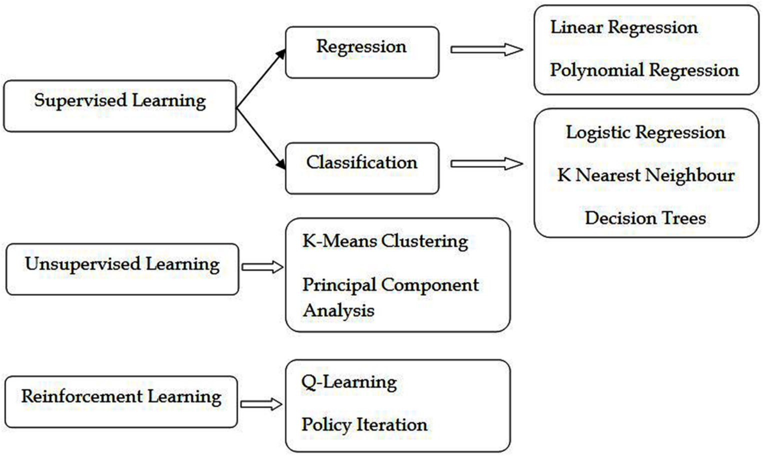
MACHINE LEARNING ALGORITHMS

This section discusses in short about the machine learning algorithms. The machine learning algorithms can be divided in three categories as supervised learning, unsupervised learning and reinforcement learning. Figure 1 shows the categories of Machine Learning algorithms. Supervised learning can be further subdivided in regression and classification. The top regression algorithms are linear regression and polynomial regression. The top classification algorithms are logistic regression, K nearest neighbour and decision trees. The popular unsupervised learning algorithms are k-means clustering and principal component analysis enforcement learning is an advanced topic and it's not recommended at the beginner level.

Supervised learning is the task of inferring a function from the training data. The training data consists of a set of observations together with its outcome. It can be further subdivided in regression and classification algorithms. Regression is used to predict numerical values and classification techniques are used to predict categorical values. Unsupervised learning is a set of algorithms used to draw inferences from data sets consisting of input data without using the outcome. The most common unsupervised learning method is cluster analysis. It is used for exploratory data analysis to find hidden patterns or groupings in data. Some of the important machines learning algorithms are discussed in short here.

- Linear regression models relationship between observation and outcome using a straight line. Root mean squared error and gradient decent is used to fit the best possible line. It provides insights into the factors that have greater influence on the outcome.
- Polynomial regression is a form of regression analysis in which the relationship between the observation and the outcome is modelled as an nth degree polynomial. The method is more reliable when the curve is built on large number of observations.
- Logistic regression is a misleading name. even though the name suggests regression but in reality it is a classification technique. it is used to estimate the probability of a binary response. It can be generalized to predict more than two categorical values also.

Figure 1. Categories of machine learning algorithms



- K nearest neighbours is a classification technique where an object is classified by a majority votes its neighbours. The observation is assigned to the class which is most common among its key nearest neighbours. The best choice of K depends upon the data. Generally larger value of K reduces the effect of noise on the classification.
- Decision tree is a decision support tool that uses a tree like model of decisions and the possible consequences. Decision trees aim to create a model that predicts by learning simple decision rules from the training data.
- K-means clustering k-means clustering aims to partition observations into K clusters. For instance the items in a supermarket are clustered in categories butter, cheese and milk in a group of dairy products. K-means algorithm does not necessarily find the most optimal configuration. The k-means algorithm is usually run multiple times to reduce this effect.
- Principal component analysis is a technique used to emphasize variation and bring out strong patterns in a data set. First principal component has the largest possible variance that is accounts for as much of the variability in the data as possible. Each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components.

MACHINE LEARNING PROCESS

Consider a task to create a system that answers the question of whether a drink is coke or pepsi. The system which answers this question is called a model and this model is created via a process called training machine learning. The goal of the machine training is to create an accurate model that answers the questions correctly most of the time. But in order to train a model the first step is to collect data to train on. The data will be collected from glasses of coke and pepsi for this example. There are many aspects of drinks that could collect data on. But for uncomplicated process pick two simple ones. First the colour as a wavelength of light and the sugar level as a percentage. These two parameters colour and alcohol are known as ML features.

The first step of ML process is gathering the data. This step is very important because the quality and quantity of data that collect will directly determine how good the predictive model can be. In this case the data that collected is colour and sugar level of each drink. This will be the training data. The next step of machine learning is data preparation. The data which are collected in first step have to be loaded into a suitable place and prepare it to use in machine learning training. First process is to put all the data together then randomize the ordering. This is required to do any pertinent visualization of the data to see if there are any relevant relationships between different variables. as well as if there are any data imbalances, for instance if the collected data has more data points about pepsi than coke then the trained model will be heavily biased toward guessing that virtually everything it sees is pepsi.

The next process is to split the data into two parts. The first part used to train the model which will be the majority of the data set. The second part will be used for evaluating the trained models performance. Do not use the same data that the model was trained on for evaluation. The next step in this workflow is choosing a model. There are many models that researchers and data scientists have created over the years as discussed in previous section. Some are very well suited for image data others for sequences such as text or music. some for numerical data and others for text-based data. In this case, data set has two features colour and sugar level percentage. A small model which is a fairly simple one is enough to get the job done.

The formula for a straight line is Y = M * X + B. where X is the input, M is the slope of the line. B is the y-intercept and Y is the value of the line at that position X. the values that available in data set to adjust or train are just M and B. where the M is that slope and B is the y-intercept. There is no other way to ascetics affect the position of the line since the only other variables are X is input and Y is output. In machine learning there are many amps since there may be many features. The collection of these values is usually formed into a matrix that is denoted W for the weights matrix. Similarly for B we arrange them together and that's called the biases

$$\text{Weights} = \begin{bmatrix} m_{1,1} & m_{1,2} \\ m_{2,1} & m_{2,2} \\ m_{3,1} & m_{3,2} \end{bmatrix}$$

$$\text{Biases} = \begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \\ b_{3,1} & b_{3,2} \end{bmatrix}$$

The training process involves initializing some random values for W and b and attempting to predict the outputs with those values. Here it is possible to compare the models predictions with the output that it should have produced and adjust the values in W and B. it's just that produce will have more accurate predictions on the next time around. So this process then repeats each iteration or cycle of updating the weights and biases is called one training step. Figure 2 represent the cycle of training a data set.

Figure 2. Cycle of training a data set

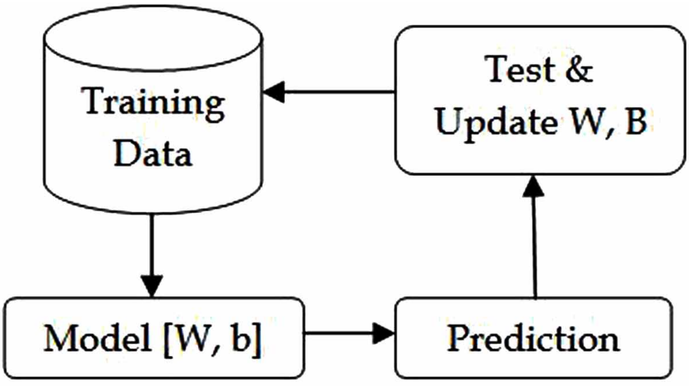
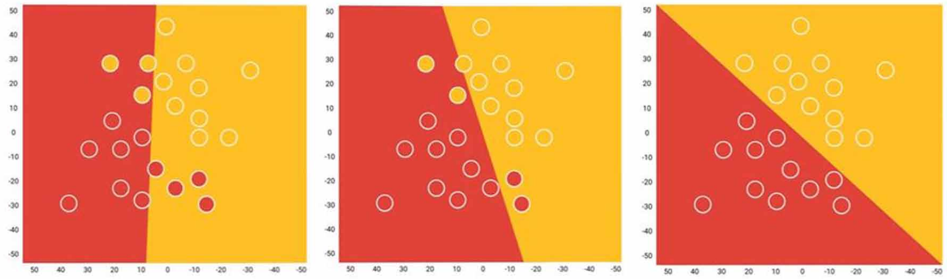


Figure 3 shows what that means more concretely for the actual data set. When start the training it's like drew a random line through the data. Then as each step of the train progresses the line moves step by step closer to the ideal separation of the coke and pepsi.

Once training is complete evaluation allows testing the model against data that has never been used for training. This metric allows seeing how the model might perform against data that has not yet seen. This is meant to be representative of how the model might perform in the real world. A good rule for a training evaluation split is somewhere on the order of 80/20 or 70/30. Much of this depends on the size of the original source data set. Once the training and the evaluation step satisfies the trainer or developer, the final step is to use the model to do something useful. Machine learning is using data to answer questions. So prediction or inference is that step where finally get to answer some questions. This is the point of all of this work where the value of machine learning is realized. Hence finally use this model to predict whether a given drink is coke or pepsi by given its colour and sugar level percentage.

Figure 3. Illustration of data set training



This is a simple example which applies the machine learning technique to predict the output from a data set which is collected. These steps of machine learning processes can be applied to other problem domains as well where the same principles apply gathering data, preparing that data, choosing a model, training it and evaluating it, parameter tuning and finally prediction.

Related Studies With Machine Learning Techniques

Existing machine learning algorithms can be categorized by the intended structure of the model. Most machine learning algorithms fall into the categories of supervised, unsupervised and reinforcement learning Abu(2012). In the first category, machine learning algorithms are provided with a labeled training data set. This set is used to build the system model representing the learned relation between the input, output and system parameters. In contrast to supervised learning, unsupervised learning algorithms are not provided with labels (i.e., there is no output vector). Basically, the goal of an unsupervised learning algorithm is to classify the sample sets to different groups (i.e., clusters) by investigating the similarity between the input samples. The third category includes reinforcement learning algorithms, in which the agent learns by interacting with its environment (i.e., online learning). Finally, some machine learning algorithms do not naturally fit into this classification since they share characteristics of both supervised and unsupervised learning methods. These hybrid algorithms (often termed as semi-supervised learning) aim to inherit the strengths of these main categories, while minimizing their weaknesses Abu(2012).

In supervised learning, a labeled training set (i.e., predefined inputs and known outputs) is used to build the system model. This model is used to represent the learned relation between the input, output and system parameters. In this subsection, the major supervised learning algorithms are discussed in the context of WSNs. In fact, supervised learning algorithms are extensively used to solve several challenges in WSNs such as localization and objects targeting Lu(2009), event detection and query processing (e.g., Jayaraman(2010), Bahrepour(2010)), media access control (e.g., Kim (2009), Shen(2008), Kulkarni(2009)), security and intrusion detection Branch(2013) and quality of service (QoS), data integrity and fault detection Moustapha(2008).

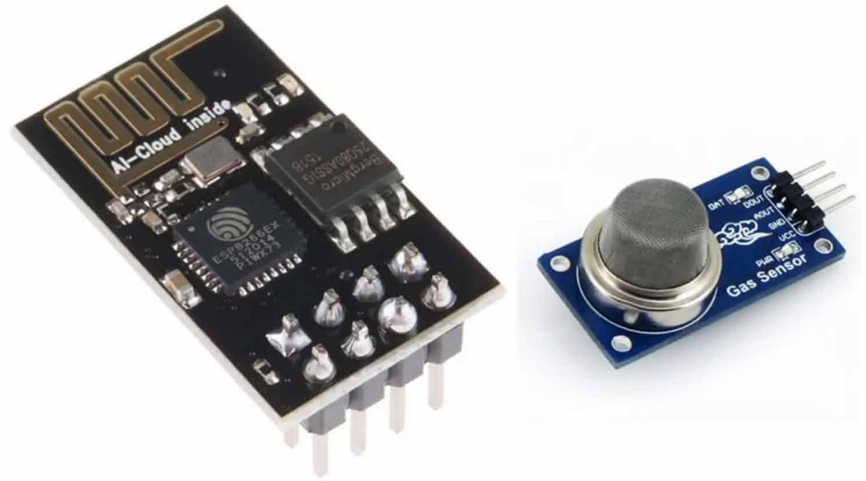
Unsupervised learners are not provided with labels (i.e., there is no output vector). Basically, the goal of an unsupervised learning algorithm is to classify the sample set into different groups by investigating the similarity between them. As expected, this theme of learning algorithms is widely used in node clustering and data aggregation problems (e.g., Masiero(2009), Rooshenas(2010), Macua(2010)). Indeed, this wide adoption is due to data structures (i.e., no labeled data is available) and the desired outcome in such problems.

Internet of Things (IoT) with Artificial Intelligence (AI)

The first part of the IoT-AI scenario is about simple sensors that give us smart solutions. Making a normal tiny sensor to measure the air quality is quite simple. But a sensor node with IoT capability will need an additional communication device. Among the detailed study on communication devices for IoT, the result suggested the device esp8266. It has multiple GPIO pins on it along with a Wi-Fi circuit. This multiple GPIO pins support to connect multiple sensors such as air quality sensor and gas sensors. Figure 4 shows the picture of esp8266 and air quality sensor. Along with these two devices it is not possible to push the data to cloud. Obviously it is necessary to integrate these two devices with a processing unit. Here the option directs with only two. One is Arduino and another is raspberry pi. Since in this smart scenario the hardware device is used only to sense and push the data to cloud for further processing, it is recommended to go with cost effective Arduino than raspberry pi.

Take an example as shown in Figure 5 to use these sensor nodes in a building to monitor the air quality among the surroundings. When start tracking the air quality in that building it is just an instant status that informs the stuff here on the south side of the building the air quality isn't great and on the north side of the building the air quality is very nice. Based on this the people work in south side of the office will go and adjust the thermometer, adjust the thermostat and the fans. Then the people on the north side are going to be too cold. This is a simple scenario where people can think about the alternative solutions based on the sensor data.

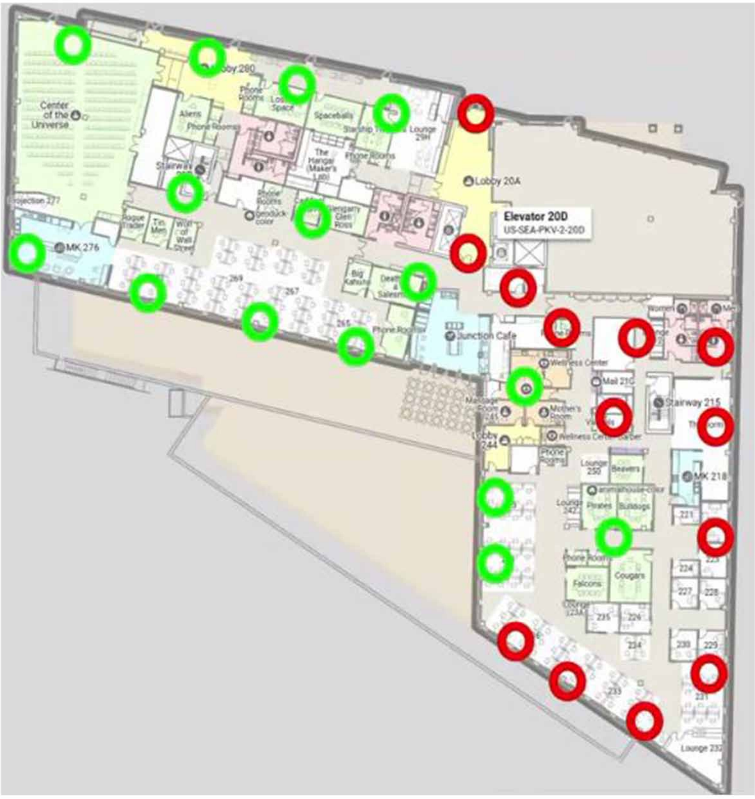
Figure 4. ESP8266 and MQ135 air quality sensor



Consider an another situations where if there was a disaster and it was like maybe a carbon monoxide leak and the sensors are going to show that carbon monoxide leak. If the people don't know fix and the leak is primarily centred around the exit, then people are being driven towards the exit they might be driven towards danger. But when using these sensors like this, it can redirect a defined solution which may be one or two. Further if problem continues, it finalizes the solution with sorry.

But think about scenarios where machine learning comes involved and it start predicting things. It will be the best times of the day to keep the fans on to maximize air quality and reduce our energy footprints. And if there was a gas leak or something like that it can predict the path of the gas leak best based on testing and machine learning models. So emergency responders would know where it is safe to go and where it is dangerous to go.

Figure 5. Deployment of sensor nodes to monitor air quality



This implementation of IoT with machine learning algorithms can be done with many open source solutions. A cloud IoT network with lot of devices can be formed with Google Cloud IoT Core. It is fully managed and allowed to easily secure, connect, manage and ingest data from all of devices. once done then it can start doing interesting things like pulling the data out using big query, using cloud ML, using cloud data lab and using cloud studio to build models and then run inference on these models and have analytics behind that.

Related Studies for Air Pollution Application

Transportation is the main daily activity of the Europeans. Each citizen travels at least one hour per day Eurostat(2007). Therefore, a lot of transportation means, such as buses, trains, cars, etc. exists in cities. This means of transport cause the emission of 12% Co2 Eurostat(2011). Moreover, road population is more than twice as deadly as traffic accidents Yim(2012) and car pollution also damaged the youth health and increased the risk of earlier deaths. This shows how much the awareness and safety of pollution are important.

The more important gasses in the air that affect the human health are ozone (O3), carbon monoxide, sulfur dioxide (SO2), nitrogen oxide, and particulate matter. The Environmental existence of these gasses is analyzed to deliver the current intensity of those gasses in the air so that more people protect themselves from these gasses. Ozone (O3) is made with three oxygen items joint together. It is too dangerous for the living tissues of the human when it contacts to them, such as, it can harm your lungs, effect to a sunburn inside your lungs, a cough, an irritated throat, or an uncomfortable feeling in your chest, Worsened Asthma, emphysema and bronchitis, and may reduce the body's ability to fight infections in the respiratory system. It is made by the reaction of volatile organic compounds (VOC), Nitrogen oxide (NO), and Nitrogen Dioxide (NO2). Therefore, nitrogen dioxide is also dangerous. As more VOC's and NO2 cause more ozone. Sunny weather, less wind, crowded traffic cause increase in ozone. Sulfur dioxide (SO2) adverse respiratory effects including bronchoconstriction and increased asthma symptoms. "Particulate matter" is a complex fusion of extremely small particles and liquid droplets. The particle can be made by acids (such as nitrates and sulfates), organic chemicals, metals, and soil or dust particles. These are so small that they can get deep into the lungs and cause serious health problems.

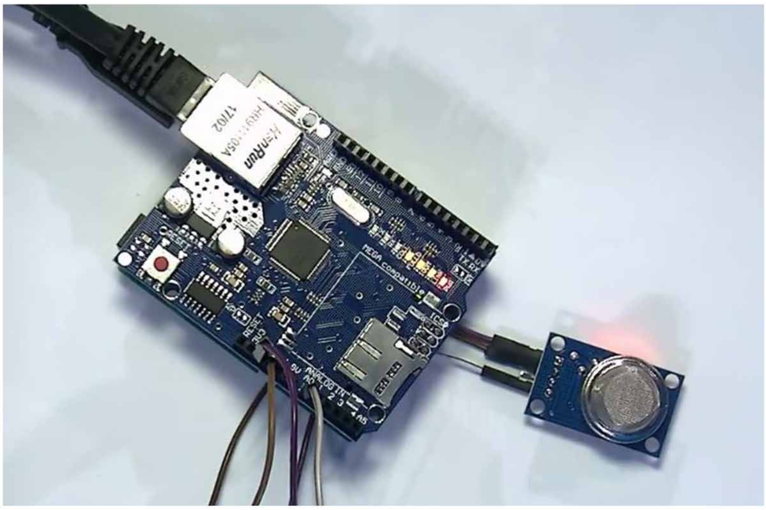
Demo on IoT With Cloud

This section discusses a little demo of environmental monitoring. As discussed in the previous section this demo requires Arduino with an air quality sensor. The Figure 6 shows the hardware specification of a single sensor node that can be used to monitor the environment. This demo uses a nodeMCU which is a type of Arduino with inbuilt Wi-Fi and an air quality sensor MQ135 which is connected to digital pin 0.

Before starting the demon make sure that a firebase running in sink machine. Using firebase the data can store and on cloud fire by getting the readings from the sensor device. So that reading of the air quality sensor has been displayed like 57, 54 and 70. This has been done with Arduino code as shown in sketch 1.

```
Sketch 1
void loop()
{
sensorValue = analogRead(0); // read analog input pin 0
Serial.println(sensorValue, DEC): // prints the value read
delay(2000); // wait 100ms for next reading
}
```

Figure 6. Hardware connection of a sensor node with Air quality sensor



This sketch reads the digital values from pin zero that this is plugged into data pin zero. And it reads the data in every two seconds. Now to write this data to firebase cloud, the URL has to be written as shown in sketch 2.

```
Sketch 2
String setting = strURL + Pg(sensorValue) + ″ HTTP/1.1″; // if
you get a connection, report back via serial:
if (client.connect(server, 80))
{
Serial.println(〝connected″); // Make a HTTP request:
client.println(setting);
client.println(hostString);
client.println(〝Connection: close″);
client.println();
}
else
{
// if you didn’t get a connection to the server:
Serial.println(〝connection failed″);
}
```

It is a simple sketch which reads the data from the sensor and sends the data to firebase. But when doing IOT, security has to be considered as an important issue since the things are just generating data and pushing it up to online database. Hence instead of pushing from the devices directly up to the databases, let the firewall to act as a proxy and then that proxy will proxy out to the databases. Hence, a secure connection between the proxy and the database will be established. Hopefully this secure connection will be enough and than a complex security algorithm in an energy constrained sensor nodes. At the end of this phase the sensor node which senses the air quality and send the data to the cloud database which in firebase in a secure communication. Now the second phase of this demo discusses the process for machine learning on the cloud scale.

Demo on IoT With Machine learning

Consider a simple scenario where a little bit smarter devices such as kabuna are used to detect the object. if the picture has been taken with a camera, it have to detect some object, find what kind of object is present in the images. Tensorflow provide an object detection API. This object detection API provides the labels of each in image and provides the boundary boxes. It is very simple to download the module file and use the API as shown in the following example. Insert the following two statements to detect the object

```
import tensorflow as tf
output_dict = sess.run(tensor_dict, feed_dict = {image_tensor:
image})
```

This API has the dictionary imported cancels with the model file of the model. When pass the images to the run method it will give the output dictionary as below.

```
output_dict['num_detections']
output_dict['detection_classes']
output_dict['detection_boxes']
output_dict['detection_scores']
```

Consider another example where the raspberry pi with the camera attached to a shopping cart. So that it can take a picture of the inside of the shopping cart and apply the object detection API to what kind of object which is detected and how many of them available in the cart. Here the cloud IOT core combined with the cloud and machine learning engine to build them a system that is that provides the production level, scalability and availability. As a summary of this scenario if you want to build a smart communal system for the shopping cart you can have a Raspberry Pi and a camera attached to the shopping cart then use the quad IOT core to collect all the data stored data or storing data on the cloud pub/sub which could be here back-end for your server and we could use the Google kubernetes engine or gke as an Orchestrator for orchestrating everything happening article outside and finally the GK we would be sending the data to the ML Benji that's where we are realigning the prediction with the object detection API. This workflow has been shown in the Figure 7.

The demonstration of this case study has been done as depicted in Figure 8. Here a smart cart has been connected with a webcam, a Raspberry Pi and display. If the kart is filled with fake eggplant and a tomato the output of the object detection API at the monitoring device would look like as shown in Figure 9.

In this example the object detection API has been trained to detect the object based on the trained data set of vegetables for this supermarket case study. Here the detection of new objects with the camera attached with raspberry pi is simple. But training machine with data set is difficult. For this example the machine learning model has been trained with the order of the shopping items that are added to basket. Once the first item has been added to the kart, the raspberry pi can trigger

Figure 7 Google workflow for IoT device data collection and analytics

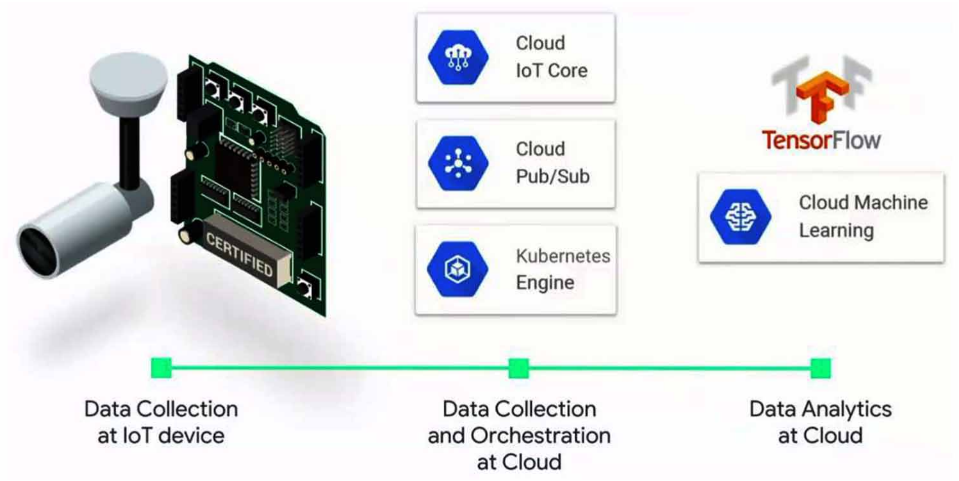
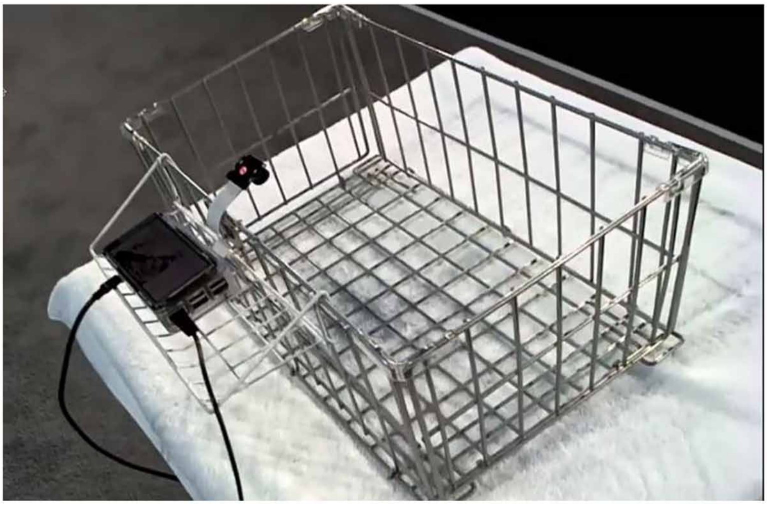


Figure 8. Shopping kart with IoT enabled raspberry pi and camera



a machine learning model that can predict what was the item added to the kart and updates the training set.

If the tomato has added to the cart one hot vector that represents the tomato in this case would have a 1.0 as a value in the vector table. Next if the eggplant is added in the cart vector table would have another 1.0. Similarly for all the new items which are added to the cart will have a value 1.0 in the vector table as shown in Figure 10.

Figure 9. Output of object detection API

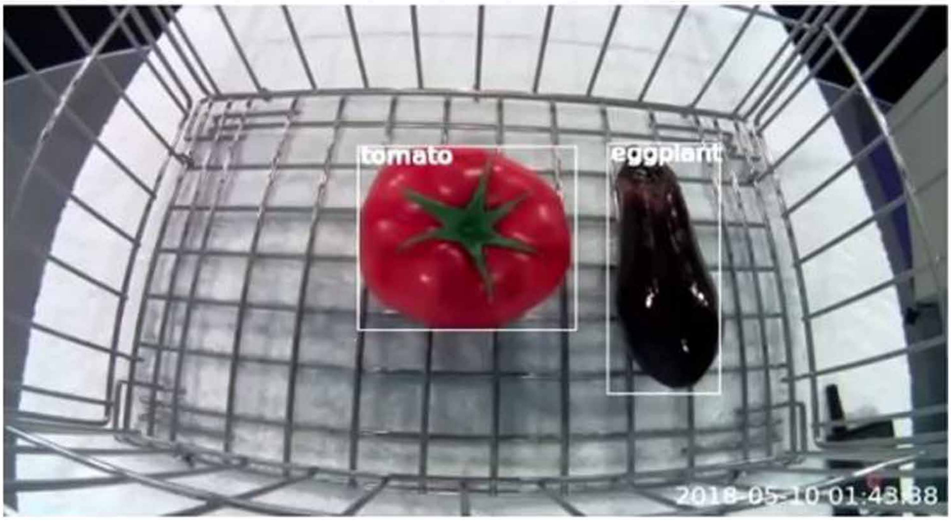
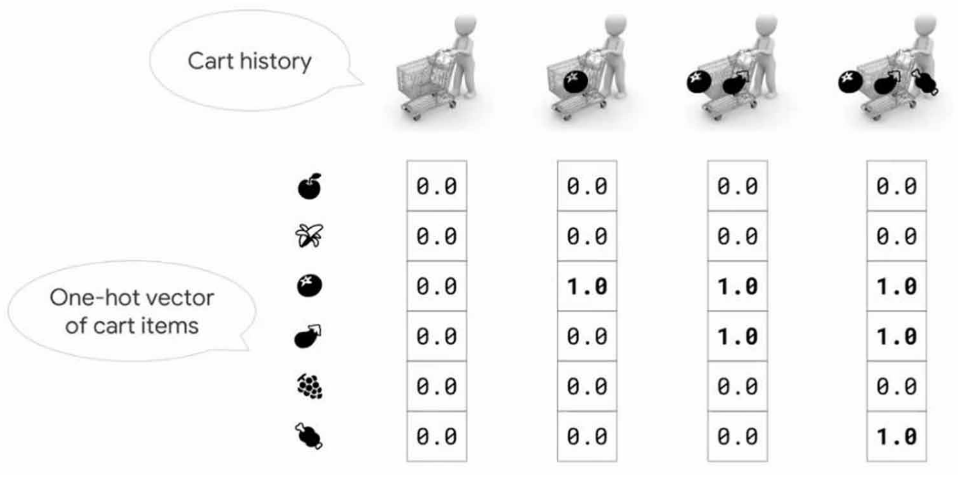


Figure 10 .Vector table of data set



	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0
	0.0	1.0	1.0	1.0
	0.0	0.0	1.0	1.0
	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	1.0

The code which to be added at the Tensorflow for updating the new added items will be

```
SHOPPING_ITEMS = [“onion”, “tomato”, “potato”, _ ]
items_table = tf.contrib.lookup.index_table_from_
tensor(SHOPPING_ITEMS) history = tf.one_hot(items_table.
lookup(featureWcart_history1), len(SHOPPING_ITEMS))
cart = tf.reshape(history, [-1, HISTORY_SIZE, len(SHOPPING_
ITEMS)])
```

In this case the convolution single dimensional composition or one decomposition-composition in machine learning is usually used to detect the certain patterns in a local group of the big data. The Figure 11 represents the 1D convolution based flatten recognition for the kart items in the super market.

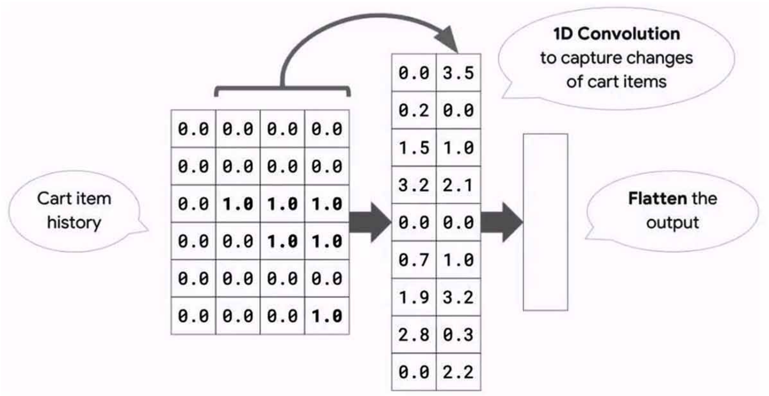
The following code has to be included in Tensorflow to find Flatten vector table is

```
cart = tf.layers.convid(cart, filters=20, kernel_size=3,
padding="valid", activation=tf.nn.relu)
cart = tf.contrib.layers.flatten(cart)
```

Here the CNN or convolutional neural network works for image recognition but you can also apply the convolution to do one single dimensional data such as time series data. In this example the one dimension decomposition on the kart Tensorflow can detect that what kind of changes that happen inside the cart items and flatten the output to get the result.

Now the kart history has been changed and also wants to take other factors such as seasonality whether it's a winter or summer or time of day. Because shoppers may want to choose different food items based on the season argue whether it's a summer hot day or whether it's a cold day. Now combine everything into a single multi-layer perception (MLP) which is a classic neural network with the three layers to predict the next two items. To add with stencil the following code has to

Figure 11. One dimensional convolution based flatten recognition



0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	**1.0**	**1.0**	**1.0**
0.0	0.0	**1.0**	**1.0**
0.0	0.0	0.0	0.0
0.0	0.0	0.0	**1.0**

0.0	3.5
0.2	0.0
1.5	1.0
3.2	2.1
0.0	0.0
0.7	1.0
1.9	3.2
2.8	0.3
0.0	2.2

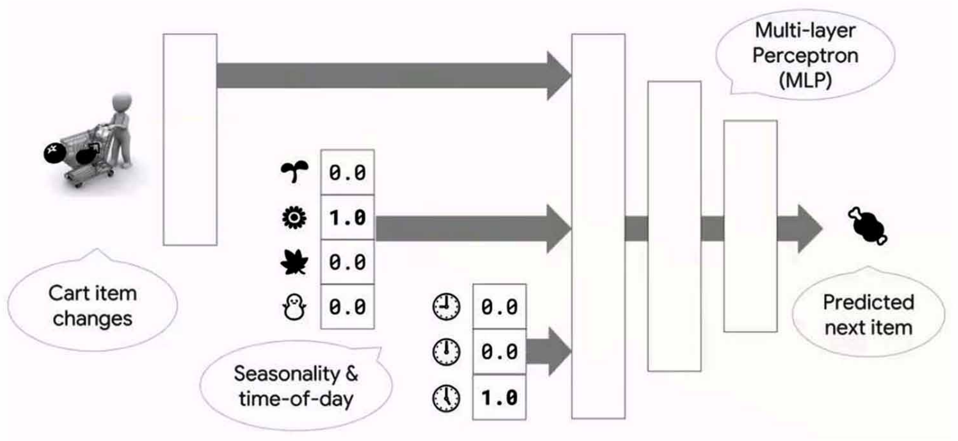
be included to concatenate with everything. The Figure 12 describes the combined stencil using multi-layer perception (MLP).

```
x = tf.concat([cart, season, time], axis=1)
x = tf.layers.dense(inputs=x, units=NUM_UNIT1, activation=tf.
nn.relu)
x = tf.layers.dense(inputs=x, units=NUM_UNIT2, activation=tf.
nn.relu)
logits = tf.layers.dense(inputs=x, units=(len(SHOPPING_
ITEMS)+1), activation=None)
```

This example shows the machine learning prediction about the item which is added to the basket will be converted as image, then based on the image the classification algorithms has been identified which item has been added and how many numbers of items, then three layer MLP takes place to predict the next item to buy and where it is present in the super market.

This a simple example how the crowd can be empowering to for the IOT devices not only for collecting data but also you can analyze it and learn some collective intelligence from it. It is not done just an IOT anymore it is a Internet of smart things. This demolition demonstration was actually built by a Google Cloud partner and they have open sourced everything's on the github. For further reference on this project please visit the github and search with smart shopping navigator to find out what kind of code to be used, where you to have the user interface to build a whole data pipeline for collecting the IOT data, analyzing it and learning the training the model.

Figure 12. Combined stencil using multi-layer perception (MLP)



SUMMARY

This chapter discusses a little bit about Internet of Things in AI and about the trends that are happening and the explosive growth that is going on the growth of actual devices and the amount of data that they're producing. In the second section discussed about the machine learning and its importance. The third and fourth sections of this chapter discussed a lit bit about the famous machine learning algorithms and the process that should be followed in machine learning approaches. And then this chapter discusses cloud scale artificial intelligence with a sensor data on an Arduino. In that section a simple procedure has been discussed that can apply a whole bunch of smart things that can writing data from Internet of Things device to cloud storage. The machine learning with IoT data section explores the Tensorflow object detection API which is useful to detect the objects with raspberry pi. In this section a simple example of super market item prediction has been demonstrated. The overall perception on this chapter discusses about the tools and procedure that can be used to apply machine learning techniques on IoT data and how a machine learning algorithm improve the computation capability of IoT into Intelligent IoT.

REFERENCES

Abu-Mostafa, Magdon-Ismail, & Lin. (2012). *Learning from data*. AMLBook.

Bahrepour, M., Meratnia, N., Poel, M., Taghikhaki, Z., & Havinga, P. J. (2010). Distributed event detection in wireless sensor networks for disaster management. *2nd International Conference on Intelligent Networking and Collaborative Systems*, 507–512. 10.1109/INCOS.2010.24

Branch, J. W., Giannella, C., Szymanski, B., Wolff, R., & Kargupta, H. (2013). In-network outlier detection in wireless sensor networks. *Knowledge and Information Systems*, *34*(1), 23–54. doi:10.100710115-011-0474-5

Eurostat. (2007). *Passenger mobility in Europe*. European Commission.

Eurostat. (2011). *Energy, transport and environment indicators*. European Commission.

Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, *29*(7), 1645–1660. doi:10.1016/j.future.2013.01.010

Jayaraman, P. P., Zaslavsky, A., & Delsing, J. (2010). *Intelligent processing of k-nearest neighbors queries using mobile data collectors in a location aware 3D wireless sensor network. In Trends in Applied Intelligent Systems* (pp. 260–270). Springer.

Jin, Gubbi, Marusic, & Palaniswami. (2014). An information framework for creating a smart city through internet of things. *IEEE Internet of Things Journal, 1*(2), 112-121.

Khan, R., Khan, S. U., Zaheer, R., & Khan, S. (2012). Future internet: the internet of things architecture, possible applications and key challenges. In *Frontiers of Information Technology (FIT)*, *2012 10th International Conference on* (pp. 257-260). IEEE. 10.1109/FIT.2012.53

Kim, M., & Park, M.-G. (2009). Bayesian statistical modeling of system energy saving effectiveness for MAC protocols of wireless sensor networks. In *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*. *Springer Berlin Heidelberg*.

Kulkarni, R. V., & Venayagamoorthy, G. K. (2009). Neural network based secure media access control protocol for wireless sensor networks. In *Proceedings of the 2009 International Joint Conference on Neural Networks, ser. IJCNN'09*. Piscataway, NJ: IEEE Press. 10.1109/IJCNN.2009.5179075

Lu, C.-H., & Fu, L.-C. (2009). Robust location-aware activity recognition using wireless sensor network in an attentive home. *IEEE Transactions on Automation Science and Engineering*, *6*(4), 598–609. doi:10.1109/TASE.2009.2021981

Macua, S., Belanovic, P., & Zazo, S. (2010). Consensus-based distributed principal component analysis in wireless sensor networks. *11th International Workshop on Signal Processing Advances in Wireless Communications*, 1–5.

Masiero, R., Quer, G., Munaretto, D., Rossi, M., Widmer, J., & Zorzi, M. (2009). Data acquisition through joint compressive sensing and principal component analysis. *Global Telecommunications Conference*, 1–6. 10.1109/GLOCOM.2009.5425458

Moustapha, A., & Selmic, R. (2008). Wireless sensor network modeling using modified recurrent neural networks: Application to fault detection. *IEEE Transactions on Instrumentation and Measurement*, *57*(5), 981–988. doi:10.1109/TIM.2007.913803

Rooshenas, A., Rabiee, H., Movaghar, A., & Naderi, M. (2010). Reducing the data transmission in wireless sensor networks using the principal component analysis. *6th International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, 133–138. 10.1109/ISSNIP.2010.5706781

Shen, Y.-J., & Wang, M.-S. (2008). Broadcast scheduling in wireless sensor networks using fuzzy hopfield neural network. *Expert Systems with Applications*, *34*(2), 900–907. doi:10.1016/j.eswa.2006.10.024

Yim, S. (2012). *Public Health Impacts of Combustion Emissions in the United Kingdom*. Cambridge, MA: Department of Aeronautics and Astronautics, Massachusetts Institute of Technology.

Chapter 9

Current Trends:
Machine Learning and AI in IoT

Jayanthi Jagannathan
Sona College of Technology, India

Anitha Elavarasi S.
Sona College of Technology, India

ABSTRACT

This chapter addresses the key role of machine learning and artificial intelligence for various applications of the internet of things. The following are the most significant applications of IoT: (1) manufacturing industry: automation of industries is on the rise; there is an urge for analyzing the energy in the process industry; (2) anomaly detection: to detect the existing fault and abnormality in functioning by using ML algorithms thereby avoiding the adverse effect during its operation; (3) smart campus: in-order to efficiently handle the energy in buildings, smart campus systems are developed; (4) improving product decisions: with the help of the predictive analytics system products are designed and developed based on the user's requirements and usability; (5) healthcare industry: IoT with machine learning provides numerous ways for the betterment of the human wellbeing. In this chapter, the most predominant approaches to machine learning that can be useful in the IoT applications to achieve a significant set of outcomes will be discussed.

DOI: 10.4018/978-1-5225-7790-4.ch009

INTRODUCTION

The Internet of Things (IoT) is the network of various physical devices in-order to exchange data and take appropriate action. In recent years the growth in technology enhances the communication between different devices are made much easier. It is estimated that there will be 30 billion devices by 2020 [Nordrum et al ., 2016]. Some of the IoT applications include automated vehicles, home automation, remote health monitoring etc. In-order to make these devices work in a smarter way or to make IoT applications more intelligent there is need for analyzing the huge amount of data using machine learning algorithm [Mahdavinejad et al., 2016]. Machine learning refers to the set of techniques meant to deal with huge data in the most intelligent way in order to derive actionable insights. Figure 1 refers to the confluence of different fields such as IoT, artificial intelligence and big data

Internet of Things

The Internet of Things (IoT), refers to the collection of inter connected everyday objects over the Internet and also to one another. It provides users with smarter and smoother experiences. Internet of Things is mainly being driven by various sensors that would possibly sense the real world data, some of the widely used sensors are temperature, pressure, gas, smoke, IR, image sensors etc. IoT platform could deliver plenty of functionalities with the intelligence by combining a set of sensors and a communication networks. Thus it could able to improve and achieve effectiveness in their autonomous functionality.

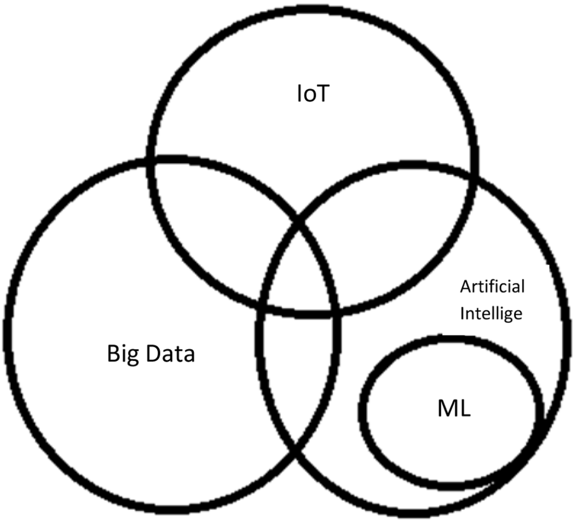
The data that is flowing across the network and devices are being stored and the same is being processed, to derive the required insights. The various stakeholders who are in need of those insights will be served on time. The data sharing is done in a secured way, only the authorized users permitted to use the same.

Let us take a worlds well know Tesla vehicles as an example. The sensors mounted in and around the car senses variety of data and derive many fact values based on the perception from the environment. Then it uploads data into a huge database. The data is further processed and send necessary signals to other vehicles or other parts.

Machine Learning

Machine Learning (ML) is one of the hottest domains of the computer science field that proves the ability of a computer system to “learn” with data, with / without being explicitly trained. It is one of the most essential applications of artificial intelligence. It concentrates on the development of set of programs that can enable data access and make them learn by themselves.

Figure 1. Confluence of IoT, artificial intelligence and big data



The ultimate view point of ML is to automate the data analysis process with the help of algorithms that are enabled with continuous learning skill. Hence ML refers to the set of techniques meant to deal with huge data in the most intelligent way in order to derive actionable insights. There are three major types of algorithms much useful are (i) Supervised (Task driven) (ii) Unsupervised (Data Driven) (iii) Reinforcement learning(learns to react to an environment)

FOREMOST USES OF MACHINE LEARNING IN IOT

- **Clustering of Data:** Clustering deals with grouping objects which are similar to each other based on an objective measure. Binary Classification (positive or negative), Logistic Regression (discrete outcome), K-Means for clustering the data are some prevalent algorithms that's being useful in clustering in Machine Learning. From any real time data to identify certain behavioral analysis is possible. Example: Medical data, data from the devices such as gyroscope, accelerometer etc.
- **Anomaly Detection:** Anomaly detection is a technique used to identify uncommon patterns that do not conform to anticipated behavior, called outliers. It finds many applications in business, from intrusion detection (identifying abnormal patterns in network traffic that could sign a hack) to

system health monitoring (spotting a malignant tumor in a scan report), and from fraud detection in credit card transactions.

- **Prediction of Data Trends:** Predictive analytics utilizes historical data to predict future happenings. Usually, historical data is used to build a mathematical model that captures significant trends. The predictive model is then used on present data to predict what will happen next, or to suggest activities to be taken to achieve the optimal outcomes.

STANDARD MACHINE LEARNING ALGORITHMS USED IN IOT APPLICATIONS

- **K-Nearest Neighbour (k-NN):** K nearest neighbours is a simple algorithm that considers all available cases and classifies new cases based on a similarity measure / distance functions. K-nearest neighbor classifier is one of the supervised classifier for performing pattern classification task. Knn address the pattern recognition problems and also the best option for deal with some of the classification related tasks. The simple edition of the K-nearest neighbor classifier algorithms is to predict the target label by finding the nearest neighbor class. The closest class will be identified using the distance measures like Euclidean distance. In an IoT application, to check whether all the sensors used for an application works normally or abnormally can be figure out using K-NN approach [3]. The average reading of neighbouring sensor can be calculated using any of the measures like Euclidean distance and hence the malfunction of the sensor node is can be predicted. Hence it can be useful at the appropriate places in an IoT environment to provide better solutions.

K-Nearest Neighbor (Knn) Algorithm Pseudocode

Let (X_i, C_i) where i = 1, 2……., n be data points.

X_i denotes feature values & C_i denotes labels for X_i for each i.

Assuming the number of classes as ‘c’

$_{Ci} \in \{1, 2, 3, \ldots\ldots, c\}$ for all values of i

Let x be a point for which label is not known, and like to find the label class using k-nearest neighbor algorithms.

Pseudocode of Knn Algorithm

```
1. Calculate "$d(x, x_i)$" $i = 1, 2, \ldots.., \mathbf{n}$; where d denotes the Euclidean distance between the points.
2. Arrange the calculated n Euclidean distances in non-decreasing order.
3. Let k be a +ve integer, take the first k distances from this sorted list.
4. Find those k-points corresponding to these k-distances.
5. Let $\mathbf{k}_i$ denotes the number of points belonging to the $i^{th}$ class among k points i.e. $k \geq 0$
6. If $k_i > k_j \; \forall \; i \neq j$ then put x in class i.
```

- **Decision Tree (DT):** A decision tree is a graph that uses a branching method to show every possible outcome of a decision. In a health care industry, to classify the severity of the diseases based on certain parameter, decision tree algorithm can be employed. Apart from this reliability of the sensor node can be identified using the features such as mean time to failure, mean time to restore. Decision Tree algorithm belongs to the folks of supervised learning algorithms. Decision tree algorithm can be used for resolving regression and classification problems too.

The main aim of classification is to predict the target class (Yes/ No). If the trained model is for predicting any of two target classes. It is known as binary classification. The main goal of regression algorithms is to predict the discrete or a continues value. In some cases, the predicted value can be used to identify the linear relationship between the attributes.

The general motive of using Decision Tree is to create a training model which can be useful to predict class or value of target variables by learning decision rules inferred from prior data(training data).Each internal node of the tree corresponds to an attribute, and each leaf node corresponds to a class label.

In decision trees, for predicting a class label for a record, it must be started from the root of the tree. Then compare the values of the root attribute with record's attribute. On the basis of comparison, the branch corresponding to that value and jump to the next node will be followed.

Pseudocode of Decision Tree Algorithm

```
1. Place the best attribute of the dataset at the root of the tree.
2. Split the training set into subsets. Subsets should be made in such a way that each subset contains data with the same value for an attribute.
```

3. Repeat step 1 and step 2 on each subset until you find leaf nodes in all the branches of the tree.

- **Neural Networks (NNs):** Neural networks can learn from multiple perceptron and take appropriate decisions which make them suitable for several challenging application. This can be used to recognize a pattern, to predict someone's behavior, recognizing objects etc. It can be more useful in many industrial sectors including medical fields, logistics tracking, smart cities and automobiles where IoT plays vital role. NN's would be useful in the classification of normal and threat patterns on an IoT Network.
- **Support Vector Machines (SVMs):** Support Vector Machine" (SVM) is a supervised machine learning algorithm which can be applied for both classification and regression challenges. However, it is widely used in to providing solution to classification problems. In an IoT application, to check the proper functioning of the sensors SVM can be applied to identify its behavior (normal / malicious). This can be done by correlating the temporal and spatial information of a node and its reading are plotted in the feature space. SVM partition this into various part and a new reading can be classified based on the gap it is going to possesses.

Support vector machine which is a supervised learning algorithm[3]. When a dataset contains features and class labels both then Support Vector Machine can be used For a dataset consisting of features set and labels set, an SVM classifier builds a model to predict classes for new examples. It assigns new example/data points to one of the classes. If there are only 2 classes then it can be called as a Binary SVM Classifier.

Types of SVM classifiers:

1. Linear SVM Classifier
2. Non-Linear SVM Classifier

Svm Linear Classifier

In the linear classifier model, it is assumed that training examples plotted in space. These data points are expected to be separated by an evident gap. It predicts a straight hyperplane dividing two classes. The main focus while drawing the hyperplane is on maximizing the distance from hyperplane to the nearest data point of either class. The drawn hyperplane is called as a maximum-margin hyperplane.

SVM Non-Linear Classifier

In the real world, the dataset is generally dispersed up to some extent. To solve this problem separation of data into different classes on the basis of a straight linear hyperplane can't be considered a good choice. For this Vapnik suggested creating Non-Linear Classifiers by applying the kernel trick to maximum-margin hyperplanes. In Non-Linear SVM Classification, data points plotted in a higher dimensional space.

- **3.5 Principal Component Analysis (PCA):** The main aim of PCA is data compression and dimensionality reduction. This helps in reducing the amount of data that is transmitted among various sensor nodes by filtering the unwanted data. Moreover it could able to simplify the problem solving approach to a greater extend as it considers/ selects only few parameters for making a decision.

APPLICATIONS

- **Manufacturing Industry**: Automation of industries is in rise, there is an urge for analyzing the energy in process industry. It needs the monitoring and diagnosis of systems in order to perform in a better and efficient manner. Monostori 2014 describes a Cyber-Physical Production Systems. The parallel development of computer science, information and communication technologies on one side and of manufacturing on the other side and their mutual influence paved ways to *Cyber-Physical Systems* (CPS). CPS is a computational entities interconnected with physical world to gathers and process data / information via internet. Autonomous cars, robotic surgery, intelligent buildings are few examples of CPS system. *Cyber-Physical Production Systems (CPPSs)*, relies on the development of computer science, information, communication technologies and manufacturing. CPPS consist of both autonomous and cooperative elements across domains such as production and logistics sub system with an establishment of man machine communication. CPPS operates on a decentralized manner. Few of the CPPSs in manufacturing industry include:
- **Intelligent manufacturing systems (IMS)**: Expected to solve unseen or unpredicted problem.
- **Biological manufacturing systems (BMS):** Develops system based on biologically inspired ideas
- Reconfigurable manufacturing systems (RMS)
- **Digital Factories (DF):** Mapping of business processes into digital

- **Holonic Manufacturing Systems (HMS):** agent-based manufacturing system with properties of autonomy and cooperation

The growth of CPPS paves way to more versatile and enormous expectation. The partial fulfilment of these expectations would represent real challenges. Monostori 2014 list few of the common R&D challenges in CPPS systems as:

- Context-adaptive and autonomous systems.
- Cooperative production systems
- Identification and prediction of dynamical systems
- Robust scheduling
- Fusion of real and virtual systems.
- Human-machine symbiosis.
- **Anomaly Detection:** To detect the existing fault and abnormality in functioning by using the ML algorithms thereby avoiding the adverse effect during its operation. The fast growing popularity of IoT led to various devices released with vulnerability which leads to different kind of attack. DIoT is a a system for **D**etecting compromised **IoT** devices [Nguyen, Thien Duc, et al., 2018]. It uses a self-learning approach (autonomous) which combines device type identification and device type specifications to classify devices based on their communicative profile and thereby detect any anomalous behavior. It detects the compromised node in a much faster and effectively (94%) manner with very few false alarms. It classify IoT malware attack into three stages as: (1) intrusion: gain unauthorized access (2) infection: attacker upload malicious code to infect the device and (3) monetization: malware takes malicious action like DDoS attack. Some of the challenges faced on anomaly detection are: (1) Dynamic threat landscape, (2) Resource limitations, (3) IoT device heterogeneity, (4) Scarcity of communications and (5) False alarms.

The growth of Internet of Things (IoT) paved ways to a number of benefits. At the same time it has various security vulnerabilities associated with it. Now a day's organization employs BYOIoT(bringing their own IoT devices) which has raised the security issues[Meidan, Yair, et al, 2017]. Therefore organization needs an intelligent system to identify the suspicious devices which are connected to the network. Suspicious IoT devices are identified using the random forest machine learning algorithm. Two major types of attacks commonly seen are:(1) Untargeted: In this approach the various IoT device are infected by malware. Cross contamination is an example of untargeted attack and (2) specifically targeted: Attacker intentionally fix the malware to the IoT device their by gaining the data in future. Such type of attack requires excellent hacking skills. Supply chain attack is an example of specifically targeted attack. The security policy of an organization should check the various IoT devices which are connected to the network by monitoring the traffic. It should check for the data origination, type of IoT device and whether it is authorized or not. If the device is found to be unauthorized appropriate action should be taken.

The Approach employed by Meidan, Yair et al, 2017 will initialize a set of authorized IoT device and structured set of traffic data are defined. Authorized IoT device are identified by using the classifier algorithm (Machine learning algorithm) by taking the TCP/IP data traffic. The classifier selected for this purpose is Random forest. It combines decision tree induction with ensemble learning. Once the classifier is trained, new streams of data traffic is given and checked whether the device is authorized one or not. Some of the common limitations are (1) the Approach is tested on a particular circumstance; if the device varies on organization then more generalized approach has to be devised. (2) The present approach describes only about TCP/IP communication technology. It does not focus on Bluetooth or Zigbee protocol and (3) the data being collected represent only normal behavior. Exceptional cases are to be considered and classifier has to be trained accordingly.

- **Smart Campus**: In-order to efficiently handle the energy in buildings, smart campus systems is developed. Whenever the user is not using the resource properly the system will intimate and make them used it in an efficient manner. Resource such as switching off the computer system, laptop, tube lights, fan can be controlled by the user mobile phone itself.

Smart Home

Home can be monitored by the data produced from various sensors fixed inside the home to make it smart [Shafie-Khah et al 2016]. For example, novel demand response (DR) methods can be useful, or customers can be alerted in the case where pollution is above its usual limit through monitoring different parameters. Cutting-edge of IoT technology facilitates smart homes and appliances including smart TVs, home security system, lighting control, fire detection, and temperature monitoring. The sensors of these appliances monitor the conditions and environment and send surveillance data to a central controller at home which enables the householder to continuously monitor and control the home even from outside and make the best decision under every circumstance [Li et al 2011].

Smart City

The main goal of a smart city is to keep the city connected together and ensure proper functioning of each subsystem with the help of information technology. The factor that facilitates the development of a city includes communicating systems, network automation and technology for measurement. Smart metering integrated Internet of Things (IoT) architecture employed in smart city applications. This can be called as Advanced Metering Infrastructures (AMI) [Lloret, Jaime, et al., 2016] . It involves communication protocol, the data format, the data gathering and big data based decision system for efficient management of electricity, water and gas their by having balance between demand and consumption. AMI act as vital role in resource (electricity, water and gas) distribution networks. The components of AMI system include hardware for communication, display, controller, sensor and software for data managements and decision support system. It provides two way communications. The author proposes a three layer approach where Layer 1 comprises of various hardware components such as meters, hub, gateway etc. Layer 2 comprises of devices responsible for receiving data at the utility side and layer 3 handles decision support system using artificial intelligent and billing system. In-order to process huge volumes of data collected through smart meters, author employs spark. By using Spark machine learning library we can performs operation likes prediction, identifying incidents and categorizing user. Three models has been formulated, (1) First model predicts the future consumption and categorize into short and medium term (2) Second model givens an alarm when something goes wrong. And (3) Apply K-means unsupervised machine learning algorithm to categorize user/ customer based on their resource consumption. Two main problem with this system are (1) false fraud incident may arise i.e. if the users consumption patterns vary an anomaly is identified (2) issues related to the data privacy i.e. based on the user behavior data are collected and analyzed pattern are extracted. Real time data may leads to privacy issues.

Meeting the essential needs of the citizens is a tough process because of the density of the population and it must be addressed by providing the ICT based IoT services. Therefore, there has been a significant development of digital devices, such as sensors, actuators, smart phones and smart appliances that motivates the vast commercial intentions of the Internet of Things (IoT). It is always likely to interconnect all devices and build communications between them through the Internet [Rathore et al 2016]. It is very mandatory to gather the information for daily management and long term planning of the city growth.

For example the real time information about any public or private transportation services must be tracked and the necessary data should be generated further analysis and based on which the decisions will be taken. The parameters that would help to track would be location, parking spaces, traffic history, pollution of different categories and areas, energy consumption etc. should be gathered constantly. IoT played a vital role in addressing these types of issues [Rathore et al., 2015]

In order to achieve higher efficiency in smart grid communications a two-way relay network with an orthogonal frequency division multiple accesses is proposed [Zhu et al ., 2015]. The global grid infrastructures are useful in connecting the smart and self-configuring devices. IoT has set of real objects that are distributed in nature .It has limited storage capacity and processing speed. It is aiming for improved reliability, performance and security of the IoT infrastructure of the smart cities [Botta et al ., 2016]. Table 1 represents the various IoT network layers and its functionality.

Actual IoT Applications for Smart Cities

The IoT uses the Internet to merge various heterogeneous things. Accordingly and for providing the ease of access, all existing things have to be linked to the Internet. The reason behind this is that smart cities include sensor networks and connection of intelligent appliances to the internet is essential to remotely monitor their treatment such as power usage monitoring to improve the electricity usage, light management, air conditioner management. To get this aim, sensors are able to be extended at various locations to gather and analyze data for utilization improvement. Botta et al 2016 illustrates the major utilizations of the IoT for a smart city. The key aims in this field of knowledge are expressed in the following subsections.

Table 1. IoT network Layers and its functionality

LAYERS	FUNCTIONS	DEVICES
PERCEPTION LAYER	Identify, detect objects, gather information, and interchange information with other devices through the Internet communication networks	Radio Frequency Identification Devices (RFID), cameras, sensors, Global Positioning Systems (GPS)
NETWORK LAYER	Forwarding data from the perception layer to the application layer under the checks of devices' capabilities, network limitation and the applications' is the job of the network layer.	Bluetooth and ZigBee which are used to carry the information from perception devices to a nearby gateway based on the competences of the communicating events [Jaradat et al 2015]. Basically a short distance service. WiFi, 2G, 3G, 4G, and Power Line Communication (PLC) carry the information over long distances based on the application.
APPLICATION LAYER	The information is received and processed for taking better decisions	smart homes, smart cities, power system monitoring, demand-side energy management, synchronization of distributed power storage, and integration of renewable energy generators [Hancke et al 2012]

- **Improving Product Decisions:** Products are developed based on the user's requirements. With the help of the predictive analytics system products are designed and developed based on the user's requirements and usability. A supply chain is the linked network of people, organizations, resources, deeds, and technologies tangled in the manufacturing and sale of a product or service. A supply chain begins with the delivery of raw materials from a supplier to a manufacturer in the first phase and ends with the delivery of the finished product or service to the end consumer in the second phase. Supply Chain Management (SCM) [1] superintends each and every point of a company's product or service, starting from initial conception to the final sale. The well managed SCM could increase the company revenue, decrease the costs. The success of any business is inseparably linked to the enactment of the supply chain.

More than fifty percent of major new business processes will incorporate some element of IoT by 2020. Maximum of 26 billion internet-connected 'smart' devices will be installed, generating some $300 billion by the end this decade. A thirty-fold increase in internet-connected physical devices will significantly amend the operation of any supply chain operates. Many globalized industries would recognize the revolution of IoT, predominantly in manufacturing, retailing and service industries. Many distributions are focused on identifying, locating, and tracking the status of assets. 58% to 77% of surveyed organizations use locating objects, containers, and personnel as the top fundamental functions of IoT solutions

The postal services use smart mailboxes in remote areas to see whether they're empty and to avoid a unwanted journey before collection. Temperature sensitive pharmaceutical products are being monitored with sensors to confirm product reliability after leaving the warehouse. Data from such sensors can be integrated with business information systems to deliver rich business intelligence. Retailers combine the physical store online presence, to improve the visibility of data at the vital points in the supply chain. A platform on a truck can transfer messages presenting exactly what products, sizes and style differences are included, not to mention the temperature or humidity goods are being transported in. Sensors are even useful in locating the specific products and staff in large

Sensors can even be used to locate the whereabouts of products and staff in large warehouse and even on the road to compute the time of arrival. Automated data capture provides real-time prominence of stock and eludes manual counting and human errors. A bidirectional remote communication is enabled with embedded sensor technologies, which is spread over one million elevators worldwide. The captured data would help technicians to make out their decisions from remote and also can also possible to initiate repair options [2]. This will definitely increase the machine uptime and customer service. The IoT is having a control over supply chain to device the external environment. Unlike previous generations of passive sensors, the IoT will allow a supply chain to control the external environment and accomplish decisions. IoT not only can transfer data about features such as the temperature and application of the machine, but also can change equipment settings and process workflow to improve performance.

The current warehouses [Steve Banker 2014] may operate manually, semi-automatically or fully automatically by " re-intellectualize" the existing control systems for handling data coming from sensors installed. The forklift is extremely reliant on the operator. The innovative smart forklift comprises sensors for speed control, anti-slip technology, and collision detection among devices and sensors. When a forklift is integrated to a Warehouse Management System, it can able to move faster and improve the productivity. Speed control is also done to ensure safety. For example, RFID tags positioned in the floor can indicate the forklift that certain warehouse section is profoundly passage by workers, and then the forklift is agreed an automatic speed limit when near to this section. The greatest cutting-edge forklifts are put up with real-time location systems that allow drivers to progress to a stated location and pick up (or put down) a load without the need for drivers to probe the location to verify that they have picked up (or delivered) the correct load.

As Phil Van Vormer clarifies, the Internet of Things (IoT) expresses the real-time prominence of the inventory. Without real-time status, inventory management trusts on predicting "Lack of real-time visibility means impossible to know about the expend time of drivers with load, even if they prefer the utmost operative route and whether enhancements could be made to how pallets flow all over the warehouse." Besides the internet of things improves inventory counting: "Manual data collection influences inventory disorder. Too many warehouse operators spend a disparate amount of time for chasing missing or misplaced pallets as a result of data entry errors. It is overcome in the connected warehouse, such problems are removed since every single pallet is tracked to the sub inch. Sensors effectively replaces the human portion, possibly leading to 100 percent inventory accuracy."

Supply chain management (SCM) [Machado et at 2016] accomplishes to enhance processes and association with other companies in the supply chain (suppliers and customers) in order to build extra value. When SCM is already profoundly supported by numerous IT solutions, the Internet of Things can be of boundless value by providing further information. One of the main challenges in SCM is reducing the bullwhip effect. A major reason of the bullwhip effect is information alteration. For a better information flow, the Internet of Things is able to activate all appropriate players in the supply chain upon the trade of a product. In customary processes, information on demand was only passed to one's direct downstream companion instead of distributing this information with the whole chain. Fifteen new sophisticated RFID chips used in the Internet of Things permit the recording of all types of manufacturing information, production date, expiry date, warranty period, after sales details consenting real time and more efficient supply chain management. If its possible to have a real-time aspect in the supply chain operation, production capacity can be higher, that leads to more productivity with the same investment.

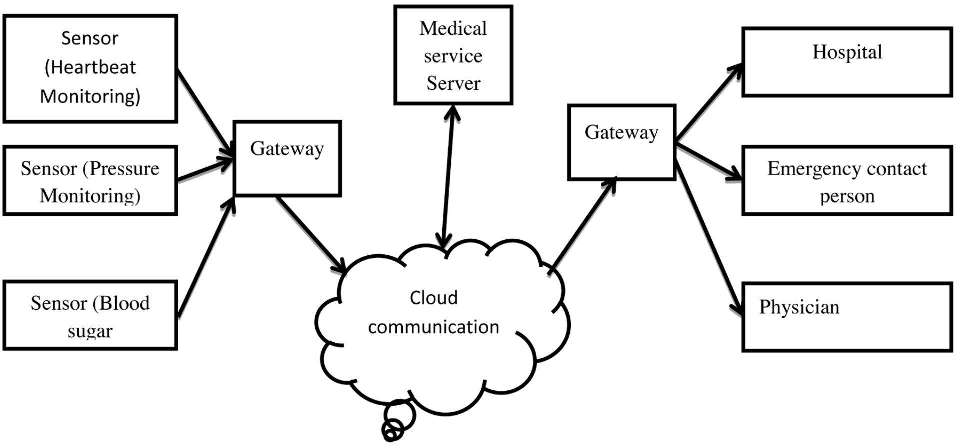
- **Healthcare Industry:** IoT with machine learning provides numerous ways for the betterment of the human wellbeing. Early diseases prediction is possible with the help of continuous monitoring and taking appropriate decision at right time. Recently, IoT has become more useful in the healthcare domain. In the Healthcare field IoT can analyze data send by the connected devices through cloud services to the medical practitioners. When the features of IoT is integrated into medical devices improves the quality and service of care for patients especially elderly people and kids [Darshan et at 2015]. IoT in healthcare can maintain patient's health record and make them accessible at anytime from anywhere. The wearable technology plays vital role in patient health monitoring service. This would help doctors to do remote monitoring and services. Since it is possible to assess over internet, on time needy service can be provided and patients life are saved.

Most of the countries are having only poor healthcare infrastructure and to extend and improve their facility further portable communication devices can be utilized. Since it is almost everyone are using portable devices for communication and it is also economical, can be used for good cause like healthcare services. The provision for monitoring the patient in the real time as well as all real time data of the patient is also analyzed and different insights are derived [Dhar et at 2014]. This will definitely help the doctors to take the clinical decisions in a faster way.

Improving the effectiveness of healthcare and the need of providing excellence care to patients is one of the thought-provoking things of modern society. Active healthcare be contingent on rapidity and correctness, supporting many people and a vast series of devices which are involving with IoT. Hence, IoT has become more useful in the area of healthcare[Catarinucci et al 2015]. Figure 1 represents a model of an IoT for healthcare domain which consist of various sensor to monitor the patient details and takes appropriate steps during emergency time.

IoT focused sensor can be used to monitor patients in an uninterrupted mode. The patient needs close care due to their bodily status, which is a noninvasive monitoring. Every patient connected in the network is closely monitored for their physiological as well as physical information that need to be analyzed using the gateways. All the relevant information will be stored cloud. Further the information will be processed for the diagnosis and prediction of disease. This will support the

Figure 2. Model of an IoT in healthcare domain



medical practitioners to improve their decision with quality care and reduced cost [Niewolny 2013].

L. Catarinucci *et al* 2015 proposed an IoT-aware architecture for smart healthcare systems . An IoT aware smart architecture for automatic monitoring and tracking of patient's personal, and biomedical data is implemented for the patients within hospitals and nursing institutes. A smart hospital system (SHS) is established, which be dependent on different technologies, precisely RFID, WSN. Smart mobiles phones are interoperating through a constrained application protocol (CoAP)/IPv6over low-power wireless personal area network (6LoWPAN)/REST infrastructure. In real time,SHS can collect,both environmental factors and patient's physiological boundary related information through an ultra-low-power hybrid sensing network (HSN) which is composed of 6 LoWPAN nodes integrating the UHF RFID functionalities. The data which is sensed, is delivered to a control center where an advanced monitoring application (MA) makes these things easily accessible by both the local and remote users via REST web service.

A term personalized healthcare [Kulkarni et at 2014] is derived based on the unique characteristics of an individual which includes biological, behavioral, social and cultural practice of well-being. This enables each and every person to follow the simple healthcare norm of "the correct care for the correct person at the correct time", which leads to better outcomes and improvement in fulfilment thus making healthcare cost-effective. A viable service focuses on the prevention, early pathology detection, and homecare instead of the costly clinical one. It verifies the overall well-being to anticipate needs and confirm agreement to healthcare plans. Internet of Things prospects to manage the personalized care services and can maintain a digital identity for every individual. Different equipment is used in healthcare, to communicate and to make the ubiquitous system-of-system. The classifications of IoT based personalized healthcare systems are Clinical care and Remote Monitoring.

EMR is a digital form of regular paper-based medical file records of an individual. EMR systems support to make available quick access to health care info remotely, at anytime and anywhere with the accessibility of IoT technology [Mohd Ibrahim Bhat et al 2017]. EMR takes away the process of structured or unstructured paper form, which could be cumbersome to access at a glance. In this paper, a patient's medical information is entered into an EMR system of health centre in the first instance. The EMR of the patient will be updated continuously throughout his or her stay in the health centre, and linked to an RFID tag. The EMR of the patient includes the following stored information: patient's bio-data, diagnostic information (from medical devices such as ECG device), medical history, prescriptions, laboratory results, blood pressure results, vital signs and medical bills. Whenever a patient seeks medical care in the health centre, the doctors will retrieve the health information of the patient from the EMR system through the use of RFID technology. Then, the retrieved information would help in to analyse and diagnose the patient's illness. Doctors can further take expert advice by sharing the information with consulting specialists if the need arises. This system ensures structure, efficiency and security because no other patient can use another's medical file.

CONCLUSION

In this chapter the most predominant approaches of machine learning that can be useful in the IoT applications to achieve significant set of outcomes will be discussed. This would guide researchers and business people to have an idea and to implement their needs in a better way. It can also help them to find out the feasibility and issues to be addressed in the project.

REFERENCES

Bank, S. (2014). *Warehouse Management Systems & Warehouse Control Systems in the Age of the Internet of Things*. SupplyChain247.

Botta, A., de Donato, W., Persico, V., & Pescapé, A. (2016). Integration of Cloud computing and Internet of Things: A survey. *Future Generation Computer Systems*, *56*, 684–700. doi:10.1016/j.future.2015.09.021

Botta, A., de Donato, W., Persico, V., & Pescapé, A. (2016). Integration of Cloud computing and Internet of Things: A survey. *Future Generation Computer Systems*, *56*, 684–700. doi:10.1016/j.future.2015.09.021

Catarinucci, L. (2015). An IoT-Aware Architecture for Smart Healthcare Systems. IEEE Internet of Things Journal, 2(6), 515-526. doi:10.1109/JIOT.2015.2417684

Darshan, K. R., & Anandakumar, K. R. (2015). A Comprehensive Review on Usage of Internet of Things (IoT) in Healthcare System. *International Conference on Emerging Research in Electronics, Computer Science and Technology (ICERECT)*, 132-136, 374–380. 10.1109/ERECT.2015.7499001

Dhar, K., Bhunia, S. S., & Mukherjee, N. (2014). Interference Aware Scheduling of Sensors in IoT Enabled Health-Care Monitoring System. *Fourth International Conference of Emerging Applications of Information Technology*, 152-157. 10.1109/EAIT.2014.50

Hancke, G., Silva, B., & Hancke, G. Jr. (2012). The Role of Advanced Sensing in Smart Cities. *Sensors (Basel)*, *13*(1), 393–425. doi:10.3390130100393 PMID:23271603

Jaradat, M., Jarrah, M., Bousselham, A., Jararweh, Y., & Al-Ayyoub, M. (2015). The Internet of Energy: Smart Sensor Networks and Big Data Management for Smart Grid. *Procedia Computer Science*, *56*, 592–597. doi:10.1016/j.procs.2015.07.250

Kulkarni, A., & Sathe, S. (2014). Healthcare applications of the Internet of Things: A Review. *International Journal of Computer Science and Information Technologies*, *5*(5), 6229–6232.

Li, X., Lu, R., Liang, X., Shen, X., Chen, J., & Lin, X. (2011). Smart community: An internet of things application. *IEEE Communications Magazine*, *49*(11), 68–75. doi:10.1109/MCOM.2011.6069711

Lloret, J., Tomas, J., Canovas, A., & Parra, L. (2016). An integrated IoT architecture for smart metering. *IEEE Communications Magazine*, *54*(12), 50–57. doi:10.1109/MCOM.2016.1600647CM

Machado, H., & Shah, K. (2016). *Internet of Things (IoT) impacts on Supply Chain*. machado2016internet.

Mahdavinejad, M. S. (2017). *Machine learning for Internet of Things data analysis: A survey*. Digital Communications and Networks.

Meidan, Y. (2017). *Detection of Unauthorized IoT Devices Using Machine Learning Techniques*. arXiv preprint arXiv:1709.04647.

Monostori, L. (2014). Cyber-physical production systems: Roots, expectations and R&D challenges. *Procedia Cirp*, *17*, 9–13. doi:10.1016/j.procir.2014.03.115

Nguyen, T. D. (2018). *IoT: A Crowdsourced Self-learning Approach for Detecting Compromised IoT Devices*. arXiv preprint arXiv:1804.07474.

Niewolny. (2013). *How the Internet of Things Is Revolutionizing Healthcare*. Freescale Semiconductors.

Nordrum, A. (2016). *Popular Internet of Things Forecast of 50 Billion Devices by 2020 Is Outdated*. IEEE.

Rathore, M. M., Ahmad, A., Paul, A., & Rho, S. (2016). Urban planning and building smart cities based on the Internet of Things using Big Data analytics. *Computer Networks*, *101*, 63–80. doi:10.1016/j.comnet.2015.12.023

Rathore, M. M., Ahmad, A., Paul, A., & Rho, S. (2016). Urban planning and building smart cities based on the Internet of Things using Big Data analytics. *Computer Networks*, *101*, 63–80. doi:10.1016/j.comnet.2015.12.023

Shafie-Khah, M., Heydarian-Forushani, E., Osório, G. J., Gil, F. A. S., Aghaei, J., Barani, M., & Catalão, J. P. S. (2016). Optimal Behavior of Electric Vehicle Parking Lots as Demand Response Aggregation Agents. *IEEE Transactions on Smart Grid*, *7*(6), 2654–2665. doi:10.1109/TSG.2015.2496796

Zhu, C., Leung, V.C.M., Shu, L., & Ngai, E.C.H. (2015). Green Internet of Things for Smart World. *IEEE Access, 3*, 2151–2162.

Chapter 10
Social Internet of Things

Sejal Atit Bhavsar
Gandhinagar Institute of Technology, India

Brinda Yeshu Pandit
Gandhinagar Institute of Technology, India

Kirit J. Modi
Ganpat University, India

ABSTRACT

Internet of things has gathered significance within the latest technology domain and trends. As a result, it offers greater ways of accessing data and utilizing intelligent systems. IoT applications are developed for specific scenarios (i.e., smart home, smart transportation, smart agriculture, e-health, etc.). Such IoT applications are inefficient for sharing data and knowledge through services. This results in an inefficient exploitation of different IoT service applications. Social internet of things (SIoT) has efficient and effective ways to support these kinds of services. A concept of social internet of things has been proposed in this chapter in order to support efficient data sharing. This chapter explores related work and literature study on social internet of things, concentrates on mapping IoT with SIoT, and describes a possible architecture for SIoT, components, layers and processes of SIoT. It also illustrates applications, where SIoT can be used, and at the end, the authors provide a few challenges related to SIoT.

DOI: 10.4018/978-1-5225-7790-4.ch010

INTRODUCTION

Internet of Things is a network of smart things, which are capable of identifying the physical world and communicate with each other without human intervention. Today people are connected with many digital things which makes their life more comfortable. The number of such devices is growing rapidly in day to day life. These devices may be actuators, sensors or any kind of physical devices, which are capable to sense the environmental change, and able to collect data and communicate with other device via wireless protocol. That's why it is known as smart device. Smart TVs, smart watches, medical devices, smartphones, sensors, security system, etc. are some examples of smart devices.

Internet of Things provides two types of communication standards i.e. human to human and thing to thing communication. It follows old client server interaction model for communication standard. Pepper, R., & Garrity, J. (2014) stated that IoT now a days does not attempt a true connection between humans and things, i.e things-to human and human-to-things, for real global computing. Each individual IoT application is established for some of real time situations, such as e-health care, traffic checking etc. This IoT application do not utilize and share other system's data for such kind of services, that prompts an ineffective utilization of the services offered by the IoT applications. This could be accomplished with the assistance of Social Internet of things by utilizing information from a few IoT applications.

Social Internet Of Things is a paradigm of "social network of intelligent objects", centered on the perception of social relationships among objects. First important point is that SIoT exploits and create social relationships among things, instead of just among humans or owners. Second, resources and services can be discovered by things on their own through social relationships to the IoT. SIoT provides distributed solution thereby reducing efforts of human. Third notable point is that SIoT does not only depend on on Web technologies, instead it is a platform for social networking services (SNS). SIoT deals with objects rather than only dealing with humans.

The organization of this chapter is as follows. Second section of this chapter is devoted to related work. This section maps slowly towards SIoT from IoT. The third section discusses existing literature review in more detail. The fourth section focuses on SIoT architecture and processes whereas fifth section is devoted to SIoT applications followed by challenges, and the last section presents the conclusion.

RELATED WORK

IoT requires technology that provides information sharing between smart devices without any human intervention. With this advancement of Internet of things, non-human objects i.e. smart devices enter into social network and begin social relations with humans. In this manner, it is essential to consider the cooperation of human to human as well as the interconnection of device to device with the goal of sharing data to IoT. Recently, there have been a significant number of research exercises that examine the possibilities of coordinating social network communication ideas into IoT environment. This section shows mapping of Internet of things to Social internet of things. Social infrastructure is now converging with the Internet of Things, which simply creates a new phenomenon known as Social IoT, where IoT meets social infrastructure.

A simple example of SIoT is household garbage pickup van. As per our daily routine, garbage pickup van has a specific route with specific time to collect garbage. But when, amount of garbage is different from its usual quantity, the whole schedule gets disturbed. In order to make things easy, a sensor can be attached to that container, which can fetch real time data of garbage weight. By using that data more amount of garbage could be collected efficiently. Currently this is being used in Santander (Spain).

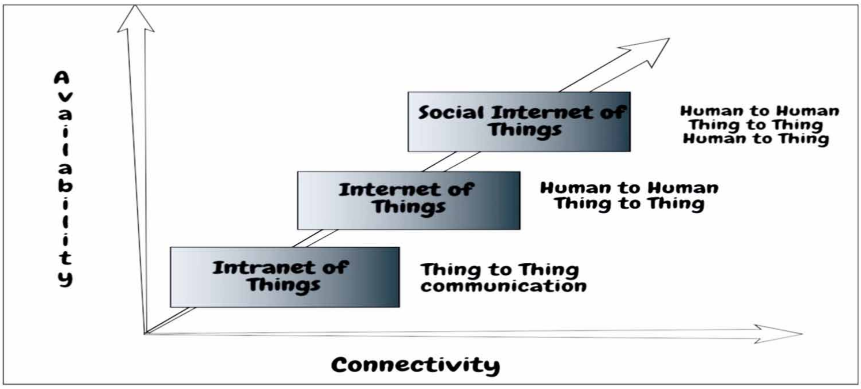
SIoT concept can also be applied with our smart home by transferring some power of one device to other which needs it. It is not necessary that they belongs to same type of object. The relationship between the heterogeneous object can be established to work efficiently with system. Following sub section discusses why SIoT is next subsequent stage for future.

SIoT Is Subsequent Stage of IoT

In future, ubiquitous computing is going to be used in variety of smart applications and services to overcome many issues that individual situation as well as in a group form i.e organizations face in daily lives via allowing things and humans to be connected with either anyone or anything, at any time and in any place. Association for Computing Machinery (ACM) stated transformative history of pervasive computing innovation as presented in figure 1, where two kind of considerations for this objective - first is increasing social interactions or connectivity and second is improving pervasiveness or availability.

In Intranet of things, things provide reactive and proactive data thereby, becoming the producer. Humans are the consumer. Hence there is low connectivity and availability. Moving one step further to Internet of things, humans offer reactive data like current location, needs, interests, demographic properties, etc. Things become

Figure 1. Transformative history of pervasive computing innovation



producer as well as consumer giving moderate connectivity and availability. Social IoT helps move us towards ubiquitous computing as things as well as humans both act as producers and consumers here. "The link establishment between humans and things in SIoT would increase availability of both elements. The elements are humans and things. It includes elements as well as promising their transparency. The two earlier mentioned notions of transparency and availability will eventually drive us to the highly pervasive world, as promoted by forthcoming motivated omnipresent computing systems." After applying Social network services to the IoT, SIoT can prompt few advantages as follows.

- The network navigability can be promised with the help of SIoT structure.
- Trustworthiness level can be recognized for leveraging the degree of interaction among things
- IoT related issues can be re-used to address solutions with the help of social networks.

According to Capgemini (2017), SIoT is a community effort where everything is an asset. SIoT offers few more assets which are discussed below:

1. **Decentralized Intelligence and Data:** As things become smarter and social, the data / information and the intelligence would become decentralized.
2. **Self-Operation, Management, and Organization:** Several "mechanisms including self-management, self-organization, self-healing, self-operation, and self-protection capabilities would be a significant part of the SIoT. Automatic system management will be important, as well as autonomic data analysis, and service revelation and creation will add to upgrade the client encounter. Smart

objects would become smarter as they can access data / information and learn from each other."

3. **Customer Experience and Management:** The level of Quality experience and Service flexibility for the customers can improve based on Social relationships and context-awareness in SIoT.
4. **Business Models:** Information sharing, connectivity sharing can be monetized giving rise to emerging business models.
5. **Sharing Economy:** With SIoT, people would no longer need to worry about being disconnected because of poor signal or bandwidth while people are on a trek – staying connected is now simple through this technology.

Social Relationships in SIoT

Based on the advantages of the SIoT presented in the previous section and the possible set of relationships highlighted in this section, Social Internet of Things is most recent extension in the field of IoT. It provides important platform to establish social relationship between interconnected objects. It provides social connection according to their common interest. Such relationship between objects are mentioned below.

In the deployment of any application social object is a central role for it. Let us consider an example for this. Marry who has just reached Canada for the first time without pre-planned journey, will have difficulty finding hotel. She starts her social mobility application to find best option near her current location. On arrival at the airport a new social relationship is established with tourism team, through which she could reach the particular place. The Social Mobility forwards Marry's queries of co-location and social relationship is collected from the available information about transportation services with availability of time and the cost of transportation through Marry's smart device which is connected with direct network or via some other network. Preferred solution is provided to her when she chooses any option from given options.

To make this type of application, each object should be connected with some social relationship to discover other relevant objects from surrounding and establish social relationship among them.

There are five kinds of relationships that govern the social internet of things as follows.

1. **Parental Object Relationship (POR):** This type of relationship is established between the same types of objects which have same generation with same manufacturer.

2. **Co-location Object Relationship (C-LOR):** This type of relationship is established between same type of object or different type of objects which are included in same environment. Good example for this is smart home devices. It has different kind of sensors with different links but they all are work for one common system.
3. **Co-Work Object Relationship (C-WOR):** This kind of relationship is set up between at least two gadgets whose functionalities are consolidated to accomplish a shared objective. This sort of relationship happens between objects that either need to contact each other to accomplish that objective, or should be in nearness of each other. e.g. sensors, Radio Frequency Identification Tag, and so on continuously exist in a similar place.
4. **Social Object Relationship (SOR):** This relationship define when the device is coming in touch with owner. For the business purpose companies are established this type of relationship.
5. **Ownership Object Relationship (OOR):** It is between smart devices with same clients. Good case for this is cell phones, video game and so forth. Literature review will be explored in the next section.

LITERATURE REVIEW

This section shows literature review on SIoT after study of following research papers.

According to Ning et al. (2011), Surveyors of this paper imagine the eventual fate of the Internet as being described by what they name Ubiquitous IoT design, which takes after the social association system (SOF) demonstrate. That work gives a smart outline of the normal IoT arrange structure. It is like a mankind neural system or social organization framework of future internet of things. It does not go for misusing the qualities of the interpersonal organizations into the IoT.

Guinard et al. (2010), found survey on Social internet of things. The reference social organization is an interpersonal organization of people and it is used by things as a foundation for benefit promotion, disclosure, and access.That momentous commitment by one means or another disregards the IoT vision in which the articles ought to collaborate suddenly to offer esteem added administrations to people.

According to Ahmad et al. (2016), The perception of 'describing human behaviour' using big Data in Social Internet of Things for recommending structure design that produces and analyses real time big data. It grabs the issue of understanding by giving response to the users that offer them the chance to increase their behaviour using the alert message taxonomy.

Paul (2016), stated Smartbuddy system for defining human behaviors using big data analytics in social internet of things.The environment provided by wearable devices and smart cities that determine human behaviours as well as human dynamics using big data. It increase the chance of their behaviour using alert message taxonomy.

Jadhav (2016) found that Wireless Home monitoring using social internet of things (SIoT) with secure and authentication. Secure authentication systems of social networking web sites like Facebook which monitors physical home environment of end user, taking into account which user can control the home environment and give home security remotely. It provides trustworthiness and security to control home system

According to Shen et al. (2017), Task-Optimized Group Search for Social Internet of Things is developed. Creating truthful associations among the objects greatly increases the efficiency of node interaction in the social IoT and helps nodes overcome perceptions of uncertainty and risk. It discriminate the normal behaviour in a unfriendly environment from the Malicious behaviour.

Atzori et al. (2017) found that A SIoT-aware approach to the resource management issue in mobile crowdsensing. Ganti, R. K., Ye, F., & Lei, H. (2011) also found crowd sensing application in SIoT. A new algorithm to address the resource management issue so that Mobile crowd sensing maximize the lifetime of the task group. Focus is on encompassing the resource allocation algorithm to other types of heterogeneous resources.

Atzori et al. (2011) mapped social structure to the internet of things towards Social Internet of Things. A level of trustfulness is permitted to navigate the communication among the billions of object. It provides crowd management using level of trustfulness.

According to Nitti et al. (2016), author explored features of cognitive radio in social internet of things. Sensing and computing individual mobile devices that collectively share data and extract information to measure and map phenomena of common interest. It satisfied unify architecture and envision the requirements.

Campagna et al. (2015) developed ThingTalk application. This application tried to work in Distributed Language for a Social Internet of Things. The possibility to implement a distributed approach for a low-complexity cooperation and the scalability feature in heterogeneous networks. Wireless services will be provided by heterogeneous networks, and thus the problem of spectrum scarcity will be severer.

Bernal Bernabe et al. (2017) found and developed architecture of a social IoT framework. Trust, transparency and user control kinds of quality parameters are achieved with this project and it also gain confidence of everyday users and developers at heart of the system. It has drawn to a close, the components and concepts are being exploited and amended in other labors with the intent to further advance the conception of a convivial Internet of Things.

According to Miori and Russo (2017) improved life quality for the elderly through the Social Internet of Things (SIoT) in Global Internet of Things Summit (GIoTS). The Elderly Monitoring accommodation system describes SOCIALIZE platform with Internet of things module, whose aim is to gather physical and environmental utilizer data. Because of this, physical and environmental data can be supervised by caregiver and medical staffs. The system has been designed to enable facile additament and/or supersession of incipient accommodations and contrivances within the environment. It represents only a commencement point for an incipient IoT vision, wherein Humans as well as things can be connected through Social Networking.

Kowshalya et al. (2017) found a trust management for reliable decision making among social objects in the Social Internet of Things. A trust management scheme is offered to facilitate automatic trustworthy decision. This trustworthy decision creates predication appearance of the objects. With the help of Social IoT trust metrics, It achieves quality of metrics like direct trust, cooperativeness, centrality, community interest, accommodation Score. This quality of service computes trustworthiness among objects. It shows the advantages of the proposed system with subsisting trust management schemes in the literature. The selective forwarding attacks are identified using trust and periodic trust updates.

Observations

Based on literature review and the survey papers explored on Social Internet of Things, it is observed that SIoT can be used in processing of Big Data, crowd sensing and measuring human behavior. It is also useful for security and trust management. Moreover, different architecture, framework and SIoT applications were also included in literature review.

ARCHITECTURE AND PROCESSES OF SIoT

Architecture of SIoT

The architecture composed of:

1. **Sensing Layer:** It is dedicated to node collaboration and data acquisition in local and short range networks
2. **Network Layer:** It is dedicated for transporting data across discrete networks
3. **Application Layer:** It is responsible for deployment of IoT applications with the middleware functionalities

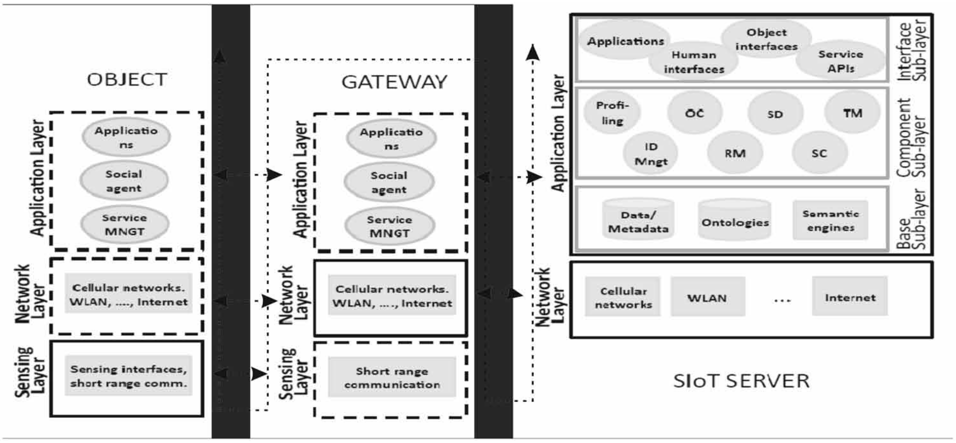
Architecture of Social Internet of Things in figure 2 exhibits the Architecture of Social Internet of Things. Social IoT Server, the Gateway, and the Object are three basic elements of SIoT architecture.

SIoT Server

SIoT Server comprises of two layers: one of them is Network layer and another one is the Application Layer. Application Layer comprises of three sublayers. The Base Sublayer comprises the database for management and storage of the data. These record the social member profiles and their relationships, also because the activities dispensed by the objects within the real and virtual worlds. Information concerning humans are also managed. Humans could also be object house owners or guests. The relevant ontologies area unit hold on in an exceedingly separate info and area unit wont to denote a semantic read of the social activities. Semantic engines extracts such kind of read through applicable. Indeed, ontology and semantic services area unit necessary to produce a machine explainable framework. This machine explainable framework is used for representing purposeful and non-functional attributes and operations of the IoT devices. During this context, many works are already shown, that can be a place to begin for the definition of an ontology to be employed in the SIoT system.

One answer is to implement the "Ontology web Language for Services (OWLs) model that gives meaningful descriptions and well-defined semantics. This has already been used because the basis of a semantic service modeling framework for the IoT. During this framework, services area unit used as associate interface that represents the IoT resources. IoT resources may be any kind of physical world devices. It supply

Figure 2. Architecture of social internet of things



associate access to the functions and capabilities of those resources. An ontology is taken into account as a basic attribute of the IoT with the role of supporting the agent who reads associate electronic tag to grasp the data in it. Agent in the OWL-S can be either man or machine. Ontologies to manage and management heterogeneous systems are investigated.

Here the people foresee that while not ontological classification associate degreed semantic annotation processes an automatic discovery are not possible. According to Mika, the importance of the ontology has been analyzed from a social network perspective as a format to represent the thing info that has relevancy to end users. There are a variety of different approaches for making semantic service descriptions. It includes Unified Service Description Language (USDL), "Semantic Annotations for WSDL (SAWSDL)," Service Modelling ontology (WSMO), "Web Service Modelling Language (WSML)," Web and semantic Annotations for representational State Transfer SA-REST. All these models are good bases that may be exploited to explain the social objects in our model. During this context, it's conjointly price mentioning the Friend-of-a-Friend project. One of the "Friend-of-a-Friend project is FOAF- WWW.foaf-project.org," that is aimed toward making an online of machine-readable pages describing humans, the links between them, and also the things they produce.

The results of this project are of specific interest for the outline of the objects' social links. Architecture of Social Internet of Things in figure 2 presents the component Sub-layer includes the tools that implement the core practicality of the Social IoT system. The ID management is geared toward distribution associate degree ID that universally identifies all the potential classes of objects. The identification is geared toward configuring manually and accordingly data i.e. static or dynamic data regarding the objects. The owner management (OC) is that the module that validate definition of the activities which will be performed by the article, the data which will be shared, similarly because the style of relationships which will be originated. In owner management, the set of objects which might access such information. The "relationship management (RM)" could be a key module within the network since the objects haven't the intelligence of humans in choosing the friendships; so, this intelligence has to be incorporated into the SIoT. Main task of this part is to permit objects to start out, update, and terminate their relationships with alternative objects. To perform all the tasks premise of the owner's management settings is required.

The "service discovery (SD)" could be a elementary element, that is aimed toward finding that objects will offer the specified service within the same approach humans explore for friendships and for any data within the social networking services. The service composition (SC) element permits the interaction between objects. Most of the time, the interaction is said to associate object that needs either to retrieve associate data concerning the important world or to seek out a selected service provided by another object. In fact, the most potential people have a tendency to see in deploying SIoT is its capability to foster such associate data retrieval. Investing on the item relationships, the service discovery procedure finds the required service, that is then activated by suggests that of this element. Last however not least, the trait management (TM) element is aimed toward understanding however the data provided by the opposite members shall be processed. Dependability is constructed on the idea of the behaviour of the item and is strictly associated with the link management module. Trustworthiness may be calculable by applying notions well-known within the literature, like spatial relation and status, that critical within the study of the social networks. The third sub-layer, that's the Interface Sub-layer, is wherever the third part interfaces to things, humans, and services are set. This sub-layer could also be mapped onto one website, deployed in a very federate approach by different sites, or deployed in a very cloud. Herein, author have a tendency to don't seem to be proposing any specific implementation resolution.

Gateway and Objects

As to the gateway and Objects systems, the mixture of layers might vary mainly counting on the device characteristics. The subsequent 3 situations are often expected. In a very easy one, a dummy Object that is equipped with a practicality of the bottom layer. A dummy object may be either a RFID tag or a presence sensing device. It is simply enabled to send easy signals to a different component i.e. the Gateway. The Gateway is provided with the entire set of functionalities of the said three layers.

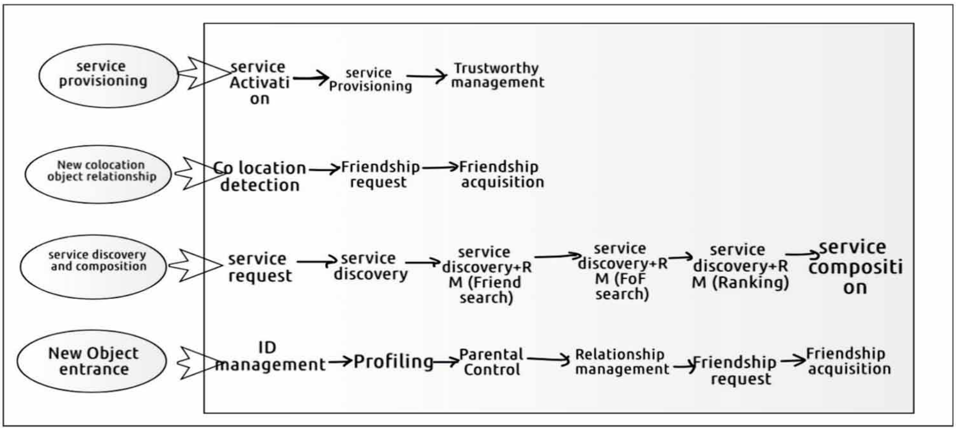
In another state of affairs, a device (e.g., a video camera) is in a position to sense the physical world data associated to send the connected knowledge over an IP network. The thing would then be set with the functionality of the Network Layer aside from that of the applying one. Consequently, there is no would like for a gateway with Application Layer functionality. An Application Layer in a very server, somewhere within the Internet, with the gateway application layer functionality would be enough. In next third state of affairs, a smart object (e.g., a smartphone) might implement the practicality of the three layers in order that the entryway is not required, except for some communication facilities targeted to take care of the net property of the thing. This can be the case of a smartphone. It has enough procedure power to perform all the three-layer operations which might have a entryway for omnipresent network property.

Whatever the situation enforced, the Application Layer encompasses the SIoT applications, still because the social agent and also the service management agent, that are conferred below. The social agent is dedicated to the communication with the SIoT servers to get and request services, to update its profile and to update friendships from the social network. It additionally implements the strategies to speak directly with different objects after they are geographically shut or once the service composition wants direct communications between objects. Finally, the service management agent is to blame for the interfaces with the humans which will management the behaviour of the thing once human action inside their social network.

SIoT Processes

Component Sub-layer contains main elements of the SIoT architecture. The sensing and networking layers are not considered by SIoT as a solution for in Internet of things, yet, to make the universe of trillions of things sensible when confronting the issue of service and data revelation. Furthermore, it goes for laying the ground for independent connections among objects for the advantage of the human client. This type of interaction is done mainly through service discovery and composition. Social Internet of Things processes in figure 3 presents interactions between the SIoT architectural elements. The four main SIoT activities are provided to understand overview of the processes. The different activities are entrance of a new object, new object relationship establishment, service discovery, service composition, and service provisioning. The label of Social Internet of Things processes in figure 3 represents the tasks involved in the analyzed activity. The parental control, profiling, account creation tasks are done by analyzed activity.

Figure 3. Social internet of things processes



"Two elements that communicate to carry out the task are identified by the associated label i2j (i; j = H, S, A, O, which stand for human, SIoT server, object agent, and object, respectively). Take note that the Gateway is not mentioned here, even though it may take part in these processes when the agent is involved. This is because, in this context, the agent is defined as the software entity that implements the application functionalities of either the Object or the Gateway."

The main architectural components in figure 3 are cited on top of the task blocks. The architectural components involve in carrying out relevant operations. Object owner performs the relevant operations like the entrance of a new object into the system, the relevant activities. Object owner communicates with the servers to perform tasks namely create the account, set the control parameters though the ID management, insert the object profile data, and object profiling components. The ID scheme should be interoperable with the fundamental identification schemes as of now being used here, for instance: IPv6 addresses, Ubiquitous code (Ucode), Universal Product Code (UPC), OpenID, Electronic Product Code (EPC), URI (Uniform Resource Identifier). The capabilities and relevant history of object through ID are added by the profiling. Due to the heterogeneity of the IoT nodes, SIoT needs to arrange SIoT members in different classes. Each class is defined on the basis of the main object features.

Class1: The large computational and communication capabilities of mobile objects are assigned to class 1. Smartphones, sensors, tablets, actuators and vehicle control units are examples of class 1.

Class2: The significant computational and communication capabilities objects are assigned to class 2. This type of objects available in static form. Displays, set top boxes, smart video cameras are examples of class 2.

Class3: The sensing capabilities objects are assigned to class 3. This type of objects are capable of providing the environment status through measurement.

Class4: The RFID- or NFC-tagged objects are assigned to class 4. Each class is then characterized by specific attributes, such as: object category. Object category additional specifies the object typology within its class though owner ID. Object position, which can be changing over the time depending on the object mobility features. Power supply status defines whether the object is either battery-powered and the battery power level is provided or not. It also provides information like socket connected or not, whether is currently connected or not, or gathers power from the environment; amount of traffic generated in terms of number of connections and overall bit-rate.

After the completion of object profiling procedure, the agent completes the process by looking for friends in Social IoT servers. The agent might run either on the question itself or in a different framework, contingent upon the protest qualities and characteristics of object.

The main relationships are established by object in this object profiling phase. Main relationships are triggered by its object profile too. It also includes relationships like parental object and ownership object relationships. During the entire lifetime of the objects, the other relationships are established later by object profiling. These relationships are mainly depend on the objects' movements, object service and object information exchanges over the SIoT.

"The application trigger the service discovery and composition phase. The applications are running either on the SIoT servers or in close relationship with the agent. These relationships run either in the gateway or in the object. The way in which this process is performed depends on the type of service that the application is looking for. The provisioning of surrounding environment information, the status of an object, the activities carried out by the object owner, besides the activation of a specific action from another object. The process continues with one of the most innovative and crucial task when the service request has been triggered by the application of designed system. The difficult task may be the serving friend search. This is related to the procedure of looking at the "friend" object's profiles to see whether the required service is provided by one or more of them."

In this procedure, the distinctive sorts of connections have not a similar pertinence to any application. For example, if the asked for benefit is of a "best work on sharing" sort, at that point the parental relationship is the most imperative. Actually, a similar issue has most likely been tended to in the past by objects having a place with a similar generation group. On account of need of a data about the encompassing condition, co-location and co-work connections are those that ought to be abused. Indeed, the comparing companions are those that most likely had event to secure a data on the environment. In the event that a companion ready to give the service has not been discovered, at that point, the diagram of fellowships is additionally slithered. Since in excess of a solitary administration might be discovered, a positioning is required. The positioning can be executed by various principles, among which: the serving object dependability, the credit/charge relationship of the cooperating objects, the protest assets (in terms of residual battery power, bandwidth, communication and computing reliability, and so on). A few approaches can be adopted in this context to rank the potential service providers.

"At the point when the described activity ends, the service composition is activated, which comprises of interfacing the requesting service with the required one." Note that the service composition includes A2A communications, while A2S communications are required during other tasks. In the new object relationship activity, two objects become aware that they are neighbors for a period of time long enough to trigger friendship. In this scenario, while referring to the co-location, co-work, and social relationships, which are activated in case of geographical proximity of the objects. The detection of this event is enabled by the use of short range communication facilities that allow two objects to detect that they are within communication range of each other. The short communications facilities e.g., NFC, Bluetooth, or ZigBee interfaces. The short communications are other possibilities to detect this event. One is the use of the localization facilities already available within the objects to track their position over the time. The localization facilities are i.e WiFi / Bluetooth triangulation, INS systems, or even GPS. An object can detect the co-location relationship with other objects during an upload of location information into its profile. The profile is available in the Social IoT server.

Co-location event is detected by object in whatever way, the object agent then requests the friendship, which may be accepted according to the owner control rules.

The service provisioning process consists in delivering the service previously discovered and composed with the requesting service. For better understanding, let us consider the scenario of a smartphone that is looking for information about radio signal coverage in the areas surrounding its current position. To accomplish its target, the smartphone drives a service discovery and composition process to look for smartphones and personal computers that have already visited the areas of interest (and are then aware of the signal strength). Once the service has been composed, the requesting agent (installed in the smartphone itself) communicates with the agents providing the relevant information to activate the service. All the way through, the service requesting agent is able to extract important information about the trustworthiness of objects that provide the services. This information is uploaded to the SIoT server and thus is available to the whole community. The next section describes SIoT applications and how they address the aforementioned challenges.

SIoT APPLICATIONS

In this era, technology has become an integral. People can't live without the social relation with technology. Below are few examples of such applications: "

"According to Haesung Lee and Joonhee Kwon (2015), John's house is placed in the block which is constructed according to modern eco-amicable principles. In the cube, each home is fitted with a keenly intellective meter and sensors in order to manage and measure energy consumption during the solid day. By designates of the gregarious IoT network, the astute meter is able to exchange info on the energy usage with reference to identification of the energy suppliers that best match the household needs, identification of the household appliances. A light in any house changes the color according to the energy saving level obtained by its owner. Relationships of the social IoT are exploited in this scenario. Another examples stated by Haesung Lee and Joonhee Kwon (2015) are as follows. Jessie is a sales representative who frequently travels by car around the city to meet her customers. Unfortunately, the traffic has increased in recent times making her driving more and more problematic. However, by exploiting the social IoT network, her car is able to gather information in real-time about traffic congestion along possible routes and to choose the best path to get to the meeting in the scheduled time. Finding the appropriate source of interesting information in the social IoT network is easy for the car by the pervasive IoT interconnection technique proposed in our study." Hannah has just bought a new notebook. In the initial phase, she has a difficult time to join to some network equipments such as printers or faxes. By exploiting the relationship of other devices belonging to the social IoT network with our proposed interconnection technique, Hannah's notebook can find a mate of the device that has already addressed the same configuration issues and fix the problems."

According to Butani, K.V, B.P & S.B (2018), If people migrate from one place to another then some communication channel may block in other countries. Some time people have to adapt that country's technology without their choice because people have to communicate with other or complete their tasks. So the magical solution of this problem is SIoT. It gives ability to control everything as per their choice with efficient ways with solutions even when the place or locations are not static. As per our general life people complete their many needs by using mobile phones. People can store many thing in smartphones. Social Internet of Things allows them to connect their device with smart cities, transportation vehicles, smart home and many more. By using SIoT, people easily communicate with different devices which are connected with us.

In home there are many appliances present like oven, blender, refrigerator, electric kettle, mixture, smoke detector etc. Refrigerator will store many groceries. When people go for shopping then people have to check each and every item with quantity so people able to analyze that what things people have to purchase. If the concept of SIoT is applied to refrigerator, than it will tell them about what groceries people need and which the present groceries with its quantity are. Even it will order groceries from your favorite store or nearby store.

If anyone is alone at home and some accident happens, then automatically their neighbors or other family members get the message of that via email, text or any social media. This thing is also possible via SIoT.

It is useful for factories. There are many machines which are working continuously. The people have to regularly provide servicing to that machines. By using Social Internet of Things, when machines are in condition that service is required then they automatically send message to the service center for the services. So maintenance issue is solved. This concept can be applied in many devices which are used in home, hospitals, transportation, agriculture, shopping centers and many more.

In shopping malls or hospitals there are lifts or escalator. If any problem occurs in that then in minimum time people have to find the solution for that. Apply Social IoT in this then, that lifts or escalator may automatically send the message to the nearest or available engineering team to solve the problem.

There is no country which have no any risks. Natural disasters, terror attacks, virus outbreaks etc. may occur any time. At times, people have no time to deliver message to each and every person. Here people can use Social IoT. It will identify the risk and send message to everyone using social media, TV channels, radio, emails etc. It is very much helpful to save the life of many people.

SIoT CHALLENGES

The major challenge for Social IoT is overabundance of activities, and ample of connections to create due to countless number of social media sites. IoT when combined with social media needs to generate constructive and quantifiable outcomes. Social IoT is beneficial only if technologies and tools are developed and designed to overcome challenges faced by Social IoT. If these kinds of technologies are implemented successfully than businesses would be easily able to deliver personalized services.

Another challenged faced is handling diverse endpoints, pattern, and styles for Social IoT data so as to find out feasible solutions. Considering smart cities one of the domain of Social IoT, it can face number of problems related to demographic, technological, social etc. factors.

Along with above mentions problems many other concerns such as trust, security and privacy are very perceptive challenges for IoT. Misuse of information and fraud will be the aftereffect if to access different IoT application an unsecured technology is used. Therefore trustworthiness and privacy of user data will play an important role in success of Social IoT.

CONCLUSION

This chapter has introduced the fundamental role of SIoT, based on the integration of environment of IoT with Social Networking. In real time Application situations, smart objects segment preeminent practices. For example, ACs in a similar neighborhood can set up social connections that can be utilized to discover answers for normal setting issues. Such issues are identified with the power or battery problems. Similarly, vehicles of same company, year and model can provide information about same electrical/mechanical problems. In different circumstances, smart devices that visit the same topographical zone can set up connections to share helpful data on the physical world. Thus, another idea of Social IoT is depicted as a universal cooperation paradigm for the data partaking in the Web of devices.

The Social Internet of Things (SIoT) can be referred to combination of social network that is people with smart devices. The main motivating factor for SIoT is to integrate smart devices into day to day activities of humans. In this chapter authors have inspected the recently emerged domain of the SIoT by reviewing social network with platform of IoT. Author also discussed architecture, relationships of SIoT and processes of SIoT in this chapter. This kind of combination will give rise to features of social networking like interactivity, recommendations, filteration etc. of users with their smart devices. This type of integration which allows user interaction with devices is the core of this chapter. Along with this they have listed some of the challenges faced while implementing Social IoT in the real world. The author will consider these challenges as future work.

REFERENCES

Atzori, L., Girau, R., Martis, S., Pilloni, V., & Uras, M. (2017, March). A SIoT-aware approach to the resource management issue in mobile crowd sensing. In *Innovations in Clouds, Internet and Networks (ICIN), 2017 20th Conference on* (pp. 232-237). IEEE.

Atzori, L., Iera, A., & Morabito, G. (2011). Social Internet of Things: Giving a social structure to the internet of things. *IEEE Communications Letters*, *15*(11), 1193–1195. doi:10.1109/LCOMM.2011.090911.111340

Atzori, L., Iera, A., Morabito, G., & Nitti, M. (2012). The social internet of things (siot)–when social networks meet the internet of things: Concept, architecture and network characterization. *Computer Networks*, *56*(16), 3594–3608. doi:10.1016/j.comnet.2012.07.010

Bernal Bernabé, J., Elicegui Maestro, I., Gandrille, E., Gligoric, N., Gluhak, A., Hennebert, C., & Nati, M. (2017). *SocIoTal-The development and architecture of a social IoT framework*. Academic Press.

Butani, K.V, B.P., & S.B. (2018). Social IOT: Network of Smart Things with Social Connections. *Journal of Engineering and Technology, 11*(2018), 60-63.

Campagna, G., Seo, J., Fischer, M., & Lam, M. S. (2015). Thing Talk: A Distributed Language for a Social Internet of Things. *Work (Reading, Mass.)*.

Ganti, R. K., Ye, F., & Lei, H. (2011). Mobile crowd sensing: Current state and future challenges. *IEEE Communications Magazine*, *49*(11), 32–39. doi:10.1109/MCOM.2011.6069707

Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, *29*(7), 1645–1660. doi:10.1016/j.future.2013.01.010

Guinard, D., Fischer, M., & Trifa, V. (2010, March). Sharing using social networks in a composable Web of Things. In PerCom Workshops (pp. 702-707). Academic Press. doi:10.1109/PERCOMW.2010.5470524

Jara, A. J., Bocchi, Y., & Genoud, D. (2014, September). Social Internet of Things: The potential of the Internet of Things for defining human behaviours. In *Intelligent Networking and Collaborative Systems (INCoS), 2014 International Conference on* (pp. 581-585). IEEE.

Kowshalya, A. M., & Valarmathi, M. L. (2017). Trust management for reliable decision making among social objects in the Social Internet of Things. *IET Networks*, *6*(4), 75–80. doi:10.1049/iet-net.2017.0021

Lee, H., & Kwon, J. (2015, November). Survey and Analysis of Information Sharing in Social IoT. In *Disaster Recovery and Business Continuity (DRBC), 2015 8th International Conference on* (pp. 15-18). IEEE. 10.1109/DRBC.2015.13

Mika, P. (2004, September). Social networks and the semantic web. In *Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence* (pp. 285-291). IEEE Computer Society.

Miori, V., & Russo, D. (2017, June). Improving life quality for the elderly through the Social Internet of Things (SIoT). In *Global Internet of Things Summit (GIoTS), 2017* (pp. 1–6). IEEE. doi:10.1109/GIOTS.2017.8016215

Ning, H., & Wang, Z. (2011). Future internet of things architecture: Like mankind neural system or social organization framework? *IEEE Communications Letters*, *15*(4), 461–463. doi:10.1109/LCOMM.2011.022411.110120

Nitti, M., Murroni, M., Fadda, M., & Atzori, L. (2016). Exploiting social internet of things features in cognitive radio. *IEEE Access: Practical Innovations, Open Solutions*, *4*, 9204–9212. doi:10.1109/ACCESS.2016.2645979

Ortiz, A. M., Hussein, D., Park, S., Han, S. N., & Crespi, N. (2014). The cluster between internet of things and social networks: Review and research challenges. *IEEE Internet of Things Journal*, *1*(3), 206–215. doi:10.1109/JIOT.2014.2318835

Chapter 11
Smart Pollution Alert System Using Machine Learning

P. Chitra
Thiagarajar College of Engineering, India

S. Abirami
Thiagarajar College of Engineering, India

ABSTRACT

This chapter proposes a novel mobile-based pollution alert system. The level of the pollutants is available in the air quality repository. This data is updated periodically by collecting the information from the sensors placed at the monitoring stations of different regions. A model using artificial neural network (ANN) is proposed to predict the AQI values based on the present and previous values of the pollutants. The ANN model processes the normalized data and predicts whether the region is hazardous or not. A novel mobile application which could be used by the user to know about the present and future pollution level could be developed using a progressive web application development environment. This mobile application uses the location information of the user and helps the user to predict the hazardous level of the pollutants in that particular location.

DOI: 10.4018/978-1-5225-7790-4.ch011

INTRODUCTION

The introduction of particulates, harmful gases and other biological molecules into the Earth's atmosphere causes air pollution, which leads to disease, death and damage to humans, damage and other living organisms such as food crops. These particles and gases that pollute the atmosphere are known as air pollutant. The air pollutant causes adverse effects on human beings and their ecosystem. Sources of air pollutants are generally either anthropogenic or natural.

Air pollutants are classified as primary or secondary. A primary pollutant is emitted directly from a source, e.g. Sulfur Dioxide (SO2), Carbon Monoxide (CO), Nitrogen Oxides (NOX), and particulate matter (PM). A secondary pollutant is not directly emitted as such, but forms when other pollutants (primary pollutants) react in the atmosphere. An example of a secondary pollutant is Ozone. When hydrocarbons are emitted and they react with NO_x in presence of sunlight, they form ozone.

The secondary pollutant Ozone has critical effects on human health such as permanent lung damage, aggravated asthma, or other respiratory illnesses. Above certain limits they also cause damage to plants, reduce the crop yield, and also increase of vegetation vulnerability to diseases.

Particulate matter (PM) is the mixture of all solid and liquid particles suspended in air, many of which are hazardous. Both organic and inorganic particles, such as dust, pollen, soot, smoke, and liquid droplets are present in this mixture. Some of them such as dust, dirt, soot, or smoke, are large enough to be seen with the naked eye while the rest are so small that they can only be detected only using an electron microscope. Fine particulate matter ($PM_{2.5}$) consisting of particles with diameter 2.5 mm or smaller, is an important pollutant among others as they are capable of penetrating deeply into the lungs and cause health problems, including the decrease of lung function, development of chronic bronchitis and nonfatal heart attacks (EPA, 2005).

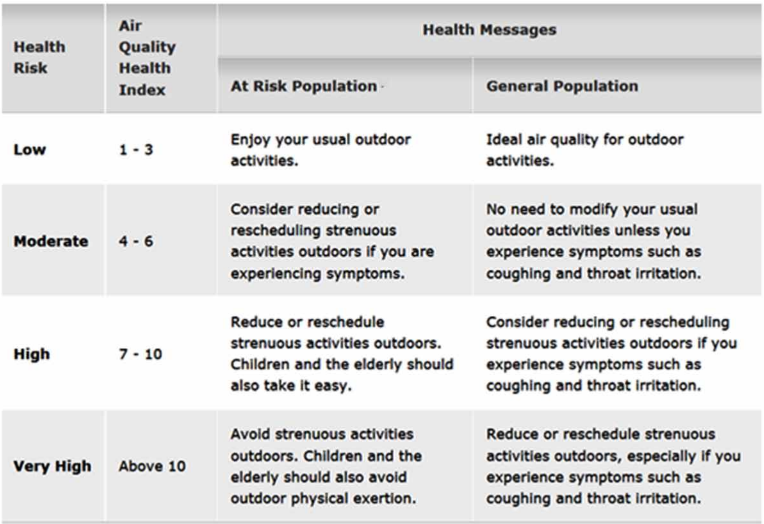
Sulphur dioxide is another eminent air pollutant whose source is from fossil combustion of industries and locomotives. Its effects on human beings include damage of respiratory system, particularly lung function, and can irritate the eyes. It mainly causes respiratory tract inflammations along with coughing, mucus secretion and also aggravates conditions such as asthma and chronic bronchitis. Also, wet deposition of it is acidic and causes acid rain that contains sulfuric acid. This badly affects the ecosystem by changing the nutrient balance in water and soil (EPA, 2005).

Nitrogen dioxide, another important air pollutant is part of a group of gaseous air pollutants produced as a result of road traffic and other fossil fuel combustion processes. Globally, it contributes to global warming and is the third most important greenhouse gas in the UK. Nitric Oxide (NO_x) gases react to form acid rain and smog and also contribute to the formation of fine particles (PM) and ground level ozone. All these, in turn affect human beings with their associated adverse health effects (EPA, 2005).

Periodic air quality evaluation could be the best way to monitor and control air pollution. The suitability of air for lives on earth depends upon its characteristics. The Air Quality Health Index (AQHI) is a health protection tool designed in Canada that helps to understand the impact of air quality on human health. As shown in Figure 1 it provides a number from 1 to 10+ to indicate the level of health risk associated with local air quality. As the number increases it indicates greater the health risk and suggests the needed precautions to be taken. The index describes the level of health risk associated with this number as 'low', 'moderate', 'high' or 'very high', and suggests steps that can be taken to reduce exposure. Building a forecast system to predict hourly average concentrations of the pollutants and thereby its AQHI would be an efficient system to protect the people especially the vulnerable groups on a daily basis from the negative shades of air pollution.

The three main factors that mainly influence the concentration of air pollution at a particular location are meteorological factors, the source of pollutants and the local topography of that location. Many air quality forecasting uses straightforward approaches like box models, Gaussian models and linear statistical models. Though, these models are easy to implement and allow for the fast calculation of forecasts, they fail to describe the interactions and non-linear relationship that handle the transportation and behavior of pollutants in the atmosphere.

Figure 1. Air quality health index table



Health Risk	Air Quality Health Index	Health Messages	
		At Risk Population	General Population
Low	1 - 3	Enjoy your usual outdoor activities.	Ideal air quality for outdoor activities.
Moderate	4 - 6	Consider reducing or rescheduling strenuous activities outdoors if you are experiencing symptoms.	No need to modify your usual outdoor activities unless you experience symptoms such as coughing and throat irritation.
High	7 - 10	Reduce or reschedule strenuous activities outdoors. Children and the elderly should also take it easy.	Consider reducing or rescheduling strenuous activities outdoors if you experience symptoms such as coughing and throat irritation.
Very High	Above 10	Avoid strenuous activities outdoors. Children and the elderly should also avoid outdoor physical exertion.	Reduce or reschedule strenuous activities outdoors, especially if you experience symptoms such as coughing and throat irritation.

The knowledge discovery and their interpretations from the huge amount of past air pollutant concentrations data and meteorological data seemed vital in the process of forecasting air quality. Machine learning that originated from the field of artificial intelligence has become popular in solving it. A large number of neural networks are used for forecasting air quality and are also found to be more advantageous than the statistical methods. However, some of their difficulties include computational expense, multiple local minima during optimization, over fitting to noise in the data, etc. Furthermore, there are no general rules to determine the optimal size of network and learning parameters, which greatly affects the prediction performance.

Model updating is another key feature in air quality forecasting that updates and refines the model along with everyday forecasting using their latest observations. The two ways for model updating are batch learning and online learning. In batch learning, whenever new data are received, it uses the past data together with the new data and performs a retraining of the model. Therefore, batch training is computationally expensive. Online learning uses only the new data to revise the model. Generally linear models easily update with batch learning or online learning. But, for non-linear methods, online learning is difficult for many formulations such as the non-linear kernel method. Also, short time update implementation using batch learning is too expensive as a non-linear model tends to have more parameters to train and the training process is much slower compared to linear models. Hence, developing non-linear updatable models for real-time air quality forecasting is remaining essential.

Machine Learning Algorithms

Machine learning is a sub-field of Artificial Intelligence. Machine learning algorithms analyze input data to predict output values within an acceptable range. As new data is fed to these algorithms, they learn and optimize their operations to improve performance, developing intelligence over time. There are four types of machine learning algorithms: supervised, semi-supervised, unsupervised and reinforcement.

Supervised Learning

In this the model is trained using previous examples. A known dataset that includes desired inputs and outputs is fed as input and the job of the algorithm is to find a method to determine how to arrive at those inputs and outputs. In this process, the algorithm identifies patterns in data, learns from observations and makes predictions. Finally, the algorithm makes predictions and is corrected by its previous knowledge and this process continues until the algorithm achieves a high level of accuracy.

Some of the common tasks followed using supervised learning is Classification, Regression and Forecasting.

- **Classification:** In this the algorithms gains knowledge from the past database and tries to determine under what category the new observations belong to. For example, when detecting credit card fraud as 'fraud' or 'safe', the program must look at existing observational data and classify the new data accordingly.
- **Regression:** In this the algorithm must infer and understand the relationships among the different variables. The regression analysis is particularly useful for forecasting/ prediction of the dependent variable using the others.

Semi-Supervised Learning

It is similar to supervised learning, but the input to the algorithm has both labeled and unlabeled data. Labeled data is essentially carries information that has meaningful tags so that the algorithm can understand the data, while unlabelled data lacks them. Using the labeled data machine learning algorithms can learn to label unlabeled data.

Unsupervised Learning

In this method of training the algorithm identify patterns from the input data. There is no meaningful information provided to the algorithm. Instead, the machine determines the correlations and relationships by analyzing available data. Through this sort of training the machine learning algorithm is left to interpret large data sets and address that data accordingly. The algorithm keeps categorizing that data into clusters in some way to describe its structure and makes it look more organized. As it learns pattern by assessing more data it gradually improves its ability to make decisions on the data. One importank task that can be performed using unsupervised learning is clustering. Clustering refers to grouping sets of similar data based on defined criteria. It for segments data into several groups and performs analysis on each data set to find patterns.

Reinforcement Learning

In reinforcement learning processes, the machine learning algorithm is provided with a set of actions, parameters and end values. By using regimented learning process it defines the rules and tries to explore different options and possibilities. It monitors and evaluates each result and determines which one is optimal based on trial and error. The algorithm is from past experiences and begins to adapt its responses to the situation in such a way to achieve the best possible result.

Online Machine Learning Algorithm

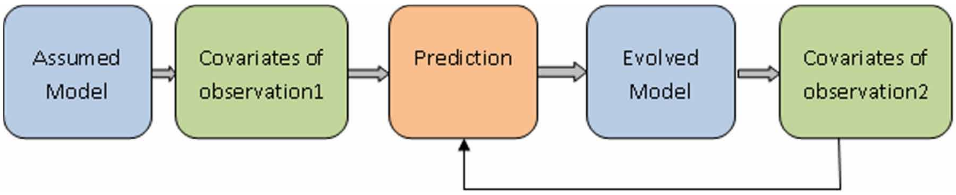
An up to date model can be obtained only if it can learn from new examples in something close to real time. Online machine learning algorithms (Figure 2) keep updating the model for every new observation made now and then. Online learning is data efficient and adaptable. It is not only fast but also has the capability to capture any new trend visible in with time.

Air Pollution Alert System

Good air quality is essential for existence and survival of life on earth. With huge developments in urbanization our ecosystem is highly endangered to pollution. Many diseases and allergies affect all life on earth. Acid rain and global warming are other effects of air pollution that makes the earth an unsuitable place to live in. Proper forecasting of air quality can help people reduce their exposure on risky days and also support the law makers to set rules accordingly to counter the increasing pollution. Hence the need for an efficient and accurate air pollution alert system remains imperative in today's world.

Various machine learning algorithms are used till date for extracting knowledge from past pollutant concentration and meteorological databases that are available in large quantity for various locations. Apart from pollutant concentration and meteorological data many researches prove that the forecasting values also depend upon the location's topography, traffic, peaks etc. This knowledge discovery using machine learning (Figure 3) helps to obtain valuable patterns in the values of air quality with respect to various parameters. These patterns in turn are efficiently utilized to predict the future air quality values which can be used to alert the people when air quality value is expected to go down. This alert system mainly helps the vulnerable group of people suffering from various diseases and allergies to reduce

Figure 2. Online Machine Learning Algorithm



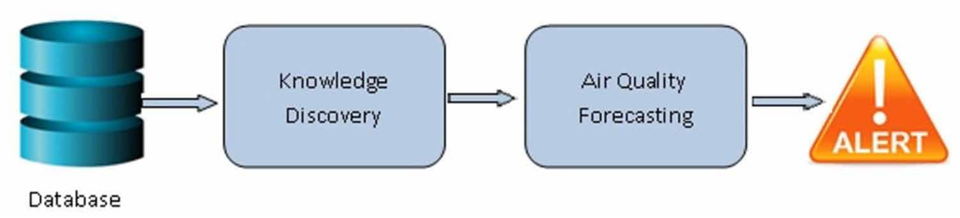
their exposure so that they can avoid the risk of their health on those risky days. Children, older adults and those who are suffering from lung and/or heart disease are especially vulnerable to the adverse effects of air pollutants and should take special precautions when the ambient air quality crosses unhealthy levels. High AQHI readings (7-10) can cause increased asthma symptoms, such as coughing, wheezing, chest tightness and the need for increased inhaler use. Planning outdoor activities by checking the AQHI could possibly minimize the health risks. Many researches has also proved this by using the clinical database- the number of patients being admitted in hospitals everyday- as one of the parameter in machine learning algorithms for knowledge discovery (Ren et al, 2018). Hence, location specific predicted air quality values and advisories must be made widely available through air pollution alert system to people in order to reduce/avoid exposure on days having unhealthy air quality.

Previous Works

Standard conventional static methods like the typical SVM algorithm is not able to process the huge data that needs frequent and continuous updating (Ghaemi et al, 2018). Once a typical SVM algorithm is trained, it works as the stationary model afterward and when new training samples are available, learning has to restart again. This process is computationally expensive and time-consuming. Online algorithms are regarded as an alternative to the conventional static methods.

A semi-supervised learning approach can be used based on a co-training framework that consists of two separated classifiers (Zheng et al, 2013). One is a spatial classifier based on an artificial neural network (ANN), which takes spatially-related features (e.g., the density of POIs and length of highways) as input to model the spatial correlation between air qualities of different locations. The other is a temporal classifier based on a linear-chain conditional random field (CRF), involving temporally-related features (e.g., traffic and meteorology). The framework consists of two major parts, offline learning and online inference

Figure 3. Air Pollution Alert System



Long short-term memory neural network extended (LSTME) model that inherently considers spatiotemporal correlations for air pollutant concentration prediction (Li et al, 2017). Long short term memory (LSTM) layers can be used to automatically extract inherent useful features from historical air pollutant data, and auxiliary data, including meteorological data and time stamp data. Unlike traditional RNNs, LSTM NNs are capable of learning long time series and are not affected by the vanishing gradient problem. Compared to traditional shallow models such as the SVR, ARMA and TDNN, deep learning-based models exhibit better prediction performance.

Feature selection and spatio-temporal semi supervised learning can be embedded in the input layer and the output layer of the deep learning neural network respectively (Qi et al, 2018) Feature selection and analysis is not to increase the prediction accuracy, but to discover the importance of different input features to the predictions of the neural networks, reveal the main relevant factors to the variation of air quality, and provide a proof from data science to support the air pollution's prevention and control.

Instead of updating the parameters of the prediction network, M-BP algorithm can be used to update the missing values of input data to minimize the prediction loss with the trained prediction neural network (Li et al, 2017). Without the M-BP, the model takes 5 hours to converge on a single Nvidia Titan GPU card, while with the M-BP, it takes 8 hours to converge.

In order to improve the processing speed along with required machine learning functionalities, Apache Spark can be employed on the Hadoop cluster (Asgari et al, 2017). Multinomial Naïve Bayes and Multinomial Logistic Regression algorithms are used for short-term air pollution forecasting. Predictive air quality risk map is generated for the next 24 hours for the whole city using inverse distance weighting (IDW) method.

Forecasting model can also be built using artificial neural network (ANN) (Gorai et al, 2017; Goyal et al, 2015). Two types of learning algorithms, feed forward back propagation (FFBP) and layer recurrent (LR) were used for training the ANN model.

A Neuro-Fuzzy model, the combination of neural network and fuzzy logic methods can also be used for air pollutant concentration prediction which is found to be more efficient than MLR and ANN (Mishra et al, 2016).

Big Data Analytics

The amount of data being processed for air quality prediction is enormous in terms of volume, velocity and variety (Zheng et al, 2013). Hence an organized platform is essential for integrating, dealing, validating and securing this data. Hadoop is one of the most popular framework (Figure 4) for distributed storage and processing of big data. Its capability of managing and analyzing massive amounts of structured and unstructured data quickly, reliably, flexibly and at low-cost is the main reason for its popularity (Chimmiri, 2016). Apache Hadoop has master-slave architecture. The master node manages the cluster state. The slave nodes are responsible for storing data and executing tasks assigned to them. Distributed File System (HDFS) and YARN resource manager are two important advances of Hadoop 2. HDFS is designed to store large amount of data across multiple available machines in redundant fashion. YARN is manages the distributed applications across Hadoop cluster and provide computing resources like CPU, memory and etc. The support for workloads provided by YARN enhances the power of Hadoop cluster through its iterative modeling. These models allow enterprises to realize near real-time processing og big data.

Spark

Spark is an in-memory distributed computing framework (Figure 5)for developing applications to perform general data analytics on distributed computing clusters. Spark speeds up the cluster computing processes through its in-memory feature. It processes by caching the dataset in memory and then performing computations at memory speeds and also by sharing data between subsequent iterations through memory (Kestelyn, 2013) Spark representation of a dataset is called RDD. RDD is a parallel data structure that lets users keep on intermediate results in memory, organize its partitioning to optimize data placement, and manipulate them using a rich set of operators and higher-level libraries like SparkSQL, Spark streaming, MLlib and GraphX. DAG is a finite directed graph of stages which are created based on various transformation applied to Spark RDDs. Each stage is comprised of tasks based on partitions of the RDD and these tasks should be executed on processing nodes. Spark supports YARN as cluster manager and its ability to read and write from and to HDFS enable Hadoop users to easily run Spark on their own Hadoop cluster.

Figure 4. Hadoop architecture

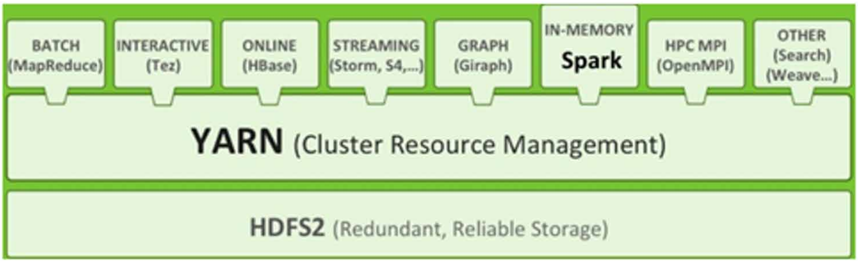
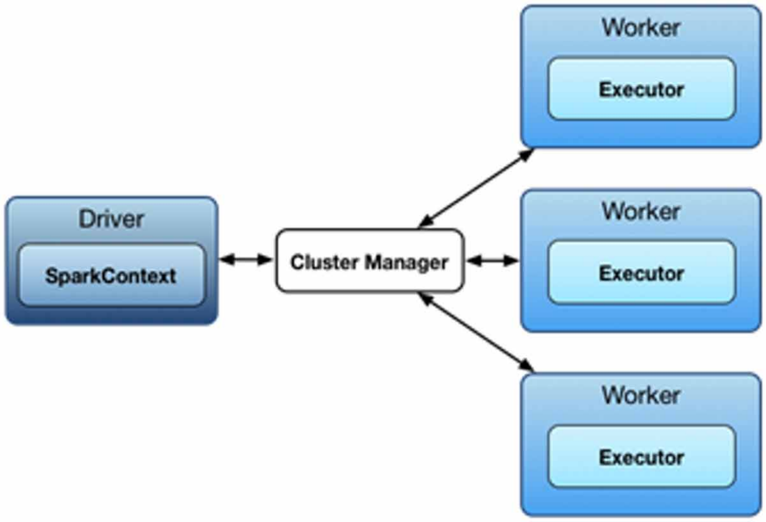

BATCH (MapReduce)
INTERACTIVE (Tez)
ONLINE (HBase)
STREAMING (Storm, S4,...)
GRAPH (Giraph)
IN-MEMORY Spark
HPC MPI (OpenMPI)
OTHER (Search) (Weave...)
YARN (Cluster Resource Management)
HDFS2 (Redundant, Reliable Storage)

Figure 5. Spark architecture



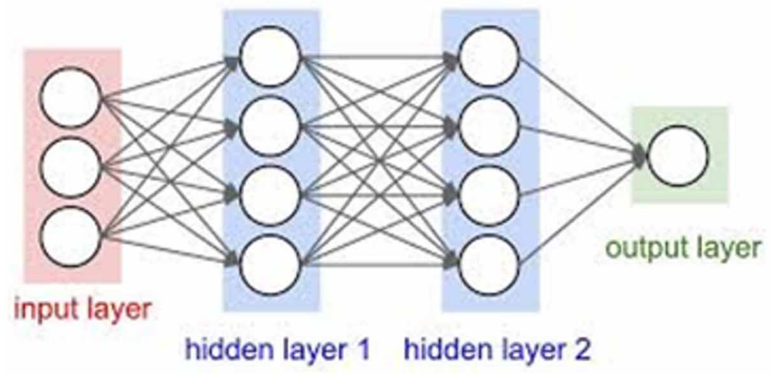
Deep Learning

A deep neural network (DNN) is an artificial neural network (ANN) with multiple hidden layers between the input and output layers (Figure 6). DNNs can model complex non-linear relationships. Deep learning can be trained in an unsupervised or supervised manner for both unsupervised and supervised learning tasks. The modern state-of-the-art deep learning is focused on training deep (many layered) neural network models using the back propagation algorithm. The most popular deep learning networks are:

- Multilayer Perceptron Networks.
- Convolutional Neural Networks.
- Long Short-Term Memory Recurrent Neural Networks.

Deep learning prototype can learn valuable representations of raw data and have spectacularly enhanced the contemporary in object detection, speech recognition, visual object recognition, and a lot of other areas. Also, deep learning models offer training scalability, stability, and generalization with big data. Due to its wide capability, Deep Learning is rapidly becoming the procedure of prime for the highest predicting precision (Ghoneim et al, 2017).

Figure 6. Deep learning architecture



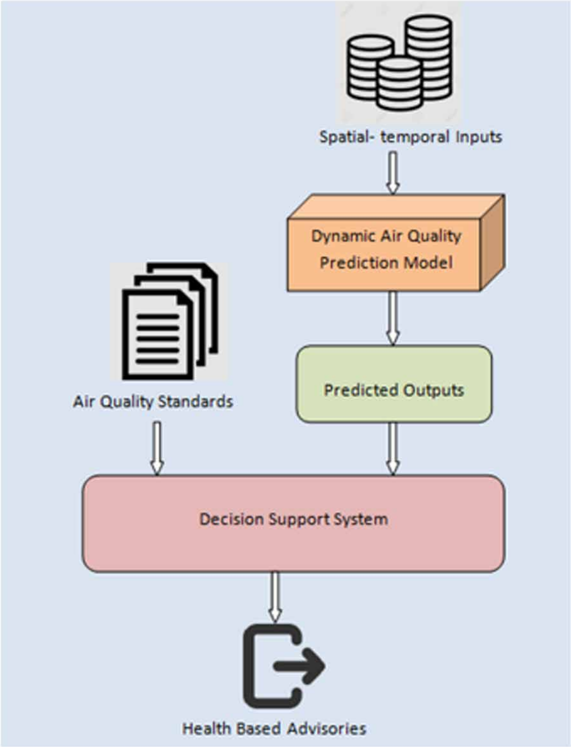
Prediction of air quality is an analysis of time series data. Traditional shallow models do not have the capacity for modeling sequential data with high accuracy rate. As time series data are more complex, high dimensional and noisy, Deep Learning is often associated with the problem of time series prediction

Proposed System

The spatial-temporal inputs considered for air quality prediction are dynamic large scale streaming, spatially expansive and behaviorally heterogeneous. Predicting air pollution under high resolution using this big dataset requires data processing technologies with high processing power and high capacity storage. In this chapter, we predict the real-time and fine-grained air quality value for an entire city based on their past air quality data reported by a limited number of existing monitor stations in the city and a variety of data sets that are observed in the city. Those observed city data could be meteorology, traffic flow, human mobility, structure of road networks, and POIs etc. Although there is no accurate relation between air quality and these factors, these models are usually based on experimental assumptions and the observed parameters that may not be applicable to all urban environments. In this chapter, we consider the meteorological data and the geographical data along with the past air quality data.

The proposed air pollution alert system (Figure 7) is a real time application that aid people to take decision regarding avoiding /reducing their exposure particularly on risky days. It provides 24 hours advanced location specific predicted values on mobile clients and website through web services. The spatial-temporal model used to predict air quality overcomes the limitation in accuracy due to the scarcity of monitoring station in urban areas.

Figure 7. Flow Chart of the Proposed System



DATA CONSIDERED FOR PREDICTION

Pollutant Data

The concentration of various pollutant collected from various monitoring stations are considered. These data is collected and is available in every air quality monitoring stations on hour basis. Due to the high cost of construction the numbers of monitoring stations are not available along all parts of a city and hence the prediction is based on the pollutant data along with how this would vary with other observed data like meteorology, traffic flow and topography of a location.

Meteorology Data

This data includes the values of wind speed, temperature and humidity for every location on hourly basis. This data is collected from the government's meteorology centre. Researches had proved that the amount of a pollutant concentration varies with these meteorology parameters.

Geographical Data

This data includes the topography features of a location such as its height, distance from the available peaks nearby, distance from the nearby roads, the traffic flow rate, various points of interests (POIs) etc. These features of a location greatly influence the concentration of the pollutants found in it. For example, the locations near to traffic roads embrace the adverse effects of air pollution.

Data Preprocessing

A more sophisticated M-BP algorithm (Li et al, 2017) capable of overcoming the missing data challenge, which provides high temporal-spatial air pollution estimation, at low computational complexity is considered. This data preprocessing is deep (has recurrent structures) and can perform sequential feature extraction automatically, which saves the step of constructing multiple classifiers. Instead of updating the parameters of the prediction network, M-BP updates the missing values of input data to minimize the prediction loss with the trained prediction neural network. Without the M-BP, the model takes 5 hours to converge on a single Nvidia Titan GPU card, while with the M-BP, it takes 8 hours to converge.

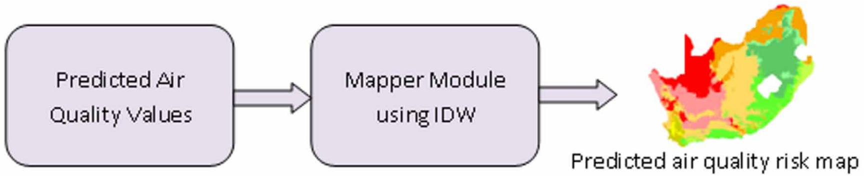
Model

Online deep learning based knowledge discovery accelerates the performance of back-propagation neural network to handle the large quantity of online air pollution streaming data taken as input for the spatial- temporal model. Hadoop based distributed computing overcomes the large amount of memory and computation power requirements for training the large scale dataset involved in air pollution forecasting. Deep Learning on Apache Spark increases the processing time of the prediction which is an indispensable need for any dynamic application like air quality prediction.

Mapper Module

The Mapper module is used to generate predictive air quality risk maps using an interpolation method. There are various interpolation methods, like Kriging, Spline and IDW (inverse distance weighting). Here, IDW method was used to create continuous and high precision spatial distribution maps for the predicted air quality values on each location for the entire city (Figure 8).

Figure 8. Mapper Module using IDW



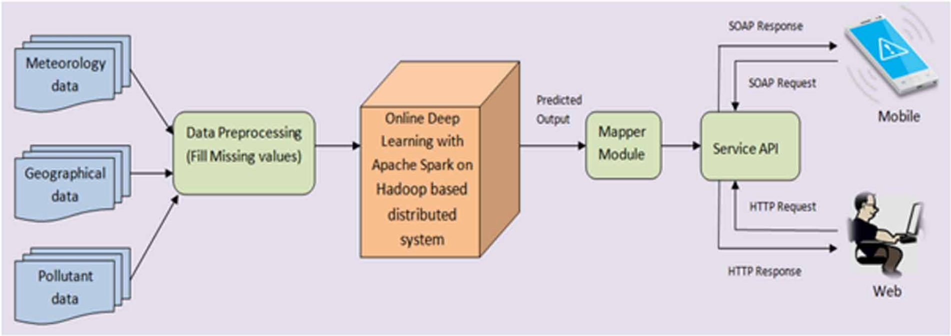
Alert System

The predictive maps and the alerts based on the air quality standards are then made available to the people via a mobile client service. The architecture of the above proposed system is as shown in Figure 9.

SUMMARY

Vast economic development with increase in population rise in cities has led to large environmental pollution problems involving air pollution, water pollution, noise and thermal pollution. Among these, air pollution has the most menacing effects on all life on earth. Air pollutant concentration keep increasing due to industrial, commercial and domestic sources. Various diseases, allergies, global warming and acid rains are some of the adverse outcomes of air pollution. Enhancement in computing technologies has started attracting increasing attention towards creating awareness in mitigating the adverse effects of pollution on human beings in both developing and developed countries. Accurate air quality forecasting can significantly reduce these grave effects of pollution on human beings and improvise their life standard.

Figure 9. Architecture of the Proposed System



Hence, the need to improve air quality forecasting has become indispensable for the betterment of the society.

In this chapter, the need for air pollution alert system is summarized and its implementation using machine learning algorithm is also elaborated. The handiness on the implementation of the fine grained real time air quality prediction with advancement in big data is well justified. Apart from the recent literature study and reviews here we also proposed an air quality evaluation based on big data analytics, machine learning models and other techniques. The proposed model is well suitable for all urban environments and could play a vital role in environmental protection against air pollution.

REFERENCES

Asgari, M., Farnaghi, M., & Ghaemi, Z. (2017, September). Predictive mapping of urban air pollution using Apache Spark on a Hadoop cluster. In *Proceedings of the 2017 International Conference on Cloud and Big Data Computing* (pp. 89-93). ACM. 10.1145/3141128.3141131

Chimmiri, M. (2016). *What is hadoop?* Retrieved December 23 2016, from http://www.hadooptpoint.com/what-is-hadoop/http://www.hadooptpoint.com/what-is-hadoop/

EPA. (2005). *Six common air pollutants*. U. S. environmental protection agency. Retrieved from http://www.epa.gov/air/urbanair/6 poll.html

Ghaemi, Z., Alimohammadi, A., & Farnaghi, M. (2018). LaSVM-based big data learning system for dynamic prediction of air pollution in Tehran. *Environmental Monitoring and Assessment*, *190*(5), 300. doi:10.100710661-018-6659-6 PMID:29679160

Ghoneim, O. A., & Manjunatha, B. R. (2017, September). Forecasting of ozone concentration in smart city using deep learning. In *Advances in Computing, Communications and Informatics (ICACCI), 2017 International Conference on* (pp. 1320-1326). IEEE. 10.1109/ICACCI.2017.8126024

Gorai, A. K., & Mitra, G. (2017). A comparative study of the feed forward back propagation (FFBP) and layer recurrent (LR) neural network model for forecasting ground level ozone concentration. *Air Quality, Atmosphere & Health*, *10*(2), 213–223. doi:10.100711869-016-0417-0

Goyal, P., Mishra, D., & Upadhyay, A. (2015). Forecasting of ozone episodes through statistical and artificial intelligence based models over Delhi metropolitan area. *Recent Researches in Applied Mathematics, Simulation and Modeling*, 111-120.

KestelynJ. (2013). Retrieved from http://blog.cloudera.com/blog/2013/11/putting-spark-to-use-fast-in-memory-computing-for-your-big-dataapplications/http://blog.cloudera.com/blog/2013/11/putting-spark-to-se-fast-in-memory-computing-for-your-big-data-applications/

Li, V. O., Lam, J. C., Chen, Y., & Gu, J. (2017, December). Deep learning model to estimate air pollution using m-bp to fill in missing proxy urban data. In *GLOBECOM 2017-2017 IEEE Global Communications Conference* (pp. 1-6). IEEE. 10.1109/GLOCOM.2017.8255004

Li, X., Peng, L., Yao, X., Cui, S., Hu, Y., You, C., & Chi, T. (2017). Long short-term memory neural network for air pollutant concentration predictions: Method development and evaluation. *Environmental Pollution*, *231*, 997–1004. doi:10.1016/j.envpol.2017.08.114 PMID:28898956

Mishra, D., & Goyal, P. (2016). Neuro-fuzzy approach to forecast NO2 pollutants addressed to air quality dispersion model over Delhi, India. *Aerosol and Air Quality Research*, *16*(1), 166–174. doi:10.4209/aaqr.2015.04.0249

Qi, Z., Wang, T., Song, G., Hu, W., Li, X., & Zhang, Z. M. (2018). Deep Air Learning: Interpolation, Prediction, and Feature Analysis of Fine-grained Air Quality. *IEEE Transactions on Knowledge and Data Engineering*, *30*(12), 2285–2297. doi:10.1109/TKDE.2018.2823740

Ren, Z., Zhu, J., Gao, Y., Yin, Q., Hu, M., Dai, L., & Li, X. (2018). Maternal exposure to ambient PM 10 during pregnancy increases the risk of congenital heart defects: Evidence from machine learning models. *The Science of the Total Environment*, *630*, 1–10. doi:10.1016/j.scitotenv.2018.02.181 PMID:29471186

Srimuruganandam, B., & Nagendra, S. S. (2015). ANN-based PM prediction model for assessing the temporal variability of PM10, PM2. 5 and PM1 concentrations at an urban roadway. *International Journal of Environmental Engineering*, *7*(1), 60–89. doi:10.1504/IJEE.2015.069266

Zheng, Y., Liu, F., & Hsieh, H. P. (2013, August). U-air: When urban air quality inference meets big data. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 1436-1444). ACM. 10.1145/2487575.2488188

Compilation of References

Abu Alsheikh, M., Lin, S., Niyato, D., & Tan, H. P. (2014). Machine learning in wireless sensor networks: Algorithms, strategies, and applications. *IEEE Communications Surveys and Tutorials*, *16*(4), 1996–2018. doi:10.1109/COMST.2014.2320099

Abu Oun. (2015). *Designing multiscale hybrid platform for testing and evaluating IoT systems*. Université de Franche-Comté.

Abu-Mostafa, Magdon-Ismail, & Lin. (2012). *Learning from data*. AMLBook.

Adjih, C., Baccelli, E., Fleury, E., Harter, G., Mitton, N., Noel, T., . . . Watteyne, T. (2016). FIT IoT-LAB: A large scale open experimental IoT testbed. *IEEE World Forum on Internet of Things, WF-IoT 2015 - Proceedings*, 459–464.

AFour Technologies. (2017). *IoT Testing Services*. Retrieved from https://afourtech.com/iot-testing-services/

Aharon, M., Elad, M., & Bruckstein, A. (2006). K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. *Signal Processing, IEEE Transactions on*, *54*(11), 4311–4322. doi:10.1109/TSP.2006.881199

Alessi, M., Giangreco, E., Pinnella, M., Pino, S., Storelli, D., Mainetti, L., ... Patrono, L. (2016). A web based virtual environment as a connection platform between people and IoT. In *Computer and energy science (SpliTech), international multidisciplinary conference on* (pp. 1–6). IEEE. doi:10.1109/SpliTech.2016.7555925

Alexander, C. (1964). *Notes on the Synthesis of Form*. Cambridge, MA: Harvard University Press.

Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys and Tutorials*, *17*(4), 2347–2376. doi:10.1109/COMST.2015.2444095

Ali, R., Dalpiaz, F., & Giorgini, P. (2010). A goal-based framework for contextual requirements modeling and analysis. *Requirements Engineering*, *15*(4), 439–458. doi:10.100700766-010-0110-z

Ambler, S. W. (2014). We need more Agile IT Now! In *Dr. Dobb's The world of software Development*. San Francisco: UBM.

Asgari, M., Farnaghi, M., & Ghaemi, Z. (2017, September). Predictive mapping of urban air pollution using Apache Spark on a Hadoop cluster. In *Proceedings of the 2017 International Conference on Cloud and Big Data Computing* (pp. 89-93). ACM. 10.1145/3141128.3141131

Atzori, Iera, & Morobito. (2010). The Internet of Things: A survey. *Journal of Computer Network, 54*(15), 2787-2805.

Atzori, L., Girau, R., Martis, S., Pilloni, V., & Uras, M. (2017, March). A SIoT-aware approach to the resource management issue in mobile crowd sensing. In *Innovations in Clouds, Internet and Networks (ICIN), 2017 20th Conference on* (pp. 232-237). IEEE.

Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A Survey. *Computer Networks*, *54*(15), 2787–2805. doi:10.1016/j.comnet.2010.05.010

Atzori, L., Iera, A., & Morabito, G. (2011). Social Internet of Things: Giving a social structure to the internet of things. *IEEE Communications Letters*, *15*(11), 1193–1195. doi:10.1109/LCOMM.2011.090911.111340

Atzori, L., Iera, A., & Morabito, G. (2017). Understanding the Internet of Things: Definition, potentials, and societal role of a fast evolving paradigm. *Ad Hoc Networks*, *56*, 122–140. doi:10.1016/j.adhoc.2016.12.004

Atzori, L., Iera, A., Morabito, G., & Nitti, M. (2012). The social internet of things (siot)–when social networks meet the internet of things: Concept, architecture and network characterization. *Computer Networks*, *56*(16), 3594–3608. doi:10.1016/j.comnet.2012.07.010

Babusiak, B., & Borik, S. (2015). Low energy wireless communication for medical devices. *38th International Conference on Telecommunications and Signal Processing (TSP)*, 444-447. 10.1109/TSP.2015.7296301

Bahga, A., & Madisetti, V. (2015). *Internet of Things-A Hands on approach*. University Press.

Bahrepour, M., Meratnia, N., Poel, M., Taghikhaki, Z., & Havinga, P. J. (2010). Distributed event detection in wireless sensor networks for disaster management. *2nd International Conference on Intelligent Networking and Collaborative Systems*, 507–512. 10.1109/INCOS.2010.24

Bandyopadhyay & Sen. (2011). Internet of Things: Applications and challenges in technology and standardization. *Springer International Journal of Wireless Personal Communications*, 49-69.

Bank, S. (2014). *Warehouse Management Systems & Warehouse Control Systems in the Age of the Internet of Things*. SupplyChain247.

Bassel, G. W., Glaab, E., Marquez, J., Holdsworth, M. J., & Bacardit, J. (2011). Functional Network Construction in Arabidopsis Using Rule-Based Machine Learning on Large-Scale Data Sets. *The Plant Cell, 23*(9), 3101–3116. doi:10.1105/tpc.111.088153

Belli, L., Cirani, S., Davoli, L., Gorrieri, A., Mancin, M., Picone, M., & Ferrari, G. (2015). Design and deployment of an IoT application-oriented testbed. *Computer*, *48*(9), 32–40. doi:10.1109/MC.2015.253

Bengio, Y. (2009). *Learning Deep Architectures for AI*. Now Publishers Inc.

Bernal Bernabé, J., Elicegui Maestro, I., Gandrille, E., Gligoric, N., Gluhak, A., Hennebert, C., & Nati, M. (2017). *SocIoTal-The development and architecture of a social IoT framework*. Academic Press.

Bertino, E., Choo, K. K. R., Georgakopolous, D., & Nepal, S. (2016). Internet of Things (IoT): Smart and secure service delivery. *ACM Transactions on Internet Technology*, *16*(4), 22. doi:10.1145/3013520

Best Practices in change, Configuration and Release Management (Report). (2010, July 14). Gartner.

Bischoff, U., & Kortuem, G. (2007). Life cycle support for sensor network applications. In *Proceedings of the 2nd international workshop on Middleware for sensor networks* (pp. 1-6). ACM. 10.1145/1376860.1376861

Bormann, C. (2014). *Test descriptions for ETSI plug tests coap 4*. Eur. Telecommunication Standards Institute London, U.K. Tech. Rep. 7-9.

Botta, A., De Donato, W., Persico, V., & Pescapé, A. (2016). Integration of cloud computing and internet of things: A survey. *Future Generation Computer Systems*, *56*, 684–700. doi:10.1016/j.future.2015.09.021

Bourne, J. (2017). *New research questions strategic importance of DevOps despite rise in usage*. CloudTech.

Branch, J. W., Giannella, C., Szymanski, B., Wolff, R., & Kargupta, H. (2013). In-network outlier detection in wireless sensor networks. *Knowledge and Information Systems*, *34*(1), 23–54. doi:10.100710115-011-0474-5

Buczak, L., & Guven, E. (2015). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys and Tutorials*, *18*(2), 1153–1176. doi:10.1109/COMST.2015.2494502

Budida, D. A. M., & Mangrulkar, R. S. (2017). Design and Implementation of Smart HealthCare System Using IoT. *International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*. 10.1109/ICIIECS.2017.8275903

Butani, K.V, B.P., & S.B. (2018). Social IOT: Network of Smart Things with Social Connections. *Journal of Engineering and Technology, 11*(2018), 60-63.

Buyya, R., & Dastjerdi, A. V. (2016). *Internet of Things: Principles and Paradigms*. Retrieved from http://www.buyya.com/papers/IoT-Book2016-C1.pdf

Campagna, G., Seo, J., Fischer, M., & Lam, M. S. (2015). Thing Talk: A Distributed Language for a Social Internet of Things. *Work (Reading, Mass.)*.

Casagras, R. (2011). *RFID and the inclusive model for the Internet of Things report*. Academic Press.

Catarinucci, L. (2015). An IoT-Aware Architecture for Smart Healthcare Systems. IEEE Internet of Things Journal, 2(6), 515-526. doi:10.1109/JIOT.2015.2417684

Chen, L. (2015). Continuous Delivery: Huge Benefits, but Challenges Too. *IEEE Software*, *32*(2), 50–54. doi:10.1109/MS.2015.27

Che, X., & Maag, S. (2013). A passive testing approach for protocols in Internet of Things. *IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*, 678–684. 10.1109/GreenCom-iThings-CPSCom.2013.124

Chimmiri, M. (2016). *What is hadoop?* Retrieved December 23 2016, from http://www.hadooptpoint.com/what-is-hadoop/http://www.hadooptpoint.com/what-is-hadoop/

Cigniti. (2018). *The Need for Testing the Internet of Things*. Retrieved from https://www.cigniti.com/blog/the-need-for-testing-the-internet-of-things/

Cigniti. (2018). *Why the Healthcare Sector Needs QA & Testing*. Retrieved from https://www.cigniti.com/blog/top-6-reasons-healthcare-sector-needs-qa-testing/

Cognizant. (2016). *The internet of things: QA unleashed*. Retrieved from https://www.cognizant.com/InsightsWhitepapers/theinternet-of-things-qa-unleashed-codex1233.pdf

Costa, P., Mottola, L., Murphy, A. L., & Picco, G. P. (2007). Programming wireless sensor networks with the teeny lime middleware. In *ACM/IFIP/USENIX International Conference on Distributed Systems Platforms and Open Distributed Processing* (pp. 429-449). Springer.

Darshan, K. R., & Anandakumar, K. R. (2015). A Comprehensive Review on Usage of Internet of Things (IoT) in Healthcare System. *International Conference on Emerging Research in Electronics, Computer Science and Technology (ICERECT)*, 132-136, 374–380. 10.1109/ERECT.2015.7499001

DataKitchen. (2017). *How to Become a Rising Star with Data Analytics*. Author.

Debois, P. (2009). *DevOpsDays Ghent*. DevopsDays.

Desnitsky, V., & Kotenko, I. (2016). Automated design, verification and testing of secure systems with embedded devices based on elicitation of expert knowledge. *Journal of Ambient Intelligence and Humanized Computing*, *7*(5), 705–719. doi:10.100712652-016-0371-6

DevOps: A Job Title or a School of Thought? (2017). Monster Career Advice.

Dhar, K., Bhunia, S. S., & Mukherjee, N. (2014). Interference Aware Scheduling of Sensors in IoT Enabled Health-Care Monitoring System. *Fourth International Conference of Emerging Applications of Information Technology*, 152-157. 10.1109/EAIT.2014.50

EPA. (2005). *Six common air pollutants*. U. S. environmental protection agency. Retrieved from http://www.epa.gov/air/urbanair/6 poll.html

Ericsson. (2011). *More than 50 billion connected devices*. White Paper 284 23-3149 Uen.

Esquiagola, J., Costa, L., Calcina, P., Fedrecheski, G., & Zuffo, M. (2017). Performance Testing of an Internet of Things Platform. *Proceedings of the 2nd International Conference on Internet of Things, Big Data and Security*, 309-314. 10.5220/0006304503090314

Eurostat. (2007). *Passenger mobility in Europe*. European Commission.

Eurostat. (2011). *Energy, transport and environment indicators*. European Commission.

Fernández-Caramés, T. M., Fraga-Lamas, P., Suárez-Albela, M., & Castedo, L. (2016). Reverse Engineering and Security Evaluation of Commercial Tags for RFID-Based IoT Applications. *Sensors (Basel)*, *17*(1), 28. doi:10.339017010028 PMID:28029119

Fielding, R. (1999). *Hypertext Transfer Protocol-HTTP/1.1, document RFC 2616*. Network Working Group. doi:10.17487/rfc2616

France, R., & Rumpe, B. (2007). Model-driven development of complex software: A research roadmap. In *2007 Future of Software Engineering* (pp. 37–54). IEEE Computer Society. doi:10.1109/FOSE.2007.14

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (2003). *Design Patterns: Elements of Reusable Object-Oriented Software with Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process by 2003*. Academic Press.

Ganti, R. K., Ye, F., & Lei, H. (2011). Mobile crowd sensing: Current state and future challenges. *IEEE Communications Magazine*, *49*(11), 32–39. doi:10.1109/MCOM.2011.6069707

Gartner IT Glossary – devops. (2015). Gartner.

Ghaemi, Z., Alimohammadi, A., & Farnaghi, M. (2018). LaSVM-based big data learning system for dynamic prediction of air pollution in Tehran. *Environmental Monitoring and Assessment*, *190*(5), 300. doi:10.100710661-018-6659-6 PMID:29679160

Ghoneim, O. A., & Manjunatha, B. R. (2017, September). Forecasting of ozone concentration in smart city using deep learning. In *Advances in Computing, Communications and Informatics (ICACCI), 2017 International Conference on* (pp. 1320-1326). IEEE. 10.1109/ICACCI.2017.8126024

Giusto, D., Iera, A., Morabito, G., & Atzori, L. (Eds.). (2010). The Internet of Things. Springer.

Gorai, A. K., & Mitra, G. (2017). A comparative study of the feed forward back propagation (FFBP) and layer recurrent (LR) neural network model for forecasting ground level ozone concentration. *Air Quality, Atmosphere & Health*, *10*(2), 213–223. doi:10.100711869-016-0417-0

Goswami, S., Misra, S., Taneja, C., & Mukherjee, A. (2014). Securing intra-communication in 6LoWPAN: A PKI integrated scheme. *2014 IEEE International Conference on Advanced Networks and Telecommuncations Systems (ANTS)*, 1–5. 10.1109/ANTS.2014.7057265

Goyal, P., Mishra, D., & Upadhyay, A. (2015). Forecasting of ozone episodes through statistical and artificial intelligence based models over Delhi metropolitan area. *Recent Researches in Applied Mathematics, Simulation and Modeling*, 111-120.

Grgić, K., Špeh, I., & Heđi, I. (2016). A web-based IoT solution for monitoring data using MQTT protocol. In *Smart Systems and Technologies (SST), International Conference on* (pp. 249-253). IEEE. 10.1109/SST.2016.7765668

Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of things (iot): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, *29*(7), 1645–1660. doi:10.1016/j.future.2013.01.010

Guinard, D., Fischer, M., & Trifa, V. (2010, March). Sharing using social networks in a composable Web of Things. In PerCom Workshops (pp. 702-707). Academic Press. doi:10.1109/PERCOMW.2010.5470524

Guinard, D., & Trifa, V. (2016). *Building the web of things: with examples in node. js and raspberry pi.* Manning Publications Co.

Gyory, N., & Chuah, M. (2017). IoTOne: Integrated platform for heterogeneous IoT devices, *International Conference on Computing, Networking and Communications (ICNC)*, 783-787. 10.1109/ICCNC.2017.7876230

Gyrard, A., Patel, P., Sheth, A. P., & Serrano, M. (2016). Building the web of knowledge with smart iot applications. *IEEE Intelligent Systems*, *31*(5), 83–88. doi:10.1109/MIS.2016.81

Hackbarth, R., Mockus, A., Palframan, J., & Sethi, R. (2016). Improving Software Quality as Customers Perceive It. *IEEE Software*, *33*(4), 40–45. doi:10.1109/MS.2015.76

Hammond, J. (2011). *The Relationship between DevOps and Continuous Delivery*. Forrester Research.

Hancke, G., Silva, B., & Hancke, G. Jr. (2012). The Role of Advanced Sensing in Smart Cities. *Sensors (Basel)*, *13*(1), 393–425. doi:10.3390130100393 PMID:23271603

Hassanalieragh, M., Page, A., Soyata, T., Sharma, G., Aktas, M., Mateos, G., … Andreescu, S. (2015). Health Monitoring and Management UsingInternet-of-Things (IoT) Sensing with Cloud-based Processing: Opportunities and Challenges. *IEEE International Conference on Services Computing*, 285-292.

He & Li. (2014). Internet of Things in Industries: A Survey. *IEEE Transactions on Industrial Informatics, 10*(4), 2232-2243.

Henze, M., Hermerschmidt, L., Kerpen, D., Häußling, R., Rumpe, B., & Wehrle, K. (2016). A Comprehensive approach to privacy in the cloud-based Internet of Things. *Future Generation Computer Systems*, *56*, 701–718. doi:10.1016/j.future.2015.09.016

Heuer, J., Hund, J., & Pfaff. (2015). Toward the web of things: Applying web technologies to the physical world. *Computer, 48*(5), 34–42.

Hillah, L. M., Maesano, A. P., De Rosa, F., Kordon, F., Wuillemin, P. H., Fontanelli, R., & Maesano, L. (2017). Automation and intelligent scheduling of distributed system functional testing. *International Journal of Software Tools for Technology Transfer*, *19*(3), 281–308. doi:10.100710009-016-0440-3

How the Internet of Things Is Improving Transportation and Logistics. (2015). *Transportation News*.

Humble, J., & Farley, D. (2011). *Continuous Delivery: reliable software releases through build, test, and deployment automation*. Pearson Education Inc.

Hung, M. (2017). Leading the IoT. *Gartner Insights on How to lead in a Connected World*. Retrieved from https://www.gartner.com/imagesrv/books/iot/iotEbook_digital.pdf

IEEE Internet Technology Policy Community. (2017). *Internet of Things (IoT) Security Best Practices*. Retrieved from https://internetinitiative.ieee.org/images/files/resources/white_papers/internet_of_things_feb2017.pdf

Internet of Things Research Study Report. (2015). Hewlett Packard Enterprise. Retrieved from www8.hp.com/h20195,/V2/GetPDF.aspx/4AA5-4759 ENW.pdf

Is DevOps a Title? (2014). DevOps.com.

Islam, S. M., Raizul, Kwak, D., Kabir, M. D., Humaun, Hossain, M., & Kwag, K-S. (2015). The Internet of Things for Health Care: A Comprehensive Survey. *IEEE Access, 3*, 678-708.

Jara, A. J., Bocchi, Y., & Genoud, D. (2014, September). Social Internet of Things: The potential of the Internet of Things for defining human behaviours. In *Intelligent Networking and Collaborative Systems (INCoS), 2014 International Conference on* (pp. 581-585). IEEE.

Jaradat, M., Jarrah, M., Bousselham, A., Jararweh, Y., & Al-Ayyoub, M. (2015). The Internet of Energy: Smart Sensor Networks and Big Data Management for Smart Grid. *Procedia Computer Science*, *56*, 592–597. doi:10.1016/j.procs.2015.07.250

Jayaraman, P. P., Zaslavsky, A., & Delsing, J. (2010). *Intelligent processing of k-nearest neighbors queries using mobile data collectors in a location aware 3D wireless sensor network. In Trends in Applied Intelligent Systems* (pp. 260–270). Springer.

Jin, Gubbi, Marusic, & Palaniswami. (2014). An information framework for creating a smart city through internet of things. *IEEE Internet of Things Journal, 1*(2), 112-121.

Jones, S., Noppen, J., & Lettice, F. (2016). *Management challenges for DevOps adoption within UK SMEs*. Academic Press.

Kao, K.-C., Chieng, W.-H., & Jeng, S.-L. (2018). Design and development of an IoT-based web application for an intelligent remote SCADA system. In *IOP Conference Series: Materials Science and Engineering* (*vol. 323*, no. 1, pp. 12-25). IOP Publishing. 10.1088/1757-899X/323/1/012025

KestelynJ. (2013). Retrieved from http://blog.cloudera.com/blog/2013/11/putting-spark-to-use-fast-in-memory-computing-for-your-big-dataapplications/http://blog.cloudera.com/blog/2013/11/putting-spark-to-se-fast-in-memory-computing-for-your-big-data-applications/

Key findings of the Teradata 2015 Global Data-Driven Marketing Survey. (2015). Retrieved from https://www.i-scoop.eu/data-driven-marketing-the-state-benefits-and-drivers-of-data-marketing/

Khan, R., Khan, S. U., Zaheer, R., & Khan, S. (2012). Future internet: the internet of things architecture, possible applications and key challenges. In *Frontiers of Information Technology (FIT), 2012 10th International Conference on* (pp. 257-260). IEEE. 10.1109/FIT.2012.53

Kim, H., Ahmad, A., Hwang, J., Baqa, H., Le Gall, F., Reina Ortega, M. A., & Song, J. S. (2018). IoT-TaaS: Towards a Prospective IoT Testing Framework. *IEEE Access: Practical Innovations, Open Solutions*, *6*, 15480–1549. doi:10.1109/ACCESS.2018.2802489

Kim, M., & Park, M.-G. (2009). Bayesian statistical modeling of system energy saving effectiveness for MAC protocols of wireless sensor networks. In *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*. *Springer Berlin Heidelberg*.

Kiruthika, J., & Khaddaj, S. (2015). Software Quality Issues and Challenges of Internet of Things. *14th International Symposium on Distributed Computing and Applications for Business Engineering and Science*, 176-179. 10.1109/DCABES.2015.51

Komaki, D., Yamaguchi, S., Shinohara, M., Horio, K., Murakami, M., & Matsui, K. (2017). Design and Implementation of a Multimedia Control and Processing Framework for IoT Application Development. *International Journal of Informatics Society*, *9*(2), 73–84.

Kowshalya, A. M., & Valarmathi, M. L. (2017). Trust management for reliable decision making among social objects in the Social Internet of Things. *IET Networks*, *6*(4), 75–80. doi:10.1049/iet-net.2017.0021

Koza, J. R., Bennett, F. H., Andre, D., & Keane, M. A. (1996). Automated Design of Both the Topology and Sizing of Analog Electrical Circuits Using Genetic Programming. In *Artificial Intelligence in Design '96*. Springer; doi:10.1007/978-94-009-0279-4_9

Kulkarni, A., & Sathe, S. (2014). Healthcare applications of the Internet of Things: A Review. *International Journal of Computer Science and Information Technologies*, *5*(5), 6229–6232.

Kulkarni, R. V., & Venayagamoorthy, G. K. (2009). Neural network based secure media access control protocol for wireless sensor networks. *Proc. Int'l Joint Conf. Neural Networks*, 3437–3444. 10.1109/IJCNN.2009.5179075

Kulkarni, V., & Reddy, S. (2003). Separation of concerns in model-driven development. *IEEE Software*, *20*(5), 64–69. doi:10.1109/MS.2003.1231154

Lee, H., & Kwon, J. (2015, November). Survey and Analysis of Information Sharing in Social IoT. In *Disaster Recovery and Business Continuity (DRBC), 2015 8th International Conference on* (pp. 15-18). IEEE. 10.1109/DRBC.2015.13

Lee, H., Grosse, R., Ranganath, R., & Ng, A. Y. (2009). Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations. *Proceedings of the 26th Annual International Conference on Machine Learning*. 10.1145/1553374.1553453

Leonardi, L., Patti, G., & Lo Bello, L. (2018). Multi-Hop Real-Time Communications Over Bluetooth Low Energy Industrial Wireless Mesh Networks. *IEEE Access: Practical Innovations, Open Solutions*, *6*, 26505–26519. doi:10.1109/ACCESS.2018.2834479

Li, V. O., Lam, J. C., Chen, Y., & Gu, J. (2017, December). Deep learning model to estimate air pollution using m-bp to fill in missing proxy urban data. In *GLOBECOM 2017-2017 IEEE Global Communications Conference* (pp. 1-6). IEEE. 10.1109/GLOCOM.2017.8255004

Lin, H., & Bergmann, N. W. (2016). IoT privacy and security challenges for smart home environments. *Information*, *7*(3), 44. doi:10.3390/info7030044

Li, S., Da Xu, L., & Zhao, S. (2015). The internet of things: A survey. *Information Systems Frontiers*, *17*(2), 243–259. doi:10.100710796-014-9492-7

Li, X., Lu, R., Liang, X., Shen, X., Chen, J., & Lin, X. (2011). Smart community: An internet of things application. *IEEE Communications Magazine*, *49*(11), 68–75. doi:10.1109/MCOM.2011.6069711

Li, X., Peng, L., Yao, X., Cui, S., Hu, Y., You, C., & Chi, T. (2017). Long short-term memory neural network for air pollutant concentration predictions: Method development and evaluation. *Environmental Pollution*, *231*, 997–1004. doi:10.1016/j.envpol.2017.08.114 PMID:28898956

Lloret, J., Tomas, J., Canovas, A., & Parra, L. (2016). An integrated IoT architecture for smart metering. *IEEE Communications Magazine*, *54*(12), 50–57. doi:10.1109/MCOM.2016.1600647CM

Loukides, M. (2012). *What is Devops?* O'Reilly Media.

Lu, C.-H., & Fu, L.-C. (2009). Robust location-aware activity recognition using wireless sensor network in an attentive home. *IEEE Transactions on Automation Science and Engineering*, *6*(4), 598–609. doi:10.1109/TASE.2009.2021981

Lu, H., Plataniotis, K. N., & Venetsanopoulos, A. N. (2011). A Survey of Multilinear Subspace Learning for Tensor Data. *Pattern Recognition*, *44*(7), 1540–1551. doi:10.1016/j.patcog.2011.01.004

Lunardi, W. T., de Matos, E., Tiburski, R., Amaral, L. A., Marczak, S., & Hessel, F. (2015). Context-based search engine for industrial iot: Discovery, search, selection, and usage of devices. *IEEE 20th Conference on Emerging Technologies Factory Automation*, 1–8.

Machado, H., & Shah, K. (2016). *Internet of Things (IoT) impacts on Supply Chain*. machado2016internet.

Machine Learning: What it is and why it matters. Retrieved from www.sas.com

Macua, S., Belanovic, P., & Zazo, S. (2010). Consensus-based distributed principal component analysis in wireless sensor networks. *11th International Workshop on Signal Processing Advances in Wireless Communications*, 1–5.

Mahdavinejad, M. S. (2017). *Machine learning for Internet of Things data analysis: A survey*. Digital Communications and Networks.

Mall, R. (2014). *Fundamentals of Software Engineering (4th ed.)*. PHI-Delhi.

Marinissen, E. J., Zorian, Y., Konijnenburg, M., Huang, C. T., Hsieh, P. H., Cockburn, P., & Verbauwhede, I. (2016). Iot: Source of test challenges. *21th IEEE European Test Symposium*, 1-10.

MarketsandMartkets. (2017). *Internet of Things (IoT) Testing Market by Testing Type (Functional, Performance, Network, Security, Compatibility, and Usability), Service Type (Professional and Managed), Application Type, and Region - Global Forecast to 2021*. Retrieved from https://www.marketsandmarkets.com/Market-Reports/iot-testing-market-51412648.html

Masiero, R., Quer, G., Munaretto, D., Rossi, M., Widmer, J., & Zorzi, M. (2009). Data acquisition through joint compressive sensing and principal component analysis. *Global Telecommunications Conference*, 1–6. 10.1109/GLOCOM.2009.5425458

Matha, M. P. (2008). *Object-Oriented Analysis and Design using UML*. Delhi: PHI-N.

Meidan, Y. (2017). *Detection of Unauthorized IoT Devices Using Machine Learning Techniques*. arXiv preprint arXiv:1709.04647.

Mellor, S. J., Clark, T., & Futagami, T. (2003). Model-driven development: Guest editors' introduction. *IEEE Software*, *20*(5), 14–18. doi:10.1109/MS.2003.1231145

Middleton, P. (2014). *Forecast: Internet of Things, Endpoints and Associated Services, Worldwide, 2014 (G00270264)*. Gartner Database.

Mika, P. (2004, September). Social networks and the semantic web. In *Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence* (pp. 285-291). IEEE Computer Society.

Miles, R., & Hamilton, K. (2006). *Learning UML 2.0*. O'Reilly.

Minerva, R. (2016). *IoT and its Challenges*. Retrieved from https://iot.ieee.org/images/files/pdf/iot_and_its_challenges_roberto_minerva.pdf

Miorandi, D., Sicari, S., De Pellegrini, F., & Chlamtac, I. (2012). Internet of Things: Vision, applications and research challenges. *Ad Hoc Networks*, *10*(7), 1497–1516. doi:10.1016/j.adhoc.2012.02.016

Miori, V., & Russo, D. (2017, June). Improving life quality for the elderly through the Social Internet of Things (SIoT). In *Global Internet of Things Summit (GIoTS), 2017* (pp. 1–6). IEEE. doi:10.1109/GIOTS.2017.8016215

Mishra, D., & Goyal, P. (2016). Neuro-fuzzy approach to forecast NO2 pollutants addressed to air quality dispersion model over Delhi, India. *Aerosol and Air Quality Research*, *16*(1), 166–174. doi:10.4209/aaqr.2015.04.0249

Monostori, L. (2014). Cyber-physical production systems: Roots, expectations and R&D challenges. *Procedia Cirp*, *17*, 9–13. doi:10.1016/j.procir.2014.03.115

Montenegro, Kushalnagar, Hui, & Culler. (2007). *Transmission of IPV6 Packets Over IEEE 802.15.4 Networks*. document RFC 4944, 2007.

Mora, H., David, G., Terol, R. M., Azorin, J., & Szymanski, J. (2017). An IoT-Based Computational Framework forHealthcare Monitoring in Mobile Environments. *Sensors (Basel)*, *17*(10), 2302. doi:10.339017102302

Moustapha, A., & Selmic, R. (2008). Wireless sensor network modeling using modified recurrent neural networks: Application to fault detection. *IEEE Transactions on Instrumentation and Measurement*, *57*(5), 981–988. doi:10.1109/TIM.2007.913803

Narudin, F. A., Feizollah, A., Anuar, N. B., & Gani, A. (2016). Evaluation of machine learning classifiers for mobile malware detection. *Soft Computing*, *20*(1), 343–357. doi:10.100700500-014-1511-6

Nasrat, P. (2011). *Agile Infrastructure*. InfoQ.

Nguyen, T. D. (2018). *IoT: A Crowdsourced Self-learning Approach for Detecting Compromised IoT Devices*. arXiv preprint arXiv:1804.07474.

Niewolny. (2013). *How the Internet of Things Is Revolutionizing Healthcare*. Freescale Semiconductors.

Ning, H., & Wang, Z. (2011). Future internet of things architecture: Like mankind neural system or social organization framework? *IEEE Communications Letters*, *15*(4), 461–463. doi:10.1109/LCOMM.2011.022411.110120

Nitti, M., Murroni, M., Fadda, M., & Atzori, L. (2016). Exploiting social internet of things features in cognitive radio. *IEEE Access: Practical Innovations, Open Solutions*, *4*, 9204–9212. doi:10.1109/ACCESS.2016.2645979

Nordrum, A. (2016). *Popular Internet of Things Forecast of 50 Billion Devices by 2020 Is Outdated*. IEEE.

One M2M Testing Framework, document oneM2M TS-0015 v2.0.0, Aug. 2016. (n.d.). Retrieved from http://www.onem2m.org/images/files/deliverables/Release2/TS-0015-Testing_Framework-V2.0.0.pdf

Ortiz, A. M., Hussein, D., Park, S., Han, S. N., & Crespi, N. (2014). The cluster between internet of things and social networks: Review and research challenges. *IEEE Internet of Things Journal*, *1*(3), 206–215. doi:10.1109/JIOT.2014.2318835

Ozay, M., Esnaola, I., Yarman Vural, F. T., Kulkarni, S. R., & Poor, H. V. (2015). Machine learning methods for attack detection in the smart grid. *IEEE Transactions on Neural Networks and Learning Systems*, *27*(8), 1773–1786. doi:10.1109/TNNLS.2015.2404803 PMID:25807571

Palmer, A. (2015). *From DevOps to DataOps*. Tamr Inc.

Patel, P., & Cassou, D. (2015). Enabling high-level application development for the Internet of Things. *Journal of Systems and Software*, *103*, 62–84. doi:10.1016/j.jss.2015.01.027

Pathak, A., Mottola, L., Bakshi, A., Prasanna, V. K., & Picco, G. P. (2007). A compilation framework for macroprogramming networked sensors. In *International Conference on Distributed Computing in Sensor Systems* (pp. 189-204). Springer. 10.1007/978-3-540-73090-3_13

Pawlak, Z. (1991). *Rough Sets: Theoretical Aspects of Reasoning About Data*. Boston, MA: Kluwer Academic Publishers. doi:10.1007/978-94-011-3534-4

Pering, T., Farrington, K., & Dahm, T. (2018). Taming the IoT: Operationalized Testing to Secure Connected Devices. *IEEE Computer*, *51*(6), 90–94. doi:10.1109/MC.2018.2701633

Peters, J. F. (2003). Design Patterns in Intelligent Systems. In N. Zhong, Z.W. Ras, S. Tsumoto, & E. Suzuki (Eds.), Foundations of Intelligent Systems, Lecture Notes in Artificial Intelligence 2871 (pp. 262-269). Springer.

Peters, J. F. (2004). Approximation space for intelligent system design patterns. *Engineering Applications of Artificial Intelligence*, *17*(4), 393–400. doi:10.1016/j.engappai.2004.04.012

Peters, J. F., Skowron, A., Stepaniuk, J., & Ramanna, S. (2002). Towards an ontology of approximate reason. *Fundamenta Informaticae*, *51*(1), 2, 157–173.

Prehofer, C., & Chiarabini, L. (2017). *From IoT Mashups to Model-based IoT*. Retrieved from https://www.w3.org/2014/02/wot/papers/prehofer.pdf

Pressman, R.S. (2009). *Software Engineering: A Practitioner's Approach*. McGraw Hill, Intl edition.

Pressman, R. S. (1998). Can Internet-Based Applications Be Engineered? Issue No. 05 - September/October: Vol. 15. DOI Bookmark. http://doi.ieeecomputersociety.org/10.1109/MS.1998.714869

Qi, Z., Wang, T., Song, G., Hu, W., Li, X., & Zhang, Z. M. (2018). Deep Air Learning: Interpolation, Prediction, and Feature Analysis of Fine-grained Air Quality. *IEEE Transactions on Knowledge and Data Engineering*, *30*(12), 2285–2297. doi:10.1109/TKDE.2018.2823740

Rahman & Shah. (2016). Security analysis of IoT protocols: A focus in CoAP. *MEC International Conference on Big Data and Smart City*, 1-7.

Rao, Saluia, Sharma, Mittal, & Sharma. (2012). Cloud computing for Internet of Things & sensing based applications. *Proceedings of Sixth International Conference on Sensing Technology,* 374-380.

Rao, N. J. M. (2017). *IoT based Remote Patient Health Monitoring System (Master's thesis)*. Kansas State University.

Rathore, M. M., Ahmad, A., Paul, A., & Rho, S. (2016). Urban planning and building smart cities based on the Internet of Things using Big Data analytics. *Computer Networks*, *101*, 63–80. doi:10.1016/j.comnet.2015.12.023

Raymor, B., & Coppen, R. (2014). *OASIS Message Queuing Telemetry Transport (MQTT) TC*. Retrieved from https://www.oasis-open.org/committees/mqtt/

Ray, P. P. (2016). *A Survey on Internet of Things Architecture" Journal of King Saud University –Computer and Information Sciences, 1319-1578*. Elsevier.

Raza, S., Trabalza, D., & Voigt, T. (2012). 6LoWPAN Compressed DTLS for CoAP. *2012 IEEE 8th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 287–9.

Raza, S., Duquennoy, S., Chung, T., Yazar, D., Voigt, T., & Roedig, U. (2011). Securing communication in 6LoWPAN with compressed IPsec. *2011 International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS)*, 1–8. 10.1109/DCOSS.2011.5982177

Razzaque, M. A., Milojevic-Jevric, M., Palade, A., & Clarke, S. (2016). Middleware for internet of things: A survey. *IEEE Internet of Things Journal*, *3*(1), 70–95. doi:10.1109/JIOT.2015.2498900

RCR-Wireless. (2016). *Testing the internet of things: Making the IoT work*. Author.

Reetz, E. S., Kuemper, D., Moessner, K., & Toenjes, R. (2013). How to test IoT-based services before deploying them into real world. *Wireless Conference (EW), Proceedings of 19th European Wireless Conference*, 1–6.

Ren, Z., Zhu, J., Gao, Y., Yin, Q., Hu, M., Dai, L., & Li, X. (2018). Maternal exposure to ambient PM 10 during pregnancy increases the risk of congenital heart defects: Evidence from machine learning models. *The Science of the Total Environment*, *630*, 1–10. doi:10.1016/j.scitotenv.2018.02.181 PMID:29471186

ResOps, daily adventures of DevOps in Research - EMBL-EBI Technical Services Cluster blog. (2018, February 8). EMBL-EBI Technical Services Cluster blog.

Rifkin, J. (2014). *The Zero Marginal Cost Society: The Internet of Things, the Collaborative Commons, and the Eclipse of Capitalism*. New York, NY: Macmillan.

Román, M., Hess, C., Cerqueira, R., Ranganathan, A., Campbell, R. H., & Nahrstedt, K. (2002). Gaia: A middleware platform for active spaces. *Mobile Computing and Communications Review*, *6*(4), 65–67. doi:10.1145/643550.643558

Roman, R., Lopez, J., & Mambo, M. (2018). A survey and analysis of security threats and challenges. *Future Generation Computer Systems*, *78*(3), 680–698. doi:10.1016/j.future.2016.11.009

Rooshenas, A., Rabiee, H., Movaghar, A., & Naderi, M. (2010). Reducing the data transmission in wireless sensor networks using the principal component analysis. *6th International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, 133–138. 10.1109/ISSNIP.2010.5706781

Rumbaugh, J., & Blaha. (2007). *Object Oriented Modelling and design with UML-2* (2nd ed.). Pearson.

Sajid, A., & Abbas, H. (2016). Data privacy in cloud-assisted healthcare systems: State of the art and future challenges. *Journal of Medical Systems*, *40*(6), 155. doi:10.100710916-016-0509-2 PMID:27155893

Saksoft. (2018). *IoT Interoperability Testing*. Retrieved from https://www.360logica.com/blog/iot-interoperability-testing

Samuel, A. (1959). Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development*, *3*(3), 210–229. doi:10.1147/rd.33.0210

Sanchez, L., Muñoz, L., Galache, J. A., Sotres, P., Santana, J. R., Gutierrez, V., ... Pfisterer, D. (2014). SmartSantander: IoT experimentation over a smart city testbed. *Computer Networks*, *61*, 217–238. doi:10.1016/j.bjp.2013.12.020

Sanger, D., & Perlroth, N. (2016). A New Era of Internet Attacks Powered by Everyday Devices, *New York Times*.

Sawyer, P., Bencomo, N., Whittle, J., Letier, E., & Finkelstein, A. (2010). Requirements -Aware Systems: A Research Agenda for RE for Self-adaptive Systems. In Requirements Engineering Conference (RE), 18th IEEE International. Sydney: IEEE Computer Society.

Shafie-Khah, M., Heydarian-Forushani, E., Osório, G. J., Gil, F. A. S., Aghaei, J., Barani, M., & Catalão, J. P. S. (2016). Optimal Behavior of Electric Vehicle Parking Lots as Demand Response Aggregation Agents. *IEEE Transactions on Smart Grid*, *7*(6), 2654–2665. doi:10.1109/TSG.2015.2496796

Shanmugasundaram. (2015). *IoT Basics and Testing Focus.* Retrieved from https://theinternetofthings.report/Resources/Whitepapers/793406e1-2095-40a5-9369-70d3df83e844_iot_basics_and_testing_focus.pdf

Shelby, Z., Hartke, K., & Bormann, C. (2014). *The Constrained Application Protocol (COAP), document RFC 7252, Internet Engineering Task Force*. IETF.

Shen, Y.-J., & Wang, M.-S. (2008). Broadcast scheduling in wireless sensor networks using fuzzy hopfield neural network. *Expert Systems with Applications*, *34*(2), 900–907. doi:10.1016/j.eswa.2006.10.024

Sicari, S., Rizzardi, A., Grieco, L. A., & Coen-Porisini, A. (2015). Security, privacy and trust in Internet of Things: The road ahead. *Computer Networks*, *76*, 146–164. doi:10.1016/j.comnet.2014.11.008

Skowron, A. (2001). Toward intelligent systems: Calculi of information granules. Bulletin of the International Rough Set Society, 5(1/2), 9-30.

Skowron, A., & Stepaniuk, J. (1998). Information granules and approximation spaces. *Proc. of the 7th Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-based Systems (IPMU'98)*, 1354-1361.

Skowron, A., Stepaniuk, J., & Peters, J. F. (2001). Hierarchy of information granules. In H.D. Burkhard, L. Czaja, H.S. Nguyen, P. Starke (Eds.), *Proc. of the Workshop on Concurrency, Specification and Programming* (pp. 254-268). Academic Press.

Sogeti, C. (2016). *HPE World Quality Report*. Retrieved from https://www.capgemini.com/resources/world-quality-report-2016-17/

Soldatos, J., Kefalakis, N., Hauswirth, M., Serrano, M., Calbimonte, J.-P., Riahi, M., & … . (2015). Openiot: Open source internet-of-things in the cloud. In *Interoperability and open-source solutions for the internet of things* (pp. 13–25). Cham: Springer.

Sommerville, I. (n.d.). *Software Engineering* (7th ed.). Pearson Education Publication.

Song, J., Kunz, A., Schmidt, M., & Szczytowski, P. (2014). Connecting and managing M2M devices in the future Internet. *Mobile Networks and Applications*, *19*(1), 4–17. doi:10.100711036-013-0480-9

Sotiriadis, S., Petrakis Euripides, G.M., Covaci, S., Zampognaro, P., Georga, E., & Thuemmler, C. (2013). *An architecture for designing Future Internet (FI) applications insensitive domains: Expressing the Software to data paradigm by utilizing hybrid cloud technology*. DOI:. doi:10.1109/BIBE.2013.6701578

Srimuruganandam, B., & Nagendra, S. S. (2015). ANN-based PM prediction model for assessing the temporal variability of PM10, PM2. 5 and PM1 concentrations at an urban roadway. *International Journal of Environmental Engineering*, *7*(1), 60–89. doi:10.1504/IJEE.2015.069266

Stankovic, J. A. (2014). Research Directions for the Internet of Things. *IEEE Internet of Things J.*, *1*(1), 3–9. doi:10.1109/JIOT.2014.2312291

Stergiou. Psannis, Kim, & Gupta. (2018). Secure integration of IoT and Cloud Computing. Future Generation Computer Systems, 78(3), 964-975.

T¨onjes, R., Reetz, E. S., Moessner, K., & Barnaghi, P. M. (2012). A test-driven approach for life cycle management of internet of things enabled services. Proceedings of Future Network and Mobile Summit, 1–8.

Taivalsaari, A., & Mikkonen, T. (2017). A roadmap to the programmable world: Software challenges in the IoT era. *IEEE Software*, *1*(1), 72–80. doi:10.1109/MS.2017.26

Tan, Z., Jamdagni, A., He, X., Nanda, P., & Liu, R. P. (2013). A system for Denial-of-Service attack detection based on multivariate correlation analysis. *IEEE Transactions on Parallel and Distributed Systems*, *25*(2), 447–456.

Tecnalia, Inspearit, Favaro, & Taneja. (2017). Software Engineering For Internet of Things. IEEE Software, 24-28.

TestingWhiz. (2018). *How Test Automation can be Helpful for IoT Applications*. Retrieved from https://www.testing-whiz.com/blog/how-test-automation-can-be-helpful-for-iot-applications

Thangavel, D., Ma, X., Valera, A., Tan, H., & Tan, C. K. (2014). Performance evaluation of MQTT and CoAP via a common middleware. *IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, 1-6. 10.1109/ISSNIP.2014.6827678

Tillmann, A. M. (2015). On the Computational Intractability of Exact and Approximate Dictionary Learning. *IEEE Signal Processing Letters*, *22*(1), 45–49. doi:10.1109/LSP.2014.2345761

Uckelmann, D., Harrison, M., & Michahelles, F. (2011). *An Architectural Approach Towards the Future Internet of Things*. Architecting the Internet of Things. doi:10.1007/978-3-642-19157-2

Urbanowicz, R. J., & Moore, J. H. (2009). Learning Classifier Systems: A Complete Introduction, Review, and Roadmap. *Journal of Artificial Evolution and Applications*, 1–25. doi:10.1155/2009/736398

Vandikas, K., & Tsiatsis, V. (2014). Performance evaluation of an IoT platform. *Proceedings - International Conference on Next Generation Mobile Applications, Services and Technologies*, 141–146.

Visoottiviseth, V., Akarasiriwong, P., Chaiyasart, S., & Chotivatunyu, S. (2017). PENTOS: Penetration testing tool for Internet of Thing devices. TENCON 2017 - 2017 IEEE Region 10 Conference, 2279-2284.

Vujović, V., & Maksimović, M. (2015). Raspberry Pi as a Sensor Web node for home automation. *Computers & Electrical Engineering*, *44*, 153–171. doi:10.1016/j.compeleceng.2015.01.019

W3C-Group. (2016). *Direct to device connectivity in the internet of things*. Retrieved from https://www.w3.org/WoT/

Wang, De, Toenjes, Reetz, & Moessner. (2012). A comprehensive ontology for knowledge representation in the internet of things. In *Proceedings of the 11th International Conference on Trust, Security and Privacy in Computing and Communications*, (pp. 1793–1798). IEEE.

Watson Internet of Things. (2016). Retrieved from IBMwww.ibm.com/internet-ofthings

Weyrich & Ebert. (2016). *Reference Architectures for the Internet of Things. IEEE Software, 33(1).*

What are known useful and misleading memes in the DevOps culture? (2017). Retrieved from devops.stackexchange.com

What is DevOps? (2014). NewRelic.com.

Whitehouse, K., Sharp, C., Brewer, E., & Culler, D. (2004). Hood: a neighborhood abstraction for sensor networks. In *Proceedings of the 2nd international conference on Mobile systems, applications, and services* (pp. 99-110). ACM. 10.1145/990064.990079

Wittstock, V., Lorenz, M., Wittstock, E., & Pürzel, F. (2012). A Framework for User Tests in a Virtual Environment. Advances in Visual Computing, 358-367.

Worthy, P., Matthews, B., & Viller, S. (2016). Trust me: doubts and concerns living with the Internet of Things. *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*, 427-434. 10.1145/2901790.2901890

Xiao, L., Li, Y., Han, G., Liu, G., & Zhuang, W. (2016). PHY-layer spoofing detection with reinforcement learning in wireless networks. *IEEE Transactions on Vehicular Technology*, *65*(12), 10037–10047. doi:10.1109/TVT.2016.2524258

Xiao, L., Xie, C., Chen, T., Dai, H., & Poor, H. V. (2016). A mobile offloading game against smart attacks. *IEEE Access: Practical Innovations, Open Solutions*, *4*, 2281–2291. doi:10.1109/ACCESS.2016.2565198

Xu, T., Wendt, J. B., & Potkonjak, M. (2014). Security of IoT systems: Design challenges and opportunities. *Proceedings of the 2014 IEEE/ACM International Conference on Computer-Aided Design, IEEE*, 417-423.

Yao, L., Sheng, Q. Z., & Dustdar, S. (2015). Web-based management of the internet of things. *IEEE Internet Computing*, *19*(4), 60–67. doi:10.1109/MIC.2015.77

Yim, S. (2012). *Public Health Impacts of Combustion Emissions in the United Kingdom*. Cambridge, MA: Department of Aeronautics and Astronautics, Massachusetts Institute of Technology.

Yue, Q., & Maode, M. (2015). An authentication and key establishment scheme to enhance security for M2M in 6LoWPANs. *2015 IEEE International Conference on Communication Workshop (ICCW)*, 2671–6

Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, *8*(3), 338–353. doi:10.1016/S0019-9958(65)90241-X

Zambonelli, F. (2006). *Towards a General Software Engineering Methodology for the Internet of Things*. Academic Press.

Zambonelli, F. (n.d.). *Towards a discipline of IoT-Oriented Software Engineering*. Retrieved from https://zdoc.site/towards-a-discipline-of-iot-oriented-software-engineering.html

Zancul, Takey, Barquet, Kuwabara, Miguel, & Rozenfield. (2016). Business process support for IoT based product-service systems (PSS). *Business Process Management Journal*, *22*(2), 305–323. doi:10.1108/BPMJ-05-2015-0078

Zhang, Zhan, Lin, Chen, Gong, Zhong, … Shi. (2011). Evolutionary Computation Meets Machine Learning: A Survey. *IEEE Computational Intelligence Magazine, 6*(4), 68–75. doi:. doi:10.1109/mci.2011.942584

Zheng, Y., Liu, F., & Hsieh, H. P. (2013, August). U-air: When urban air quality inference meets big data. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 1436-1444). ACM. 10.1145/2487575.2488188

Zhu, C., Leung, V.C.M., Shu, L., & Ngai, E.C.H. (2015). Green Internet of Things for Smart World. *IEEE Access, 3*, 2151–2162.

Ziegler, S., Crettaz, C., Ladid, L., Krco, S., Pokric, B., Skarmeta, A. F., & Jung, M. (2013). Lecture Notes in Computer Science: Vol. 161-172. *IoT6 Moving to an IPv6-based future IoT. The Future Internet Assembly.*

Related References

To continue our tradition of advancing information science and technology research, we have compiled a list of recommended IGI Global readings. These references will provide additional information and guidance to further enrich your knowledge and assist you with your own research and future publications.

Aasi, P., Rusu, L., & Vieru, D. (2017). The Role of Culture in IT Governance Five Focus Areas: A Literature Review. *International Journal of IT/Business Alignment and Governance, 8*(2), 42-61. doi:10.4018/IJITBAG.2017070103

Abdrabo, A. A. (2018). Egypt's Knowledge-Based Development: Opportunities, Challenges, and Future Possibilities. In A. Alraouf (Ed.), *Knowledge-Based Urban Development in the Middle East* (pp. 80–101). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-3734-2.ch005

Abu Doush, I., & Alhami, I. (2018). Evaluating the Accessibility of Computer Laboratories, Libraries, and Websites in Jordanian Universities and Colleges. *International Journal of Information Systems and Social Change*, *9*(2), 44–60. doi:10.4018/IJISSC.2018040104

Adeboye, A. (2016). Perceived Use and Acceptance of Cloud Enterprise Resource Planning (ERP) Implementation in the Manufacturing Industries. *International Journal of Strategic Information Technology and Applications*, *7*(3), 24–40. doi:10.4018/IJSITA.2016070102

Adegbore, A. M., Quadri, M. O., & Oyewo, O. R. (2018). A Theoretical Approach to the Adoption of Electronic Resource Management Systems (ERMS) in Nigerian University Libraries. In A. Tella & T. Kwanya (Eds.), *Handbook of Research on Managing Intellectual Property in Digital Libraries* (pp. 292–311). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-3093-0.ch015

Adhikari, M., & Roy, D. (2016). Green Computing. In G. Deka, G. Siddesh, K. Srinivasa, & L. Patnaik (Eds.), *Emerging Research Surrounding Power Consumption and Performance Issues in Utility Computing* (pp. 84–108). Hershey, PA: IGI Global. doi:10.4018/978-1-4666-8853-7.ch005

Afolabi, O. A. (2018). Myths and Challenges of Building an Effective Digital Library in Developing Nations: An African Perspective. In A. Tella & T. Kwanya (Eds.), *Handbook of Research on Managing Intellectual Property in Digital Libraries* (pp. 51–79). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-3093-0.ch004

Agarwal, R., Singh, A., & Sen, S. (2016). Role of Molecular Docking in Computer-Aided Drug Design and Development. In S. Dastmalchi, M. Hamzeh-Mivehroud, & B. Sokouti (Eds.), *Applied Case Studies and Solutions in Molecular Docking-Based Drug Design* (pp. 1–28). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-0362-0.ch001

Ali, O., & Soar, J. (2016). Technology Innovation Adoption Theories. In L. Al-Hakim, X. Wu, A. Koronios, & Y. Shou (Eds.), *Handbook of Research on Driving Competitive Advantage through Sustainable, Lean, and Disruptive Innovation* (pp. 1–38). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-0135-0.ch001

Alsharo, M. (2017). Attitudes Towards Cloud Computing Adoption in Emerging Economies. *International Journal of Cloud Applications and Computing*, *7*(3), 44–58. doi:10.4018/IJCAC.2017070102

Amer, T. S., & Johnson, T. L. (2016). Information Technology Progress Indicators: Temporal Expectancy, User Preference, and the Perception of Process Duration. *International Journal of Technology and Human Interaction*, *12*(4), 1–14. doi:10.4018/IJTHI.2016100101

Amer, T. S., & Johnson, T. L. (2017). Information Technology Progress Indicators: Research Employing Psychological Frameworks. In A. Mesquita (Ed.), *Research Paradigms and Contemporary Perspectives on Human-Technology Interaction* (pp. 168–186). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-1868-6.ch008

Anchugam, C. V., & Thangadurai, K. (2016). Introduction to Network Security. In D. G., M. Singh, & M. Jayanthi (Eds.), Network Security Attacks and Countermeasures (pp. 1-48). Hershey, PA: IGI Global. doi:10.4018/978-1-4666-8761-5.ch001

Anchugam, C. V., & Thangadurai, K. (2016). Classification of Network Attacks and Countermeasures of Different Attacks. In D. G., M. Singh, & M. Jayanthi (Eds.), Network Security Attacks and Countermeasures (pp. 115-156). Hershey, PA: IGI Global. doi:10.4018/978-1-4666-8761-5.ch004

Anohah, E. (2016). Pedagogy and Design of Online Learning Environment in Computer Science Education for High Schools. *International Journal of Online Pedagogy and Course Design*, *6*(3), 39–51. doi:10.4018/IJOPCD.2016070104

Anohah, E. (2017). Paradigm and Architecture of Computing Augmented Learning Management System for Computer Science Education. *International Journal of Online Pedagogy and Course Design*, *7*(2), 60–70. doi:10.4018/IJOPCD.2017040105

Anohah, E., & Suhonen, J. (2017). Trends of Mobile Learning in Computing Education from 2006 to 2014: A Systematic Review of Research Publications. *International Journal of Mobile and Blended Learning*, *9*(1), 16–33. doi:10.4018/IJMBL.2017010102

Assis-Hassid, S., Heart, T., Reychav, I., & Pliskin, J. S. (2016). Modelling Factors Affecting Patient-Doctor-Computer Communication in Primary Care. *International Journal of Reliable and Quality E-Healthcare*, *5*(1), 1–17. doi:10.4018/IJRQEH.2016010101

Bailey, E. K. (2017). Applying Learning Theories to Computer Technology Supported Instruction. In M. Grassetti & S. Brookby (Eds.), *Advancing Next-Generation Teacher Education through Digital Tools and Applications* (pp. 61–81). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-0965-3.ch004

Balasubramanian, K. (2016). Attacks on Online Banking and Commerce. In K. Balasubramanian, K. Mala, & M. Rajakani (Eds.), *Cryptographic Solutions for Secure Online Banking and Commerce* (pp. 1–19). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-0273-9.ch001

Baldwin, S., Opoku-Agyemang, K., & Roy, D. (2016). Games People Play: A Trilateral Collaboration Researching Computer Gaming across Cultures. In K. Valentine & L. Jensen (Eds.), *Examining the Evolution of Gaming and Its Impact on Social, Cultural, and Political Perspectives* (pp. 364–376). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-0261-6.ch017

Related References

Banerjee, S., Sing, T. Y., Chowdhury, A. R., & Anwar, H. (2018). Let's Go Green: Towards a Taxonomy of Green Computing Enablers for Business Sustainability. In M. Khosrow-Pour (Ed.), *Green Computing Strategies for Competitive Advantage and Business Sustainability* (pp. 89–109). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-5017-4.ch005

Basham, R. (2018). Information Science and Technology in Crisis Response and Management. In M. Khosrow-Pour, D.B.A. (Ed.), Encyclopedia of Information Science and Technology, Fourth Edition (pp. 1407-1418). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2255-3.ch121

Batyashe, T., & Iyamu, T. (2018). Architectural Framework for the Implementation of Information Technology Governance in Organisations. In M. Khosrow-Pour, D.B.A. (Ed.), Encyclopedia of Information Science and Technology, Fourth Edition (pp. 810-819). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2255-3.ch070

Bekleyen, N., & Çelik, S. (2017). Attitudes of Adult EFL Learners towards Preparing for a Language Test via CALL. In D. Tafazoli & M. Romero (Eds.), *Multiculturalism and Technology-Enhanced Language Learning* (pp. 214–229). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-1882-2.ch013

Bennett, A., Eglash, R., Lachney, M., & Babbitt, W. (2016). Design Agency: Diversifying Computer Science at the Intersections of Creativity and Culture. In M. Raisinghani (Ed.), *Revolutionizing Education through Web-Based Instruction* (pp. 35–56). Hershey, PA: IGI Global. doi:10.4018/978-1-4666-9932-8.ch003

Bergeron, F., Croteau, A., Uwizeyemungu, S., & Raymond, L. (2017). A Framework for Research on Information Technology Governance in SMEs. In S. De Haes & W. Van Grembergen (Eds.), *Strategic IT Governance and Alignment in Business Settings* (pp. 53–81). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-0861-8.ch003

Bhatt, G. D., Wang, Z., & Rodger, J. A. (2017). Information Systems Capabilities and Their Effects on Competitive Advantages: A Study of Chinese Companies. *Information Resources Management Journal*, *30*(3), 41–57. doi:10.4018/IRMJ.2017070103

Bogdanoski, M., Stoilkovski, M., & Risteski, A. (2016). Novel First Responder Digital Forensics Tool as a Support to Law Enforcement. In M. Hadji-Janev & M. Bogdanoski (Eds.), *Handbook of Research on Civil Society and National Security in the Era of Cyber Warfare* (pp. 352–376). Hershey, PA: IGI Global. doi:10.4018/978-1-4666-8793-6.ch016

Boontarig, W., Papasratorn, B., & Chutimaskul, W. (2016). The Unified Model for Acceptance and Use of Health Information on Online Social Networks: Evidence from Thailand. *International Journal of E-Health and Medical Communications*, *7*(1), 31–47. doi:10.4018/IJEHMC.2016010102

Brown, S., & Yuan, X. (2016). Techniques for Retaining Computer Science Students at Historical Black Colleges and Universities. In C. Prince & R. Ford (Eds.), *Setting a New Agenda for Student Engagement and Retention in Historically Black Colleges and Universities* (pp. 251–268). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-0308-8.ch014

Burcoff, A., & Shamir, L. (2017). Computer Analysis of Pablo Picasso's Artistic Style. *International Journal of Art, Culture and Design Technologies*, *6*(1), 1–18. doi:10.4018/IJACDT.2017010101

Byker, E. J. (2017). I Play I Learn: Introducing Technological Play Theory. In C. Martin & D. Polly (Eds.), *Handbook of Research on Teacher Education and Professional Development* (pp. 297–306). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-1067-3.ch016

Calongne, C. M., Stricker, A. G., Truman, B., & Arenas, F. J. (2017). Cognitive Apprenticeship and Computer Science Education in Cyberspace: Reimagining the Past. In A. Stricker, C. Calongne, B. Truman, & F. Arenas (Eds.), *Integrating an Awareness of Selfhood and Society into Virtual Learning* (pp. 180–197). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2182-2.ch013

Carlton, E. L., Holsinger, J. W. Jr, & Anunobi, N. (2016). Physician Engagement with Health Information Technology: Implications for Practice and Professionalism. *International Journal of Computers in Clinical Practice*, *1*(2), 51–73. doi:10.4018/IJCCP.2016070103

Carneiro, A. D. (2017). Defending Information Networks in Cyberspace: Some Notes on Security Needs. In M. Dawson, D. Kisku, P. Gupta, J. Sing, & W. Li (Eds.), Developing Next-Generation Countermeasures for Homeland Security Threat Prevention (pp. 354-375). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-0703-1.ch016

Cavalcanti, J. C. (2016). The New "ABC" of ICTs (Analytics + Big Data + Cloud Computing): A Complex Trade-Off between IT and CT Costs. In J. Martins & A. Molnar (Eds.), *Handbook of Research on Innovations in Information Retrieval, Analysis, and Management* (pp. 152–186). Hershey, PA: IGI Global. doi:10.4018/978-1-4666-8833-9.ch006

Chase, J. P., & Yan, Z. (2017). Affect in Statistics Cognition. In *Assessing and Measuring Statistics Cognition in Higher Education Online Environments: Emerging Research and Opportunities* (pp. 144–187). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2420-5.ch005

Chen, C. (2016). Effective Learning Strategies for the 21st Century: Implications for the E-Learning. In M. Anderson & C. Gavan (Eds.), *Developing Effective Educational Experiences through Learning Analytics* (pp. 143–169). Hershey, PA: IGI Global. doi:10.4018/978-1-4666-9983-0.ch006

Chen, E. T. (2016). Examining the Influence of Information Technology on Modern Health Care. In P. Manolitzas, E. Grigoroudis, N. Matsatsinis, & D. Yannacopoulos (Eds.), *Effective Methods for Modern Healthcare Service Quality and Evaluation* (pp. 110–136). Hershey, PA: IGI Global. doi:10.4018/978-1-4666-9961-8.ch006

Cimermanova, I. (2017). Computer-Assisted Learning in Slovakia. In D. Tafazoli & M. Romero (Eds.), *Multiculturalism and Technology-Enhanced Language Learning* (pp. 252–270). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-1882-2.ch015

Cipolla-Ficarra, F. V., & Cipolla-Ficarra, M. (2018). Computer Animation for Ingenious Revival. In F. Cipolla-Ficarra, M. Ficarra, M. Cipolla-Ficarra, A. Quiroga, J. Alma, & J. Carré (Eds.), *Technology-Enhanced Human Interaction in Modern Society* (pp. 159–181). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-3437-2.ch008

Cockrell, S., Damron, T. S., Melton, A. M., & Smith, A. D. (2018). Offshoring IT. In M. Khosrow-Pour, D.B.A. (Ed.), Encyclopedia of Information Science and Technology, Fourth Edition (pp. 5476-5489). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2255-3.ch476

Coffey, J. W. (2018). Logic and Proof in Computer Science: Categories and Limits of Proof Techniques. In J. Horne (Ed.), *Philosophical Perceptions on Logic and Order* (pp. 218–240). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2443-4.ch007

Dale, M. (2017). Re-Thinking the Challenges of Enterprise Architecture Implementation. In M. Tavana (Ed.), *Enterprise Information Systems and the Digitalization of Business Functions* (pp. 205–221). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2382-6.ch009

Das, A., Dasgupta, R., & Bagchi, A. (2016). Overview of Cellular Computing-Basic Principles and Applications. In J. Mandal, S. Mukhopadhyay, & T. Pal (Eds.), *Handbook of Research on Natural Computing for Optimization Problems* (pp. 637–662). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-0058-2.ch026

De Maere, K., De Haes, S., & von Kutzschenbach, M. (2017). CIO Perspectives on Organizational Learning within the Context of IT Governance. *International Journal of IT/Business Alignment and Governance, 8*(1), 32-47. doi:10.4018/IJITBAG.2017010103

Demir, K., Çaka, C., Yaman, N. D., İslamoğlu, H., & Kuzu, A. (2018). Examining the Current Definitions of Computational Thinking. In H. Ozcinar, G. Wong, & H. Ozturk (Eds.), *Teaching Computational Thinking in Primary Education* (pp. 36–64). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-3200-2.ch003

Deng, X., Hung, Y., & Lin, C. D. (2017). Design and Analysis of Computer Experiments. In S. Saha, A. Mandal, A. Narasimhamurthy, S. V, & S. Sangam (Eds.), Handbook of Research on Applied Cybernetics and Systems Science (pp. 264-279). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2498-4.ch013

Denner, J., Martinez, J., & Thiry, H. (2017). Strategies for Engaging Hispanic/Latino Youth in the US in Computer Science. In Y. Rankin & J. Thomas (Eds.), *Moving Students of Color from Consumers to Producers of Technology* (pp. 24–48). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2005-4.ch002

Devi, A. (2017). Cyber Crime and Cyber Security: A Quick Glance. In R. Kumar, P. Pattnaik, & P. Pandey (Eds.), *Detecting and Mitigating Robotic Cyber Security Risks* (pp. 160–171). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2154-9.ch011

Dores, A. R., Barbosa, F., Guerreiro, S., Almeida, I., & Carvalho, I. P. (2016). Computer-Based Neuropsychological Rehabilitation: Virtual Reality and Serious Games. In M. Cruz-Cunha, I. Miranda, R. Martinho, & R. Rijo (Eds.), *Encyclopedia of E-Health and Telemedicine* (pp. 473–485). Hershey, PA: IGI Global. doi:10.4018/978-1-4666-9978-6.ch037

Doshi, N., & Schaefer, G. (2016). Computer-Aided Analysis of Nailfold Capillaroscopy Images. In D. Fotiadis (Ed.), *Handbook of Research on Trends in the Diagnosis and Treatment of Chronic Conditions* (pp. 146–158). Hershey, PA: IGI Global. doi:10.4018/978-1-4666-8828-5.ch007

Doyle, D. J., & Fahy, P. J. (2018). Interactivity in Distance Education and Computer-Aided Learning, With Medical Education Examples. In M. Khosrow-Pour, D.B.A. (Ed.), Encyclopedia of Information Science and Technology, Fourth Edition (pp. 5829-5840). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2255-3.ch507

Elias, N. I., & Walker, T. W. (2017). Factors that Contribute to Continued Use of E-Training among Healthcare Professionals. In F. Topor (Ed.), *Handbook of Research on Individualism and Identity in the Globalized Digital Age* (pp. 403–429). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-0522-8.ch018

Eloy, S., Dias, M. S., Lopes, P. F., & Vilar, E. (2016). Digital Technologies in Architecture and Engineering: Exploring an Engaged Interaction within Curricula. In D. Fonseca & E. Redondo (Eds.), *Handbook of Research on Applied E-Learning in Engineering and Architecture Education* (pp. 368–402). Hershey, PA: IGI Global. doi:10.4018/978-1-4666-8803-2.ch017

Estrela, V. V., Magalhães, H. A., & Saotome, O. (2016). Total Variation Applications in Computer Vision. In N. Kamila (Ed.), *Handbook of Research on Emerging Perspectives in Intelligent Pattern Recognition, Analysis, and Image Processing* (pp. 41–64). Hershey, PA: IGI Global. doi:10.4018/978-1-4666-8654-0.ch002

Filipovic, N., Radovic, M., Nikolic, D. D., Saveljic, I., Milosevic, Z., Exarchos, T. P., ... Parodi, O. (2016). Computer Predictive Model for Plaque Formation and Progression in the Artery. In D. Fotiadis (Ed.), *Handbook of Research on Trends in the Diagnosis and Treatment of Chronic Conditions* (pp. 279–300). Hershey, PA: IGI Global. doi:10.4018/978-1-4666-8828-5.ch013

Fisher, R. L. (2018). Computer-Assisted Indian Matrimonial Services. In M. Khosrow-Pour, D.B.A. (Ed.), Encyclopedia of Information Science and Technology, Fourth Edition (pp. 4136-4145). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2255-3.ch358

Fleenor, H. G., & Hodhod, R. (2016). Assessment of Learning and Technology: Computer Science Education. In V. Wang (Ed.), *Handbook of Research on Learning Outcomes and Opportunities in the Digital Age* (pp. 51–78). Hershey, PA: IGI Global. doi:10.4018/978-1-4666-9577-1.ch003

García-Valcárcel, A., & Mena, J. (2016). Information Technology as a Way To Support Collaborative Learning: What In-Service Teachers Think, Know and Do. *Journal of Information Technology Research*, *9*(1), 1–17. doi:10.4018/JITR.2016010101

Gardner-McCune, C., & Jimenez, Y. (2017). Historical App Developers: Integrating CS into K-12 through Cross-Disciplinary Projects. In Y. Rankin & J. Thomas (Eds.), *Moving Students of Color from Consumers to Producers of Technology* (pp. 85–112). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2005-4.ch005

Garvey, G. P. (2016). Exploring Perception, Cognition, and Neural Pathways of Stereo Vision and the Split–Brain Human Computer Interface. In A. Ursyn (Ed.), *Knowledge Visualization and Visual Literacy in Science Education* (pp. 28–76). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-0480-1.ch002

Ghafele, R., & Gibert, B. (2018). Open Growth: The Economic Impact of Open Source Software in the USA. In M. Khosrow-Pour (Ed.), *Optimizing Contemporary Application and Processes in Open Source Software* (pp. 164–197). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-5314-4.ch007

Ghobakhloo, M., & Azar, A. (2018). Information Technology Resources, the Organizational Capability of Lean-Agile Manufacturing, and Business Performance. *Information Resources Management Journal*, *31*(2), 47–74. doi:10.4018/IRMJ.2018040103

Gianni, M., & Gotzamani, K. (2016). Integrated Management Systems and Information Management Systems: Common Threads. In P. Papajorgji, F. Pinet, A. Guimarães, & J. Papathanasiou (Eds.), *Automated Enterprise Systems for Maximizing Business Performance* (pp. 195–214). Hershey, PA: IGI Global. doi:10.4018/978-1-4666-8841-4.ch011

Gikandi, J. W. (2017). Computer-Supported Collaborative Learning and Assessment: A Strategy for Developing Online Learning Communities in Continuing Education. In J. Keengwe & G. Onchwari (Eds.), *Handbook of Research on Learner-Centered Pedagogy in Teacher Education and Professional Development* (pp. 309–333). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-0892-2.ch017

Gokhale, A. A., & Machina, K. F. (2017). Development of a Scale to Measure Attitudes toward Information Technology. In L. Tomei (Ed.), *Exploring the New Era of Technology-Infused Education* (pp. 49–64). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-1709-2.ch004

Grace, A., O'Donoghue, J., Mahony, C., Heffernan, T., Molony, D., & Carroll, T. (2016). Computerized Decision Support Systems for Multimorbidity Care: An Urgent Call for Research and Development. In M. Cruz-Cunha, I. Miranda, R. Martinho, & R. Rijo (Eds.), *Encyclopedia of E-Health and Telemedicine* (pp. 486–494). Hershey, PA: IGI Global. doi:10.4018/978-1-4666-9978-6.ch038

Gupta, A., & Singh, O. (2016). Computer Aided Modeling and Finite Element Analysis of Human Elbow. *International Journal of Biomedical and Clinical Engineering*, *5*(1), 31–38. doi:10.4018/IJBCE.2016010104

H., S. K. (2016). Classification of Cybercrimes and Punishments under the Information Technology Act, 2000. In S. Geetha, & A. Phamila (Eds.), *Combating Security Breaches and Criminal Activity in the Digital Sphere* (pp. 57-66). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-0193-0.ch004

Hafeez-Baig, A., Gururajan, R., & Wickramasinghe, N. (2017). Readiness as a Novel Construct of Readiness Acceptance Model (RAM) for the Wireless Handheld Technology. In N. Wickramasinghe (Ed.), *Handbook of Research on Healthcare Administration and Management* (pp. 578–595). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-0920-2.ch035

Hanafizadeh, P., Ghandchi, S., & Asgarimehr, M. (2017). Impact of Information Technology on Lifestyle: A Literature Review and Classification. *International Journal of Virtual Communities and Social Networking*, *9*(2), 1–23. doi:10.4018/IJVCSN.2017040101

Harlow, D. B., Dwyer, H., Hansen, A. K., Hill, C., Iveland, A., Leak, A. E., & Franklin, D. M. (2016). Computer Programming in Elementary and Middle School: Connections across Content. In M. Urban & D. Falvo (Eds.), *Improving K-12 STEM Education Outcomes through Technological Integration* (pp. 337–361). Hershey, PA: IGI Global. doi:10.4018/978-1-4666-9616-7.ch015

Haseski, H. İ., Ilic, U., & Tuğtekin, U. (2018). Computational Thinking in Educational Digital Games: An Assessment Tool Proposal. In H. Ozcinar, G. Wong, & H. Ozturk (Eds.), *Teaching Computational Thinking in Primary Education* (pp. 256–287). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-3200-2.ch013

Hee, W. J., Jalleh, G., Lai, H., & Lin, C. (2017). E-Commerce and IT Projects: Evaluation and Management Issues in Australian and Taiwanese Hospitals. *International Journal of Public Health Management and Ethics*, *2*(1), 69–90. doi:10.4018/IJPHME.2017010104

Hernandez, A. A. (2017). Green Information Technology Usage: Awareness and Practices of Philippine IT Professionals. *International Journal of Enterprise Information Systems*, *13*(4), 90–103. doi:10.4018/IJEIS.2017100106

Hernandez, A. A., & Ona, S. E. (2016). Green IT Adoption: Lessons from the Philippines Business Process Outsourcing Industry. *International Journal of Social Ecology and Sustainable Development*, *7*(1), 1–34. doi:10.4018/IJSESD.2016010101

Hernandez, M. A., Marin, E. C., Garcia-Rodriguez, J., Azorin-Lopez, J., & Cazorla, M. (2017). Automatic Learning Improves Human-Robot Interaction in Productive Environments: A Review. *International Journal of Computer Vision and Image Processing*, *7*(3), 65–75. doi:10.4018/IJCVIP.2017070106

Horne-Popp, L. M., Tessone, E. B., & Welker, J. (2018). If You Build It, They Will Come: Creating a Library Statistics Dashboard for Decision-Making. In L. Costello & M. Powers (Eds.), *Developing In-House Digital Tools in Library Spaces* (pp. 177–203). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2676-6.ch009

Hossan, C. G., & Ryan, J. C. (2016). Factors Affecting e-Government Technology Adoption Behaviour in a Voluntary Environment. *International Journal of Electronic Government Research*, *12*(1), 24–49. doi:10.4018/IJEGR.2016010102

Hu, H., Hu, P. J., & Al-Gahtani, S. S. (2017). User Acceptance of Computer Technology at Work in Arabian Culture: A Model Comparison Approach. In M. Khosrow-Pour (Ed.), *Handbook of Research on Technology Adoption, Social Policy, and Global Integration* (pp. 205–228). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2668-1.ch011

Huie, C. P. (2016). Perceptions of Business Intelligence Professionals about Factors Related to Business Intelligence input in Decision Making. *International Journal of Business Analytics*, *3*(3), 1–24. doi:10.4018/IJBAN.2016070101

Hung, S., Huang, W., Yen, D. C., Chang, S., & Lu, C. (2016). Effect of Information Service Competence and Contextual Factors on the Effectiveness of Strategic Information Systems Planning in Hospitals. *Journal of Global Information Management*, *24*(1), 14–36. doi:10.4018/JGIM.2016010102

Ifinedo, P. (2017). Using an Extended Theory of Planned Behavior to Study Nurses' Adoption of Healthcare Information Systems in Nova Scotia. *International Journal of Technology Diffusion*, *8*(1), 1–17. doi:10.4018/IJTD.2017010101

Ilie, V., & Sneha, S. (2018). A Three Country Study for Understanding Physicians' Engagement With Electronic Information Resources Pre and Post System Implementation. *Journal of Global Information Management*, *26*(2), 48–73. doi:10.4018/JGIM.2018040103

Inoue-Smith, Y. (2017). Perceived Ease in Using Technology Predicts Teacher Candidates' Preferences for Online Resources. *International Journal of Online Pedagogy and Course Design*, *7*(3), 17–28. doi:10.4018/IJOPCD.2017070102

Islam, A. A. (2016). Development and Validation of the Technology Adoption and Gratification (TAG) Model in Higher Education: A Cross-Cultural Study Between Malaysia and China. *International Journal of Technology and Human Interaction*, *12*(3), 78–105. doi:10.4018/IJTHI.2016070106

Islam, A. Y. (2017). Technology Satisfaction in an Academic Context: Moderating Effect of Gender. In A. Mesquita (Ed.), *Research Paradigms and Contemporary Perspectives on Human-Technology Interaction* (pp. 187–211). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-1868-6.ch009

Jamil, G. L., & Jamil, C. C. (2017). Information and Knowledge Management Perspective Contributions for Fashion Studies: Observing Logistics and Supply Chain Management Processes. In G. Jamil, A. Soares, & C. Pessoa (Eds.), *Handbook of Research on Information Management for Effective Logistics and Supply Chains* (pp. 199–221). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-0973-8.ch011

Jamil, G. L., Jamil, L. C., Vieira, A. A., & Xavier, A. J. (2016). Challenges in Modelling Healthcare Services: A Study Case of Information Architecture Perspectives. In G. Jamil, J. Poças Rascão, F. Ribeiro, & A. Malheiro da Silva (Eds.), *Handbook of Research on Information Architecture and Management in Modern Organizations* (pp. 1–23). Hershey, PA: IGI Global. doi:10.4018/978-1-4666-8637-3.ch001

Janakova, M. (2018). Big Data and Simulations for the Solution of Controversies in Small Businesses. In M. Khosrow-Pour, D.B.A. (Ed.), Encyclopedia of Information Science and Technology, Fourth Edition (pp. 6907-6915). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2255-3.ch598

Jha, D. G. (2016). Preparing for Information Technology Driven Changes. In S. Tiwari & L. Nafees (Eds.), *Innovative Management Education Pedagogies for Preparing Next-Generation Leaders* (pp. 258–274). Hershey, PA: IGI Global. doi:10.4018/978-1-4666-9691-4.ch015

Jhawar, A., & Garg, S. K. (2018). Logistics Improvement by Investment in Information Technology Using System Dynamics. In A. Azar & S. Vaidyanathan (Eds.), *Advances in System Dynamics and Control* (pp. 528–567). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-4077-9.ch017

Kalelioğlu, F., Gülbahar, Y., & Doğan, D. (2018). Teaching How to Think Like a Programmer: Emerging Insights. In H. Ozcinar, G. Wong, & H. Ozturk (Eds.), *Teaching Computational Thinking in Primary Education* (pp. 18–35). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-3200-2.ch002

Kamberi, S. (2017). A Girls-Only Online Virtual World Environment and its Implications for Game-Based Learning. In A. Stricker, C. Calongne, B. Truman, & F. Arenas (Eds.), *Integrating an Awareness of Selfhood and Society into Virtual Learning* (pp. 74–95). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2182-2.ch006

Kamel, S., & Rizk, N. (2017). ICT Strategy Development: From Design to Implementation – Case of Egypt. In C. Howard & K. Hargiss (Eds.), *Strategic Information Systems and Technologies in Modern Organizations* (pp. 239–257). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-1680-4.ch010

Kamel, S. H. (2018). The Potential Role of the Software Industry in Supporting Economic Development. In M. Khosrow-Pour, D.B.A. (Ed.), Encyclopedia of Information Science and Technology, Fourth Edition (pp. 7259-7269). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2255-3.ch631

Karon, R. (2016). Utilisation of Health Information Systems for Service Delivery in the Namibian Environment. In T. Iyamu & A. Tatnall (Eds.), *Maximizing Healthcare Delivery and Management through Technology Integration* (pp. 169–183). Hershey, PA: IGI Global. doi:10.4018/978-1-4666-9446-0.ch011

Kawata, S. (2018). Computer-Assisted Parallel Program Generation. In M. Khosrow-Pour, D.B.A. (Ed.), Encyclopedia of Information Science and Technology, Fourth Edition (pp. 4583-4593). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2255-3.ch398

Khanam, S., Siddiqui, J., & Talib, F. (2016). A DEMATEL Approach for Prioritizing the TQM Enablers and IT Resources in the Indian ICT Industry. *International Journal of Applied Management Sciences and Engineering*, *3*(1), 11–29. doi:10.4018/IJAMSE.2016010102

Khari, M., Shrivastava, G., Gupta, S., & Gupta, R. (2017). Role of Cyber Security in Today's Scenario. In R. Kumar, P. Pattnaik, & P. Pandey (Eds.), *Detecting and Mitigating Robotic Cyber Security Risks* (pp. 177–191). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2154-9.ch013

Khouja, M., Rodriguez, I. B., Ben Halima, Y., & Moalla, S. (2018). IT Governance in Higher Education Institutions: A Systematic Literature Review. *International Journal of Human Capital and Information Technology Professionals*, *9*(2), 52–67. doi:10.4018/IJHCITP.2018040104

Kim, S., Chang, M., Choi, N., Park, J., & Kim, H. (2016). The Direct and Indirect Effects of Computer Uses on Student Success in Math. *International Journal of Cyber Behavior, Psychology and Learning*, *6*(3), 48–64. doi:10.4018/IJCBPL.2016070104

Kiourt, C., Pavlidis, G., Koutsoudis, A., & Kalles, D. (2017). Realistic Simulation of Cultural Heritage. *International Journal of Computational Methods in Heritage Science*, *1*(1), 10–40. doi:10.4018/IJCMHS.2017010102

Korikov, A., & Krivtsov, O. (2016). System of People-Computer: On the Way of Creation of Human-Oriented Interface. In V. Mkrttchian, A. Bershadsky, A. Bozhday, M. Kataev, & S. Kataev (Eds.), *Handbook of Research on Estimation and Control Techniques in E-Learning Systems* (pp. 458–470). Hershey, PA: IGI Global. doi:10.4018/978-1-4666-9489-7.ch032

Köse, U. (2017). An Augmented-Reality-Based Intelligent Mobile Application for Open Computer Education. In G. Kurubacak & H. Altinpulluk (Eds.), *Mobile Technologies and Augmented Reality in Open Education* (pp. 154–174). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2110-5.ch008

Lahmiri, S. (2018). Information Technology Outsourcing Risk Factors and Provider Selection. In M. Gupta, R. Sharman, J. Walp, & P. Mulgund (Eds.), *Information Technology Risk Management and Compliance in Modern Organizations* (pp. 214–228). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2604-9.ch008

Landriscina, F. (2017). Computer-Supported Imagination: The Interplay Between Computer and Mental Simulation in Understanding Scientific Concepts. In I. Levin & D. Tsybulsky (Eds.), *Digital Tools and Solutions for Inquiry-Based STEM Learning* (pp. 33–60). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2525-7.ch002

Lau, S. K., Winley, G. K., Leung, N. K., Tsang, N., & Lau, S. Y. (2016). An Exploratory Study of Expectation in IT Skills in a Developing Nation: Vietnam. *Journal of Global Information Management*, *24*(1), 1–13. doi:10.4018/JGIM.2016010101

Lavranos, C., Kostagiolas, P., & Papadatos, J. (2016). Information Retrieval Technologies and the "Realities" of Music Information Seeking. In I. Deliyannis, P. Kostagiolas, & C. Banou (Eds.), *Experimental Multimedia Systems for Interactivity and Strategic Innovation* (pp. 102–121). Hershey, PA: IGI Global. doi:10.4018/978-1-4666-8659-5.ch005

Lee, W. W. (2018). Ethical Computing Continues From Problem to Solution. In M. Khosrow-Pour, D.B.A. (Ed.), Encyclopedia of Information Science and Technology, Fourth Edition (pp. 4884-4897). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2255-3.ch423

Lehto, M. (2016). Cyber Security Education and Research in the Finland's Universities and Universities of Applied Sciences. *International Journal of Cyber Warfare & Terrorism*, *6*(2), 15–31. doi:10.4018/IJCWT.2016040102

Lin, C., Jalleh, G., & Huang, Y. (2016). Evaluating and Managing Electronic Commerce and Outsourcing Projects in Hospitals. In A. Dwivedi (Ed.), *Reshaping Medical Practice and Care with Health Information Systems* (pp. 132–172). Hershey, PA: IGI Global. doi:10.4018/978-1-4666-9870-3.ch005

Lin, S., Chen, S., & Chuang, S. (2017). Perceived Innovation and Quick Response Codes in an Online-to-Offline E-Commerce Service Model. *International Journal of E-Adoption*, *9*(2), 1–16. doi:10.4018/IJEA.2017070101

Liu, M., Wang, Y., Xu, W., & Liu, L. (2017). Automated Scoring of Chinese Engineering Students' English Essays. *International Journal of Distance Education Technologies*, *15*(1), 52–68. doi:10.4018/IJDET.2017010104

Luciano, E. M., Wiedenhöft, G. C., Macadar, M. A., & Pinheiro dos Santos, F. (2016). Information Technology Governance Adoption: Understanding its Expectations Through the Lens of Organizational Citizenship. *International Journal of IT/Business Alignment and Governance, 7*(2), 22-32. doi:10.4018/IJITBAG.2016070102

Mabe, L. K., & Oladele, O. I. (2017). Application of Information Communication Technologies for Agricultural Development through Extension Services: A Review. In T. Tossy (Ed.), *Information Technology Integration for Socio-Economic Development* (pp. 52–101). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-0539-6.ch003

Manogaran, G., Thota, C., & Lopez, D. (2018). Human-Computer Interaction With Big Data Analytics. In D. Lopez & M. Durai (Eds.), *HCI Challenges and Privacy Preservation in Big Data Security* (pp. 1–22). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2863-0.ch001

Margolis, J., Goode, J., & Flapan, J. (2017). A Critical Crossroads for Computer Science for All: “Identifying Talent” or “Building Talent,” and What Difference Does It Make? In Y. Rankin & J. Thomas (Eds.), *Moving Students of Color from Consumers to Producers of Technology* (pp. 1–23). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2005-4.ch001

Mbale, J. (2018). Computer Centres Resource Cloud Elasticity-Scalability (CRECES): Copperbelt University Case Study. In S. Aljawarneh & M. Malhotra (Eds.), *Critical Research on Scalability and Security Issues in Virtual Cloud Environments* (pp. 48–70). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-3029-9.ch003

McKee, J. (2018). The Right Information: The Key to Effective Business Planning. In *Business Architectures for Risk Assessment and Strategic Planning: Emerging Research and Opportunities* (pp. 38–52). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-3392-4.ch003

Mensah, I. K., & Mi, J. (2018). Determinants of Intention to Use Local E-Government Services in Ghana: The Perspective of Local Government Workers. *International Journal of Technology Diffusion*, *9*(2), 41–60. doi:10.4018/IJTD.2018040103

Mohamed, J. H. (2018). Scientograph-Based Visualization of Computer Forensics Research Literature. In J. Jeyasekar & P. Saravanan (Eds.), *Innovations in Measuring and Evaluating Scientific Information* (pp. 148–162). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-3457-0.ch010

Moore, R. L., & Johnson, N. (2017). Earning a Seat at the Table: How IT Departments Can Partner in Organizational Change and Innovation. *International Journal of Knowledge-Based Organizations*, *7*(2), 1–12. doi:10.4018/IJKBO.2017040101

Mtebe, J. S., & Kissaka, M. M. (2016). Enhancing the Quality of Computer Science Education with MOOCs in Sub-Saharan Africa. In J. Keengwe & G. Onchwari (Eds.), *Handbook of Research on Active Learning and the Flipped Classroom Model in the Digital Age* (pp. 366–377). Hershey, PA: IGI Global. doi:10.4018/978-1-4666-9680-8.ch019

Mukul, M. K., & Bhattaharyya, S. (2017). Brain-Machine Interface: Human-Computer Interaction. In E. Noughabi, B. Raahemi, A. Albadvi, & B. Far (Eds.), *Handbook of Research on Data Science for Effective Healthcare Practice and Administration* (pp. 417–443). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2515-8.ch018

Na, L. (2017). Library and Information Science Education and Graduate Programs in Academic Libraries. In L. Ruan, Q. Zhu, & Y. Ye (Eds.), *Academic Library Development and Administration in China* (pp. 218–229). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-0550-1.ch013

Nabavi, A., Taghavi-Fard, M. T., Hanafizadeh, P., & Taghva, M. R. (2016). Information Technology Continuance Intention: A Systematic Literature Review. *International Journal of E-Business Research*, *12*(1), 58–95. doi:10.4018/IJEBR.2016010104

Nath, R., & Murthy, V. N. (2018). What Accounts for the Differences in Internet Diffusion Rates Around the World? In M. Khosrow-Pour, D.B.A. (Ed.), Encyclopedia of Information Science and Technology, Fourth Edition (pp. 8095-8104). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2255-3.ch705

Nedelko, Z., & Potocan, V. (2018). The Role of Emerging Information Technologies for Supporting Supply Chain Management. In M. Khosrow-Pour, D.B.A. (Ed.), Encyclopedia of Information Science and Technology, Fourth Edition (pp. 5559-5569). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2255-3.ch483

Ngafeeson, M. N. (2018). User Resistance to Health Information Technology. In M. Khosrow-Pour, D.B.A. (Ed.), Encyclopedia of Information Science and Technology, Fourth Edition (pp. 3816-3825). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2255-3.ch331

Nozari, H., Najafi, S. E., Jafari-Eskandari, M., & Aliahmadi, A. (2016). Providing a Model for Virtual Project Management with an Emphasis on IT Projects. In C. Graham (Ed.), *Strategic Management and Leadership for Systems Development in Virtual Spaces* (pp. 43–63). Hershey, PA: IGI Global. doi:10.4018/978-1-4666-9688-4.ch003

Nurdin, N., Stockdale, R., & Scheepers, H. (2016). Influence of Organizational Factors in the Sustainability of E-Government: A Case Study of Local E-Government in Indonesia. In I. Sodhi (Ed.), *Trends, Prospects, and Challenges in Asian E-Governance* (pp. 281–323). Hershey, PA: IGI Global. doi:10.4018/978-1-4666-9536-8.ch014

Odagiri, K. (2017). Introduction of Individual Technology to Constitute the Current Internet. In *Strategic Policy-Based Network Management in Contemporary Organizations* (pp. 20–96). Hershey, PA: IGI Global. doi:10.4018/978-1-68318-003-6.ch003

Okike, E. U. (2018). Computer Science and Prison Education. In I. Biao (Ed.), *Strategic Learning Ideologies in Prison Education Programs* (pp. 246–264). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2909-5.ch012

Olelewe, C. J., & Nwafor, I. P. (2017). Level of Computer Appreciation Skills Acquired for Sustainable Development by Secondary School Students in Nsukka LGA of Enugu State, Nigeria. In C. Ayo & V. Mbarika (Eds.), *Sustainable ICT Adoption and Integration for Socio-Economic Development* (pp. 214–233). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2565-3.ch010

Oliveira, M., Maçada, A. C., Curado, C., & Nodari, F. (2017). Infrastructure Profiles and Knowledge Sharing. *International Journal of Technology and Human Interaction*, *13*(3), 1–12. doi:10.4018/IJTHI.2017070101

Otarkhani, A., Shokouhyar, S., & Pour, S. S. (2017). Analyzing the Impact of Governance of Enterprise IT on Hospital Performance: Tehran's (Iran) Hospitals – A Case Study. *International Journal of Healthcare Information Systems and Informatics*, *12*(3), 1–20. doi:10.4018/IJHISI.2017070101

Otunla, A. O., & Amuda, C. O. (2018). Nigerian Undergraduate Students' Computer Competencies and Use of Information Technology Tools and Resources for Study Skills and Habits' Enhancement. In M. Khosrow-Pour, D.B.A. (Ed.), Encyclopedia of Information Science and Technology, Fourth Edition (pp. 2303-2313). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2255-3.ch200

Özçınar, H. (2018). A Brief Discussion on Incentives and Barriers to Computational Thinking Education. In H. Ozcinar, G. Wong, & H. Ozturk (Eds.), *Teaching Computational Thinking in Primary Education* (pp. 1–17). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-3200-2.ch001

Pandey, J. M., Garg, S., Mishra, P., & Mishra, B. P. (2017). Computer Based Psychological Interventions: Subject to the Efficacy of Psychological Services. *International Journal of Computers in Clinical Practice*, *2*(1), 25–33. doi:10.4018/IJCCP.2017010102

Parry, V. K., & Lind, M. L. (2016). Alignment of Business Strategy and Information Technology Considering Information Technology Governance, Project Portfolio Control, and Risk Management. *International Journal of Information Technology Project Management*, *7*(4), 21–37. doi:10.4018/IJITPM.2016100102

Patro, C. (2017). Impulsion of Information Technology on Human Resource Practices. In P. Ordóñez de Pablos (Ed.), *Managerial Strategies and Solutions for Business Success in Asia* (pp. 231–254). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-1886-0.ch013

Patro, C. S., & Raghunath, K. M. (2017). Information Technology Paraphernalia for Supply Chain Management Decisions. In M. Tavana (Ed.), *Enterprise Information Systems and the Digitalization of Business Functions* (pp. 294–320). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2382-6.ch014

Paul, P. K. (2016). Cloud Computing: An Agent of Promoting Interdisciplinary Sciences, Especially Information Science and I-Schools – Emerging Techno-Educational Scenario. In L. Chao (Ed.), *Handbook of Research on Cloud-Based STEM Education for Improved Learning Outcomes* (pp. 247–258). Hershey, PA: IGI Global. doi:10.4018/978-1-4666-9924-3.ch016

Paul, P. K. (2018). The Context of IST for Solid Information Retrieval and Infrastructure Building: Study of Developing Country. *International Journal of Information Retrieval Research*, *8*(1), 86–100. doi:10.4018/IJIRR.2018010106

Paul, P. K., & Chatterjee, D. (2018). iSchools Promoting "Information Science and Technology" (IST) Domain Towards Community, Business, and Society With Contemporary Worldwide Trend and Emerging Potentialities in India. In M. Khosrow-Pour, D.B.A. (Ed.), Encyclopedia of Information Science and Technology, Fourth Edition (pp. 4723-4735). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2255-3.ch410

Pessoa, C. R., & Marques, M. E. (2017). Information Technology and Communication Management in Supply Chain Management. In G. Jamil, A. Soares, & C. Pessoa (Eds.), *Handbook of Research on Information Management for Effective Logistics and Supply Chains* (pp. 23–33). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-0973-8.ch002

Pineda, R. G. (2016). Where the Interaction Is Not: Reflections on the Philosophy of Human-Computer Interaction. *International Journal of Art, Culture and Design Technologies*, *5*(1), 1–12. doi:10.4018/IJACDT.2016010101

Pineda, R. G. (2018). Remediating Interaction: Towards a Philosophy of Human-Computer Relationship. In M. Khosrow-Pour (Ed.), *Enhancing Art, Culture, and Design With Technological Integration* (pp. 75–98). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-5023-5.ch004

Poikela, P., & Vuojärvi, H. (2016). Learning ICT-Mediated Communication through Computer-Based Simulations. In M. Cruz-Cunha, I. Miranda, R. Martinho, & R. Rijo (Eds.), *Encyclopedia of E-Health and Telemedicine* (pp. 674–687). Hershey, PA: IGI Global. doi:10.4018/978-1-4666-9978-6.ch052

Qian, Y. (2017). Computer Simulation in Higher Education: Affordances, Opportunities, and Outcomes. In P. Vu, S. Fredrickson, & C. Moore (Eds.), *Handbook of Research on Innovative Pedagogies and Technologies for Online Learning in Higher Education* (pp. 236–262). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-1851-8.ch011

Radant, O., Colomo-Palacios, R., & Stantchev, V. (2016). Factors for the Management of Scarce Human Resources and Highly Skilled Employees in IT-Departments: A Systematic Review. *Journal of Information Technology Research*, *9*(1), 65–82. doi:10.4018/JITR.2016010105

Rahman, N. (2016). Toward Achieving Environmental Sustainability in the Computer Industry. *International Journal of Green Computing*, *7*(1), 37–54. doi:10.4018/IJGC.2016010103

Rahman, N. (2017). Lessons from a Successful Data Warehousing Project Management. *International Journal of Information Technology Project Management*, *8*(4), 30–45. doi:10.4018/IJITPM.2017100103

Rahman, N. (2018). Environmental Sustainability in the Computer Industry for Competitive Advantage. In M. Khosrow-Pour (Ed.), *Green Computing Strategies for Competitive Advantage and Business Sustainability* (pp. 110–130). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-5017-4.ch006

Rajh, A., & Pavetic, T. (2017). Computer Generated Description as the Required Digital Competence in Archival Profession. *International Journal of Digital Literacy and Digital Competence*, *8*(1), 36–49. doi:10.4018/IJDLDC.2017010103

Raman, A., & Goyal, D. P. (2017). Extending IMPLEMENT Framework for Enterprise Information Systems Implementation to Information System Innovation. In M. Tavana (Ed.), *Enterprise Information Systems and the Digitalization of Business Functions* (pp. 137–177). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2382-6.ch007

Rao, Y. S., Rauta, A. K., Saini, H., & Panda, T. C. (2017). Mathematical Model for Cyber Attack in Computer Network. *International Journal of Business Data Communications and Networking*, *13*(1), 58–65. doi:10.4018/IJBDCN.2017010105

Rapaport, W. J. (2018). Syntactic Semantics and the Proper Treatment of Computationalism. In M. Danesi (Ed.), *Empirical Research on Semiotics and Visual Rhetoric* (pp. 128–176). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-5622-0.ch007

Raut, R., Priyadarshinee, P., & Jha, M. (2017). Understanding the Mediation Effect of Cloud Computing Adoption in Indian Organization: Integrating TAM-TOE- Risk Model. *International Journal of Service Science, Management, Engineering, and Technology*, *8*(3), 40–59. doi:10.4018/IJSSMET.2017070103

Regan, E. A., & Wang, J. (2016). Realizing the Value of EHR Systems Critical Success Factors. *International Journal of Healthcare Information Systems and Informatics*, *11*(3), 1–18. doi:10.4018/IJHISI.2016070101

Rezaie, S., Mirabedini, S. J., & Abtahi, A. (2018). Designing a Model for Implementation of Business Intelligence in the Banking Industry. *International Journal of Enterprise Information Systems*, *14*(1), 77–103. doi:10.4018/IJEIS.2018010105

Rezende, D. A. (2016). Digital City Projects: Information and Public Services Offered by Chicago (USA) and Curitiba (Brazil). *International Journal of Knowledge Society Research*, *7*(3), 16–30. doi:10.4018/IJKSR.2016070102

Rezende, D. A. (2018). Strategic Digital City Projects: Innovative Information and Public Services Offered by Chicago (USA) and Curitiba (Brazil). In M. Lytras, L. Daniela, & A. Visvizi (Eds.), *Enhancing Knowledge Discovery and Innovation in the Digital Era* (pp. 204–223). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-4191-2.ch012

Riabov, V. V. (2016). Teaching Online Computer-Science Courses in LMS and Cloud Environment. *International Journal of Quality Assurance in Engineering and Technology Education*, *5*(4), 12–41. doi:10.4018/IJQAETE.2016100102

Ricordel, V., Wang, J., Da Silva, M. P., & Le Callet, P. (2016). 2D and 3D Visual Attention for Computer Vision: Concepts, Measurement, and Modeling. In R. Pal (Ed.), *Innovative Research in Attention Modeling and Computer Vision Applications* (pp. 1–44). Hershey, PA: IGI Global. doi:10.4018/978-1-4666-8723-3.ch001

Rodriguez, A., Rico-Diaz, A. J., Rabuñal, J. R., & Gestal, M. (2017). Fish Tracking with Computer Vision Techniques: An Application to Vertical Slot Fishways. In M. S., & V. V. (Eds.), Multi-Core Computer Vision and Image Processing for Intelligent Applications (pp. 74-104). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-0889-2.ch003

Romero, J. A. (2018). Sustainable Advantages of Business Value of Information Technology. In M. Khosrow-Pour, D.B.A. (Ed.), Encyclopedia of Information Science and Technology, Fourth Edition (pp. 923-929). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2255-3.ch079

Romero, J. A. (2018). The Always-On Business Model and Competitive Advantage. In N. Bajgoric (Ed.), *Always-On Enterprise Information Systems for Modern Organizations* (pp. 23–40). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-3704-5.ch002

Rosen, Y. (2018). Computer Agent Technologies in Collaborative Learning and Assessment. In M. Khosrow-Pour, D.B.A. (Ed.), Encyclopedia of Information Science and Technology, Fourth Edition (pp. 2402-2410). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2255-3.ch209

Rosen, Y., & Mosharraf, M. (2016). Computer Agent Technologies in Collaborative Assessments. In Y. Rosen, S. Ferrara, & M. Mosharraf (Eds.), *Handbook of Research on Technology Tools for Real-World Skill Development* (pp. 319–343). Hershey, PA: IGI Global. doi:10.4018/978-1-4666-9441-5.ch012

Roy, D. (2018). Success Factors of Adoption of Mobile Applications in Rural India: Effect of Service Characteristics on Conceptual Model. In M. Khosrow-Pour (Ed.), *Green Computing Strategies for Competitive Advantage and Business Sustainability* (pp. 211–238). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-5017-4.ch010

Ruffin, T. R. (2016). Health Information Technology and Change. In V. Wang (Ed.), *Handbook of Research on Advancing Health Education through Technology* (pp. 259–285). Hershey, PA: IGI Global. doi:10.4018/978-1-4666-9494-1.ch012

Ruffin, T. R. (2016). Health Information Technology and Quality Management. *International Journal of Information Communication Technologies and Human Development*, *8*(4), 56–72. doi:10.4018/IJICTHD.2016100105

Ruffin, T. R., & Hawkins, D. P. (2018). Trends in Health Care Information Technology and Informatics. In M. Khosrow-Pour, D.B.A. (Ed.), Encyclopedia of Information Science and Technology, Fourth Edition (pp. 3805-3815). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2255-3.ch330

Safari, M. R., & Jiang, Q. (2018). The Theory and Practice of IT Governance Maturity and Strategies Alignment: Evidence From Banking Industry. *Journal of Global Information Management*, *26*(2), 127–146. doi:10.4018/JGIM.2018040106

Sahin, H. B., & Anagun, S. S. (2018). Educational Computer Games in Math Teaching: A Learning Culture. In E. Toprak & E. Kumtepe (Eds.), *Supporting Multiculturalism in Open and Distance Learning Spaces* (pp. 249–280). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-3076-3.ch013

Sanna, A., & Valpreda, F. (2017). An Assessment of the Impact of a Collaborative Didactic Approach and Students' Background in Teaching Computer Animation. *International Journal of Information and Communication Technology Education*, *13*(4), 1–16. doi:10.4018/IJICTE.2017100101

Savita, K., Dominic, P., & Ramayah, T. (2016). The Drivers, Practices and Outcomes of Green Supply Chain Management: Insights from ISO14001 Manufacturing Firms in Malaysia. *International Journal of Information Systems and Supply Chain Management*, *9*(2), 35–60. doi:10.4018/IJISSCM.2016040103

Scott, A., Martin, A., & McAlear, F. (2017). Enhancing Participation in Computer Science among Girls of Color: An Examination of a Preparatory AP Computer Science Intervention. In Y. Rankin & J. Thomas (Eds.), *Moving Students of Color from Consumers to Producers of Technology* (pp. 62–84). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2005-4.ch004

Shahsavandi, E., Mayah, G., & Rahbari, H. (2016). Impact of E-Government on Transparency and Corruption in Iran. In I. Sodhi (Ed.), *Trends, Prospects, and Challenges in Asian E-Governance* (pp. 75–94). Hershey, PA: IGI Global. doi:10.4018/978-1-4666-9536-8.ch004

Siddoo, V., & Wongsai, N. (2017). Factors Influencing the Adoption of ISO/IEC 29110 in Thai Government Projects: A Case Study. *International Journal of Information Technologies and Systems Approach*, *10*(1), 22–44. doi:10.4018/IJITSA.2017010102

Sidorkina, I., & Rybakov, A. (2016). Computer-Aided Design as Carrier of Set Development Changes System in E-Course Engineering. In V. Mkrttchian, A. Bershadsky, A. Bozhday, M. Kataev, & S. Kataev (Eds.), *Handbook of Research on Estimation and Control Techniques in E-Learning Systems* (pp. 500–515). Hershey, PA: IGI Global. doi:10.4018/978-1-4666-9489-7.ch035

Sidorkina, I., & Rybakov, A. (2016). Creating Model of E-Course: As an Object of Computer-Aided Design. In V. Mkrttchian, A. Bershadsky, A. Bozhday, M. Kataev, & S. Kataev (Eds.), *Handbook of Research on Estimation and Control Techniques in E-Learning Systems* (pp. 286–297). Hershey, PA: IGI Global. doi:10.4018/978-1-4666-9489-7.ch019

Simões, A. (2017). Using Game Frameworks to Teach Computer Programming. In R. Alexandre Peixoto de Queirós & M. Pinto (Eds.), *Gamification-Based E-Learning Strategies for Computer Programming Education* (pp. 221–236). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-1034-5.ch010

Sllame, A. M. (2017). Integrating LAB Work With Classes in Computer Network Courses. In H. Alphin Jr, R. Chan, & J. Lavine (Eds.), *The Future of Accessibility in International Higher Education* (pp. 253–275). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2560-8.ch015

Smirnov, A., Ponomarev, A., Shilov, N., Kashevnik, A., & Teslya, N. (2018). Ontology-Based Human-Computer Cloud for Decision Support: Architecture and Applications in Tourism. *International Journal of Embedded and Real-Time Communication Systems*, *9*(1), 1–19. doi:10.4018/IJERTCS.2018010101

Smith-Ditizio, A. A., & Smith, A. D. (2018). Computer Fraud Challenges and Its Legal Implications. In M. Khosrow-Pour, D.B.A. (Ed.), Encyclopedia of Information Science and Technology, Fourth Edition (pp. 4837-4848). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2255-3.ch419

Sohani, S. S. (2016). Job Shadowing in Information Technology Projects: A Source of Competitive Advantage. *International Journal of Information Technology Project Management*, *7*(1), 47–57. doi:10.4018/IJITPM.2016010104

Sosnin, P. (2018). Figuratively Semantic Support of Human-Computer Interactions. In *Experience-Based Human-Computer Interactions: Emerging Research and Opportunities* (pp. 244–272). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2987-3.ch008

Spinelli, R., & Benevolo, C. (2016). From Healthcare Services to E-Health Applications: A Delivery System-Based Taxonomy. In A. Dwivedi (Ed.), *Reshaping Medical Practice and Care with Health Information Systems* (pp. 205–245). Hershey, PA: IGI Global. doi:10.4018/978-1-4666-9870-3.ch007

Srinivasan, S. (2016). Overview of Clinical Trial and Pharmacovigilance Process and Areas of Application of Computer System. In P. Chakraborty & A. Nagal (Eds.), *Software Innovations in Clinical Drug Development and Safety* (pp. 1–13). Hershey, PA: IGI Global. doi:10.4018/978-1-4666-8726-4.ch001

Srisawasdi, N. (2016). Motivating Inquiry-Based Learning Through a Combination of Physical and Virtual Computer-Based Laboratory Experiments in High School Science. In M. Urban & D. Falvo (Eds.), *Improving K-12 STEM Education Outcomes through Technological Integration* (pp. 108–134). Hershey, PA: IGI Global. doi:10.4018/978-1-4666-9616-7.ch006

Stavridi, S. V., & Hamada, D. R. (2016). Children and Youth Librarians: Competencies Required in Technology-Based Environment. In J. Yap, M. Perez, M. Ayson, & G. Entico (Eds.), *Special Library Administration, Standardization and Technological Integration* (pp. 25–50). Hershey, PA: IGI Global. doi:10.4018/978-1-4666-9542-9.ch002

Sung, W., Ahn, J., Kai, S. M., Choi, A., & Black, J. B. (2016). Incorporating Touch-Based Tablets into Classroom Activities: Fostering Children's Computational Thinking through iPad Integrated Instruction. In D. Mentor (Ed.), *Handbook of Research on Mobile Learning in Contemporary Classrooms* (pp. 378–406). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-0251-7.ch019

Syväjärvi, A., Leinonen, J., Kivivirta, V., & Kesti, M. (2017). The Latitude of Information Management in Local Government: Views of Local Government Managers. *International Journal of Electronic Government Research*, *13*(1), 69–85. doi:10.4018/IJEGR.2017010105

Tanque, M., & Foxwell, H. J. (2018). Big Data and Cloud Computing: A Review of Supply Chain Capabilities and Challenges. In A. Prasad (Ed.), *Exploring the Convergence of Big Data and the Internet of Things* (pp. 1–28). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2947-7.ch001

Teixeira, A., Gomes, A., & Orvalho, J. G. (2017). Auditory Feedback in a Computer Game for Blind People. In T. Issa, P. Kommers, T. Issa, P. Isaías, & T. Issa (Eds.), *Smart Technology Applications in Business Environments* (pp. 134–158). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2492-2.ch007

Thompson, N., McGill, T., & Murray, D. (2018). Affect-Sensitive Computer Systems. In M. Khosrow-Pour, D.B.A. (Ed.), Encyclopedia of Information Science and Technology, Fourth Edition (pp. 4124-4135). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2255-3.ch357

Trad, A., & Kalpić, D. (2016). The E-Business Transformation Framework for E-Commerce Control and Monitoring Pattern. In I. Lee (Ed.), *Encyclopedia of E-Commerce Development, Implementation, and Management* (pp. 754–777). Hershey, PA: IGI Global. doi:10.4018/978-1-4666-9787-4.ch053

Triberti, S., Brivio, E., & Galimberti, C. (2018). On Social Presence: Theories, Methodologies, and Guidelines for the Innovative Contexts of Computer-Mediated Learning. In M. Marmon (Ed.), *Enhancing Social Presence in Online Learning Environments* (pp. 20–41). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-3229-3.ch002

Tripathy, B. K. T. R., S., & Mohanty, R. K. (2018). Memetic Algorithms and Their Applications in Computer Science. In S. Dash, B. Tripathy, & A. Rahman (Eds.), Handbook of Research on Modeling, Analysis, and Application of Nature-Inspired Metaheuristic Algorithms (pp. 73-93). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2857-9.ch004

Turulja, L., & Bajgoric, N. (2017). Human Resource Management IT and Global Economy Perspective: Global Human Resource Information Systems. In M. Khosrow-Pour (Ed.), *Handbook of Research on Technology Adoption, Social Policy, and Global Integration* (pp. 377–394). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2668-1.ch018

Unwin, D. W., Sanzogni, L., & Sandhu, K. (2017). Developing and Measuring the Business Case for Health Information Technology. In K. Moahi, K. Bwalya, & P. Sebina (Eds.), *Health Information Systems and the Advancement of Medical Practice in Developing Countries* (pp. 262–290). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2262-1.ch015

Vadhanam, B. R. S., M., Sugumaran, V., V., V., & Ramalingam, V. V. (2017). Computer Vision Based Classification on Commercial Videos. In M. S., & V. V. (Eds.), Multi-Core Computer Vision and Image Processing for Intelligent Applications (pp. 105-135). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-0889-2.ch004

Valverde, R., Torres, B., & Motaghi, H. (2018). A Quantum NeuroIS Data Analytics Architecture for the Usability Evaluation of Learning Management Systems. In S. Bhattacharyya (Ed.), *Quantum-Inspired Intelligent Systems for Multimedia Data Analysis* (pp. 277–299). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-5219-2.ch009

Vassilis, E. (2018). Learning and Teaching Methodology: "1:1 Educational Computing. In K. Koutsopoulos, K. Doukas, & Y. Kotsanis (Eds.), *Handbook of Research on Educational Design and Cloud Computing in Modern Classroom Settings* (pp. 122–155). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-3053-4.ch007

Wadhwani, A. K., Wadhwani, S., & Singh, T. (2016). Computer Aided Diagnosis System for Breast Cancer Detection. In Y. Morsi, A. Shukla, & C. Rathore (Eds.), *Optimizing Assistive Technologies for Aging Populations* (pp. 378–395). Hershey, PA: IGI Global. doi:10.4018/978-1-4666-9530-6.ch015

Wang, L., Wu, Y., & Hu, C. (2016). English Teachers' Practice and Perspectives on Using Educational Computer Games in EIL Context. *International Journal of Technology and Human Interaction*, *12*(3), 33–46. doi:10.4018/IJTHI.2016070103

Watfa, M. K., Majeed, H., & Salahuddin, T. (2016). Computer Based E-Healthcare Clinical Systems: A Comprehensive Survey. *International Journal of Privacy and Health Information Management*, *4*(1), 50–69. doi:10.4018/IJPHIM.2016010104

Weeger, A., & Haase, U. (2016). Taking up Three Challenges to Business-IT Alignment Research by the Use of Activity Theory. *International Journal of IT/Business Alignment and Governance, 7*(2), 1-21. doi:10.4018/IJITBAG.2016070101

Wexler, B. E. (2017). Computer-Presented and Physical Brain-Training Exercises for School Children: Improving Executive Functions and Learning. In B. Dubbels (Ed.), *Transforming Gaming and Computer Simulation Technologies across Industries* (pp. 206–224). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-1817-4.ch012

Williams, D. M., Gani, M. O., Addo, I. D., Majumder, A. J., Tamma, C. P., Wang, M., ... Chu, C. (2016). Challenges in Developing Applications for Aging Populations. In Y. Morsi, A. Shukla, & C. Rathore (Eds.), *Optimizing Assistive Technologies for Aging Populations* (pp. 1–21). Hershey, PA: IGI Global. doi:10.4018/978-1-4666-9530-6.ch001

Wimble, M., Singh, H., & Phillips, B. (2018). Understanding Cross-Level Interactions of Firm-Level Information Technology and Industry Environment: A Multilevel Model of Business Value. *Information Resources Management Journal*, *31*(1), 1–20. doi:10.4018/IRMJ.2018010101

Wimmer, H., Powell, L., Kilgus, L., & Force, C. (2017). Improving Course Assessment via Web-based Homework. *International Journal of Online Pedagogy and Course Design*, *7*(2), 1–19. doi:10.4018/IJOPCD.2017040101

Wong, Y. L., & Siu, K. W. (2018). Assessing Computer-Aided Design Skills. In M. Khosrow-Pour, D.B.A. (Ed.), Encyclopedia of Information Science and Technology, Fourth Edition (pp. 7382-7391). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2255-3.ch642

Wongsurawat, W., & Shrestha, V. (2018). Information Technology, Globalization, and Local Conditions: Implications for Entrepreneurs in Southeast Asia. In P. Ordóñez de Pablos (Ed.), *Management Strategies and Technology Fluidity in the Asian Business Sector* (pp. 163–176). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-4056-4.ch010

Yang, Y., Zhu, X., Jin, C., & Li, J. J. (2018). Reforming Classroom Education Through a QQ Group: A Pilot Experiment at a Primary School in Shanghai. In H. Spires (Ed.), *Digital Transformation and Innovation in Chinese Education* (pp. 211–231). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2924-8.ch012

Yilmaz, R., Sezgin, A., Kurnaz, S., & Arslan, Y. Z. (2018). Object-Oriented Programming in Computer Science. In M. Khosrow-Pour, D.B.A. (Ed.), Encyclopedia of Information Science and Technology, Fourth Edition (pp. 7470-7480). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2255-3.ch650

Yu, L. (2018). From Teaching Software Engineering Locally and Globally to Devising an Internationalized Computer Science Curriculum. In S. Dikli, B. Etheridge, & R. Rawls (Eds.), *Curriculum Internationalization and the Future of Education* (pp. 293–320). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2791-6.ch016

Yuhua, F. (2018). Computer Information Library Clusters. In M. Khosrow-Pour, D.B.A. (Ed.), Encyclopedia of Information Science and Technology, Fourth Edition (pp. 4399-4403). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-2255-3.ch382

Zare, M. A., Taghavi Fard, M. T., & Hanafizadeh, P. (2016). The Assessment of Outsourcing IT Services using DEA Technique: A Study of Application Outsourcing in Research Centers. *International Journal of Operations Research and Information Systems*, *7*(1), 45–57. doi:10.4018/IJORIS.2016010104

Zhao, J., Wang, Q., Guo, J., Gao, L., & Yang, F. (2016). An Overview on Passive Image Forensics Technology for Automatic Computer Forgery. *International Journal of Digital Crime and Forensics*, *8*(4), 14–25. doi:10.4018/IJDCF.2016100102

Zimeras, S. (2016). Computer Virus Models and Analysis in M-Health IT Systems: Computer Virus Models. In A. Moumtzoglou (Ed.), *M-Health Innovations for Patient-Centered Care* (pp. 284–297). Hershey, PA: IGI Global. doi:10.4018/978-1-4666-9861-1.ch014

Zlatanovska, K. (2016). Hacking and Hacktivism as an Information Communication System Threat. In M. Hadji-Janev & M. Bogdanoski (Eds.), *Handbook of Research on Civil Society and National Security in the Era of Cyber Warfare* (pp. 68–101). Hershey, PA: IGI Global. doi:10.4018/978-1-4666-8793-6.ch004

About the Contributors

D. Jeya Mala has a Ph.D. in Software Engineering with Specialization on Software Testing and is Associate Professor in Thiagarajar College of Engineering, –a leading educational and philanthropic institution in Tamil Nadu, India. She had been in the industry for about 4 years. She has a profound teaching and research experience of more than 15 years. She has recieved "Best Techno Faculty Award" from ICT Academy of Tamil Nadu, a Govt. Initiative. Also, she is the proud recipient of "Best Poster Award - 2016" from Indian Science Congress Association, an Initiative of DST, Govt. of India. She has also received 'Certificate of Excellence' from IIT-Bombay in successfully completing the FDP101x and FDP201x with score above 95%. She has published a book on "Object Oriented Analysis and Design using UML" for Tata McGraw Hill Publishers. She has published more than fifty (50) papers about her research works at leading international journals and conferences such as IET, ACM, Springer, World Scientific, Computing and Informatics etc. As a researcher, Dr. Jeya Mala had investigated practical aspects of software engineering and object oriented paradigms for effective software development. Her work on Software Testing has fetched grants from UGC under Major Research Project scheme. Her dissertation has been listed as one of the best Ph.D. thesis in the CSIR – Indian Science Abstracts. She has successfully guided numerous Software Development based projects for the IBM- The Great Mind Challenge (TGMC) contest. The project she has mentored during 2007, has received national level Best Top 10 Project Award – 2007, from IBM. Currently she is guiding Ph.D. and M.Phil research scholars under the areas of Software Engineering and optimization techniques. She is a life member of Computer Society of India and an invited member of ACEEE. She forms the reviewer board in Journals like IEEE Transactions on Software Engineering, Elsevier – Information Sciences, Springer, World Scientific, International Journal of Metaheuristics etc. She has organized several sponsored national level conferences and workshops, notably she is one of the organizers of "Research Ideas in Software Engineering and Security (RISES'13) – A run-up event of ICSE 2014 sponsored by Computer Society of India". She has been listed in Marquis Who's Who list in 2011 onwards. She has chaired several national and international conferences notably she is the organizing chair of International Congress on Software Engineering held at Thailand during 2014. She is a proud recipient of several laurels from industries like Honeywell, IBM and Microsoft for her remarkable contributions in the field of Software Development and Object Orientation.

* * *

Sejal A. Bhavsar is an Assistant Professor at Gandhinagar Institute of Technology, India. She is pursuing her PhD from Ganpat University and received her Mtech in CSE from the Nirma University in 2012. Her research areas of interest are Internet of Things, Cloud and Fog computing.

Jasmine K. S. is currently working as Associate Professor in the Department of MCA, R. V. College of Engineering, Bangalore got 23 years of teaching experience. She has authored 70+ research papers in the national and international level conferences and journals. She is also organizing committee member /Advisory board member/ Reviewer for various journals and conferences. She also chaired sessions for conferences. She has authored book and Book chapters published by international publishers namely, Lap LAMBERT Academic Publishing, Germany, In-Tech publishers, Vienna, Macmillan Publishers and IGI International. She served as the Coordinator for ISO, NBA & NAAC. Her research interests include Software reuse, Software testing and Experimental software engineering. Dr. Jasmine is the member of many professional societies such ISTE, CSI, ISCA, IAENG, IDES and IFERP Dr.Jasmine has been honored with several awards both nationally and Internationally, namely, Global award titled 'Best Academic Researcher -2012' by Association of Scientists, Developers and Faculties(ASDF), an international professional body at Puducherry, Tamilnadu, "Excellence in Technology Research Award" by EET CRS Research & Branding Company, Noida, Delhi, under 'Technology achievement awards-2014','Outstanding Faculty Award-2015' & Distinguished Woman in Science-2018 by Venus International foundation (VIFFA), Centre For Advanced Research and Design, Chennai, 'Senior Women Educator & Scholar Award-2017' by National Foundation For Entrepreneurship Development (NFED), Coimbatore, Tamil Nadu, 'Prof. Indira Parikh 50 Women in Education Leaders-2018' by World Education Congress, Mumbai. She also has been honored by Rashtreeya Sikshana Samithi Trust, Bangalore for 'Excellence in Education' on many occasions.

Kirit J. Modi is an Associate Professor at Ganpat University, India. He received his PhD in CSE from the Nirma University in 2016. His current research interests include service-oriented computing and cloud computing.

Chitra P. is currently working as Professor in the Department of Computer Science & Engineering, Thiagarajar College of Engineering. She completed her B.E from Madurai Kamaraj University during 1995; subsequently she worked as lecturer and completed her M.E and Ph.D in CSE during 2004 and 2011, respectively. She is a reviewer for many national and international peer reviewed journals and member of technical program committee for many IEEE national and international conferences. She has under her credits many publications in reputed international conferences and journals in the areas of distributed systems, cloud computing, Multicore architectures.

Brinda Y. Pandit is an Assistant Professor at Gandhinagar Institute of Technology, India. She is pursuing her PhD from Ganpat University and received her Mtech in Information techonology from the Ganpat University in 2012. Her research areas of interest are Web usage mining and Social Internet of Things.

P. B. Pankajavalli is an Assistant Professor in the Department of Computer Science, School of Computer Science and Engineering, Bharathiar University, Coimbatore, Tamil Nadu, India. She obtained her Post Graduation MCA under Bharathiar University in 2003 and M.E., degree under Anna University in 2011and Ph.D., degree in Computer Science from Mother Teresa Women's University in the year 2013. She has published 30 papers in journals and in Conferences both at National and International level. She has received Best Teacher Award from Lions Club, Erode. Her areas of interest include Ad-hoc Networks, Wireless Sensor Networks and Internet of Things.

K. Sridhar Patnaik is working as Associate Professor in the department of Computer Science and Engineering, BIT, Mesra, Ranchi. He has done BE(EE) from NIT, Silchar, Assam (India) and ME(SE), BIT, Mesra, Ranchi, Jharkhand (India). He obtained his PhD in Engineering from BIT, Mesra, Ranchi, Jharkhand (India). He has a work and research experience of 15 years. His areas of interest include Intelligent System Design, Software Engg, Soft Computing, Image Analysis, IoT and Machine Learning. He has a number of publications in national and international journals. He is member of professional bodies like ISTE, IE(I), ACM, CI Lab-Univ. of Manitoba, Winnipeg, Canada.

P. Priakanth is a Professor in the Department of Computer Technology-UG, Kongu Engineering College, Perundurai, Tamil Nadu, India. He received MCA degree in Computer Applications from Bharathiyar University, Coimbatore in 1998. He received ME degree in Computer Science and Engineering from Anna University, Chennai in 2005. He received Ph.D degree in Information and Communication Engineering from Anna University, Chennai in 2010. His research area includes Ad-hoc Networks, Wireless Communications, Wireless Sensor Networks and its applications.

Abirami S. is currently a full time research scholar in the area Deep Learning. She completed her B.E from National Engineering College, Kovilpatti in 2010 and her M.E from Anna University, Coimbatore in 2012.

Anitha Elavarasi S. is an Assistant professor in the Department of Computer Science and Engineering, Sona College of Technology, Salem, Tamilnadu. She is having eleven years of teaching experience. She has published more than 20 papers in the National and International journals and conferences. She is currently working in the Healthcare domain and she is one of the technical team member for developing consultancy projects in the Web and Mobile App domain. Her area of interest includes web technology, software engineering, Database Technologies, Data mining, big data analytics, IoT etc. She had completed her PhD in the design and development of clustering algorithm for categorical data. Her passion is reading books, learns new technologies and music.

S. Gopikrishnan is an Assistant Professor in the department of Information Technology at Karpagam College of Engineering, Coimbatore, India. He received his B.E. (2009) and M.E. (2012) degrees in Computer Science and Engineering from Anna University, Chennai. He holds Ph.D. (2018) degree at Computer Science and Engineering from Anna University, Chennai. His professional career is comprised of a total 8 years of work experience in academia and industry. His research interests span wireless networks, cyber physical systems, security and privacy, social networks, and algorithms and have published high quality papers on these fields. He is a guest editor of various International journals and distinguished member in technical societies.

Karthick Selvaraj is a PhD Research Scholar in the Department of Computer Science, School of Computer Science and Engineering, Bharathiar University, Coimbatore, Tamil Nadu. He has received Post Graduation M.Sc. Computer Science in 2017 from Bharathiar University (UD), Coimbatore, Tamil Nadu. He has published research papers in International Journals and presented papers in International and National Conferences. His area of specializations and research interests are Wireless Sensor Networks, Internet of Things and Analysis of Algorithms.

Itu Snigdh received her PhD in Wireless sensor networks in 2015 and Master's degree in Software Engg. in 2000 from B.I.T Mesra (Ranchi). She completed her bachelor's degree in electrical engineering from BIT Sindri in 1996. She is currently working as an Asst. Professor in the Dept. of Computer Science and Engineering at BIT Mesra for 15 years. Her areas of interest include software Engineering, Internet of Things, Wireless Sensor Networks, and Database Management Systems.

Index