UNIVERSITAT POLITÈCNICA DE CATALUNYA (UPC)

Departament d'Arquitectura de Computadors (DAC)
EPSEVG Campus

Adaptive Learning-based Resource Management Strategy in Fog-to-Cloud

This dissertation is submitted for the degree of

Doctor of Philosophy* in *Computer Architecture

Doctoral thesis by:

Souvik Sengupta

Thesis Advisor:
Dr. Jordi García Almiñana

Thesis Co-Advisor:
Dr. Xavi Masip-Bruin

June, 2020

“The highest education is that which does not merely give us information but makes our life in harmony with all existence." - **Rabindranath Tagore**

Universitat Politècnica de Catalunya (UPC BarcelonaTech)
Department of Computer Architecture

Adaptive Learning-based Resource Management Strategy in Fog-to-Cloud

by *Souvik Sengupta*

Abstract

Keywords— *Fog-to-Cloud, Internet of Things, Smart computing, Machine Learning, Forecasting, Prediction, Distributed Database, Resource management, Resource Allocation*

Technology in the twenty-first century is rapidly growing up. All the things and devices surrounding us are getting connected with the network. Eventually, all of these things and devices are becoming more intelligent than previous. Importantly, all of these precise and continuous developments are driving us into a new smart computing world. Significantly, to build a more intelligent, smarter and agile computing platform, lots of new computing architectures are emerging. Fog-to-Cloud (F2C) is among one of them, which emerges to ensure the commitment for bringing the higher computing facilities near to the edge of the network and also help the large-scale computing system to be more intelligent. As the F2C is in its infantile state, so there are lots of issues that remain to be addressed. The biggest challenge for this computing paradigm is to build a proper and efficient resource management mechanism. Mainly, to address this challenge, in this work, we have given our sole interest for designing the initial architectural framework to build a proper, adaptive and efficient resource management mechanism in F2C.

F2C has been proposed as a combined, coordinated and hierarchical computing platform. A vast number of heterogeneous devices are participating in any F2C-enabled system. Efficient utilization of those devices is always a key concern. However, their versatility creates a massive challenge for effectively handling them. Even following any large-scale smart computing system, it can easily recognize that the system can offer various kind of services among its consumers. Notably, all of the services are served for different purposes. Significantly, for efficiently provide the services, it is essential to perform some tasks. Typically, each task has different requirements, which have to be satisfied. Therefore, it is necessary to know the characteristics of participating devices and system offered services before building the effective and resource management mechanism in F2C-enabled system. However, considering these facts, initially, we have given our intense focus for identifying and defining the taxonomic model for all the participating devices and system involved services-tasks.

In any F2C-enabled system consists of a large number of small Internet-of-Things (IoTs). They are involved in generating a continuous and colossal amount of sensing-data by capturing various environmental events. Notably, this sensing-data is one of the key ingredients for various smart services (e.g., traffic monitoring, fire-emergency service, air pollution monitoring, etc.) which have been offered by the F2C-enabled system. Besides that, resource statistical information is also playing a crucial role, for efficiently providing the services among the system consumers. Typically, continuous monitoring of participating devices can generate a massive amount of resource statistical information in the F2C-enabled system. Eventually, with the help of this information, it becomes much easier to know the device's availability and suitability for executing some tasks to offer some services. Therefore, ensuring better service facilities for any latency-sensitive services (e.g., e-health monitoring, traffic monitoring, etc.), it is essential to securely distribute the sensing-data and resource statistical information over the network. Hence, considering these matters, in this work, we also proposed and designed a secure and distributed database framework for effectively and securely distribute the data over the network.

Interestingly, to build an advanced and smarter system is necessarily required an effective mechanism for the utilization of system resources. Notably, effective utilization of system resources not only ensures to reduce the unnecessary resource consumption, but it also improves the overall system performance. Typically, the utilization and resource handling process mainly depends on the resource selection and allocation mechanism. Importantly, on-demand resource selection and allocation is a pretty tough job for any large-scale distributed system. Significantly, the prediction of resources (e.g., RAM, CPU, Disk, etc.) usage and performance (i.e., in terms of task execution time) can help to identify their suitability for executing some tasks. Therefore, adopting the machine learning (ML) techniques can be much more useful for designing an advanced and sophisticated resource allocation mechanism in the F2C-enabled system. However, adopting and performing the ML techniques in F2C-enabled system can be a challenging task. Especially, the overall diversification and many other issues can pose a massive challenge for successfully performing the ML techniques in any F2C-enabled system. By identifying those issues and following the architecture of F2C-enabled smart system, in this work, finally, we have proposed and designed two different possible architectural schemas for performing the ML techniques in the F2C-enabled system. So that, we can achieve an adaptive, advance and sophisticated resource management mechanism in the F2C-enabled system. However, our proposals are the initial footmarks for designing the overall architectural framework for resource management mechanism in F2C-enabled system. Hence, in our proposals, we have opened various issues; in future which can be helpful for the willing researchers to continue the investigation over the various open challenges.

Dedication

In memories of ***Shri Bibekanada Sengupta*** and ***Shri Mohit Kumar Khan***......

Acknowledgement

Long in the making with many more rewrites and edits to follow, my work is indebted to the support and generosity of a range of interlocutors: family, advisors, colleagues, research staff, and friends. I will try to do justice in these acknowledgements and in my footnotes to the many people who have guided me along the way.

My Special thanks to my parents, companion and all other family members; who stand beside me in every second of this journey. Without the endless support and love from my parents *Chandana* and *Bhaskar Sengupta*, I would not have been able to accomplish this work. Without the unconditional friendship, love, and encouragement from my companion *Ankana*, it was not possible for me to finish this work. Thank you, *Ankana* for calmly and patiently sharing all of my frustrations and sorrows. Thanks to *Mani*, *Amma*, and *Didun*; whose blessings have turned this challenging path into much easier way for me. The countless affections of *Mosai*, *Masimoni*, *Fulmosai*, *Fulmasi*, *Kutti meso*, and *Kutti*; gave me the strength and confidence in this challenging journey. An exceptional thanks to my maternal-uncle *Dr. Asis Khan*, and maternal-aunt *Suparna Bardhan* for all of your wise advice, and unconditional help to improve my writing skills. Also, I would like to extend my thanks to all of my cousins: *Esha*, *Rimo*, *Tuya*, *Riku*, and *Vicky* for being my biggest cheerleaders. In addition, I am also thankful to my all other family members.

I want to express my sincere gratitude to my PhD advisors, *Dr. Jordi García Almiñana* and *Dr. Xavier Masip-Bruin*, for their valuable tutorship, support and guidance, which helps me to grow as a research scientist. Their dedication, invaluable discussions and timely advice certainly made this work possible. Thank you, *Professors* for supporting me during the past three years and giving me the freedom to chase my dream. I extend my heartiest thanks to my committee members for serving as my committee members, at this complicated situation. Also, many thanks to my colleagues at CRAAX Lab, with whom I had the pleasure of working over more than three years. Moreover, I would also like to thank all the administrative staffs at UPC for their support during the course of my dissertation.

Many thanks to *Sarang*, with whom I have worked in collaboration since my second year in CRAAX, contributing to the publication of several ideas. In addition, I thank all of my friends those who supported me to achieve my goal. Thank you *Alex*, *Amir*, *Pargat Ji*, *Bawa Paji*, *Kiko*, *Carlos*, and *Sergi*; you guys have never let me think that I am living too far away from my family. Also, my deepest gratitude to *Dr. Suman Shankar Bhunia*; he was the first one who unconditionally guided me to start my career in the research domain.

This work has been supported by several organizations. My sincere acknowledgement and thanks to all of them. Especially, the H2020 European Union mF2C project with reference 730929, the Spanish Ministry of Economy and Competitiveness and the European Regional Development Fund, under contract TEC2015-66220-R(MINECO/FEDER), the Spanish Ministry of Science, Innovation and Universities and the European Regional Development Fund (FEDER) under contract RTI2018-094532-B-I00, and Technical University of Catalonia (UPC).

Souvik Sengupta, June 2020
Barcelona, Spain

Contents

List of Figures

List of Tables

Abbreviations

AC Agent Controller.

CAU Control Area Unit.

CC Cloud computing.

ClA Cloud Agent.

DDB distributed database.

EC Edge computing.

F2C Fog-to-Cloud.

FA Fog Area.

FC Fog computing.

FEL Fog Employee Layer.

FML Fog Manager Layer.

IaaS Infrastructure as a Service.

ICTs Information and communications technologies.

IoT Internet of Things.

LFn Leader Fog node.

ML Machine Learning.

NFn Normal Fog node.

PaaS Platform as a Service.

PM Platform Manager.

QoS Quality of Service.

RM Resource Management.

SaaS Software as a Service.

SC Smart Computing.

SDN Software-defined networking.

SFDDM Secure F2C Distributed Database Management system.

SLA service-level agreement.

SPA Student Project Allocation.

TOPSIS Technique for Order of Preference by Similarity to Ideal Solution.

VMs Virtual Machines.

WSN Wireless Sensor Network.

Chapter 1: Overview

The introductory chapter gives an insight into the topic of the research area. The chapter begins with the preamble to the research topic with further strengthen on the aims of the research project. Eventually, it describes the overall structure of the dissertation of the research.

1.1 Introduction

In modern days Information and communications technologies (ICTs) are facing fast and rapid development. Importantly, modern networking technologies are becoming more ubiquitous and efficient. As a result, it helps to rise-up the new and more sophisticated computing paradigms for ensuring to deliver high-performance computational services within a quicker time. Realizing this fact, in 2016 a famous computer scientist Hanna Wallach said in an article [1] that, *"We are on the cusp of a new era in computational social science"*. In that paper, he has clearly articulated how the ongoing development and integration of computation and communication technologies driving our society towards a new intelligent world.

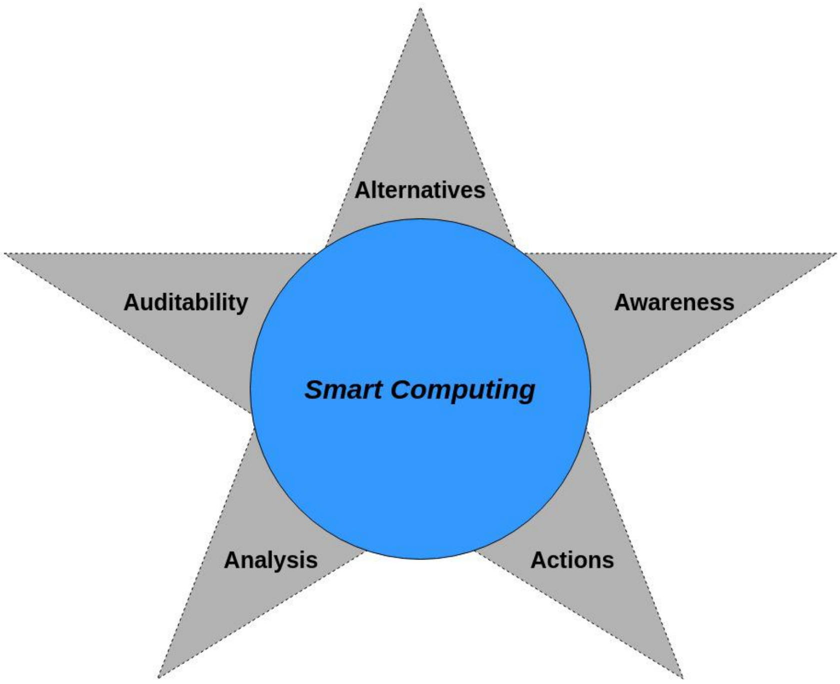


Figure 1.1: Five key stages of intelligent activity: The five A's of Smart Computing (SC)

Significantly, all these certain developments changing the view of traditional computational mechanism rapidly. Since the internet technology has developed and it integrated with computation technology; the traditional computation becomes more ubiquitous and gaining more power. Also, with the help of the advancement of communication and networking technologies, the concept of distributed computing is getting more widespread than previously. Therefore, the conventional computation mechanism is evolving and emerging a new computing paradigm, known as *Smart Computing (SC)*. By understanding this bloomingness of smart computing concept, in 2009, A.H. Bartels and his fellow researchers tried to consolidate the smart computing concept by making the generalized definition. According to them, smart computing adds the capabilities of real-time situational awareness and automated analysis to the existing technologies. They said that smart computation is a unified computation mechanism, where *"A new generation of integrated hardware, software, and network technologies that provide IT systems with real-time awareness of the real world and advanced analytics to help people make more intelligent decisions about alterna-*

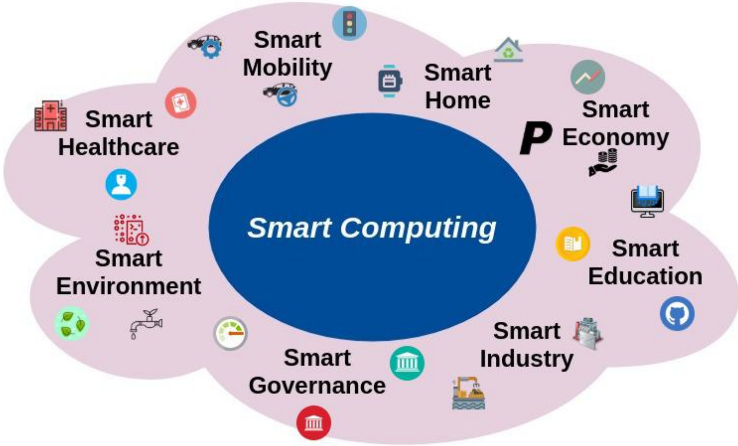


Figure 1.2: Different application domains of Smart Computing (SC) paradigm

tives and actions that optimize business processes and business balance sheet results" [2]. However, in that article, they clearly mentioned that smart computing concept is not a new computing methodology; it is an evolutionary extension of past generations of computing. Basically, by combing the five critical stages of intellectual activity (as shown in Fig. 1.1), the existing past technologies are getting smarter and intelligent and becoming the *SC* paradigm. Recent days, we are already surrounded by several smart notions, such as smart cities, smart industry, smart farms, all in all, building the main concept of SC paradigm, as illustrated in Fig. 1.2. Some of them (e.g., Smart City) is getting popular and buzzing up all over the world. Most importantly, lots of research and developing activities are going on to develop these concepts and make them smarter than previous. The main motivation behind the ongoing development is to ensure a better quality of lives [3].

The rise of Internet of Things (IoT) technology has significantly fueled the development of ubiquitous computing and turned the traditional computing platform into intelligent computing paradigm. It is helping the existing system to be smarter, self-adaptive and knowledgeable. In meanwhile, connecting the surrounding devices with the internet and deploying several sensors and actuators in the environment, started the massive amount of data flow over the network, and it poses a big challenge for the existing system for managing those massive amounts of data. On the other hand, one of the primary demands for any smart computing platform is to ensure better service facilities in quick response time. For that reason, it is essential to process those data, near to the end-users [4]. Also the ability of context-awareness [5], [6] is one of the key aspects for any intelligent system. Therefore, the integration of various existing computing concept (i.e., Distributed computing, Centralized computing) is a natural demand for any SC paradigm [7], [8]. Thus, defining an efficient SC platform essentially means appropriately integrating various kind of existing technologies, where some of such technologies (e.g., Cloud computing (CC)) provide the best facilities to store and process a massive amount of continuously capturing data. Some other (e.g., IoT and Wireless Sensor Network (WSN)) provide the best facilities for producing and capturing those data from various kind of sources of the real world. Furthermore, new technologies (e.g., Fog computing (FC) and Edge computing (EC)) provide the most appropriate facilities to process those data near to the end-users or edge devices, speeding up performance as well as reducing network traffic through the system. Considering and combining all these facilities provided by the various existing technologies helps in to create better and more efficient and intelligent system [9].

For bringing more intelligence into the existing smart systems, a new integrated computing paradigm has been conceptualized and proposed by Masip-Bruin et al. [10]. By combining all the concepts of cloud, fog and edge computing, and IoT, the authors proposed a new combined and hierarchical computing platform which is known as *Fog-to-Cloud (F2C)*. According to the authors, F2C is an emerging and developing concept, which is rising to help any smart system to be more intelligent. As the F2C is in its infantile state, therefore still lots of development works are essential for solving many remaining challenges. The main motivation of the F2C paradigm is to provide exceptional service facilities and bring more intelligence to any smart system. That means, the F2C must ensure better system resources utilization for providing the services, by enhancing the Quality of Service (QoS). Thus, it is an essential demand to have a proper and effective resource management mechanism in the F2C paradigm. Since the F2C is in the developing stage, so we realized that some extra efforts are necessary to build an effective and proper resource management mechanism in the F2C paradigm, and that is the core foundation of this research work. In this thesis, by addressing various existing issues, we propose an adaptive learning-based resource management mechanism in F2C computing paradigm for effectively managing any smart system resources and helps the existing smart system to be more intelligent.

1.2 Problem statement

The F2C is emerging to ensure exceptional service facilities and bringing more smartness to any SC paradigm. For keeping these promises, a proper, efficient and effective resource utilization mechanism has to be designed and developed in F2C. Since the F2C still is in the early stage, the most significant problem is to utilize the system resources effectively and efficiently manage them. Importantly, the effective and efficient utilization of system resources can be attained in many dimensions (i.e., reduction of unnecessary resource consumption, improvement of QoS factors, enhancing the security and privacy functionalities, cost reduction, performance improvement, etc.) [5], [7], [8], [11]–[17]. Typically, in the F2C-enabled system, defining the efficient resource management mechanism is a complicated job, which has many open issues. For instances, the growth of the smart city is massive, and interestingly a vast amount of various connected devices are working there. Since the popularity and growth of the smart city are increasing, significantly, the number of connected devices is also increasing. A critical analysis made by the famous technological company *Ericsson* that, they are predicting by 2022 there would be more than twenty-nine (29) billion active connected devices (i.e., depicted in Fig. 1.3) working in the surroundings, of which around 18 billion would be related to various IoT devices [18]. Thus, it can be easily realizable that the diversification of the massive amount of connected devices is going to pose a major challenge for designing a proper strategy to utilize and manage them effectively. Also, these IoTs (i.e., sensory devices) are going to play a crucial role in the modern smart city. They are mainly responsible for continuous capturing various environmental events and generating data. IoTs are one of the primary sources for producing a massive amount of data flow through over the network in the system.

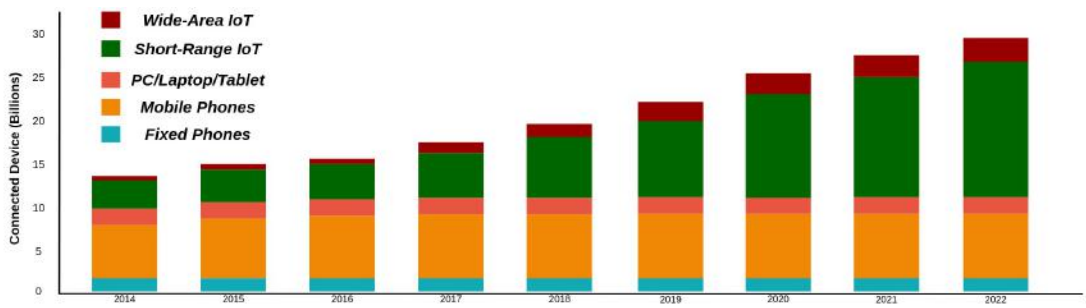


Figure 1.3: Forecasting the number of active connected devices by 2022 - provided by Ericsson INC.

Most importantly, data is one of the necessary ingredients for any smart services (e.g., smart healthcare services). So it is quite evident that without properly managing those data, it is impossible for building a useful resource management mechanism in any F2C-enabled system. However, mishandling the data can create a negative impact on system performance. Unfortunately, since the F2C is in the infantile stage, thus it does not have the proper and secure data storing mechanism, which also creates another challenge for designing effective and intelligent resource management mechanism in F2C paradigm.

Significantly, in any smart system, selecting and allocating the proper resource(s) is not only enhance the service quality but also it ensures effective utilization of resources in any dimension; which have described in earlier. Along with the diversity of F2C system resources, the hierarchical, distributed and combined architecture of F2C system make a vital challenge for design an effective, sophisticated and intelligent resource management mechanism in F2C computing paradigm. Thus, considering all of these facts, we realized that before solving the main challenge in F2C computing paradigm, it is necessary to solve other existing related issues. For that purpose, we pointed out and described all the existing sub-problems. Therefore in this research work, initially, we investigated the following sub-problems. Then, by addressing the existing sub-problems, we propose a new adequate resource management mechanism in F2C computing paradigm.

- ***SP1.*** **Diversity among the participating system resources and variety in the system services; creates a massive challenge for any smart computing system:** Following any large-scale smart computing system (e.g., smart city), it can be easily determined that a massive amount of various devices (i.e., from small mobile devices to powerful computing devices) are participating in the system. Each of them has different specifications, and they are different in nature. Also, any modern smart computing system offering various kind of services (e.g., smart traffic management, smart waste management, etc.) to its consumers. Each service has different service-demands, and to successfully provide the service(s), all of these demands essentially needs to be satisfied. Therefore, without having the proper knowledge about the characteristics of participating resources and service-demands, it becomes a difficult job to properly manage the participating system resources and turn the existing system into more intelligent.

- ***SP2.*** **Ensuring the latency-sensitive context-awareness nature throughout the computing paradigm:** One of the key characteristics of any smart system is to offer the latency-sensitive context-aware services effectively. Typically, in a large-scale smart system, it is a challenging job. The reason behind that, primarily the service needs to be executed near to the end-users. Secondly, for the execution of service required data needs to be stored near to the end-users. For a large-scale smart system, where various kind of devices are participating, it is a complicated and challenging job to address these two challenges adequately and help to make the system more intelligent.

- ***SP3.*** **Efficiently and securely managing the data over the network:** An enormous amount of participating computing and sensory devices is continuously generating data in any large-scale smart computing platform. These generated data are the basic ingredients of various services. To promptly provide any context-aware services, it is necessary to manage and distribute the data over the network securely and also keep it near to the end-users. Eventually, effective, secure and proper distribution of data over the network is also essential for any large-scale smart computing paradigm. However, the hierarchical, distributed and combining characteristics of the F2C computing platform is making a massive challenge for securely distributing the data over the system.

- ***SP4.*** **Efficient resource utilization improves the quality of overall system:** For avoiding the irrelevant system resources consumption, it is necessary to utilize the system resources efficiently and intelligently. For a big-scale smart computing platform, where billions and millions of various computing devices are working; it is a pretty tough job for ensuring to avoid unnecessary resource consumption by choosing the appropriate resource(s) for executing some task(s). Hence, defining and designing an effective and adaptive resource utilization mechanism is one of the elementary demands for this computing paradigm.

Undoubtedly, we believe that by addressing these four main sub-problems and few other existing issues, we can reach to our goal and able to define more precise, effective and advanced resource management mechanism for F2C paradigm; explicitly which also helps the existing smart system to be more intelligent.

1.3 Objectives

The famous American astronaut Neil Armstrong correctly defined the meaning of research. He said that "*In much of society, research means to investigate something you do not know or understand*." That means, according to him, the meaning of 'research' is a process of works for chasing the unknown or invisible things for knowing it. Significantly, one of the famous author and philanthropist Tony Robbins said, "*Setting goals is the first step in turning the invisible into the visible*." Similarly, like every other research works, we have some specific goals. By achieving these goals, we want to fulfil some objectives. Before discussing them, we identify that our objectives can be classified into two main categories: *Generalized* objective and *Technical* objective. In the below, we are going to discuss them briefly.

- **Generalized objective** - By accomplishing this work, our generalized and primary objective is to design the architectural framework for adequately and intelligently managing the F2C-enabled smart system resources. For that purpose, it is essential to design an adaptive, efficient and sophisticated resource allocation mechanism in the F2C paradigm, which can ensure to intelligently choose the best-fitted resource(s) for executing task(s) and provide the service(s) enhancing the quality matters.
- **Technical objectives** - To present the architectural framework for a sophisticated resource management strategy, we have a couple of 'Technical' or 'Problem specific' objectives, which we want to achieve by accomplishing this work. These objectives are as follows:
  - ***TO1:*** Identify the taxonomy of F2C system resources and based on that design, the generalized resource description model, to have a complete view of resources classification.
  - ***TO2:*** Formalize the aggregated resource description model in the different hierarchy of the F2C computing paradigm, for knowing the overall capacity.
  - ***TO3:*** Define and propose the 'Service-Task' classification in the F2C paradigm that describes the characteristics of services and requirements of tasks.
  - ***TO4:*** Design the resource status tracking mechanism for the coordinated F2C platform in order to have continuous fresh information about the available system resources. Also, manage the collected resource information, service-related information, and sensing-data distributively and securely over the F2C platform.
  - ***TO5:*** Primarily, based on the service-task requirements, match the suitability for selecting the best-fitted resource(s) and allocate them to execute some task(s).
  - ***TO6:*** Continuous monitoring of the service-task execution to get the actual resource(s) usage and task-execution time information; therefore, providing this information as the feedback for improving the matching preferences between service-task and resource.
  - ***TO7:*** Based on re-evaluated matching preferences, allocate the appropriate resource(s) to execute some task(s) and offered the service(s) in the F2C paradigm, and that helps to identify the optimal solution for efficiently allocating the resource(s) in the F2C paradigm.

1.4 Contribution of this dissertation

To adequately address the existing challenges in the F2C paradigm and fulfil our *Generalized* and *Technical* objectives, in this dissertation work, we have proposed some solutions and performed some jobs for justifying our proposals. In the following, we are going to briefly explain those proposals and jobs, supported by the list of accepted publications.

- ***Job1:*** Significantly for obtaining our *generalized* objective, we design an architectural framework for effectively managing the F2C resources. Notably, with the help of our architecture, we can able to utilize the system resources adaptively. For defining the whole architectural framework and fulfill our objectives, we also design various architectural modules and sub-modules, which are collaboratively addressing many issues to design and develop an architectural framework of adaptive, sophisticated and efficient resource management mechanism in F2C paradigm.

  Publications -

  - "An Architecture for Resource Management in a Fog-to-Cloud Framework", accepted in *24th International European Conference on Parallel and Distributed Computing (Euro-Par 2018), F2C-DP Workshop*. DOI: 10.1007/978-3-030-10549-5_22

- ***Job2:*** For achieving the *TO1, TO2,* and TO3, initially we investigate a significant number of related works. Then, based on the identified characteristics, we propose the taxonomic models for F2C system resources, F2C system offer services and their corresponding tasks. After that, based on the resource taxonomic model, we also propose and develop the resource description models for properly organizing the system resources information. Notably, the proper organization of this information is helping us to realize the overall computing capacity of any F2C-enabled system.

  Publications -

  - "Taxonomy and Resource Modeling in Combined Fog-to-Cloud Systems", accepted in *Future Technologies Conference (FTC) 2018.* DOI: 10.1007/978-3-030-02686-8_52
  - "Essentiality of Resource and Service-Task Characterization in the Coordinated Fog-to-Cloud Paradigm", accepted in *2018 IEEE 7th International Conference on Smart Communications in Network Technologies (SaCoNeT-2018).* DOI: 10.1109/SaCoNeT.2018.8585732

- ***Job3:*** Similarly, to attain the *TO4*, we designed and developed an architectural module for efficiently monitoring the system resources. In the F2C-enabled system, this module is continuously generating a massive amount of data and creating the information pool. Along with its data, the information pool is playing a crucial role in the service-task execution process. Therefore, the secure distribution of the data along with the information pool is essential. Hence, to satisfy this demand, we propose and develop a secure distributed database framework in the F2C-enabled system.

  Publications -

  - "Essentiality of Managing the Resource Information in the Coordinated Fog-to-Cloud Paradigm", published in *International Journal of Communication Systems (John Wiley & Sons Ltd).* DOI: 10.1002/dac.4286
  - "SFDDM: A Secure Distributed Database Management in Combined Fog-to-Cloud Systems", accepted in *2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (IEEE CAMAD 2019).* DOI: 10.1109/CAMAD.2019.8858458

- ***Job4:*** Designing and defining effective and sophisticated resource management strategies in the F2C paradigm essentially demands to focus on the resource allocation, provisioning and scheduling problem. Especially, allocating the best-fitted resource(s) is not only helps to execute the task(s) adequately but also it implicitly help to design and define a proper resource management strategy. For that reason, before allocating the resource(s), it is essential to forecast their suitability and calculate the fitness ranking for executing some task(s). Considering that fact and achieving our *TO5, TO6,* and *T07*, we given our focus to predicting the resource usages and performances. Then based on the forecasted information and system resources information, we calculate the fitness rank for each system resources. Finally, based on the calculated fitness rank, we allocate the best-fitted system resource(s) for executing the requested task(s) to provide the F2C service(s). However, considering the distributed, hierarchical and coordinated nature of F2C-enabled system, we realize that searching the best-fitted resource(s) is a crucial task. Notably, we comprehend that in F2C-enabled system, the searching and allocating process could be done in two ways: centralized and distributed. Therefore, in this thesis work, we propose and describe both of the ways thoroughly.

Publications -

- "An Architectural Schema for Performance Prediction using Machine Learning in the Fog-to-Cloud Paradigm", accepted in *2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON 2019).* DOI: 10.1109/UEMCON47517.2019.8992939
- "Collaborative learning-based Schema for Predicting Resource Usage and Performance in F2C", submitted in the ranked international conference.

1.5 Delimitations of scope and key assumptions

In this research work, we have explored more than state of the art in the area of resource management in the F2C-enabled SC paradigm. We realized that for achieving all of our objectives, we have to perform several jobs, which have been mentioned in the earlier section. Notably, by performing all of these jobs, not only we achieve all of our objectives but also we able to solve many research challenges. In the next, we are going to describe, which research challenges have been addressed in this dissertation work. Also, we briefly explain our plan for addressing those challenges. Here, we also clarify which research topics have been excluded in this thesis work. In short, we also explain the applicability of our proposals.

For achieving our main objective, initially, we propose and outline the architectural framework for adaptively managing the resources of F2C-enabled system. In that, framework, we introduce different architectural modules (i.e., *Categorization, Resource Sharing, Policy, Information Collector, Adaptive Resource Allocator*) and sub-modules (i.e., *Resource Categorization, Service-Task Categorization, Resource Collector, Perceptive, Mapper, Forecasting*). Notably, all of these modules and sub-modules are collectively functioning to build the adaptive resource management mechanism in the F2C-enabled system. Typically, every modules and sub-modules have different functionalities. Interestingly, some of this module (i.e., *Categorization*) and sub-modules (i.e., *Resource, Service-Task*) have been developed for solving some research challenges (e.g., classification related issues), where the development and functionalities of some others modules (i.e., *Information Collector*) and sub-modules (i.e., *Resource Collector, Perceptive*) creates some research challenge (e.g., secure distribution of data) in the F2C paradigm. However, in this thesis work, we perform several jobs for addressing all of these research challenges.

Selection and allocation of the best-fitted resource(s) is another critical step for making an efficient and sophisticated resource management mechanism. Importantly, learning from the previous knowledge and forecasting the system resource's usability can improve the resource selection and allocation process. Eventually, that also helps to build an adaptive resource management mechanism. Considering this fact, in our proposed architectural framework, we design and develop an architectural module (i.e., *Adaptive Resource Allocator*) which is responsible for performing this step. However, the hierarchical, distributed and combining nature of the F2C computing paradigm, makes a considerable difficulty for the F2C-enabled system to adequately perform the resource selection and allocation process. Following that fact, we realize that the selection and allocation process can either be done in the logically centralized location or can be done collaboratively in the distributed locations. Therefore, in this thesis work, we give our intense attention to continue the investigation the above mentioned two strategies thoroughly. Typically, working on those two strategies help us to propose and build two different architectural schemas for performing the adaptive resource selection and allocation process in the F2C paradigm.

Despite all of these, in any modern computing paradigm, resource sharing operations and system policies are also playing a massive role in the resource management mechanism. Considering this fact, we also present two different modules (i.e., *Resource Sharing, Policy*) in our proposed architectural framework of managing the F2C resources. Since we explained earlier, all the objectives of this work, so we mainly give our focus to solve the three following issues: 1. *Categorization-related issues*, 2. *Monitoring and secure data storing related issues*, and 3. *Adaptively resource selection and allocation related issues*. Therefore, we consider that investigating the resource sharing operations, and defining system policies, are out of the scope of this thesis work. Interestingly, some of the proposals (e.g., Categorization, monitoring mechanism, etc.) of this thesis work have been adopted and implemented in the mF2C project (Towards an Open, Secure, Decentralized and Coordinated Fog-to-Cloud Management Ecosystem) [19], [20].

1.6 Thesis structure

This thesis manuscript has been divided into eight chapters. In the below, we provide a brief description of the content of each chapter.

- **Chapter 1:** Provides a synopsis to this research work developed in this thesis, including primary motivation, problem statement, objectives, the scope of this research and organizational information of this thesis.
  - *Section 1.1* presents the introductory statements about the smart computing paradigm and introduce the Fog-to-Cloud computing concept in the smart computing context. Also, the preliminary motivation of this work has been briefly explained in this section.
  - *Section 1.2* describes the current issues in the Fog-to-Cloud computing paradigm, within a smart computing context.
  - *Section 1.3* explains the generalized and technical objectives of this work.
  - *Section 1.4* introduces all the contributions of the dissertation for fulfilling our generalized and technical objectives and also present the list of outcomes.
  - *Section 1.5* shows the scope of this research work. In this section, we briefly describe and clarify the boundaries that have set for the study.
  - *Section 1.6* manifests the organizational structure of this thesis work.
- **Chapter 2:** Introduces the detailed architectural description of F2C continuum, accentuating its goals and the necessity for coordinating the cloud, edge/fog, and IoT resources. Further, this chapter presents the essentiality for the new resource management schema in the F2C paradigm.
  - *Section 2.1* provides a brief conceptual description and main characteristics of cloud and fog/edge computing paradigm and IoT technology.
  - *Section 2.2* presents the thorough architectural description of combined F2C paradigm. In this section, we define various related concepts (i.e. Fog Area (FA), Fog Manager Layer (FML) and Fog Employee Layer (FEL), etc.) and also identify the functionalities of different layer resources (i.e., Cloud Layer resources, FML resources, FEL resources, etc.).
  - *Section 2.3* highlights the urgency for designing a new and intelligence resource management mechanism in the F2C continuum, in order to improve the smartness for any F2C-enabled smart computing system.
- **Chapter 3:** Mainly conveys the vision for identifying various related works. Significantly, by reviewing those associated works helps to strength the ideas for addressing the various existing issues in the combined Fog-to-Cloud computing paradigm.

- *Section 3.1* provides an extensive literature study about the existing resource management techniques in the various computing platform (i.e., cloud, edge/fog, IoT, etc.). In this section, we mainly give our focus to identify which factors or aspects are majorly related to the resource management mechanism in any major computing paradigm.
- *Section 3.2* presents a comprehensive review of the classification or characterization model for various distributed and smart systems' resources and services. By analyzing the various characterization model, we try to understand how the resource and service characterization or classification model is playing an important role, for designing the effective resource management mechanism in any computing paradigm.
- *Section 3.3* highlights the thorough study for securely distributing the data over the network in various computing paradigm. Mainly, in this section, by reviewing the related works, we try to understand the necessity of having a new secure data distribution framework for the F2C continuum.
- *Section 3.4* investigates various resource allocation and provisioning mechanisms in the different modern computing platform. By thoroughly scrutinizing them, we get a clear vision for designing a self-adaptive resource allocation and provisioning mechanism in F2C platform; which could ensure the improve service execution and reduce the unnecessary resource consumption. That explicitly helps the system to build more intelligent resource management mechanism and improve the smartness for the existing system.
- *Section 3.5* helps to identify the essentiality for having a forecasting module in F2C paradigm. By reviewing, various related works, we realized why the prediction of resource usage is essential for building an effective, and intelligent resource management mechanism. Also, investigating various related works, helping to understand how the resource usage prediction can be applied to the F2C computing system.

• **Chapter 4:** Is devoted to envisioning the newly proposed resource management schema in F2C paradigm.

- *Section 4.1* thoroughly describes the architectural schema of the resource management mechanism in the F2C paradigm. We also comprehensively point-out all the essential functionalities which must be attained by our proposed schema.
- *Section 4.2* introduces the various architectural modules (i.e., Categorization, Information Collector, Adaptive Resource Allocator) and sub-modules (i.e., Resource Categorization, Service-Task Categorization, Resource Collector, Perceptive, Mapper, Forecasting), which are potentially helping to build an effective and advanced resource management mechanism in the F2C paradigm.

• **Chapter 5:** Presents the detailed architectural and functional description of the Categorization Module. Later on, in this chapter, we thoroughly described how our proposal integrated into the EU H2020 funded research project.

- *Section 5.1* describes, the functionalities of Categorization Module. In this section, we comprehensively present how this module is working in our proposing resource management framework. Finally, we justified that making the Categorization Module is the preliminary step for designing the advanced and intelligent resource management mechanism in F2C paradigm.
- *Section 5.2* presents the paths for integrating our proposed Categorization Module in the ongoing European research project (mF2C) [21]. Specifically, in this section, we describe how the various findings from our research works, are helping the ongoing mF2C project to build a more sophisticated and intelligent resource management mechanism.

- **Chapter 6:** Thoroughly demonstrates the insight view of the Information Collector Module. In this chapter, we explain how does the development of Information Collector Module manifesting a new research challenge in the F2C system. Later on, we provide a solution for addressing that challenge.
  - *Section 6.1* describes, how does the Information Collector Module is working to generate the information pool and also revealing a research challenge for distributing this pool over the network. Consecutively, for addressing this challenge, we proposed and developed an architectural schema for securely distribute the information over the network as well as create a secure distributed information pool.
  - *Section 6.2* presents the experimental justification of our proposed and developed architectural schema for securely managing the distributed information pool over the network.
- **Chapter 7:** Presents a comprehensive description of the Adaptive Resource Allocator Module, and its functionalities. Also, in this section, we intensively focus on addressing different related issues and different mechanisms for developing the sophisticated Adaptive Resource Allocator Module. Then, we explain how this module can help to allocate the proper resources.
  - *Section 7.1* provides a comprehensive description of the Adaptive Resource Allocator Module, and its functionalities. In this section, we also present the intelligent ways of resource selection and allocation process in the F2C continuum. Here, also we introduce two different techniques for implementing the machine learning operations to select and allocate the computing resources in F2C paradigm.
  - *Section 7.2* summaries the first technique for implementing the machine learning operations in the F2C paradigm. Initially, we consider all the machine learning operations essentially needed to be done in the cluster-based logically centralized location, where all the data reside. Also, in this section, by presenting some evaluation tests, we justified the effectiveness of our proposal.
  - *Section 7.3* provides a complete insight view of implementing the collaborative machine learning operations in the F2C continuum. In this section, we describe how does collaborative machine learning operations can be adopted in the F2C paradigm. Also, by performing some comparative tests between the cluster-based logically centralized approach and collaborative learning approach, we justified the essentiality of the adoption.
- **Chapter 8:** Suggesting the new avenues for future works and by summarizing the proposed ideas of the thesis draw the concluding remarks.
  - *Section 8.1* summarizes the research outcomes and conclusion of the thesis.
  - *Section 8.2* describe the future opportunities for extending and improve the proposed ideas. So that we can be able to enhance the intelligence of the system.

Chapter 2: Background of Fog-to-Cloud continuum

This chapter aims to provide a thorough structural description of Fog-to-Cloud computing paradigm. We discuss the various structural components of the F2C model and explain how the F2C can be implemented in a smart computing paradigm (e.g., smart city). We also briefly explain how the F2C deployment can help increase the intelligence and effectiveness for any large scale smart computing paradigm. By comprehensively following the characteristics of any smart computing platform and the F2C computing architecture in this chapter, we have given our focus to identify the necessity for designing a novel resource management mechanism for improving the QoS factor and the smartness for the system.

2.1 State-of-the-art: Cloud, Fog/Edge computing, and IoT

According to Masip-Bruin et al. [21], the Fog-to-Cloud (F2C) computing platform is the composite and combined form of three different existing technologies - Cloud computing, Fog/Edge computing and Internet of Things (IoT). Individually, all of these technologies are playing some crucial role in developing a more advanced and intelligent system to improve the quality of lives of this society. So, before moving on to discuss the architectural schema of F2C paradigm, it is relevant to briefly focus on those technologies and identify their characteristics and functionalities.

2.1.1 Cloud computing

In the late 1996, a small technological company named Compaq Computer, first introduced the *Cloud Computing* concept as '*future of internet business*' [22]. Later, as the technology advanced, big giants company like Google and Amazon etc. started using the cloud computing terminology to denote the facilities for unlimited accessing software, computer power, and files over the Web. Realizing the popularity of cloud computing, in 2011, the National Institute of Standards and Technology (NIST) releases the reference architecture and also presented a proper definition of cloud computing framework [23], [24]. According to NIST, *Cloud computing (Fig. 2.1) is a computational model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.* They also mentioned that, in the cloud computing paradigm, resources are logically and conceptually located in a centralized location. With the help of cloud computing facilities, it is possible to easily handle a massive volume of data processing, filtering, aggregation and storage. Typically with the help of three service models (i.e., cloud *Software as a Service (SaaS)*, cloud *Platform as a Service (PaaS)*, and cloud *Infrastructure as a Service (IaaS)*), the cloud can efficiently provide its facilities among the consumers. Through the SaaS model, the cloud ensures to run various applications on the cloud infrastructure for providing the various services among its consumers. Significantly, the SaaS model free the cloud consumers from installing various applications or software in their local machine. Rather than, consumers can access the cloud resources for executing various applications and get the services. Whereas, through the PaaS model, cloud guarantees the consumers for providing an environment to develop, test and deploy their (i.e., consumer) own applications or software. Though, in that case, consumers are only permissible for modifying their developed and deployed software or applications, but they are not legitimate for controlling or modify the existing cloud framework. Therefore the IaaS model exists to provide the facilities of administration and control over the existing cloud framework. With the help of the IaaS service model, cloud helping its consumers for arbitrary deploy and develop the various software and applications by dynamically scale-up and down of existing cloud resources (i.e., Storage, Computing, Network etc.). Most importantly, through the IaaS service model, the cloud is enabling the facilities of the virtualized environment and offering it among its consumers.

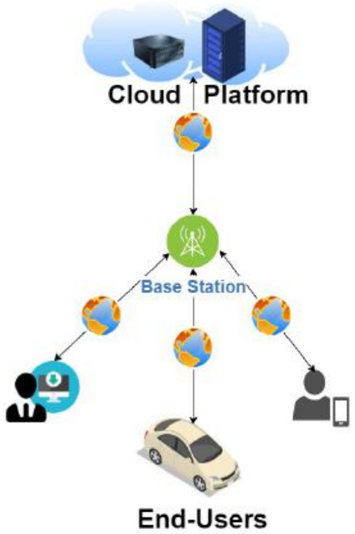


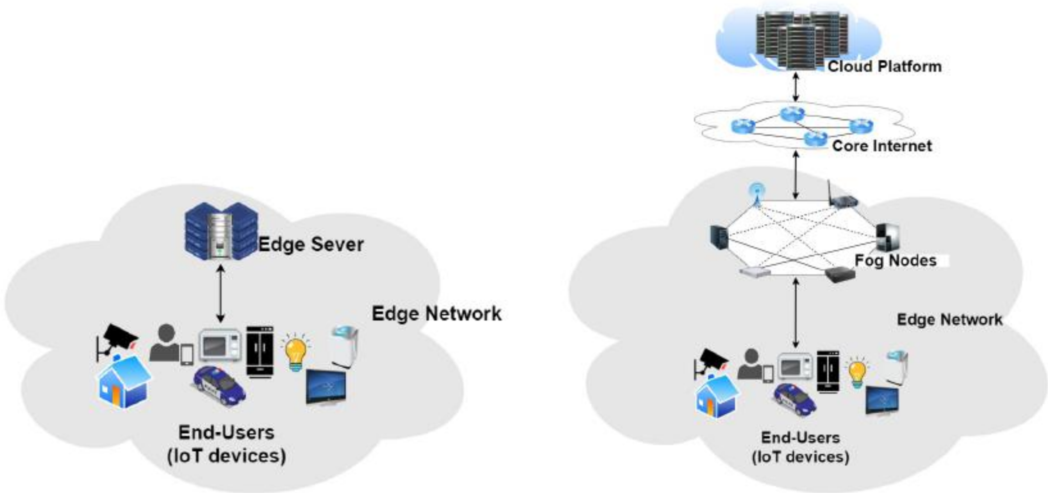
Figure 2.1: Conceptual diagram of Cloud computing (CC)

Conforming to the NIST proposed reference architecture [23], the cloud can be deployed into four different way. Based on the cloud organizational policies and management cloud can be classified into four deployment model, such as: *Public Cloud, Private Cloud, Community Cloud* and *Hybrid Cloud*. For the public cloud model, cloud resources (ex. computing, storage, network, applications, etc.) are typically provided by some third-party service providers. In that case, cloud facilities are accessible for general public users over the web world. Usually, accessing public cloud services are less expensive. Whereas, private cloud infrastructure is solely used for an organization or individual personal purpose. In this case, infrastructural resources, data and applications are only accessible to specific and permitted consumers. Accessing the private cloud facilities are remain to be expensive the public cloud facilities. Significantly, community cloud facilitates organizations with having similar type of requirements for sharing cloud infrastructural resources (ex. storage, processing, etc.) and applications. This kind of cloud model can be managed by some permitted organization members or third-party members. In regards to private cloud facilities, the community cloud services remain to be less expensive for a group of organizations or personals those who have similar requirements. Interestingly, in the cloud deployment model, there is another kind of model is exist, which is known as the hybrid cloud model. Basically, the hybrid cloud is the combination and integrated form of two or more aforementioned cloud deployment model. In the case of the hybrid cloud model, infrastructure resources and applications can be handled by the internal and external cloud consumers. Hybrid cloud model, sensitive data can be stored privately, and it has the facilities to pack up those confidential data externally in the public cloud to recover the data in case of the system failures. Whereas, all the insensitive data remain to be public in the hybrid cloud model.

On-demand self-service facilities, scalability and elasticity nature, unlimited resource (e.g., computing, storage, network etc.) availability and accessing facilities, multi-tenancy nature and many other characteristics of cloud computing turned it to leading and accessible technology into the modern computing paradigm. Though the cloud offers several attractive facilities among its consumers; but it has some disabilities for efficiently providing various latency-sensitive services (ex. real-time traffic monitoring, emergency disaster management, etc.) among the end-users. Conceptually, the cloud computing architecture is centralized, and generally, cloud resources are located far from the end-users. So, that distance not only makes an issue for the reliableness and availability of cloud resources but also create security and privacy concerns for the end-users. So, for addressing these issues and many other open challenges, researchers come up with a new solution in the distributed computing system. They proposed to bring the cloud facilities near to the end-users. So, that is how the concept of edge or fog computing emerged.

2.1.2 Fog and Edge computing

Demand for effectively spreading and bringing the cloud facilities near to the end-users for diminishing the communication latency between the end-users or consumers and cloud resources, along with improved reliability and security-privacy facilities simulated the advent of computing facilities near to the end-users [25]. Many similar concepts have emerged, like as: Edge computing (EC) and Fog computing (FC), etc. Though the 'Fog computing' and 'Edge computing' are both ensuring to bring the computational and storage capacity near to the verge of the network; but these paradigms are not fully identic, and they have the clear and prominent differences [26]. According to the famous cloud strategic consultant David Linthicum, *"Edge computing (Fig. 2.2(a)) brings the computational processing capabilities close to the data source, and it does not need to be sent to a remote cloud or other centralized systems for processing. Whereas Fog computing (Fig. 2.2(b)) is a standard that defines how edge computing should work, and it facilitates the operation of computing, storage and networking services between end devices and cloud computing data centers"* [27]. Moreover, "fog is inclusive of cloud, core, metro, edge, clients, and things, and fog seeks to realize a seamless continuum of computing services from the cloud to the things rather than treating the network edges as isolated computing platforms, and fog envisions a horizontal platform that will support the common fog computing functions for multiple industries and application domains, including but not limited to traditional telco services" [28]. Significantly, the intention and motivation of these two computing paradigms are similar, *bringing the cloud facilities (i.e., computational, storage etc.) near to the end-users so that communication latency between system resources and end-users can be avoided and the issues regarding the availability, reliability and security can be handled appropriately for the end-users*.



(a) Edge computing (b) Fog computing

Figure 2.2: Conceptual diagram of: (a) Edge computing (EC), (b) Fog computing (FC)

Most importantly, fog or edge resources are geographically distributed over the network. So that enables the geo-distributed communication, computation and storage facilities over the Web. As the fog or edge devices have the intercommunication, so they can efficiently manage the data handover for the moving devices. Also, by bringing the computation and storage facilities fog or edge devices are not only securing to provide the latency-sensitive services efficiently but at the same time, it is enabling more provision towards the end-users for securely managing their sensitive personal data. In comparison with the cloud resources, fog, or edge resources are cheaper to access. In both cases (i.e., fog or edge) computation and processing are done at the local place or near to consumers, so usually, the communication cost in the fog or edge remains lower than the cloud. Also, many other features like: scalability, location-awareness, interoperability etc., are helping the fog and edge computing to be the more popular solution for offering location-aware latency-sensitive distributed services (ex. disaster management, real-time traffic monitoring etc.). Besides all of these advantages, some issues exist in the fog and edge computing paradigm. For example, diversity in the fog and edge devices pose a massive challenge for managing and controlling those devices. Unlikely the cloud resources, the fog and edge devices are mainly resource-constrained. So, continuous accessing or using those devices not only limiting their task-execution capabilities but also draining the battery power and reducing their life-span. Thus, efficient task allocation, scheduling and execution are also the essential demands for improving the life-span of the fog or edge devices. Also, in a large-scale IoT-based system, i.e., smart city scenario, as a large number of various devices are participating; therefore it is quite impossible to efficiently provide various location-aware latency-sensitive distributed services for fog and edge computing without any support from cloud computing. Thus the integration and coordination of cloud and fog/edge is a necessary demand for effectively provide the location-aware latency-sensitive distributed services in the large-scale IoT-based smart system.

2.1.3 Internet-of-Things (IoT)

In late 1999, the Internet of Things (IoT) term coined by one of the technocrats of the Procter & Gamble Company (P&G) named Kevin Ashton. The main motivation of IoT development is to enable seamless integration between the cyber world and the physical world [29]. Developing various IoT applications (i.e., Fig. 2.3) not only integrating the surrounding physical devices with the web-world but also improving the quality of living. Most importantly, in IoT, the 'things' can be referred to both as they physical or virtual entity. Basically, IoT is often specified as the integration of various standards and enabling technologies with different sensing, connectivity, actuating, and other capabilities. Also, IoT can be defined as "a network of items, where the sensors are embedded and connected with the internet" [30]. As the IoT has a huge potentiality for designing an intelligent system, therefore it increases the popularity of IoT for use in our daily life. Though the IoT devices offer several attractive facilities; but in a large-scale smart system like - smart city scenario, it is tough to manage and control those devices adequately. Especially the diversity, lack of global standardization in IoT architecture and communication creates a massive challenge for any large-scale smart system scenario. Also, the IoT devices are mainly in-charge for the continuous capturing of various environmental events and transform them into data. Therefore, they are mainly responsible for creating a massive data-flow in a large-scale smart system. Adequate addressing these aforementioned issues and many other open challenges (i.e., security issues) are necessary demands for any smart system. For that purpose, it is also necessary to build an integrated, combined and collaborative computing platform.

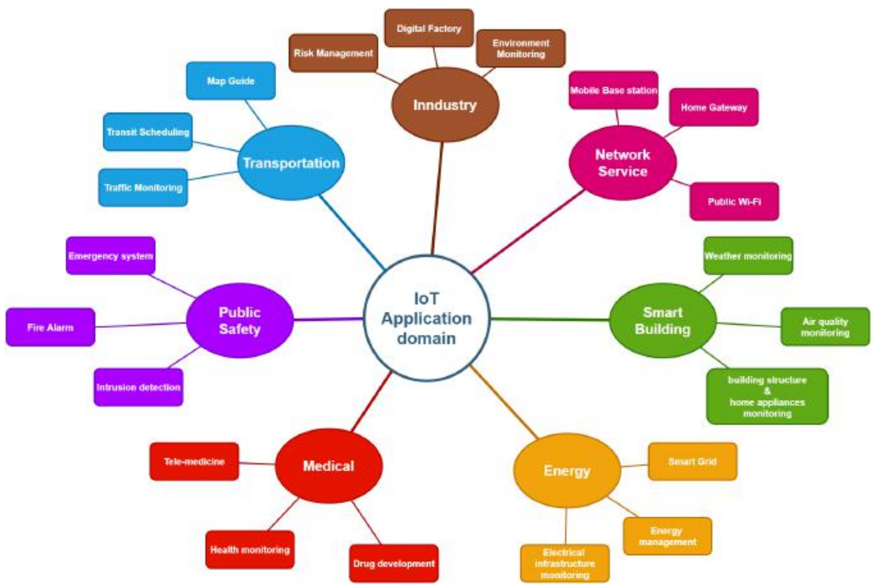


Figure 2.3: IoT application domain

2.2 Structural schema of the F2C paradigm

Following the non-stoppable development of ICT domain, Masip-Bruin et al. proposed the concept of Fog-to-Cloud (F2C) computing in an article [10] in 2016. According to the authors, the F2C has been developed by combining three existing technological facilities: Cloud computing, Fog/Edge computing and Internet of Things. In that paper, they introduced the F2C as the combined and integrated computational framework. But it could be logical that one could have a question that - *why we need such integration?* For answering this kind of question, the authors did a thorough background study of the cloud computing, fog/edge computing and IoT platform. In that article, they correctly pointed out the limitations of cloud and fog/edge computing. Also, in their paper, authors have briefly explained how IoTs become an indispensable pillar of any smart computing platform (e.g., smart city).

All modern ICT developments have the ultimate objective, that is improving the intelligence of existing systems and offer better services among the subscribers in order to improve the quality of lives. Having unlimited resource, storage and computation facilities makes the cloud computing platform as the proprietor in the modern ICT domain. Despite all of these remarkable facilities still, a couple of significant limitations remain unsolved in the cloud paradigm, concerning edge devices. Considering a large-scale smart computing scenario, i.e., smart city, and following the cloud computing structure, the authors have already identified the unsolved issues of cloud-enabled widespread smart computing paradigm. As the edge devices are located far from the cloud resources, it becomes a challenging job for providing the services among the edge devices without compromising the performance metrics. Also, due to the bandwidth limitation in the edge of the network, it is tough for the cloud resources to provide the latency-sensitive services effectively among the edge devices. For addressing these challenges, the fog and edge computing emerge.

The biggest significance of fog/edge computing is bringing the computation capability near to the verge of the network. Which is not only facilitates better utility experience for latency-sensitive services but also helping the overall system to be more advanced and intelligent. Another major characteristic of the fog/edge computing is that it can adequately handle the mobility of end-users. Interestingly, the mobility linked to fog/edge computing bringing the higher level of volatility in case of both processed data and the overall system infrastructure. For a global prospect of views that create some enormous challenges for managing the whole system infrastructure. Even, due to the low computational capabilities and higher volatility nature, it is very much challenging job for the fog/edge devices to ensure a higher value of reliableness and privacy. Also, compared to the cloud computing paradigm, still, the fog/edge computing platform far reaches from having a precise and generalized global business model. So in a large-scale IoT-enabled smart system to overcome these challenges, Masip-Bruin et al. proposed a combinational and integrated computing platform by considering the cloud and fog/edge computing facilities. In their proposed architecture, fog/edge devices are responsible for leveraging the cloud functionalities to end-users, and also ensuring the local process, compute and analyze data that have been captured by small sensory devices. Effectively, with the help of their newly proposed architecture, also a broad set of various latency-sensitive services (e.g., emergency e-health service, disaster management, etc.) can be efficiently offered to the end-users.

According to Masip-Bruin et al., the F2C is a hierarchical, combined and integrated computing platform. To better understand the hierarchical representation of this computing platform, in Fig. 2.4, we depicted the hierarchical architectural representation of the F2C paradigm. Basically, it is the composite of three different layers' of resources. In their proposed computing platform, the upmost layer resources have the highest computational capabilities and unlimited resource facilities (i.e., computing, network, power and storage resource). This uppermost layer is notably known as *Cloud Layer*, and the resources that are working in this layer are called *Cloud Layer resource*. Whereas the bottommost layer of their computing architecture is consist of various and a large number of small sensory and actuating resources. These resources are mainly known as *Edge-IoT device*. The architectural layer where the edge-IoT devices are residing is recognized as *Edge-IoT Layer*. Between these two layers, the *Fog Layer* resources or fog resources are residing. Computing, storage, communication, power etc. capacities of fog

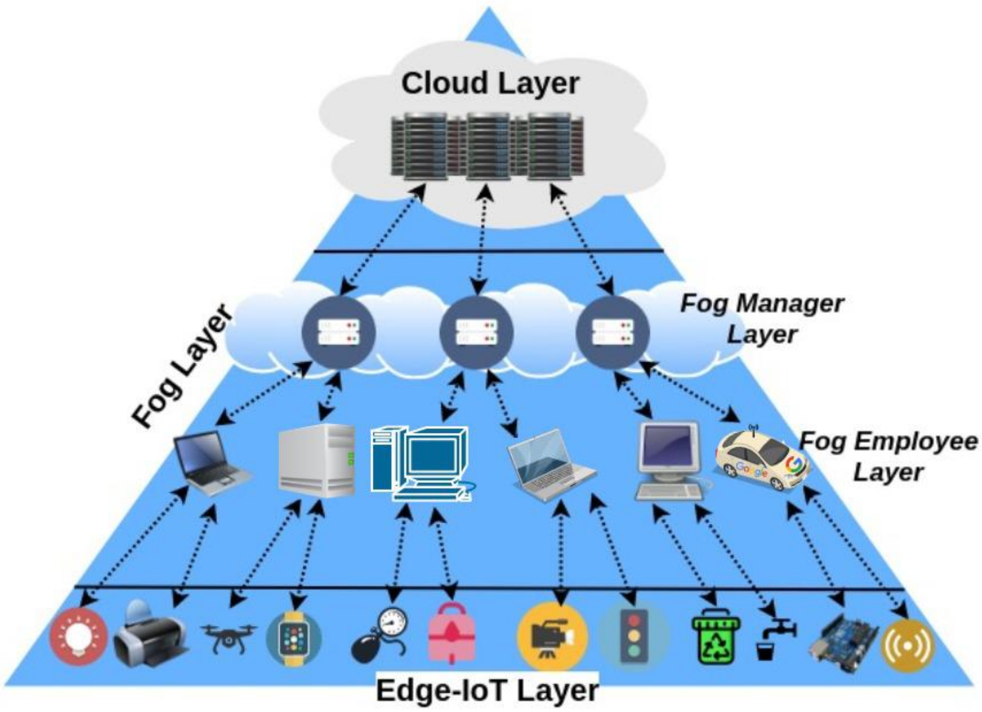


Figure 2.4: Hierarchical F2C architecture

resources are limited. Conforming to Masip-Bruin et al. proposed computing architecture, these resources are working collaboratively with the cloud resources for bringing the cloud functionalities near to the verge of the network.

However, the layered and hierarchical computing architecture is not a new concept in the modern computing paradigm. Even in [31], [32], authors have also proposed some similar computing architecture for extending the cloud facilities to reduce the task-execution time and efficiently provide latency-sensitive services. Interestingly, in these works, the authors have proposed a more static computing platform, where services can only be offered by the separate task-execution in any of one relevant layer of resources. Whereas, in F2C services can dynamically offer by joint task-execution in cloud and fog resources, as well as separate task-execution in the suitable cloud or fog resources. Significantly, this dynamicity feature and the coordinated management facilities of the F2C is not only ensuring to offer better service facilities but giving a huge advantage to the F2C computing platform to create the evident difference with other hierarchical cloud-based computing platforms. Undoubtedly, these features and facilities help the F2C to be the indispensable solution for making more advanced and intelligent large-scale IoT-based system.

For better understanding, how the F2C is incorporating with the large-scale IoT-based system (e.g., smart city) in Fig. 2.5 we have already illustrated an architectural diagram of F2C-enabled smart city scenario. Typically in any modern smart city for accessing city-provided various services, a large number of citizens participating in the system with their various computing devices. Also, following any smart city scenario, one can easily understand that lots of diverse sensors, actuators and networking devices are deployed either by the authorized organizations or by some individuals. Altogether for efficiently managing those devices and effectively provide all the different services to the end-users (i.e., citizens), the F2C become a predominant computing architecture for any large-scale IoT-enable system like as the smart city. From that diagram (Fig. 2.5), it can easily figure out that, in a modern F2C-enabled smart city scenario the several small areas emerge. These small areas are responsible for bringing the F2C facilities near to the verge of the network and helping to efficiently serve city-provided various services among the citizens. These small areas are individually known as Fog Area (FA). Each FA, consist of diverse

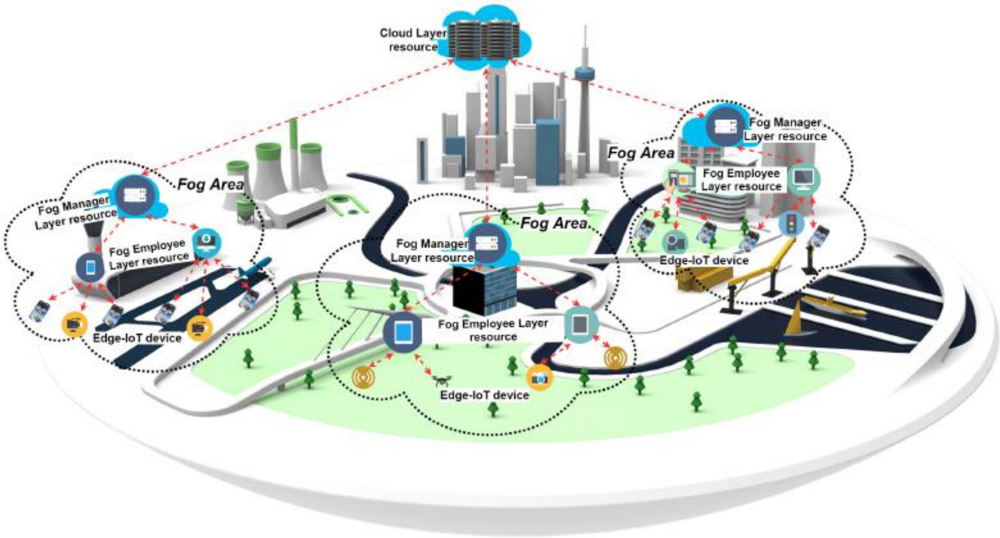


Figure 2.5: F2C-enabled Smart City

fog resources and edge-IoT (i.e., sensor and actuator) devices. From the topological or hierarchical perspective of the F2C paradigm, it can easily identify that for each FA in the F2C-enabled smart city scenario, some of the fog resources are working to manage the corresponding FA's participating devices (i.e., other fog resources and Edge-IoT devices). By deep observation of Fig. 2.4 & 2.5, it can easily recognize that those fog resources are basically acting like manager of other fog resources and edge-IoT devices. Also, these resources are working as the gateway for other fog resources and edge-IoT devices, to communicate with the cloud resources. We named these managerial fog resources as *Fog Manager Layer (FML) resource*. Significantly, FML resources are liable for effectively provide the F2C facilities among the other fog resources, within their scope. Following Fig. 2.5, it also can realize that except those FML resources, also a large number of small computing devices or typically the other fog resources are working in any FA. These resources are mainly responsible for performing some small computational tasks (e.g., initial sensed data processing and aggregation etc.) and bringing other F2C facilities (e.g., security and primary storage facilities for storing captured data) near to the end-users. Also, they are working as the end-point for the Edge-IoT devices. We titled these fog resources as *Fog Employee Layer (FEL) resource*. This classification of fog resources motivates to further classify the *Fog Layer* of F2C architecture into two sub-layers. In Fig. 2.4, considering this fact we have already illustrated those two sub-layers (i.e. *Fog Manager Layer (FML)* & *Fog Employee Layer (FEL)*).

2.3 Essentiality for designing adaptive resource management

The F2C is getting hype for ensuring to bring more intelligence and advancement in the smart computing platform. As technology evolves, the demand for using smart things in our daily life is increasing. Significantly, that needs gradually driving us into a new world, where everything surrounding us is getting connected through the network. New forecasted information comes out from International Data Corporation (IDC) [33], they are predicting that by 2025 more than forty-two (42) billions of various smart connected devices or things would be activated all over the globe. These devices would generate almost seventy-nine (79) zettabytes (ZB) of data in the same year. Most importantly, efficiently handling this massive volume of data and managing the vast amount of diverse devices is a pretty challenging job for any modern computing system. The history of the rise of F2C lies in resolving these challenges. Therefore to keep its commitment and bring more smartness to the existing smart computing platform F2C needs to ensure a more advanced, sophisticated and effective resource management mechanism to adequately manage the connected devices, as well as their generated data.

Distributed, coordinating, and ubiquitous nature of F2C, are helping F2C to be an effective emerging solution for making a more sophisticated and intelligent smart computing paradigm. Most importantly, F2C has been represented as hierarchical, layered and combined computing architecture, where each layer consists of a considerable amount of diverse computing and connected devices. These devices not only have different characteristics and resource capabilities, but also they have different communication technology. They are using various networking technology (e.g., Bluetooth, WiFi, 3G, 4G, Ethernet, etc.) for connecting to other devices as well as the web world. Also, these devices are capable of executing various kind of apps and running various kind of software. Even the small and resource-constraint edge-IoT devices (e.g., sensors and actuators) have a tremendous amount of versatility in their nature. This kind of diversity is not only turning the F2C into the unique computing architecture, but at the same time, it creates a significant issue for the F2C to manage the system resources in an efficient way. Importantly, in any smart computing system, QoS is one of the fundamental metrics to measure the intelligence level of the system. The QoS explicitly depends on the service-level agreement (SLA) of the system. Also, it depends on many factors such as service execution latency, resource performance, reliability, scalability/flexibility, security etc. So it is a pretty tough job for F2C to maintain the high QoS value in the highly distributed and diversified computing platform (e.g., Smart City).

Once in a time, Sir Stephen Hawkins said *"Intelligence is the ability to adapt to change"*. So to bring more smartness and advancement in the existing system, F2C needs to come up with a more sophisticated, dynamic and adaptive resource management mechanism. Therefore, it can be possible to utilize all the system resources effectively and reduce unnecessary resource consumption without degrading the QoS metrics. Considering all of these facts and demands, we realize that, it is a necessary and most urgent job to design an effective and adaptive resource management solution for the F2C paradigm.

Chapter 3: State of the art

State of the art is the level of highest development achieved in a specific market, application domain, science or technology within a particular time. State of the art often refers as SOTA. Da Anunciação Marco author of the 'My Blog and You', quoted that "State of the Art is the frenetic and relentless pursuit of doing what its best at that time!". SOTA is nothing but a thorough analysis of the most recent contributions to a field. In this chapter, we mostly scrutinize all the potential research works for identifying their contributions to build efficient resource management mechanism in various existing computing platforms (i.e., Cloud computing, Edge computing, Fog computing etc.). The outcome of this chapter implicitly helps us to materialize our ultimate vision.

3.1 Resource management in modern computing paradigm

Resource Management (RM) - a series of processes which deals with the accumulation and release of resources [34]. Efficient management or utilization of system resources is the primary steps for making an advanced and sophisticated computing paradigm. Interestingly, efficient resource management involves discovering and identifying all available system resources, selecting appropriate resources, and partitioning and provisioning them to optimize the utility function which can be in terms of overall performance, cost, energy efficiency, information accuracy, coverage, reliability, etc. [35]. However, designing and defining an efficient resource management mechanism is not an easy task for any large-scale distributed computing system. Notably, overall diversification can pose a massive hurdle to effectively manage the system resource of any modern distributed computing platform [36]. For example, F2C is a combined and distributed computing architecture, which has been formed by the integration of cloud, edge/fog and IoT resources [10]. So, diversification is a common characteristic of any F2C-enabled system. Also, F2C represented as a hierarchical computing architecture. Thus, diversity, hierarchical and many other natures (e.g., dynamism) of F2C create a colossal objection for effectively managing the system resources.

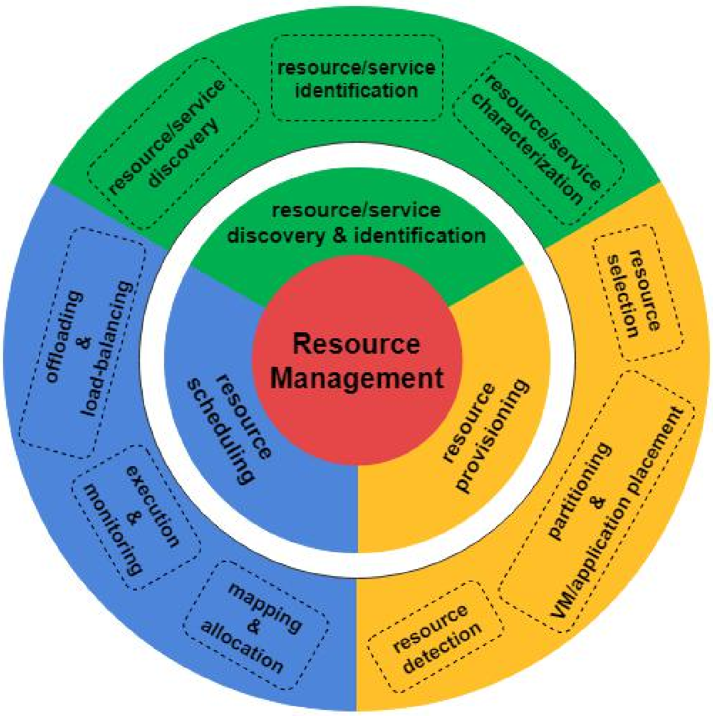


Figure 3.1: Taxonomy for various resource management courses in any distributed system

Although, resource management mechanism of any distributed system is an integrated process of accomplishing many several courses. Effective accomplishment of those courses potentially solved many issues for designing effective and advanced resource management mechanism of any computing paradigm. After performing an exhaustive literature survey, we realized that all of these courses could be classified as follows - 1. *resource/service discovery & identification, related courses*, 2. *resource provisioning related courses*, and 3. *resource scheduling related courses* [36]–[39]. The classification of those courses is depicted in Fig. 3.1. Significantly, properly attaining all these three courses can solve many issues for managing the system resources. For example, effective resource/service discovery and identification mechanism can lead to determining the characteristics of system-involving resources and services.

Discovery and identification of system resources and services are the initial steps for designing and defining advanced resource management mechanism in any distributed computing system [40]. Interestingly, after doing comprehensive literature study in various computing platform (e.g., cloud, fog/edge, IoT etc.), we realized that the exploration of resources and services implicitly helps to grasp in-depth knowledge about the types and features of them. Importantly, that in-depth knowledge helps to build a better mechanism for managing the system involving resources and services. Significantly, in any service-oriented and distributed computing system (i.e., cloud, edge, fog, etc.), the concept of *task* and *service* are closely related to each other [41]. Services (i.e., e-Health, e-Banking, e-Learning, etc.) can be offered by executing some tasks in various computing resources. However, resource-requirements of each task is different. Therefore, without perceiving the task requirements, it is also a difficult job for organizing and managing the system resource in accordance with tasks demand. Thus, knowing the characteristics of services is also helpful for understanding the task's involvement. So, before establishing a proper resource management schema in any service-oriented and distributed computing system, it is also necessary to know the system resource classification as well as identify the system-involving service characteristics and their corresponding tasks requirement [38], [42]–[45].

Resource provisioning is another most essential course of the whole resource management procedure. The meaning of resource provisioning is detection, selection and arrangement of appropriate resources in accordance with the service/task requirements [46]. Optimal provisioning of the resources can ensure better utilization and reduce the unnecessary consumption of resources [47]. Interestingly, in any large-scale distributed computing platform (e.g., Smart City), heterogeneity of system resources makes a tremendous obstacle to choose the appropriate resources to satisfy the service/task demands [36]. Unfortunately, inefficient provisioning of resources degrades the overall performance of any computing platform. Especially, under-provisioning of resources may cause the service-level agreement (SLA) violation, and over-provisioning of resources may force to increase the unnecessary resource consumption. Most importantly, for both of the cases, the QoS factor can be degraded for the overall system [48]. In the different computing paradigm, different techniques are helping to arrange the resources effectively. For example, virtualization is playing a crucial role in the cloud computing platform [49]. The basic idea of the virtualization is to managing the workload of cloud resources to make the overall platform more effective, efficient, economical, and scalable. Most importantly, with the help of Virtual Machines (VMs), the cloud platform can able to offer a dynamic and scalable environment to offer on-demand services. Therefore, VMs placement is a critical issue for making an efficient resource provisioning mechanism in the cloud platform [50]. Whereas, fog or edge computing extends the cloud facilities by placing applications closer to the end-users and edge devices [51]. Notably, the placement of those applications has an essential influence on the performance of the computing architecture [52]. So, adequate addressing of all these issues can lead to building a proper resource provisioning mechanism in any coordinated computing system, like F2C.

Besides the resource provisioning, and resource/service discovery and identification; resource scheduling is another required course for the resource management process. Basically, it is the itinerary for managing the events and system resources [53]. Most importantly, the utility function (i.e., performance, cost, energy-efficiency, information accuracy, coverage, reliability, etc.) of any system heavily depends on the resource scheduling mechanism [54]. Proper allocation of resources, effective execution of tasks, and efficient arrangement of workload balancing among the system resources help to define a proper resource scheduling mechanism in any large-scale distributed computing system [55]. Interestingly, machine learning techniques have the most significant influence on the accomplishment of these three main courses. Advance knowing of workload predictions, service requirements and resource behaviours; help to select the appropriate resources for executing the tasks for efficiently offer the services in large-scale distributed systems [56].

Considering all of these facts and all the open issues in the F2C-enabled smart system; we decide to given our focus for addressing some of the matters, for building a framework of adaptive resource management mechanism in F2C-enabled smart systems. For that purpose, in the remaining sections of this chapter, we investigate some of the related works. Notably, all of these works have been classified into four groups, as follows: a) *Classification and Characterization*, related works; b) *Monitoring and Secure data distribution* related works; c)*Resource allocation mechanism*, related works; d) *Resource usage forecasting and performance prediction*, related works.

3.2 Classification and Characterization

Identifying the characteristics of resources and services and also determining the task's requirements, are the initial steps for making a proper resource management schema in any computing system. Therefore, in the next, following up various related works in different computing platforms (e.g. cloud, fog and edge, grid, mobile computing, etc.), and technologies (e.g. WSN, IoT, SDN), we tried to realize the core strategies for classifying system resources and services-tasks.

3.2.1 In resource aspects

For any computing platform, proper utilization of resources undoubtedly facilitates an optimal task execution for offering the services, and hence helps to build a useful resource management solution. Most importantly, to manage the whole set of system resources, it is very much essential to identify their characteristics and properly classified them to make a resource catalogue. Thus, it is necessary to determine a resource classification and taxonomy, for a scenario combining fog and cloud resources, like the one, we envisioned by the F2C. Notably, we have not found too many past works, where researchers have done their investigation over the combined, coordinated and hierarchical computing platform like F2C. Therefore, we considered most of the related computing platforms (i.e., cloud, fog and edge, grid, mobile computing, etc.) and technologies (i.e., IoT, WSN, etc.), which could be the part of the smart computing system and typically involved for making the broad, integrated and coordinated F2C computing platform. In our previous work [57], we put together a comprehensive literature survey, highlighting the resource characteristics for distinct computing paradigms and also observed several interesting findings. We found that each researcher has different interests and objectives when defining, or suggesting their resources classification. Some of them from a system architecture perspective and according to their platform scope (cloud, fog, IoT, etc.), some others from a resources and services management perspective, or some others concerned about understanding the system behaviours and features, and addressing the various challenges of those systems. Also, in some cases, we found that the researchers have focused on designing or modelling the architecture of the smart computing platform ([58], [59]). So, following all of these research papers and also investigating the reference architectures of various joint-venture research works ([60]–[67]) we can identify some parameters, which are potentially helping us to identify the resource characteristics of any smart computing system properly. Following are the parameters:

- Hardware: Processor, Memory, Storage, Power, GPU, and FPGA.
- Software: Operating System, Application and APIs, and Database Product Information.
- Network: Bandwidth, Technology and Standards, Networking System, Communicating Techniques, and Deployment Model.

- Security: Authenticity, Intrusion, Privacy, Trust, Encryption, Denial of Service attack, Identification, Accountability, Confidentiality, and Authorization.
- Data: Source and Owner, Type and Size, Content-Format, Analysis Type, Processing Framework or Infrastructure. Also, the type of Sensors, Actuators, and RFID tags are significant, but these all are explicitly related to Data.
- Cost: Computation, Communication, and Deployment.
- Features and Behaviors: Scalability, Fault-Tolerance, Location-Awareness, Availability, Agility / Real-Time, Autonomy, Mobility, Multi-Tenancy, Reliability, Context Information, Programmability and Virtualization, Ubiquitous, Heterogeneity, On-Demand, Geo-Distribution, Flexibility / Interoperability, Transparency / Openness, and Proximity.

According to the upper mentioned parameters, we have built all of our finding tables (Table 3.1, 3.2, 3.3, and 3.4), where all the fields for these categories (columns in the table) have been filled up according to the description of each research work (rows in each table). The box corresponding to each category (i.e., cloud, fog and edge, IoT, and other computing platforms) which is addressed in the associated research work has been shadowed, so providing a graphical snapshot of what categories are more relevant for the different research areas and, therefore, allowing the extraction of several valuable findings, as presented in the next.

	Features & Behaviors																			Cost			Data					Security			Network					Software			Hardware				
Paper Reference Number	Proximity	Transparency & Openness	Flexibility / Interoperability	Geo-Distribution	On-Demand	Heterogenity	Ubiquitous	Programmability & Virtualization	Context Information	Reliability	Multi-tenancy	Mobility	Autonomy	Agility / Real-time	Availability	Location Awareness	Fault Tolurence	Scalability	Deployment	Computation	Communicating	I/O Type	Analysis Type	Content Format	Type & Size	Source & Owner	Methodology	Requirements	Attack	Deployment Models	Communicating Techniques	Technology & Standrads	Applicable Networking System	Bandwidth	DataBase product Info	Apps & APIs	OS	GPU & FGPA	Power	Processor	Storage	Memory
	Explicitly mentioned about the Resource characterisics																																									
[69]		✓		✓	✓		✓					✓		✓		✓	✓												✓	✓		✓			✓				✓	
[79]		✓		✓	✓		✓							✓			✓									✓		✓						✓	✓			✓	✓	✓
[71]		✓		✓			✓		✓					✓		✓																✓						✓	✓	✓
[38]				✓	✓		✓							✓			✓	✓	✓	✓												✓		✓	✓	✓				✓	✓	
[68]		✓		✓		✓	✓	✓	✓	✓		✓		✓		✓		✓	✓	✓								✓		✓	✓				✓		✓					
[74]		✓		✓	✓		✓	✓	✓			✓		✓		✓	✓	✓	✓	✓															✓	✓				✓	✓	✓
[75]		✓		✓	✓		✓	✓		✓				✓		✓	✓													✓					✓				✓	✓	✓	✓
[76]		✓		✓	✓		✓		✓			✓		✓		✓	✓													✓					✓							
[81]							✓												✓	✓	✓									✓										✓	✓	✓	✓
[77]					✓									✓				✓	✓	✓						✓			✓				✓						✓	✓	✓
[83]							✓																											✓						✓	✓	✓
[82]							✓					✓						✓	✓	✓									✓		✓		✓						✓	✓	✓
[78]							✓													✓													✓			✓			✓	✓	✓
	Implicitly mentioned about the Resource characterisics																																									
[80]		✓	✓	✓	✓	✓	✓	✓	✓	✓		✓		✓		✓	✓				✓					✓	✓							✓	✓	✓	✓		✓	✓	
[70]		✓		✓	✓	✓	✓		✓			✓		✓		✓	✓	✓	✓	✓															✓	✓	✓			✓	✓	✓
[73]	✓	✓		✓	✓	✓	✓	✓	✓		✓	✓		✓	✓	✓	✓	✓	✓	✓						✓	✓			✓	✓		✓	✓	✓							
[43]		✓	✓	✓	✓		✓	✓	✓	✓	✓	✓		✓		✓	✓											✓		✓				✓								
[84]			✓	✓	✓		✓		✓										✓	✓	✓	✓																			✓	✓	✓
[85]				✓			✓																																		✓	✓	✓

Table 3.1: Finding table for Cloud-associated works

Cloud-associated works: Cloud computing is an architectural model that leverages on-demand provisioning and management of computing, and which defines the set of good practices for building a flexible and dynamic set of computing resources to be delivered as a service. In our investigation, we have found that many researchers have given their focus for understanding the characteristics of cloud resources [68]–[78]. Also, some of the works have been done on managing the cloud resources [43], [74], [79], and cloud services [72]. However, in these works, researchers have also focused on discussing the cloud paradigm architecture and service models or identifying the challenges for cloud computing. Thus, it helps to understand the characteristics of cloud resources. Notably, we have found a couple of research works focused on identifying the characteristics of the cloud computing paradigm [38], [80]. We have also found, in [70], [81], that the researchers have presented some basic concepts for the definition of a pricing model for the cloud. There are lots of research works which identify the challenges and issues for the cloud computing platform [43], [73]–[76], [79] and, most importantly, we have found that lots of work have been focused on managing the cloud resources and services [38], [69], [72], [74], [75], [79]. Interestingly, in our investigation, we have found that there are many research works ([81]–[83]), where researchers have explicitly presented the resource characteristics and classifications for the cloud paradigm. We have also found some of the research works ([84], [85]) where initially they present the different strategy for managing the resources in cloud paradigm and then, they implicitly present the characteristics of the resources of the cloud paradigm. Investigating over all these works, helped us to identify the different features and characteristics of cloud resources. Notably, from the finding table 3.1, we can easily understand that besides the boundless resource (e.g., RAM, CPU, storage) availability nature, the virtualization and on-demand availability are some of the key features of cloud resources. We also found that cost (i.e., charges for access and utilization of resources), is one of the crucial aspects for classifying the cloud resources [72]. Indeed, several works in cloud computing have proposed some cost models for system resources and services [68], [72].

	Features & Behaviors																		Cost			Data					Security			Network					Software			Hardware				
Paper Reference Number	Proximity	Transparency & Openness	Flexibility / Interoperability	Geo-Distribution	On-Demand	Heterogenity	Ubiquitous	Programmability & Virtualization	Context Information	Reliability	Multi-tenancy	Mobility	Autonomy	Agility / Real-time	Availability	Location Awareness	Fault Tolerence	Scalability	Deployment	Computation	Communicating	I/O Type	Analysis Type	Content Format	Type & Size	Source & Owner	Methodology	Requirements	Attack	Deployment Models	Communicating Techniques	Technology & Standrads	Applicable Networking System	Bandwidth	DataBase product Info	Apps & APIs	OS	GPU & FGPA	Power	Processor	Storage	Memory
	Explicitly mentioned about the Resource characterisics																																									
[86]	✓			✓		✓		✓		✓	✓	✓	✓	✓		✓	✓	✓																	✓	✓				✓	✓	✓
[92]	✓		✓	✓		✓		✓	✓	✓	✓	✓		✓	✓	✓		✓										✓		✓		✓		✓					✓			✓
[93]										✓		✓										✓					✓	✓			✓	✓		✓		✓	✓		✓	✓	✓	✓
[94]	✓			✓		✓	✓	✓		✓	✓	✓		✓	✓		✓	✓	✓	✓	✓	✓			✓		✓	✓			✓	✓							✓	✓	✓	✓
	Implicitly mentioned about the Resource characterisics																																									
[91]	✓		✓	✓			✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓				✓		✓	✓	✓	✓			✓	✓			✓		✓	✓	✓	✓
[87]	✓	✓	✓	✓		✓		✓		✓	✓	✓	✓	✓	✓	✓	✓	✓														✓	✓						✓	✓	✓	✓
[95]	✓		✓	✓		✓	✓	✓		✓		✓		✓		✓		✓								✓	✓	✓	✓					✓					✓		✓	
[96]	✓		✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓					✓					✓				✓		✓				✓				
[88]	✓	✓	✓	✓		✓		✓			✓	✓		✓		✓		✓										✓				✓		✓	✓	✓				✓	✓	✓
[25]	✓		✓	✓		✓		✓	✓			✓		✓		✓																✓		✓						✓	✓	✓
[89]	✓		✓	✓		✓		✓	✓	✓		✓	✓	✓		✓		✓										✓		✓	✓	✓		✓		✓	✓		✓	✓		
[100]	✓			✓		✓	✓	✓	✓			✓		✓		✓		✓														✓										
[101]				✓		✓		✓	✓			✓		✓											✓						✓	✓		✓		✓	✓			✓	✓	✓
[97]				✓				✓		✓		✓		✓														✓				✓										
[98]										✓					✓												✓	✓						✓						✓	✓	✓
[99]	✓		✓	✓		✓	✓			✓				✓	✓			✓				✓						✓	✓		✓	✓										
[90]	✓			✓								✓		✓		✓			✓									✓			✓			✓					✓	✓	✓	✓

Table 3.2: Finding table for Fog and Edge associated works

Fog and Edge associated works: Several authors have published their work with some contributions related to resources categorization in the scope of fog and edge computing. In [86]–[90], the authors present mainly some architectural contributions and, as part of these, they address some classification issues. In [91]–[94] the researchers propose explicitly one taxonomy for fog computing platforms. Besides, we have observed that many researchers focus on identifying features, behaviours and attributes in fog computing, as in [25], [88], [89], [92], [95], [96], where they discuss their challenges and characteristics. Interestingly, in [93], they present a taxonomy for a connected object, which is mostly suggested as the taxonomy for edge and fog devices.

Note that in our literature, we have not only focused on those papers discussing the classification or taxonomy of the resources, but also on those research works which concentrate on managerial-related issues in the fog computing paradigm. Managing resources and services properly is essential to run a system efficiently. For instance, in [86], the authors briefly discuss some policy-based service orchestration techniques to manage the services in the fog computing paradigm efficiently. Authors in [89], [92], [96] present a resource management strategy for fog computing systems, wherein [92] authors are focused on devices management in the fog computing paradigm, and in [89], [96] they discuss on the resources and network management in fog computing platforms. In this literature review, we have also found that most researchers consider some use-cases to present their work and, as part of this, they also mention some particular types of devices, which are involved in their proposed system [25], [86], [87], [89], [91], [92]. In some papers, we have also seen that authors have already shown their concerns to overcome from mobility-related issues ([97]) and security-privacy related issues ([98], [99]). Also, authors in [91]–[94] present some taxonomy for fog computing platforms by considering various use-cases and applications. Impressively, we also found a few works [100], [101], where the researchers have expressly shown their concerns for identifying the characteristics of edge resources. However, most interestingly, we have not seen any research work where researchers have explicitly presented some general ontology for a complete fog computing platform, though we found a couple of research works where researchers have given some classifications and taxonomy for such platforms and their resources. As a summary, in table 3.2, summarizes our findings to identify the characteristics of fog and edge resources. Notably, we found that edge and fog, resources are geographically distributed over the network [57], [80], [89]. Thus, the distribution nature elevates to provide real-time services in edge or fog computing platform. In summary, we may quickly assess that there are a significant variety and diversity of system resources, which undoubtedly makes resource categorization a challenging task for fog and edge paradigms.

IoT associated works: IoT is a revolutionary concept, which immensely changes the traditional computing and networking system. In our investigation, we have found mainly researchers have given their focus to solve the architectural and managerial related issues in the IoT paradigm. For fulfilling their objectives, they have mainly done their research on various IoT-application domains and use-cases. Interestingly, by performing their research work on those application domains and use-cases, the researchers propose their views, taxonomy, classification and ontology for the IoT resources. Like in [45], [102]–[108], researchers have proposed resource management solutions. Also, in their work, they have identified various kind of resource model for IoT paradigms. Besides, we also found some works [45], [102], [103], [105], [106], [109]–[111], where researchers have explicitly given their concern for identifying the features of IoT resources. We have also found some other works [45], [105], [107], [108], [110], [111], where researchers have given their concern to manage the IoT system resources efficiently. More interestingly, we have also found some research works [105], [109], [110] where researchers propose some classification model for IoT resources, by considering different parameters. Notably, in some other works [103], [104], authors discuss the ontology of the IoT paradigm.

Reviewing these works in the IoT paradigm, we found that in most cases, the researchers have given their focuses to identify the functional characteristics of their system resources. In some works ([102], [110], [112]), researchers have explicitly defined the functional characteristics. Whereas in other works, either by describing the whole computing paradigm ([113]–[115]) or reviewing the various related works ([108], [109], [116]), the authors have implicitly presented the functional characteristics of the IoT resources. We found that in most of the IoT based research works, the researchers have also given their focuses to identify the data-related information (I/O related, data size and source information). In our literature, we also found that some of the researchers have focused on ensuring network security and data privacy in the IoT platform ([105], [117], [118]). Also, we have seen that some of the researchers have explicitly or implicitly described the hardware characteristics of their system resources ([45], [113], [118]–[121]) and discussed the various networking or communicating standards and techniques ([45], [109], [114], [116]) which are involved in the IoT platform. So, in summary, considering all of

	Features & Behaviors																		Cost			Data					Security			Network					Software			Hardware				
Paper Reference Number	Proximity	Transparency & Openness	Flexibility / Interoperability	Geo-Distribution	On-Demand	Heterogenity	Ubiquitous	Programmability & Virtualization	Context Information	Reliability	Multi-tenancy	Mobility	Autonomy	Agility / Real-time	Availability	Location Awareness	Fault Tolurence	Scalability	Deployment	Computation	Communicating	I/O Type	Analysis Type	Content Format	Type & Size	Source & Owner	Methodology	Requirements	Attack	Deployment Models	Communicating Techniques	Technology & Standrads	Applicable Networking System	Bandwidth	DataBase product Info	Apps & APIs	OS	GPU & FGPA	Power	Processor	Storage	Memory
	Explicitly mentioned about the Resource characterisics																																									
[102]	✔	✔	✔	✔		✔	✔	✔	✔	✔		✔	✔	✔	✔	✔	✔	✔				✔										✔					✔	✔		✔		
[103]				✔		✔	✔		✔					✔		✔						✔		✔	✔																	
[104]						✔								✔		✔						✔						✔														
[106]				✔		✔	✔							✔	✔	✔						✔																				
[107]		✔	✔			✔		✔		✔				✔		✔		✔				✔										✔		✔		✔						
[45]	✔	✔	✔	✔		✔	✔			✔		✔	✔			✔	✔	✔				✔						✔		✔	✔	✔				✔			✔	✔	✔	✔
[110]	✔			✔		✔				✔		✔	✔	✔	✔	✔						✔												✔					✔			
[111]						✔	✔									✔						✔								✔												
[119]			✔			✔								✔																	✔						✔		✔	✔	✔	✔
[112]										✔				✔	✔					✔																						
	Implicitly mentioned about the Resource characterisics																																									
[109]	✔	✔	✔	✔		✔				✔			✔	✔		✔	✔	✔	✔		✔	✔								✔	✔	✔	✔	✔								
[105]			✔	✔		✔	✔		✔	✔		✔	✔	✔	✔	✔		✔				✔	✔		✔		✔	✔	✔			✔		✔					✔			
[108]	✔						✔	✔	✔	✔		✔				✔												✔				✔							✔			
[115]			✔			✔		✔														✔		✔								✔										
[120]						✔																					✔		✔										✔	✔	✔	✔
[114]	✔					✔		✔						✔								✔									✔	✔				✔						
[121]									✔					✔				✔		✔					✔			✔			✔			✔					✔	✔	✔	✔
[117]						✔		✔																	✔		✔	✔	✔		✔											
[113]				✔		✔	✔		✔	✔			✔	✔	✔	✔		✔							✔			✔											✔	✔	✔	✔
[116]	✔		✔			✔	✔						✔			✔		✔				✔			✔		✔				✔											
[118]																											✔		✔		✔	✔							✔	✔	✔	✔

Table 3.3: Finding table for IoT-associated works

these works in table 3.3, we have summarized the characteristics of system resources of various IoT-based system domain. Significantly, after investigations of various IoT-related works, we realized that, though the IoT devices are resource-constrained; but they are the primary source of producing a large amount of data in any large-scale computing system (e.g., smart city). Typically, these data are the main ingredient for the various services (e.g., e-health, traffic monitoring, etc.), which are offered by the large-scale computing system (e.g., smart city).

Other Computing platform associated works: Besides the cloud, fog and edge computing platforms and IoT technology there are many other computing platforms (e.g., mobile computing), and technologies (e.g., WSN), might potentially involved in making a large-scale smart system (e.g. smart city). However, investigating of these works could somehow enhance the findings obtained to achieve our main objectives. Therefore, we have given our focus on some research works on the following computing platforms: grid computing (e.g., [122]) and mobile computing (e.g. [123]), and other technologies: wireless sensor networks (WSN) (e.g., [124]) and Software-defined networking (SDN) (e.g., [125]). As our primary intention is to identify the classification, taxonomy and ontology of system resources of various computing paradigms, here we have put our focuses on those works where researchers have given more concern to identify the resource classification and propose some taxonomy, and following on the architectural and managerial issues of those systems.

Investigating on these works, we have found that in some cases ([122], [124]–[129]) the authors have explicitly put their focus to identifying the classification, characterization, or taxonomy for their system resources. Also, we have found some works ([130]–[135]) that by discussing the system architecture or by proposing the computational

	Features & Behaviors																		Cost			Data					Security			Network					Software			Hardware				
Paper Reference Number	Proximity	Transparency & Openness	Flexibility / Interoperability	Geo-Distribution	On-Demand	Heterogenity	Ubiquitous	Programmability & Virtualization	Context Information	Reliability	Multi-tenancy	Mobility	Autonomy	Agility / Real-time	Availability	Location Awareness	Fault Tolurence	Scalability	Deployment	Computation	Communicating	I/O Type	Analysis Type	Content Format	Type & Size	Source & Owner	Methodology	Requirements	Attack	Deployment Models	Communicating Techniques	Technology & Standrads	Applicable Networking System	Bandwidth	DataBase product info	Apps & APIs	OS	GPU & FGPA	Power	Processor	Storage	Memory
	Explicitly mentioned about the Resource characterisics																																									
[126]		✓		✓		✓		✓							✓	✓		✓																							✓	✓
[122]			✓	✓		✓		✓					✓		✓	✓		✓													✓	✓		✓		✓	✓			✓	✓	✓
[124]				✓		✓								✓		✓	✓	✓																						✓		✓
[127]				✓				✓	✓	✓		✓			✓	✓										✓						✓							✓	✓	✓	✓
[128]										✓				✓	✓		✓								✓									✓								✓
[125]								✓								✓															✓	✓		✓						✓	✓	
[137]										✓						✓									✓		✓	✓	✓		✓								✓	✓		✓
[129]									✓							✓						✓					✓				✓	✓		✓								
	Implicitly mentioned about the Resource characterisics																																									
[130]	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓						✓								✓	✓	✓								✓	✓	✓
[131]		✓	✓		✓	✓		✓							✓	✓						✓								✓												
[134]								✓							✓	✓				✓												✓		✓					✓	✓		✓
[135]			✓	✓					✓																						✓			✓					✓	✓	✓	✓
[132]				✓						✓				✓	✓			✓							✓		✓				✓	✓	✓	✓								
[136]						✓	✓		✓								✓											✓			✓											
[123]			✓			✓			✓	✓						✓						✓						✓			✓	✓										
[138]									✓								✓										✓			✓						✓	✓			✓		✓
[133]		✓		✓		✓		✓										✓				✓			✓							✓							✓			
[139]										✓																	✓				✓	✓										

Table 3.4: Finding table for Other Computing platform associated works

model in various computing paradigms, they have implicitly considered the characteristics of their system resources. Even in some works, we found that authors did some extensive survey ([123], [136]) to identify the various challenges and gaps of their considering system, and then proposed some new solutions ([134], [135]) to address them and overcome those challenges. Interestingly, we found in various computing platforms and research works that researchers have given their profound concerns to provide the best possible security-privacy features for the end devices ([123], [132], [136]–[139]) and tried to find out a solution for reducing the energy consumption for their system resources ([133]–[135], [139]). So, considering all of these works, in table 3.4 we have presented a summary table to understand the resource characteristics, classification or taxonomy in other related computing paradigms.

After analyzing all the finding tables, different synoptic findings can be extracted. We have organized these findings following the same categories used to organize the table, which is: Hardware related findings, Software related findings, Networking related findings, Security-related findings, Data related findings, IoT devices related findings, Cost-related findings, and Features related findings. Next, we present each of these findings.

- **Hardware related findings**. We have observed that most researchers in fog/edge computing and IoT based research works are concerned about efficient power management. That has sense, since many small devices at the edge (for instance, smartphones) as well as many sensors, are battery-powered. In addition, except for the IoT based research works, almost every other project considers the hardware components (i.e., processor, memory, storage) in the system architecture definition. Generally, IoT based research works do not consider hardware components in detail due to the simplicity of its components, except in [45], [119]; in these two papers, the researchers have explicitly mentioned the resource classification of IoT paradigm. Where in [45] researchers proposed some resources management strategy and then measure their efficiency, and in

[119] the authors have explicitly explained the characteristics of IoT devices to provide the classification of OSes for low-end devices. Beside that, we also found in the IoT domain, in some works ([113], [118]) that researchers have implicitly explained the resource characterization for the IoT paradigm. Also interestingly, we have only found three papers ([80], [96], [102]) where researchers consider other hardware components, such as GPU and FPGA, to represent the characteristics of their considered computing platform.

- **Software related findings**. Concerning the software related issues, we have found that software is not essential in most IoT based research. Exceptionally, only in a few of these works ([45], [107], [114], [119]), the authors have considered the APIs and OSes involved in their system resources to create a catalogue of services, and in [102] to understand the characteristics of system resources. Conversely, this attributes category gains relevancy in the cloud-related systems, where the software to be classified and organized in more detailed. It is also relevant for some fog computing-based research works ([86], [88], [89], [93]) where researchers are concerned about identifying the various heterogeneous OS, apps, and APIs involved with the fog and edge devices. Interestingly, we also found that in [122] the author considered the OS details and various software components (i.e., editor, compiler info, middleware apps, etc.) to present the ontological description of the grid resources. Similarly, in [138], researchers have thoroughly discussed different apps and APIs, which are involved in their proposed robust, secure and high-performance network operating system.
- **Networking related findings**. The network-related attributes are impressive. Mainly, for fog and edge computing research works, with the main concern about network bandwidth management. As the variety in fog/edge and IoT devices is extensive, many researchers are focused on identifying the standards and protocols for these paradigms. We also found that in some papers, such as [87], [91], [109], they discuss various types of networking devices (e.g., switches, routers, set-top boxes) to present their system architecture. However, most interestingly, we have observed that a few cloud-based research works are also interested in network bandwidth to manage cloud resources ([38], [43], [69], etc.) and discovering resources [71] or services[72].
- **Security related findings**. The most relevant security attributes addressed are authenticity, privacy, trustiness, and confidentiality. Also, concerning the security category, most research works are either concerned in several security aspects, or none. It is also worth noticing that the DOS attack is only considered as an ultimate security challenge in one research paper from the fog computing paradigm [91]. In any case, from this literature survey, Security has been observed as one of the major considered set of attributes.
- **Data related findings**. Data are one of the key components of any computing paradigm, but, surprisingly, we found that this attribute is not widely considered in our reviewed literature. Only in the IoT-related research works has been partially considered, addressing concepts such as data source, size, and content type information. In the IoT category, characterization and type identification of various IoT elements (sensors, actuators, etc.) is the most common and simplified objective. We also observed in [80], [93], to address the connected objects and the cloud-IoT paradigm, researchers have focused on identifying the type of sensors, actuators, and RFID in their systems. Indeed, these two papers focus on integrated fog/edge and cloud platforms, so this fact is quite natural. Besides that, we also found one exception [84] in cloud paradigm, where researchers have given their concern to identify the I/O type to present the cloud resource auctioning mechanism.
- **Cost related findings**. It is undoubtedly one of the least considered finding category. It is being considered mainly in cloud-based research works. Indeed, it is evident that cost is a crucial factor that should be considered before presenting some new proposal for making the cloud much more efficient. Exceptionally, we also found some of the non-cloud-based research works [90], [91], [94], [109], [112], [121], [134], where researchers also gave their concern to identify the cost-related and service provisioning features.

- **Features related findings**. The features and behaviours category includes a variable set of specific attributes. We observe that the cloud computing paradigm has some unique features, such as fault-tolerance, centralization, on-demand resource allocation, etc.; but no concern for transparency and openness, nor geographical-related issues. In other computing platforms (i.e., fog and edge, IoT, grid, and WSN) resources are geographically distributed, and near to the edge of the network, so they provide real-time services, and also these systems are very much aware of their working location. We also found that virtualization is playing a crucial role in cloud and fog computing paradigms.

Finally, these findings helped us to understand the characteristics and features of resources of different related computing paradigm and technologies. Undoubtedly, all of these pieces of information would help us to identify the characteristics and features of F2C resources.

3.2.2 In service-task aspects

Discovering characteristics of system offered services and identifying the requirements of tasks execution helps to effectively manage the system resources for any computing paradigm. Therefore, we continue our further investigation to identify the characteristics of services corresponding tasks requirements in various computing platforms (e.g. cloud, fog and edge, etc.) and technologies (e.g., IoT). Notably, we found a couple of cloud-based research works, where before presenting the proper resource management strategy, the researchers give their utmost concern for identifying the characteristics of the system resources, as well as the service-task features which are involved to their considering system. For example, in [43], the authors presented a resource management technique for managing the resources in the cloud platform. In that paper, they measured the performance of their proposed strategy and identified that the performance of their proposed resource management technique could be varied according to the specifications of participating system resources as well as the service-task demands. Also, in [38], the authors did a review of the literature to find out the primary vital aspects potentially involved in providing a QoS-aware automatic resource management strategy in the cloud. In the review, they found that by identifying the proper classification of system resources, tasks requirements and many other aspects, the appropriate QoS-aware automatic resource management strategy could be built.

In [44], the authors proposed a hybrid approach for a task scheduling model in the heterogeneously distributed computing paradigm. Taking into account the diverse characteristics of their system resources and tasks, the authors introduced their strategical model and validated it in a simulator tool. Not only in the case of fog and cloud paradigm we found researchers considering the resource and service-task characteristics to define the resource management technique, but also we have found lots of IoT-based research works, where researchers have given their concern, before proposing the adequate resource management strategy. For example, in [45] before presenting the proper resource management mechanism in the IoT-based smart city scenario, they have given their focus to identifying the characteristics of the system resources as well as the services. Also, we found in a couple of research works (i.e., [140], [141]), the fact that before defining the cost model for the corresponding computing paradigm, the researchers considered the characteristics of the system resources and identified service-task demands.

So, following all of these works, it is quite clear that resource and service-task characterization is essential and relevant, before designing an adequate resource management strategy in the F2C-enabled smart environment scenario. Eventually, investigating these works would definitely help us to make the taxonomic model for F2C resources and services-tasks.

3.3 Monitoring and Secure data distribution

Several issues seem necessarily needed to be addressed for building up a proper resource management strategy in any large-scale smart system (i.e., smart city). By reviewing several related works, in earlier, we found that the primary step for making a proper resource management strategy is to determine the characteristics of participating system resources and also the service execution requirements. Notably, in many works (e.g. [142]–[144]), we found that to build a more sophisticated and more efficient resource management mechanism; researchers have proposed some effective resource monitoring mechanism for continuously monitor the available system resources and organize the resource statistical information accurately over the system.

Importantly, in different computing platforms (e.g., cloud, fog and edge, etc.), we found many researchers have given their profound concerns for designing an effective monitoring mechanism for their system resources. For example, in [78], the authors presented that, how the cloud resource description and continuously monitoring the system resources, help to design better resource utilization and load-balancing in the Cloud-NFV platform. Also, in [144], the researchers have proposed an architectural model for monitoring their system resources in fog and mobile cloud platforms. In that paper, the authors said that efficient system resources monitoring is necessary. However, in that work, the researchers also mentioned that an appropriate distributed mechanism for storing and collecting these monitored resources information is a necessary demand for designing an effective resource management mechanism in their considering system. Similarly, the authors in [144] said that effective monitoring the system resources and distributing the collected resource statistical information could help their system to be more agile and helps to offer better latency-sensitive services. Therefore, not only the resource monitoring mechanism but also the resource information collection and distribution are the significant issues that also need to be addressed. Thus, realizing this demand for a better data collection mechanism in the distributed computing system, authors in [145] present an extensive study for efficiently collecting the data in the VANET platform, which could potentially help to design a useful data collection and storing mechanism in any large-scale distributed system. After reviewing these works, it is undoubtedly clear that building a proper mechanism to monitor the system resources is a necessary demand for making the effective resource management mechanism. Additionally, in any large-scale smart system (i.e., smart city), a vast number of small IoT devices are participating. They are continuously capturing various environmental events and producing a massive amount of sensing-data. Notably, along with the monitoring information, this data is also considered as one of the ingredients for various services (e.g., e-health, traffic monitoring, etc.). Typically, both of these data are generating a tremendous amount of data flow over the system. So, building a proper and secure data storing and managing mechanism is necessary before setting up an adequate resource management technique in any large-scale smart computing domain (e.g., smart city).

Importantly, we found some contributions are dealing with solutions for data management in various computing platforms (i.e., cloud and fog). In [146], [147], authors focus on an efficient data management in a fog computing ecosystem. More concretely, in [146] authors show their concerns to find out the best possible way to appropriately replicate the data over the distributed database system without issuing inconsistency issues. On the other hand, authors in [147] propose to design a suitable data-model aimed at ensuring efficient data distribution to the overall distributed computing framework. However, none of these two contributions considered security functionalities and provisioning over the data management. Another contribution [148] focuses on building an appropriate framework for managing the huge amount of data in the federated fog/edge and cloud platform. However, the work is not considering any security implementation over the data management in the federated fog/cloud. In [149], authors design a framework to bring the full cloud facilities near to the edge minimizing the data-query response time, which helps to optimize the performance of latency-sensitive situation-awareness services. However, the proposed architecture relies on computational resources from volunteering contributors what can cause high-security risks. In [150], a holistic vision-centric approach, proposing an architectural vision for managing the data is introduced, including novel hypothetical leveraged data-centric model approaches for managing the system data over the distributed network in a trustworthy fashion. Nonetheless, authors only propose an architecture with no details about security deployment and trust establishment between layers such as IoT, fog and cloud. Other papers, put special attention to highlight potential concerns to design access control policies [151] aimed at ensuring the security and privacy needs [152] for specific distributed database frameworks. However, in these proposals, access control and security provisioning are handled by a centralized component, becoming a single point of failure that can compromise the whole system. Therefore, after revisiting current works, we may conclude that an architectural solution for a secure and distributed management and access of the data in combined fog/cloud systems is a must and which is yet demanding specific efforts from the scientific community.

3.4 Resource allocation mechanism

For any computing paradigm, in the resource management mechanism, one of the most crucial steps is selecting the proper resources for allocating them to execute some tasks and provide some services. Typically, for a large-scale distributed and smart system (e.g. smart city), the heterogeneity in system resources and services creates a massive challenge for the resource selection and allocation operations. Notably, proper and quick selection and allocation of the system resources, are essential demands for effectively offer the latency-sensitive services. Also, proper selection and effective allocation of system resources, enhance the intelligence level of the overall system. Considering all these facts, many researchers have given their concerns for designing and building some advanced and sophisticated resource allocation techniques in various computing platforms (e.g. cloud, fog and edge, grid, etc.) [153]–[156]. Following various works, we realized that all of the resource allocation procedure could be classified into various aspects (e.g., objective aspects, awareness and enhancement aspects, and resource allocation adaption aspects, etc.).

In our literature study, we found that for conforming some objectives (i.e., energy optimization, resource consumption optimization, cost optimization, service-latency optimization, etc.) many researchers have done their works for designing some effective resource allocation mechanism in various computing platforms (i.e., cloud, fog and edge, grid, mobile computing etc.) [16], [34], [81], [89], [140], [154], [157]–[160]. Also, we found several works in different computing paradigms, where researchers have proposed the resource allocation mechanisms by considering various awareness and enhancement related aspects (i.e., SLA, QoS, context and mobility aware, and security-privacy enhancements, etc.) [6], [34], [161]–[164]. Apart from these, we found that in some of the works, researchers have given more concerns to identify the resource allocation adaption processes in various computing paradigms. Typically these processes can be further classified into main two approaches: reactive and predictive [165]. According to these researchers, in the case of reactive resource allocation approaches, the current state of resources has to be measured before allocating them. Whereas in the case of predictive approaches, the system is able to priorly forecast the computational load and the overall performance of system resources before allocating them [166]. In a predictive approach, the system is able to measure the performance and computational load of system resources priorly; so this approach explicitly helps for designing a better and intelligent resource management mechanism [167]. Besides these two approaches, we found the hybrid approach [168] for allocating the system resources.

Significantly, after performing the investigation on several works, we realized that most of the researchers had proposed different techniques for choosing the best-fitted resource(s) for successfully executing some task(s). Most importantly, the selection procedure for the best-fitted resource(s), can eventually be done based on some ranking or score of the available system resources. Interestingly, the calculation of the resource's rank depends on various aspects (i.e., resource demand matching, cost information, etc.) [52], [169]–[175]. Typically, in a large-scale smart system (e.g., smart city), the diversity of system resources poses a massive challenge to rank them efficiently. Considering this issue, many researchers have done their research works for making the multi-attribute based resource ranking mechanism [176]–[179]. However, for fulfilling their objectives, they have adopted some well-known methodology, i.e., TOPSIS [176], [180]–[182]. Apart from all of these works, we have found some other research works in different computing paradigm, where researchers have proposed their resource allocation mechanisms based on the auction-based resource selection procedures [84], [183], [184]. Importantly, we have found some works where researchers considered the game theory approach for allocating their system resources [159], [185].

3.5 Resource usage forecasting and performance prediction

For improving the intelligence level of any computing platform, it is essential to build a predictive approach based resource management mechanism for forecasting the resource usage and performance in advance accurately. Significantly, to fulfill this purpose, a vast number of research works have been already done in the cloud computing domain (e.g., [160], [186]–[189]) aimed at improving the resource management strategy. Also, in most of the cases, researchers provide a more specific solution to improve their prediction model for their proposed systems. For example, in [186], researchers use an existing ML approach to build some prediction model for improving the forecasting of resource usage in the cloud computing platform. Likewise, in other computing domains (i.e., smart grids, mobile ad-hoc networks) [190], [191], we also found some works, where researchers show their concerns to forecast the resource usage to reduce the workload for their system resources. Besides these, we also found some research works in the IoT and edge/fog computing domains (e.g., [192], [193]), where researchers put the focus on estimating the resource usage for upcoming tasks to reduce the load of their system resources. Even, we found that a significant amount of research works (e.g., [162], [194]–[197]) have been done in various other computing platforms, where researchers proposed some ML techniques for improving the resource provisioning and management.

After reviewing these works, we found one common thing, mentioned by them all; that is data. Most importantly, without the data, it is impossible to train any machine learning model. Even the selection of wrong data can easily reduce the accuracy-level of the trained ML model. So, for that reason, it is necessary to process the data and clean it. Significantly, data processing in large-scale smart system (e.g., smart city), can be done in a logically centralized location or distributed locations. Indeed, taking advantages of these two data processing mechanisms, ML techniques can also be performed in two ways: in a logically centralized location and distributed locations.

Furthermore, for edge or fog computing the prediction has to be done near to the end-users. For that reason, the data have to reside near to the processing node. Significantly, building a large-scale distributed and intelligent computing system inherently demands the effective sharing of smartness over the system [198]. Therefore, in the large-scale distributed system, it is needed to engage all computing resources in the learning and prediction processes [199]. With the help of global coordination and also by the processing of their own data, all computing resources can build their local prediction model for forecasting resource usage and performance (task execution-time). Following this way, resources are not only capable of taking the quick decision, but also they explicitly help to protect the data privacy policy [200]. This technique has been adopted in some of the research works for building the advance resource management mechanism [199], [201] in various individual computing platforms (such as edge or IoT). Nonetheless, performing this distributed learning and prediction technique is a very complex work for any large-scale smart system [201]. Notably, the heterogeneous nature of participating devices poses a massive challenge to perform the technique successfully. However, we have not found any prior work in any combined and coordinated computing platform (as in the F2C-based smart computing platform) where this technique has been adopted to effectively manage all the diverse system resources and build a proper resource management schema.

Chapter 4: Resource Management schema

The primary purpose of this chapter is to highlight our proposed resource management architectural schema. A thorough description of various architectural components and their functionalities would help to perceive the general resource handing operation in the F2C domain.

4.1 Architectural and Functional description

The Fog-to-Cloud (F2C) computing framework is emerging to both provide higher functional efficiency for latency-sensitive services and also help the large-scale smart computing system (e.g., smart city) to be more intelligent. In such a scenario, a vast number of heterogeneous resources, including computing devices and IoT sensors, are required to be worked in coordination for providing the best facilities. Thus, appropriately managing the set of available system resources is one of the most critical and challenging tasks in this computing framework. Also, it must be considered that F2C-enabled large-sized smart system offers different types of services (e.g., e-Health facilities, traffic monitoring, fire emergency, etc.). Most interestingly, all of these services are offered by executing several tasks. Notably, every task has different resource requirements, which have to be satisfied for successfully delivering the services. So, considering both the resource and service aspects, it is simple to realize the massiveness of heterogeneity in any large-scale smart system. Hence, the nature of diversification poses a considerable challenge to effectively manage and utilize the system resources in any F2C-enabled large-sized smart system. However, effective management and utilization policy of system resources can revive to build a more advanced and smarter system. Also, proper utilization of system resources can prolong the resource's life-span and helps to reduce their unnecessary consumption [192]. Properly utilize the system resources essentially needs adequate resource allocation and scheduling policies [202]. Notably, in any large-scale and distributed smart computing system, on-demand resource allocation is a critical process and one of the most crucial steps for designing an advanced and intelligent resource management mechanism [203]. Especially, proper allocation of available system resources based on their current and future demand for executing some computational tasks is a challenging job. In order to address this issue, many researchers proposed different techniques [204]–[206], which can be classified into two approaches: reactive and predictive [165]. According to these researchers, in the case of reactive resource allocation approaches, the current state of resources has to be measured before allocating them. Whereas in the case of predictive approaches, the system can priorly forecast the computational load and the overall performance of system resources before allocating them [166]. In a predictive approach, the system can measure the performance and computational load of system resources priorly; so this approach explicitly helps for designing a better and intelligent resource management mechanism [167]. Thus, to perform some sort of forecasting operation, it is necessary to design a framework of machine-learning (ML) based resource management mechanism. Eventually, that would help the overall system to be more intelligent and advanced. Thus, considering all of these facts, we realized the necessity to design an architectural framework for managing the resources in the F2C-enabled computing system. However, before designing the architectural framework, it is essential to ascertain all the functional requirements which have to be attained by our proposing architectural framework for managing the F2C resources. Therefore, in the next paragraph, we have explained all the prerequisite for developing our proposing architectural framework.

In this thesis work, our main objective is to propose and design an advanced and sophisticated architectural framework for adequately managing the F2C resources. Notably, for keeping our promises and fulfilling our objectives following functionalities have to be attained by our proposing framework:

- *Capabilities of classifying and characterizing the system resources and service-task.*
- *Ability to figure-out the resource description models for gathering the full knowledge of F2C-enabled system.*

- *Must have useful information collection facilities for making the accurate information pool. Notably, accurate information pool explicitly helps in the resource selection and allocation procedures. Hence, in the F2C it is necessary to securely distribute the information pool over the network to offer better performance in latency-sensitive services.*
- *Need to have the skills for intelligently predict, select and allocate the best-fitted resources for executing some tasks and provide the services. However, smart selection and allocation ensure the enhancement of QoS factors and reduction of unnecessary resource consumption.*
- *Competency to distribute the smartness over the hierarchical and coordinated F2C computing paradigm.*

Considering all the aforementioned prerequisites, in this dissertation, primarily, we gave our intense focus on designing the overall resource management architectural framework. Also given our focus on identifying the various architectural modules and sub-modules, which can collaboratively help to build a proper mechanism for managing the resources in the F2C paradigm. Later on, in this thesis work, we gave our concentration for solving various research challenges which are incorporated in various modules of our proposing architectural framework. Notably, by addressing those research challenges, we can able to develop a more precise, advanced and sophisticated resource management mechanism in the F2C-enabled system.

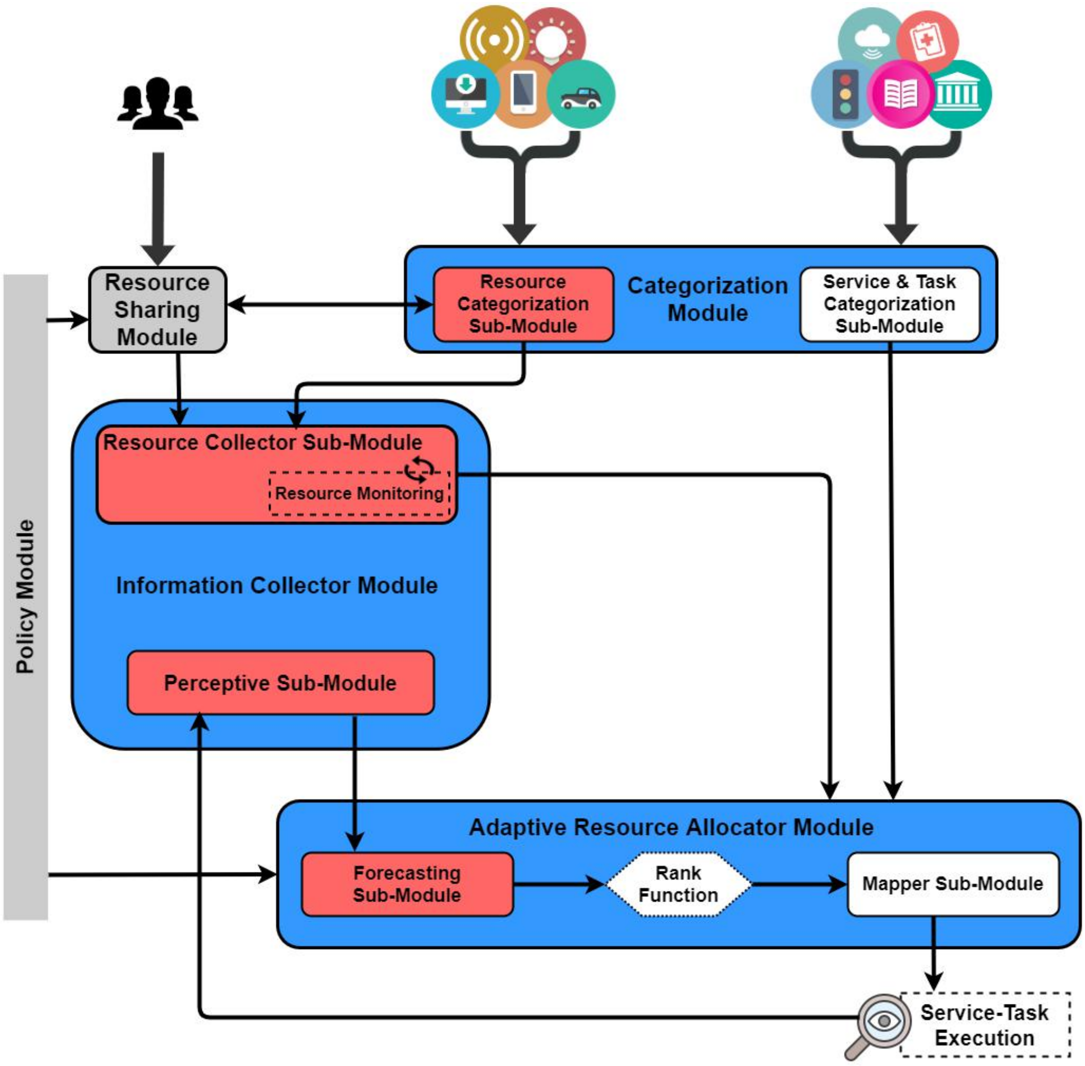


Figure 4.1: Proposed resource management framework in Fog-to-Cloud (F2C)

4.2 Resource Management Modules

In Fig. 4.1, we depicted our proposing structure of the resources management strategy for the F2C-enabled computing system. According to the diagram, four other modules along with the *Policy Module*, are playing the pivotal roles to utilize and manage the system resources appropriately and effectively offer the services among the subscribers of any F2C-enabled smart system. These four different modules are: 1. *Categorization Module*, 2. *Resource Sharing Module*, 3. *Information Collector Module*, and 4. *Adaptive Resource Allocator Module*. In the next of this section, we explain the functionalities of these four modules, in the context of smart city scenario.

4.2.1 Categorization Module

In earlier mentioned, the most significant challenge for managing resources in any F2C-enabled smart city scenario is diversity. Variety of the whole set of resources from the bottom edge up to the cloud, is one of the crucial characteristics of any F2C-enabled system. Significantly, many different services are offered among the smart city's consumer. Undoubtedly, any F2C-enabled smart city becomes much more smarter, by successfully and efficiently delivering all of these services among its citizens. Importantly, each service has different characteristics (i.e., free or chargeable services), contexts (i.e., governmental, health, educational, etc.) and even different requirements (i.e., resource requirements for executing the tasks). Thus it creates an enormous diversity over the whole system. Therefore, this diversification poses some notable difficulties for building up the proper resource utilization strategy in the F2C paradigm. Even it makes a significant challenge for an appropriate matching between the service requirements and available system resources. So, before forming the useful resources management strategy, it is pretty relevant to identify the classification and categorization of the system resources and services involved in the F2C paradigm. Thus considering this fact, in our resource management framework, we sketched and built a module named *Categorization Module*. This module is responsible for identifying the characteristics of system resources and services. This module is consists of two sub-modules.

- *Resource Categorization Sub-Module* - This sub-module is in charge of identifying the characteristics of participating system resources, and also helps to classify the system resources based on their identified characteristics information. Notably, the outcome result from this sub-module is helping to create the resource information pool for the whole system. Thus it explicitly helps for building an effective resource handling mechanism.
- *Service & Task Categorization Sub-Module* - Whereas this sub-module is responsible for analyzing the characteristics of the system's services. Following any large-scale smart system scenario (e.g., smart city), it can be easily realizable that a vast number of different services are offered among the system subscribers. Notably, the characteristics of each service are different. For offering these services, it is essential to perform and accomplish some tasks. Importantly, each of the tasks has different requirements. Some of them required to be executed for a long time. Even some of the tasks have to be performed in a short time. Interestingly, some of the tasks demanding substantial computational facilities, where some of them can be executed in the resource-constrained computational devices. Therefore, the *Service & Task Categorization Sub-Module* performing in our proposed resource management architecture for identifying and gathering all this information.

4.2.2 Resource Sharing Module

In any F2C-enabled smart computing system, seeks to provide resources to different latency-sensitive applications. It is a challenging issue for any F2C-service providers to satisfy all the resource requests effectively. Even, optimally provisioning the available resources is also a challenging job for any F2C-enabled system. Resource sharing among different F2C-service providers can address the limitation, as mentioned above. Importantly, by following some policies, individual F2C-service providers can rent their available and excrescence resources to the other service providers or F2C-resource requester.

Following any F2C-enabled smart city scenario, it can easily recognize that besides the cloud-service providers, many individuals and small organizations are participating in the system with their computing and networking devices. They can contribute or 'rent' their devices for executing some requested tasks to serve various smart services effectively. Also, many of the participants are joining in the F2C system for only accessing some smart services. Therefore, considering these facts, we realized that all the devices might participate as either 'Consumer', 'Contributor', or 'Dichotomous'. Interestingly, before contributing their resource components to the F2C system, it is necessary to define the proper resource-sharing model. Based on the owner preference, system policies, and availability of their resource-components, a sharing model can be determined. Notably, we identified all the available resources in the F2C system could be shared by adopting any of the following two-ways - a) *Containerization*, or b) *Conventional Virtualization* [207]–[209].

Significantly, following the F2C-enabled system, we realized that resource-sharing is playing a crucial role in successfully utilizing the available system resources and efficiently offer the various smart services among its subscriber. Therefore, we proposed a functional module named *Resource Sharing Module* in our resource management framework. By considering the system policies, user preferences and many other aspects, this module is responsible for choosing the appropriate technique for sharing the available resources. Once the module created the shared copy of actual resources, it passes the information to the *Resource Collector Sub-Module* for creating the resource information pool. Thus it helps the overall resource management process to handle and utilize the available system resources adequately.

4.2.3 Information Collector Module

The overall information about available system resources and their current status helps to manage them properly. Therefore, the collection of the resource-information is one of the mandatory steps before selecting and allocating them for executing some tasks to achieve some services. Realizing this demand, helped us to design and build one particular module in our proposed resource management framework. This module is known as *Information Collector Module*. The primary purpose of this module is to continuously monitor the available system resources, as well as observing the Service-Task execution procedure. Notably, monitoring of the system resources has some intense impact over the different courses of resource management mechanism in any distributed system. Efficient monitoring of the system resources can help to provision and schedule the system resource appropriately. Also, an effective monitoring mechanism can improve the overall scalability (i.e., load-balancing, failover recovery, etc.) of the system, as well as it explicitly helps to design a useful cost or billing techniques [143].

Significantly, following overall F2C-enabled system architecture, we identified that this module is a composite of two different sub-modules: 1) *Resource Collector Sub-Module*, and 2) *Perceptive Sub-Module*. Typically both of the sub-modules are in charge of monitoring the system resources. However, these two sub-modules are following two different types of resource monitoring mechanism, which are essentially needed to be performed for designing an adequate resource management mechanism. The initial monitoring can be done for creating the resource information pool. Notably, resources are consumed for executing some tasks. So, it is also essential to keep track of those resource usage-related information for designing an adaptive resource management mechanism in F2C paradigm. Thus the second type of monitoring can be performed for collecting the resource-usage information. Typically, both of the monitoring can be done by following any of the three different strategies - a) *Push*, b) *Pull*, and c) *Hybrid* [143]. Basically, in the *Push* strategy, the resource monitoring agent runs in every computing node and pushing the resource information to their managerial computing nodes, in a specific time interval. Whereas, in *Pull* strategy, the resource information is obtained dynamically whenever the request has been received. This strategy is mostly a triggered based approach. In this strategy, computing nodes are sending their information after requesting the information from their managerial node. Besides these two strategies, also we have found another strategy which is a combinational form of the two aforementioned strategies. This combinational strategy is known as *Hybrid* strategy.

Nonetheless, considering all of these aspects in our proposed resource management framework, we decided to adopt the *Push* strategy for *Resource Collector Sub-Module*. However, enriched by the outcomes of the *Resource Sharing Module* and *Resource-Categorization Sub-Module*, this sub-module is collaboratively making the resource-information pool. Most importantly, in the F2C platform, this sub-module is in charge of generating the overall landscape of the system. Whereas, the *Perceptive Sub-Module* is in charge of monitoring the state of the system resources. Notably, this sub-module is mainly responsible for keep tracking and collecting the resources usage (e.g. RAM usage, CPU usage, Disk I/O usage, etc.) and performance (e.g. task execution time) related information for the execution of the tasks. Whenever the task execution is proceeding, this sub-module is being activated and starting to perform. Thus, this sub-module is following the *Pull* strategy for monitoring the state of the resources and capturing the resource usage information. Finally, after collecting this information, this sub-module sends it to the *Forecasting Sub-Module*. Hence, after receiving this information, it is getting much more accessible for the *Forecasting Sub-Module* to accurately predict the working load, usage and performance of available system resources. Thus the accurate prediction helps to utilize the system resources properly. Significantly, one of the common feature of *Resource Collector Sub-Module* and *Perceptive Sub-Module* is that both of the sub-modules are generating an enormous volume of data to the F2C-enabled system.

4.2.4 Adaptive Resource Allocator Module

In our proposed resource management framework, the *Adaptive Resource Allocator Module* is the core module for allocating the appropriate resources for executing some tasks. It is also responsible for effectively scheduling the tasks among the available system resources to balance their workload efficiently. According to Fig. 4.1, initially, this module is getting the current status of the available system resources from the *Resource Collector Sub-Module*. Interestingly, the *Service & Task Categorization Sub-Module* informed this module with the information of requested task requirements. Once the module received all the information, then using the *Rank Function* it starts to calculate the ranking for the available system resources. The rank can be calculated considering the global cost or business model [12], [161], [210], attributes of available system resources [211]–[213], and following the resource requirements [173], [214], [215]. Significantly, in the hierarchical, combined and coordinated F2C-enabled computing system, one of the biggest challenges is to design the global cost or business model. In the coordinated F2C platform, the global cost or business model can be defined by considering different aspects (i.e., task execution cost, usability cost, etc.).

After calculating the rank of available system resources, based on the system policies, the *Mapper Sub-Module* select, map and allocate the proper resource(s) for executing the requested task(s). During the task(s) execution process, the *Perceptive Sub-Module* is monitoring the allocated resource(s) for collecting the resource usage (e.g. RAM usage, CPU usage, Power usage, Network usage, Disk I/O, etc.) and performance (e.g., task execution time) related information. Then, the *Perceptive Sub-Module* send this information to the *Forecasting Sub-Module*. With the help of this information, the *Forecasting Sub-Module* can able to predict future resource usage and performance. Thus, this predicted information can help the *Rank Function* for re-evaluating the rank for system resources. Hence, by following this adaptive mechanism, it is possible to achieve better resource utilization and management mechanism in the F2C paradigm.

Moreover, in this thesis work, our main intention is to design the preliminary version of the adaptive resource management framework in F2C computing paradigm. For that purpose, we have done this work. Importantly, in order to validate this framework and solve some of the relevant related issues, we have given our intense focuses to continue our investigation further. Typically, in our proposed framework, we have introduced some of the functional modules and their sub-modules. Combinedly, all of these modules and sub-modules, are helping to design adequate, advanced and sophisticated resource management mechanism. So, we realized the demand for continuing the research on those modules/sub-modules for solving various existing issues to design an appropriate resource management schema in F2C paradigm. In Fig. 4.1, we marked some of the modules by blue colour and some of the corresponding sub-modules by the red colour. Also, some of the sub-modules we marked by the white colour. Whereas, by the grey colour we also mark some of the modules. For continuing the further investigation, we give our highest priority on the blue coloured modules and corresponding red coloured sub-modules. Whereas, after doing some initial investigations, we adopted some of the existing concepts to deploy the white coloured sub-modules for designing our proposed resource management framework. Importantly, investigation on the grey coloured modules is the out of the scope of this work. Therefore, we have considered some of the existing solutions for building those grey coloured modules to design the adaptive resource management mechanism in F2C paradigm. In the following chapters of this dissertation, we are going to discuss all the related research challenges for developing the blue coloured modules and corresponding red coloured sub-modules. There, we also propose different solutions for successfully addressing the existing research challenges for developing the aforementioned modules and sub-modules.

Chapter 5: Categorization Module

Focusing on the F2C-enabled smart city scenario, in the previous chapter thoroughly described our proposed resource management framework. We have seen that different architectural modules and sub-modules are involved in our proposed framework for efficiently managing the system resources. The Categorization Module is one among them. According to our proposed resource management framework, this module is consist of two sub-modules. One is Resource Categorization, and another one is Service & Task Categorization Sub-Module. These two sub-modules are in charge of identifying the characteristics of participating system resources, helping to create the resource information pool, and responsible for analyzing the characteristics of the system's services as well as identifying corresponding task-execution requirements. However, we identified that by accomplishing these jobs, the Categorization Module is solving a few critical research challenges in the F2C continuum. Therefore, initially in this chapter, we point out all the research challenges, which are incorporated with the Categorization Module. Later on, we also thoroughly described our proposals to successfully addressed those research challenges. Finally, we illustrate how our proposals have been successfully integrated into the EU H2020 funded mF2C[1] project for solving different research challenges.

5.1 Architectural and Functional description of Categorization Module

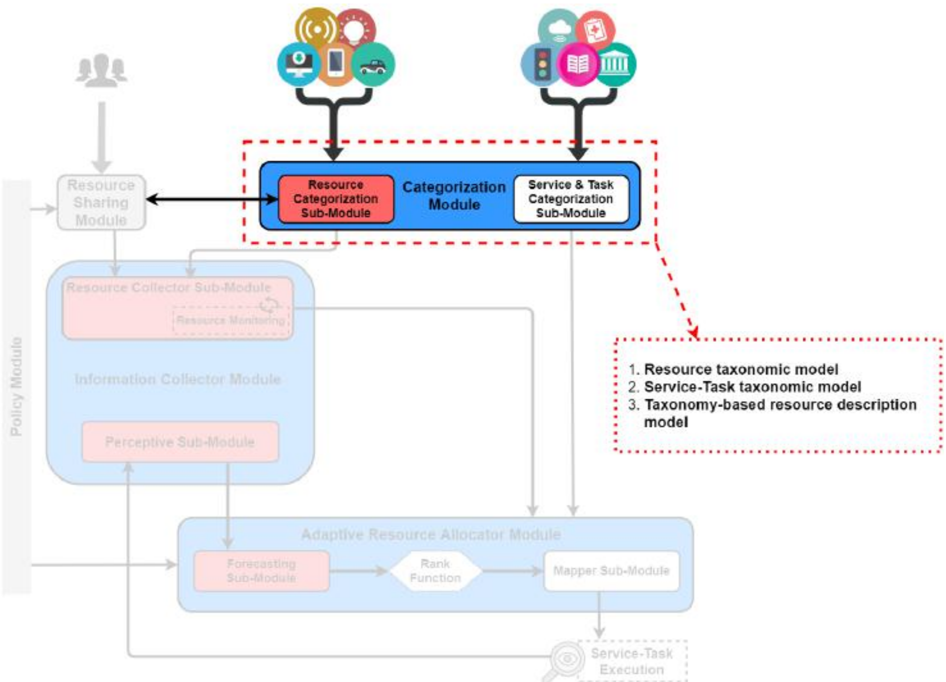


Figure 5.1: Research challenges addressing by the Categorization Module

The enormous heterogeneity for the whole set of resources from the edge up to the cloud makes resources management in Fog-to-Cloud a challenging effort. From a broad perspective, it is pretty evident that the closer to the top (i.e., cloud) the larger the capacities are. Thus, we may undoubtedly assess that computation, processing and storage capabilities are higher in the cloud than in the fog, and higher in the fog than in the edge. Interestingly, in the F2C envisioned scenario, this assessment is even more elaborated, leveraging the different layers foreseen for fog. Indeed, in F2C different layers are identified to meet different characteristics of distinct devices. Even following

[1] https://www.mf2c-project.eu/

any large-scale F2C-enabled smart system (i.e., smart city), it can be easily recognized that it offers various smart services (e.g., e-Health, e-Governance, smart garbage collection, etc.) among its subscribers. Notably, all these services are offered by accomplishing some tasks. Thus, the accomplishment of the tasks can be done by fulfilling their requirements. Typically, every task has different requirements, and also each of the services is offered to enable various facilities in different purpose. Considering all the aforementioned facts, we proposed and designed the *Categorization Module* in our resource management framework. In our proposed architecture, this module is responsible for solving three different research challenges in both resource and service-task circumstances. In Fig. 5.1, we have presented three different research challenges which must have to be solved for making the advanced and effective resource management mechanism. These three different challenges are as follows: 1. Identifying the *Resource taxonomic model*, 2. Determining the *Service-Task taxonomic model*, and 3. Defining the *Taxonomy-based resource description model* in any F2C-enabled system. Next, in this section, we thoroughly describe how those challenges have been addressed before developing the adaptive and sophisticated resource management framework in F2C paradigm.

5.1.1 Resource taxonomic model

In the F2C paradigm, devices can participate in the F2C-enabled system as either 'Consumer', 'Contributor', or 'Dichotomous'. A device acts as a 'Consumer', gets into the F2C system to execute services, being a pure resources consumer. While acting as 'Contributor', a device offers its resources for running some applications to offer some services. Finally, some devices can act as 'Dichotomous'. These 'Dichotomous' devices not only accessing some services but also contributing with their resources to support services execution. Thus, according to the participation role, devices in the F2C system can be classified into the aforementioned three distinct types. However, following different service-based systems (e.g., smart city), we have found that for participating in any smart city, devices must have an entry point or software or application portal. Hence, following the F2C-enabled smart city scenario, we have identified that devices can participate in the system in two ways: either they have 'application' or 'software' installed on their device; or, they can join the system by connecting to another device which has the 'application' or 'software'. Notably, devices that are endowed with the F2C enabled 'software' or 'application', considered as the F2C resources. Also, other participating devices joining the system by connecting with F2C resources, are referred to as attached resource components of the corresponding F2C resource.

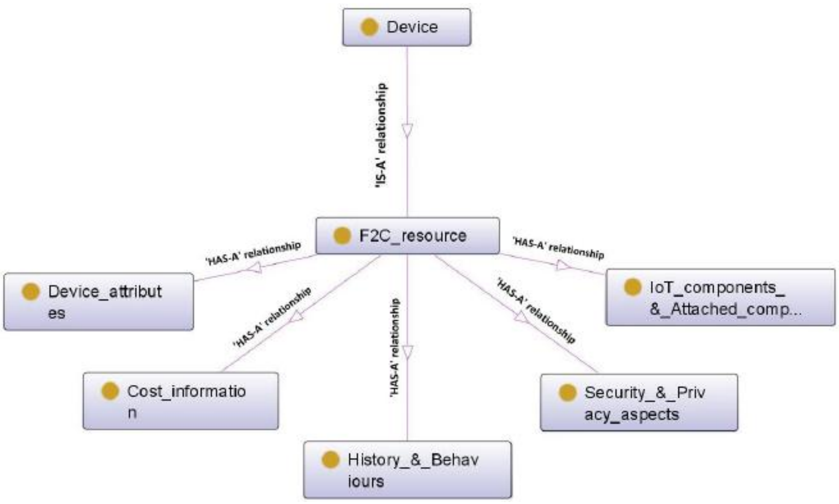


Figure 5.2: Taxonomic model for the F2C resources.

Following the F2C-enabled smart city scenario, we realized that before moving further for designing an effective resource management mechanism, it is essential to identify and discover the resource characteristics of the F2C resources. Thus, a taxonomic model of F2C resources can give detailed information about the characteristics of F2C resources. Therefore, a taxonomic model for F2C resources is a necessary demand. Considering this fact, we present a taxonomic model for F2C resources in Fig. 5.2. To that end, in order to present the ontology-based taxonomic model in F2C paradigm, we adopt the classification method proposed by Perez [216]. According to their proposal, modelling elements are divided into five basic modellings original language: classes, relations, functions, axioms, and instances. The ontology model O, is shown as:

$$O = \{C, R, F, A, I\} \quad (1)$$

C represents the class or concepts and can be further classified and subdivided into a kind of basic class Ci. R represents the collection of relations, mainly containing four basic types: part-of, kind-of, instance-of and attribute-of. F represents the collection of functions which can be formalized as:

$$F = C_1 \times C_2 \times C_3 \times \ldots \times C_{n-1} \rightarrow C_n \quad (2)$$

A represents the collection of axioms, and I represents the collection of instances. Based on the ontological model described above, we have analyzed the basic elements of parameters C (class) and R (relation), according to the attributes and expected behaviour for the whole set of resources in an F2C system. This analysis will help to constitute the resource description model for F2C-enabled system.

According to our proposed taxonomic model (in Fig. 5.2), each F2C resource associated with five (5) main attributes or characteristics. In Fig. 5.2, each of them represents as the individual class. The characteristics or attributes of F2C resources are: **Device attributes** (i.e., hardware, software, network specifications and device-type information), **Cost information** (i.e., chargeable device, non-chargeable device), **History and Behavioural information** (i.e., participating role, mobility, lifespan, reliability, information of the device location etc.), **Security and Privacy aspects** (i.e., device hardware security, network security and data security), and **IoT and Attached components** (i.e., sensors, actuators, RFID tags, attached resource components etc.). Notably, considering the aforementioned information, we are classifying all the F2C system resources. Interestingly, by collecting this information, it is possible to gather the full knowledge about the F2C system resource. Besides, adequate maintenance of this information for all F2C resources helps us identify the full system capacity. However, unfortunately, collecting this information is not an easy job for any F2C-enabled computing paradigm. Especially, it is very tough to automatically grab the characteristics related information from the extremely resource-constrained Edge-IoTs (i.e., sensors, actuators, RFID tags, etc.).

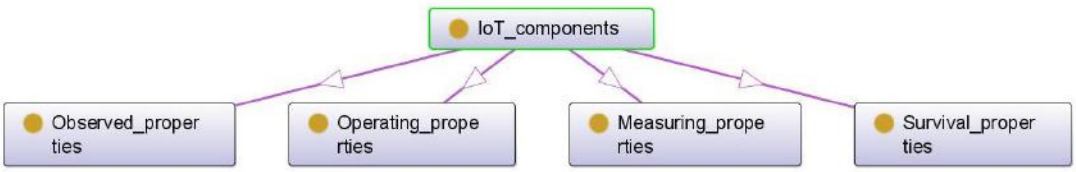


Figure 5.3: Property-based Edge-IoT classification.

Notably, in the F2C-enabled smart city, all the Edge-IoT components can participate following two ways: either by attaching with the F2C-enabled devices or as an integral part of the F2C-enabled devices. Most importantly, the IoTs are considering their associated F2C-enabled device as the gateway for connecting to the F2C-enabled computing platform. Interestingly, by deep reviewing the modern technology and smart system scenario, we have identified that the IoTs (more precisely, the sensors) are mostly responsible for providing a significant amount of data on the system. So, by considering our F2C-enabled smart city scenario, we have seen that primarily the IoTs are transmitting this raw data to their associated F2C-enabled device, for further analysis and processing. Importantly, these raw data contains all the characteristics information of the IoT (i.e., sensors, actuators), and as well as the captured sensing-data. After analyzing these raw-data, we have identified the significant properties of the Edge-IoTs. By considering all the potential issues of any F2C-enabled smart environment scenario and based on the determined properties (i.e., Observed, Operating, Measuring, and Survival properties), we are classifying all the Edge-IoTs for the F2C-enabled computing paradigm. This classification is presented in Fig. 5.3. Reviewing the F2C-enabled smart computing platform, we have identified that mainly the sensors are responsible for supplying a vast amount of captured sensing-data to the system. They are continuously capturing the data from various environmental events (i.e., temperature measurement, humidity measurement, etc.) and transmitting these sensing-data to their associated F2C-device for further processing. However, for further processing on these data, it is essential to identify the features of collecting data and the collector object (i.e., the sensor itself).

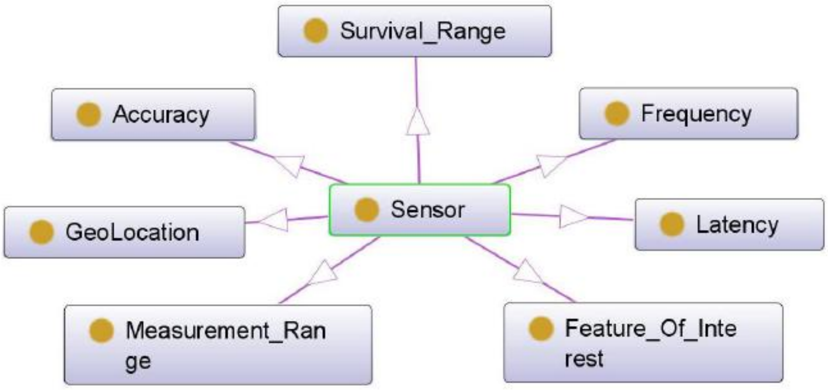


Figure 5.4: Sensor classification in the F2C-enabled smart system

For any IoT-based system, by analyzing the captured raw data, the Semantic Sensor Network (SSN) ontology provides the facilities for identifying and describing the sensors [67]. So, in our work, we adopted this methodology for classifying all the participating sensors for F2C-enabled smart system. Following the functionalities of the F2C-enabled smart system, considering the various related works (i.e., [217], [110], [218]), and focusing on the SSN ontology, we have identified that the sensors can be classified according to the following information: **Feature of Interest** (i.e., in an observation of the weight of a person, the feature of interest is the person, and the quality is weight), **Survival Range** (i.e., working lifespan), **Accuracy** (i.e., the maximum difference that will exist between the actual value), **Geo-Location** (i.e., location of the sensory device), **Latency** (i.e., delay for transmitting the captured data), **Frequency** (i.e., the rate for the amount of capturing and sending data), **Measurement Range** (i.e., the highest value that a sensor can able to measure). Meanwhile, in Fig. 5.4, we represented the sensor classification model for any F2C-enabled smart environment scenario. However, the characterization of the F2C resources not only helps us identify the attributes of the resources but also, collectively all this information provides us with the greater knowledge of the complete resources information of any F2C-enabled smart computing scenario. Moreover, having this kind of aggregated information helps us to define the proper resource management mechanism in the F2C-enabled system. Interestingly, following our proposed F2C resource management framework (Fig. 4.1), it can be easily realized that the process of characterization and classification of the F2C resources is performed by the *Resource Categorization Sub-Module*.

5.1.2 Service-Task taxonomic model

Another important job of the *Categorization Module* is to identify the characteristics of services and discover the task requirements. Significantly, identifying their characteristics and requirements can ease the job to search the appropriate resources for them. Notably, following any service-based computing system (e.g., smart city), the 'Service' and 'Task' are quite closely related to each other. In [219], the authors identify the meaning of 'Task', as performing some certain job(s) or function(s). Whereas, in [41], the authors define the 'Service' as a composite

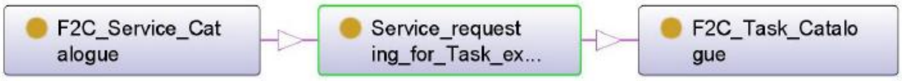
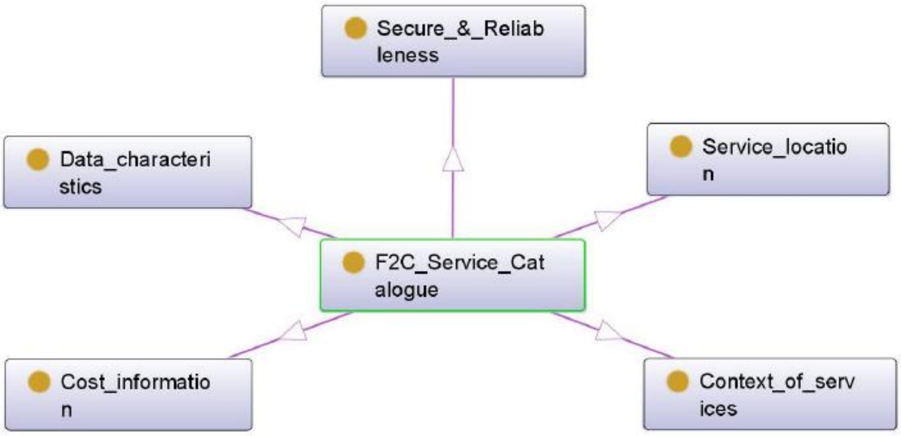
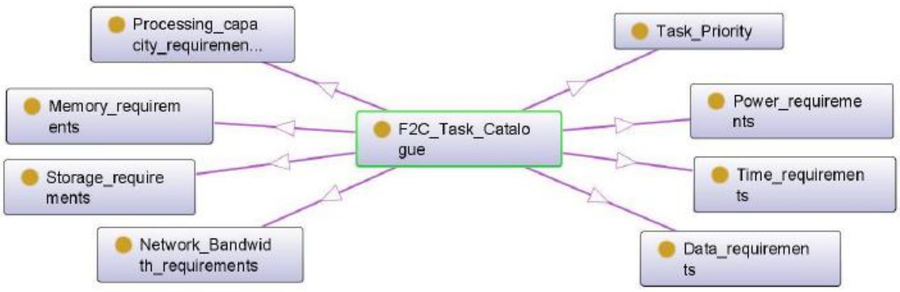


Figure 5.5: Relationship between Service and Task in the F2C paradigm

made up of small blocks of functionalities. According to them, any service-based system is offering the service(s) to its subscribers by executing some task(s). Therefore, following the F2C-enabled smart city scenario, we have identified that like other service-based computing paradigms, here also the 'Service' and 'Task' are related to each other.



(a) Classification of the F2C service



(b) Classification of the F2C task

Figure 5.6: Taxonomic model: (a) F2C Service, (b) F2C Task

Fig. 5.5 represents the relationship between the 'Service' and 'Task', in the F2C-enabled system. Most importantly, we also found that service characterization is the composite form of two steps: one is the service classification, and the other is the task classification. So, in our platform, we are considering this characterization process as the 'Service-Task Characterization'. Furthermore, as we have already identified that the diversification is a vital aspect of any F2C-enabled smart environment scenario, this aspect is also applicable to the service concept. So, considering all potential issues in the F2C-enabled smart city scenario and also following the various characteristics and attributes of the services initially, we classify all F2C services according to five different aspects. Following are the considering aspects for the service classification: **Context of services** (i.e., governmental, educational, transport, etc. related services), **Service location** (from where the services are offered, i.e., cloud or fog), **Secure and Reliableness** (i.e., based on the security preferences, services can be classified), **Data characteristics** (i.e., based on the amount of data processing requirement, services can also be classified), and **Cost information** (i.e., based on the service offering cost, services can be further classified into chargeable or non-chargeable services). In Fig. 5.6(a), we presented the classification model for the services; those are involved in any F2C-enabled computing paradigm.

In earlier, we have identified that like any other service-based computing paradigm; services are also offered by executing some task(s) on the F2C-enabled smart system. Also, we have identified that, in any F2C-enabled smart system, the service characterization is the composite form of two steps, where the task classification is also essential to define the 'Service-Task' taxonomic model in any F2C-enabled smart environment scenario. So, similarly, tasks can also be classified according to their **Execution requirements** (i.e., network bandwidth capacity, time requirements, processor requirements, storage requirements, memory requirements, etc.) and their **Priority** (i.e., high, medium, or low). In the below, we represented the classification model for the F2C-task(s) in Fig. 5.6(b).

So, in case of any F2C-enabled smart system, by considering these two taxonomic models, it can be possible to build the 'Service Model' in F2C-enabled smart system. Also, considering the proposed taxonomic models, helping to gather full knowledge and information about the services. Most importantly, by adequately maintaining all these information, we can easily achieve to estimate the resource requirements for the whole system, and explicitly that might also help to identify the proper way to manage the system resources for providing the better services. Importantly, to validate our proposed resource management schema, we have given our most profound concerns to solve all the F2C resource-related issues. Therefore, developing and implementing the service model is out of the scope of this work.

5.1.3 Taxonomy-based resource description for F2C

The process of classification and characterization of F2C resources, not only helps to identify their attributes; also by accomplishing this process, the *Resource Categorization Sub-Module* generates a massive amount of data into the system. Moreover, it is worthy of organizing those data properly before storing it. However, organizing those data in the hierarchical F2C-enabled smart system might generate two different resource description model (i.e., Fig. 5.7). The initial one is the *Generalized resource description model* - for a single F2C resource, and the second one is the *Aggregated resource description model* - a combined form of the multiple F2C resources.

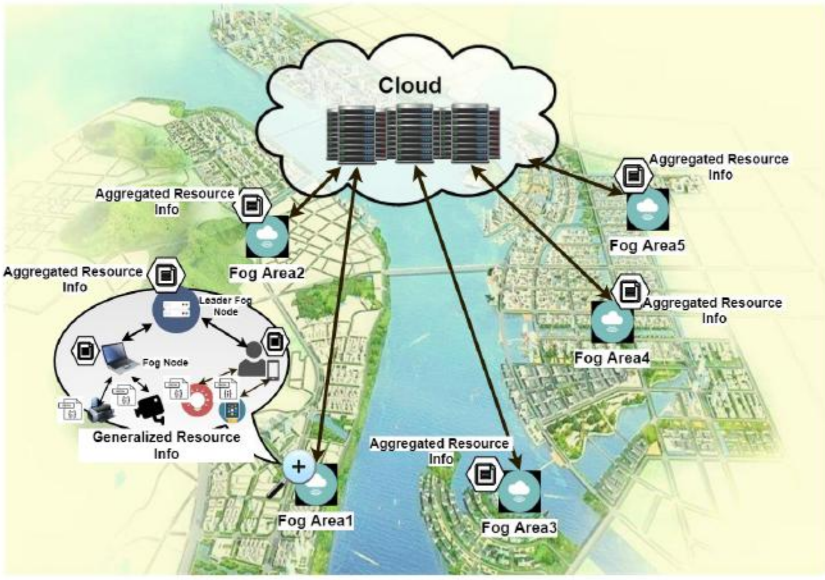


Figure 5.7: Resource information sharing from Fog to Cloud: generalized to aggregated resource info

5.1.3.1 Generalized resource description model:

Earlier, we explained that for joining into the F2C-enabled smart system, devices must have a specific entry point or 'software' or 'application'. By installing this specific software or application devices can participate in the F2C-enabled system. Interestingly, the devices have specific software or application are known as F2C resource and denoted as R_{F2C}. Other devices, those have not the 'software' or 'application', can also participate in the system by attaching to the F2C resources. Notably, considering the previously proposed F2C resource taxonomic model and all these facts, we have present the generalized resource description model for an F2C enabled-device, as follows:

```
R_F2C =
        <
          user_name; device_id;
          Device_attributes: <
               Hardware_components: <
                            Storage_information;
                            Main_memroy_information;
                            Processor_information;
                            Power_source_information;
                            GPU_&_FPGA_information
                            >;
               Software_components: <
                            Apps_&_APIs: <
                                     F2C_app: <
                                             cloud_resource_app;
                                             fog_resource_app
                                             >;
                                     Other_apps_&_APIs
                                     >;
                            Operating_system
                            >;
               Network_information: <
                            Bandwidth_information;
                            Networking_standards_information
                            >;
               Resource_type: <Physical_device;
                                   Virtual_device>
              >;
          IoT_components_&_Attached_components: <
               Sensors;
               Actuators;
               RFID_tags;
               Other_attached_device_components
               >;
          Security_&_Privacy_aspects: <
               Device_hardware_security;
               Network_security;
               Data_privacy
               >;
          Cost_information: <
               Chargeable_device; Non-Chargeable_device
               >;
          History_&_Behaviors: <
               Participation_role; Mobility; Life_span;
               Reliability;
               Information_of_the_device_Location;
               resource_sharing_information
               >
        >
```

To share the resource information efficiently with other F2C enabled resources, we adopt the JSON language to make a standard and formatted description file. In Fig. 5.8, we represent the resource description file of an F2C enabled laptop, based on the JSON language. The description file contains the detailed information about the hardware (i.e., total and current available storage, RAM information), software (i.e., OS information, F2C app information, etc.), IoT and attached components (sensors and other connected device information), history & behavioural (i.e., current location information, participation role etc.) information of the F2C enabled laptop.

```
{
  "user_name":"craax_user123",
  "device_id":11078934576,
  "Device_attributes": {
       "Hardware_components": {
   "Storage_information_(_in_MB_)": {
"Total":122880,
"Available":965890
                       },
   "Main_Memory_information_(_in_MB_)": {
"Total":32768,
"Available":13968
      },
   "Processor_information": {
"Processor_maker":"Intel Core i7-8550U CPU @ 1.80GHz",
"Available_percentage_of_processor":90.7
"Processor_architecture":" X86_64"
   },
.
.
},
        "Software_components": {
   "Operating_system":"Windows-10-10.0.16299-SP0",
   "Apps_&_APIs": {
"F2C_app":"fog_resource_app",
"Other_apps_&_APIs": {
"Adobe Acrobat Reader DC",
"AMD Software",
.
.
     }
   }
},
.
.
},
  "IoT_components_&_attached_components": {
.
.
},
.
.
}
```

Figure 5.8: The JSON-formatted resource description file for a F2C-enabled laptop : An example

5.1.3.2 Aggregated resource description model:

As shown in Fig. 5.7 several fog areas may be included in a smart city, each of them providing F2C services to the citizens. However, in each fog area, correct management of the whole set of resources is essential to make the overall system to be accurate and efficient. Unfortunately, since each fog area is built with distinct resources. Depending upon the amount of participating F2C resources, the overall capacity of processing, storage, power and networking techniques may differ for each fog area. Thus it is endowing each particular fog area with distinct characteristics and features. This scenario makes the management of all fog areas notably challenging, thus it creates a massive difficulties for building an efficient F2C system. To mitigate this problem, a clear description of the entire set of capacities and characteristics of each individual fog area is mandatory.

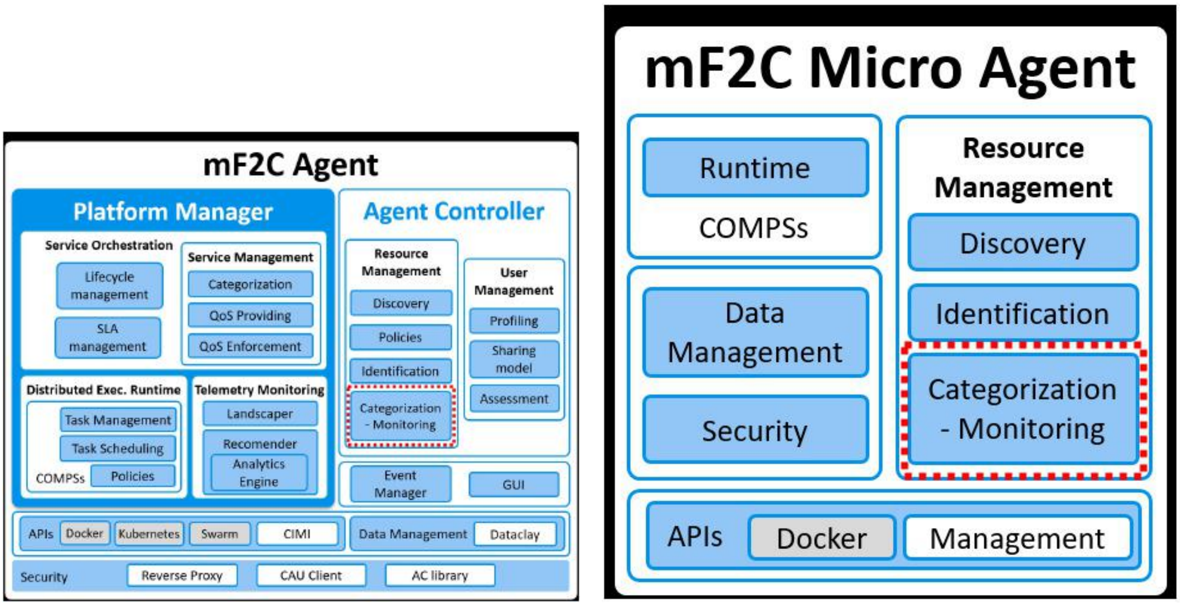
Previously we defined that, in the F2C system, devices those are sharing their resources can participate in the system as - 'Contributor', or 'Dichotomous'. Let's consider Fig. 5.7 as an illustrative scenario to depict that cooperative scenario. We may see that 'Fog Area1', contains one leader fog node and two fog node devices (i.e., smartphone, laptop) along with other connected devices (i.e., printer, bulb etc.). Let's consider that the two fog node devices and the leader fog node are participating in the system as 'Dichotomous'. In this case, the two fog node devices are sharing their resource information with the leader fog node. Thus, once the leader fog node receives the resource information for the two fog node devices, it aggregates all the information along with its own resource components information to form the resource information for the particular fog area. Then, the leader fog node shares this aggregated information to the higher layer in the F2C architecture. To make it work an strategy to aggregate the resources information must be defined. To that end, next, we propose a generalized aggregated resource description model for the F2C system. We identify the aggregated resource description model as aRD_{F2C}, and its structure is described as following:

```
aRD_F2C =
        <
          fog_node_id; fog_area_id;
          total_number_of_the_attached_F2C_enabled_resources;
          main_memory_capacity_info_(_in_MB_): <
                total_available_main_memory;
                F2C_resource_with_highest_main_memory;
                F2C_resource_with_lowest_main_memory
          >;
          storage_capacity_info_(_in_MB_): <
                total_available_storage_size;
                F2C_resource_with_highest_storage_size;
                F2C_resource_with_lowest_storage_size
          >;
          processor_info: <
                processing_capacity_info_(_in_percentage_): <
                      average_of_processing_capacity;
                      F2C_resource_with_highest_processing;
                      F2C_resource_with_lowest_processing
                >;
                processor_core _info_(_number_of_cores_): <
                      average_of_total_number_of_cores;
                      F2C_resource_with_highest_processor_core;
                      F2C_resource_with_lowest_processor_core
                >
          >;
          gpu_capacity_(_in_MB_): <
                total_available_gpu_capacity;
                F2C_enabled_resource_with_highest_gpu;
                F2C_enabled_resource_with_lowest_gpu
          >;
          power_info_remaining_time_(_in_seconds_): <
                average_time_of_power_remain;
                F2C_resource_with_highest_power_remain;
                F2C_resource_with_ lowest_power_remain
          >;
          IoT_&_other_attached_devices_info: <
                sensors_type_info;
                actuators_type_info;
                RFID_tag_type_info;
                other_attached_device_info;
          >;
          Security_&_Privacy_score: <
                average_score_for_F2C_resource;
                F2C_enabled_resource_with_highest_score;
                F2C_enabled_resource_with_lowest_score
          >
        >
```

By following this aggregated resource information, it can be easily drawn that it is quite different from the generalized resource description model of a single F2C resource. After getting all the resource information of a fog area, the leader fog node of the respective area is aggregating all of the information, and it is making an aggregated description file according to the upper mentioned model. The aggregated description file only contains the information about leader fog node id, fog area id, total number of fog nodes, the total capacity of main memory, storage, GPU etc., information about the highest and lowest main memory, storage, processing, GPU capacity of the F2C enabled fog node of the respective fog area and so on. Then after creating the aggregated resource information model, the leader fog node shares this information with the upper layer resources of the F2C paradigm. However, defining and designing both of these description models are giving us some dividend for adequately organizing the system resources information, as well as it is helping us to understand the overall computation capacity of any F2C-enabled system easily. Notably, following the proposed description models, in this thesis work and mF2C project [20], we are collecting the resource information for creating the information pool.

5.2 Integration of our proposals to the EU H2020 funded project

The H2020 European Union mF2C project has partially supported this research work. Therefore some of our proposals have been adopted and implemented in the mF2C project. The project is focused on designing and developing a hierarchical and decentralized management architecture for the fog-to-cloud scenario, which is intended to be secure, robust, scalable and efficient [220]. According to the project consortium members, the mF2C system intended to enabling a coordinated management solution for all the participating edge and cloud resource, with a view of optimizing services execution. The main component of the mF2C architecture is known as *Agent*. Deploying *Agent* on different computing nodes helps them to participate in the mF2C system. Notably, each *Agent* includes various set of functional blocks. In short, the mF2C management solution is a software suite to be deployed on the different devices willing to participate in mF2C. Interestingly, all those functional blocks have been split up into two different functional components: Agent Controller (AC), and Platform Manager (PM). The PM provides the high-level functionalities, responsible for inter-agent communications and can make decisions with a more global view. On the other hand, the AC functionalities have a more local scope, dealing with local resources and services [220].



(a) mF2C Agent

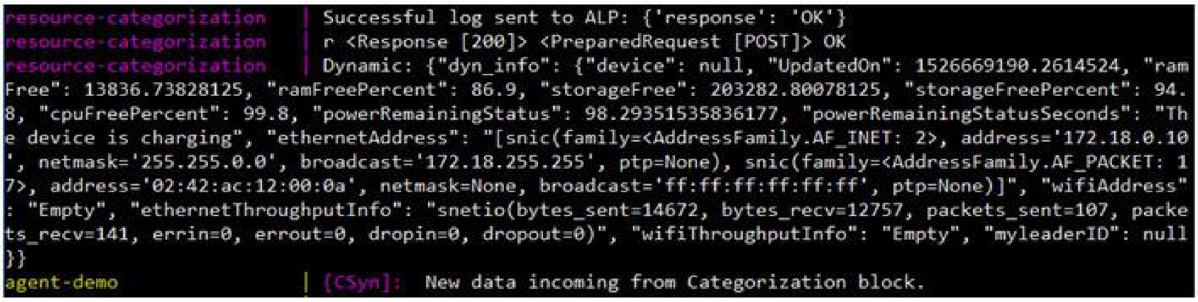
(b) mF2C Micro Agent

Figure 5.9: Architectural diagram: (a) mF2C agent, (b) mF2C Micro Agent

From an architectural perspective, the set of mF2C devices are distributed to different hierarchical layers. Even their resource capabilities are different in nature. Considering this fact, the consortium members have realized to build different versions of *Agent*. Basically, *Agent* is the default version, which is used by most devices within the architecture. Whereas, the cloud resources are using a slightly modified version of the standard *Agent*, which is known as - *Cloud Agent*. While the simplified *Micro Agent* is designed for use in constrained devices (e.g. Raspberry Pi). Significantly, the consortium has partially adopted and implemented our proposals for building the default *Agent* in the system. In Fig. 5.9(a) & 5.9(b), we present their proposed architectural diagram for the mF2C *Agent* and *Micro Agent*. In both of these figures, we have marked the *Categorization-Monitoring* functional block with red coloured dotted rectangles. For developing this functional block, the consortium has been partially adopted our proposals. Precisely, by combining the concepts and functionalities of our proposed architectural module/sub-modules (i.e., *Resource Categorization Sub-Module* and *Information Collector Module*) the *Categorization-Monitoring* functional block has been developed in the mF2C project.

The *Categorization-Monitoring* functional block is an integral part of *Resource Management* block. The *Categorization-Monitoring* block is mainly responsible for collecting the available resource components (e.g., RAM, CPU/GPU, Disk, Power, bandwidth, etc.) information for the participating devices and continuously monitoring them. The collection of this information helps to classify all the mF2C devices properly. Eventually, continuous monitoring of the devices, helping the overall system to build a resource information pool. Thus, feeding up by this information, other functional blocks (i.e., Lifecycle Management, Recomender, etc.) are participating in managing the tasks and offering the services efficiently. Also, in the upper layer mF2C devices (e.g., cloud and fog), this *Categorization-Monitoring* block is in charge of aggregating the information of lower layer mF2C devices.

(a) Static resource information



(b) Monitoring dynamic resource information

(c) Aggregated Resource Information

Figure 5.10: Resource information of mF2C device: (a) Static Info, (b) Dynamic Info, (c) Aggregated Info

Therefore, for offering the resource monitoring and info aggregation services, we built this functional block by using some simple Python packages (e.g., psutil[2], py-cpuinfo[3], etc.), and finally we dockerized[4] this functional block for working with other containerized functional blocks to build the whole mF2C *Agent* stack. In Fig. 5.10, we have presented the sample outcomes of the *Categorization-Monitoring* block. Where Fig. 5.10(a), shows the total resource components information (i.e., static resource information) of an mF2C device. While Fig. 5.10(b) presents continuous monitoring information (i.e., dynamic resource information), and Fig. 5.10(c), demonstrate the aggregated resource information (e.g., Fog-Area).

[2]https://pypi.org/project/psutil/
[3]https://pypi.org/project/py-cpuinfo/
[4]https://www.docker.com/

Chapter 6: Information Collector Module

The main characteristic of the Information Collector Module is to continuously monitor the participating system resources in order to keep track of the total availability of the computational capability. Notably, by continuous observing the system resources and service-task executions, this module is responsible for capturing a massive amount of information and helping to build the information pool in the F2C continuum. Significantly, the creation of the proper information pool and secure distribution of this information pool are the essential requirements for defining and designing a proper resource selection and allocation process in the F2C paradigm. Thus for satisfying these requirements, in this chapter, we have proposed an architectural schema for securely store and distribute the information over the network. Finally, by performing some evaluation tests, we justified the effectiveness of our proposed schema.

6.1 Revealing research challenge and solution

In any large-scale F2C-enabled smart system (e.g., smart city), a massive amount of heterogeneous devices are working. Notably, before managing them, it is essential to know their characteristics and collect this information. Significantly, collecting the devices characteristics-related information, helping to build a resource information pool in the system. Interestingly, the resource information pool helps in the process of effectively handling the system resources in the F2C paradigm. However, continuous monitoring of the system resources is also an essential demand for managing them properly. Typically, continuous monitoring of all the participated resources is

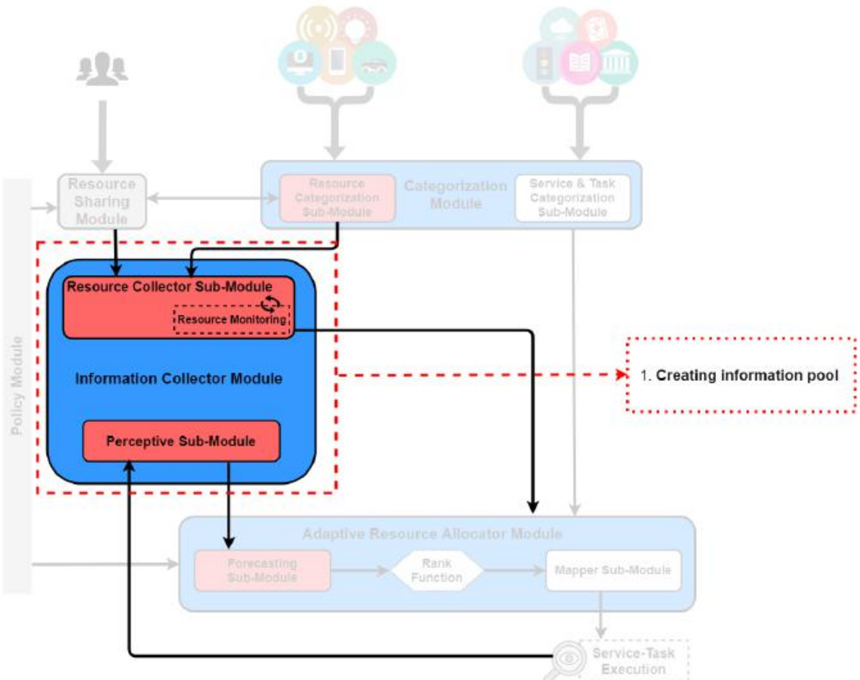


Figure 6.1: Revealing research challenge by development of Information Collector Module

generating a massive amount of resource statistical data in the F2C-enabled system. Hence, this resource statistical data along with the characterization information of participating resources, making a more rigorous, enriched and extensive resource information pool. Following our proposed resource management framework (Fig. 4.1), it can easily comprehend that mainly the *Resource Categorization Sub-Module*, *Resource Collector Sub-Module*, and *Perceptive Sub-Module* are responsible for making this kind of exhaustive resource information pool in the F2C-enabled system. Also, following any large-scale F2C-enabled smart system it is easily realizable that, a massive amount of Edge-IoTs (e.g., sensor) are working in the system. They are also capturing various environmental events and generating a colossal amount of sensing-data. Similarly, like resource information, this sensing-data is also considered as one of the ingredients for running various applications to offer different services among the subscribers of any F2C-enabled system. Notably, both of these data are creating a massive information pool.

Importantly, by observing our proposed resource management architecture, it can be easily recognizable that development and functionalities of the *Information Collector Module* is implicitly revealing one crucial research challenge, which is described in Fig. 6.1. Typically, for any F2C-enabled smart system, secure distribution of the information pool is a necessary demand. Remarkably, the secure distribution of information pool over the network is ensuring to prominently supply the ingredients (i.e., data) for any services. Particularly, the secure distribution of information pool improves the service qualities for any latency-sensitive applications. Significantly, secure data distribution over the network helps to search the proper resources for allocation to execute some tasks and provide some services. Considering these facts, in the next sub-sections, we propose a secure DDB schema in a large-scale F2C-enabled smart system and explain how our proposal improves the overall system to provide the latency-sensitive situation-aware services among the subscribers effectively.

6.1.1 Proposed schema for secure data storing in F2C

There are many scenarios where the coordinated management of fog and cloud resources brings in substantial benefits, such as e-health, industry 4.0, smart transportation or smart cities, just to name a few. Particularly, latency-sensitive situation-awareness applications put together a broad set of highly demanding services with a significant impact on smart cities scenarios. In fact, through a vast deployment of smart systems and devices, cities are evolving towards an improved and sustainable living facility offering a much better quality of life by reducing environmental pollution and also solving many other issues [221]. Several efforts are already made aimed at building up smart solutions for improving life quality in modern cities. For example, authors in [222] proposed a methodology for using WSN technology for reducing air pollution in modern cities. In the proposed scenario, many resource-constrained devices at the edge collect data to measure air quality. These data, in a smart city context, must be processed near to the edge for adequately providing the latency-sensitive situation-aware services (e.g., traffic monitoring, fire emergency).

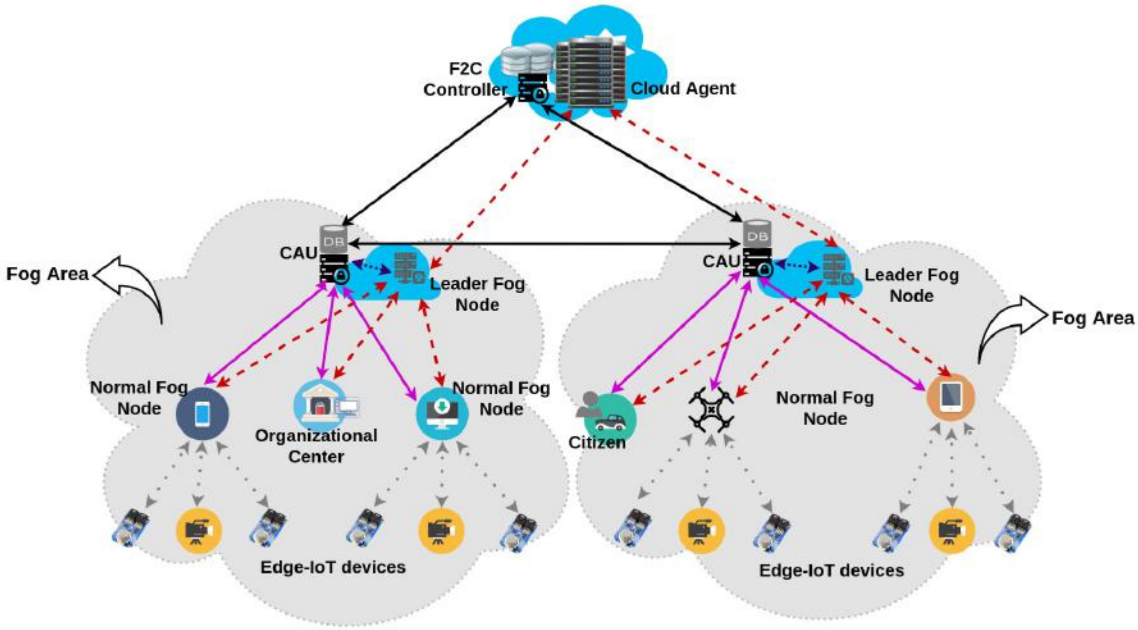


Figure 6.2: F2C-based execution model for monitoring traffic and air quality in a Smart City scenario

As described earlier, the coordinated management of cloud and fog resources may help to achieve that set of requirements, taking real-time decisions about the set of resources to use according to the real needs, be it at the cloud or the fog. Hence, when putting together the needs required by the envisioned highly demanding scenario and the features offered by the F2C model, we indeed notice that F2C may be an excellent management paradigm on which, a solution may rely on. For the sake of illustration, the proposed architectural solution is applied for air quality and traffic monitoring service in smart cities. Fig. 6.2 presents a topological approach of the tentative infrastructure required to deploy the air and traffic service on a city matching the F2C model. As shown in the picture, there are many quality measurement sensors and surveillance cameras geographically distributed over the city, aimed at measuring the air quality level and monitoring the traffic in real-time. Assisted by the surveillance cameras, information about the total number of vehicles moving in the city for a given time may also be collected. In practice, considering the volume of sensors and the volume of vehicles, a large volume of data is expected.

Moreover, for a much better service experience in a smart city domain, it is necessarily required to share the captured data with existing local government organization (e.g., meteorological dept., traffic controlling dept., etc.) and also among the citizens. Unfortunately, processing and securely sharing this high volume of data in real-time is pretty difficult when considering that edge devices are resource-constrained systems with poor storage and processing capacities, mainly due to two key factors. First, it is evident that it can be an excellent candidate to handle these needs, but bandwidth limitations and large distances may make the cloud, as a standalone solution, not to be the ultimate one. Second, unreliable communication between edge devices and cloud brings additional complexity in the system while also increasing the chances for data losses.

6.1.1.1 Architectural schema & components of SFDDM

In order to alleviate all the upper-mentioned issues, we develop a modified F2C-based execution model, supported by an architectural schema named - Secure F2C Distributed Database Management system (SFDDM). The SFDDM mainly consists of three (3) layers (i.e., Cloud, Fog, and Edge-IoT) and six (6) types of architectural elements (i.e., Cloud Agent, F2C Controller, Leader Fog node, Control Area Unit, Normal Fog node, and Edge-IoT device), as discussed below. Notably, in Fig. 6.3, we use different colours of arrows for denoting the various communication purpose between the different architectural elements of SFDDM. For instance, the red coloured dotted arrow indicates the control plane communication between the Cloud Layer and Fog Layer resources. Whereas, the black coloured arrow indicates the data plane communication between the Control Area Unit (CAU) and F2C Controller. In that figure, we used the violet coloured to denote the communication between CAUs and Leader Fog node (LFn), and purple coloured arrow for presenting the communication between CAUs and Normal Fog node (NFn). Also, we use the grey coloured arrow to show the combined data and control plane communication between NFn and Edge-IoT devices.

Cloud Layer includes Cloud Agent (ClA) and F2C Controller: The *Cloud Layer* is considered as the most resource-enriched layer. It is the uppermost layer of our proposing architectural schema, as shown in Fig. 6.3. According to our schema, two-types of architectural elements are working in this layer, the F2C Controller and the Cloud Agent (ClA). The F2C Controller is the master node acting as a certificate authority providing authentication for distributed Control Area Unit (CAU) located at the edge of the network. F2C Controller gives authorization for security provisioning at the edge of the network to the distributed CAUs. Also, it acts as the data center node for our proposing distributed database-oriented architectural schema. On the other hand, ClA is the cloud node, which is basically a composite form of cloud resources. It is responsible for providing the various kind of cloud facilities, including the storage facilities for historical data. The F2C Controller has secure intercommunication with the ClA. Key details and functionalities of the F2C Controller and CAUs are further extended in next.

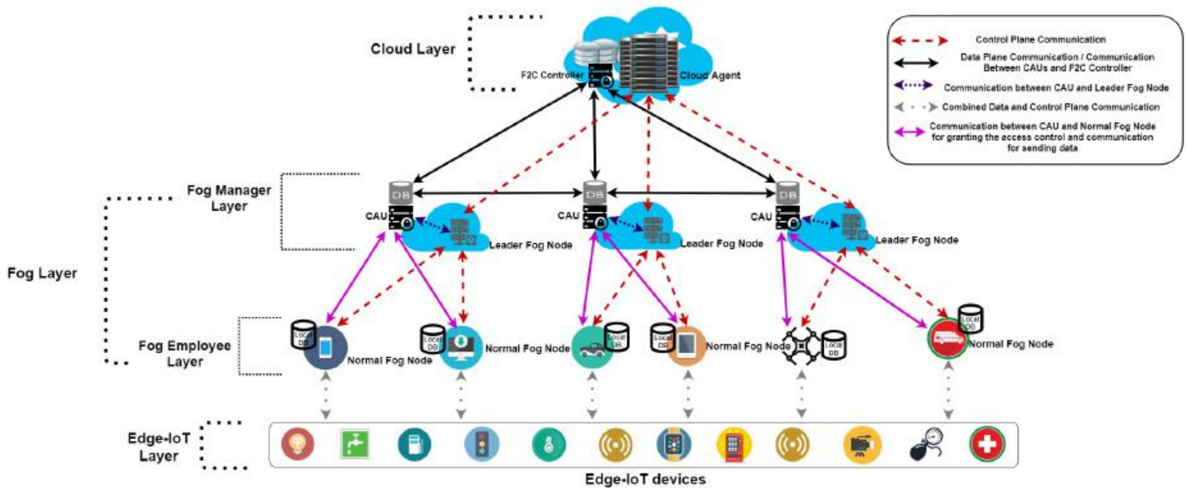


Figure 6.3: CAU-based Distributed Database in the F2C scenario

Fog Layer with Control Area Unit (CAU), Leader Fog node (LFn), and Normal Fog node (NFn): In the proposed architecture, the *Fog Layer* is mainly responsible for bringing the cloud facilities near to the edge of the network. According to the architecture schema shown in Fig. 6.3, this layer is a composite layer of two different sub-layers. The upper sub-layer is called as the Fog Manager Layer (FML). Another sub-layer resides below of the FML, referred to as the Fog Employee Layer (FEL). The FML sub-layer consists of two different architectural elements - the Control Area Unit (CAU) and the Leader Fog node (LFn). The CAUs are responsible for securely managing and storing the captured data, and whereas the LFns are responsible for bringing other cloud facilities near to the end-users. The FEL sub-layer elements are comparatively more resource-constrained than the FML sub-layer elements, known as Normal Fog node (NFn). The NFns act as the end-point or gateway for the Edge-IoT devices (e.g., sensor nodes). From a topological perspective, we may consider that in a modern F2C-enabled smart city scenario, several small areas may emerge to properly provide F2C services near to the edge of the network. These small areas are individually known as Fog Area (FA). In the proposed architectural schema, we consider that each fog area has one CAU and one LFn, respectively responsible for providing the security functionalities, storing the data near to the edge and managing other devices (i.e., NFn) within their scope.

Edge-IoT Layer consists of Edge-IoT devices: The *Edge-IoT Layer* is the bottom layer of the proposed architectural schema (Fig. 6.3), and consists of various kind of small Edge-IoT devices (e.g., sensor nodes, surveillance camera). These devices are responsible for both the continuous capturing of various environmental events (such as temperature sensing) and the execution of some actions (such as automatic fire extinguisher), according to some instructions.

Moreover, by recognizing the need for a robust data security management at edge devices, particularly in data-sensitive applications, the proposed architecture designs a distributed database cluster deployed on top of all CAUs and the F2C Controller. Next, we briefly describe the main security features of the proposed architectural schema.

6.1.1.2 Security services

The proposed data management security schema leverages the decoupled distributed security framework introduced in [223], briefly discussed in terms of its main components next:

F2C Controller: The F2C Controller resides at cloud and acts as the master node for the distributed CAUs deployed at the FML. The F2C Controller provides authentication, authorization, access control and secure channels for distributed CAUs. Once authorized, CAUs are responsible for providing security in their corresponding FAs.

Control-Area-Unit (CAU): The CAUs (deployed at the fog layer), are authenticated and authorized by the F2C Controller. CAUs are responsible for providing distributed authentication and access control, as well as for enabling the secure channels for their corresponding fog areas. Then, CAUs become trustable to act as distributed security controllers at other fog nodes (i.e., LFn and NFn) to provide the security requirements within their scope. CAUs provide security services for LFn, NFn and even other edge and IoT devices. All the CAUs are communicating with each other through a secure intercommunication channel. The CAUs, after being authenticated and authorized from the F2C Controller, can provide security at the edge without relying on the F2C Controller at the cloud.

In the proposed architectural schema, the communication between all CAUs, CAUs and F2C Controller, LFn and CAU, as well as NFn and CAU are implemented through transport layer security (TLS); hence all communications are handled through secure channels. CAUs are in charge of providing certificate authorities for their corresponding fog areas. Indeed, by using X.509 public-key certificates with the RSA cryptography algorithm, CAUs provide distributed authentication within their scope. Moreover, CAUs provide access control (Role-based) to the distributed database to prevent any unauthorized access. Finally, before storing the data, the CAUs encrypt the data by implementing the advanced encryption standard (AES), thus preventing any data leakage, eavesdropping, and any type of passive and active attack over the database.

6.1.2 Functionalities of our proposed schema

The set of expected services provided by the proposing SFDDM architectural schema relies on two algorithms (Algorithm 1 & 2). Algorithm 1, describes the data aggregation procedure, while Algorithm 2 describes the secure data storing and retrieving procedure in the proposing SFDDM schema. In both of these algorithms, MQ_s represents an array-list for saving the aggregated data. Whereas, in Algorithm 1, Q_s represents an array-list for temporarily saving the raw captured data. We also consider that the raw captured data those have the same capture time (C_t) and capture location (C_l), store to the Q_s for further processing. Likewise in Algorithm 2, we use MQ_r to denote the responded data packet. Primarily, MQ_s, Q_s and MQ_r are considered as individual empty array-list. In Algorithm 2, InV is an array-list for representing the index value. Initially, it also remains as an empty array-list.

In our envisioned scenario, considering latency-sensitive situation-awareness services in a smart city as an illustrative example, all Edge-IoT devices are responsible for both continuous capturing the different environmental events (i.e., air pollution measurement, passing vehicles counting) and generating (i.e., collecting) the data. Afterwards, once the data is generated, the Edge-IoT devices immediately send this data to their connected NFn. Once the NFn receives this data, it starts aggregating this data. To that end, the SFDDM solution adopts the statistical average method. Thus, based on the data context and with the help of data capturing time (C_t) and capturing location (C_l), the aggregation is being done. Most importantly, by doing so, it is possible to detect faulty data. Then the clean data packet is built for sending to the upper layer in the SFDDM architectural schema. At this step the Algorithm 1 stops working.

In order to store and retrieve the data, the Algorithm 2 is deployed. According to this algorithm, before sending or retrieving the data, all NFns must be authenticated, before sending the data packet to the CAU of their corresponding fog area. Once the authorized NFn gets the accessing permission from the corresponding CAU, then it can start sending the data packet (MQ_s). In order to ensure privacy and data integrity, we encrypt the (MQ_s), before storing it against the index value ($InV = [C_l, C_t]$). To that end, for retrieving the data, also all query makers (e.g., organizational centre, citizen) need to register themselves and claim the access permission from the CAU of their corresponding fog area. Once they get access permission, they can start querying for the desired data. When a CAU gets the query from the query makers, then the CAU starts looking for the data in the distributed database

Algorithm 1 For preparing the aggregated data in a NFn

Initial Consideration: $MQ_s = [\,], Q_s = [\,],\ C_l,\ C_t$

procedure AGGREGATION_OF_DATA
  Check the data sources for context
  if contexts are same **then**
    check the values of C_l *and* C_t *for every data sources*
    if *All the* C_l *and* C_t *are same* **then**
      Add all the data into Q_s
      Perform data aggregation on Q_s
      Estimate Mean, Median, and Mode of Q_s
      Calculate the difference between Mode value and each data of Q_s
      if *difference* > 1, *for any data in* Q_s **then**
        Send alert sms to LFn and Add Estimated Mode value to the MQ_s
      else *Add the Estimated Mode value to the* MQ_s
      end if
    else *Add all the data in* MQ_s
    end if
  else *Add all the data in* MQ_s
  end if
end procedure

Algorithm 2 For Storing/Retrieving data

Initial Consideration: $MQ_s = [\,],\ MQ_r = [\,], InV = [\,]$

procedure DATA_STORING/RETRIEVING
  Check the authentication of NFn and register them, if not registered
FIRST PHASE: **For Storing Data**
  Aggregate the data in NFn and insert it in MQ_s, against the InV
  Send the MQ_s, and InV to CAU
  Confirm the accessing permission of NFn for storing data in distributed DB
  Encrypt the MQ_s and store it into the distributed DB (in a table) considering the InV value
SECOND PHASE: **For Retrieving data**
  Confirm the accessing permission of NFn for retrieving data from distributed DB
  Make the query from other NFn, considering the desired InV value and table name to the CAU
  Then, CAU find the encrypted data from it's database, and then decrypt the data and also put it in the MQ_r
  CAU send the MQ_r to the NFn
end procedure

with the help of the mentioned InV. After finding the data from the distributed database cluster, the CAU decrypts the data and prepares the response packet (MQ_r) for sending to the query makers. Following this way, our schema securely delivers data to the query makers.

6.2 Performance Evaluation of the proposed schema

To evaluate the performance of our proposed architectural schema, we have configured our project testbed [21] and emulated our schema. In addition, we have also used the Python-based simulation tool (YAFS [224]) to generate the traditional Cloud-based and Edge/Fog-based execution models, in order to compare our performance with the traditional execution models. We have set up a prototype of the F2C-enabled smart city scenario in our research lab, where the Cloud Layer elements (i.e., F2C Controller and Cloud Agent) are hosted on a server with Intel Xeon family E5-2620 V4 series (clock speed @3GHz), 96GB RAM, 1TB Hard Drive, and running on Ubuntu 16.04LTS Linux. In our experimental testbed, all the Fog Layer elements (i.e., CAUs, LFn and NFn) are relatively small computing devices: Raspberry Pi3 B+ models, running on Ubuntu 16.04LTS Linux and with Cortex A53 @ 1.4GHz processor, 1GB SD-RAM, and 64GB micro-SD card. Similarly, we are hosting the LFn and NFn on some Raspberry Pi Zero models, working with 1GHz single-core CPU, 512MB RAM and, for storage purposes, we are using 8GB microSD for each of the LFn and NFn. For storing the captured data and securely distributing it over the network, we are creating a distributed database cluster over all the CAUs and F2C Controllers. For that purpose, we are using the containerized Apache Cassandra *(Dockerized-Cassandra)*. Tests have been performed implementing the multi-datacenter, multi-node based Cassandra cluster over the considered distributed schema. Security in our proposing schema has been ensured by implementing the aforementioned security functionalities. Similarly, we have customized the YAFS simulator to create the traditional cloud-based and fog-based smart city scenario, considering the same resources specification (i.e., RAM size, CPU specs., bandwidth), in order to compare with our proposed schema. For, cloud-based and traditional edge/fog based execution model, we are permanently storing the data in the cloud. Whereas, in the traditional edge/fog based model, before permanent storing of the data, the fog nodes are caching and processing this data for a limited amount of time. The feasibility of our schema has been validated performing the test over several amounts of distinct data packets. Individually, the average size of each of the packet is 30kB. By imitating the smart city scenario, we understand that at the edge of the network, the communication bandwidth is reduced. In our proposed schema, we consider the maximum connection speed between Edge-IoT devices and the NFn is 2 MBps. Similarly, the maximum connection speed between NFn and LFn/CAUs is 5-7 MBps. Moreover, the maximum connection speed between different FML's architectural elements (e.g. LFn and CAUs) and also with the cloud layer's element is 11 MBps. A thorough observation at any smart city scenario, easily shows that bandwidth utilization is one of the most critical issues to manage. Considering this requirement, we perform our first evaluation test to show the performance of our proposed schema.

6.2.1 Bandwidth utilization: Cloud vs Fog vs SFDDM

SFDDM aggregates the raw captured data in the NFn, before being sent to the distributed database. After that, the aggregated data is sent to be stored in the distributed database, located at the FML. Eventually, this reduces the overall bandwidth utilization for our proposing schema. In addition, as the distributed database cluster is residing near to the edge, it also reduces the data transmission time for storing the data into the database. In Fig. 6.4, we present the performance evaluation of our model. Here, the deep magenta coloured line represents the performance of the SFDDM schema, and the green coloured line and the red coloured line represent the performance of fog-based and cloud-based execution model, respectively. We observe that our model reduces the data packet transmission time by approximately 33% compared to the cloud-based execution model and approximately 44%

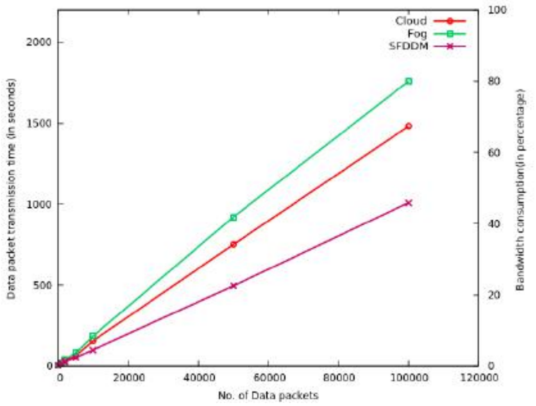
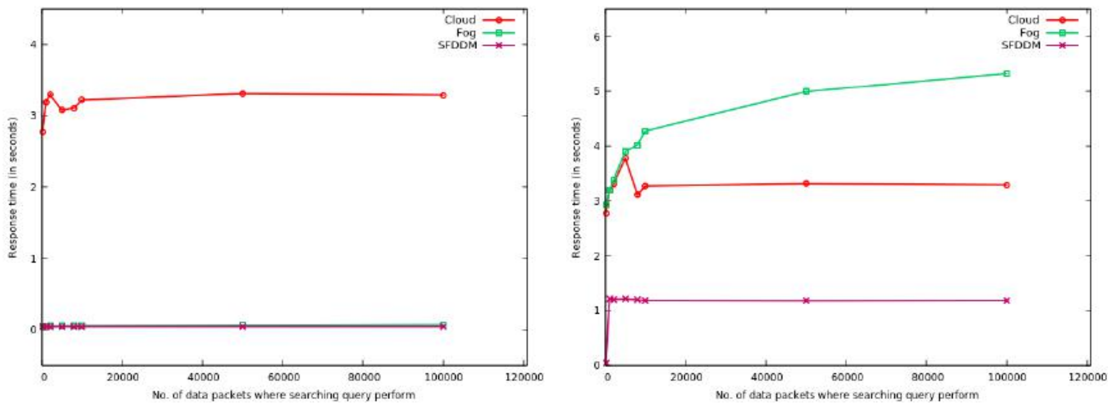


Figure 6.4: Data transmission vs Bandwidth usage

compared to the traditional fog/edge-based execution model. By performing this test, it is clearly shown that the traditional fog/edge-based execution model is the unsuitable candidate for transmitting data into the datacenter. In the traditional fog/edge-based execution model, the datacenter is located in the cloud and to store the captured data it needs to pass through the fog/edge layer. That added some extra time for transmitting the data for the fog/edge-based execution model.

6.2.2 Query response time: Cloud vs Fog vs SFDDM



(a) Data within local scope

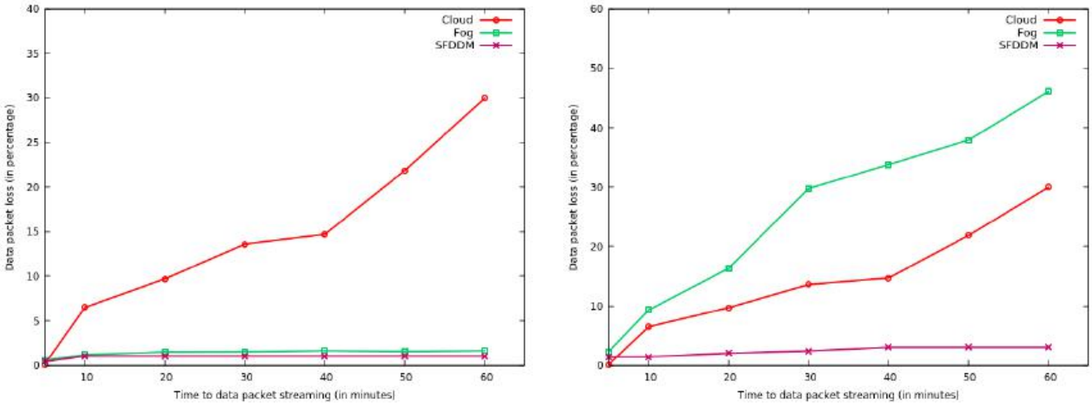
(b) Data in remote area

Figure 6.5: Query response time: (a) Data within the local scope, (b) Data in remote area

We consider two different cases to measure the query response time. In the first case (Fig. 6.5(a)), we consider that the data source and the query makers (i.e., organizational centre or citizen) are within the same scope (i.e., same fog area). In the second case (Fig. 6.5(b)), we consider the query for searching the data is made either from a remote area or from a different scope (i.e., different fog areas).

In the first case (Fig. 6.5(a)), we have found that, since the data is residing within the same scope, the query response time for the normal fog-based execution model and our SFDDM schema is quite similar; however, the query response time at cloud is extremely high. Interestingly, in the second case (Fig. 6.5(b)), we have found that our schema performs much better than the regular fog-based execution model. As the SFDDM has the distributed database near to the edge, the data is dispersed and therefore, closer to the edge of the network. However, in case of the normal fog-based execution model, the query has travelled through the upper layer (e.g., cloud layer), so this usually takes a long time and degrades the performance for normal fog-based execution model. The traditional cloud-based execution model again has higher response time because the data is residing far from the query makers.

6.2.3 Data loss: Cloud vs Fog vs SFDDM



(a) Data within the local scope

(b) Data is in the remote area

Figure 6.6: Data loss: (a) Data within the local scope, (b) Data is in the remote area

Finally, we measure the data loss. To that end, we consider the same two cases for some video streaming applications in a smart city scenario. In the first experiment (Fig. 6.6(a)), the data (image) source node (which hosts the surveillance camera) and the monitoring node (i.e., organizational centre) are residing within the same scope (i.e., in the same fog area), and in the second case (Fig. 6.6(b)), both nodes are residing in different scopes (i.e., in different fog areas). For both cases, we found that our SFDDM schema provides better performance than either traditional cloud-based or fog-based execution models. Indeed, implementing the distributed security functionalities and database near to the edge is guaranteeing a better performance for our SFDDM schema.

Our proposed SFDDM provides enhanced features and performance for any latency-sensitive situation-awareness applications in any F2C-enabled smart system. By performing some tests, we have measured the performance of the SFDDM schema and compared with those in the cloud-based and fog-based schema. Notably, this research work is presented as the primary steps for making an efficient and secure F2C-based system. By developing this schema, we have already advanced for addressing secure data distribution-related issues. Thus, by accomplishing this work, we also able to design an effective way for securely collecting and storing the data through the overall system. Significantly, by ensuring all of the aforementioned facilities, SFDDM massively helps the *Information Collector Module* for working efficiently in our proposed resource management framework. Eventually, that also help to build a more secure, advanced and enriched information pool in the F2C-enabled smart system; which ease the process of appropriate resource searching and mapping.

Chapter 7: Adaptive Resource Allocator Module

The Adaptive Resource Allocator Module is the main module in our proposed resource management schema, which is responsible for selecting and allocating the appropriate resources for executing some tasks. This module has the capability for intelligently and adaptively scheduling the tasks among the available system resources. Notably, this module enables the scope for implementing the machine learning operations in the resource selection and allocation process to the F2C paradigm. Thus, by discussing the architectural and functional aspects of the Adaptive Resource Allocator Module, initially in this chapter, we thoroughly present the smart way of the resource selection and allocation process in F2C paradigm. Considering the large scale F2C-enabled system, we realize that in any F2C-enabled system, all the machine learning operations can be done in different ways. Importantly, in this chapter, we propose two different approaches for implementing the machine learning operations in the F2C paradigm to intelligently select and allocate the appropriate resources for executing some tasks and provide the services. Later on, in this chapter, we also present the experimental results for validating our proposals.

7.1 Architectural and Functional details of Adaptive Resource Allocator Module

Adaptive Resource Allocator Module is one of the core modules for our proposed resource management framework. According to the service-task requirements, this module is responsible to find-out the best-fitted available resource(s) for executing the requested task(s) in the F2C-enabled system. Thus, the main job of this module is to build an advanced resource allocation technique in the F2C-enabled smart system. However, the resource allocation process heavily depends upon the suitability matching between the requested task requirements and available system resources. Also, the resource allocation process relies upon the selection of best-fitted resource(s).

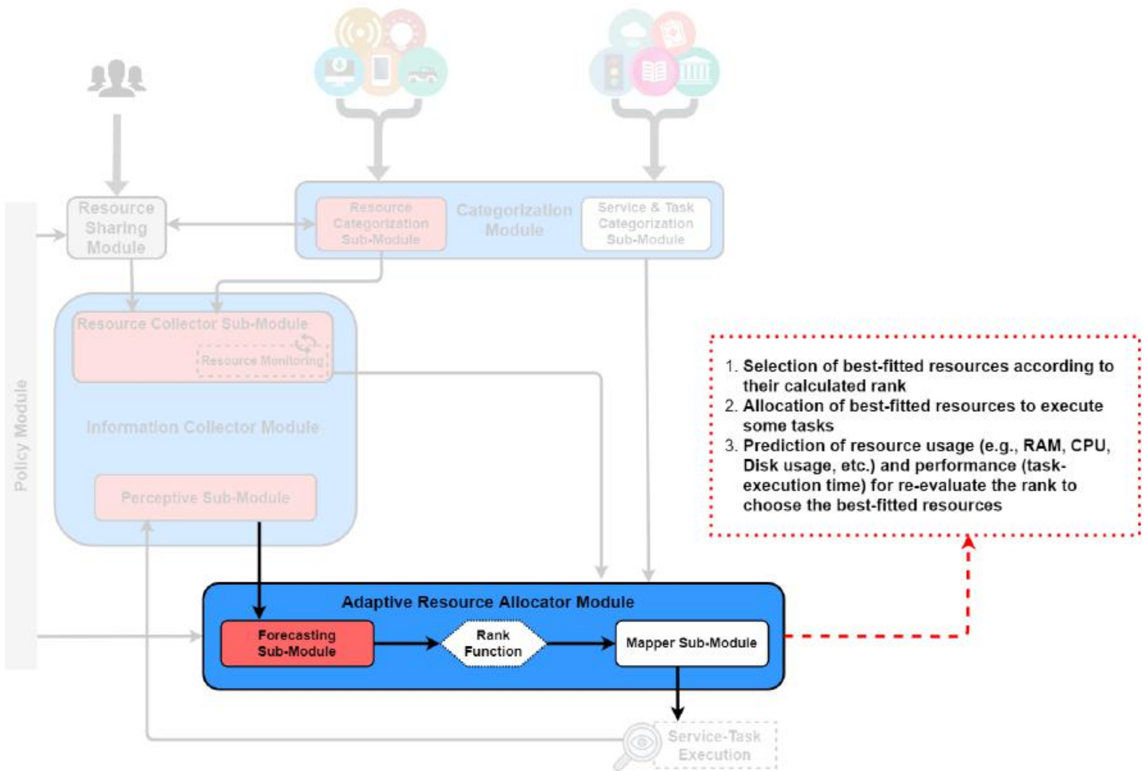


Figure 7.1: Research challenges addressing by Adaptive Resource Allocator Module

However, in any large-scale and distributed smart computing system (e.g. F2C-enabled smart city), on-demand resource allocation is a critical process and one of the most crucial steps for designing an advanced and intelligent resource management mechanism [203]. Especially, proper allocation of available system resources based on their current and future demand for executing some computational tasks is a challenging job. In order to address this issue, many researchers proposed different techniques [204]–[206], which can be classified into two approaches: reactive and predictive [165]. According to these researchers, in the case of reactive resource allocation approaches, the current state of resources has to be measured before allocating them. Whereas in the case of predictive approaches, the system can priorly forecast the computational load and the overall performance of system resources before allocating them [166]. In a predictive approach, the system can measure the performance and computational load of system resources priorly; so this approach explicitly helps for designing a better and intelligent resource management mechanism [167]. Thus, to perform some sort of forecasting operation, it is necessary to design a schema of Machine Learning (ML) based resource management mechanism. Eventually, that would help the overall system to be more intelligent and advanced.

Considering all these aspects, in our proposed resource management architecture, we designed and developed the *Adaptive Resource Allocator Module*. This functional architectural module is mainly responsible for selecting the best-fitted resource(s) to execute some task(s), based upon the prior experiences. Notably, we found that this module has been evolved to address three different research challenges of the resource allocation procedure in F2C paradigm. In Fig. 7.1, we described those three challenges as follows: 1. *Selection of best-fitted resources according to their calculated rank*, 2. *Allocation of best-fitted resource(s) to execute some task(s)*, and 3. *Prediction of resource usage(e.g., RAM, CPU, Disk usage, etc.) and performance(task execution time) for re-evaluate the resource rank to choose the best-fitted resources*. Typically, to address the first two challenges in this thesis work, we have already adopted some of the existing strategies (e.g., [185], [225], [226]), which have been discussed earlier. Essentially, by following the architecture of F2C-enabled system, we realized that the prediction-based re-evaluation of resource ranking is a very complex work and challenging issue, which need to be addressed properly. Hence, in this work, we have given our intense focus to solve the third challenge.

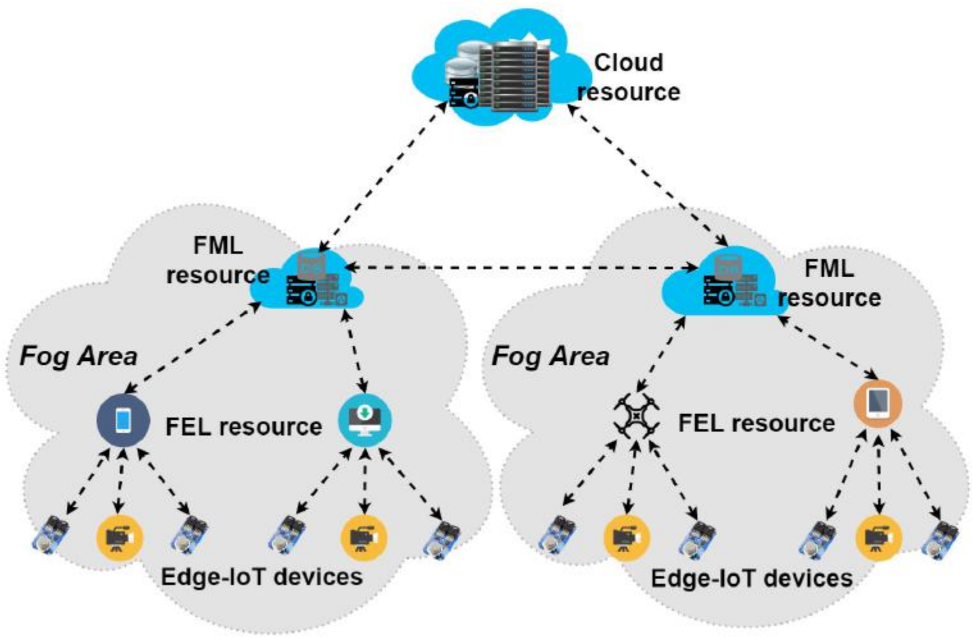


Figure 7.2: Prototype of F2C-enabled execution model in a smart city domain

The F2C is represented as hierarchical, combined and layered computing architecture. To better understanding the functional architecture of F2C-enabled smart computing platform, in Fig. 7.2, we depicted the architectural diagram of F2C-enabled smart city scenario. Notably, F2C is an emerging computational platform. Like as many other modern computing platforms (e.g. cloud, Grid); performs the ML-based resource allocation technique is an essential demand for designing an adaptive resource management mechanism in the F2C-enabled system. Vitally, ML technology relies on training the prediction model with a large amount of historical data. Alternatively, in case of unavailability of these data, models can be learning on-the-fly, understanding that in initial stages, the prediction will not be that accurate but will improve with the time and system use. In this case, such learning can be done in two ways; either in the logically centralized location or distributedly near to the data sources locations. Truly, both of these techniques have several advantages and disadvantages in terms of computation, resource consumption, service latency, data privacy etc. So, considering all of these facts, in this thesis work, we have decided to investigate further on these two techniques. Importantly, we have already implemented the multi-node, multi-datacenter based secure DDB cluster over the FML and cloud resources. So, all the required data has been distributed over the cluster. Thus in the first technique, we assume that by taking the advantages of secure DDB cluster, FML and Cloud resources can both perform ML operations. For the seek of simplification, in that case, we consider the ML operations have been performed in the logically centralized location. Whereas, in other technique, we assume that, all the F2C resources (e.g., cloud, FML, and FEL resources) can able to perform the ML operations with the help of their local data. In that scenario, with the assistance of global coordinations, all the F2C resources can able to perform the ML operations distributedly near to the data sources. Typically, we called the first technique as the *Cluster-based logically centralized approach for forecasting resource usage and performance*, whereas in the second technique we named it as the *Collaborative learning-based approach for forecasting resource usage and performance*. Later on, we have given our focus for identifying how these two techniques can help design the advanced and sophisticated *Adaptive Resource Allocator Module* in our proposed resource management framework.

7.1.1 Resource selection and allocation procedure

Before continuing the investigation on the topic of different techniques for performing the ML operations to improve the resource allocation, it is relevant to define how the resource selection and allocation process can be made in F2C paradigm. Typically, the resource allocation process heavily depends upon the suitability matching between the requested task requirements and available system resources. Also, the resource allocation process relies upon the selection of best-fitted resource(s). Therefore, the calculation of suitability matching score is a crucial step before allocating the suitable resource. Considering this fact and similarly like the [185] work, we adopted the *Student Project Allocation (SPA)* problem to calculate the match score (γ_r) of the resources. After that, for selecting the best-fitted resource(s), we calculate their rank. For that purpose, we consider a cost-model [225], to determine the global cost information (c_r) of the system resources. Then, by endorsing the multi-criteria decision analysis-based *Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS)* method [226], and considering the match score (γ_r) and the global cost information (c_r), we compute the rank of the resources (R_r). Finally, based on the rank of the resources (R_r), we allocate them to execute some tasks. The following steps are performed for allocating the system resources:

Step 1: *Search for local resources*; if they do satisfy the request, then stop searching (the resources have been found); otherwise perform the next steps.

Step 2: *Request to the nearest Fog Manager Layer (FML) or Cloud resource for searching the appropriate resources*; FML or Cloud resource, looks to the distributed database (DDB) cluster in search of appropriate resources.

Step 3: *In the DDB cluster calculate the matching score*; using the SPA algorithm, the available resources can be grouped and classified into three categories: highly match, moderate match, and low match, as:

$$(matching\_score \quad : \quad \gamma_r \ = \ [3, 2, 1])$$

Step 4: *Considering the global cost information and matching score determine the rank of resources*; endorsing the TOPSIS method and considering the global cost information, calculate the ranking for each resource, as:

$$(resource\_rank \quad : \quad R_r \; = \; (\gamma_r, \; c_r))$$

Step 5: *Based on the rank choose the appropriate resource*; based on the ranking (R_r), the appropriate resource must be chosen for the mapping with the requested service.

7.2 Cluster-based logically centralized approach for forecasting resource usage and performance

For the sake of better understanding the prediction mechanism in the F2C paradigm, we have already depicted the thorough architectural schema in Fig. 7.3. The presented architecture is an initiatory approach for building a prediction-based resource management mechanism in the F2C paradigm. As the F2C is still in a development stage, there are many opportunities for improvement. For example, to build a prediction-based resource allocation technique in the F2C paradigm, we have introduced many components (i.e., Smart Cache, Task Scheduler) in our architecture, but confirming the final placement of these components and the coordination among them is still a challenging job. Certainly, to improve the proposed architecture, that challenge (and many other) needs to be addressed. In this work, we propose the initial version of the F2C architectural schema for performing the prediction process to the FML or Cloud resources. According to our proposing architecture (Fig. 7.3), similarly like as cloud layer resources, also the FML resources (i.e., mainly LFn and CAU) can involve in the forecasting process to predicting resource usage and execution time for upcoming computational tasks. By thorough observation of Fig. 7.3, it can be easily recognized that the communication purpose of various components is different. Therefore, in that figure, we used different colour arrows to denote that. Notably, the brief description of the communication purposes has been described in that figure. In this section by following the proposed architectural schema (Fig. 7.3), we are going to briefly describe the various components of F2C resource, which are potentially involved in the forecasting/prediction process. Then, by briefly discussing the prediction/forecasting process in F2C paradigm, we are going to explain how the F2C resources can be adequately allocated for executing some tasks.

7.2.1 F2C resource components: Involved for predicting resource usage and performance

We earlier described that the F2C is a hierarchical, combined and distributed computing platform. Interestingly, in the F2C paradigm, many components are working together to achieve higher accuracy for predicting resource usage and execution time, what might lead to building an intelligent and proper resource management mechanism in the F2C paradigm. Next, we focus on those resource components briefly.

Smart Cache This is the core component, which is involved in forecasting/predicting resource usage and performance. It consists of two sub-components *Trainer component* and *Predictor component*. The job of the *Trainer component* is to build and train the machine learning (ML) model, whereas the *Predictor component* is responsible for forecasting the resource usage and performance assisted by the trained model. In our proposed architecture, as the ClA and LFn are responsible for performing some computational tasks and controlling other F2C resources, so it is pretty much evident that the Smart Cache is an integral part of these two kinds of F2C resource components.

Task Scheduler Based on the prediction result and following the requested computational task information, this component helps to choose and allocate the appropriate F2C resource(s) for performing the task(s). Basically,

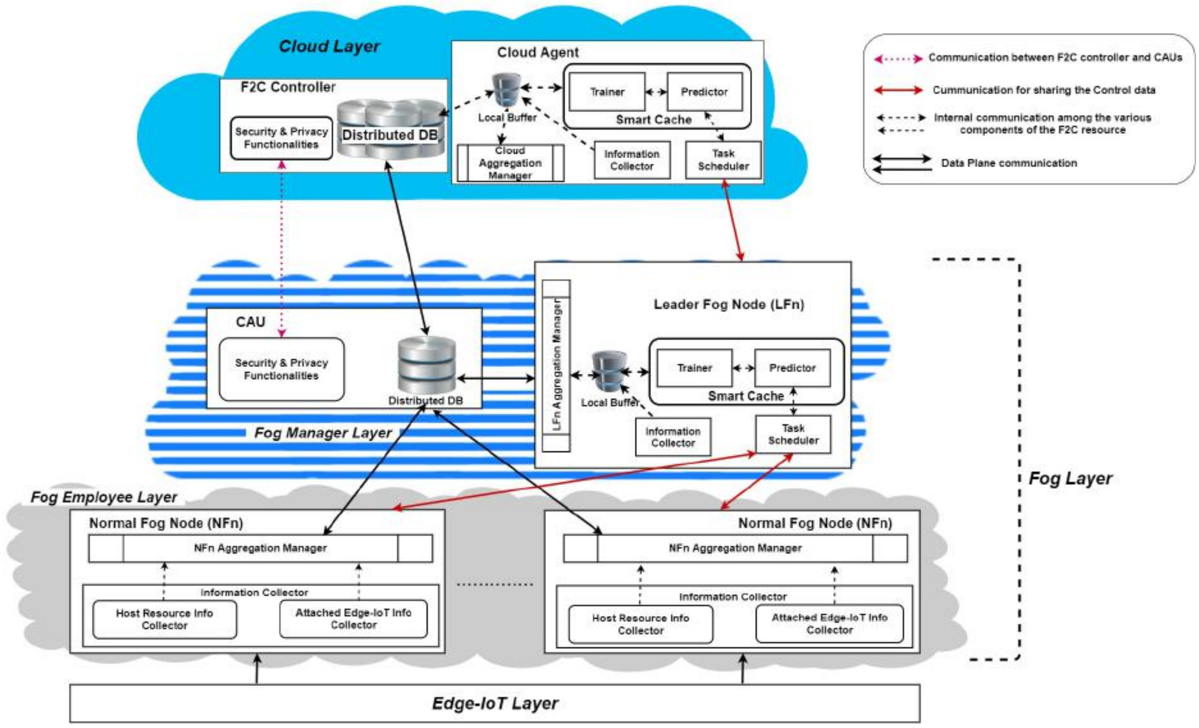


Figure 7.3: Architectural schema for Cluster-based logically centralized approach for predicting the resource usage and performance in F2C

it is a provisioning component, which helps to find out the suitable F2C resource for executing the requested task(s). Similarly, like as the Smart Cache component, it is also residing inside the ClA and LFn and assisting them to choose the proper F2C resources for executing some requested tasks.

Aggregation Manager As the name suggested, this component is in charge of aggregating the current state information and behavioural information of F2C resources and sending that aggregated information to the nearest DDB cluster node (i.e., CAU, F2C Controller). With the help of the Aggregation Manager component, F2C resources can aggregate the monotonous sensing-data and helping to store them to the distributed database for further analysis.

Information Collector This component is a kind of monitoring component, which is responsible for continuously collecting the host resource information and also collecting the captured sensing-data from the attached Edge-IoT devices.

Besides all of these resource components, the fact of keeping and distributing the data over the network (i.e., implementing the distributed database over the F2C controller and CAUs) will improve the prediction procedure, also eventually helping to speed-up the resource allocation procedure in the F2C paradigm.

7.2.2 Prediction-based resource management mechanism: Steps for forecasting and allocating the F2C resource

For any modern computing platform (i.e., fog computing) predicting the system resource usage and performance is the initial footprint to design an intelligent and proper resource provisioning and management mechanism [227]. By considering this fact, in Fig. 7.4, we present a flow-chart diagram to show the process of prediction-based resource management mechanism in the F2C paradigm. As we have already designed a secure DDB cluster over the cloud and FML resources, so by retrieving the data from the cluster, both the cloud and FML resources can perform the ML techniques. In pursuance of simplification, in that case, we consider the ML operations are performing in the logically centralized location (i.e., cluster node - FML or Cloud). Notably, following any smart computing platform and considering all the challenges for defining the proper and smart resource management mechanism of our proposed F2C paradigm, we understand that the type of target classes (e.g., RAM usage, CPU usage, task execution time) are known to us. Since the type of target classes for prediction are known, we adopted the *supervised* machine learning technique, to predict the resource usage and execution time for upcoming requested tasks.

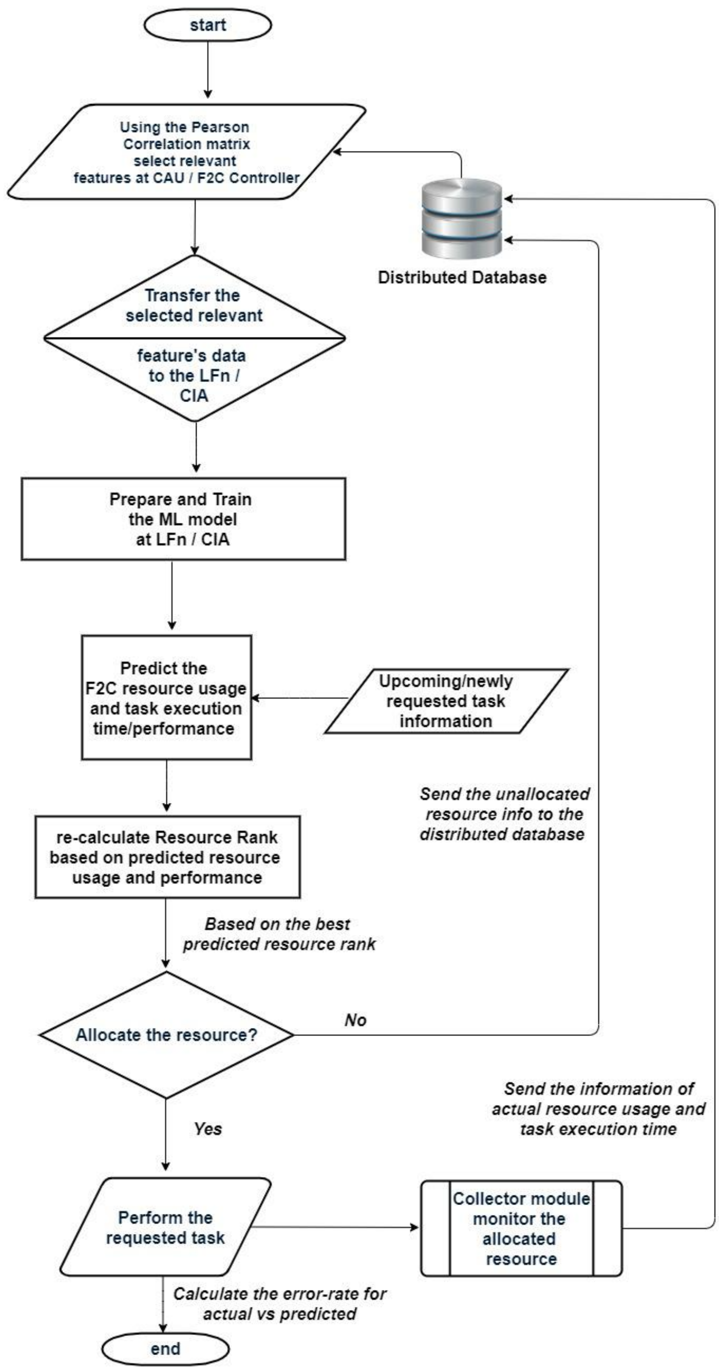


Figure 7.4: Workflow for Forecasting-based resource allocation

Typically, the dataset is a vital ingredient for the ML techniques. So before designing and training the ML model for forecasting future behaviour and states of the system resources, it is indispensable to either clean-up or process the available dataset and finds out the relevant categories of data. More precisely, it is necessary to identify all the features which have an impact on the target variable. From the relevant set of existing techniques for selecting the relevant features, we adopted the *Pearson correlation coefficient* matrix [228] for identifying the relevant features, which have a high impact on the target classes (e.g., RAM usage, CPU usage, task execution time). The *Pearson correlation coefficient* (r_{xy}) for some given paired data (e.g., $(x_1, y_1), \ldots, (x_n, y_n)$) is described as follows:

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{7.1}$$

where n is the sample size; x_i, y_i are the individual sample points indexed with i; and $\bar{x}$, $\bar{y}$ are *sample mean*. In the proposed F2C architecture, as the data is distributed over the CAUs and the F2C Controller, it is relevant to perform the data filtering process in the CAUs and F2C Controller. After that, the filtered data is transferred to LFn and ClA for performing the prediction process. In our F2C paradigm, LFn and ClA both have a particular component (i.e., Smart Cache) for performing this prediction process.

After getting the filtered data, based on some existing ML technique, the *Smart Cache* (more precisely *Trainer component*) build a trained ML model for predicting the upcoming/newly requested tasks. Then, based on this trained ML model, the *Predictor component* forecasts both the system resource usage ($Ru_p{}'$) and execution time ($Ex_p{}'$) for the available F2C resources, in order to perform the newly requested task. Then, before allocating the newly requested task (considering the predicted resource usage ($Ru_p{}'$) and execution time ($Ex_p{}'$)), the *Task Scheduler* component calculate the individual ranking (i.e., $R_r = (Ru_p{}', Ex_p{}')$) for all the available F2C resources and after that, it (i.e., *Task Scheduler*) chooses the best possible ranked (R_r) resource for executing the newly requested task. Once the task has been allocated and performed, the monitoring component (as earlier described all F2C resources (i.e., ClA, LFn and NFn) have a special kind of monitoring component, referred to as *Information Collector*), collects the actual resource usage and complete task execution time information. Then with the help of *Aggregation Manager* and *Local Buffer*, it sends this information to the nearest node (i.e., CAU or F2C Controller) of the distributed database cluster. Significantly this information improves the accuracy level to choose the best-fitted resource for executing the next upcoming task.

7.2.3 Performance Evaluation: For Cluster-based logically centralized approach

Building the ML-based prediction model and improving its accuracy-level is the initial step for designing a proper and smart resource management mechanism in any computing paradigm. So, considering that fact in this work, we focus on building an accurate ML-based prediction model and evaluating its performance in our considering F2C domain. For that purpose, we already configured our project testbed [21] for performing some tests and evaluate the accuracy of our proposing ML-based mechanism. Then taking a vast amount of images, we execute some simple image-recognition application [229] in the F2C resources (except CAUs and F2C Controller), to generate the raw dataset[1] for predicting the resource usage and execution time in the F2C paradigm.

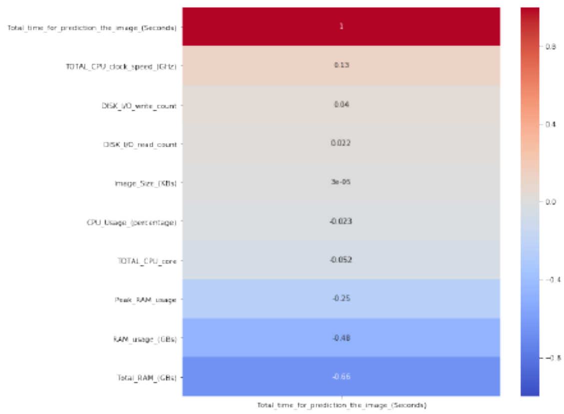
In our research lab, we have set up a prototype of the F2C-enabled smart city scenario, where the Cloud Layer elements (i.e., F2C Controller and Cloud Agent) are hosted on a server with Intel Xeon family E5-2620 V4 series (clock speed @3GHz), 96GB RAM, 1TB Hard Drive, and running on Ubuntu 16.04LTS Linux. The Fog Layer resources (i.e., CAUs, LFn and NFn) are relatively small computing devices, and following Fig. 7.2, it can be easily seen that the number of participating fog layer resources are quite higher than the cloud layer resources. Most importantly, they are hugely diverse [230]. For that reason and to avoid the unnecessary resource shortage issues, we created a significant number of VMs (approx.∼ 32) in two relatively high-end computing machines.

[1]https://gitlab.com/data-snoopers/appload

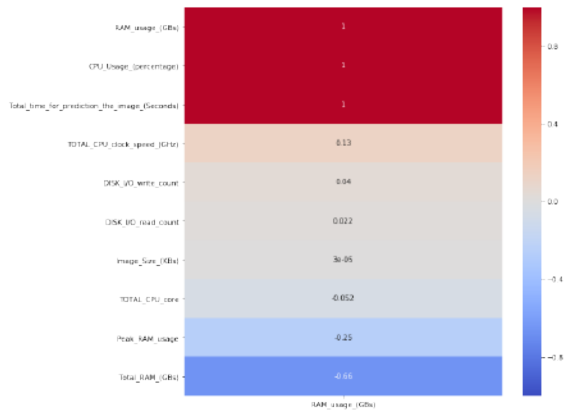
The specification of every VM is mostly different from others. Some VMs are relatively a bit resource-enriched, and some are not so. In our testbed, we considered those resource-enriched VMs as the FML sub-layer resources (i.e., CAUs and LFn) and except them, all others are considered as NFn resources. All fog layer resources are running on Ubuntu 16.04LTS Linux. We implemented a distributed database cluster over all the CAUs and F2C Controller. For that purpose, we are using the containerized Apache Cassandra *(Dockerized-Cassandra)*. Tests have been performed implementing the multi-datacenter, multi-node based Cassandra cluster over the considered distributed schema.

From the observation of a smart city scenario, one can easily understand that at the edge of the network, the communication bandwidth is reduced. Considering this fact, in our proposed schema, we limited the maximum connection speed between Edge-IoT devices and the NFn to 2 MBps. Likewise, the maximum connection speed between NFn and LFn/CAUs is 5-7 MBps. Moreover, the maximum connection speed between different FML's architectural elements (e.g. LFn and CAUs) and also with the cloud layer's element is 11 MBps. A thorough observation at any smart city scenario, quickly shows that bandwidth utilization is one of the most critical issues to manage. Considering this setup, we perform the tests to validate the effectiveness of our proposing architectural schema.

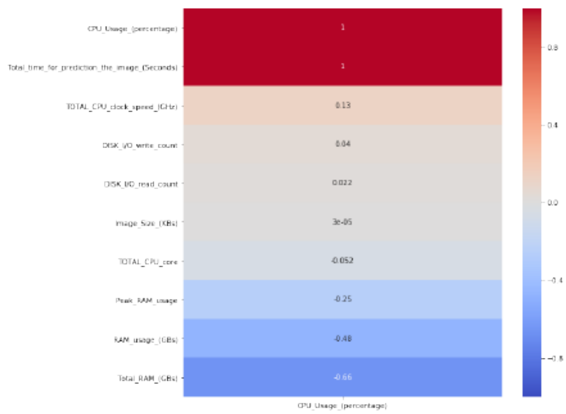
7.2.3.1 Dataset generation for train the ML model:



(a) Performance/Task Execution time prediction



(b) RAM usage prediction



(c) CPU usage prediction

Figure 7.5: Feature Selection for: (a) Performance/Execution time prediction, (b) RAM usage prediction, (c) CPU usage prediction

We earlier identified that cleaning the raw dataset and filtering the appropriate data is the primary step for performing some ML technique. According to our proposed architecture, this process is done on CAUs or F2C Controller. As our main goal is to predict the performance of F2C resources, first we filter our collected raw dataset, using the *Pearson correlation coefficient* matrix to find out those features having a strong influence for predicting the task execution time. After filtering the raw dataset, we may conclude that in our experiment, task execution time is critically dependent on the RAM related information (i.e., Total RAM size and RAM usage) of the system resources. Also, it is minorly correlated with the CPU information (i.e., CPU clock speed) of F2C resources. In Fig. 7.5(a), we represented the Pearson correlation heatmap for showing the correlation of all other independent variables with the output variable: execution time (i.e., Total_time_for_prediction_the_image_(Seconds)). Following that figure (Fig. 7.5(a)) and considering the general scaling of correlation values [231], it can be easily concluded that the task execution time has a moderate negative correlation with RAM features of F2C resources (i.e., Total RAM size and RAM usage). Also, the task execution time has a minor positive correlation with CPU information (i.e., CPU clock speed).

For any computing system, the task execution time is strongly related to the system resources [163]. Hence, we can conclude that before designing an intelligent and proper resource management mechanism for the F2C system, it is pretty relevant to predict the resource usage for the available system resources, what would implicitly help to reduce the resource consumption of the whole F2C system. From the earlier experiment, as we found that the task execution time is significantly correlated with the RAM related information of an F2C resource; so we perform the same procedure (i.e., *Pearson correlation coefficient*) to identify the features which have strongly correlated with RAM usage information. In Fig. 7.5(b), we presented the heatmap to understand which features are strongly correlated with RAM usage. Interestingly, we found that CPU usage and task execution time both have a very high positive correlation with the RAM usage of F2C resources. So, then we also follow the same procedure to identify all the features; those have a strong correlation with the CPU usage of F2C resources. In Fig. 7.5(c), we depicted the heatmap to identify all the features which correlate with the CPU usage of F2C resources.

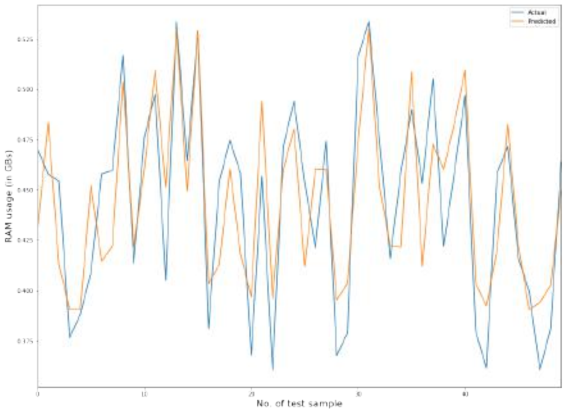
Surprisingly we found that the Storage information (i.e., Disk I/O) of F2C resources, has not any significant correlation with the task execution time, RAM usage and CPU usage information. So then we assume that it is worthily to predict the RAM usage, CPU usage and task execution time for designing a proper and intelligent resource management mechanism for the F2C system. To that end, according to our proposed F2C model, after this filtering process, the cleaned and filtered data must be transferred/copied to the LFn or ClA for further processing.

7.2.3.2 Resource usage and performance prediction accuracy:

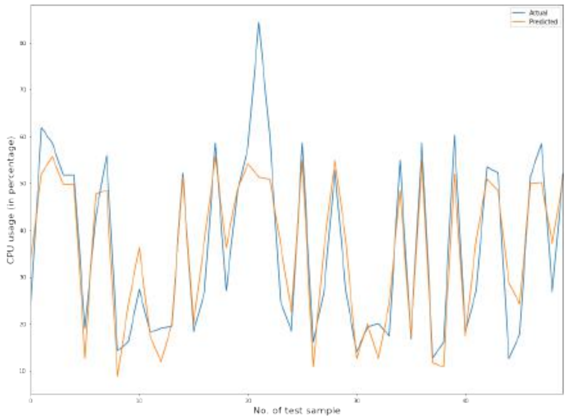
Following the F2C paradigm, we understand that, due to the resource-constrained nature of fog layer resources, it is very much tough for them to perform some additional heavy computation work for prediction. For that reason, it is necessary to choose an appropriate and low computation complexity-based ML technique for our F2C paradigm. As the linear regression ensures to consume less amount of resources for performing[232]; so to predict the resource usage and performance of the F2C resources, we adopted the *multivariate linear regression* ML technique [233]. Based on this ML techniques, we build and train the ML model for predicting the resource usage and execution time in our F2C paradigm. Importantly, evaluating the accuracy of the prediction model is highly necessary to justify the effectiveness of our proposed architectural schema. For that purpose, we calculate cost function value ($J(\Theta)$) to evaluate the prediction quality, as follows:

$$J(\Theta) = J(\theta_0, \theta_1, ..., \theta_n) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 \tag{7.2}$$

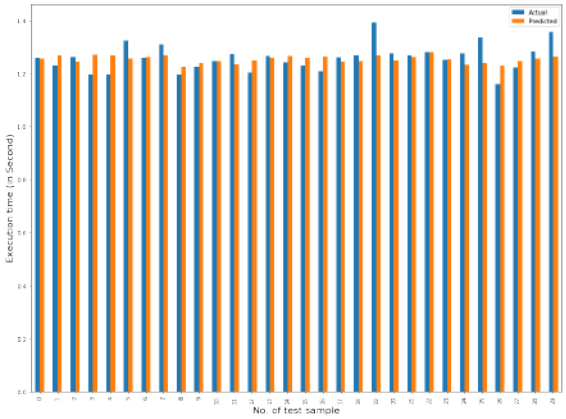
Where m is the number of sample data, $h_\theta(x)$ is the hypothetical value, which has been predicted by our ML model and y is the actual value. Now, to build more accurate prediction model for our F2C paradigm, we adopted the *gradient descent* algorithm [234] to reduce the value of the cost function ($J(\Theta)$).



(a) RAM usage



(b) CPU usage



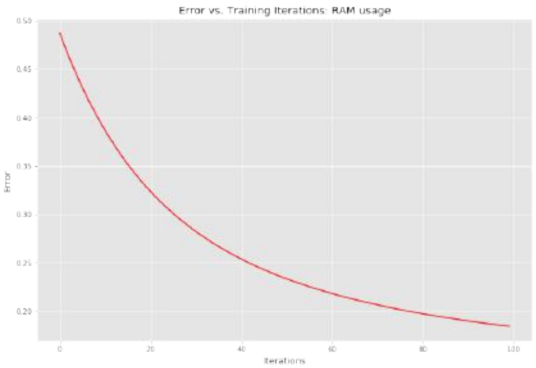
(c) Performance/Task execution time prediction

Figure 7.6: Model's performance evaluation: Actual vs Predicted

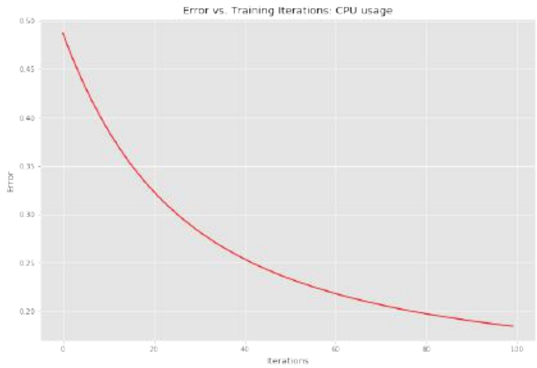
It has been earlier mentioned that before choosing the appropriate resource for executing the newly requested task, it is pretty relevant to predict/forecast the execution time for the newly requested task, and resource usage (e.g., RAM and CPU usage) of the available F2C resources. For that purpose, we already built and trained our ML model for predicting the execution time and future resource usage. Then, we perform some tests to validate the efficiency of our model. In Fig. 7.6(a), 7.6(b) & 7.6(c) we present the test results (i.e., predicted vs actual) to show the accuracy of our ML model for predicting the resource usage and future task execution time in F2C paradigm.

During our experiments, we found that initially, the cost function values of our ML model were a bit higher. Initially, the difference values for all the three cases (e.g., RAM usage, CPU usage and execution time prediction) were approx. $\sim$ 4%, 13% & 8% respectively. So then, to reduce the error (i.e., cost function value) and build a more accurate prediction model, we adopted the *Multivariate Gradient Descent* algorithm to train the predictive functions of our ML model (i.e., more precisely, the theta (Θ) parameters of the hypothesis function ($h_\theta(x)$). After performing some iteration of training, we have seen that our training model can accurately predict the resource usage (e.g., RAM usage and CPU usage) and also the execution time for the F2C resources. In Fig. 7.7(a), 7.7(b), & 7.7(c), we represent the accuracy evaluation for our prediction model. We found that for all three cases (i.e., RAM usage, CPU usage and execution time) almost after hundred (100) iterations of training, our ML model is able to accurately perform and reduce all the three cost function values[2] (e.g., for RAM usage, CPU usage, execution time) to 1.8%, 7% & 4.6% respectively.

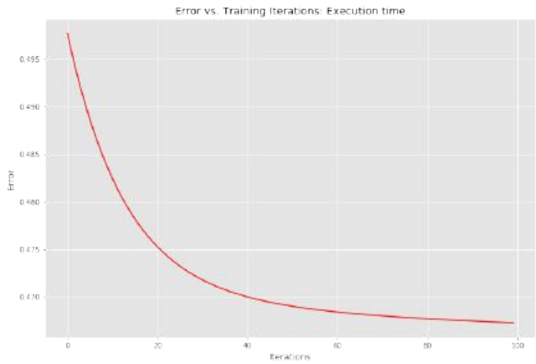
[2]https://github.com/resourceusageds/prediction



(a) RAM usage



(b) CPU usage



(c) Performance/Task execution time

Figure 7.7: Accuracy evaluation in terms of cost function value: Error vs Training iteration

By performing all these operations, we ensure the accuracy of our predicting model in the F2C paradigm. Most importantly, an accurate prediction helps to choose the best-fit resources for the upcoming tasks and also eventually helps to design a proper resource management mechanism for the F2C paradigm. Surprisingly, in our experiment, we have found the disk I/O information has not too much impact on predicting the execution time. We generated the raw dataset by executing a single type of application (e.g., image-recognition), so it might happen that, for executing this application the disk I/O is not so influencing for changing the task execution time. So, in future, it is necessary to execute other kinds of applications and investigate the correlational value between the task execution time and disk I/0. As this work is the initial step to building the prediction based resource management mechanism in the F2C paradigm, we assumed that measuring and predicting the network I/O is out of scope for this work.

7.3 Collaborative learning-based approach for forecasting resource usage and performance

One of the key features of any F2C-enabled smart system is to offer better utility experiences in case of latency-sensitive situation-aware services. For that purpose, it is essential to distribute intelligence capacity over the whole network of the F2C system. Thus, in the raised of sudden environmental events, the system can make smart decisions and adequate actions. Importantly, the intelligence of any system strengthens by learning from the previous observation and accurate prediction of the future. Notably, spreading intelligence over the whole network means engaging the participating system resources into the learning and prediction process with global coordination. That means, all the system resources can participate in this integrated process, and initially build their prediction model by processing over their local data. Then, the global prediction model builds by aggregating all these local models. Adopting this collaborative process not only helps to build a more accurate prediction model and reduce decision latency, but also ensure the data locality and privacy policies in any F2C-enabled smart system.

7.3.1 Architectural Description: Collaborative prediction mechanism in F2C

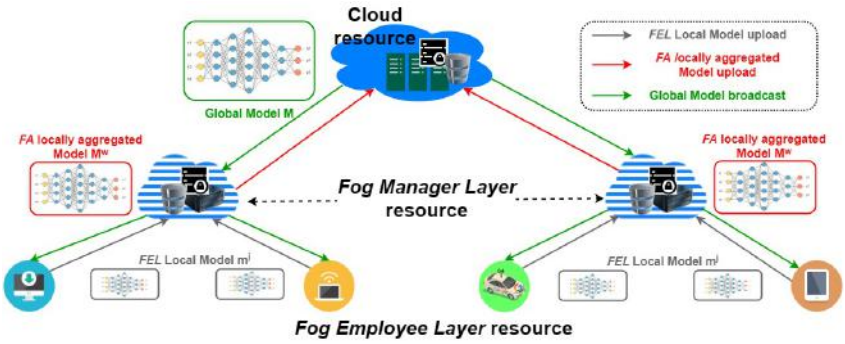


Figure 7.8: Collaborative learning procedure in F2C

Following any smart computing platform, it is easily recognizable that precisely two types of data are essential for executing some tasks and efficiently provide the various smart services. These two types of data are resources statistical data and collected data from various environmental events (e.g., sensing-data). The resource statistical data is sensitive for the system resources control, and sharing these data can disclose the privacy of such resources. Notably, continuous monitoring of the system resources and collection of the resource statistical data can explicitly help to manage the system resources adequately. Ultimately, that leads to building a useful resource allocation mechanism for executing some tasks and offer better services in the F2C-enabled smart system. On the other hand, the nature of continuously generated sensing-data is more open, and these data are one of the essential ingredients for any smart services. Most importantly, sharing and distributing this sensing-data over the network is mostly needed for running various applications to offer different latency-sensitive situation-aware services. For that reason, similarly, like our previous work [235], here we also deployed a secure, clustered and distributed database over the FML and cloud resources.

Resources statistical data is essential to build the prediction model for adequately managing the F2C system resources. However, sharing this information can breach current data-privacy policies (e.g., GDPR). Considering this fact, we adopt the collaborative ML concept [199] in our system. Fig. 7.8 depicts the incorporation of the collaborative ML technique in the F2C-enabled smart system. Initially, a global prediction model (M_0) has been built into the cloud layer. Then, the initial global model is broadcasted all over the network. The green lines in the figure denote the broadcasting of the global model. Then, every fog resources start processing the local instance of the global model with their own local and private data. After a given time (i), the locally updated model is uploaded to the upper layer for further aggregation to build the new global model. In Fig. 7.8, the red lines denote the FA model uploading process, and the grey lines suggest the FEL model uploading process. In the F2C-enabled smart system, aggregating of predicting model is a two-step process. In the first step, for a particular FA, all the locally updated prediction model of FEL resources (m_i^j) are aggregating with the locally updated model of corresponding FML resource to build the FA locally aggregated model (M_i^w). Whereas, in the second step, all M_i^w have been collected and aggregated to the cloud layer for making the new global prediction model (M_i). The algorithm 3 explains the process of collaborative ML technique in the F2C-enabled smart system. We have adopted the FedAvg technique [236] for aggregating the local models.

Algorithm 3 FedAvg-based model training and update process

Initial State: *generate parametrized global model* M_0

1: **procedure** UPDATED_PREDICTION_MODEL_CREATION

2: **Fog Area (FA) side:**

3: **foreach** FA, distribute the M_0 among FML resources

4: **end for**

5: **each** FA, consist of $\sum_{n=0}^{j} r_n$ no. of FEL resources

6: **foreach** FA, distribute the M_0 to its FEL resources

7: **foreach** FML & FEL resources, perform the local model update for i time

8: Generate local m_i^j, where j means no. of locally updated model by FEL resources

9: Upload m_i^j to the corresponding FML resource

10: **end for**

11: **end for**

12: **foreach** FA, FML resource aggregate all of its corresponding FEL resource's m_i^j along with its own updated model, then using FedAvg mechanism and generate M_i^w, where i denotes the given time for model update and w denotes the FA no.

13: **end for**

14: **foreach** FA, FML resource store its own aggregated model to the distributed DB cluster

15: **end for**

16: **Cloud-side**: make Query for all M_i^w

17: Further aggregate all M_i^w using FedAvg mechanism

18: Generate new global model M_i and replace old M_0

19: **repeat** From step 2 to 13 with updated global model (M_i), until accuracy reached to a certain threshold value

20: **end procedure**

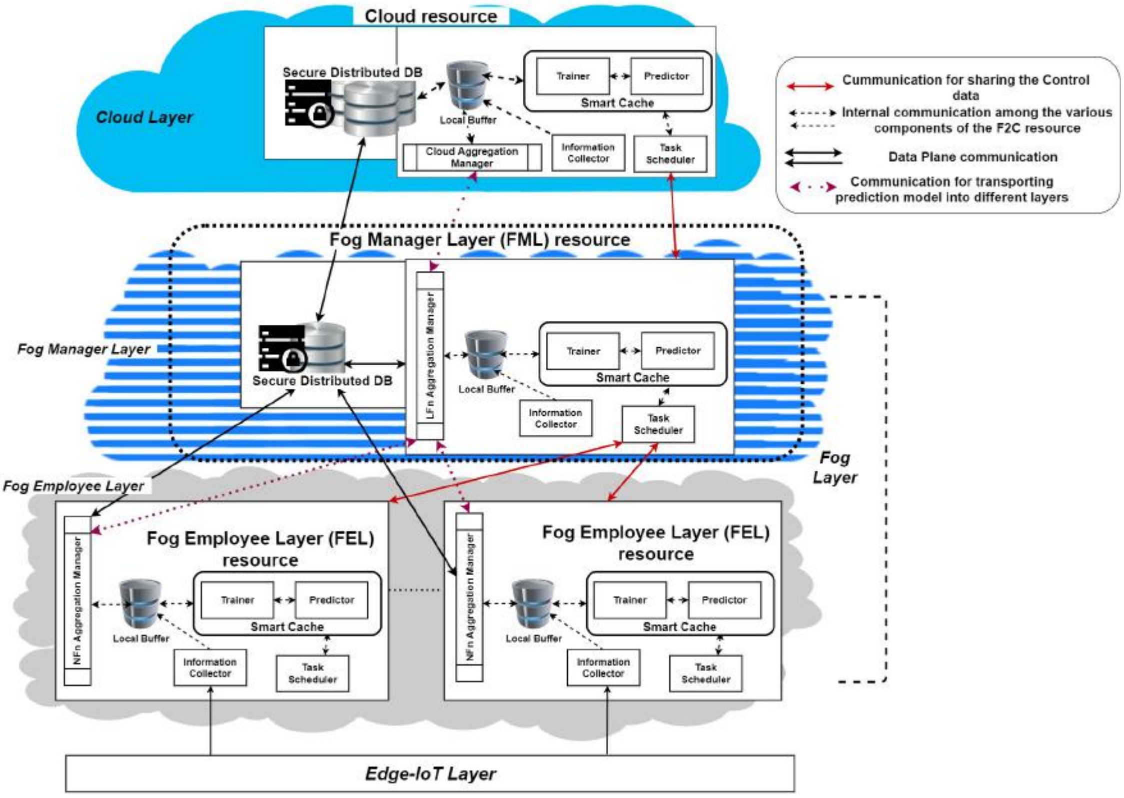


Figure 7.9: Architectural schema for Collaborative learning-based resource usage and performance prediction in F2C

Fig. 7.9 illustrates the collaborative learning-based resource management mechanism in the F2C-enabled smart system. It presents a detailed description of how the various components of the F2C system resource help in building the collaborative learning-based resource management mechanism. In our architecture, cloud resources are responsible for making the initial M_0. After, the creation of the M_0, it is being distributed among the various FML resources through the secure distributed database cluster. Once the model is shared, then it is further broadcasted to the FEL resources. After that, all the fog resources (FML and FEL resources) individually train the local copy of the global model in their Trainer component and locally test the accuracy level of the updated global model through the Predictor component. After a given time interval, all the local models are uploaded to upper layer resources for further aggregation in making of a new global model. Compositely these two components are known as Smart Cache. Based on the Predictor component's output, the Task Scheduler component can manage the upcoming tasks for all of our computational resources (either cloud and fog resources). Eventually, that helps to allocate the tasks into the best-fitted system resources for the execution. Notably, the Information Collector is helping the computing resources to separate their private and public data. Interestingly, for all the computing resources this component is also helping to temporary store the separated data into their local database (the Local Buffer). Besides that, all fog resources are also responsible for sending and storing the captured sensing-data, which is generated by the Edge-IoT devices (sensors). With the help of the Aggregation Manager component, computing resources can aggregate the monotonous sensing-data and helping to store them to the distributed database for further analysis. Even the Aggregation Manager component is responsible for transporting the prediction model between the various layer of resources. Notably, by thoroughly observing Fig. 7.9, it can be easily noticeable that we use different colours of arrows for denoting the various communication purpose between the different architectural components. Importantly, the brief description of their purposes has been described in that figure.

7.3.2 Performance Evaluation: For Collaborative learning-based approach

7.3.2.1 Testbed set-up:

We have built a prototype of the F2C-enabled smart city paradigm in our research lab and considered it as our testbed. In a real scenario, a vast number of citizen and organizations can associate to the F2C-enabled smart city, with their networking (e.g., routers, switches) and computing devices. Even a large number of FAs can emerge to serve the F2C facilities over the whole network efficiently. Following that, we built two FAs in that prototype of the F2C-enabled smart city. We consider each of the FA belongs to two different organizations. Also, all the exclusive FAs have the FML resource and a few FEL resources. Besides that, each FA consists of a massive number of Edge-IoT devices (sensor nodes). These Edge-IoT devices are mainly generating a large amount of sensing-data by continuously capturing of various environmental events (e.g., temperature, humidity, air pollution, etc.). Importantly, this sensing-data is also essential for offering various services (e.g., fire emergency, traffic monitoring, etc.) in the smart city domain. Gravely, this sensing-data might not disclose any private information (e.g., resource statistical data) of devices, and it can be shareable over the network.

We have hosted cloud resources on a server with Intel Xeon family E5-2620 V4 series (clock speed @3GHz), 96GB RAM, 1TB Hard Drive, and running on Ubuntu 16.04LTS Linux. All the FML resources are hosted in Raspberry Pi3 B+ models, running on Ubuntu 16.04LTS Linux and with Cortex A53 @ 1.4GHz processor, 1GB SD-RAM, and 64GB micro-SD card. Whereas, the FEL resources are mounted in Raspberry Pi Zero models, working with 1GHz single-core CPU, 512MB RAM and 8GB micro-SD storage. For storing the sensing-data and securely distributing it over the network, we have implemented a distributed database cluster over the FML and cloud resources. For that purpose, we are using the multi-datacenter, multi-node based containerized Apache Cassandra (Dockerized-Cassandra) cluster. To populate the real situation, we limit the bandwidth at the edge between 2-5 MBps, and the maximum connection speed between FML resources and the cloud layer resources is 11 MBps. A thorough observation of any smart city scenario shows that bandwidth utilization is one of the most critical issues to manage. Especially, limiting bandwidth poses a massive challenge for efficiently provide smart services in the smart city domain. Considering this fact, in our proposed schema, we tried to overcome this open issue. For that purpose, we performed some tests to measure the overall bandwidth utilization and data transmission time. To conduct these tests and measure the comparative performance between our proposed schema and centralized learning-based schema, we continuously generate some sensing-data packets for sending them to the distributed database cluster. Individually these data packets are almost 30kB in size and tagged with their generation timestamp. Measuring the packet transmitting time and overall bandwidth utilization implicitly helps to realize the load of the bandwidth. Notably, higher transmitting time and utilization represent the high bandwidth load, where lower utilization and transmitting time indicates the reduced bandwidth load. We use the Wireshark [237], an open-source network analyzer tool for measuring the bandwidth utilization of the whole system.

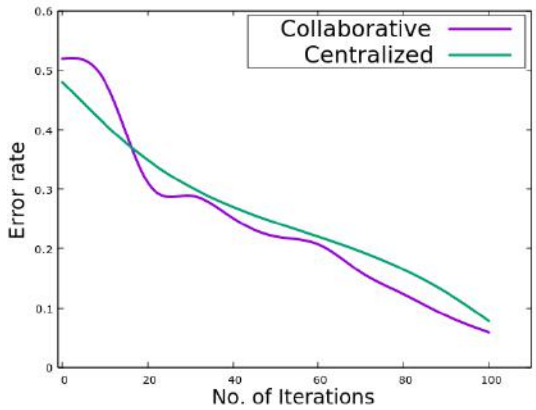
We have compared the accuracy of the ML technology in our proposed schema with respect to a centralized-learning schema. First, we run one simple image-classification app on every computational resource to generate our dataset[3]. Observing the issues for managing the system resources in our test scenario, we decided to use the supervised ML technique for both cases with multivariate linear regression with Stochastic Gradient Descent for optimizing the loss function [238] of prediction. Also, to successfully perform the ML technique, we take support from the Tensorflow framework [239].

7.3.2.2 Experimental results:

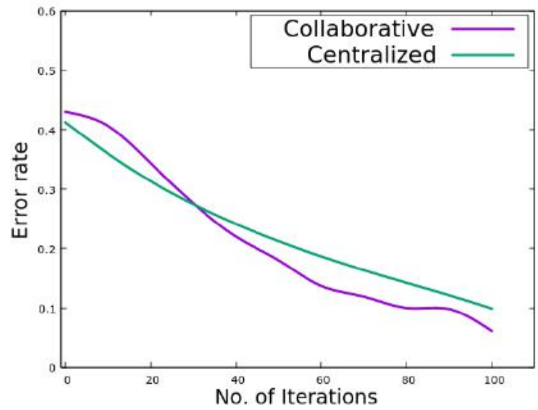
We perform two sets of different experimental and comparative study. In the first study, we perform the ML technique for predicting resource usage, considering CPU, Disk, RAM, and execution time for executing some tasks

[3]https://gitlab.com/data-snoopers/appload

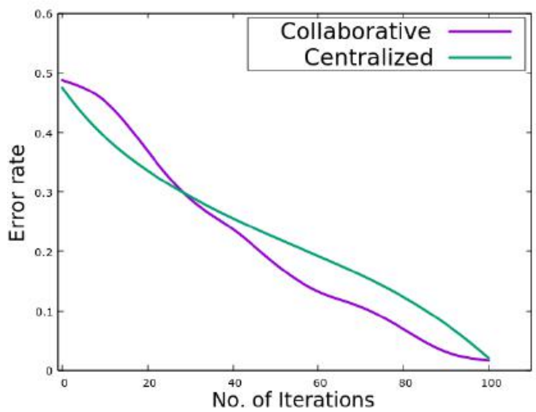
in both cases. Fig. 7.10(a), 7.10(b), 7.10(c), depicted the comparison study between our proposed collaborative learning-based schema and the centralized learning-based prediction schema for predicting the resource usage to executing some tasks. Whereas, in Fig. 7.10(d) presents the comparison of task execution-time for both of the cases. In those figures, the purple line presents the performance of the newly envisioned schema, and the green line presents the performance of the centralized learning-based schema.



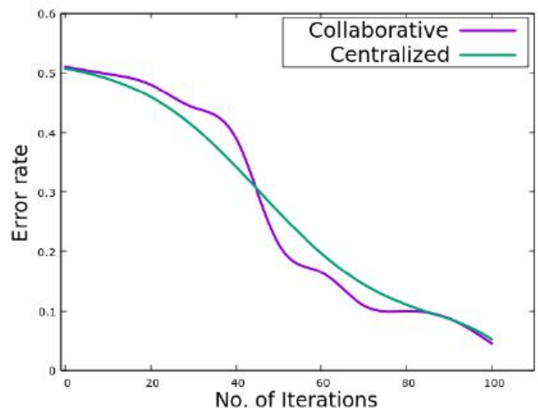
(a) CPU usage



(b) Disk usage



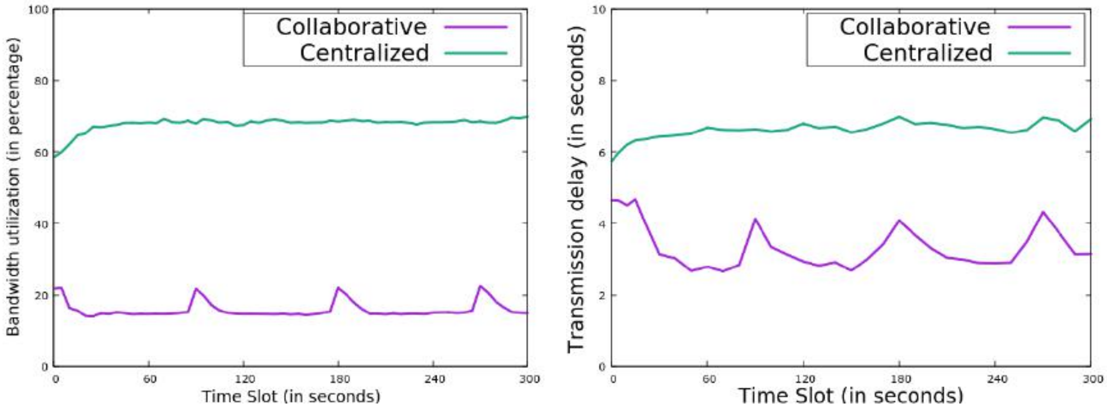
(c) RAM usage



(d) Execution time (Performance)

Figure 7.10: Comparison Study between Collaborative vs Centralized learning: ((a)-(d)) Prediction efficiency

In our proposed schema, we are not sending the resource statistical data in the centralized location. Instead of that, every individual computing nodes are responsible for learning and building their local model with the help of their own data. Finally aggregating those local model helping to build an accurate and useful global model. Notably, the whole process is required a few rounds of communication, that is known as training iteration for building the global prediction model in our proposed schema. However, for the centralized learning-based schema, all the data needed to be transferred into the centralized location. Then, the prediction model is trained and tested to a centralized location. Primarily, the centralized model is built by iterating over a large number of distinct data, so it has a higher accuracy level. Thus, initially, the centralized learning-based schema is outperforming over our proposed schema. However, after a few training iterations (approx. 30-40), the error rate of the prediction model for our proposed schema becomes slightly lower than the centralized learning-based schema. The reason behind the improvement is that, rather than sending the continuous resource statistical data, we are locally processing those data to build the local model. After that, we are sharing the local parametric information (local model) to build the global model. Therefore, in our schema, the chances of losing the resource statistical data due to the bandwidth limitation, are reduced. On the other hand, in the centralized schema, we are continuously sending the resource statistical information. That creates an extra burden for the network and increase the chances of data loss. Hence, some data loss can create some negative impact to reduce the accuracy level of the prediction model. Therefore, it justified the improvement in accuracy level of our proposed schema.



(a) Bandwidth Utilization

(b) Data transmission Delay

Figure 7.11: Comparison Study between Collaborative vs Centralized learning: ((a)-(b)) Network load

As earlier mentioned, in this proposed schema, we are not sending the resource statistical data to the database cluster. Therefore, the overall bandwidth utilization for our proposed schema remains lower than the centralized schema. Fig. 7.11(a) shows that the overall bandwidth utilization in our proposed schema has been reduced to one-third of the centralized schema. Besides that, continuously sending resource statistical data to the database cluster creates an extra burden for the network channel. Especially in the limited bandwidth arena (the edge), it creates an extra delay for sending the other data (sensing-data). Fig. 7.11(b), presents the comparison of data transmission delay for our proposed schema and the centralized learning-based schema. In both cases, we have considered the distance between the data source and the centralized location is identical. In these two figures, the purple line presents the performance of our proposed schema, where the green line presents the performance of the centralized learning-based schema. The spikes in the purple line represent the increased bandwidth utilization and transmission delay for the process of preparing the new global model and distributing it over the network. In our proposed schema, the data transmission delay is almost reduced to half of the centralized schema. Combining these two figures (7.11(a) and 7.11(b)), we can see that in our proposed schema, the network load has been considerably reduced. And finally, we have estimated the data volume reduction by keeping the resource statistical data locally. In this case, each node in the federated learning system performs its local learning, so they do not need to send the monitoring data, but need to send some parametric data of the new learning. We have estimated that in our proposed schema, around ninety-five per cent (95%) of resource information-related data flow has been reduced with respect to the centralized schema.

Notably, by accomplishing these works, we are able to take the initial steps for designing an advanced and sophisticated *Adaptive Resource Allocator Module* module to build an adaptive resource management framework in the F2C-enabled smart system. Initially, considering the architecture (i.e., Fig. 7.2) of F2C-enabled smart system (e.g. smart city), we realized that there are two ways for performing the ML operations in this system. Firstly, the ML can be performed in a logically centralized location (i.e., Fig. 7.3). Whereas, in another way (i.e., Fig. 7.9), all the available computing nodes in the system can participate in the ML operations and collaboratively performs. In this section, we have already highlighted these two possibilities. Our evaluation tests justified the several advantages of our proposals. We have also found some challenging issues in both of our proposed solutions. For examples, in collaborative learning mechanism, ensuring the fairness of the prediction model is a bit tricky and challenging job. Notably, in our proposed logically centralized learning mechanism, all the FML and Cloud resources are capable of performing the ML operations. Therefore, computation offloading and scheduling is an essential demand for balancing the work-load between various FML and Cloud resources. Solving these and many other open issues can constitute the foundation of future works.

Chapter 8: Conclusion & Future work

In this chapter, primarily, we summarized all of our contributions. Later on, we present all the remaining open issues, which are derived from the work developed in this thesis and based-on those issues, we sketched potential future lines of work which might be considered by other researchers.

8.1 Conclusion

The unprecedented development of Internet of Things (IoT) is turning our society into the digital world and connecting every small and big thing of our surroundings, driving us into an intelligent world. Certainly, the demand for ubiquitous computing has been growing up to build a truly smart computing paradigm. Thus, many new computing concepts have been developed and emerged. The primary motivation of all these developments is to build smarter and intelligent computing platform for improving the quality of lives. Notably, one of the key characteristics of any smart computing platform or system is to efficiently offer latency-sensitive situation-aware services among its consumers. Considering, any large-scale smart system (e.g. smart city), it can be easily realizable that efficient offering those services is a pretty challenging job. Especially when a large number of diverse and heterogeneous devices are participating in the system. For solving this issue, a new coordinated and combined computing architecture has been proposed by Masip-Bruin et al. [10], which is known as *Fog-to-Cloud (F2C)*. Significantly, F2C is in its infantile state; thus a proper resource utilization and handling is a necessary demand of this combined and coordinated computing architecture. Hence, considering this demand, in this dissertation, we performed several works for achieving our main objective to define and design the architectural framework, and it is corresponding architectural components (i.e., *Categorization Module, Resource Sharing Module, Information Collector Module, Adaptive Resource Allocator Module*) to manage the system resources adequately.

Considering the overall diversification of any F2C-enabled system, in this work, we proposed the taxonomic model of participating resources, as well as system-involved services and tasks. Later on, based on the resource taxonomic model, we designed the resource description models for organizing the system resources information, as well as it is helping us to understand the overall computation capacity of any F2C-enabled system easily. Hence, according to the proposed description models, in this dissertation work and mF2C project [20], we are collecting the resource information for creating the information pool. Notably, along with the resource information, the sensing-data generating a massive information pool in the F2C-enabled system. Typically, this information pool is playing a crucial role in the resource(s) allocation and task(s) execution process. Therefore, in this work, we designed and developed a secure, distributed database framework, which ensures, the secure distribution of the information pool over the network. Also, tasks execution in the F2C-enabled system is heavily related to the proper resource selection and allocation process. Adequate resource allocation can ensure the effective execution of the requested tasks and offer better services among the system's consumers. Undoubtedly, on-demand resource allocation in any F2C-enabled system is a complex and challenging job. Especially to build a more intelligent system, it is essential to reduce the unnecessary resource consumption in the overall system, without degrading the overall performance of the system. Hence, considering all of these aspects, in this work, we have proposed and developed an architectural module (i.e., *Adaptive Resource Allocator*) for effectively addressing the resource allocation related issues. Importantly, in our proposal, Machine Learning (ML) technique is playing a pivotal role in priorly select the appropriate resource(s) for executing some task(s). Nonetheless, the ML technique can be implemented for the F2C-enabled system in two ways: centralized and distributed. Thus, in this thesis work, we have also proposed and developed two different architectural schemas for performing the ML-based resource allocation procedure in F2C-enabled system. Typically, in the first approach, the ML technique is performing over the cluster-based logically centralized location, where the information is securely distributed. Significantly, in our second approach, taking into account the data-privacy concerns and following the F2C-enabled system, we adopted the collaborative ML technique for performing the resource allocation procedure in a distributed way. As a summary, the contributions of the dissertation are as follows:

1. *Design of adaptive resource management framework:* In this dissertation, primarily to address our main objective, by thoroughly investigating on several related works, we proposed the initial and the basic architectural structure of sophisticated and adaptive resource management mechanism in any F2C-enabled system. Our work is the initial footsteps for making the advanced solution of distributed resource management in any F2C-enabled system.

   Outcome:

   - S. Sengupta, J. Garcia, and X. Masip-Bruin, "An Architecture for Resource Management in a Fog-to-Cloud Framework", in 24th International European Conference on Parallel and Distributed Computing (Euro-Par 2018), F2C-DP Workshop, Springer, 2018, pp. 275-286.

2. *Characterization-based taxonomic model:* By addressing the characterization related issues, we proposed the taxonomic model of participating system resources as well as system-involved services and tasks.

   Outcomes:

   - S. Sengupta, J. Garcia, and X. Masip-Bruin, "A Literature Survey on Ontology of Different Computing Platforms in Smart Environments", in the Research reports management application of the Computer Architecture Department, Universitat Politècnica de Catalunya (UPC BarcelonaTech).
   - S. Sengupta, J. Garcia, and X. Masip-Bruin, "Essentiality of Resource and Service-Task Characterization in the Coordinated Fog-to-Cloud Paradigm", in 2018 International Conference on Smart Communications in Network Technologies (SaCoNeT), IEEE, 2018, pp. 249-254.

3. *Taxonomy-based resource description model:* Later, considering the defined resource taxonomic model, we have designed the generalized and aggregated resource description model of F2C resources.

   Outcome:

   - S. Sengupta, J. Garcia, and X. Masip-Bruin, "Taxonomy and Resource Modeling in Combined Fog-to-Cloud Systems", in proceedings of the Future Technologies Conference, Springer, 2018, pp. 687-704.

4. *Build-up a secure and distributed information pool over the network:* Importantly, for making an efficient resource management mechanism is necessarily required the resource information pool. Also, for successfully offering various services (e.g., e-health service, traffic monitoring service, etc.) the system needs the captured sensing-data. Thus, following any F2C-enabled smart system, we understand that a massive amount of data is flowing all over the system. Necessarily, this data needs to be protected and effectively distributed over the system. So that, in-requirement, the data can be processed fast and offer the services in a quick time. Considering that fact, we have already proposed and designed a secure distributed database management framework over the network for creating an information pool. Notably, with the help of our proposed distributed database framework in the F2C-enabled system, we can achieve approximately thirty-three per cent (33%) and forty-four per cent (44%) better bandwidth utilization, compared to the traditional cloud-based and fog/edge-based smart system, respectively. Also, with the help of our proposal in F2C-enabled system, we achieved three (3) times faster query response time compared to the tradition cloud-based system.

Outcomes:

- S. Sengupta, J. Garcia, and X. Masip-Bruin, "Essentiality of Managing the Resource Information in the Coordinated Fog-to-Cloud Paradigm", International Journal of Communication Systems, 2019, vol. 33, no. 10: e4286.
- S. Sengupta, S. Kahvazadeh, X. Masip-Bruin, and J. Garcia, “SFDDM: A Secure Distributed Database Management in Combined Fog-to-Cloud Systems”, in 2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks(CAMAD), IEEE, 2019, pp. 1–7.

5. *Construction of adaptive resource allocation:* The F2C is blooming to ensure better service facilities in any smart system by efficiently offering the latency-sensitive situation-aware services among the system’s consumers. Also, another vital characteristic of the F2C is to bring intelligence or smartness near to the verge of the network. Notably, bringing the smartness near to the edge and distributing it over the network, implicitly helps to utilize the F2C resources properly. Especially, prior forecasting of their resource usage (i.e., RAM usage, CPU usage, Power usage, Disk usage, Bandwidth usage, etc.) and performance (i.e., in terms of task execution time) helps to choose the appropriate F2C resources for executing some tasks and offer some services among the system consumers. Hence, Machine Learning (ML) is playing a crucial role in allocating F2C resources. Considering this fact, in this thesis, we have adopted the ML techniques for designing and adaptive and sophisticated resource allocation mechanism. Following the F2C computing architecture in the smart city scenario, we realized that there are two ways that the ML techniques can be implemented in any F2C-enabled smart system (e.g., smart city). Either the ML can be performed in the logically centralized location, or it can be performed in a distributed fashion. So, considering all these aspects, in this thesis, we have already proposed two different architectural solutions for implementing the ML techniques in any F2C-enabled smart system, to achieve the distributed resource management mechanism. In our first proposal, we design an architectural schema, where the ML technique is performing over the secure distributed database cluster. To simplify it, we called this as *Cluster-based logically centralized approach* for allocating the system resources. Notably, in our second proposal, we design such an architectural framework, where the ML technique can be performed in a distributed way. Taking into account the data-privacy concerns and following the F2C computing architecture, we realized the demand for implementing the collaborative ML technique in our considering system. Thus we also, design and develop another architectural schema, which is named as *Collaborative learning-based approach* for allocating the system resources. Importantly, implementing the collaborative ML technique ensures not only the data-privacy aspect but also it reduces the overall bandwidth consumption by one-third, and also reduces the data transmission delay by half, compared to the cluster-based logically centralized approach.

Outcomes:

- S. Sengupta, J. Garcia, X. Masip-Bruin, and A. Prieto-González, "An Architectural Schema for Performance Prediction using Machine Learning in the Fog-to-Cloud Paradigm", in 2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON 2019), IEEE, 2019, pp. 0994-1002.
- S. Sengupta, J. Garcia, and X. Masip-Bruin, “Collaborative learning-based Schema for Predicting Resource Usage and Performance in F2C Paradigm”, submitted in 2020 IEEE Global Communications Conference.

8.2 Future opportunities

In this section, we sketch remaining open challenges derived from the work developed in this dissertation and mark most of the potential future opportunities of work which might be considered by other researchers. For that purpose, we classified all the potential opportunities into a few categories. Next, in the section, by identifying those challenges, we tried to provide some guidelines for future works.

Opportunities to work on resource sharing related issues: Virtualization is one of the most significant and demanding features for many modern computing platforms (e.g., cloud computing). By virtualizing the system resources and infrastructure, many computing platforms offer higher service facilities among its subscribers. Notably, virtualization allows a single physical instance of a resource or an application to be shared among its subscribers, based on some certain partitioning policies [240], [241]. Importantly, following the F2C-enabled system and considering the massive diversity of the system resources, we realized that, in any F2C-enabled system, resource sharing is not an easy task.

In any F2C-enabled system, multiple devices from different participants (i.e., individuals or organizations) can join the system for offering their idle resources. Importantly, for enabling this facility in any F2C-enabled system, it is essential to have a proper global business model and specific policies for sharing the available idle resources in the system. Eventually, working on these issues can potentially lead to building a more advanced and sophisticated resource sharing mechanism in any F2C-enabled system.

Opportunities to work on policy related issues: In this thesis work, we have not given too much concern for defining the global policy model to build the resource management mechanism in F2C-enabled system. Notably, policies are playing a crucial role in service-oriented computing platform for efficiently utilizing the system resources [242], [243]. Policies have a considerable impact on defining resource sharing and partitioning [244]. Also, based on the enforced system policies, data can be distributed over the system as well as it can be accessed for further processing [245]. Even depending on the system policies, services and tasks can be prioritized and scheduled [246]. Also, policies have the greatest influence for setting the security and privacy functionalities of the systems [164]. Thus policies have a significant impact on resource utilization and management techniques. Therefore, it is essential to define a global policy module in the F2C system for efficiently managing the participating devices.

Opportunities to work on resource selection procedure: In our work, for selecting the appropriate F2C resources, first, we calculate the rank of them based on the matching score [185] and global cost model [225]. Notably, for calculating the rank of the F2C resources, we have adopted the TOPSIS methodology [226]. Then based on the calculated rank, we allocate the best-fitted F2C resources for executing some tasks. As our primary motivation was to define and build the overall architectural structure of the resource management mechanism in the F2C-enabled system; therefore, we have adopted those aforementioned techniques for allocating the F2C resources. Importantly, in the F2C-enabled system, various devices are participating by contributing their resources for executing some tasks and offer some services. Significantly, some individuals own some of the devices, and also some organizations own some of the other devices. Unfortunately, there is no such business model (or, cost model) available in the F2C system. Most importantly, F2C is a coordinated and combined computing platform which has been emerging by the integration of cloud, fog/edge, and IoT facilities. Therefore, we realized that a unified business model (or, cost model) is necessarily required.

Also, in this work for the allocation of F2C resources, we adopted the TOPSIS method. Typically, the traditional TOPSIS method has some flaws. Notably, the traditional TOPSIS method work in the finite set of alternatives and have several disadvantages: correlations between criteria, uncertainty in obtaining the ranks (weights) only by objective methods or subjective methods and the possibility of alternative closed to the ideal point and nadir point concurrently [247]. Also, TOPSIS does not provide for rank (weight) elicitation, and consistency checking for judgments [248]. Significantly, in the real scenario, any F2C-enabled system is massively dynamic. Therefore, calculating the rank of F2C resources by TOPSIS methodology could be challenging. Thus, it is necessary to further investigating on this topic.

Opportunities to work on forecasting mechanism: In order to fulfil our objectives and build an adaptive resource utilization and management mechanism, we have adopted the ML techniques. Following the architecture of F2C-enabled system, we realized that in F2C, ML techniques could be performed in two ways: either in the logically centralized location or in the distributed locations. Therefore, we have given our focus on identifying the functional schema for both of these two ways. Significantly, data is the main ingredient for the ML techniques. By processing over the data, the prediction model can be trained to perform the forecasting. Therefore, preparing data is the essential operation, that needs to be performed to efficiently execute the ML techniques in the F2C-enabled system. Significantly, in the proposed distributed ML-based resource management mechanism, all the computing devices in the F2C system are participating in building their own prediction model by processing over their local data and with the help of global coordination. Notably, after building the local prediction models, all of the models are collected and aggregated in a centralized location for making the global prediction model. Importantly, processing over the tampered or corrupted data can lead to building some unfair local models. Typically aggregating the unfair local models can also build an unfair global model. Eventually, that creates a massive negative impact over the system and in-total it would degrade the performance of the system. Hence, tackling the prediction model's fairness issue in the F2C-enabled system is an important and challenging job. However, in future, welling researchers can focus on to solve this issue for building more sophisticated, adequate, advanced and intelligent resource management mechanism in any F2C-enabled system.

Besides all of these issues, mobility is also a key concerning issue, that needs to be addressed. Most importantly, we consider that addressing the mobility of the devices is out of the scope of our work. However, in the real scenario, a massive number of mobile computational devices can participate in any F2C-enabled smart system (e.g. smart city). Efficiently utilizing them for executing some tasks and provide some services is a vital issue. Apart from that, many other open issues (i.e., security-related), that can be addressed in the future work for improving our proposed resource management architecture.

References

[1] H. Wallach, “Computational social science”, *Computational Social Science*, vol. 307, 2016.

[2] A. H. Bartels, E Daley, A Parker, B Evelson, and C Muteba, “Smart computing drives the new era of it growth”, *Forrester Inc*, 2009.

[3] D. Cook and S. K. Das, *Smart environments: Technology, protocols and applications*. John Wiley & Sons, 2004, vol. 43.

[4] J. Pan and J. McElhannon, “Future edge cloud and edge computing for internet of things applications”, *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 439–449, 2018.

[5] O. B. Sezer, E. Dogdu, and A. M. Ozbayoglu, “Context-aware computing, learning, and big data in internet of things: A survey”, *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 1–27, 2018.

[6] E.-S. Cho, T.-S. Yoon, J.-H. Choi, J.-Y. Paik, and S. Helal, “An integrated formal model for context-aware systems”, in *2013 IEEE 37th Annual Computer Software and Applications Conference Workshops*, IEEE, 2013, pp. 163–168.

[7] M. Maulana, “An internet of things framework for smart energy in buildings: Designs, prototype, and experiments”, *Buletin Inovasi ICT & Ilmu Komputer*, vol. 2, no. 1, 2015.

[8] J. Jin, J. Gubbi, S. Marusic, and M. Palaniswami, “An information framework for creating a smart city through internet of things”, *IEEE Internet of Things journal*, vol. 1, no. 2, pp. 112–121, 2014.

[9] S. Chun, S. Shin, S. Seo, S. Eom, J. Jung, and K.-H. Lee, “A pub/sub-based fog computing architecture for internet-of-vehicles”, in *2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, IEEE, 2016, pp. 90–93.

[10] X. Masip-Bruin, E. Marín-Tordera, G. Tashakor, A. Jukan, and G.-J. Ren, “Foggy clouds and cloudy fogs: A real need for coordinated management of fog-to-cloud computing systems”, *IEEE Wireless Communications*, vol. 23, no. 5, 2016.

[11] R. Pakdel and J. Herbert, “Adaptive cost efficient framework for cloud-based machine learning”, in *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, IEEE, vol. 2, 2017, pp. 155–160.

[12] C.-C. Lin and J.-W. Yang, “Cost-efficient deployment of fog computing systems at logistics centers in industry 4.0”, *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4603–4611, 2018.

[13] J. He, J. Wei, K. Chen, Z. Tang, Y. Zhou, and Y. Zhang, “Multitier fog computing with large-scale iot data analytics for smart cities”, *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 677–686, 2017.

[14] F. Xu, H. Zheng, H. Jiang, W. Shao, H. Liu, and Z. Zhou, “Cost-effective cloud server provisioning for predictable performance of big data analytics”, *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 5, pp. 1036–1051, 2018.

[15] Z. Li, T. N. Hung, S. Lu, and R. S. M. Goh, “Performance and monetary cost of large-scale distributed graph processing on amazon cloud”, in *2016 International Conference on Cloud Computing Research and Innovations (ICCCRI)*, IEEE, 2016, pp. 9–16.

[16] J. Shao, Q. Qiao, Y. Xu, S. Zhai, and H. Liu, “Operation and maintenance costs allocation model of grid project based on performance value”, in *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, IEEE, 2019, pp. 1809–1812.

[17] J. Emeras, S. Varrette, V. Plugaru, and P. Bouvry, “Amazon elastic compute cloud (ec2) vs. in-house hpc platform: A cost analysis”, *IEEE Transactions on Cloud Computing*, 2016.

[18] Ericsson, *Internet of things forecast (report)*, https://www.ericsson.com/en/mobility-report/internet-of-things-forecast, 2016.

[19] R. Sosa, R. Sucasas, A. S. ENG, A. Leckey, C. Cordeiro, J. Jensen, C. Ketley, S. Crompton, and F. Carpio, “D5. 1 mf2c reference architecture (integration it-1)”,

[20] C. Cordeiro, S. Stanovnik, M. Cankar, J. Jensen, E. M. T. UPC, D. G. WSL, A. Salis, R. Bulla, P. C. ENG, A. Queralt, *et al.*, “D5. 2 mf2c reference architecture (integration it-2)”,

[21] X. Masip-Bruin, E. Marín-Tordera, A. Juan-Ferrer, A. Queralt, A. Jukan, J. Garcia, D. Lezzi, J. Jensen, C. Cordeiro, A. Leckey, *et al.*, “Mf2c: Towards a coordinated management of the iot-fog-cloud continuum”, in *Proceedings of the 4th ACM MobiHoc Workshop on Experiences with the Design and Implementation of Smart Objects*, ACM, 2018, p. 8.

[22] A. Regalado, “Who coined ‘cloud computing’”, *Technology Review*, vol. 31, 2011.

[23] F. Liu, J. Tong, J. Mao, R. Bohn, J. Messina, L. Badger, and D. Leaf, “Nist cloud computing reference architecture”, *NIST special publication*, vol. 500, no. 2011, p. 292, 2011.

[24] P. Mell, T. Grance, *et al.*, “The nist definition of cloud computing”, 2011.

[25] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, “Fog computing and its role in the internet of things”, in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, ACM, 2012, pp. 13–16.

[26] A. Yousefpour, C. Fung, T. Nguyen, K. Kadiyala, F. Jalali, A. Niakanlahiji, J. Kong, and J. P. Jue, “All one needs to know about fog computing and related edge computing paradigms: A complete survey”, *Journal of Systems Architecture*, 2019.

[27] D. Linthicum. (2019). Edge computing vs. fog computing: Definitions and enterprise uses, [Online]. Available: https://www.cisco.com/c/en/us/solutions/enterprise-networks/edge-computing.html.

[28] M. Chiang, S. Ha, I Chih-Lin, F. Risso, and T. Zhang, “Clarifying fog computing and networking: 10 questions and answers”, *IEEE Communications Magazine*, vol. 55, no. 4, pp. 18–20, 2017.

[29] A. Čolaković and M. Hadžialić, “Internet of things (iot): A review of enabling technologies, challenges, and open research issues”, *Computer Networks*, vol. 144, pp. 17–39, 2018.

[30] S. Madakam, V. Lake, V. Lake, V. Lake, *et al.*, “Internet of things (iot): A literature review”, *Journal of Computer and Communications*, vol. 3, no. 05, p. 164, 2015.

[31] S.-C. Wang, K.-Q. Yan, S.-S. Wang, and C.-W. Chen, "A three-phases scheduling in a hierarchical cloud computing network", in *2011 Third International Conference on Communications and Mobile Computing*, IEEE, 2011, pp. 114–117.

[32] R. Yu, Y. Zhang, S. Gjessing, W. Xia, and K. Yang, "Toward cloud-based vehicular networks with efficient resource management", *IEEE Network*, vol. 27, no. 5, pp. 48–55, 2013.

[33] M. Shirer and C. MacGillivray. (2019). The growth in connected iot devices is expected to generate 79.4zb of data in 2025, according to a new idc forecast, [Online]. Available: https://www.idc.com/getdoc.jsp?containerId=prUS45213219.

[34] E. Arianyan, H. Taheri, and S. Sharifian, "Novel energy and sla efficient resource management heuristics for consolidation of virtual machines in cloud data centers", *Computers & Electrical Engineering*, vol. 47, pp. 222–240, 2015.

[35] R. Buyya and A. V. Dastjerdi, *Internet of Things: Principles and paradigms*. Elsevier, 2016.

[36] A. Chowdhury and S. A. Raut, "A survey study on internet of things resource management", *Journal of Network and Computer Applications*, vol. 120, pp. 42–60, 2018.

[37] M. Kumar, S. Sharma, A. Goel, and S. Singh, "A comprehensive survey for scheduling techniques in cloud computing", *Journal of Network and Computer Applications*, 2019.

[38] S. Singh and I. Chana, "Qos-aware autonomic resource management in cloud computing: A systematic review", *ACM Computing Surveys (CSUR)*, vol. 48, no. 3, p. 42, 2016.

[39] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective", *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[40] G. Tanganelli, C. Vallati, and E. Mingozzi, "Edge-centric distributed discovery and access in the internet of things", *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 425–438, 2017.

[41] T. Graves, "The service-oriented enterprise", *Colchester: Tetradian Books*, 2009.

[42] L. Ni, J. Zhang, C. Jiang, C. Yan, and K. Yu, "Resource allocation strategy in fog computing based on priced timed petri nets", *IEEE Internet of Things Journal*, vol. 4, no. 5, 2017.

[43] S. Mustafa, B. Nazir, A. Hayat, S. A. Madani, *et al.*, "Resource management in cloud computing: Taxonomy, prospects, and challenges", *Computers & Electrical Engineering*, vol. 47, pp. 186–203, 2015.

[44] M.-A. Vasile, F. Pop, R.-I. Tutueanu, V. Cristea, and J. Kołodziej, "Resource-aware hybrid scheduling algorithm in heterogeneous distributed computing", *Future Generation Computer Systems*, vol. 51, pp. 61–71, 2015.

[45] L. Sanchez, L. Muñoz, J. A. Galache, P. Sotres, J. R. Santana, V. Gutierrez, R. Ramdhany, A. Gluhak, S. Krco, E. Theodoridis, *et al.*, "Smartsantander: Iot experimentation over a smart city testbed", *Computer Networks*, vol. 61, pp. 217–238, 2014.

[46] S. Singh and I. Chana, "Cloud resource provisioning: Survey, status and future research directions", *Knowledge and Information Systems*, vol. 49, no. 3, pp. 1005–1069, 2016.

[47] N. Kherraf, H. A. Alameddine, S. Sharafeddine, C. M. Assi, and A. Ghrayeb, "Optimized provisioning of edge computing resources with heterogeneous workload in iot networks", *IEEE Transactions on Network and Service Management*, vol. 16, no. 2, pp. 459–474, 2019.

[48] M. A. Salahuddin, A. Al-Fuqaha, and M. Guizani, "Reinforcement learning for resource provisioning in the vehicular cloud", *IEEE Wireless Communications*, vol. 23, no. 4, pp. 128–135, 2016.

[49] S. Prakash *et al.*, "Role of virtualization techniques in cloud computing environment", in *Advances in Computer Communication and Computational Sciences*, Springer, 2019, pp. 439–450.

[50] X. Liu, B. Cheng, Y. Yue, M. Wang, B. Li, and J. Chen, "Traffic-aware and reliability-guaranteed virtual machine placement optimization in cloud datacenters", in *2019 IEEE 12th International Conference on Cloud Computing (CLOUD)*, IEEE, 2019, pp. 91–98.

[51] M. Goudarzi, H. Wu, M. S. Palaniswami, and R. Buyya, "An application placement technique for concurrent iot applications in edge and fog computing environments", *IEEE Transactions on Mobile Computing*, 2020.

[52] G. Li, J. Wu, J. Li, K. Wang, and T. Ye, "Service popularity-based smart resources partitioning for fog computing-enabled industrial internet of things", *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4702–4711, 2018.

[53] S. Goyal, S. Bawa, and B. Singh, "Energy optimised resource scheduling algorithm for private cloud computing", *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 23, no. 1-2, pp. 115–123, 2016.

[54] M. Xu, A. V. Dastjerdi, and R. Buyya, "Energy efficient scheduling of cloud application components with brownout", *IEEE Transactions on Sustainable Computing*, vol. 1, no. 2, pp. 40–53, 2016.

[55] X. Wang, K. Wang, S. Wu, S. Di, H. Jin, K. Yang, and S. Ou, "Dynamic resource scheduling in mobile edge cloud with cloud radio access network", *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 11, pp. 2429–2445, 2018.

[56] E. Cortez, A. Bonde, A. Muzio, M. Russinovich, M. Fontoura, and R. Bianchini, "Resource central: Understanding and predicting workloads for improved resource management in large cloud platforms", in *Proceedings of the 26th Symposium on Operating Systems Principles*, 2017, pp. 153–167.

[57] S. Sengupta, J. Garcia, and X. Masip-Bruin, "A literature survey on ontology of different computing platforms in smart environments", *arXiv preprint arXiv:1803.00087*, 2018.

[58] F. Cicirelli, A. Guerrieri, G. Spezzano, and A. Vinci, "An edge-based platform for dynamic smart city applications", *Future Generation Computer Systems*, vol. 76, pp. 106–118, 2017.

[59] F. Cicirelli, G. Fortino, A. Guerrieri, G. Spezzano, and A. Vinci, "Metamodeling of smart environments: From design to implementation", *Advanced Engineering Informatics*, vol. 33, pp. 274–284, 2017.

[60] O. Consortium *et al.*, "Openfog reference architecture for fog computing", *Architecture Working Group*, 2017.

[61] EdgeX-Foundary, *Edgex project documentation*, https://www.edgexfoundry.org/wp-content/uploads/sites/25/2017/12/EdgeX-Introduction-Full-12.28.17.pdf, 2018.

[62] AMASS, *Amass project deliverables*, https://www.amass-ecsel.eu/content/deliverables, 2016.

[63] ONF, *Project documentation of the open networking foundation (onf)*, https://www.opennetworking.org/reference-designs/, 2018.

[64] ERIgrid, *Project documentation and deliverables of the erigrid*, https://erigrid.eu/dissemination/, 2014.

[65] SOSCIP, *Project information of the soscip*, https://www.soscip.org/projects/, 2012.

[66] J. Swetina, G. Lu, P. Jacobs, F. Ennesser, and J. Song, "Toward a standardized common m2m service layer platform: Introduction to onem2m", *IEEE Wireless Communications*, vol. 21, no. 3, pp. 20–26, 2014.

[67] M. Compton, P. Barnaghi, L. Bermudez, R. GarcíA-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, A. Herzog, *et al.*, "The ssn ontology of the w3c semantic sensor network incubator group", *Web semantics: science, services and agents on the World Wide Web*, vol. 17, pp. 25–32, 2012.

[68] E. Arianyan, M. R. Ahmadi, and D. Maleki, "A novel taxonomy and comparison method for ranking cloud computing software products", *International Journal of Grid and Distributed Computing*, vol. 9, no. 3, pp. 173–190, 2016.

[69] D. Weerasiri, M. C. Barukh, B. Benatallah, Q. Z. Sheng, and R. Ranjan, "A taxonomy and survey of cloud resource orchestration techniques", *ACM Computing Surveys (CSUR)*, vol. 50, no. 2, p. 26, 2017.

[70] L. Youseff, M. Butrico, and D. Da Silva, "Toward a unified ontology of cloud computing", in *Grid Computing Environments Workshop, 2008. GCE'08*, IEEE, 2008, pp. 1–10.

[71] C. Pittaras, M. Ghijsen, A. Wibisono, P. Grosso, J. Van Der Ham, and C. de Laat, "Semantic distributed resource discovery for multiple resource providers", in *Semantics, Knowledge and Grids (SKG), 2012 Eighth International Conference on*, IEEE, 2012, pp. 225–228.

[72] M. Zhang, R. Ranjan, A. Haller, D. Georgakopoulos, M. Menzel, and S. Nepal, "An ontology-based system for cloud infrastructure services' discovery", in *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2012 8th International Conference on*, IEEE, 2012, pp. 524–530.

[73] N. Fernando, S. W. Loke, and W. Rahayu, "Mobile cloud computing: A survey", *Future generation computer systems*, vol. 29, no. 1, pp. 84–106, 2013.

[74] S. Singh and I. Chana, "A survey on resource scheduling in cloud computing: Issues and challenges", *Journal of grid computing*, vol. 14, no. 2, pp. 217–264, 2016.

[75] B. Jennings and R. Stadler, "Resource management in clouds: Survey and research challenges", *Journal of Network and Systems Management*, vol. 23, no. 3, pp. 567–619, 2015.

[76] F. Moscato, R. Aversa, B. Di Martino, T.-F. Fortiş, and V. Munteanu, "An analysis of mosaic ontology for cloud resources annotation", in *Computer Science and Information Systems (FedCSIS), 2011 Federated Conference on*, IEEE, 2011, pp. 973–980.

[77] K. Yongsiriwit, M. Sellami, and W. Gaaloul, "A semantic framework supporting cloud resource descriptions interoperability", in *2016 IEEE 9th International Conference on Cloud Computing (CLOUD)*, IEEE, 2016, pp. 585–592.

[78] K. Tutschku, V. A. Mehri, A. Carlsson, K. V. Chivukula, and J. Christenson, "On resource description capabilities of on-board tools for resource management in cloud networking and nfv infrastructures", in *2016 IEEE International Conference on Communications Workshops (ICC)*, IEEE, 2016, pp. 442–447.

[79] S. M. Parikh, N. M. Patel, and H. B. Prajapati, "Resource management in cloud computing: Classification and taxonomy", *arXiv preprint arXiv:1703.00374*, 2017.

[80] A. Botta, W. De Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and internet of things: A survey", *Future Generation Computer Systems*, vol. 56, pp. 684–700, 2016.

[81] Z. Á. Mann, "Allocation of virtual machines in cloud data centers—a survey of problem models and optimization algorithms", *Acm Computing Surveys (CSUR)*, vol. 48, no. 1, p. 11, 2015.

[82] B. Di Martino, G. Cretella, A. Esposito, A Willner, A Alloush, D. Bernstein, D. Vij, and J Weinman, "Towards an ontology-based intercloud resource catalogue–the ieee p2302 intercloud approach for a semantic resource exchange", in *2015 IEEE International Conference on Cloud Engineering*, IEEE, 2015, pp. 458–464.

[83] E. Hachicha and W. Gaaloul, "Towards resource-aware business process development in the cloud", in *2015 IEEE 29th International Conference on Advanced Information Networking and Applications*, IEEE, 2015, pp. 761–768.

[84] W. Shi, L. Zhang, C. Wu, Z. Li, and F. Lau, "An online auction framework for dynamic resource provisioning in cloud computing", *ACM SIGMETRICS Performance Evaluation Review*, vol. 42, no. 1, pp. 71–83, 2014.

[85] M. Mishra, A. Das, P. Kulkarni, and A. Sahoo, "Dynamic resource management using virtual machine migrations", *IEEE Communications Magazine*, vol. 50, no. 9, pp. 34–40, 2012.

[86] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, "Fog computing: A platform for internet of things and analytics", in *Big data and internet of things: A roadmap for smart environments*, Springer, 2014, pp. 169–186.

[87] A. Munir, P. Kansakar, and S. U. Khan, "Ifciot: Integrated fog cloud iot: A novel architectural paradigm for the future internet of things.", *IEEE Consumer Electronics Magazine*, vol. 6, no. 3, pp. 74–82, 2017.

[88] M. H. Syed, E. B. Fernandez, and M. Ilyas, "A pattern for fog computing", in *Proceedings of the 10th Travelling Conference on Pattern Languages of Programs*, ACM, 2016, p. 13.

[89] E. Baccarelli, P. G. V. Naranjo, M. Scarpiniti, M. Shojafar, and J. H. Abawajy, “Fog of everything: Energy-efficient networked computing architectures, research challenges, and a case study”, *IEEE access*, vol. 5, pp. 9882–9910, 2017.

[90] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, “Vehicular fog computing: A viewpoint of vehicles as the infrastructures”, *IEEE Transactions on Vehicular Technology*, vol. 65, no. 6, pp. 3860–3873, 2016.

[91] R. Mahmud, R. Kotagiri, and R. Buyya, “Fog computing: A taxonomy, survey and future directions”, in *Internet of everything*, Springer, 2018, pp. 103–130.

[92] C. Perera, Y. Qin, J. C. Estrella, S. Reiff-Marganiec, and A. V. Vasilakos, “Fog computing for sustainable smart cities: A survey”, *ACM Computing Surveys (CSUR)*, vol. 50, no. 3, p. 32, 2017.

[93] B. Dorsemaine, J.-P. Gaulier, J.-P. Wary, N. Kheir, and P. Urien, “Internet of things: A definition & taxonomy”, in *Next Generation Mobile Applications, Services and Technologies, 2015 9th International Conference on*, IEEE, 2015, pp. 72–77.

[94] R. K. Naha, S. Garg, D. Georgakopoulos, P. P. Jayaraman, L. Gao, Y. Xiang, and R. Ranjan, “Fog computing: Survey of trends, architectures, requirements, and research directions”, *IEEE access*, vol. 6, pp. 47 980–48 009, 2018.

[95] S. Yi, C. Li, and Q. Li, “A survey of fog computing: Concepts, applications and issues”, in *Proceedings of the 2015 workshop on mobile big data*, ACM, 2015, pp. 37–42.

[96] P. Varshney and Y. Simmhan, “Demystifying fog computing: Characterizing architectures, applications and abstractions”, in *Fog and Edge Computing (ICFEC), 2017 IEEE 1st International Conference on*, IEEE, 2017, pp. 115–124.

[97] T. N. Gia, A. M. Rahmani, T. Westerlund, P. Liljeberg, and H. Tenhunen, “Fog computing approach for mobility support in internet-of-things systems”, *IEEE Access*, vol. 6, pp. 36 064–36 082, 2018.

[98] P. Hu, H. Ning, T. Qiu, H. Song, Y. Wang, and X. Yao, “Security and privacy preservation scheme of face identification and resolution framework using fog computing in internet of things”, *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1143–1155, 2017.

[99] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, “A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications”, *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1125–1142, 2017.

[100] E. Ahmed, A. Ahmed, I. Yaqoob, J. Shuja, A. Gani, M. Imran, and M. Shoaib, “Bringing computation closer toward the user network: Is edge computing the solution?”, *IEEE Communications Magazine*, vol. 55, no. 11, pp. 138–144, 2017.

[101] G. Premsankar, M. Di Francesco, and T. Taleb, “Edge computing for the internet of things: A case study”, *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1275–1284, 2018.

[102] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, “Internet of things: A survey on enabling technologies, protocols, and applications”, *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[103] H. Zhang and C. Meng, “A multi-dimensional ontology-based iot resource model”, in *Software Engineering and Service Science (ICSESS), 2014 5th IEEE International Conference on*, IEEE, 2014, pp. 124–127.

[104] S. Wang, Y. Hou, F. Gao, and S. Ma, “Ontology-based resource description model for internet of things”, in *Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2016 International Conference on*, IEEE, 2016, pp. 105–108.

[105] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of things (iot): A vision, architectural elements, and future directions”, *Future generation computer systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[106] S. Zhao, Y. Zhang, L. Yu, B. Cheng, Y. Ji, and J. Chen, “A multidimensional resource model for dynamic resource matching in internet of things”, *Concurrency and Computation: Practice and Experience*, vol. 27, no. 8, pp. 1819–1843, 2015.

[107] S. Haller, A. Serbanati, M. Bauer, and F. Carrez, “A domain model for the internet of things”, in *Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCom), IEEE International Conference on and IEEE Cyber, Physical and Social Computing*, IEEE, 2013, pp. 411–417.

[108] B. Kantarci and H. T. Mouftah, “Sensing services in cloud-centric internet of things: A survey, taxonomy and challenges”, in *Communication Workshop (ICCW), 2015 IEEE International Conference on*, IEEE, 2015, pp. 1865–1870.

[109] E. Ahmed, I. Yaqoob, A. Gani, M. Imran, and M. Guizani, “Internet-of-things-based smart environments: State of the art, taxonomy, and open research challenges”, *IEEE Wireless Communications*, vol. 23, no. 5, pp. 10–16, 2016.

[110] A. Rayes and S. Salam, “The things in iot: Sensors and actuators”, in *Internet of Things From Hype to Reality*, Springer, 2017, pp. 57–77.

[111] M. Bermudez-Edo, T. Elsaleh, P. Barnaghi, and K. Taylor, “Iot-lite: A lightweight semantic model for the internet of things”, in *Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld), 2016 Intl IEEE Conferences*, IEEE, 2016, pp. 90–97.

[112] R. Dilli, A. Argou, M. Pilla, A. M. Pernas, R. Reiser, and A. Yamin, “Fuzzy logic and mcda in iot resources classification”, in *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, ACM, 2018, pp. 761–766.

[113] M. A. Razzaque, M. Milojevic-Jevric, A. Palade, and S. Clarke, “Middleware for internet of things: A survey”, *IEEE Internet of things journal*, vol. 3, no. 1, pp. 70–95, 2016.

[114] P. Hu, H. Ning, L. Chen, and M. Daneshmand, “An open internet of things system architecture based on software-defined device”, *IEEE Internet of Things Journal*, 2018.

[115] J. Takalo-Mattila, J. Kiljander, F. Pramudianto, and E. Ferrera, “Architecture for mixed criticality resource management in internet of things”, in *Proceedings of 2014 TRON Symposium (TRONSHOW)*, IEEE, 2014, pp. 1–9.

[116] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of things: Vision, applications and research challenges", *Ad hoc networks*, vol. 10, no. 7, pp. 1497–1516, 2012.

[117] J. Singh, T. Pasquier, J. Bacon, H. Ko, and D. Eyers, "Twenty security considerations for cloud-supported internet of things", *IEEE Internet of things Journal*, vol. 3, no. 3, pp. 269–284, 2016.

[118] Y. Yang, L. Wu, G. Yin, L. Li, and H. Zhao, "A survey on security and privacy issues in internet-of-things", *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1250–1258, 2017.

[119] O. Hahm, E. Baccelli, H. Petersen, and N. Tsiftes, "Operating systems for low-end devices in the internet of things: A survey", *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 720–734, 2016.

[120] A. Ometov, S. V. Bezzateev, J. Kannisto, J. Harju, S. Andreev, and Y. Koucheryavy, "Facilitating the delegation of use for private devices in the era of the internet of wearable things", *IEEE Internet of Things Journal*, vol. 4, no. 4, pp. 843–854, 2017.

[121] J. E. Siegel, S. Kumar, and S. E. Sarma, "The future internet of things: Secure, efficient, and model-based", *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2386–2398, 2018.

[122] M. Kaur, "Semantic resource discovery with resource usage policies in grid environment", *International Journal of Computer Science Issues (IJCSI)*, vol. 9, no. 5, p. 301, 2012.

[123] A. Kamilaris and A. Pitsillides, "Mobile phone computing and the internet of things: A survey", *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 885–898, 2016.

[124] S Oteafy and H. S. Hassanein, "Resource re-use in wireless sensor networks: Realizing a synergetic internet of things", *Journal of Communications*, vol. 7, no. 7, pp. 484–493, 2012.

[125] Q. Zhou, C.-X. Wang, S. McLaughlin, and X. Zhou, "Network virtualization and resource description in software-defined wireless networks", *IEEE Communications Magazine*, vol. 53, no. 11, pp. 110–117, 2015.

[126] K. Karaoglanoglou and H. Karatza, "Directing requests in a large-scale grid system based on resource categorization", in *Performance Evaluation of Computer & Telecommunication Systems (SPECTS), 2011 International Symposium on*, IEEE, 2011, pp. 9–15.

[127] M. Sharifi, S. Kafaie, and O. Kashefi, "A survey and taxonomy of cyber foraging of mobile devices", *IEEE Communications Surveys & Tutorials*, vol. 14, no. 4, pp. 1232–1243, 2012.

[128] J. Ousterhout, A. Gopalan, A. Gupta, A. Kejriwal, C. Lee, B. Montazeri, D. Ongaro, S. J. Park, H. Qin, M. Rosenblum, *et al.*, "The ramcloud storage system", *ACM Transactions on Computer Systems (TOCS)*, vol. 33, no. 3, p. 7, 2015.

[129] T. Pflanzner and A. Kertész, "A taxonomy and survey of iot cloud applications", *EAI ENDORSED TRANSACTIONS ON INTERNET OF THINGS*, vol. 3, no. 12, Terjedelem–14, 2018.

[130] J. Qadir, N. Ahmed, and N. Ahad, "Building programmable wireless networks: An architectural survey", *EURASIP Journal on Wireless Communications and Networking*, vol. 2014, no. 1, p. 172, 2014.

[131] K. Lee and D. Hughes, "System architecture directions for tangible cloud computing", 2010.

[132] Y. Li, X. Cheng, Y. Cao, D. Wang, and L. Yang, "Smart choice for the smart grid: Narrowband internet of things (nb-iot)", *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1505–1515, 2018.

[133] Y. Xu and A. Helal, “Scalable cloud–sensor architecture for the internet of things”, *IEEE Internet of Things Journal*, vol. 3, no. 3, pp. 285–298, 2016.

[134] Y. Dong, K. Wan, and Y. Yue, “A dynamic resource supply model towards cyber physical system (cps)”, in *2014 International Symposium on Computer, Consumer and Control*, IEEE, 2014, pp. 183–186.

[135] A.-C. Orgerie, M. D. d. Assuncao, and L. Lefevre, “A survey on techniques for improving the energy efficiency of large-scale distributed systems”, *ACM Computing Surveys (CSUR)*, vol. 46, no. 4, p. 47, 2014.

[136] A. M. Ortiz, D. Hussein, S. Park, S. N. Han, and N. Crespi, “The cluster between internet of things and social networks: Review and research challenges”, *IEEE Internet of Things Journal*, vol. 1, no. 3, pp. 206–215, 2014.

[137] T. Winkler and B. Rinner, “Security and privacy protection in visual sensor networks: A survey”, *ACM Computing Surveys (CSUR)*, vol. 47, no. 1, p. 2, 2014.

[138] S. Shin, Y. Song, T. Lee, S. Lee, J. Chung, P. Porras, V. Yegneswaran, J. Noh, and B. B. Kang, “Rosemary: A robust, secure, and high-performance network operating system”, in *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, ACM, 2014, pp. 78–89.

[139] Z. Brabec, J. Holecek, and T. Hruza, “Overview of applicability of ict technologies for smart grids”, in *Proceedings of 15th International Conference MECHATRONIKA*, IEEE, 2012, pp. 1–5.

[140] E. N. Alkhanak, S. P. Lee, R. Rezaei, and R. M. Parizi, “Cost optimization approaches for scientific workflow scheduling in cloud and grid computing: A review, classifications, and open issues”, *Journal of Systems and Software*, vol. 113, pp. 1–26, 2016.

[141] L. Yu, H. Shen, K. Sapra, L. Ye, and Z. Cai, “CoRE: Cooperative end-to-end traffic redundancy elimination for reducing cloud bandwidth cost”, *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 2, 2017.

[142] H. Chen, X. Fu, Z. Tang, and X. Zhu, “Resource monitoring and prediction in cloud computing environments”, in *2015 3rd International Conference on Applied Computing and Information Technology/2nd International Conference on Computational Science and Intelligence*, IEEE, 2015, pp. 288–292.

[143] S. K. Battula, S. Garg, J. Montgomery, and B. H. Kang, “An efficient resource monitoring service for fog computing environments”, *IEEE Transactions on Services Computing*, 2019.

[144] F. Mourlin and C. Mahmoudi, “Monitoring architecture for fog and mobile cloud”, in *2018 17th International Symposium on Parallel and Distributed Computing (ISPDC)*, IEEE, 2018, pp. 109–117.

[145] B. Pourghebleh and N. Jafari Navimipour, “Towards efficient data collection mechanisms in the vehicular ad hoc networks”, *International Journal of Communication Systems*, e3893, 2019.

[146] R. Mayer, H. Gupta, E. Saurez, and U. Ramachandran, "Fogstore: Toward a distributed data store for fog computing", in *2017 IEEE Fog World Congress (FWC)*, IEEE, 2017, pp. 1–6.

[147] T. Kudo, "Fog computing with distributed database", in *2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA)*, IEEE, 2018, pp. 623–630.

[148] M. Malensek, S. L. Pallickara, and S. Pallickara, "Hermes: Federating fog and cloud domains to support query evaluations in continuous sensing environments", *IEEE Cloud Computing*, vol. 4, no. 2, pp. 54–62, 2017.

[149] M. Ryden, K. Oh, A. Chandra, and J. Weissman, "Nebula: Distributed edge cloud for data intensive computing", in *2014 IEEE International Conference on Cloud Engineering*, IEEE, 2014, pp. 57–66.

[150] E. M. Schooler, D. Zage, J. Sedayao, H. Moustafa, A. Brown, and M. Ambrosin, "An architectural vision for a data-centric iot: Rethinking things, trust and clouds", in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, IEEE, 2017, pp. 1717–1728.

[151] M. Gupta, F. Patwa, and R. Sandhu, "An attribute-based access control model for secure big data processing in hadoop ecosystem", in *Proceedings of the Third ACM Workshop on Attribute-Based Access Control*, ACM, 2018, pp. 13–24.

[152] L. Lyu, K. Nandakumar, B. Rubinstein, J. Jin, J. Bedo, and M. Palaniswami, "Ppfa: Privacy preserving fog-enabled aggregation in smart grid", *IEEE Transactions on Industrial Informatics*, 2018.

[153] G. Luo, Z. Zhang, D. Wu, X. Li, and G. Liu, "Research on optimization allocation of manufacturing resource services in the cloud environment", in *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, IEEE, 2017, pp. 1–6.

[154] J. Zhang, X. Hu, Z. Ning, E. C.-H. Ngai, L. Zhou, J. Wei, J. Cheng, B. Hu, and V. C. Leung, "Joint resource allocation for latency-sensitive services over mobile edge computing networks with caching", *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4283–4294, 2018.

[155] L. Yin, J. Luo, and H. Luo, "Tasks scheduling and resource allocation in fog computing based on containers for smart manufacturing", *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4712–4721, 2018.

[156] Z. Cao, J. Lin, C. Wan, Y. Song, Y. Zhang, and X. Wang, "Optimal cloud computing resource allocation for demand side management in smart grid", *IEEE Transactions on Smart Grid*, vol. 8, no. 4, pp. 1943–1955, 2016.

[157] L. Gu, D. Zeng, S. Guo, A. Barnawi, and Y. Xiang, "Cost efficient resource management in fog computing supported medical cyber-physical system", *IEEE Transactions on Emerging Topics in Computing*, vol. 5, no. 1, 2017.

[158] B. Yang, W. K. Chai, Z. Xu, K. V. Katsaros, and G. Pavlou, “Cost-efficient nfv-enabled mobile edge-cloud for low latency mobile applications”, *IEEE Transactions on Network and Service Management*, vol. 15, no. 1, pp. 475–488, 2018.

[159] H. Zhang, Y. Xiao, S. Bu, D. Niyato, F. R. Yu, and Z. Han, “Computing resource allocation in three-tier iot fog networks: A joint optimization approach combining stackelberg game and matching”, *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1204–1215, 2017.

[160] H. A. Chaudhari, C. S. Thaker, and J. Patel, “Performance improvisation through resource usage optimization in cloud environment”, in *Proceedings of the 2014 International Conference on Information and Communication Technology for Competitive Strategies*, ACM, 2014, p. 45.

[161] M. Maurer, I. Brandic, and R. Sakellariou, “Self-adaptive and resource-efficient sla enactment for cloud computing infrastructures”, in *2012 IEEE Fifth International Conference on Cloud Computing*, IEEE, 2012, pp. 368–375.

[162] J. Plachy, Z. Becvar, and E. C. Strinati, “Dynamic resource allocation exploiting mobility prediction in mobile edge computing”, in *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, IEEE, 2016, pp. 1–6.

[163] C. Delimitrou and C. Kozyrakis, “Quasar: Resource-efficient and qos-aware cluster management”, in *ACM SIGARCH Computer Architecture News*, ACM, vol. 42, 2014, pp. 127–144.

[164] C. Choi, J. Choi, and P. Kim, “Ontology-based access control model for security policy reasoning in cloud computing”, *The Journal of Supercomputing*, vol. 67, no. 3, pp. 711–722, 2014.

[165] X. LIU and R. BUYYA, “Resource management and scheduling in distributed stream processing systems: A taxonomy, review and future directions”, *ACM Computing Surveys*, vol. 1, no. 1, 2018.

[166] W.-C. Chien, C.-F. Lai, and H.-C. Chao, “Dynamic resource prediction and allocation in c-ran with edge artificial intelligence”, *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 4306–4314, 2019.

[167] P.-O. Östberg, J. Byrne, P. Casari, P. Eardley, A. F. Anta, J. Forsman, J. Kennedy, T. Le Duc, M. N. Marino, R. Loomba, *et al.*, “Reliable capacity provisioning for distributed cloud/edge/-fog computing applications”, in *2017 European conference on networks and communications (EuCNC)*, IEEE, 2017, pp. 1–6.

[168] A. Hameed, A. Khoshkbarforoushha, R. Ranjan, P. P. Jayaraman, J. Kolodziej, P. Balaji, S. Zeadally, Q. M. Malluhi, N. Tziritas, A. Vishnu, *et al.*, “A survey and taxonomy on energy efficient resource allocation techniques for cloud computing systems”, *Computing*, vol. 98, no. 7, pp. 751–774, 2016.

[169] J. Steenstra, A. Gantman, K. S. Taylor, and L. Chen, *Dynamic resource matching system*, US Patent 8,688,148, 2014.

[170] Ü. Şengül, M. Eren, S. E. Shiraz, V. Gezder, and A. B. Şengül, “Fuzzy topsis method for ranking renewable energy supply systems in turkey”, *Renewable energy*, vol. 75, pp. 617–625, 2015.

[171] X. Chen, Q. Shi, L. Yang, and J. Xu, "Thriftyedge: Resource-efficient edge computing for intelligent iot applications", *IEEE network*, vol. 32, no. 1, pp. 61–65, 2018.

[172] H. Chen, F. Wang, M. Migliavacca, L. O. Chua, and N. Helian, "Complexity reduction: Local activity ranking by resource entropy for qos-aware cloud scheduling", in *2016 IEEE International Conference on Services Computing (SCC)*, IEEE, 2016, pp. 585–592.

[173] H. Cao, Y. Zhu, L. Yang, and G. Zheng, "A efficient mapping algorithm with novel node-ranking approach for embedding virtual networks", *IEEE Access*, vol. 5, pp. 22 054–22 066, 2017.

[174] S. Bharti, K. K. Pattanaik, and P. Bellavista, "Value of information based sensor ranking for efficient sensor service allocation in service oriented wireless sensor networks", *IEEE Transactions on Emerging Topics in Computing*, 2019.

[175] W. Chen and W. Xiao, "Cost-efficient task scheduling for parallel applications on heterogeneous cloud environment", in *2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, IEEE, 2019, pp. 1651–1657.

[176] M. H. Shirvani, N. Amirsoleimani, S. Salimpour, and A. Azab, "Multi-criteria task scheduling in distributed systems based on fuzzy topsis", in *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*, IEEE, 2017, pp. 1–4.

[177] G. Peng, H. Wang, J. Dong, and H. Zhang, "Knowledge-based resource allocation for collaborative simulation development in a multi-tenant cloud computing environment", *IEEE Transactions on Services Computing*, vol. 11, no. 2, pp. 306–317, 2016.

[178] A. O. Kotb, Y.-C. Shen, X. Zhu, and Y. Huang, "Iparker—a new smart car-parking system based on dynamic resource allocation and pricing", *IEEE transactions on intelligent transportation systems*, vol. 17, no. 9, pp. 2637–2647, 2016.

[179] D. T. Nguyen, L. B. Le, and V. Bhargava, "Price-based resource allocation for edge computing: A market equilibrium approach", *IEEE Transactions on Cloud Computing*, 2018.

[180] M. A. Senouci, M. S. Mushtaq, S. Hoceini, and A. Mellouk, "Topsis-based dynamic approach for mobile network interface selection", *Computer Networks*, vol. 107, pp. 304–314, 2016.

[181] G. Büyüközkan and S. Güleryüz, "Multi criteria group decision making approach for smart phone selection using intuitionistic fuzzy topsis", *International Journal of Computational Intelligence Systems*, vol. 9, no. 4, pp. 709–725, 2016.

[182] M. S. García-Cascales and M. T. Lamata, "On rank reversal and topsis method", *Mathematical and Computer Modelling*, vol. 56, no. 5-6, pp. 123–132, 2012.

[183] P. Samimi, Y. Teimouri, and M. Mukhtar, "A combinatorial double auction resource allocation model in cloud computing", *Information Sciences*, vol. 357, 2016.

[184] L. Ding, L. Chang, and L. Wang, "Online auction-based resource scheduling in grid computing networks", *International Journal of Distributed Sensor Networks*, vol. 12, no. 10, 2016.

[185] H. Zhang, Y. Zhang, Y. Gu, D. Niyato, and Z. Han, "A hierarchical game framework for resource management in fog computing", *IEEE Communications Magazine*, vol. 55, no. 8, pp. 52–57, 2017.

[186] J. S. Hirwa, U. Rugwiro, M. Stammers, and C. Gu, "Cloud-based clusters: Multivariate optimization techniques for resource performance prediction", in *2016 15th International Symposium on Parallel and Distributed Computing (ISPDC)*, IEEE, 2016, pp. 165–171.

[187] T. Mehmood, S. Latif, and S. Malik, "Prediction of cloud computing resource utilization", in *2018 15th International Conference on Smart Cities: Improving Quality of Life Using ICT & IoT (HONET-ICT)*, IEEE, 2018, pp. 38–42.

[188] S Kumaraswamy and M. K. Nair, "Intelligent vms prediction in cloud computing environment", in *2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon)*, IEEE, 2017, pp. 288–294.

[189] M. Dabbagh, B. Hamdaoui, M. Guizani, and A. Rayes, "Efficient datacenter resource utilization through cloud resource overcommitment", in *2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, IEEE, 2015, pp. 330–335.

[190] B. P. Hayes and M. Prodanovic, "State forecasting and operational planning for distribution network energy management systems", *IEEE Transactions on Smart Grid*, vol. 7, no. 2, pp. 1002–1011, 2015.

[191] S. S. Chaudhari and R. C. Biradar, "Resource prediction using wavelet neural network in mobile ad-hoc networks", in *2014 International Conference on Advances in Electronics Computers and Communications*, IEEE, 2014, pp. 1–6.

[192] M. Aazam, M. St-Hilaire, C.-H. Lung, I. Lambadaris, and E.-N. Huh, "Iot resource estimation challenges and modeling in fog", in *Fog Computing in the Internet of Things*, Springer, 2018, pp. 17–31.

[193] C. N. Le Tan, C. Klein, and E. Elmroth, "Location-aware load prediction in edge data centers", in *2017 Second International Conference on Fog and Mobile Edge Computing (FMEC)*, IEEE, 2017, pp. 25–31.

[194] M. Aazam and E.-N. Huh, "Dynamic resource provisioning through fog micro datacenter", in *2015 IEEE international conference on pervasive computing and communication workshops (PerCom workshops)*, IEEE, 2015, pp. 105–110.

[195] F. Bi, S. Stein, E. Gerding, N. Jennings, and T. La Porta, "A truthful online mechanism for allocating fog computing resources", in *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, International Foundation for Autonomous Agents and Multiagent Systems, 2019, pp. 1829–1831.

[196] F.-H. Tseng, X. Wang, L.-D. Chou, H.-C. Chao, and V. C. Leung, "Dynamic resource prediction and allocation for cloud data center using the multiobjective genetic algorithm", *IEEE Systems Journal*, vol. 12, no. 2, pp. 1688–1699, 2017.

[197] H. Xu, H. Huang, R. S. Khalid, and H. Yu, “Distributed machine learning based smart-grid energy management with occupant cognition”, in *2016 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, IEEE, 2016, pp. 491–496.

[198] Z. Zhang, T. Song, L. Lin, Y. Hua, X. He, Z. Xue, *et al.*, “Towards ubiquitous intelligent computing: Heterogeneous distributed deep neural networks”, *IEEE Transactions on Big Data*, 2018.

[199] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, “When edge meets learning: Adaptive control for resource-constrained distributed machine learning”, in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, IEEE, 2018, pp. 63–71.

[200] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, *et al.*, “Communication-efficient learning of deep networks from decentralized data”, *arXiv preprint arXiv:1602.05629*, 2016.

[201] J. Ren, H. Wang, T. Hou, S. Zheng, and C. Tang, “Federated learning-based computation offloading optimization in edge computing-supported internet of things”, *IEEE Access*, vol. 7, pp. 69 194–69 201, 2019.

[202] S. Shekhar and A. Gokhale, “Dynamic resource management across cloud-edge resources for performance-sensitive applications”, in *Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, IEEE Press, 2017.

[203] K. Ahmed, D. Bigagli, Z. Hu, and J. Wang, *Resource manager for managing the sharing of resources among multiple workloads in a distributed computing environment*, US Patent 9,632,827, 2017.

[204] B. Baek, J. Lee, Y. Peng, and S. Park, “Three dynamic pricing schemes for resource allocation of edge computing for iot environment”, *IEEE Internet of Things Journal*, 2020.

[205] V. Pilloni and L. Atzori, “Consensus-based resource allocation among objects in the internet of things”, *Annals of Telecommunications*, vol. 72, no. 7-8, pp. 415–429, 2017.

[206] F. Hussain, M. Y. Hassan, M. S. Hossen, and S. Choudhury, “System capacity maximization with efficient resource allocation algorithms in d2d communication”, *IEEE Access*, vol. 6, pp. 32 409–32 424, 2018.

[207] J. Bhimani, Z. Yang, M. Leeser, and N. Mi, “Accelerating big data applications using lightweight virtualization framework on enterprise cloud”, in *2017 IEEE High Performance Extreme Computing Conference (HPEC)*, IEEE, 2017, pp. 1–7.

[208] A. Abuabdo and Z. A. Al-Sharif, “Virtualization vs. containerization: Towards a multithreaded performance evaluation approach”, in *2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA)*, IEEE, 2019, pp. 1–6.

[209] Z. Tao, Q. Xia, Z. Hao, C. Li, L. Ma, S. Yi, and Q. Li, “A survey of virtual machine management in edge computing”, *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1482–1499, 2019.

[210] J. L. de Souza Toniolli and B. Jaumard, "Resource allocation for multiple workflows in cloud-fog computing systems", in *Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing Companion*, 2019, pp. 77–84.

[211] X. Cheng, S. Su, Z. Zhang, H. Wang, F. Yang, Y. Luo, and J. Wang, "Virtual network embedding through topology-aware node ranking", *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 2, pp. 38–47, 2011.

[212] H. Cao, L. Yang, and H. Zhu, "Novel node-ranking approach and multiple topology attributes-based embedding algorithm for single-domain virtual network embedding", *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 108–120, 2017.

[213] S. K. Garg, S. Versteeg, and R. Buyya, "A framework for ranking of cloud computing services", *Future Generation Computer Systems*, vol. 29, no. 4, pp. 1012–1023, 2013.

[214] X. Li, H. Wang, B. Ding, X. Li, and D. Feng, "Resource allocation with multi-factor node ranking in data center networks", *Future Generation Computer Systems*, vol. 32, pp. 1–12, 2014.

[215] J. M. Suit, *Assigning virtual machines to business application service groups based on ranking of the virtual machines*, US Patent 9,354,960, 2016.

[216] A. Gomez-Perez, M. Fernández-López, and O. Corcho, *Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web*. Springer Science & Business Media, 2006.

[217] A. Herzog and A. Buchmann, "Predefined classification for mixed mode environments", *Behavioral Ecology*, pp. 134–141, 2009.

[218] B Annex, D Annex, and E Annex, "Open geospatial consortium", 2015.

[219] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third 'chime'speech separation and recognition challenge: Dataset, task and baselines", in *IEEE Workshop on Automatic Speech Recognition and Understanding*, IEEE, 2015.

[220] X. Masip, E. Marín, A. Juan, A. Queralt, A. Jukan, C. Cordeiro, J. Jensen, M. McGrath, A. Salis, D. Guilhot, *et al.*, "The mf2c project: Challenges & lessons learnt", 2019.

[221] M. Thuzar, "Urbanization in southeast asia: Developing smart cities for the future?", *Regional Outlook*, p. 96, 2011.

[222] M. S. Jamil, M. A. Jamil, A. Mazhar, A. Ikram, A. Ahmed, and U. Munawar, "Smart environment monitoring system by employing wireless sensor networks on vehicles for pollution free smart cities", *Procedia Engineering*, vol. 107, pp. 480–484, 2015.

[223] S. Kahvazadeh, V. B. Souza, X. Masip-Bruin, E. Marin-Tordera, J. Garcia, and R. Diaz, "Securing combined fog-to-cloud system through sdn approach", in *Proceedings of the 4th Workshop on CrossCloud Infrastructures & Platforms*, ACM, 2017, p. 2.

[224] I. Lera and C. Guerrero, *Simulation tool: Yet another fog simulator (yafs)*, 2017. [Online]. Available: https://yafs.readthedocs.io/en/latest/index.html.

[225] M. H. Y. Moghaddam and A. Leon-Garcia, "A fog-based internet of energy architecture for transactive energy management systems", *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1055–1069, 2018.

[226] G.-H. Tzeng and J.-J. Huang, *Multiple attribute decision making: methods and applications*. CRC press, 2011.

[227] S. Agarwal, S. Yadav, and A. K. Yadav, "An efficient architecture and algorithm for resource provisioning in fog computing", *International Journal of Information Engineering and Electronic Business*, vol. 8, no. 1, 2016.

[228] H. Xu and Y. Deng, "Dependent evidence combination based on shearman coefficient and pearson coefficient", *IEEE Access*, vol. 6, pp. 11 634–11 640, 2017.

[229] M. Olafenwa, J. Olafenwa, and K. vcar, *Olafenwamoses/imageai*, 2019. [Online]. Available: https://github.com/OlafenwaMoses/ImageAI.

[230] S. Sengupta, J. Garcia, and X. Masip-Bruin, "Taxonomy and resource modeling in combined fog-to-cloud systems", in *Proceedings of the Future Technologies Conference*, Springer, 2018, pp. 687–704.

[231] D. E. Hinkle, W. Wiersma, and S. G. Jurs, *Applied statistics for the behavioral sciences*. Houghton Mifflin College Division, 2003, vol. 663.

[232] A. Singh, N. Thakur, and A. Sharma, "A review of supervised machine learning algorithms", in *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, IEEE, 2016, pp. 1310–1315.

[233] R. Singh, R. K. Umrao, M Ahmad, M. Ansari, L. Sharma, and T. Singh, "Prediction of geomechanical parameters using soft computing and multiple regression approach", *Measurement*, vol. 99, pp. 108–119, 2017.

[234] M. Andrychowicz, M. Denil, S. Gomez, M. W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, and N. De Freitas, "Learning to learn by gradient descent by gradient descent", in *Advances in neural information processing systems*, 2016, pp. 3981–3989.

[235] S. Sengupta, S. Kahvazadeh, X. Masip-Bruin, and J. Garcia, "Sfddm: A secure distributed database management in combined fog-to-cloud systems", in *2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, IEEE, 2019, pp. 1–7.

[236] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konecny, S. Mazzocchi, H. B. McMahan, *et al.*, "Towards federated learning at scale: System design", *arXiv preprint arXiv:1902.01046*, 2019.

[237] L. Chappell, *Wireshark network analysis*. Podbooks.com, Llc, 2012.

[238] L. Pillaud-Vivien, A. Rudi, and F. Bach, "Statistical optimality of stochastic gradient descent on hard learning problems through multiple passes", in *Advances in Neural Information Processing Systems*, 2018, pp. 8114–8124.

[239] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, "Tensorflow: A system for large-scale machine learning", in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 2016, pp. 265–283.

[240] Z. Wang, X. Zhu, and S. Singhal, "Utilization and slo-based control for dynamic sizing of resource partitions", in *International Workshop on Distributed Systems: Operations and Management*, Springer, 2005, pp. 133–144.

[241] Z.-W. Chen, H. Lei, M.-L. Yang, Y. Liao, and J.-L. Yu, "Improved task and resource partitioning under the resource-oriented partitioned scheduling", *Journal of Computer Science and Technology*, vol. 34, no. 4, pp. 839–853, 2019.

[242] E. Albert, F. S. de Boer, R. Hähnle, E. B. Johnsen, R. Schlatte, S. L. T. Tarifa, and P. Y. Wong, "Formal modeling and analysis of resource management for cloud architectures: An industrial case study using real-time abs", *Service Oriented Computing and Applications*, vol. 8, no. 4, pp. 323–339, 2014.

[243] C. Li, Y. Xue, J. Wang, W. Zhang, and T. Li, "Edge-oriented computing paradigms: A survey on architecture design and system management", *ACM Computing Surveys (CSUR)*, vol. 51, no. 2, pp. 1–34, 2018.

[244] J. L. Lopes, R. S. de Souza, C. F. R. Geyer, C. A. da Costa, J. L. Barbosa, A. M. Pernas, and A. C. Yamin, "A middleware architecture for dynamic adaptation in ubiquitous computing.", *J. UCS*, vol. 20, no. 9, pp. 1327–1351, 2014.

[245] D. Demilo, D. B. Ferriera, R. Viswanathan, R. Mahajani, B. Lincoln, and J. McKernan, *Policy driven cloud storage management and cloud storage policy router*, US Patent 8,799,322, 2014.

[246] R. V. Lopes and D. Menascé, "A taxonomy of job scheduling on distributed computing systems", *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 12, pp. 3412–3428, 2016.

[247] Q. Xu, Y.-B. Zhang, J. Zhang, and X.-G. Lv, "Improved topsis model and its application in the evaluation of ncaa basketball coaches", *Modern Applied Science*, vol. 9, no. 2, p. 259, 2015.

[248] E. Roszkowska and T. Wachowicz, "Application of fuzzy topsis to scoring the negotiation offers in ill-structured negotiation problems", *European Journal of Operational Research*, vol. 242, no. 3, pp. 920–932, 2015.