UNIVERSITY OF NOVI SAD
FACULTY OF SCIENCES
DEPARTMENT OF
MATHEMATICS AND INFORMATICS

Saša Pešić

# An Advanced Resource Management and Indoor Positioning Edge Computing Architecture

– Doctoral Dissertation –

## Napredna (edge computing) softverska arhitektura za upravljanje resursima i unutrašnje pozicioniranje

– Doktorska Disertacija –

Novi Sad, 2020

# Contents

# List of Figures

# List of Tables

# Shorthands and Abbreviations

| | |
|---|---|
| **IoT** | Internet of Things |
| **IIoT** | Industrial Internet of Things |
| **ECP** | Edge Computing |
| **IPS** | Indoor Positioning System |
| **RMF** | Resource Management Framework |
| **SoTA** | State-of-The-Art |
| **BLEMAT** | Bluetooth Low Energy Microlocation Asset Tracking |
| **M2M** | Machine-to-Machine |
| **GW** | Gateway |
| **LBS** | Location-based Services |
| **API** | Application Programming Interface |
| **AI** | Artificial Intelligence |
| **ML** | Machine Learning |
| **DLT** | Distributed Ledger Technology |
| **DoS** | Denial of Service |
| **MQTT** | Message Queuing Telemetry Transport |
| **RSSI/RSS** | Relative Signal Strength Indication |
| **BLE** | Bluetooth Low Energy |
| **SBC** | Single Board Computer |
| **GEMAT** | Geo-fencing Micro-location Asset Tracking |
| **GWAP** | Access Point |
| **GED** | Graph Edit Distance |
| **BMS** | Building Management System |
| **SBM** | Smart Building Management |
| **ANN** | Artificial Neural Network |
| **RNN** | Recurrent Neural Network |
| **LSTM** | Long-short Term Memory |
| **SSID** | Wireless Service Set Identifier |
| **UUID** | Universally Unique Identifier |
| **HVAC** | Heating, ventilation, and air conditioning |
| **GN** | Girvan-Newman algorithm |
| **LM** | Louvain method |

# Acknowledgments

First and foremost I need to thank my advisors, Dr. Miloš Radovanović, and prof. Dr. Mirjana Ivanović for the time we spent working on this thesis and all research articles that came before it. They taught me valuable lessons on writing more clearly and being more accessible to the reader. I would also like to thank the members of the committee, Dr. Vladimir Kurbalija, Dr. Miloš Savić and Dr. Dragan Bošković for their time spent on reading and commenting on this thesis.

I thank the VizLore Labs Foundation for continuous support in the writing of this thesis and carrying out experiments, specifically Milenko Tošić, Ognjen Iković, and Dr. Dragan Bošković. The work that contributed to this thesis was also partially funded by the Serbian Ministry of Education, Science, and Technological Development (project number: OI174023).

I am grateful to many of my colleagues at the Faculty of Sciences and the VizLore Labs foundation for a pleasant work environment and many discussions. I am additionally grateful to my parents, my wife and friends for many good moments over the years.

# Preface

The complexity of the Internet of Things (IoT) systems and the tasks they handle require changes in the way that resources are managed and services are delivered. Therefore, the concept of edge computing was introduced in the sphere of IoT systems to improve the scalability, reactivity, efficiency, and privacy of IoT systems. However, the edge computing concept alone is not enough to support the dynamism of a typical IoT system. Robust frameworks should be put in place to perform physical and logical resource management, decision-making processes failover and handover management and security, and system health management.

Location detection and tracking of indoor objects present a complex problem because it largely depends on the context within which the indoor tracking system operates – interior space type and layout, dynamic and static physical obstacles and signal fluctuation and interference, pose serious challenges in achieving satisfactory precision in indoor positioning systems (IPSs). Besides, for IoT systems, one of the basic requirements is minimalist utilization of resources (storage, memory, processor) as well as the processing of data as close to their source as possible to optimize the decision-making process. Above all, the IoT IPS must dynamically adapt to the physical context in which it operates to enhance the functionality of the system, and because of the dynamic nature of the IoT systems themselves – IoT systems consist of devices that frequently connect/leave the IoT system (i.e. node churn), have different physical and logical characteristics, and communicate frequently with other devices that are/are not part of the observed IoT system.

The mixture of problems and requirements for creating an IoT platform with an embedded system for indoor positioning and object tracking creates a solid ground for a serious research and engineering endeavor. Therefore, this thesis covers the creation of an advanced IoT platform for an IPS that includes the detection, positioning, and indoor tracking of objects based on the Bluetooth and WiFi protocols. Therefore, the research within the thesis is divided into two major tasks:

A Building a Resource Management Framework (RMF) for automated management and load-balancing of resources designed for a real-world IoT platform;

# Dissertation Outline

To elaborate on the research tasks mentioned in the prefaces, this thesis will be divided into two major parts.

*PART I* will aim to shed light on IoT and edge computing systems and accompanying computing and architectural paradigms, their definition, areas of application, and common use-cases, as well as operational, business, economical, social challenges and benefits. It will aim to illustrate modern needs and requests in building IoT systems and current State-of-The-Art (SoTA) approaches to designing them. Additionally, it will especially discuss the security and privacy topics of IoT and edge computing systems. It will encompass research, design, and implementation of an MQTT-based Resource Management Framework for Edge Computing systems that handle: resource management, failover detection and handover administration, logical and physical workload balancing and protection, and monitoring of physical and logical system resources designed for a real-world IoT platform. It will also discuss modern requests for such frameworks, current SoTA approaches, and offer a solution in the form of a software framework, with minimal implementation and communication overhead.

*PART II* will aim to elaborate on IPS, their definition, deployment types, commonly used positioning techniques, areas of application, and common use-cases, as well as operational, business, economic, social challenges, and benefits. It will specifically discuss designing IPS for the typical IoT infrastructure. It will offer insights to modern IPS requests, current SoTA in solving them, and underline original approaches from this thesis. It will elaborate on the research, design and authors' implementation of an IPS for the IoT – Bluetooth Low Energy Microlocation Asset Tracking (BLEMAT), including its software engines (collections of software components) for: indoor positioning, occupancy detection, visualization, pattern discovery and prediction, geofencing, movement pattern detection, visualization, discovery and prediction, social dynamics analysis, and indoor floor plan layout detection.

# Part I

# Edge Computing: Security and Resource Management

# Chapter 1

# Internet of Things and Edge Computing

IoT is an emerging hot topic of technological, economic, social, and industrial importance. Among many of the technological phenomena that are present today, the exponential growth of the Internet of Things usage will most definitely result in a great technological transformation that will forever change the world that we live in. Driven by artificial intelligence, cognitive computing and new solutions for device-to-device connectivity as well as rising technologies concerning big data and data analytics, the adoption of the Internet of Things concept is accelerating rapidly.

This chapter will elaborate on the definitions, architectures, opportunities, challenges, and impacts of IoT and Edge Computing platforms. It is structured as follows: Section 1.1 introduces the terms IoT and ECP, and elaborates on comprehensive and many definitions academia and industry provide for them; Section 1.2 discusses typical IoT/ECP system architectural designs and components; finally, Section 1.3 provides a refreshing perspective on opportunities, challenges and impact of IoT/ECP in the areas of industry (Subsection 1.3.1) and global economy (Subsection 1.3.2).

## 1.1 Definitions

Before IoT, the concept of cloud computing has brought a revolution in how we build our applications [10]. Even though the cloud offers numerous advantages, it does present several concerns such as security, privacy, availability of data and services, reliability, and performance [35]. IoT systems aim to solve some of these challenges.

The term IoT refers to all devices of everyday life that are connected to the Internet and that have some kind of intelligence. It is applicable in domestics, all kinds of appliances that communicate with the user, biomedical control systems, trends in consumer use, automotive industry, etc. Smart cities are great incubators for IoT systems [217] that make urban living more enticing, such as agile and convenient transportation systems, eco-friendly heat, and electricity managements [98] and energy-efficient buildings [133]. Driven by artificial intelligence, cognitive computing and new solutions for machine-to-machine (M2M) connectivity as well as rising technologies concerning big data and data analytics, the adoption of the IoT concept has accelerated rapidly. The idea of IoT rests in connected devices acting autonomously, communicating between themselves on a global level, to make human lives more convenient [60]. The innovation of IoT relies on billions of low-powered sensors that monitor processes, take measurements, gather data, and communicate with other sensors and data centers. All devices that are connected to the Internet are a potential data source.

Much of the data in the IoT is processed using cloud computing resources. However, data provides the most value when it is interacted with real-time. Cloud computing is good for offline analysis of large data sets which provides the basis for proactive management of processes and resources. However, apart from this strategic nature of data analysis, most systems and processes in the IoT domain require real-time actions and cannot risk delays introduced by transferring all the data to a cloud platform for analysis. That is why data processing functionalities are being moved to the edge, which represents a computing paradigm better known as **edge computing**. This leads to providing real-time actionable insights that transform into major business benefits. Based on what the scientific community agrees to be characteristic of edge computing, the following *sequential* definition of what *edge computing* is can be devised. Edge computing is:

A an extension of the cloud paradigm,

B a network of widely geographically and/or densely distributed heterogeneous and ubiquitous devices describing a system architecture paradigm that rests on low and predictable latency, location awareness, real-time interaction, and big data analytics,

C which are accomplished by bringing computational resources and logic closer to actionable data at the edge of the network and providing automatically manageable, scalable and elastic resources and services,

D thus creating a space for deploying hierarchically organized, mobile and reactive systems that run close to end-users and are capable of real-time big data streaming and processing, while efficiently preserving privacy at all levels.

There are other terms describing edge computing-like architectures, and many other observations of what it comprises. While some refer to the edge computing to be this whole lower layer of an IoT system having IoT gateways, sensors, etc., other also use the term *mist computing* [161, 216] to divide the resource richer (an IoT gateway, e.g. Raspberry Pi) from resource poorer devices (a temperature sensor, IBeacon, etc. ). Similarly, *fog computing* is used as a term. While some authors make a clear separation of edge and fog computing, others consider it a synonym. Dolui et al. [36] provide an extensive insight into the comparison of edge computing and related computing paradigms, and is worth taking a look at. Despite the different views and definitions, the idea is the same – sharing of computational resources will be a central element that is categorically necessary for a functional IoT ecosystem. Thus, following this idea, in this thesis, these terms are considered equivalent.

## 1.2 Architectures and System Design

An IoT architecture contains a plethora of elements: network and decision-making gateways, sensors and actuators, communication and data storage protocols, cloud services and data centers, etc. Since the architectural complexity of an IoT system is high, these various types of software and hardware components must be included systematically into a sophisticated and unified network, which is an IoT system.

Designing an IoT system (and its architecture) consists of 5 building (design) stages (see Figure 1.1):

1. Sensors and actuators
2. IoT gateways
3. Edge computing
4. Cloud computing
5. Human control

**(1) Sensors** can transform information obtained from the outer world into data for further processing. Thus, sensors represent the first building block of IoT architectures, since IoT is all about connecting the physical and digital worlds. Gaining knowledge about various data representations that are going to be present in an IoT system is necessary to continue to architect other components and workflows. Actuators can intervene in the physical worlds – they can switch on the light and change the temperature in a house. Sensors and actuators cover everything needed to interact with the physical context of the deployed IoT system. At this stage, raw data gets collected and sent further.

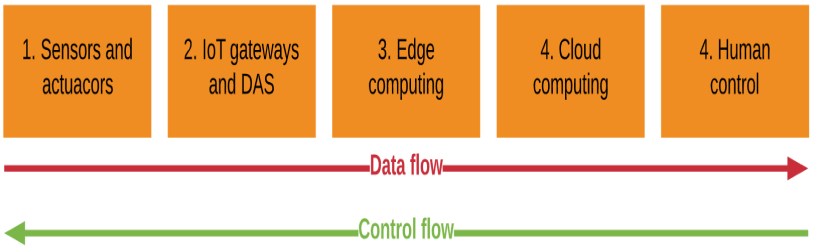



**Figure 1.1:** IoT architectures 5 building stages.

**(2) IoT gateways** are typical starting points for information processing, once the information has been captured by a sensor, or before it needs to be fed to an actuator. They are often deployed in the physical context of the IoT system, work in wireless networks, and transmit the data further. The crucial importance of this stage is to process the vast amount of data collected previously (1) and package it for further analysis. Stage (2) makes data both digitized and aggregated.

**(3)** Any light-to-heavy **processing** done **at the edge** of the network i.e. on devices that rest in the same physical context as the sensors, actuators, and gateways from (1) and (2) is considered edge computing. Edge components/systems perform enhanced data analytics and pre-processing here (machine-learning, filtering, and visualization, etc.). Some IoT systems can be architected solely with components from stages (1), (2), and (3), however, that is rarely the case. Such systems are pure edge computing systems. IoT Gateway components (such as RaspberryPi, Arduino, etc.) are not always enough for fast or complex data processing tasks. Thus, while a large percent of tasks could be carried out at the edge, some will still require more resources (storage, computation, etc.).

**(4) The analysis, management, and storage of data** is handled at the last stage and is typically carried out by leveraging cloud resources (Hardware as service, Software as a service, Platform as a service components). This state enables in-depth processing. After meeting all the standards of quality and requirements, the information is sent back to the physical world (in a processed and precisely analyzed form).

For IoT systems requiring strict human-based control or otherwise requires controlled user inputs, the last stage includes initiating **(5) control over the IoT architecture** components for system administrators and/or end-users. For system administrators, this stage includes enabling system management and administration and control over hardware and software components for all previous stages (1, 2, 3, 4). For end-users, this stage includes workflows to send inputs or commands down the architectural stages (to stage 1).

## 1.3 Opportunities, Challenges and Impact

The beneficial effects of IoT will be important for individuals, enterprises, and policymakers, ranging from helping municipalities minimize healthcare costs and enhance the quality of life, lowering carbon emissions, increasing access to schooling in rural underserved areas, and enhancing travel safety. In the rest of this section opportunities, challenges, and impact of IoT will be explored through real-world use-cases in the industry (Section 1.3.1) and economy (Section 1.3.2). This section helps the reader explore various applications of IoT to better understand the context of this thesis.

### 1.3.1 Industry

The Industrial IoT (IIoT), much like the Internet of Things in general, encompasses multiple use cases, industries, and applications. Initially focused on maximizing operational efficiency, rationalization, automation and maintenance, the IIoT provides plenty of opportunities in the fields of automation, optimization, smart manufacturing and smart business, asset performance management, maintenance, industrial regulation, the movement towards on-demand service models, new ways of serving customers and the development of new revenue models.

The IIoT can be described as devices, computers, and people that use advanced data analytics to allow intelligent industrial operations for transformational business outcomes. The IIoT has become the largest and most important part of the IoT, but from an investment perspective, user applications will pick up the pace soon. Currently, in the overall IoT picture, the IIoT is far more relevant and advanced compared to IoT systems being built for small to medium scale use cases (e.g. smart homes). This is due to the fact that industrial IoT use-cases yield larger overall market sizes and profits and have the potential to transform manufacturing operations to significantly increase efficiency and lower costs (which has a positive effect for both customers and producers). Companies today are looking for a competitive advantage, and the benefits of embracing IIoT impact everything from maintenance, to supplier logistics to employee workflows to product delivery.

Three large markets are benefiting greatly from IIoT: manufacturing, transportation and logistics, and energy and utilities. Data presented in the remaining of this section is collected from corporate research reports: IDC [30, 29], DHL and Cisco [122]). Location-based services, which are the basis for the second part of this thesis will also be explored here from an industrial usability standpoint as they are heavily intertwined in the three aforementioned industries.

Manufacturing is the largest industry from an IoT spending perspective. In 2016, the manufacturing operations accounted for $102.5 billion in IoT spending on a total of $178 billion, all IoT use cases included (in manufacturing). With a combined outlay of $178, manufacturing is by far the largest sector on the IIoT,

and the manufacturing operations category outweighs all other IoT use-case investments in all sectors, including customers. Two other IoT verticals that are relevant for expenditure in manufacturing, in addition to operations, are asset management and maintenance and field service (according to IDC's report [30]).

Transportation is the second-largest sector from the perspective of spending on the IoT. Transport and logistics (T&L) companies are looking to step up the value chain with advanced IoT-enabled communication and monitoring processes, workflows, and systems. The transportation industry has reached $78 billion in IoT spending and is expected to continue rising rapidly, as is the case with the IoT manufacturing sector. The biggest use case in transport is freight tracking, good for a vast majority of overall IoT transportation spending with a total of $55.9 billion, and remains a key business driver until 2020. Looking at the overall growth of the IIoT in transport and logistics, we see the increasing emergence of a digital supply chain and connected logistics reality, which is at the same time one of the challenges facing the manufacturing industry and the T&L market as many players do not have a digital strategy in place and are urged to accelerate their digital transformation efforts. The main drivers for IIoT in T&L are IT defense, communication systems, supply chain management systems, and vehicle and transport tracking. The IIoT is a leading force in the connected logistics environment along with cloud analytics and freight tracking.

Oil and gas, smart grid, and many other advances and use cases in the overall energy and infrastructure sector are also a big part of the IIoT. Utilities alone are the third industry from the IoT spending sense, having hit a total of $69 billion in 2016, according to previously reported data from IDC [29]. There is also one technology sector here that stands out: the smart energy and gas grid, which accounted for $57.8 billion in 2016. The IIoT plays a key role in the overall digital transition in many parts of the broad ecosystem towards a digital supply chain and value chain products, which naturally also touches on retail and consumer-facing aspects. Smart grids, however, are crucial in providing network transmission and distribution from a strictly industrial internet perspective. Others include plant performance, maintenance, and data-driven opportunities arising from smart grids and operations and services allowed by IoT.

Location-based services (LBSs) refer to software services that use geospatial data to provide location information to other systems and/or users. When it comes to smart space services, the role of geospatial technologies is highly significant. LBS is critical for many businesses, as well as government organizations, to get real insights from data linked to a specific location where activities take place. One of the most important and useful features is the spatial patterns that location-related data and services can offer, where the position is a common denominator in all these activities and can be leveraged to better understand patterns and connections.

LBS deployments have the potential to improve almost every aspect of the infrastructure of a smart city: traffic and transit control, energy efficiency, environmental planning, real estate growth, crime reduction, event management, etc. Smart cameras, autonomous vehicles, and drones all use the information on the location to gather and handle data. Digital twin technology and indoor positioning systems are also enabled by LBSs. Digital twins refer to any kind of handling digital representations of objects from the real world. In supply chain and logistics systems, LBS enables tracking products to travel inside a factory, office, or the whole supply chain to enhance overall operations. Adding a location to asset monitoring systems helps improve productivity with cost-reducing knowledge about the position in real-time. LBS can assist in monitoring, analyzing, and applying meaning to asset position data for use in predictive analysis, supply chain management, or position-based market research.

### 1.3.2 Economy

The overarching connectivity of things in the Internet of Things (IoT) presents an appealing environment for innovation and business ventures, thus creating fertile grounds for a new type of economic landscape [141]. This is referred to as the M2M economy, and this term defines the new layer of the digital economy facilitated by M2M payments in the IoT. M2M economy refers to the market of devices (hardware, such as sensors and actuators), connectivity and network services, and managed services that are generated by the interconnection of things (data analytics, value-added applications). Micropayments (intuitively small transaction values) between connected IoT devices enable a new layer of the economy, bringing convenience, faster settlements, and automatization to the end-user in a new, simpler manner [120, 132]. In an M2M economy landscape, smart vehicles might cooperate to marshall traffic [123] or negotiate for a parking spot [53], delivery drones might settle delivery priorities using micro-payments [45], and sensors might request access to higher network bandwidth for urgent or prioritized data transfers [3].

To proliferate the M2M economy would have to embrace a structured set of basic requirements for participating devices. Firstly, energy consumption, network connectivity, and bandwidth as basic machine operation needs have to be meet [176]. Safety and security should be ensured [7] as IoT devices in an M2M economy should have a reliable infrastructure [94], access to oracles [166], trusted execution environments for operations, predictive maintenance and quantum computing resistance to some degree. Next, defining unique digital identities for all participating devices, to make all M2M communication and collaboration transparent and traceable, is of utmost importance. Finally, the peak of the requirements set

includes self-learning, self-optimization, and self-actualization of devices based on machine learning principles.

Machines will become the customers/stakeholders of the future. Using Application Programming Interfaces (APIs) as software intermediaries, Distributed Ledger Technology (DLT), and Artificial Intelligence (AI) the design of financial/economical collaboration channels are bound to shift from user to machine-centric. AI will manage the rating Key Point Interfaces (KPIs) for consumer goods, flexibility and price, and DLT will reduce the latency between transactions and settlements, especially in the field of the supply chain by cutting out the most intense and time-consuming middleman. As DLT reshapes the financial services industry, other industries are aligning. Still, many are viewing DLT as a cost or means to perform risk mitigation. On the other hand, some are beginning to conceptualize an entirely new frontier of services and models of monetization. The intersection of the physical world and IoT with blockchain is one of those frontiers. Essentially, blockchain enables trustless transaction management between network parties. The absence of a trusted intermediary enables faster reconciliation among transacting parties, and the heavy use of cryptography facilitates authoritativeness in the network [31]. Blockchain represents a change in architecture, from a centralized trust system to decentralized, and with that change, there is an associated change in market structure. The low cost of transacting, security and integrity of data, verifiable transactions and identities, transparency, and traceability create fertile grounds for a new, decentralized M2M economy. With built-in cryptocurrencies, blockchains provide a way for participating devices to engage in economic transactions while at the same time incentivizing them [197]. This is enabled by using smart contracts for reconciliation in M2M micro-payments between devices, executing payments, and automatically handling any eventual disputes. Smart contracts comprise conditions that must be met for autonomous transacting. They are stored and run on a blockchain, which makes them decentralized and secure [149]. Smart contracts provide a necessary framework that enables running M2M transactions in a secured and scalable ecosystem, thus also attending to many of the security challenges that IoT currently faces. Conclusively, the blockchain might provide the optimal backbone for connected IoT devices to securely exchange data and transact autonomously in an M2M economy.

Conclusively, global payment systems are ready for disruption, with one of many important drivers being the ampleness of devices and potential M2M transactions enabled by the IoT. By bypassing the fixed cost component of payments in current payment systems, alternatively, transaction costs are vastly reduced on a larger scale, this directly creating a space for bigger transaction volumes and M2M economy escalation. Costless payment mechanisms, that bypasses current financial structures, will be another driving point for the proliferation of the M2M economic landscape. DLT will disintermediate many industries that rest on trusted interme-

diaries and 3rd parties to mediate disputes and business processes. By providing a trustless system with no intermediaries, DLT creates opportunities for people to interact in manners that were not possible before. This behavioral, societal, and economic change will initiate large-scale changes to a multitude of industries that have long dictated society and will cue the beginning for new ones, based on a decentralized future.

Embracing the concept of the M2M economy will have a large influence on global trade flows, and therefore have economic and geopolitical implications. In 2015, it was estimated that every second, another 127 devices are connected to the Internet [32]. Connected home applications (home automation security and video surveillance), white goods, and tracking applications will represent 48% of the total M2M connections by 2022, showcasing the pervasiveness of M2M in our lives [27]. Cellular IoT connections are expected to soar to 3.5B in 2023 [15], increasing at a compound annual growth rate of 30%. IDC projects worldwide IoT technology spending to reach $745 billion in 2019 and $1.2 trillion in 2022 [191]. Concerning application and acceptance, 90% of senior executives in technology, telecommunications, and media, say that IoT is crucial for their business to some extent in 2018, and 750 technological leaders confirm forecasts that IoT will drive the biggest business transformation in the next 3 years [218].

Use cases that need actualization of the M2M economy are not lacking. Electric vehicles (EVs) could make fast, secure, friction-less payments to charging stations using blockchain [188], and EV owners could even offer their private charging stations to others for a fee, creating a P2P EV charging network [131]. Other potential energy use cases for M2M transactions include energy/power trading between smart grids and homes – IoT devices can trade energy between themselves or purchase energy from the local or utility grid when certain conditions are met (e.g. price is lowest). By tokenizing electricity, people could share the ownership or output of their renewable energy projects, and potentially increases the liquidity of local and global renewable energy finance markets [109, 145].

Autonomous vehicles present another landscape that can benefit from enabled M2M economy. Vehicles paying for parking, bidding, or negotiating for parking spaces aid smart transportation management use-cases [184]. Modern vehicles are equipped with automatic malfunction detection systems, and in extension they could as well order spare parts, schedule a maintenance appointment and pay for all of that autonomously. Other land transport applications might include cargo logistics [173], optimizing supply chains [102, 92], automatic fuel purchase, emission trading [84] etc.

Traditional trade finance is often tied to particularly high costs and demanding procedures, due to a paper-heavy bureaucratic process and the challenges of coordinating the plethora of players implicated in a trade transaction [49]. In the IoT/Blockchain M2M economy landscape, this topic has been tackled by a

plethora of companies, making advancements in improving the security of traditional trade finance transactions and streamlining and digitalizing processes (especially letters of credit) as well facilitating *know-your customer* (KYC) processes and easing supply chain finance products and models (essDocs[1], Bolero[2], Barclays[3], Waves[4], we.trade[5], etc.). Streamlining financial flows between buyers, sellers, and financiers, and enhancing the speed, transparency, security, and reliability of supply-chain financing and markets is another key driving point for M2M economies to have a global impact. This is possible by leveraging DLT, AI, IoT, and smart contracts. Having a landscape where data is of extreme value, IoT devices might engage in trading collected data for a fee or a service in return. Devices can engage in trading power and storage capabilities to other devices, as demonstrated with Golem blockchain[6].

IoT and advances in AI are making it possible for devices, sensors, and actuators to operate autonomously and communicate directly without human intervention. This leads to many potential M2M economy use cases throughout industries. Industrial machines could be paying 3D printers to print replacement parts. In an M2M economy, industries may be able to leave asset management and maintenance to the assets themselves [54].

All of the aforementioned are entirely new business models that were not possible with traditional financial systems. Due to the disruptive power of IoT, DLT, and AI, the M2M economy is offering various means to make human lives more comfortable, while at the same time offering new value on a global scale through many industry verticals and horizontals.

[1] essDOCS – `www.essdocs.com`
[2] Bolero – `www.bolero.net`
[3] Barclays – `www.barclayscorporate.com`
[4] Waves – `www.wavesplatform.com`
[5] we.trade – `www.we-trade.com`
[6] Golem – `www.golem.network`

# Chapter 2

# Security and Privacy Concerns

IoT security and, especially, security in edge computing systems are an emerging discipline. Much of the security issues that exist in the cloud, also exist in the edge. These include identity spoofing, authorization, and authentication, denial of service attacks, etc. In the real world, it is more difficult to handle edge security than cloud security, since edge devices are deployed in places that are normally out of rigorously controlled environments, typical for data centers. Furthermore, edge computing systems need to employ distributed security mechanisms, since having a central body introduces a single point of failure. Additionally, these mechanisms need to be computationally undemanding and their implementations lightweight so they could be deployed on resource-poor devices: smartwatches, single-board computers, even single-sensor devices with microchips and embedded software.

This chapter will elaborate on the peculiar security context of IoT/ECP systems, issues and requirements, and introduce a novel categorization of security solutions for IoT/ECP systems. The chapter is structured as follows: Section 2.1 provides answers to the question "What is uniquely challenging when it comes to the security of IoT/ECP systems?"; Section 2.2 underlines security engineering approaches most suitable for IoT/ECP systems; Section 2.3 provides a SoTA overview of actual categorization approaches for security solutions in IoT/ECP systems; Section 2.4 summarizes one of the original contributions of the thesis, a methodological security categorization framework for IoT/ECP-like architectures and systems called 'CAAVI-RICS'; finally, Section 2.5 draws conclusions from Section 2.4 and underlines the framework's advantages and shortcomings.

## 2.1 Specific Security Context of Edge Computing Platforms

The smart devices in our homes, cars, hospitals, offices, airplanes, and different electronic wearable devices, are all effectively endpoints in the IoT and can be used to increase the efficiency of resources management. In edge computing, processes lead to reactive, and near real-time decision-making. On the other hand, by constantly gathering data, observing and monitoring trends, and complex data analytics, edge computing together with cloud computing constructs powerful proactive systems that adapt to every operational context. This, being the breakthrough cloud/edge computing methodology has made in terms of responsiveness and reactiveness of distributed systems, is also the biggest challenge in overcoming existing and upcoming security threats.

The IoT is giving every device a possibility to connect to the Internet. With this potential, come great security concerns. Through the evolution of computing, we have seen endless cycles of security and privacy awareness, from off-the-grid computers, through first online personal computers, and now IoT. Today organizations typically build systems with several layers of security: security policies, firewalls, intrusion detection systems, behavioral patterns analysis, etc. As we enter a new era of computing brought by web 4.0, IoT, and edge computing, we must reexamine the whole concept of security.

Securing an edge computing/IoT system is very different from securing a data center. IoT security introduces a large attack surface through massive scale device deployments. Many IoT devices (sensors, consumer items) are deployed on a large scale. As a result, the potential number of interconnected links between them cannot be accounted for. Furthermore, many of these devices will need to dynamically communicate with other, external devices unpredictably.

Next, often a security breach can happen a long time before it is detected. There are vulnerable operating systems that are rarely updated. Commonly, if one type of device has a security issue, then most likely other similar devices have the same issue which can thus be massively exploited. As devices' lifetime goes, typical IoT devices such as sensors have a very long lifetime, which can lead to them outliving implemented security mechanisms. As a result, many security concerns must be taken into account when designing a product or a system for IoT.

When you add billions of IoT devices generating zettabytes of data daily – together they provide a major surface for generating disorder, launching data theft attacks, or simply maliciously invading and collapsing a system from the inside by flooding communication channels or hurting protocols and processes in other ways. To conclude, coming up with an efficient security framework for an IoT edge computing system is much more difficult than handling traditional distributed systems. Cyber-threats may come from many sources, focus on many different

layers and technologies, and the major security task here is to leverage the dense distribution of computational resources properly to utilize an efficient distributed security framework.

However, securing an edge computing platform does bring some important benefits over traditional cloud-centric IoT systems: sensitive data can be kept at the edge level at all times without the need for transferring to cloud, thus enhancing privacy; the edge level can provide more contextual information about security and privacy than traditional cloud-centric IoT systems – information about the operating environment is easy to collect at the edge level; cloud communication overhead is reduced, making the attacks on that link less probable; proximity and intelligence at the edge of the network enable real-time interaction, predictable network latency, and clock synchronization. Security in edge computing systems must leverage these benefits.

## 2.2 Engineering Security for an Edge Computing Platform

There are hundreds of communication and data exchange protocols and frameworks currently used throughout the IoT/edge computing – some running on resource-rich devices, some on resource-poor devices. But the thing they have in common is the lack of infrastructural security – among securing the protocols and devices themselves, there is usually no extremely efficient way of tracking and analyzing every microtransaction in the system. In edge computing systems, a single misguided and unnoticed malicious microtransaction can bring the whole system down. In terms of transaction granularity, a microtransaction refers to the lowest level, the smallest transaction between more modules on the same device, or multiple devices – i.e. a sensor sending temperature measurement to a smartwatch. When a sensor takes in a reading, and filters it, or computes the average over last minute, that is a microtransaction. Further, each reading that a sensor computational module receives from the sensor temperature measurement module is a microtransaction. In conclusion, if the source of such a malicious action is not detected and banned from the network immediately the network could sustain to receive amassed malicious breaches.

In general, we have to reconsider the idea of security – not only resource-rich devices being compromised can bring disaster to the system. It comes down to the fact that there is yet a lot of infrastructural modeling to be done before we are in a place where collaboration and communication between fellow edge nodes are carried out seamlessly. Haphazard security engineering must not be a practice in such systems that merge the physical and the digital world – furthermore, it can be fatal.

Considering the high-density distribution of heterogeneous nodes in the edge, security in the system itself must be autonomous. Security features of an edge computing system should not depend on any cloud-based, or otherwise centralized system, in an ideal edge environment. Yes, cloud services should be included in the deeper analysis of suspicious node behavior, etc. (and derive new and update existing security policies), but reactions to malicious actions should happen instantly when detected, on the edge level of the system, regardless of connectivity to a cloud platform, and without the involvement of system administrators.

### 2.2.1 Security Issues

In an edge-computing architecture, end-to-end security must cover all devices from the cloud level to the edge of the network (gateways, sensors, actuators). However, the security of edge computing systems has to start with secure device hardware. If a device is not trustworthy at the time of deployment, the whole infrastructure becomes unsecured and unstable. When and only when trusted edge nodes are deployed, a secure communication layer encompassing all nodes can be built, thus creating a secure end-to-end edge computing infrastructure.

An edge computing system should secure itself by authenticating the identity of all users, data consistency and integrity, and the availability of system services. An edge system should have the ability to revoke connections and handle key management anonymously, as well as to insert real-time constraints and monitoring them [82]. It is, however, very important that this process is anonymous and stateless to protect sensitive user data (location, identity, etc.) even from the edge computing system itself. For a highly efficient edge computing system, it is also relevant to aggressively analyze activities of both users and administrators, and adjust topology, bandwidth allocation, and traffic policies real-time [155]. These and other security challenges can be categorized as related to physical security or intrusions and will be explained in the upcoming two subsections.

#### Physical Security

One thing to consider for edge computing like systems is hardware security. Hardware security must be the first building block of a trustworthy system. If and only if there are trusted hardware components then there can also be trusted firmware and software components. Since edge computing devices are often deployed in areas outside of rigorous surveillance physical security is important. Device hardware refers to the *things* part of the *Internet of things*: sensors, actuators, etc. These devices‘ main goal is to collect data from different sources, depending on the use case. Depending on the volume of the data, different types of devices can be used (simple sensor, collection of sensors maintained by a computer, a gateway, etc.). Among data collecting, disseminating the data through a communication channel

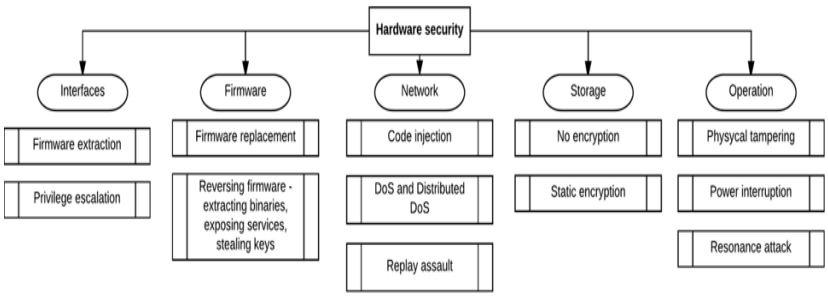



**Figure 2.1:** Summarizing edge computing devices' physical security concerns.

is essential at this layer. At this layer, possible security issues mostly include physical tampering, and they can be grouped by what is the attackers objective (also see Figure 2.1 for the summary):

- **Device interfaces** – an attack can be launched to extract firmware from the device, to gain or raise privileges (also called privilege escalation), but also, physically disturbing the interfaces is an option.

- **Device firmware** – an attacker could replace device firmware with a malicious version through device interfaces. Next, if the firmware is extracted, one can reverse the firmware to look for vulnerabilities like hardcoded credentials, sensitive files, etc. Through emulating firmware and analyzing firmware binaries an attacker can use the knowledge to expose defenseless service (web, ssh), steal encryption keys, etc.

- **Device network services** – attacks focused on code injection, DoS and DDoS attacks, and possible Replay assaults.

- **Device local storage** – if local information is not encrypted that can cause a lot of issues. Also, while accessing information trustworthiness of the information seeker must be checked. One can also exploit the fact that there could be static encryption.

- **Device operation** – focused on physical attacks like interrupting the devices' power, resonance attacks (sensors are made to operate on different frequencies), etc.

Several solutions to the problems described above include:

- **On-boot peripherals check**, making device ports inaccessible, implementing physical locks to keep people from accessing the device, disabling unused interfaces, disabling Universal Plug and Play (UPnP);

- **Boot-loader protection** – secure boot (utilize cryptography code-signing techniques to ensure that the device only executes code produced by the devices Original equipment manufacturer (OEM) or other trusted parties). There are a vast number of USA-based patents handling only this particular subject, originating from 1997 until today [33, 90];

- **Secure firmware updates** – related to boot-loader protection, secure firmware updates ensure the device is running trustworthy code and prevents any attacks seeking to exploit the device's firmware update process.

- **Dynamic data encryption** – instead of statically securing data with hard-coded certificates/credentials, the process should be rather dynamic.

- **Secure communication** – utilization of protocols such as Transport Layer Security (TLS), Datagram TLS (DTLS), IP-Sec add authentication, and in-motion data protection to edge computing devices. They make eavesdropping much more difficult, making it a hard task to steal communication, passwords, device configuration files, or other sensitive data.

- **Local data protection** – Above mentioned security protocols do ensure protection in transit but do not ensure protection while on the device. All sensitive data on the device must be encrypted, in case the device should be accessed by an unauthorized party, discarded or stolen. Consider the example of someone accessing an office printer storing thousands of printed documents on its integrated memory.

In the end keeping the device in a safe, controlled, environment is always the best practice.

### Intrusions

Two types of system-wide intrusions will be discussed: outside and inside intrusions.

An outside intrusion is a network intrusion – an unauthorized activity on the hosting network. In most cases, such unwanted activity consumes network resources predetermined for other uses, and nearly always threatens the security of the network and/or its data. So, outside intrusions refer to typical security aspects of a network system which is connected to the Internet. They include asymmetric routing attacks, buffer overflow attacks, protocol-specific attacks (ARP, IP, TCP, UDP, ICMP, etc.), traffic flooding, and malicious software (trojans, malware, worms, etc.) that are received from remote locations and cause harm to the system. These kinds of attacks are often prevented by a pre-installed firewall or antivirus software, although, we have to be aware of the fact that edge nodes, especially

those resource-constrained cannot run a firewall, and thus might have to rely on the closest firewall/antivirus application on a resource-richer device. The percentage of prevented networking attacks can be increased by utilizing proper network monitoring: checking suspicious network packets, using different intrusion detection techniques [51], even training machine learning models upon empirical data to be able to recognize such intrusions. For example, a malware attack called Zero Day can be identified with a machine learning technique [28].

Outside attacks also refer to attacking web pages and insecure web APIs. When user input is not properly validated it can lead to code injection (SQL injection for example) that can cause unauthorized access to data that can compromise the entire underlying database. It can also lead to forwarding modified data in the system to different nodes. Another important potential security issue concerning web applications is the hijacking of sessions and cookies, as well as embedding malicious scripts inside websites: cross-site scripting.

An outside attack can also come in the form of a denial of service attack (DoS). DoS attacks, and especially distributed DoS attacks bring a new form of security challenges to IoT edge systems. A DoS attack aims to make a service unavailable. An example would be jamming a wireless network signal, flooding nodes with data they cannot handle, etc. When a cloud platform is the main actor in the decision-making process, then DDoS attacks can cause the complete unavailability of system services.

Wireless networks are very often unattended and hostile environments. The ease of access here increases the possibility for a DoS and selective forwarding attack leading to impersonation, message replay, and message distortion issues. One could protect such a network with an efficient key management service, secure routing, and regular auditing of network traffic. Data could be protected by different obfuscation techniques, and the possibility of attack could be lowered by limiting the number of connections as well as using SSL for secure connections. If a wireless access point is not properly secured, it can result in many forms of attack on the system: Sybil attack, illegal bandwidth usage (flood attack), or Man in the middle attack that intercepts network traffic and can alter data or terminate communication.

When talking about data encryption it is of utmost importance for the integrity of the system looking from the inside, as well as outside. For edge computing systems, Guo et al. [194] mention it is important to distinguish between two types of data in the system: sensitive data that will be protected with complex security algorithms and archival data, that will also be protected, but the hijacking of such data stream could not bring harm to the system.

Inside intrusions refer to uncharacteristic behavior of the systems‘ participants inside the system (rogue nodes, man-in-the-middle attacks, etc.) that cause internal disturbance of trustworthiness and integrity, leading to possible incorrect decision

making, false data input/output, privilege escalation, but also data theft, identity theft, etc. Inside intrusions also refer to authentication and authorization issues, edge nodes, and sensors spoofing (inside the edge system itself).

## 2.3 Related Work on Categorizations of Security Approaches

This section gives a SoTA review of research incentives creating different overviews and categorizations of security in IoT and edge computing systems.

Mahmoud et al. [125] discuss the countermeasures for securing IoT through authentication, trust establishment, federated architectures, and security awareness. In their taxonomy of security and threats in IoT, Babar et al. [6] discuss the solutions in identification, communication, physical threats, embedded security, and storage management. An algorithmic overview of solutions is presented by Cirani et al. [26] where solutions are presented through the following chapters: security protocols, lightweight cryptographic algorithms, key distribution, and management, secure data aggregation, and authorization. Nguyen et al. [142] provide a review of only communication protocols for IoT, classifying them to symmetric and asymmetric security protocols.

Kumar et al. [96] are dividing edge security to (1) Network, (2) Data, (3) Access control, (4) Privacy, and (5) Attackers Interest in Private Data. Data security features homomorphic encryption and searchable encryption; access control security features Attribute-based encryption; privacy discusses privacy of data and location stressing the importance of edge computing, and attackers interest in private data describes different stakeholders in the private data tampering area.

Khan et al. [83] have identified 12 security categories to formulate a systematic review. These are: (1) Advanced Persistent Threats – describing attacks whereby the aim is to compromise a company's infrastructure with the desire to steal data and intellectual property; (2) Access Control Issues; (3) Account Hijacking; (4) Denial of Service (DoS); (5) Data Breaches; (6) Data Loss; (7) Insecure APIs; (8) System and Application Vulnerabilities – describing the problems of exploitable bugs arising from software; (9) Malicious Insider – mentioning regular users with malicious intents; (10) Insufficient Due Diligence; (11) Abuse and Nefarious Use; and (12) Shared Technology Issues – features sharing infrastructures, platforms or application domains. At this moment this is the most complete systematic review of security challenges and proposed solutions.

Yi et al. [215] are reviewing security through seven chapters: (1) trust and authentication, (2) network security, (3) secure data storage, (4) secure and private data computation, (5) user privacy, (6) access control, and (7) intrusion detection. Trust and authentication feature mentioning trust models, rogue fog nodes, and

authentication methods (PKI), network security focuses on SDN, secure data storage discusses homomorphic encryption and searchable encryption, TPA, and LT code. Secure and private data computation is focusing on verifiable computing, privacy discusses data, usage, and location privacy. Access control features cryptography solutions as well as Attribute-based encryption and intrusion detection is describing monitoring and analyzing log files, access control policies, and user login information.

While papers mentioned in this section provide extensive overviews of security of IoT and edge computing systems, they do not analyze the abstractions above the categories of security or form an overview methodology that can correspond to not just the proposed categorization (by certaion authors), but others as well. Also, these papers often discuss IoT systems as a mixed blend of cloud and edge computing. These will be addressed in the upcoming sections.

## 2.4 Methodological Security Overview Framework for ECP Architectures – CAAVI-RICS Model

There are three essential rules for the design of ECP security solutions: (1) be efficient, responsive, and resource preserving; (2) act at the edge layer in collaboration with peers; (3) dynamically adapt and advance through feedback. In the rest of this section, the thesis will elaborate on CAAVI-RICS, the proposed methodological framework for discussing security aspects of computer systems, focused on IoT and ECP. These security aspects will be discussed from the point of view of potential for implementation in IoT and ECP systems, generalized to all similar systems and architectures. For researchers in the area, the CAAVI-RICS frameworks offers a novel taxonomy of security solutions for IoT and ECP systems, highlighting the key security concerns of computer systems in general (CAAVI). For IT practitioners, this categorization offers deeper technological insight into aspects behind every CAAVI concern, the RICS, while explaining which solutions can be implemented in IoT and ECP architectures and in what capacity.

CAAVI is an acronym for Credibility, Authentication, Authorization, Verification, and Integrity. The CAAVI principles are in detail discussed through considering 4 aspects (RICS): (1) Rationale (what is it and why is it important?), (2) Influence (how does it affect the overall system well-being if (not) implemented correctly), (3) Concerns (what problems does it bring?) and (4) Solutions (review of current SoTA solutions). This discussion is summarized in this thesis. Detailed elaboration of the CAAVI-RICS model has been presented in three papers by the author of this thesis so far [153, 154, 152]. Thus, in this part of the thesis, the model will be presented through key concepts and illustrations.

**Credibility** is the process of creating trustworthy relationships between devices. It is indispensable in decision-making processes and allows for the development of autonomous channels of communication between resource-constrained edge nodes. A distributed system is positively affected by a well-established legitimacy and structure of action as it facilitates straightforward, mutual collaboration between devices and enables more effective management functions. Consequently, autonomous communication is allowed – the establishment of channels of communication between devices without prior knowledge of one another. Concerns over breach and misrepresentation of credibility in ECP systems can be observed in two aspects. At the hardware level credibility can be compromised by tampering with peripherals, sensors, etc. when the system can be led to provide false information. Many security risks at the hardware level can be triggered by node misconduct (e.g. due to parts malfunction, etc.). At the software level, credibility can be undermined by tampering with system functionality (e.g. malicious code injection, identity spoofing, etc.). Although a higher degree of reliability is needed in some systems where trust is more important (e.g. patient support and care systems), other systems (e.g. home automation systems) may aim to reduce the complexity of safety and security procedures to improve the efficiency of their functionalities. In terms of credibility, different hardware solutions [203, 202, 170], lightweight trust management, and computational frameworks [164, 190] and sacrifice [135] and remote attestation [23] approaches can be applied. The Credibility principle of the CAAVI model has been presented in detail in the cited article [153]. The summary of the results of RICS applied to the credibility principle is given in Figure 2.2.

**Authentication** is the method of determining the claimed identity of the node by verifying the evidence provided. The evidence provided is typically called an authentication credential. There are three key pieces of information that authentication schemes are typically based on, and that are expected from the nodes: knowledge, possession, and condition [69]. Knowledge refers to something known to the node (e.g. password, PIN), possession refers to something owned (e.g. certificate) and condition refers to something issued (e.g. MAC address, serial ID). Proper authentication enables more robust devices and secure communication between system components and end-users. Deploying new services, apps and their updates to existing networks will be effortless because the devices are already authenticated. Authenticated devices can be used to transmit instructions, firmware updates, and security upgrades to other devices while executing concurrent network authentication concurrently. Complex authentication mechanisms provide enhanced security but induce lower system performance and higher network latency. Based on how robust authentication is, the level of decision-making is often affected. Authentication schemes are evolving from single to multi-authentication [71, 128], peer-to-peer solutions are available, and the importance of developing authentication

| Rationale | Influence |
|---|---|
| • Rests on four pillars: unique identity, foreseeable behavior, continuity of positive intentions and reputation-based trustworthiness<br>• Information credibility is directly tied to device credibility<br>• Credibility can be calculated and devices can be assigned a credibility index<br>• Credibility is key to establishing trust<br>• Credibility-assessment management frameworks are necessary | *Positive:*<br>• Enables cooperation, efficient process handling<br>• Enables autonomous communication<br>• Easies device onboarding phase<br>*Negative:*<br>• Results in exposed devices and sensitive data<br>• Can lead to performance and decision-making quality decrease<br>• Repels perspective users and customers |
| • Malicious nodes can form or enter the system<br>• Device's credibility can be compromised on both hardware and software levels<br>• Rogue, misbehaving or faulty nodes can form or enter the system<br>• Fake information and false messages spreading<br>• Sharing sensitive information over public Internet access points | • Distinguish between security requirement for different subsystems<br>• Credibility takes root at the hardware level (HPC, PUF)<br>• Lightweight global trust calculation<br>• Misbehavior detection through logistics trust and STING algorithms<br>• Supporting frameworks to inspect, grant and revoke credibility in real-time<br>• Dynamically allocating security levels to positively influence performance |
| Concerns | Solutions |

**Figure 2.2:** Credibility summary.

schemes and algorithms to meet the abundance of contextual information (physical and logical context of system operation) is recognized [210, 214, 195]. Furthermore, Public-Key Infrastructure (PKI) is still a hot topic for authentication in ECP systems [100], and blockchains are making a debut as of recent [196]. The Authentication principle of the CAAVI model has been presented in detail in the cited article [154]. The summary of the results of RICS applied to the authentication principle is given in Figure 2.3.

**Authorization** is a security mechanism of specifying access rights-privileges to system resources. Authentication protects the system from uncharted malicious entities, while authorization provides an authenticated device or user with secure access to the system 's services. Access control policies (ACPs) are a form of authorization that are used to establish user/client rights or access thresholds for specific system services. In latency-relevant use-cases of ECP, fast authorization provides flexibility in decision-making and resource access management. Besides, dynamic access control mechanisms provide greater value, enhanced functionality, and new levels of convenience and utility. To have a positive impact, authorization must be flexible enough in IoT: multiple forms of access control must be supported; authorization data must have the potential to be easily migrated and aggregated; origin tracing of data and activities must be enabled at all times. Most

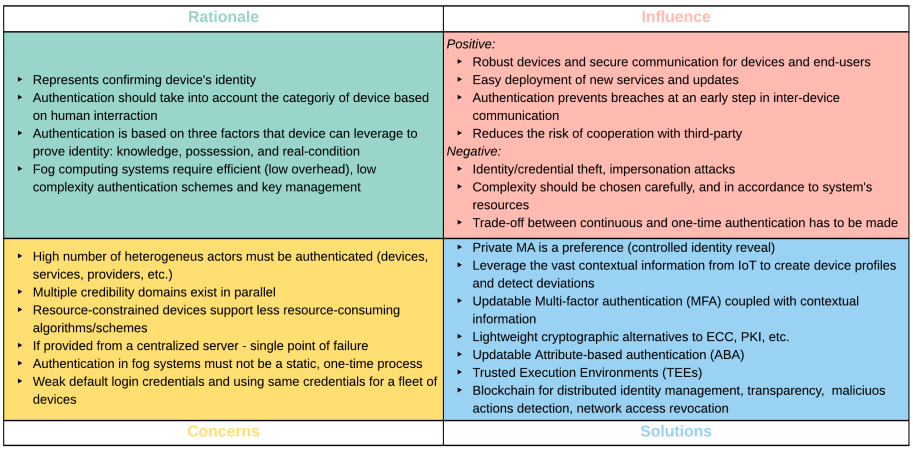



**Figure 2.3:** Authentication summary.

current authorization systems depend on trustworthy central authorities [186] and, if the central authority is compromised, attackers will disrupt authorization policies across the network. The management of authorization policies for devices that are only intermittently connected to the underlying network may present a concern and should be handled carefully. Solutions can be grouped connected to aspects such as the granularity of authorization, peer-level delegation [70], decentralized identity management [85], and blockchain-based solutions for enabling provenance for all of these aspects [5]. Existing standards must also be examined (OAuth 2.0, CoAP – Constrained Application Protocol, DTLS – Datagram Transport Layer Security, etc. [179]). The summary of the results of RICS applied to the authorization principle is given in Figure 2.4.

**Verification** is the process of establishing the truth, accuracy, or validity of the system actions and their results. After already considering CAA (Credibility, Authentication, and Authorization) security layers, verification must further minimize the probability of a malicious action in the system through rigorous analysis of current system-wide devices' behavior. In verification, two frameworks should be put in place in parallel: device and user behavior profiling and actions intent evaluation (and storage of such information). When discussing verification, the focus in CAAVI-RICS will be on intrusion detection approaches falling into two categories: rule-based and behavior-based intrusion detection systems (IDS). Deployment of IDS often includes the deployment of an Intrusion Prevention System (IPS) [14]. The IPS is a software component that can prevent an otherwise successful attack on the system. IDSs can reveal a wide range of issues: handling data in violation of on-site security policies, unauthorized transmission of data outside the network (spyware, keyloggers), viruses, trojans, and malware infections that have

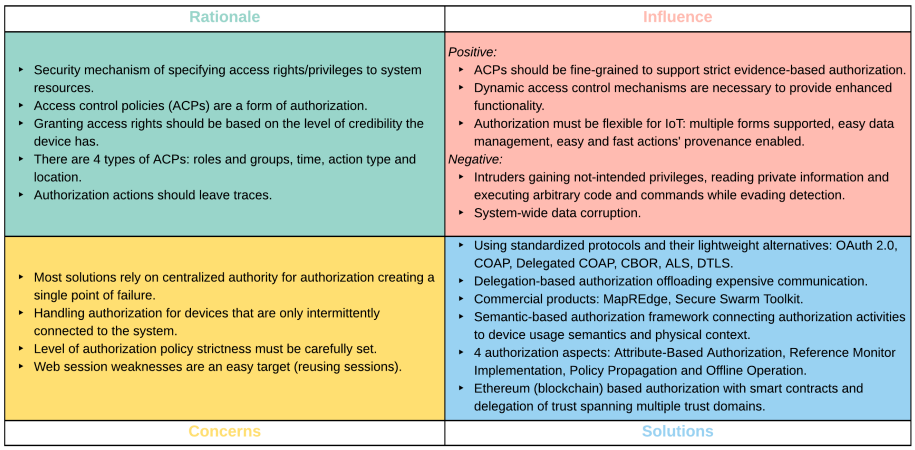



**Figure 2.4:** Authorization summary.

gained control of the system's internal resources, etc. The effect of the IDS on the functionality of cyber-physical networks is determined by both detection and reaction power, and their association with intruder strength and observed behaviour. In ECP environments, the IDS must not be a passive system. If the IDS and IPS fail in their intents, unprevented malicious actions can lead to data tampering, identity theft, etc. The exploitation of actuators has not been discussed so far. This problem is probably the most significant safety issue for the industrial IoT systems because actuators represent 'connectors' in the cyber-physical environment. Next, the form/type of IDS must be deliberately considered and the trade-offs in storage/computation must be used in the evaluation. In the ECP environments, due to lower computing capacities, the underlying system applying the IDS may not be able to have a reasonable data analytics response rate. Effective task-delegation mechanisms should be put in place to respond to this issue when a complicated IDS is active inside a system. Verification solutions are presented from the standpoint of IDS and IPS approaches. Lightweight anomaly detection approaches powered by various machine-learning algorithms are explored [177, 171, 193], together with network traffic real-time analysis and profiling [66], outsourcing IDS [134], and commercial solutions [169, 9, 1]. The summary of the results of RICS applied to the verification principle is given in Figure 2.5.

**Integrity** of a system refers to the capability of performing correctly according to the original specification of the system under various alerting (i.e. adversarial) conditions. The integrity of a system also rests on the integrity of data within, which is the focus of the Integrity principle of the CAAVI-RICS model. The integrity of data is protected when data has not been maliciously changed during transit or in storage. Data integrity can be summarized through addressing 4 as-

| Rationale | Influence |
|---|---|
| ‣ Process of establishing the truth, accuracy, or validity of the system's action and their results.<br>‣ System actions must be properly verified and persisted for auditing.<br>‣ Verification of user and device behaviour needs to be a low-latency, distributed service, relying on edge devices.<br>‣ In verification two frameworks should be put in place in parallel: device/user behaviour profiling and actions intent evaluation and storage.<br>‣ Verification is based on intrusion detection systems (IDS) in two categories: rule-based and behaviour-based. Intrusion detection is often accompanied by intrusion prevention systems (IPS). | *Positive:*<br>‣ IDS can reveal handling data in violation with in-place security policies, unauthorized data transmission (spyware, keyloggers), virus infections .<br>‣ IDS should be connected to existing knowledge about network architecture, applications, participants and security. In dynamic environments such as IoT, this ground-truth must also be continually reviewed and updated.<br>*Negative:*<br>‣ IDS/IPS should not incur latency in the network traffic. An effective trade-off must be made between storage costs and computational complexity.<br>‣ Handling data at the edge might impose a large computational overhead on edge nodes |
| ‣ Non-prevented malicious actions can lead to data tampering, identity theft, etc. Vast volume of data directly impacts network traffic, making it difficult for IDS to function timely.<br>‣ A special focus in industrial IoT should be put on securing hardware, software, and behaviour around actuators.<br>‣ Rule-based IDS need more storage to detect known attacks with the downside of not being able to detect new attacks. Behavioral-based IDS can detect new attacks but their complexity is higher than that of rule-based IDS. IDS might need human evaluation to progress faster.<br>‣ Multi-hop characteristic of network packets transmission | ‣ Lightweight, probabilistic anomaly detection techniques for low-resource IoT devices based on game theory, RF profiling and RSS monitoring.<br>‣ Commercial products include SNORT, OpenWIPS-NG, Suricata.<br>‣ Automata-based IDS detect jam-attacks, false-attacks, and reply-attacks.<br>‣ A fuzzy logic-based IDS framework where each network device relies on an agent component to assess the infection state of each of its immediate neighbors.<br>‣ Plug and protect approaches, focused on ease of deployment, portability, minimum configuration, and versatility. |
| Concerns | Solutions |

**Figure 2.5:** Verification summary.

pects: (1) confidentiality, (2) authenticity, (3) freshness, and (4) reliability. By 2022, IoT data are expected to make up 45% of all traffic on the Internet [72]. Considering that IoT data must aim to be error-prone while platforms retain confidentiality, authenticity, freshness, and reliability. High integrity data brings many benefits: faster problem-solving, cost-effectiveness and efficiency, quality data analytics to identify business opportunities and competitive advantages, etc. Data integrity issues that may arise include tampering, mis-versioning, corruption, etc. Compromising data integrity is worse than data theft because corrupting the data on which an entity relies could cause it to act in a risky and unsafe manner. IoT devices can malfunction on their own and start sending faulty data or even stop broadcasting. Besides, in IoT, it is important that data aggregation schemes reduce power consumption, avoid traffic congestion, and maximize data usability. If one data source in the aggregated data consists of tampered data, the whole aggregation can be labeled as unusable. Some IoT use-cases need to address data freshness very seriously – drones and self-driving cars need real-time information to work properly. Security solutions for data integrity are focused on lightweight encryption and decryption schemes [79, 63], ABE variants [76, 57], lightweight existing cryptography algorithm alternatives [185], and blockchain-based frameworks [112, 39]. The summary of the results of RICS applied to the integrity principle is given in Figure 2.6.

## 2.5 Bridging CAAVI Principles

The CAAVI principles – Credibility, Authentication, Authorization, Verification, and Integrity have been described in that particular order on purpose, as illustrated

| Rationale | Influence |
|---|---|
| ‣ Refers to the capability of performing correctly according to the original specification of the system under various adversarial conditions. The integrity of a system also rests on the integrity of data within.<br>‣ Data integrity can be summarized by addressing 4 aspects: confidentiality, authenticity, freshness, and reliability. | *Positive:*<br>‣ Data with a high degree of integrity brings many benefits: faster problem solving, cost-effectiveness and efficiency, quality data analytics to identify business opportunities and competitive advantage, etc.<br>*Negative:*<br>‣ Data can be tampered with while in storage, and accidental data versioning can occur.<br>‣ Based on false, corrupted or bad-quality data bad code-level decisions are made, that can result in catastrophic outcomes.<br>‣ Low-quality data is inaccurate, non-compliant to regulatory standards, uncontrolled, unsecured, static (not updated), and dormant (not used). |
| ‣ Corrupting the data on which an entity relies could cause it to act in a risky and unsafe manner.<br>‣ IoT devices can malfunction on their own and start sending out false data or stop broadcasting at all.<br>‣ Data segmentation must be handled carefully to avoid segment/aggregation contamination.<br>‣ Some IoT use-cases need to address data freshness very seriously -- drones and self-driving cars need real-time information to work properly.<br>‣ Long/multiple processing pipelines can lead to lower data reliability. | ‣ Dynamic Tree Chaining and Geometric Star Chaining provide authenticity, integrity, sampling uniformity, system efficiency, and application flexibility to IoT data communication.<br>‣ Lightweight hardware-level integrity leveraging PUFs and random time sequences.<br>‣ Lightweight cryptographic suites: ECC-based encryption, ABE, identity-based cryptography, and encryption, AES, DESL, PRESENT, Twine, HLA.<br>‣ Blockchain offers a scalable, resilient and reliable approach for ensuring the integrity of IoT data (FlowFence). |
| Concerns | Solutions |

**Figure 2.6:** Integrity summary.

in Figure 2.7. The former principles allow the latter, and the latter demand the former. Nevertheless, the engineering considerations must be observed separately for each principle, and the specified order should be followed when integrating the principles. First, by engineering the mechanisms to ensure credibility, we ensure that the trustworthiness of the network increases the effectiveness of communication. By addressing Authentication and Authorization, we ensure that there are special, verifiable credentials for each node allowing them to communicate and access data. Engineering of concepts behind Verification protects the system from internal misbehavior and external malicious acts by ensuring system behavior is reliable, consistent, and controlled on an ongoing basis. Lastly, data integrity is obtained in part from the correct engineering of CAAV principles, but also mechanisms at the integrity layer themselves ensure data security, privacy, reliability, completeness, accuracy, and consistency. Designing and bridging CAAVI principles is critical to system security engineering, as the (in)correct design of an earlier principle impacts the latter significantly.

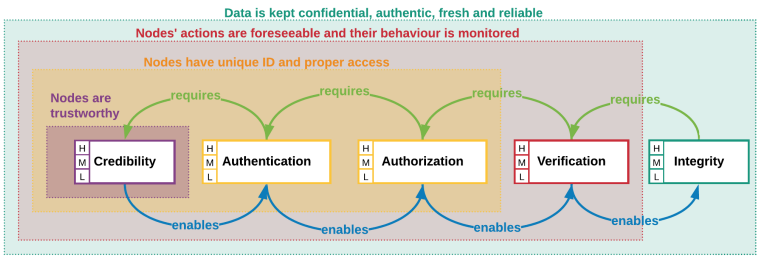



**Figure 2.7:** CAAVI principles bridging.

Furthermore, it is important to highlight that the level of security for each principle can be modeled separately. For example, an IoT system operating in the context of a hospital, having access to patients' health record and information private under doctor-patient confidential and/or HIPPA, would need a very strong design behind each of the CAAVI aspects. On the other hand, a home automation system for turning the light on/off would need far less complex security mechanisms and data integrity checks. Thus, the level of security inside a system heavily depends on the system and use-case description: criticality, time sensitivity, and reactiveness requirements, data sensitivity and data/user/device privacy requirements, available computational resources, architectural and deployment approach, and the physical and logical operational contextual information availability.

# Chapter 3

# An MQTT-based Resource Management Framework

In this chapter a Resource Management Framework for edge computing systems handling fail-rescue of software components, infrastructure and software health checking and platform workload re-distribution is described. The RMF with workflows and fail-recover scenarios described in the following sections is a generic framework best-suited to IoT platforms with event-driven, publish-subscribe communication mechanisms. As such, the framework will be presented in the operational context of one such IoT/ECP system.

The work in this chapter is structured as follows: Section 3.1 deepens the topic of why context-aware resource management frameworks are needed for IoT/ECP systems; Section 3.2 underlines current SoTa and research gaps related to Section 3.1; Section 3.3 presents a functional architecture of an MQTT-based RMF specifically designed for reactive IoT/ECP systems, and also an original scientific contribution of this thesis; Section 3.4 elaborates on the standard operational workflows of the RMW explained in Section 3.3; Section 3.5 presents a subset of fail-recover scenarios inside the RMF; finally, Section 3.6 concludes the chapter with showcasing a simulated experiment and initial results of the RMF.

## 3.1 Context-aware Resource Management at the Edge

Much of the data in the IoT is processed using cloud computing resources. However, data provides the most value when it is interacted with in real-time. Cloud computing is good for offline analysis of large data sets which provides the basis for proactive management of processes and resources. However, apart from this strategic nature of data analysis, most systems and processes in the IoT domain require real-time actions and cannot risk delays introduced by transferring

all the data to a cloud platform for analysis. That is why data processing functionalities are being moved to the edge, which represents a computing paradigm better known as edge computing (ECP). This leads to providing real-time actionable insights that transform into major business benefits. The complexity of IoT systems and tasks that are put before them require shifts in the way resources and service provisioning are managed. The concept of edge computing is introduced to enhance IoT systems' scalability, reactivity, efficiency, and privacy. The state-of-the-art in resource management for edge computing systems mainly focuses either on smart actuation enabled through insightful data analysis and machine learning or on managing the edge system itself to improve performance and efficiency. The proposed architectural solution showcases how one software architecture can be used to achieve both. Proof of concept experiments that are executed on a real-world edge computing testbed validate this solution's performance in improving the resilience and responsiveness of the edge computing system when there are operational context and topology changes. Furthermore, the framework addresses the recovery of failed decision-making processes as well, impacting the overall health of the underlying system.

Network management approaches such as software-defined networking, and advanced routing protocols provide a certain level of robustness. If the IoT gateways only perform data formatting and cloud platform bridging, then these network management solutions are very effective in overcoming topology disruptions. But when gateways are responsible for data analytics and reactive decision-making, what is going to happen? Load balancing and handover workflows allow the topology to be adapted as a result of node or communication link failures, to compensate for the failed processes. Failed decision-making processes cannot be compensated that easily. One solution is a mechanism for failover towards the cloud platform. In particular for edge computing systems that rely on data privacy and low latency, this is not always possible nor desirable. The edge computing systems need a solution that enables participating nodes to take over not only data streams from their failed peers but also data analytics and decision-making processes.

The complexity of IoT systems and tasks that are put before them require shifts in the way resources and service provisioning are managed. Network management approaches like software-defined networking and advanced routing protocols provide a certain level of robustness. If the network of IoT gateways (GWs) performs only data formatting and bridging to cloud resources, then these network management solutions are very effective in overcoming topology disturbances. However, when IoT GWs are responsible for reactive decision-making (DM) directly influencing the operational environment of the system these network management solutions are ineffective in keeping the system functionalities *alive* (i.e. address failed DM/actuation processes). One solution to this problem would be to provide a failover mechanism towards cloud resources. However, that is not always a possibility

nor is it desirable for ECP systems since they rely on data privacy and low latency. Therefore, ECP systems need a solution that allows participating nodes to reinstate not only failed/missing data streams from their peers but also data-analytics (DA) and DM processes. In this thesis, the author is proposing a Resource Management Framework (RMF) for ECP systems that provides a decentralized solution to this challenge. The RMF is tested in a simulated environment.

In this chapter, an edge computing solution for resource management of context-aware decision-making processes distributed between IoT gateways is presented. The solution performs decision-making process management for smart actuation, based on analysis of sensory data streams, and context-informed edge computing resource and service provisioning management based on topology and operational changes. Our architectural solution showcases the first version of a Resource Management Framework – a generic framework for software resource orchestration best-suited to IoT platforms with event-driven, publish-subscribe communication mechanisms. Proof of concept experiments that are executed in a simulated edge computing testbed validate the solution's performance in improving the resilience and responsiveness of the edge computing system when there are operational and topology changes. Furthermore, the framework addresses the recovery of failed decision-making processes, impacting the overall health of the underlying IoT system.

## 3.2 Current SoTA and Research Gaps

As state-of-the-art goes, research papers addressing resource management can be divided into two groups: (1) addressing a lower-level (specific) solution, and (2) addressing a high-level architectural approach. Papers from group (1) can be further categorized into resource management techniques focusing on (a) storage, (b) network, (c) computational resources, and (d) software components. This thesis describes a RMF from a higher-level standpoint (2), focusing on a re-distribution approach for software components of the underlying IoT/ECP architecture (d), while approaches used for (a), (b) and (c) are not discussed in this thesis.

Papers addressing solutions from group (2) include container-based allocation of resources [165], infrastructure health audit [127], etc. Solutions for resource management (1) in IoT/ECP, in categories (a), (b), and (c) are not lacking. Kim et al. describe an efficient XML-based classification scheme for data storing/processing of heterogeneous data for IoT [86]. Sehgal et al. proposed a set of IP-based network management protocols implemented on resource-constrained devices [178]. Many research papers are addressing computational resource allocation and offloading management for IoT/ECP systems to improve their performance [105, 129, 17]. Hong et al. presented a thorough survey of such approaches [68].

On the other hand, solutions managing failing software components and logical components of ECP architectures are under-researched. In their study, Alreshidi et al. underlined that future research on architecting logical components (software) of IoT platforms will be focused on architectural patterns that support automation and reusability to dynamically adapt IoT software [4]. As a movement towards solving a part of that challenge, Leiba et al. use blockchain, smart contracts, and Zero-knowledge proof to build a decentralized network for the propagation of IoT software update [101]. However, secure software updates are a small part of the bigger problem that is fail-rescue strategies of processes inside an IoT platform.

Taking into account the above-mentioned research, this thesis focuses on architecting a high-level resource management framework for re-distribution of running DA and DM processes throughout an event-driven, publish-subscribe communication mechanism-based IoT platform. To the best of this author's knowledge, this is the first solution focusing exclusively on the Message Queuing Telemetry Transport (MQTT) protocol. Thus, the main contributions with addressing such a framework rest in (1) providing a higher-level RMF for the IoT based on a lightweight communication protocol (MQTT), as contrary to the above-mentioned papers that are focused on a specific resource management task; (2) providing step-by-step detailed description for resolving sophisticated failure scenarios; (3) discussing recover scenarios not only for failure but also in workload re-distribution to achieve optimal operational capacity.

## 3.3 Functional Architecture

The high-level architecture of the proposed resource management framework for recovering ECP systems is displayed in Figure 3.1.

The actors of the RMF can be categorized into physical and software components. Among the physical components, there are typical IoT platform-connected hardware: sensors, actuators, and devices. Sensors and Actuators can be connected to multiple Devices by wire and/or wirelessly – e.g. Actuators 1 and 2 are connected wirelessly to Device 1 and connected by a wire to Device 2. A Device has a hardware and a software stack – hardware stack describes the set of wireless technologies the Device is capable of communicating on and the software stack describes the set of software components (IoT platform and RMF related) deployed on the Device.

The software stack for each Device consists of two groups of software modules: (1) providing IoT platform functionalities (multiple soft sensors, the MQTT broker), and (2) providing RMF functionalities (Controller, the Soft Sensors Controller Module (SSCM) and the System Control Table (SCT)).

In the observed IoT platform, the soft sensors approach is used to enable scalable and hierarchically distributed DA and DM processes across the platform and

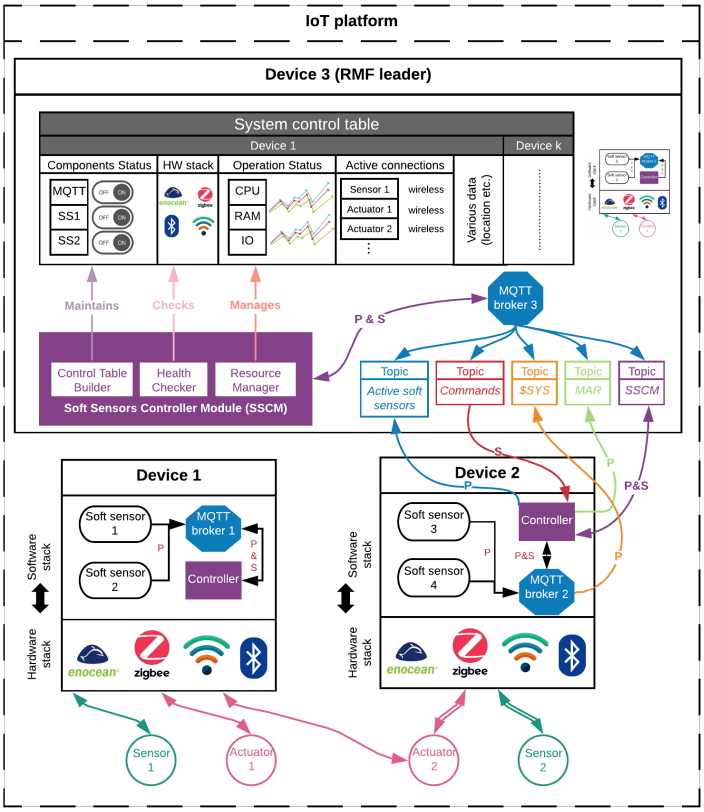



**Figure 3.1:** Resource Management Framework high-level workflow.

the RMF as well [47, 192]. Soft sensors represent self-contained software modules performing simple analytical processes and exposing services and/or data via MQTT. They are hierarchically deployed which allows the creation of analytical chains where each step in the DA chain provides deeper insight into the context of the managed IoT system. 1st layer soft sensors are deployed right on top of physical sensors and actuators, 2nd layer soft sensors read, preprocess, and selectively forward data to interested parties, 3rd layer soft sensors perform aggregation, and finally 4th layer soft sensors perform DM.

Publish/subscribe mechanisms are a suitable choice for machine-to-machine correspondence due to loose coupling and simple communication architecture [81]. One of the widely used communication protocols is the MQTT protocol which transports messages between multiple clients through a central broker. It has a strong footing in IoT and ECP systems [183, 137]. The RMF's default communication protocol is MQTT, handling all communication flows: soft sensor to soft

sensor, soft sensor to device and device to device (and ultimately platform to platform). In the RMF there are 5 critical topics:

- **Active soft sensors** – used to propagate information about soft sensors failure/recovery to the SSCM
- **Commands** – used to propagate commands to and from RMF devices
- **$SYS** – used to propagate information about device and communications operation (device CPU, RAM, storage, MQTT broker status, etc.)
- **MAR i.e. Manual Action Required** – used to propagate information to the platform administrator that the RMF did not resolve a failure issue itself and manual action is required
- **SSCM** – used to distribute the SCT from RMF leader to other devices

A Controller module is a critical functioning part of the RMF. The soft sensors approach in deploying DA and DM mechanisms ensures high independence of the ECP system. To separate the RMF management functionalities from the underlining IoT platform, the Controller module handles all RMF-related communication: failed processes notification, command propagation, RMF leader voting, etc.

The SSCM is in charge of three operational tasks: (1) Building the system's control table, (2) Running continuous health checks, and (3) handle processes failure, redistribution, and ensuring fail-safe system operation. Throughout the operation, if the IoT platform, the Control Table Builder (1) maintains the system's control table – persisting and continuously updating information about the status of the entire IoT platform per device: running physical components and soft sensors, MQTT broker status, etc. Health Checker (2) updates the same control table with resource consumption information on the one hand, and on the other runs regular workload balance checks and high-risk analysis of potentially failing devices. The Resource Manager (3) does not perform DA but serves as a proxy to issue two types of commands: (a) actions based on the results of the analysis from (2) Health Checker and (b) reactions to failed soft sensors information that reaches the SSCM from the IoT platform belonging devices.

## 3.4 Standard Operational Workflows

RMF operational workflows are categorized into 5 groups, depending on their role in the RMF. These groups of workflows are (1) SCT update and propagation, (2) RMF Leader voting and role switch, (3) System health status data collection, (4) System health and workload distribution analysis and (5) Fail-recover scenarios handling. These groups and belonging processes are displayed in Figure 3.2. It is

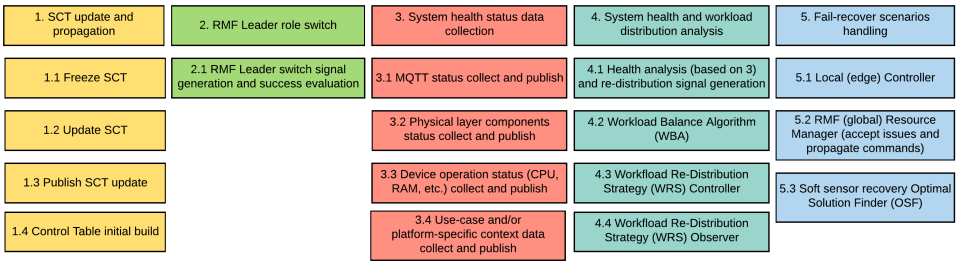



**Figure 3.2:** RMF standard operational workflow grouping.

important to highlight that every process inside RMF runs as a soft sensor. Thus, RMF processes that fail are also recovered with other RMF-established workflows. In the rest of this Section, the RMF-specific workflows will be elaborated.

**1. SCT update and propagation** – Processes from this group are in charge of manipulating the SCT, which includes freezing the table (table does not take in updates), updating the table's contents, publishing SCT updates via MQTT to IoT Controllers and a process to gather initial SCT information once the RMF is setup. SCT is distributed to all devices every $k$ seconds;

**2. RMF Leader role switch** – RMF leadership role is awarded to devices of the RMF in a round-robin fashion. The Controller script of the RMF leader issues a command for a role switch every $k$ seconds. The success of that command is evaluated through analyzing SCT updates (whether RMF role status was updated).

**3. System health status data collection** – A set of processes enabling context-building inside the RMF through collecting operational status information about running communications (MQTT brokers and bridges), device operation status (CPU/RAM/Storage usage), physical component status (active sensor connections, communication channels, etc.)

**4. System health and workload distribution analysis** – Processes in this group are in charge of ensuring the health and optimal operation of the underlying edge computing platform. This is achieved through constant cross-referencing of current health parameters with defined optimal ones (by platform administrator) and reacting when a deviation is detected. WBA, WRS Controller, and WRS Observer handle activities related to executing and evaluating the success of re-distribution strategies. Health Checker performs regular health checks of all system components every $k$ seconds;

**5. Fail-recover scenarios handling** – A set of processes covering activities for recovering failed components of the underlying edge computing platform. For enabling recovery, all Devices host all soft sensors that they are capable of running (technology and access compatibility can be a problem for $1^{st}$ layer soft sensors). All soft sensors send keep-alive pings to the Controller every $k$ seconds. Controller

(local) and Resource Manager (global) immediately react to all Publish/Subscribe actions triggering other RMF functionalities.

The RMF operates as an integral part of an IoT platform. However, on top of RMF a Platform administrator management portal is envisioned to enable IoT platform administrators to issue manual commands through the RMF when needed. Also, there is a time limit for all fail-rescue scenarios predefined by platform administrators. If these limits are exceeded the Manual Action Requirement event is triggered and the platform administrator is notified.

## 3.5 Fail-recover Scenarios Description and Handling

To fully understand the RMF and its inner workings this section will present a detailed workflow from the standpoint of how RMF handles recovery of failed processes and devices and workload re-distribution. Our proposed RMF addresses 9 fail-recover scenarios:

T1. RMF leader switch;

T2. Soft sensor failure;

T3. MQTT broker failure;

T4/T5/T6. CPU/RAM/Storage peak;

T7. RMF leader failure;

T8. Controller failure;

T9. Device failure.

To visualize how RMF handles these scenarios, three scenarios will be explained and visualized. Steps taken from the moment the scenarios are triggered up to when they are resolved are displayed in Figure 3.3. Software specification for all other triggers/scenarios (T1–T9) is available in the RMF as well, but will not be further discussed in this thesis.

**Scenario 1** covers the failure of a soft sensor of any hierarchical layer (T2. Soft sensor failure). As soon as the Controller stops receiving keep-alive pings from a failed soft sensor (step 2) a *FAIL* message is posted to the *Active soft sensors* topic on a local MQTT broker (step 3). This message is immediately transferred to all remote MQTT brokers bridged to this MQTT broker. The message is received by the Control Table Builder (CTB) which updates the SCT. The Resource Manager of the Device that is currently the RMF leader will receive this message as well

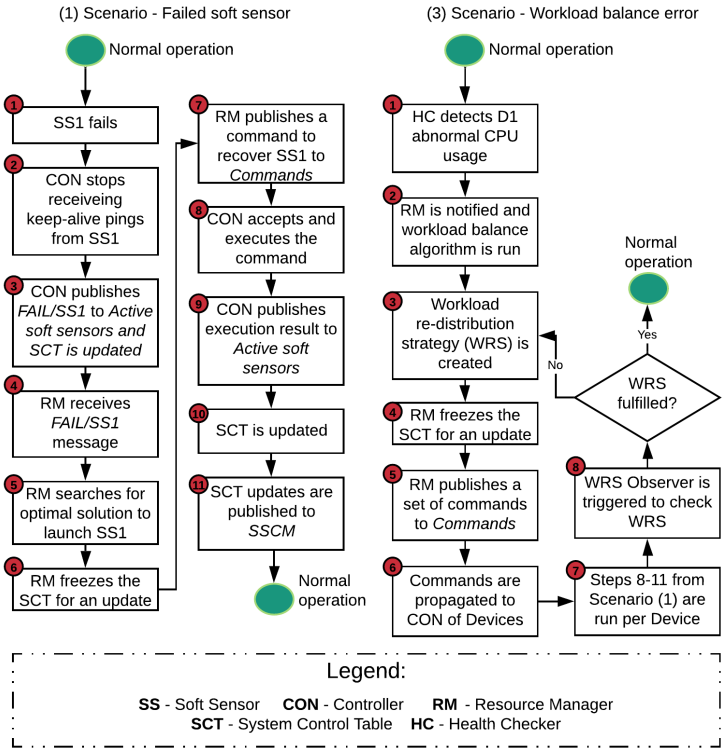



**Figure 3.3:** Example: Fail-recover scenarios handling inside RMF.

(step 4) and will search for the optimal solution for recovering the failed soft sensor (step 5). The optimal solution is found in 2 steps: (a) find all devices capable of running failed soft sensor; (b) launch the failed soft sensors on the device with lowest average workload (CPU, storage, and RAM utilization). Once the solution has been obtained, the SCT is frozen and no other topology updates are received at this point (step 6). The Resource Manager will propagate the solution in a form of a system command, e.g. 'start $\text{SoftSensor}_\text{M}$ on $\text{Device}_\text{N}$', to the *Commands* topic (step 7). This information is published to all subscribed Controllers through MQTT bridges. The Controller of $\text{Device}_\text{N}$ is the only one that will react to this message and execute the system command (step 8). The execution of the command will be repeated after failure. If the command executed successfully the Controller posts an *ACTIVE* message to the *Active soft sensors* topic referencing $\text{SoftSensor}_\text{M}$ (step 9). The Control Table Builder of the RMF leader will update the SCT (step 10) and this update is published to the *SSCM* topic (step 11). All devices are subscribed to the SSCM topic and this update is propagated to their local copies of SCT.

**Scenario 2** covers the failure of a Device in the observed infrastructure (T9. Device failure). This scenario is not shown in Figure 3.3 since it is a repetition

of the Scenario 1 workflow per device. In this version of the RMF, a brute-force approach is envisioned for recovering of soft sensors – the next version of the RMF will have to consider workload balancing at this point.

**Scenario 3** is triggered when the Health Checker (HC) of the RMF leader device detects a deviation in a device's behavior (step 1), e.g. abnormal CPU usage on $\text{Device}_\text{N}$ (T4/T5. CPU/RAM peak). This and similar behavior deviations can happen due to many reasons: malfunction, physical tampering, malicious attacks, etc. However, the RMF adopts a strict policy to react first and notify later. This is done so that the soft sensors and platform business processes can converge quickly, while the platform administrator is notified of the problem with a device to manually check out the issue and issue manual workload redistribution commands if needed later. Resource Manager of RMF leader is notified through the *Commands* topic that running of the Workload Balance Algorithm (WBA) is required for $\text{Device}_\text{N}$ (step 2). WBA analyzes the running soft sensors on $\text{Device}_\text{N}$ and all other devices capable of running 1 or more soft sensors of $\text{Device}_\text{N}$. Based on this analysis the WBA outputs (step 3) a Workload Re-distribution Strategy (WRS) and creates two new controlling scripts to take care of the execution of the WRS: (a) WRS Controller and (b) WRS Observer. The WRS controller is in charge of executing the WRS through communicating with the RM to issue commands to appropriate devices. The WRS controller will trigger the WRS Observer when the WRS has been implemented. Since SCT is about to be updated shortly, it is frozen, as before, to prevent other actions (step 4). Based on the WRS the Resource Manager of RMF leader issues a set of system commands to stop/start soft sensors on different devices, through the *Commands* topic (step 5). Controllers of all devices affected by the WRS will receive these system commands, execute them, and publish new information to *Active Soft Sensors*, thus updating the SCT. These updates are pushed to all devices through the *SSCM* topic (steps 6, 7). Upon the completion of the WRS, the WRS Controller notifies the WRS Observer. The WRS Observer analyzes the WRS outputs versus the state in the SCT and concludes whether the WRS has been implemented successfully (step 8). Based on that analysis, a decision is made whether a new WRS needs to be created to compensate for the eventual problems (returning to step 3). This loop of events will execute a maximum 3 times before a MAR event is fired.

## 3.6 Experiments, Results and Discussion

Proof-of-concept experiments are carried out in a simulated environment (written in Python) with 50 devices with Raspberry Pi 4 capabilities. Each simulated physical device has a random number of soft sensors (2–50), and a random distribution of connected physical sensors (2–10). An event (T2–T9) is triggered every 60-90

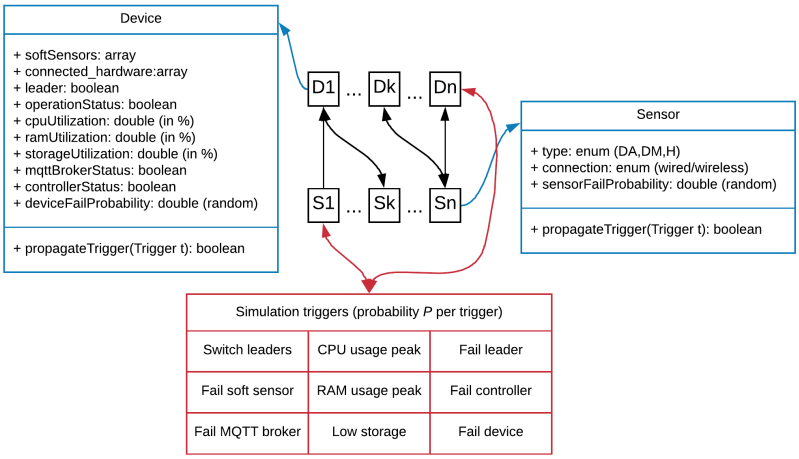



**Figure 3.4:** RMF simulation high-level overview.

seconds and executes with a $P$ probability (per trigger), while the simulation runs non-stop – meaning that every 90 seconds a trigger might execute.

RMF leadership role switch (T1) is triggered every 300 seconds. $P$ for events T2 and T3 is set to 60%, for T4-T6 to 30% and T7-T9 to 10%. This is set up empirically, considering the possibilities for these types of event to occur in a real-world physical environment.

The simulation was programmed to reset after every fail-recover scenario. A high-level architecture of the simulation is displayed in Figure 3.4. As part of this simulation, verification that RMF successfully reacts to scenarios described in Section 3.5 is carried out.

The simulation was run for 72 hours non-stop. The results of the simulation are recorded in Table 3.1.

Although the simulation was short, results are affirmative of the effectiveness of the proposed RMF solution.

Numbers in the last row of Table 3.1 are rounded to the higher integer. The average number of triggers executed is 245, and a total of executed triggers is 1715. In 36% of cases the trigger was not executed in the simulation framework due to specified probabilities. As an upside, on average, for every 245 triggers propagated, 241 were successfully resolved by the RMF, leading up to 98% of autonomously resolved issues inside the observed simulated IoT platform.

However, several downsides need to be accounted for. The total number of manual actions required (MAR) over the 72 hours was 26 – this number is still high for an RMF to be considered automated and autonomous. In the simulation MAR was in most cases caused by 2 issues: **Case (1)** RMF leadership role change signal synced with the trigger causing a failure to update the SCT prop-

**Table 3.1:** RMF 72-hour simulation results.

| Trigger type | Triggered no. | Resolved no. | MAR no. | Average delay (ms) |
|---|---|---|---|---|
| T1 | 299 | 295 | 4 | 1204 |
| T2 | 214 | 204 | 10 | 2892 |
| T3 | 260 | 255 | 5 | 989 |
| T4/T5/T6 | 267 | 267 | 0 | 6620 |
| T7 | 210 | 210 | 0 | 2476 |
| T8 | 215 | 211 | 4 | 1001 |
| T9 | 249 | 246 | 3 | 1705 |
| **Average** | **245** | **241** | **4** | **2412** |

erly; **Case (2)** Multiple device failure occurs and a 1$^{st}$ layer soft sensor cannot be started due to missing connectivity with other viable devices. While case (2) is out of RMF's reach and impact, case (1) has to be accounted for. Next, triggers T4/T5/T6 are associated with Health Checker (HC) and Workload Balance Algorithm (WBA). Although they were resolved in all of the cases the average delay is much higher than for other triggers. That is caused by the loop in the scenario handling strategy – only 29% of triggers T4/T5/T6 were resolved in the first loop, 60% in the second and 11% in the final (third) loop. This resulted in a high average delay, yielding a high average delay in total (3915ms). Scenarios handling triggers T4/T5/T6 need to be handled with more sophistication and WBA needs to be re-calibrated to create finer Workflow Re-distribution Strategies (WRS). Since WBA currently checks devices capable of running the failed components in a sequential manner, adding randomness to this process could yield positive results. Furthermore, by allowing flexibility in the system's optimal health parameters, e.g. an allowed percentage of increase for CPU/RAM. Finally, in general, the average delays for all triggers are high for an industry-grade IoT platform and need to be cut approximately ten-fold to be considered for such platforms, especially for critical use-cases [2, 136].

As part of the future work there will be focus on increasing the sophistication of the RMF through three aspects: (1) inspect and reduce MAR, (2) refine the Workload Balance Algorithm and Workload Re-distribution Strategy Controller and Observer components, and (3) work to reduce RMF's average delay mainly through (2) since redistribution issues cause highest delays. A milestone in RMF implementation and a part of the future work is the integration of RMF with a commercial IoT platform.

# Part II

# BLEMAT: Bluetooth Low-Energy Microlocation Asset Tracking Edge Computing Software Architecture

# Chapter 4

# Indoor Positioning: Introduction

Indoor positioning is a widespread term referring to location determination workflows/systems that operate on top of an infrastructure of devices deployed in a closed space. To make the location information in such systems useful, the location information is typically fed into some kind of application software. Indoor positioning technologies enable a number of location-based solutions, including real-time location tracking, wayfinding, inventory management, and localization systems for first responders.

The work in this chapter is structured as follows: Section 4.1 gives common definitions for indoor positioning and indoor positioning systems; Section 4.2 provides an overview of most commonly used indoor positioning techniques and analyzes the required hardware and software components; Section 4.3 distinguishes typical deployment options for indoor positioning systems, and underlines advantages and shortcomings for each; Section 4.4 covers specific design requirements and challenges for implementing indoor positioning systems in edge computing environments; Finally, Section 4.5 discusses the related work and research gaps in positioning techniques, filtering of positioning data and deployment options for indoor positioning systems.

## 4.1 Definitions

Indoor positioning is an umbrella term referring to a variety of systems and services that provide location-based services (LBS). An indoor positioning system (IPS) represents a network of devices that are used to locate people or objects where GPS and other satellite systems lose accuracy or fail, such as within multistory houses, airports, parking garages, and underground areas. A broad range of techniques and technologies are used to provide indoor positioning ranging from already installed reconfigured technologies such as smartphones, WiFi and Bluetooth antennas, digital cameras, and clocks; to specifically built installations with

strategically positioned relays and beacons within a specified space. The easiest way to explain IPS is that it is like GPS for indoor environments. IPS may be used to identify individuals or items inside houses, usually via a mobile device, such as a tablet or a smartphone. While the technology is newer than GPS, services using IPS are quickly gaining popularity in places such as shopping malls, hospitals, airports, and other indoor locations where navigation and other LBS have proven to be indispensable. Using advanced mathematical algorithms, IPS technology leverages internal sensors in smartphones to measure the device's inner location. Through cleverly integrating the incoming data from these sensors, a very precise location can be measured, with little or no delay, resulting in seamless user experience.

IPS can provide indoor navigation and/or location tracking and these two use-cases differ a lot in terms of deployment, capabilities, and computational complexity. Indoor navigation means directing people's location in challenging, unfamiliar and complex buildings. The user of an indoor navigation system on his/her smart device gets his/her location displayed on a map. On the map, a path to the chosen destination is shown after selecting a destination or point of interest. Indoor navigation service would continuously update the user's location in and around the house so that the app always displays the current position on the road. The effect is indoor navigation with turn-by-turn. Indoor navigation systems or way-finding apps are used at railway stations and airports, in office buildings and hospitals, shopping centers, multi-functional areas, and sports stadiums.

Location tracking refers to acquiring the current and/or continuous position of a user/object in the observed space. It is commonly used in a range of use-cases from mobile/static equipment tracking and geofencing to inferring and predicting positions and routes of the observed indoor space users (people).

## 4.2 Positioning Techniques

Indoor positioning technologies fit into the four main categories: proximity, trilateration, fingerprinting, and motion. Most of these techniques can be used independently but can also be mixed to achieve greater precision. Detecting user/object proximity is dependent on either direct contact or proximity between a receiver and a computer. Much of the time, except for Wi-Fi, it is used on the network side and has server-side monitoring capability. There is a range of possible solutions:

- QR Codes, NFC or RFID tags can be read by smartphones of NFC/RFID readers. Stickers containing QR codes, NFC stickers, or RFID tags are linked to a precise location in the building.

- Bluetooth Low Energy (BLE) beacons send a signal which can be read while the device is in the region of the BLE beacon signal emission. The more

beacons deployed, the more accurate the location would be – user position is tied to the location of the beacon with the strongest signal.

- Visible Light Communication (VLC) is also called Li-Fi. For example, an LED lamp can send an invisible signal that can be perceived either by a smartphone camera or by a dedicated receiver.
- Wi-Fi Access Points (WAPs) can be used to locate any Wi-Fi connected device. The user position is determined in the same manner as for BLE devices.
- With ultrasound devices, that are typically deployed on top of existing audio systems (in stores or shopping malls), proximity location is deduced the same manner as for BLE and Wi-Fi.

Trilateration positioning uses signal strengths computed as distances between several emitters and a receiver, to calculate the position of the receiver. The distances are either calculated by Relative Signal Strength Indication (RSSI) or Time of Flight (ToF) based algorithms. RSSI enables measurement of distance based on attenuation of the radio wave that follows the Inverse-Square physical law. Calculating distances does not require complicated calculations, but the accuracy is low due to signal degradation – the sensitivity of the signal to physical obstacles like walls, doors, people, and other signal interference on the same frequency. ToF measures the distance between an emitter and a receiver based on the time difference between a signal's emission and its return to the sender. ToF includes roundtrip communication and complex signal processing that requires dedicated chipsets as opposed to RSSI. ToF results in better accuracy and a higher computational cost. Trilateration is typically coupled with the following technologies:

- BLE and ultrasound-based technology, although classified primarily as proximity technologies, can be used for trilateration-based positioning. Distances are typically computed on the client-side (e.g. smartphone) using RSSI.
- Wi-Fi can also use trilateration on the client-side (smartphone), or the server-side. The latter is much more reliable, but these location services are only available on rather expensive Wi-Fi (e.g. Cisco CMX or Cisco Meraki) equipment. Normally, distances are calculated using RSSI but shortly an expansion of the 802.11 standards will have ToF support.
- Ultra-Wide Band is another new technology that gives very precise positioning information, thanks to the use of ToF to measure distances between receiver and emitter. Nonetheless, there are many disadvantages to this technology: lack of standardization, smartphones are typically not equipped with UWB, high costs.

Technologies for fingerprinting positioning use signal measurements across buildings to determine one's location. It implies that there is an almost unique signal which can be registered for every position. Then, to work out the current location, it is possible to equate the currently received signals with existing records. Fingerprinting techniques can be used either alone or to boost the precision of other positioning technologies. Besides, fingerprinting from several sources may be combined simultaneously to increase the accuracy (e.g. Wi-Fi plus BLE signals). Fingerprinting requires a large overhead in manual system deployment due to fingerprints collection. Fingerprinting can be used as:

- BLE and Wi-Fi fingerprinting is most used in industrial location tracking systems. It requires spectral image scanning of the indoor space and a database for their storage.

- Magnetic positioning-based fingerprinting relies on each building or structure having a specific magnetic fingerprint, based on how the building materials influence and distort the Earth's otherwise permanent magnetic field. Those patterns may be explicitly applied to a floor plan for a house. In department stores, schools, malls, airports, and other indoor areas, mobile owners can then be precisely located. Magnetic positioning can maintain 1–2 meters in indoor spaces.

- Photo fingerprinting requires image processing of the interior of the building. It works well when there are significant distinctions between floors and furnishings. It does not prove reliable overtime except when fingerprinting comes from permanent high-resolution cameras able to refresh records regularly.

Motion positioning inside closed spaces operates on the same concepts but with different technologies. Since conventional Inertial Measurement Units (IMUs) cannot be used, smartphone sensors are used for the identification and quantification of movements (compass, accelerometers, barometers, pedometers). Algorithms such as Kalman filters process data that are used to measure relative movement from those sensors. The problem with such techniques is low accuracy due to small size sensors and accumulated errors.

## 4.3 IPS Deployment Types

Deployment of radio-based IPSs traditionally requires the deployment of multiple fixed pieces of equipment transmitting a signal (emitters), such as a Bluetooth beacon. They are physically dispersed at the observed closed space, and the device that is being tracked (e.g. smartphone) is the midpoint of the system (scanner)

– it is responsible for beacon scanning, carrying out calculations and in the end positioning itself on the space map [44, 189, 43, 103, 200]. This is a typical deployment for large shopping malls, stadiums, etc. since the deployment of beacons on a large scale is rather cheap, and it is safe to rely upon that nowadays every user device (smartphone, even smartwatches) has the capabilities to perform location determination calculations. However, this approach has several issues:

1. It often includes an excessive fingerprinting phase, to create a spectral map throughout the space. For large spaces, this is an enormous manual task.

2. RF-based pieces of equipment (e.g. beacons) that are deployed are at a fixed point in the indoor space, and they cannot be reconfigured easily.

3. System recalibration can take a long time to finish and requires a considerable amount of downtime (i.e. change the advertising interval of every beacon, create a new fingerprint map, etc.)

4. System cannot adapt to contextual changes in the managed environment (i.e. different flow of people throughout a day/week/year, new physical obstacles, etc.)

In IoT edge gateway devices are abundant, which can serve as radio signal scanners while performing other IoT functions in the system they are a part of (e.g. SBCs such as Raspberry Pi). If they are used for signal scanning, then the emitter can be any device capable of sending radio signals (Wi-Fi, Bluetooth, ZigBee, or any other). Edge devices such as a Raspberry Pi are resource-capable and can perform advanced filtering, calculations, even small-scale machine learning tasks. Furthermore, they can be reconfigured easily, networked in a mesh-like topology to handle a large number of requests, and protect data and increase data privacy at the edge level. An IoT IPS could, thus, rely on edge devices to serve as radio signal scanners, bringing important advantages over the above-mentioned deployment topology:

1. **Positioning calculations** happen on the scanner devices, rather than user-devices or other objects being tracked, making those devices unburdened with any calculations. The calculations happen in a controlled, edge environment, increasing data handling and privacy capabilities, as well as system reactivity.

2. **Seamless communication** – scanners are not just used to perform IPS related calculations, but can as well be used in other IoT functions as data aggregators and processors, as well as actuators. They can seamlessly communicate with other systems and hardware (e.g. sensors and actuators).

3. **Manual fingerprinting** can be avoided if the system resorts to detecting the tracked asset's proximity to the scanners. When a tracked asset is close to a scanner its immediate location in the space is known by comparing its fingerprints to he scanner's fingerprints (without performing additional calculations). For fingerprint matching, machine learning models can be trained – to match fingerprints of a tracked asset at a specific time to a specific scanner. The signal spectral map of the indoor space is calculated between the scanners periodically, making space context updates reactive to changes in the environment. Then, positioning techniques can then be used as a secondary source to increase the accuracy of the system and pinpoint location. Since proximity has been established before, in this case, closest scanners can be used to achieve maximum accuracy. Scanners can, individually, control each aspect of the system at every point in time – they can change scanning parameters as well as enact system-wide updates.

4. **System re-calibration** is carried out as a background process with no downtime. This includes re-scanning the indoor space to detect new scanners and learning their fingerprints to be used for fingerprinting-based proximity detection. The system can work with previously set parameters and can update as soon as the re-calibration process had finished.

## 4.4 Requirements and Challenges

Indoor positioning is a difficult problem, as well-established and robust outdoor positioning techniques do not work inside buildings. GPS waves can not penetrate receivers properly due to walls and roofs, resulting in a significant degradation in the localization approximation that may exceed 10 meters.

Performing accurate indoor location determination presents with a set of challenges: choice of technology and signal characteristics, pre-deployment effort, selection of filtering algorithms, and selection of machine learning algorithms.

Most current solutions are based on Wi-Fi or Bluetooth. Both of these technologies operate on a 2.4 GHz frequency band that is highly susceptible to noise and interference. Bluetooth uses radio frequencies (RF) to send signals between devices wirelessly. The presence of humans, metal objects, or other obstacles or RF reflective surfaces causes disturbance in signal propagation. Other electrical equipment that emits strong RFs may do the same. Because Wi-Fi uses the same 2.4 GHz bandwidth, these two signals often interfere with each other as well. Besides, the geolocation calculated based on Wi-Fi or Bluetooth signal propagation may be incorrect for several reasons: signal disturbance and obstruction, multipath signal distortion, and hardware or software device design.

In terms of pre-deployment efforts, several issues are connected to IPS. Deployment of indoor positioning and asset tracking systems traditionally requires multiple fixed beacons to be stationed in the observed indoor space, where the device being tracked is the midpoint of the system – it is responsible for beacon scanning, computations, and ultimately positioning itself on the space map [44, 189, 43, 103, 200]. This approach has several problems: it often involves an excessive fingerprinting phase to create a spectral map across space; the beacons deployed are fixed and can not be re-configured or changed easily; the re-calibration of the system can take a long time to complete and requires a considerable amount of downtime (i.e. changing the advertising interval of each beacon, creating a new fingerprint map, etc.).

Specifying a fitting filtering model for signal perturbation is a well-researched topic and there are many existing approaches [189, 103, 106]. Although there are many popular and widely-used filtering approaches for location determination: averages, moving averages, Kalman filtering, auto-regressive moving averages, etc., it is up to the use-case requirements at hand to provide the most relevant information on the best filtering approach. Besides, it is usually the case that many methods of filtering are combined to reach a final position estimation. These combinations need to be handled carefully. There are two distinct approaches in filtering data for location determination that can be considered: heuristic and statistical filtering. If there is a domain knowledge of the use-case, heuristic rules can be used to filter data and thus infer thresholds for accuracy, location age, speed and location time, etc. Alternatively, statistical filtering, which involves estimating the location based on historical data (location time-series) is used. Recursive Bayesian estimator, Kalman filter, Particle filter are representatives of the statistical filtering approach for IPSs [20].

With recognizing the importance and advantages of machine-learning approaches in indoor positioning, there is a challenge in modeling an approach that can best work with the proposed filtering models while overcoming the problems of constant signal deviances. Based on data volume and structure, appropriate supervised and unsupervised machine learning algorithms must be selected. While supervised methods may be used to perform fingerprinting or area presence detection, unsupervised methods may be used to detect clusters of locations and to infer the user presence and movement information. Keeping in mind that traditional fingerprinting requires a big pre-deployment effort, there is also a challenge in eliminating or diminishing the time required to carry out this step in IPS.

## 4.5 Current SoTA and Research Gaps

In the positioning literature, machine learning algorithms have widespread usage in estimating positions [113]. To be able to guarantee high location estimation

accuracy and precision, most machine learning algorithms require a large number of carefully labeled samples [207, 19].

Academic research has shown that IPS can not be accurate without contextual knowledge and aggressive filtering. Li et al. [103] present a low-cost, Bluetooth-based, 3D indoor positioning environment, where the least square method is used for estimating linear and non-linear parameters of the given regression model. By fusing the Bluetooth beacons and the Pedestrian Dead Reconning (PDR) technique to provide meter-level positioning with the help of the Extended Kalman Filter (EKF) system, Li, et al. [106] acquired an accuracy of 2 meters. Wang et al. [200] propose a Bluetooth positioning computation method based on the weighted K-nearest neighbors and an adaptive bandwidth mean shift that achieves high precision. Cheng et al. [43], use Kalman filtering, while Jianyong et al. [78] propose Gaussian RSSI filtering and optimization of positioning computations based on Taylor series expansion. Solutions considering Bluetooth and Wi-Fi data sources combination [95, 34] are proposed concerning the clear advantages of the approach that is data fusion from multiple sources. While following the trend of machine learning and consecutive filtering to different trilateration approaches, the solutions mentioned above are not space-agnostic – all reference nodes and the reference space must be known at the time of deployment. Besides, they mostly need a large contextual dataset at the beginning, i.e. a Wi-Fi or Bluetooth spectral map of the entire space.

The standard and most widely used approach in building an IPS is ML-based fingerprinting. Fingerprinting requires a major pre-deployment effort to record the spectral map of the space being observed. This is a major issue in the adoption and practical application of fingerprinting as a standard for location determination. When it comes to the initial spectral map creation of the active signal sources and fingerprinting recalibration, there are a few studies that aspired to reduce the human effort required for it. By analyzing and studying raw, crowdsourced, and unlabeled data Gu et al. [59] present a novel semi-supervised Deep Extreme Learning Machine (SDELM) algorithm, which takes the advantages of semi-supervised learning, Deep Learning (DL), and Extreme Learning Machine (ELM). With the combination of these approaches, the positioning computations can be improved both in the feature extraction process and in the classifier that estimates the position. Zhou et al. [219] and Yang et al. [213] have utilized user activities and movement to estimate the final location – an accuracy comparable to other mentioned fingerprinting approaches was achieved. However, these methods still require a large amount of labeled data to ensure location accuracy. Lastly, reducing human effort in data collection, context understanding, and space mapping is an imperative [75], and it is a topic discussed in this thesis, in abundance.

# Chapter 5

# BLEMAT: Preliminaries

BLEMAT, Bluetooth Low Energy Microlocation Asset Tracking is a semi space-agnostic, context-aware fog computing system that performs real-time indoor positioning, smoothing and filtering, fingerprinting, and floor plan layout detection accompanied with various complex data analytics and prediction and forecasting tasks [156]. The system achieves high accuracy and precision in position estimation while maintaining low resource utilization. BLEMAT is a collection of software engines that can be deployed as an integral part of any IoT device (e.g. Raspberry Pi, Arduino, etc.) with enabled Bluetooth and Wi-Fi communication modules.

The work in this chapter is structured as follows: Section 5.1 then presents a high-level functional architecture with a component diagram; Section 5.2 presents the BLEMAT workflows running at the lower-level (at the edge) on IoT controllers; Finally, Section 5.3 elaborates on active BLEMAT deployments, their scale, location and hardware/software details.

## 5.1 Functional Architecture

At the edge layer, the BLEMAT distributed edge computing system consists of multiple IoT gateways running an instance of BLEMAT connected in a full- mesh topology. This indicates that between every two IoT gateways there is an established connection ready to be used. The ECP paradigm aims to move computing processes to the edge of the network. Because of redundant network connections, systems in a mesh topology are capable of better managing high amounts of traffic and adapt to node failure issues as well as other topology changes [144]. BLEMAT, while utilizing the practices of both ECP and mesh paradigms, accumulates the advantages of both, thus creating a robust and efficient system resilient to node outage and communication overhead spikes. The communication module enables bridging

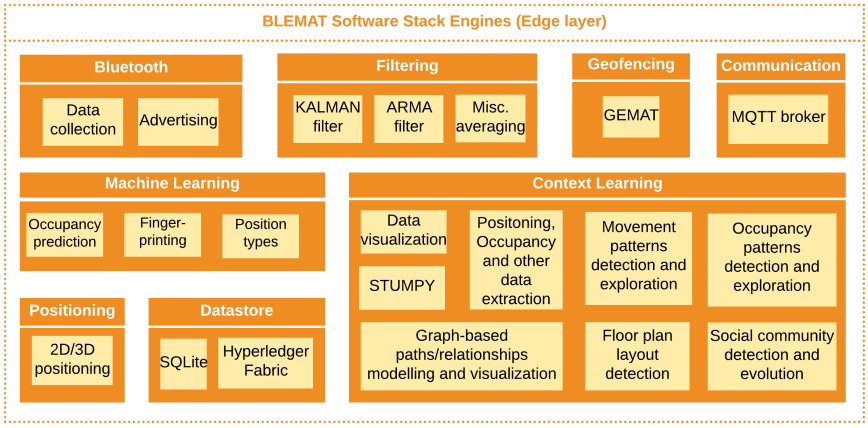



**Figure 5.1:** BLEMAT high-level functional architecture.

communication of different BLEMAT deployments when needed. The high-level functional architecture of BLEMAT is displayed in Figure 5.1

In BLEMAT (deployed on an edge device) there are 8 separate Software Stack Engines. Each will be briefly explained in the next few paragraphs.

(1) The **Bluetooth engine** consists of processes enabling the core functionalities of BLEMAT: a collection of surrounding Bluetooth signals and Bluetooth Advertising. In the current form of BLEMAT, each IoT gateway running BLEMAT must be equipped with a Bluetooth module. Bluetooth advertising enables the BLEMAT software architecture to learn about the spectral image around other IoT gateways, detect newly joined BLEMAT controllers, etc. It is important to note: BLEMAT is first and foremost a Bluetooth-based positioning system, however, it could effortlessly be setup to use Wi-Fi signals. In that case, the Bluetooth module must not be equipped on an IoT Gateway running BLEMAT.

(2) The **Positioning engine** can solve 2D and 3D localization problems using basic multilateration. Additionally, BLEMAT solves the localization problems using a *clustering combinatorial multilateration* approach – by calculating a position of a beacon for all combinations of IoT gateways that detect it, clustering the obtained positions in 2D/3D, and extracting centroids of the densest cluster as the final position. Where this approach is considered resource-consuming, regular multilateration is possible. Alternatively, BLEMAT is capable of performing positioning by resorting to fingerprinting if that is the system setup, as the machine learning engine is equipped with workflows to train such models.

(3) The **Data filtering** engine consists of pipelines for data filtering, where data can be supplied in various formats (signal data, positioning data, occupancy data, etc.). Filtering approaches include different time-series filtering techniques (auto-regressive moving average (ARMA), KALMAN filter, etc.), and dedicated

**fingerprints**

| position | | devices | |
|---|---|---|---|
| | | **bssid** | **level** |
| **floor** | 0 | BLESSID-1 | -81 |
| **X** | 12.5 | BLESSID-2 | -56 |
| **Y** | 4.5 | BLESSID-3 | -51 |
| | | BLESSID-4 | -64 |
| | | BLESSID-5 | -42 |
| | | **bssid** | **level** |
| **floor** | 0 | BLESSID-1 | -76 |
| **X** | 13 | BLESSID-2 | -49 |
| **Y** | 5 | BLESSID-3 | -57 |
| | | BLESSID-4 | -45 |
| | | BLESSID-5 | -67 |

**Figure 5.2:** Sample fingerprinting dataset.

filtering techniques for dedicated data types (e.g. occupancy data filtering based on supplied parameters).

(4) The **Communication engine** connects the entire system via Mosquitto MQTT brokers [111]. MQTT bridges exist between every gateway and are created automatically upon BLEMAT startup – a list of IP addresses or fully qualified DNS must be supplied for each device. For BLEMAT to function without problems, the MQTT configuration also needs to be updated with a set of local and remote topics, and their mapping.

(5) The **Machine Learning engine** includes workflows enabling the training of ML models capable of performing fingerprinting. A sample dataset, a subset of the fingerprints dataset from the business office deployment site (see Section 5.3.1) is displayed in Figure 5.2. Based on such inputs BLEMAT machine learning engines can train an ML model that can be based on the spectral image (RSSI level from multiple IoT gateways with BLEMAT deployed), output a position that best matches the spectral image. For fingerprinting, BLEMAT uses a probabilistic Naive Bayes classifier because it is among the simplest Bayesian network models and uses insignificant resources for training. Thus, it can be trained at the edge, even on IoT Gateways.

As an example, it takes a Raspberry Pi 4 (A-72 ARM, Quad Core, 1.5 GHz processor) approximately 6 minutes to train an ANN on a dataset of 10.000 records for an image recognition problem (recognizing numbers in images) [146] with the Python ann library [1]). Being a much simpler problem, and since the dataset is significantly smaller ( 400 records), the training of a Naive Bayes fingerprinting model for an office space of $80m^2$ on the same device takes approximately 30-45

[1] ANN library: https://pypi.org/project/artificial-neural-network/

| position | visits | time_from | time_to | duration | type |
|---|---|---|---|---|---|
| X 4.24<br>Y 4.14 | 89 | 1559727434 | 1559728034 | 4 | urban |
| X 5.17<br>Y 4.04 | 16 | 1559727434 | 1559728034 | 112 | non-urban |

**Figure 5.3:** Sample position classifier dataset.

seconds. However, since the training is not essentially required to be performed at the edge, it can be outsourced to a resource-richer machine which is more efficient.

The machine learning engine includes also the ML model for predicting the occupancy of the observed indoor space. This is an LSTM Neural Network model, that can be trained on edge, but on resource richer devices. This model was verified in a residential building deployment site (see Section 5.3.2) and is explained in-depth in Section 6.2.4. The third member of the machine learning engine is the Naive Bayes position-type classifier. This classifier accepts a dataset that represents how many certain positions were visited in the observed indoor space, for every $1m^2$. This ML model is used to decide whether a certain location in the observed space is a transitional, urban area, where Beacons are expected to express moving behavior (i.e. hallway), or a non-urban area where Beacons usually do not move a lot (i.e. a meeting room). This model impacts the filtering of the position estimations over some time, by suggesting whether the Beacon is currently in motion, or it is most likely not moving. Acting upon that result, the filtering components can further refine the Beacons trajectory in the observed space. A sample dataset used to train this model is displayed in Figure 5.3.

(6) Built on top of BLEMAT, the **geofencing Micro-location Asset Tracking (GEMAT) framework** is a semi-unsupervised geofencing system that constantly learns about the operational context of an environment where it is deployed [159]. The GEMAT allows for detection of entrance and exit event to certain indoor secured areas. The engine is in-depth described in Section 6.3.

Currently, BLEMAT's **Storage engine** supports (7) storing data in a relational SQLite database and an instance of a private and permissioned Hyperledger Fabric blockchain. Hyperledger Fabric is an enterprise-grade private (permissioned) distributed ledger platform that provides modularity and versatility for a wide range of use cases in the industry. While being first and foremost an IPS, BLEMAT also is a proof-of-concept implementation of the author's previous work on blockchain frameworks for the IoT [158, 160].

Lastly, the (8) **Context learning engine** in BLEMAT has 7 integral modules:

1. Positioning and occupancy data extraction and transformations;
2. Objects' movement patterns detection, persistence, and exploration;

3. Occupancy patterns detection and exploration (with an instance of STUMPY);
4. Set of techniques for data visualization (2D and 3D illustrations of beacons physical dispersion, movement paths, occupancy, heatmaps, etc.);
5. Graph-based paths/social relationships modeling;
6. Floor-plan layout modeling and room-level layout detection (approximation);
7. Social community detection and evolution tracking.

All of these modules expose their data and services to the local infrastructure through the MQTT broker running at the edge layer. Alongside this, there is webserver, a Python-Flask application providing the same through REST APIs, as that is the best way for an external Administration Dashboard to receive aggregated information.

## 5.2 Workflow at the Edge Layer

BLEMAT workflow (see Figure 5.4) starts with IoT gateways (in BLEMAT context the *BLE scanners*) collecting all available Beacon signals and forming a vector of RSSI readings for each Beacon (**Bluetooth Engine**). After applying a preset averaging approach on the signals over a certain period (e.g. 5 seconds) in the **Filtering Engine** the calculated average value is sent to a certain topic on the local MQTT broker (running on the IoT gateway) for each Beacon, ending up in the **Positioning Engine**. By design, averaging is performed in three steps: (1) the RSSI vector values are sorted; (2) the top and bottom 10% values are removed (deemed as outliers); (2) for the rest of the values the arithmetic mean $\bar{x}$ is calculated. Other averaging approaches can be applied here, as there is more of them available inside BLEMAT. The frequency of messages from the *BLE scanner* depends on the system parameters setup by the BLEMAT administrator (also driven by the accuracy requirements of the system), but also from the advertising interval of the observed BLE beacons (30-5000 ms).

The **Filtering Engine** is subscribed to messages that the *BLE Scanner* component, a part of the **Bluetooth Engine**, sends to the MQTT broker. By default, there is an initialized Kalman filter for each of the Beacons being tracked in the system. The filter acts upon the values it receives and smooths peaks and eventual disturbances in the received values. Before acting on the value, the Kalman filter first checks with the **Context learning Engine** component whether the parameters of the filter need to be fine-tuned (based on position type classification information). The **Context learning Engine** returns information to the Kalman filter component

whether the parameters of the Kalman filter need to be tuned, in particular, if the filtering needs to be more or less aggressive based on the position type.

When filtering has been performed, the resulting RSSI values are sent to the *Distance calculator* component, part of the **Positioning Engine**. The component calculates the distance in meters based on predefined parameters from current RSSI in dBm, using the equations written below. In the nature of RSSI there is exponential decrease in value compared to the increase of distance [41]. To account for that, the following RSSI attenuation model is used for distance estimations [116]:

$$RSSI_{dbM} = -10 \cdot n \cdot log_{10}d + A$$

where *n* is the signal propagation constant, also named propagation exponent; *d* is the distance from the sender; *A* is the received signal strength at a distance of one meter. Next, the relationship between distance *d* and RSSI can be expressed as:

$$d_{meters} = 10 \cdot \frac{RSSI_{dbM} - A}{-10 \cdot n}$$

Once the distance has been calculated the value is published to the local MQTT broker.

The *Position estimator* components is a part of the **Positioning Engine** as well, and is in charge of performing positioning computations. For a specific Beacon, it receives distance in meters from all neighboring gateways that captured the Beacon signal and computes a position (it needs a minimum of 3 gateways to capture a Beacon's position). Positions that are out of space boundaries are immediately discarded. For a Beacon, all positions captured while BLEMAT operates are also sent to the *Kalman filter* component – another instance of the Kalman filter, in charge of continuously filtering position estimations. This particular instance of the Kalman filter checks the peaks in position estimations (i.e. gross changes in position are smoothed to compensate for possible faulty measurements or calculations), compares the current position estimation with previous ones, and can either smooth or discard the position. Finally, the result in the form of coordinates ($x$, $y$, $z$) is sent to the local MQTT broker. The local MQTT broker publishes the position with other remote MQTT brokers.

This workflow is active on each of the controllers at the edge layer. To save bandwidth, only one of the gateways runs the Python-Django REST API-based service that communicates with the **Administration Dashboard**, answering to various requests for data. If the gateway goes offline, the process of running this application is smoothly handed over to another gateway. This is done in a Round-robin fashion, and the detection of failed gateways is checked with Last Will And Testament messages (core functionality of the MQTT protocol). BLEMAT IoT gateways form a full wireless-mesh topology, where the node that is communication with the Administration Dashboard is always the node that acts as a mesh

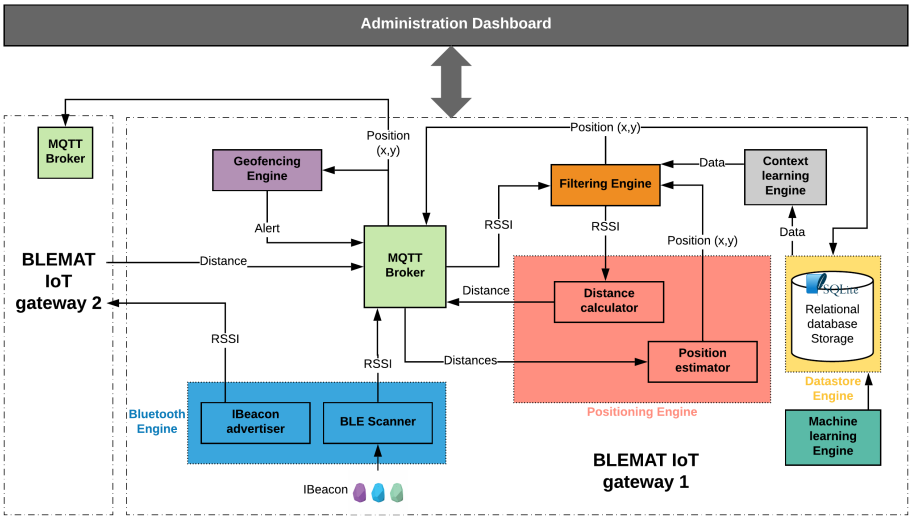



**Figure 5.4:** BLEMAT edge layer workflow.

gateway (system leader) at a given moment within the network topology. The usage of wireless-mesh topology approach increases the system's reliability, redundancy, nodes self-discovery success, and network self-healing. All the above mentioned steps run non-stop, which helps BLEMAT to extract fresh information and build knowledge of the context. Furthermore, all calculations are handled on the edge layer increasing security, data integrity and privacy, as well as trust.

## 5.3 Active Deployments

BLEMAT is actively being used in two physical locations. One of the deployment sites is a small office space occupied by VizLore Labs Foundation in Novi Sad, Serbia (described in Section 5.3.1). The other one is a residential building, part of a large residential complex in the USA (described in Section 5.3.2).

### 5.3.1 Deployment 1: Office Space

The physical layout of the BLEMAT deployed in the first deployment site is shown in Figure 5.5. Deployed IoT gateways with BLEMAT installed (i.e. BLEMAT scanners) are highlighted with red rectangles. Circles of other colors represent detected beacons over time.

IoT gateways are ARM-based processing boards (Raspberry Pi 4) capable of both scanning for BLE signals and advertising their BLE packets. Five IoT gateway devices are deployed at known, fixed points of the physical office space. Each

IoT gateway is capable of performing beacon scanning and advertising, contextual information collection, and communication with the rest of the system. The deployment site is an $80m^2$ office space with two separate rooms/offices and a kitchen, connected by a hallway and separated by a concrete wall. The flow of people in the office is rather dynamic, and so is the Wi-Fi spectral image, so there is considerable noise in signal propagation.

### 5.3.2 Deployment 2: Residential Building

This deployment site is a 5-building US-based residential complex with cloud-managed networking and IoT infrastructure providing Internet access and IoT services. The network infrastructure is comprised of 5 distribution centers (DC) with fiber-optic connectivity between each center. The infrastructure provides wireless and fixed Internet access to 149 apartments. Each distribution center includes UPSs, core routers, switches, and indoor/outdoor Wi-Fi 802.11b/g/n access points, deployed across the entire residential complex. Every wireless Access Point (AP) in addition to providing wireless Internet access, provides a multitude of IoT services. APs are based on commercial hardware and open-source software (OpenWRT). Each AP is equipped with a 1 802.11/b/g/n Wi-Fi interface and 1 Bluetooth Low Energy interface.

BLEMAT is deployed in the largest residential building in the complex (i.e. central building). The central building has 16 floors (ground plus 15) and 149 apartments (plus 12 common rooms on the ground floor). This entire building is modeled in both 3D and 2D (see Figure 5.6). In the 3D model, every apartment is modeled as an arbitrarily sized cuboid defined with its edges in programming code. However, when plotting the entire building with deployment and positioning data, the 2D model is used to achieve better visibility. The left side of Figure 5.6 shows

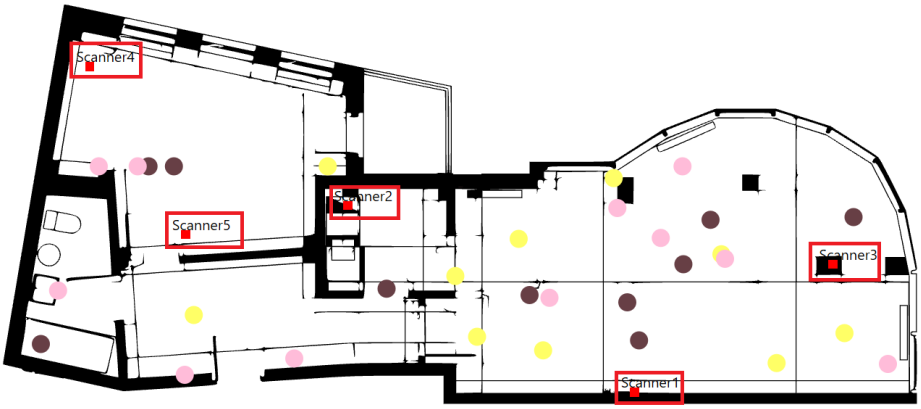



**Figure 5.5:** BLEMAT deployment site 1 – VizLore Labs Foundation office space

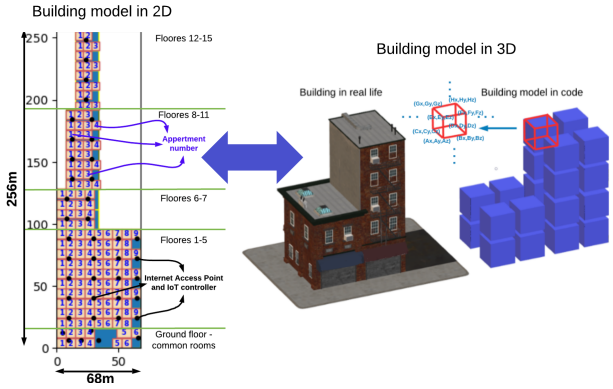



**Figure 5.6:** BLEMAT deployment site 2 – central building

all 2D representations of floor plans in the building, stacked on top of each other: the Ground floor is 68m long and 16m wide (abv. 68x16m), with 12 common rooms (kitchen, gym, computer room, offices, etc.); floors 1–5 are also 68x16, with 17 apartments per floor; floors 6 and 7 are 34x16m with 8 apartments per floor; floors 8–11 are 26.5x16 with 7 apartments per floor; lastly, floors 12–15 are 19x16m with 5 apartments per floor. Every floor is 2.5m high, making the total building height of 40m, which is not visible in Figure 5.6. There are 42 APs deployed in the central building. The distribution of APs per floor is shown in Figure 5.6 – the exact locations of the APs are depicted with a black dot.

# Chapter 6

# Software Engines Implementation Details

Inside BLEMAT there is a range of dedicated software engines (collection of software components), each in charge of procuring a certain analysis of the observed indoor space. While some are related to demystifying the physical context of the indoor space (flow of people, spectral maps, positions of obstacles, etc.), others are related to exploring behavior and social dynamics/interaction of the observed people and deducing indoor space usage patterns to aid in increasing the overall efficiency inside the observed space. This chapter will elaborate on all such software engines implemented inside BLEMAT.

The work in this chapter is structured as follows: Section 6.1 presents workflows in charge of modeling and analyzing movement patterns of the observed objects/people; Section 6.2 provides insights to workflows on the occupancy detection and prediction software engine; Section 6.3 explores the geofencing software engine and its capabilities; Finally, Section 6.4 covers processes in charge of inferring the floorplan layout by only using BLEMAT-calculated positioning data and signal parameters; Finally, Section 6.5 unravels the workflows behind social dynamics analysis, quantifying social relationships, and finding and observing social communities.

## 6.1 Modeling and Analysis of Movement Patterns

BLEMAT's **Context learning Engine** supports modeling and analysis of movement patterns for the observed Beacons and uses graphs as data structures to achieve that. Using graphs to model a beacon's behavior in BLEMAT requires the use of two distinctive types of graphs. On the one hand, to represent tenant paths, a linear/path graph called **tenant path graph** is used. A tenant path is a collection

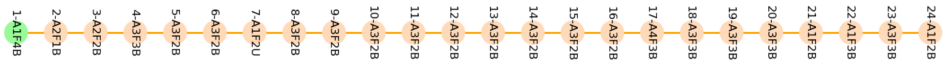



**Figure 6.1:** Example of 24-hour tenant path graph.

of rooms/apartments in the observed building where the tenant was detected at a certain time, aggregated hourly. For an observation period of 24 hours, the tenant path graph will have 24 nodes, each node representing the most common visited room/apartment since the last node. On the other hand, to represent social relationships and tenant behavior through building rooms'/apartments' occupancy data a weighted graph called **tenant behavior graph** is used. A tenant behavior graph is a collection of nodes representing rooms/apartments in the observed building the tenant has visited. Each node provides information about the number of visits, accumulated visiting time, etc. The definitions of the graph types and some representative examples follow below:

**Definition 1** *Tenant path graph $G_{(T_B,n)} = (V, E)$ for tenant T (represented by beacon B) is a connected graph with 2 nodes of vertex degree 1, and the other $n-2$ nodes of vertex degree 2. $V$ is a set of vertices and their labels and $E$ is a set of edges. A path graph is a graph that can be drawn so that all of its vertices and edges lie on a single straight line [58].*

An example of a tenant path graph for an observation period of 24 hours is displayed in Figure 6.1. Vertices are labeled with the step number and apartment label – 1-A1F4B is the first step, 24-A1F2B is the last step in the 24-hour observation period, separated by an underscore.

**Definition 2** *Tenant behavior graph $G_{(T_B,A)} = (V, E, W, V_A)$ for tenant T (represented by beacon B) from apartment $A$ is a weighted graph. The following are part of Definition 2:*

- *$V$ is a set of vertices and their labels;*
- *$E$ is a set of edges;*
- *$W$ is a relation between vertices associated with edges of specific weight;*
- *Weight $w_i$ for each edge $e$ between vertices $V_i$ and $V_j$ is defined as $n \in N$. $n$ is the number of transitions from $V_i$ to $V_j$ and vice-versa;*
- *$V_R$ is a relation of attributes corresponding to each vertex of $V$. $V_R['Stays']$ is an attribute of vertex $V$ specifying the number of hours tenant $T$ (represented by beacon $B$) spent in apartment $A$ during the observation period.*

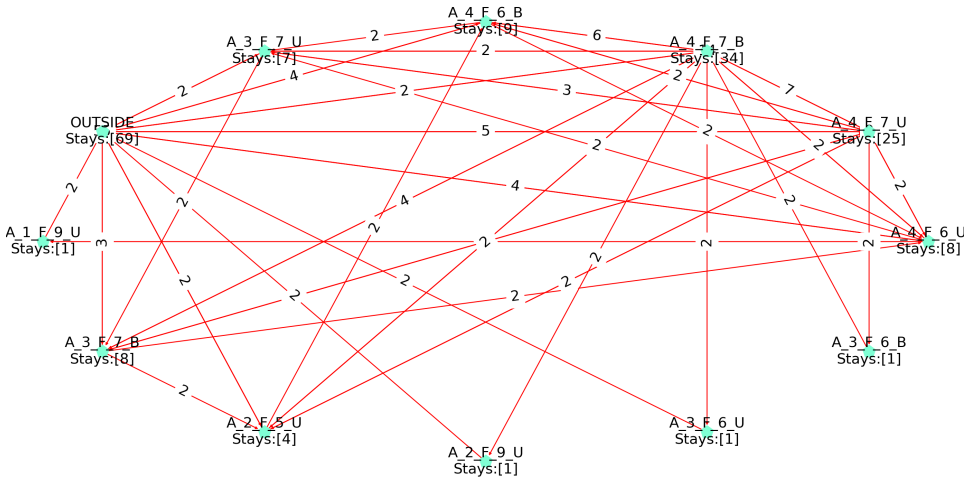



**Figure 6.2:** Example of a tenant behavior graph.

Figure 6.2 displays an example of the weighted behavior graph for the same tenant/beacon, aggregated for one week. Vertices are labeled the same as in the tenant path graph, edges are associated with weights, and there are additional vertex attributes.

### 6.1.1 SoTA Approaches

Positioning data is core data for an IPS, hence we can observe all other data as metadata, complementing positioning data to give a deeper understanding of the indoor space context that is observed. In this thesis, the modeling of occupancy, movement paths, and behavior using various graph types is considered. Graph-based space modeling, data acquisition, and information extraction models have been researched for IPS. Jensen, et al. [74] proposed a base graph and mapping model to represent the topology of indoor space at different levels. Werner et al. [208] provided a novel idea for graph-based data structure modeling for scalable and flexible geo-location querying. This thesis focuses on using graphs for both data structure and indoor environment modeling, path persistence and exploration, as well as semantic information extraction. Besides, the approaches from this thesis differentiate from the above-mentioned articles by providing modeling and complex analysis of the social behavior characteristics of indoor space users.

Using new data models to increase the amount of contextual information about the observed indoor space is crucial. Analyzing and predicting the movement patterns of observed objects/people in IPS has a high research interest [163], as it directly impacts resource optimization – crucial for a BMS. Learning indoor movement habits enables prediction and intelligent control of heating and lighting [48]. Furthermore, the movement patterns can be used to extrapolate information about

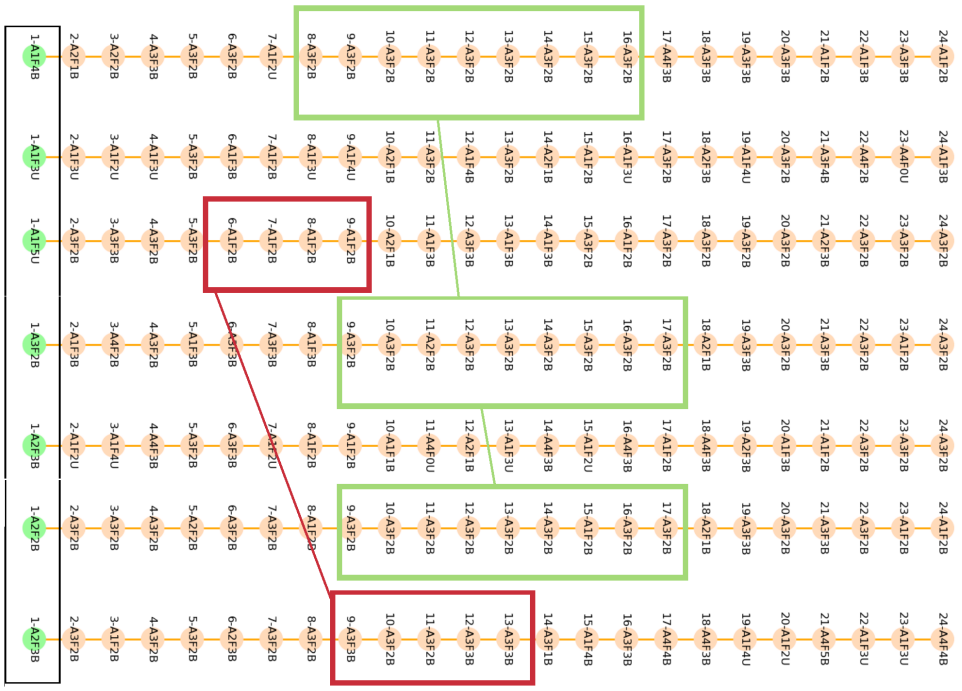


**Figure 6.3:** 7-day tenant's paths.

the type of activities in the indoor space [91]. The ability to accurately predict the movement trajectory of people holds potential benefits for many other applications as well, such as aged care [87] and retail (e.g. customer flocking [89], behavior analysis [211]).

In this thesis, the focus is on novel data models for IPS in residential buildings, their visualization and usability in BMS's decision-making processes to increase overall tenant well-being. Movement patterns prediction will not be researched, however, there are existing data acquisition workflows that can easily support that functionality within BLEMAT.

### 6.1.2 Generating Tenant Path Graphs

The generation of periodic tenant path graphs begins by building graph vertices. Given a certain time granularity (i.e. 1 vertex per hour, 10 minutes, etc.), in this process, every path graph vertex is associated with the apartment in which the tenant was detected the most times for the time granularity chosen. As an example, Figure 6.3 displays path graphs for one tenant/beacon, with a time granularity of 1 hour, for one week, starting at November $1^{st}$.

From this summarized movement behavior one can infer information about most visited apartments, and visually detect movement patterns. The most visited

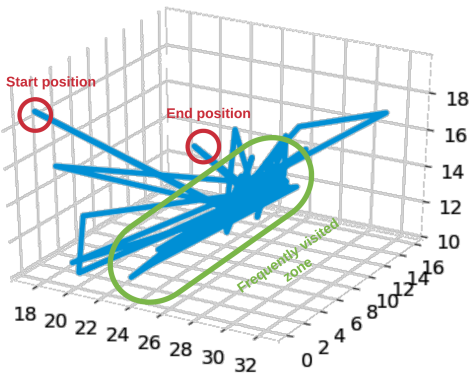



**Figure 6.4:** 1-day 3D path.

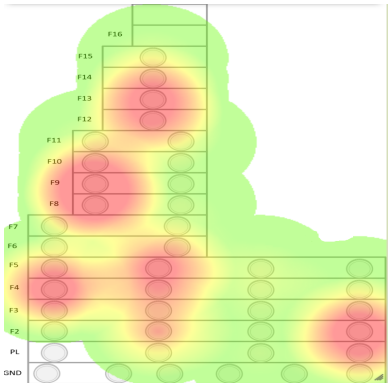



**Figure 6.5:** 1-day heatmap of tenants' movement activity.

apartment is the home apartment of the beacon/tenant – Figure 6.3 displays that the tenant usually spends his/her nights (from midnight to 8 AM) in apartment $A3F2B$, also highlighted in green. The tenant had 66 positions captured in this apartment, which is 35% of all positions. Now, the next most visited apartments can be inhered as a commonly visited friend, however, one must be aware of the floor plan layout and the fact that IPS are not 100% accurate most of the time due to signal perturbation. Due to that fact, the next most visited apartment, if it is next to the tenant's home apartment, can also be interpreted as a positioning inaccuracy. In the observed residential building, in 82% of beacons detected the home apartment is self-evident.

Conclusively, this tenant's weekly behavior does not display any interaction outside the building – in Figure 6.3 there are no *OUT* vertices. This type of tenant behavior can infer an emergency, especially if the movement is minimally physically dispersed in the observed area.

BLEMAT enables the creation of 3D tenant movement path graphs (visualized in Figure 6.4) encompassing the entire building's 3D area. Although this kind of visualization can become crowded with data points, it can provide certain insights: for one tenant one can extract frequently visited, i.e. *hot* zones; if one overlaps multiple paths information about frequent intersection zones and points can be inferred; based on this representation one can create indoor space heat maps based on the frequency of visits and similar [180].

Tenant path graphs can be applied to the 2D building model representation to display a heat map at the building level or the floor level. All tenants' movement activity is captured as a heat map in Figure 6.5, and additionally a subset of the movement activity per floor is displayed in Figure 6.6.

### 6.1.3 Graph-based Detection of Movement Patterns

Occupancy patterns differ significantly from movement patterns – the former describe how the indoor space is occupied, and the latter how users of the indoor space interact with it. These movement patterns represent interaction patterns both between users and users and the indoor space. Detecting the existence of movement patterns is important for efficient resource utilization of both the indoor space and its users. A machine's/person's movement pattern can be used to detect deviations in real-time and signal an emergency/unusual situation or to proactively utilize resources (i.e. turn on lights, call an elevator, etc.). In an emergency scenario, the information about a certain person's movement pattern can be used to build individual, guided escape routes that will be more efficient since they will guide through well-known indoor space areas (when applicable). Collective observation of movement patterns can be used to plan emergency escape routes more efficiently. In this part of the thesis, experiments on how tenants' movement patterns can be extracted will be showcased.

The similarity of graphs per time interval (days, weeks) is calculated as an average similarity between every 2 consecutive graphs. In general, the expression can be formulated as:

$$S = \frac{\sum_{i=1}^{k} \textit{similarity}(G_i, G_{i+1})}{k} \tag{6.1}$$

where if $G_i$ is a weekday and $G_{i+1}$ is a weekend day (and vice versa) their similarity calculation is omitted. That is done empirically, based on previous research on tenant's behavior in such residential buildings. Since tenants' activity during the weekends is more stochastic than during the working week [157], comparing workday to weekend behavior yields unstable results.

In Eq. 6.1 *similarity* represents an interchangeable similarity function. BLEMAT is generating tenant path graphs constantly. While graphs present a visually satisfactory approach to modeling path-like data (both in 2D and 3D), comparing

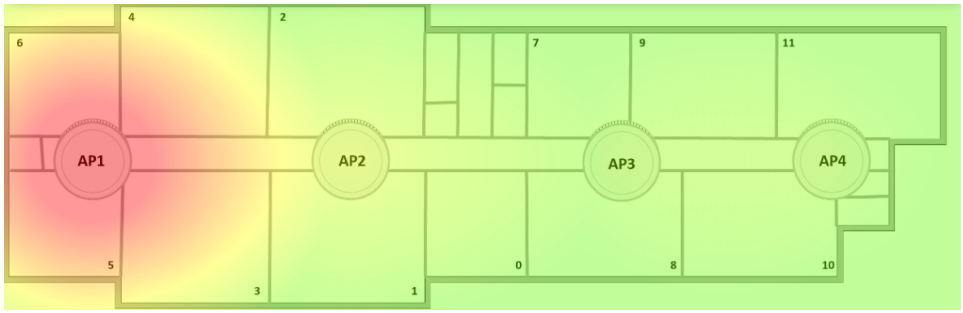



**Figure 6.6:** 1-day heatmap of tenants' movement activity per floor.

graphs is not a trivial problem. Graph-similarity algorithms, such as Graph-edit distance (GED), are computationally and time-consuming, and cannot be applied in a workflow requiring real-time or near-real-time information retrieval. Using the fact that BLEMAT-defined path graphs are straightforward linear graphs, we are transforming graphs into sentences, mapping each vertex to one word in the sentence. So, graphs are used for visualization, but word vectors are used for similarity calculations, therefore transforming a graph-similarity problem to a string-similarity problem, saving minutes in processing but obtaining satisfactory results. On newly formed sentences different string similarity measures are applied to obtain average similarity of paths for a certain tenant: Percentage similarity, Cosine, and Levenshtein similarity.

Percentage similarity calculates the overlapping percentage of paths, minding word order. Cosine similarity measures the orientation of two n-dimensional sample vectors irrespective to their magnitude. It is calculated by the dot product of two numeric vectors, and it is normalized by the product of the vector lengths so that output values close to 1 indicate high similarity. For two paths $P1$ and $P2$ cosine similarity are defined as:

$$\cos(\mathbf{P1}, \mathbf{P2}) = \frac{\mathbf{P1P2}}{\|\mathbf{P1}\|\|\mathbf{P2}\|} = \frac{\sum_{i=1}^{n} \mathbf{P1}_i \mathbf{P2}_i}{\sqrt{\sum_{i=1}^{n} (\mathbf{P1}_i)^2} \sqrt{\sum_{i=1}^{n} (\mathbf{P2}_i)^2}} \tag{6.2}$$

Lastly, Levenshtein similarity measures distance between two sentences as the minimum number of single-character edits required to change one word into the other.

For the tenant/beacon showcased in both Figures, 6.3 and 6.30 average one-week paths percentage similarity is 50%, cosine similarity is 0.62 and Levenshtein similarity is 17 – although there is an indication of some pattern-like movement behavior the numbers are not high enough to indicate certain pattern existence. However, other tenants' beacons displayed in Figure 6.8, which shows a small subset of all beacons, show higher similarity scores where patterns exist with a higher probability. When running for all observed tenants/beacons (58 in June, 110 in November and 118 in December), the existence probability of tenant weekly movement patterns is displayed in Figure 6.7. The number of observed tenants changes over time due to fluctuations in the occupancy of the apartments in the building. Since we are dealing with a building mostly occupied with university students, in June the occupancy is lower due to less university activities. In November and December the occupancy is more consistent. We can infer patterns existence where average percentage similarity is $> 70\%$, cosine similarity $> 0.7$, and Levenshtein similarity $< 20$. The three similarity metrics yield consistent results: if patterns are inferred by one of the metric other would behave accordingly. The numbers are set based on the authors' empirical knowledge of the observed physical environment and extensive and repetitive running of the experiments described in this research.

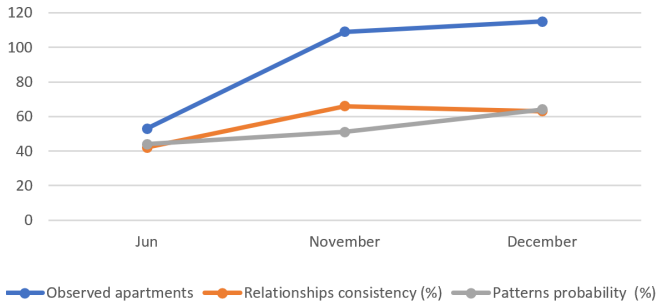



**Figure 6.7:** Periodic tenants behavioral characteristics.

The figure also shows the relationships consistency, however, this will be explored and elaborated on in Section 6.5.1.

The simple graph representations of tenant path graphs make them suitable for increasing computational performance by using string instead of graph-similarity functions. Converting graphs to sentences and running string-similarity functions saves 2–6 seconds per one pair of compared graphs.

### 6.1.4 Discussion

With performed experiments it was showcased how different IPS metadata can be extracted and used to model the behavior of tenants in a residential building. In this section, the applicability of achieved experimental results, with an accent on impact, is further discussed.

By modeling tenant movement paths a visual overview of the tenant's movement in 2D and 3D is provided. Alongside, by overlapping multiple tenant path graphs, intersection points and/or areas can be pinpointed, leading to the discovery of potential frequently used indoor areas. Finally, by transforming path graphs into sentences similarity algorithms on two consecutive sentences can be run – which is faster than for two consecutive graphs. The three resulting similarity metrics (overlapping percentage, Levenshtein, and cosine similarity) combined give a probabilistic approximation of the possibility of pattern existence, thus inferring the existence of behavioral patterns for the entire building. This information, combined with the occupancy patterns detection and forecasting [157], can be used to

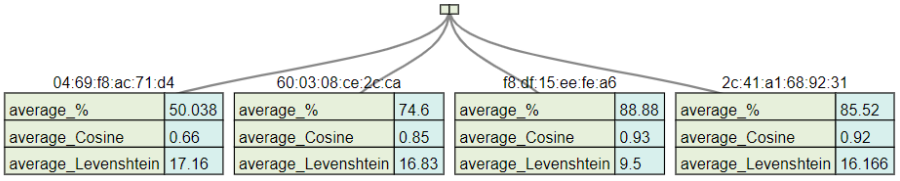



**Figure 6.8:** 1-week observed paths similarity for a subset of tenants' beacons.

drive a series of SBM decisions. Correlating movement with occupancy patterns gives a better outlook on the tenants' interaction with the observed indoor space which can be used for decision-making related to efficient resource utilization (elevators, electricity, HVAC). Movement patterns can be used to infer activity or inactivity of tenants, indicating a potential human-emergency situation. Furthermore, by knowing a tenant's movement pattern, custom emergency escape routes can be built, that would overlap the areas of space that are best known to the tenant with the shortest escape route, taking into account also the real-time congestion and flocking of people.

## 6.2 Occupancy Detection, Prediction and Data Analytics

Operating costs of buildings are a significant expense for all businesses, and it is vital to find a way to run these facilities as efficiently as possible. IoT-enabled Building Management Systems (BMSs) provide means for process and resource use automation leading to overall efficiency improvements. Inferring spatial and temporal occupation in all its forms (binary, numerical, or continuous) is one of the key contextual inputs required for smart BMSs. The design, implementation, and experimental validation of a smart building occupancy detection and forecasting solution are shown in this part of the thesis. The presented solution comprises three main building blocks: (1) An edge computing indoor positioning system (BLEMAT) which, combined with wireless access network monitoring processes, produces indoor location information in a semi-unsupervised manner; (2) Data analysis and pattern searching pipelines responsible for fusing data coming from different smart building and networking systems and deriving information on temporal and spatial occupancy patterns; (3) Long short-term memory (LSTM) neural networks trained to predict occupancy patterns in different areas of a smart building. Experimental validation confirms that the proposed solution can provide the information needed for smart building management systems to detect and predict occupancy.

### 6.2.1 SoTA Approaches

Despite the abundance of application scenarios, modeling the occupancy of buildings remains a tough, costly, and error-prone task [143]. The approaches for solving the occupancy forecasting problem differ primarily in six aspects: (1) choice of ML models, (2) choice of data sources fo extract occupancy from, (3) determination of the prediction's temporal characteristics, and (4) definition of occupancy monitoring (detection, counting, tracking or recognition of behaviors); (5) the accessibility of complete and original datasets for experiment reproducibility and (6) the sophistication and extend of data analytics and pattern extraction techniques

applied to occupancy data. SoTA work will be analyzed from the standpoint of these 6 aspects.

Existing SoTA research can be categorized into two approaches based on the model of occupancy forecasting. On one hand, there are probabilistic models, where occupancy is stochastically modeled, and on the other hand, one can resort to using standard ML techniques.

A multitude of research work addresses the problem of occupancy monitoring from the probabilistic perspective [212, 124, 108]. Dong et al. approach the problem of occupancy prediction (binary and concrete) to optimize HVAC systems by using expectation maximization, a probabilistic simplified finite-state automata with binary states, a general system problem solver and uncertain basis functions [38]. It is not mentioned what the occupancy prediction model 's maximum temporal forecasting capabilities are, and no occupancy data analytics/pattern search functionalities are showcased. Dataset is built from data (not public) coming from occupancy sensors. Li et al. use the Markov models (MM) to predict short-term occupancy at 15, 30, 60 minutes, and 24 hours at room and house level [107]. Passive Infrared ( PIR) sensors are used to extract occupancy data (dataset not public). Analytics on the occupancy data are presented, however, without occupancy pattern recognition. Binary and concrete occupancy is forecasted. Likewise, Chitu et al. use MM to predict binary occupancy based on PIR data (dataset is not public) [24]. Information about the maximum temporal characteristic of the forecasting model is not disclosed. To control the indoor household temperature in a simulated environment, Lu et al. use Hidden MM to predict occupancy probabilities through the behavioral characteristics of tenants (departures, arrivals) [117]. Occupancy data comes from motion sensors and door sensors. Two public datasets are used in conjunction with the newly collected data described in the research. Occupancy types and patterns have been analyzed to conclude that the periodicity of patterns has an impact on framework capabilities. Ryu et al. worked on predicting the number of occupants with hidden MM, but instead, they have used data about occupancy profiles, $CO_2$ concentration, and energy use of smart lighting systems and appliances (dataset is not public) [172]. Occupancy patterns have been analyzed to create different room occupancy profiles. Another approach to tackling the problem of occupancy monitoring is the use of standard ML techniques. Previous studies related to occupancy monitoring with ML models have successfully predicted occupants' behavior in different formats: binary classification [174, 148, 38](occupied or unoccupied), concrete values [38, 21, 162, 201](e.g. number of occupants), as well as time-series [212, 124, 108]). Both unsupervised and supervised ML techniques have been used in this area – the latter to learn and forecast occupancy based on historical data and the former to find occupancy patterns [110]. Qolomany et al. propose to detect the number of people at a given time and location at 15, 30, and 60 minute

time intervals by using only the Wi-Fi dataset [162]. Predictions are made by the Autoregressive Integrated Moving Average and LSTM ML models. However, no datasets or data analytics techniques applied in regards to patterns of occupancy are presented. Mamidi et al. and Ekwevugbe et al. both use ANNs to infer the number of occupants and binary occupancy [126, 40]. Mamidi et al. use motion, CO2, sound, light, and door sensors to predict occupancy up to an hour in advance, while Ekwevugbe et al. use temperature, motion, and sound data (datasets are not public). Occupancy patterns have not been analyzed. Chen et al. predicted the number of occupants (dataset is not public) on data collected from a university campus building. ANN and SVM showed good performance in predicting the number of occupants during the forecast timeframe up to 2h, without a deeper analysis of the occupancy patterns [21].

Conclusively, the core problem of occupancy modeling, which is learning patterns of occupancy and forecasting occupancy schedules, has rarely been addressed due to both the highly stochastic behavior of occupants and the lack of high-quality, statistically relevant datasets. In their work Liang et al. present a two-fold approach to this problem: firstly, they perform clustering of occupancy patterns with unsupervised techniques, and secondly they perform classification of inferred daily profiles of the number of occupants in the observed building [110]. Similarly, Capozzoli et al. used K-means and decision trees to model occupancy profiles to reduce the energy consumption of HVAC systems [11]. Chang et al. infer occupancy patterns from lighting switch data identifying 5 typical occupancy patterns in the office building, based on the average daily 24-hour employee presence profiles in their [16] cubicles.

Most of the SoTA research does not make datasets publicly available, although they are of high quality ( i.e. University of Florida campus, De Montfort University campus, etc.). Such datasets would make a significant contribution to the field of indoor positioning as well as occupancy modeling. The capacity of the SoTA models for occupancy forecasting is mostly short-term and concrete (up to 24 hours, forecasting the number of people), and no SoTA research focuses on long-term binary occupancy forecasting (i.e. 1–2 weeks ahead). The choice of prediction models varies between probabilistic, ML, and deep learning models. Deeper insights into occupational analytics and pattern search and analysis are rarely mentioned in combination with occupancy forecasting and additional, more sophisticated occupancy data analytics techniques are omitted. Existing SoTA research focuses on either occupancy prediction or occupancy pattern mining, not directly linking the two in any way.

To fill the gaps mentioned above, the scientific contributions of this thesis regarding both occupancy detection and prediction, as well as challenges in occupancy pattern analysis research are:

1. Design and implementation of a **specialized subsystem of BLEMAT with straightforward, lightweight and novel approaches to occupancy forecasting**, data analytics, visualization, and pattern-search pipelines, requiring minimal effort for deployment on existing smart building infrastructures;

2. A **forecasting model and results for long-term binary occupancy forecasts (1 week ahead)** based solely on Bluetooth and Wi-Fi usage data from a smart building, requiring no additional occupancy (or other) sensors;

3. **Supplying pipelines for both autonomous and user-driven occupancy pattern matching and extraction** per the desired area of granularity (building, floor or room);

4. **Linking occupancy forecasting results (2) directly with occupancy data analysis (1,3)** to acquire real-time and strategic (future) SBM insights;

5. **Provide high-quality real-world positioning and occupancy datasets** for experiments (public datasets that will be updated regularly via Zenodo) [151].

### 6.2.2 Occupancy Data Acquisition

For the entire residential building (see section 5.3.2), BLEMAT provides means for Bluetooth-based positioning, real-time and historic location data analysis, advanced geofencing capabilities, and Bluetooth scanning and advertising. LBSs built on top of BLEMAT provide insights into how spaces are used in the residential complex, in particular common rooms such as laundry rooms, gyms, cafeteria, etc.

For the set of experiments described in the rest of this section positioning and Wi-Fi access data were collected from the central building alone (see Sec-

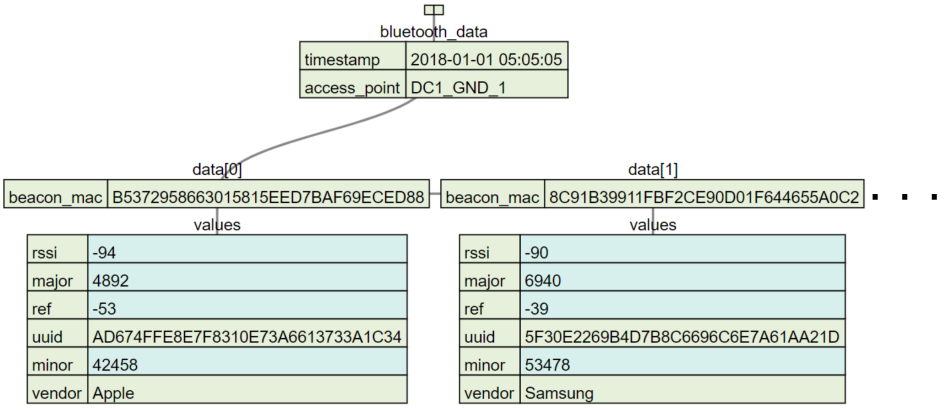



**Figure 6.9:** Bluetooth device output signal strength snapshot.

tion 5.3.2). The precise 3D location has been captured for every Bluetooth beacon detected inside the building, in a longer period (during BLEMAT operation). For some devices, there is information about the device type, seller, etc. while this is omitted (not advertised by the particular Bluetooth advertising packet of the device) for other devices, usually by design. Since it is the largest in the residential complex, the data experiments in this building will have statistical relevance and will give a general idea of how data collected in the entire complex can be used to provide better SBM services and to increase the efficiency of the building resources. Currently, two relevant time-based tasks are scheduled to collect data for experiments in the central building: Bluetooth devices signal strength snapshot and Wi-Fi hotspots usage and access data.

Bluetooth devices signal strength snapshot is the result of time-based tasks performed every 10 minutes inside BLEMAT. Overall, this building-wide snapshot provides information from every AP that includes a list of Bluetooth Device Information objects (BDI) acquired through BLEMAT scanning from the AP. Specifically, one instance BDI model includes the MAC address of the Bluetooth device, the Received Signal Strength (RSS) in dBm from the scanning AP, the 1-meter reference RSS, the major version, the minor version, the vendor, and the Universally Unique Identifier (UUID). The exact representation of the data collected is shown in Figure 6.9. The snapshot was captured at a certain timestamp, from a particular BLEMAT-enabled AP. The result of the snapshot is an arbitrary length data array. Note that the MAC addresses of the beacons and the UUID are scrambled concerning the privacy of the users, and timestamps are shifted.

Wi-Fi hotspots usage and access snapshot is the result of time-based tasks executed every 20 minutes on the cloud platform running on top of the SBM to manage the network infrastructure (not operated nor controlled by BLEMAT). The task uses proprietary APIs to collect network usage data for each AP and core routers and store results in the predefined cloud data store. Overall, this snapshot gathers information on how the Wi-Fi network is used throughout the building at a specific time. The following information is collected for each connected device: the location of the device in terms of the AP and Wireless Service Set Identifier (SSID) to which the device is connected, the apartment number of the user that is most likely the owner of the connected device, its MAC address, the device uptime, the device idle time and the amount of data downloaded and uploaded during the device uptime. Device idle time refers to the period since the last activity of the device has been detected on the network. Besides, the snapshot includes information about the device category (desktop, laptop, tablet, smartphone, etc.), the operating system (Android, IoS, etc.), and device model information (e.g. iPhone 7, iPad Air 2, etc.). The exact representation of the data collected with the Wi-Fi hotspots usage and access snapshot is shown in Figure 6.10. The snapshot was captured at the given timestamp. The result of the snapshot is again, an arbitrary length data array.

Note that the devices' MAC addresses are scrambled concerning user privacy, and timestamps are shifted. In the rest of the section, experiments will be performed on the fused data from these two tasks. The data from Bluetooth and Wi-Fi snapshots observed in this study contains data for approximately 6 weeks during May-June 2019.

### 6.2.3 Data Preprocessing

The final result of the preprocessing of data acquired from snapshots described in the previous Section will be information on apartment occupancy, per apartment, for a given range of dates/times and for a given time-frequency (i.e. show occupancy for apartment X, between 6/4/2019 and 6/11/2019, per hour). To achieve this level of information granularity, several steps need to be taken:

1. Extraction of unique beacons detected over some time;
2. Calculation of distances from AP to beacons;
3. Determining the 3D position of each beacon;
4. Filtering of determined positions;
5. Determining the apartment label (apartment-level position) for each beacon;
6. Apartment-level occupancy dataset construction;
7. Wi-Fi usage data cross-matching and merging with (6).

In Step 1, all AP snapshots are aggregated for a short period (10–15 minutes) and non-duplicated information about Bluetooth beacons inside the building is extracted. The following information is extracted for each detected beacon: mac

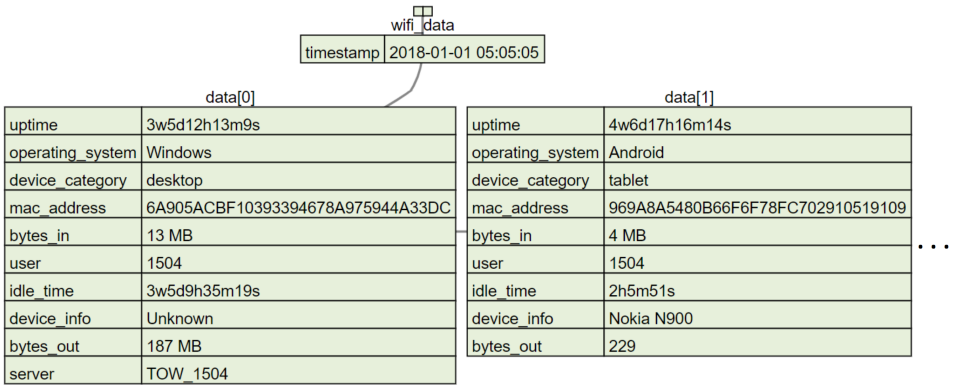



**Figure 6.10:** Access and use the snapshot output of Wi-Fi hotspots.

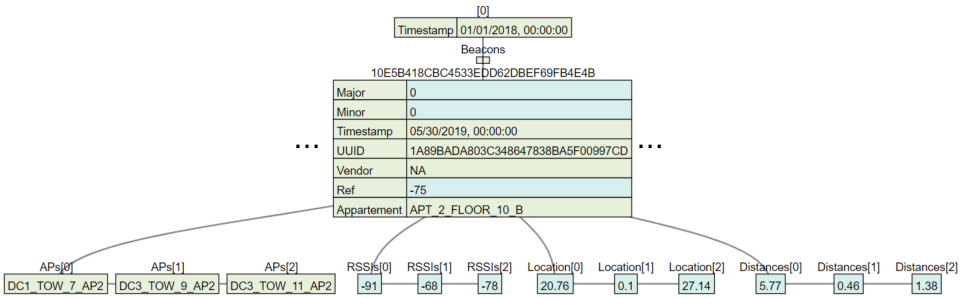



**Figure 6.11:** Development of occupancy dataset after step 5.

address, list of APs that can detect the beacon, list of AP-to-beacon RSSI in dBm, beacon minor and major versions, UUID, and reference RSSI in dBm at 1 meter (see Figure 6.9).

For every beacon detected in Step 1, Step 2 consists of calculating the distance between the beacon and the AP that has detected the beacon. The data model is extended with a list of calculated distances, corresponding to the indexes of the list of APs and RSSIs. The distance is calculated as described in the positioning engine of BLEMAT (see section 5.1).

In Step 3, the exact 3D position for each beacon is extracted from the data acquired from two previous steps. The position is calculated using the 3D multi-lateration model described in Section 5.1. inside BLEMAT's Positioning Engine. Step 4 includes the filtering of the positioning results that are misplaced in the 3D model of the building (outliers). Step 5 provides an apartment-level location for each beacon within the building (see Figure 6.11). This is made possible by having a predefined 3D mapping function for each apartment within the building consisting of its 8 edges (see Figure 5.6). In Step 6, everything is ready to extract the dataset containing apartment-level occupancy. The occupancy is extracted for each apartment, per given date and time as well as time-frequency. Step 7, also the final step in data pre-processing workflow is to cross-match the apartment-occupancy data obtained from Wi-Fi usage data and to merge the occupancy (based on Wi-Fi data) extracted per apartment with the Bluetooth-based occupancy dataset obtained from Step 6.s For a given time range, if it is detected that the user device had actively been using the AP (idle time less than 20 seconds, see Figure 6.10, attribute *idle*) that is inside the tenant 's apartment (the tenants can have their own APs), this apartment is labeled as occupied based on Wi-Fi access data (if not already labeled as occupied from Bluetooth data). The entire collection of different datasets is publicly available through Zenodo. Datasets starting with the keyword *raw* represent data collected and formatted before step 6. These datasets contain detected beacons and their 3D positions. The datasets starting with the keyword *occupancy* relate to the final datasets obtained after all 7 steps have been executed. Concerning the privacy of tenants, only Bluetooth data is made available, while Wi-Fi data

is not disclosed. Besides, all timestamps and mac addresses of beacons have been scrambled.

### 6.2.4 Multi-step Occupancy Forecasting

Long Short Term Memory (LSTM) Recurrent Neural Network (RNN) is used for multi-step occupancy forecasting. RNNs facilitate the creation of time-dependent and sequential data functions, such as stock market forecasting, text generation, etc. Nevertheless, RNNs suffer from the problem of vanishing gradients, which limits the learning of long data sequences [65]. Unlike other ML algorithms, LSTM RNNs are capable of automatically learning features from sequential data, support multivariate data out of the box, and can output variable-length sequences that can be used for multi-step forecasting. LSTMs overcome the problem of vanishing gradients by creating a relationship between the forget gate activation and the gradient calculations, thus providing a path for information flow through the forget gate for important information [80].

#### LSTM NN Architecture

The main components of the LSTM RNN are the memory cells and its gates (forget, input, and output). The LSTM Gating Structure allows information to be saved over several time-steps and, by extension, allows gradients to pass through multiple time-steps. The LSTM gates protect the LSTM unit from misleading signals: the input gate protects against insignificant input events, the forget gate helps to dismiss the previous memory content and the output gate decides whether to expose the output of the memory cell. The three LSTM gates have sigmoid activation functions for the $[0, 1]$ restriction [80].

Several parameters are used to build a proper LSTM RNN architecture. The first step is to determine the number of recurrent LSTM layers and the number of neurons per layer. The shape of the input is specified as a 1D occupancy array (0,1). Dropout for LSTM layers is defined to prevent overfitting of the model [22]. As we deal with the problem of binary classification, the logistic activation function,

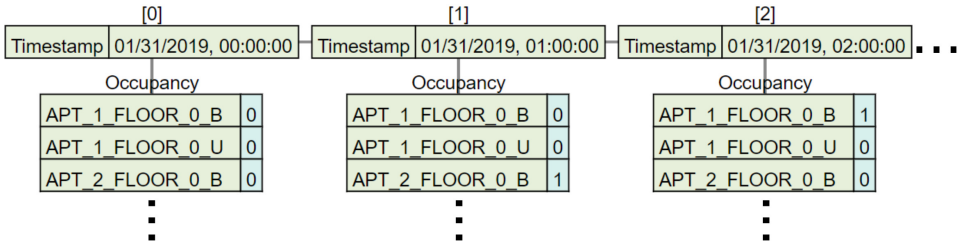



**Figure 6.12:** Final dataset for building occupancy.

or *sigmoid*, is a viable candidate to shape the output between 0 and 1. Finally, the Dense layer is stacked at the end to get the resulting classes (0,1) comparable to the input occupancy labels. The compilation of the constructed network transforms the basic sequence of layers into a set of matrix transformations [62]. To minimize the loss function (Mean Squared Error), Adam Optimizer is used, which is efficient for LSTM RNNs and requires little memory [88].

Network training requires the setting of 2 additional parameters: the number of epochs and the batch size. The number of epochs is the number of times the learning algorithm runs through the entire training dataset. The batch size is the number of samples processed before the network model is updated. Both parameters have been empirically adjusted by analyzing how they affect forecasting performance.

LSTM RNNs were built and trained with the Keras library [62] on a machine with the Intel Quad-Core i7 processor and 16 Gb of RAM, which is part of the edge network. In the BLEMAT design, once trained, the model weights file is transferred to the edge gateway(s) from where it can be used for forecasting (using TensorFlow Lite [1]). This way of using the model is verified in the BLEMAT deployment in the office space (see Section 5.3.1) as BLEMAT is deployed on a resource-richer SBCs (Raspberry Pi 3). However, this was not possible on the testbed in the observed residential building – SBCs used to run BLEMAT are highly limited in terms of both storage, memory, and processing capacity.

#### Forecasting Problem and Goal

The forecasting problem is framed as the forecast of individual apartment's occupancy given the historical data for that apartment (requires building and training 1 LSTM RNN per apartment) 1 week ahead. Moving forecast window sizes are used – based on $1 \ldots k$ weeks, occupancy for the week $k+1$ is forecasted to find the optimal input-to-forecast window. For occupancy forecast the largest dataset (6 weeks) is used, with two granularities: occupancy recorded every 10 minutes and 1 hour. Besides, three distinctive datasets (by granularity level) were formed: the first dataset contains all days, the second dataset contains only working days, while the third dataset contains only weekend days.

Through experiments with LSTM RNN forecasting in this research, it has been verified that LSTM RNNs can be used to predict longer occupancy and unoccupancy patterns with satisfactory results and that the forecasting of 1 week ahead based on $k$ previous weeks yields satisfactory forecasting results. This research argues that, when datasets are split into workdays and weekends, the forecasting model improves in both cases and that forecasting improves for datasets with lower occupancy granularity (10 minutes). The work in this thesis also shows that the decision on the input-to-forecast window size severely impacts the outcome of the

[1] TensorFlow Lite: https:/www.tensorflow.org/lite

| Exp. | Dataset type | Occupancy granularity | ITF week ratio | LSTM layers | Neurons | Dropout | Epochs | Training time |
|---|---|---|---|---|---|---|---|---|
| 1 | all days | 1 hour | 3-1 | 2 | 100 | 0.2 | 50 | 200s |
| 2 | all days | 10 min | 2-1 | 2 | 100 | 0.2 | 50 | 50s |
| 3 | workdays | 10 min | 2-1 | 3 | 50 | 0.2 | 100 | 80s |
| 4 | workdays | 10 min | 3-1 | 2 | 100 | 0.2 | 50 | 200s |
| 5 | workdays | 10 min | 2-1 | 5 | 200 | 0.2 | 500 | 500s |
| 6 | weekends | 10 min | 2-1 | 1 | 100 | 0.2 | 50 | 50s |

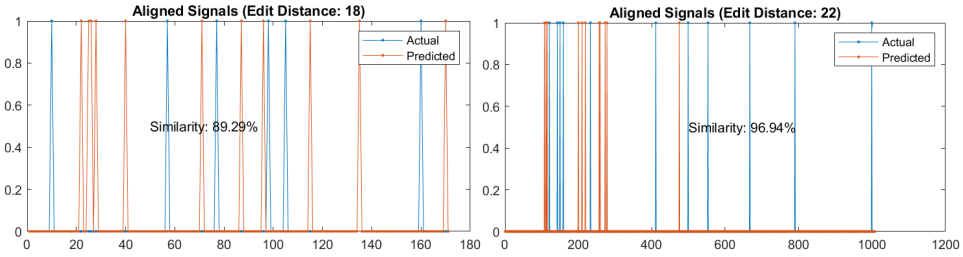
**Table 6.1:** Experiments summary.

forecast. Finally, the trained models are tested using metrics other than RMSE, which are more fitting for the comparison of time-series.

### 6.2.5 LSTM RNN Training and Evaluation

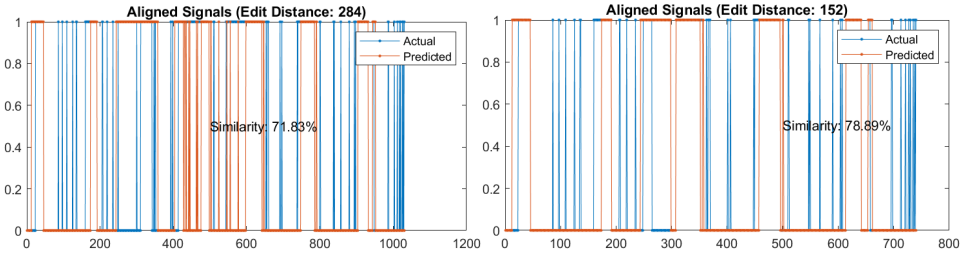
Training of LSTM RNN for forecasting occupancy data 1 week ahead has been trained and evaluated on the datasets mentioned above. Specifically, the experiments were run with different parameters: type of dataset (all days, weekend days, working days), occupancy granularity (1h, 10min), NN training parameters (number of LSTM layers, number of neurons per layer, dropout, batch size, number of epochs) and input training data in weeks (2 and 3). All experimental networks have been trained on data of all apartments. In this section, a subset of the results of the training (for some apartments) is shown to highlight the conclusions of the forecasting objectives. Table 6.1 shows all LSTM networks used in training and evaluation through occupancy forecasting experiments (training time is shown per apartment).

Prediction evaluation was performed on weeks that the model did not receive as inputs for learning (see ITF week ration in Table 6.1. The forecasting success measurement metric used by default for recurrent NNs is RMSE, the square root of the average squared difference between the forecasted and the actual values. For binary sequences (time-series), other measurement criteria for the identification of similarities between real and forecasted time-series are considered in this study. Since behavior trends are supposed to be identified in occupancy data, metrics that measure similarity based on subsequence occurrences are more suitable. Edit Distance on Real Sequence (EDR) is such a metric [42]. The EDR between two numerical time-series is measured as the number of operations (replace, insert, delete) required to convert one time-series to the other. If the Euclidean distance between two points $x_i$ and $y_i$ is less than epsilon (0.1 in this case), they are considered identical ($d = 0$), otherwise they will be considered different ($d = 1$). The similarity percentage of two input time-series can be determined based on EDR outputs. EDR is chosen as a more suitable metric as RMSE is prone to outliers, specifically spikes after longer occupancy/unoccupancy. Therefore, in the case of occupancy forecasting, it is more important to determine whether the longer subsequences are correctly forecast than to observe the error for every single data point.




**(a)** 1h occupancy granularity **(b)** 10min occupancy granularity

**Figure 6.13:** Forecasted occupancy for *APT_4_FLOOR_4_B*.




**(a)** All days **(b)** Only workdays

**Figure 6.14:** Forecasted occupancy for *APT_8_FLOOR_1_B*.

Other relevant approaches for characterizing the similarity between time-series, such as the Longest Common Subsequence and Edit distance with Real Penalty, are also discussed by Kurbalija et al. [97] and Geler et al. [52].

The main differentiators for forecasting results are the ITF week and occupancy granularity ratios. Firstly NNs for Exp. 1–2 were trained to test this hypothesis. As an example, the results of Exp. 1–2 are shown in Figures 6.13a and 6.13b, for *APT_4_FLOOR_4_B*. The X-axis is the time in hours/minutes, the Y-axis is the binary occupancy. As can be seen, the smaller occupancy granularity indicates a marginal increase in the forecasting performance, both in terms of RMSE (0.14 over 0.33) and EDR similarity (96% over 89%). Exp. 3–5 RNNs are trained on workdays dataset for 10min occupancy granularity. For Exp. 3–5 all LSTM parameters (except Dropout) were adjusted, to show there is no significant effect on the forecast. Furthermore, tests were run to see whether the forecasting model improves when not all days are used to forecast the next week. As an example, see Figures 2 and 3 for *APT_8_FLOOR_1_B* – Figure 6.14a represents similarity when all 7 days are used to train the model and yield forecasts, and Figure 6.14b when only working days are used – the EDR similarity increased. Exp. 6 was run with weekends dataset only, with default LSTM RNN parameters described above. However, in most of the apartments for the weekends' dataset, the RMSE is larger

(>0.5) and the EDR similarity is smaller (<65%), such as in *APT_8_FLOOR_1_B* – EDR similarity is 64.17% and RMSE is 0.69 (see Figures 6.14a and 6.14b). Occupancy activity during the weekends is more stochastic than during the working week, and the NN has little data to learn from (most of it with no regularities).

**Forecasting Conclusive Remarks**

All experiments have confirmed that adjusting the number of LSTM layers, neurons per LSTM layer, and epochs, as well as the dropout, does not significantly affect the forecast. The number of LSTM layers was varied between 1 and 5, with 2 layers showing the best results. The number of neurons was modified between 50 and 200, resulting in an optimum amount of 100 neurons per layer. The number of epochs was varied between 10 and 500, while the 50 epoch training period proved to be optimal. For the number of epochs below 50, the forecast time-series values do not converge close enough to 0 and 1, while for the number of epochs greater than 50 there are no significant changes in the forecast results. With this setup, LSTM RNN training for all apartments required 8.94 hours for the 3-1 ITF week ratio (≈200 sec per NN) and 2.36 hours (≈50 sec per NN) for the 2-1 ITF week ratio. It is relevant to note that three LSTM RNNs have been trained per apartment.

Longer intervals of occupancy and unoccupancy are typically correctly predicted, although small-term peaks are frequently skipped (see Figure 6.14a and Figure 6.14b).

In 66% of apartments, the ITF week ratio of 2 to 1 yields better results than a 3 to 1 week ratio (≈0.1 RMSE). This error is negligible while reducing the training time to 3-4 seconds per epoch is significant. For 75% of apartments, taking into account the 3-1 ITF week ratio, an increase of forecasting based on10min occupancy granularity over 1h is found. The ITF week ratio and occupancy granularity can be observed individually, per apartment, which means that the relationship between them can be learned from forecast similarity and RMSE (Exp. 1–7). This information enables selecting the best ITF week ratio and occupancy granularity individually, for each apartment. The ITF week ratio of 1-1 yields the worst forecasting results. Conclusively, 2-1 ITF week ratio and 10min occupancy granularity can be chosen as default – training time is less and error is negligible in most cases.

When 2 NNs are trained per apartment (1 for weekends, 1 for workdays) instead of 1 for the whole week, the predicted results increase in 75% of apartments for workdays and in 33% for weekends (see Figure 6.14a and Figure 6.14b). EDR was used to evaluate the success of forecasting rather than RMSE. RMSE is not directly related to the similarity of the EDR – lower RMSE does not imply a higher EDR similarity and vice versa.

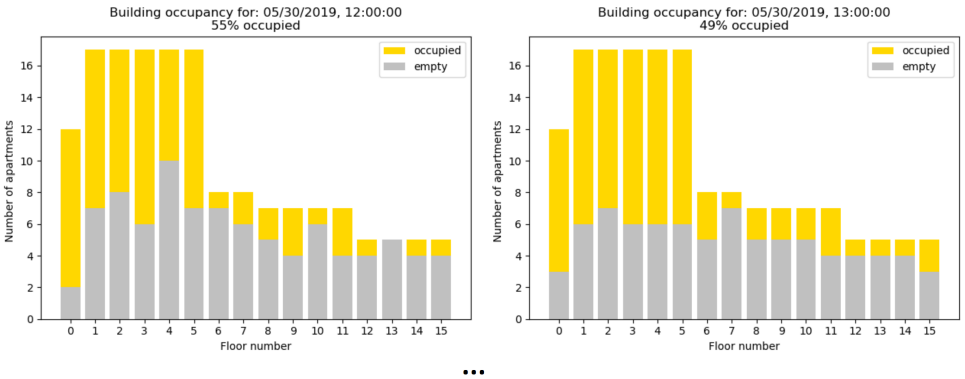
Finally the average EDR similarity for the entire building was calculated to be 68.24% for Exp. 2, 76.45% for Exp. 3–5 and 59.47% for Exp. 6. Although the fleet of trained NNs in the central building showed satisfactory performance in each case

(considering the complexity of the problem), the current maximum EDR similarity is 75.45% for working days. Results could be improved by collecting more historical data for RNNs and combining occupancy data with other data sources (social, environmental).

### 6.2.6 Visualization and Data Analytics

In addition , this section will present feedback on how these observations can be integrated into the SBM of the observed BLEMAT deployment site in order to save costs and adjust the usage of resources to the building / tenant needs.

BLEMAT can compute the occupancy percentage of the entire building per floor to visualize the occupancy of the building for that time range. As can be seen in Figure 6.15, approximately ≈50% of the building is occupied. The X-axis shows floor numbers, and the Y-axis shows the number of apartments that are occupied. Projected occupancy data such as this may be used to assist with SBM decision-making issues that rely on building occupancy in general, such as the amount of meals to be served in the common kitchen. Next, the information about the number/percentage of occupied apartments per floor for a given time range can be obtained (see Figure 6.16). The X-axis shows apartment labels, and the Y-axis shows binary occupancy. Figure 6.16 shows occupancy for the third floor, from 00–03 AM (from a 24h time-series visualized per hour). It is evident that the occupancy per apartment does not shift (with the exception of 1 point, circled red), which is accounted for by the fact that it is recorded during the night time and that the tenants are sleeping (user devices remain in the apartment). The information shown in Figure 6.16 displays lower granularity than in Figure 6.15, thus, it could be used in more fine-grained decision-making processes. Since the exact occupancy per apartment is forecasted, this information can be used to group apartments per floor and, if applicable, manage certain resources in order to diminish costs (scale




...

**Figure 6.15:** Average building occupancy over time.

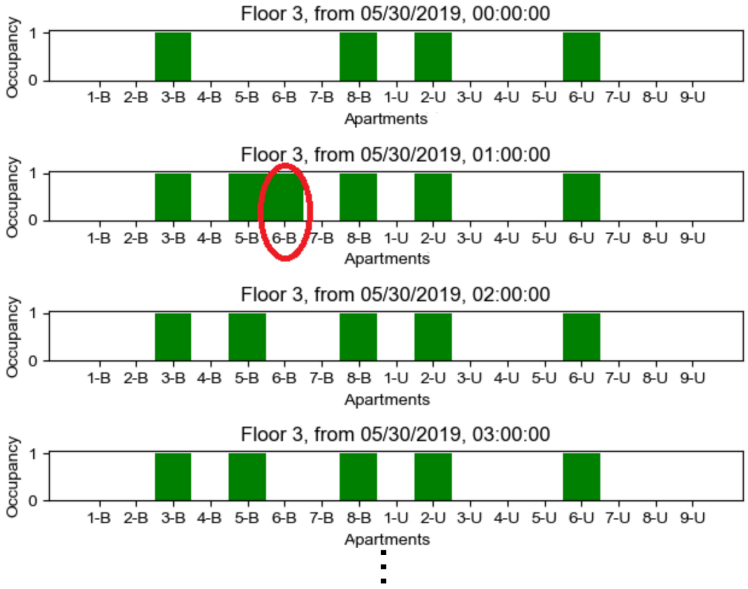



**Figure 6.16:** Floor occupancy per time range.

down internet access provisioning services, disable nodes [67] or enhance HVAC management based on the projected occupancy of apartments [104]). This data can also be helpful in emergency scenarios [46] where there is a need to check the occupancy of the entire building to help in efficient organization of emergency aid (i.e. firefighters) or planning agile emergency routes [119]. Moving on to even smaller granularity of occupancy data visualization, individual apartment occupancy can be plotted (see Figure 6.17). The plot's X-axis shows the time of the day (00–24) and the Y-axis shows binary occupancy. The plot on the left of Figure 6.17 shows occupancy for the apartment with the label *APT_4_FLOOR_4_U* for 1 week. The right plot in Figure 6.17 shows the same data for another apartment. Taking a look

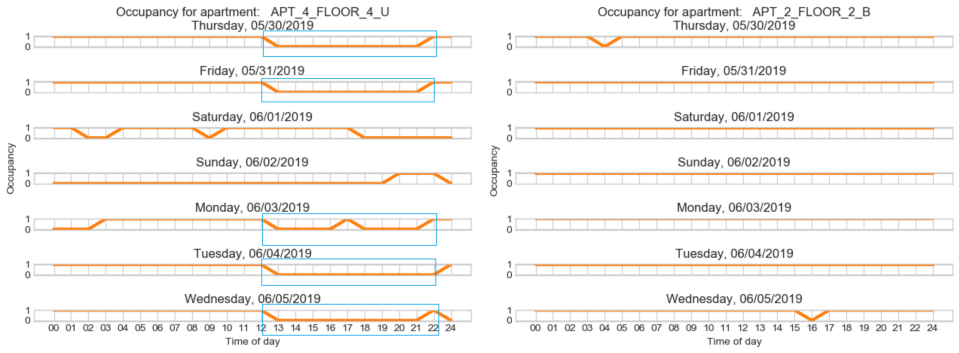



**Figure 6.17:** Apartment-level occupancy for a week.

at the left plot of Figure 6.17, some parts of the plot are bordered with a blue rectangle. At this level of information granularity, patterns are beginning to emerge – the occupancy of apartment *APT_4_FLOOR_4_U* follows a certain pattern: during the working days it is unoccupied between 13h–21h, while the occupancy is stochastic during the weekend. On the other hand, the apartment *APT_2_FLOOR_2_B* is occupied most of the time – this could be an indication that the device is static (i.e. Amazon Alexa, Siri or a smart TV) and not a mobile device (e.g. smartphone). On the forecasted data , different pattern matching techniques can be run at this granularity – these will be discussed in later sections.

### 6.2.7 Occupancy Patterns Exploration Approaches

This section provides insight into the search and clustering of occupancy patterns with BLEMAT, to find appropriate patterns to consider for an SBM to improve resource utilization control functions. This is also the ultimate objective for SBM – to extract and forecast occupancy data. The ability to infer occupancy trends and the actions of tenants can influence the operating efficiency of both resource usage (i.e. smart lighting, HVAC) and cost-saving strategies [204, 37].

Considering the occupancy patterns searching, it is most important to deduce the time delimiter period. In this work, a span of 1 week is chosen as a representative time period for pattern occurrences, for individual apartment-level tenant occupancy patterns. A week is an acceptable choice as it is the most common delimiter of how the behaviors and obligations of individuals are structured in time ( i.e. training, work, school). To avoid discovering false patterns, workflows for pattern searching should be repeatedly performed with fresh data.

#### Detecting and Clustering Patterns of Longest Occupancy/Unoccupancy

There are two steps to this analytical task: the first step is to detect the longest sequences when the apartment has been occupied/unoccupied, and the second step is to find regularity when these sequences are combined over a given period of time. 1 week is used as a time delimiter in this study and we will look at the individual days, split hourly (00-24) to define and uncover regularities in the individual tenant's behavioral patterns.

First of all, we tackle the question of detecting the longest sequences of apartment occupancy/unoccupancy. Since apartment occupancy for a day is given as a binary array of length 24 (1 or 0 for every hour), finding the longest occupancy/unoccupancy sequences comprises of finding the longest subsequences of 1 and/or 0. Longest sequences of unoccupied status (0) for apartment *APT_4_FLOOR_4_U* for 1 week are displayed in the plot on the left side of Figure 6.18, by days. On this plot, the X-axis shows the hour of the day, and the Y-axis displays the length of the longest sequence. For example, on Tuesday, the longest

sequence of unoccupied status started at 13:00 (see X-axis) and lasted for 10 hours straight (see Y-axis). A similar subsequence repeats for Wednesday, Thursday, and Friday. This particular apartment has already been a part of previous analysis (see Figure 6.17) and some regularities in its occupancy have already been discussed. Visualization in step 1 is beneficial, but autonomous mechanisms need exist in order to provide data on which apartments have specific occupancy trends and which simply do not. Thus, a clustering method was implemented in step 2 to detect relevant occupancy/unoccupancy trends. This approach is displayed in the plot on the right side of Figure 6.18. For clustering, K-means is used, and $K$ was varied between 3 and 4. $K = 2$ is too small for a good K value because it might not put real outliers to appropriate clusters, but $K = 3$ and $K = 4$ have both shown good results for finding clusters of patterns that are indeed representative. The cluster analysis of patterns of longest sequences of unoccupied status for apartment *APT_4_FLOOR_4_U* was run with $K = 3$ and has found 2 significant clusters. A cluster is determined to be significant if it has 2 or more elements, although, this also can be varied depending on what type of pattern repentance are observed – in this case, we are trying to find patterns of a tenant that goes to work or school, and usually these types of responsibilities occur at least twice a week. The most significant cluster, **Cluster 1** has 4 points (2 points overlap; the cluster has 4 elements) and corresponds to the pattern also observed in the plot on the left of Figure 6.18 (Tuesday–Friday). Notice that it is possible to perform step 1 and step 2 independently. It is more informative, however, to run them in a batch, because step 2 complements the knowledge from step 1. In addition, these steps should be performed periodically for all apartments, to check whether and how the individual habits may have changed.

By running these steps on all apartments very specific information per apartment can be extracted. It can be verified if the tenant from a specific apartment goes to job/attends school (by finding relevant unoccupancy patterns that repeat

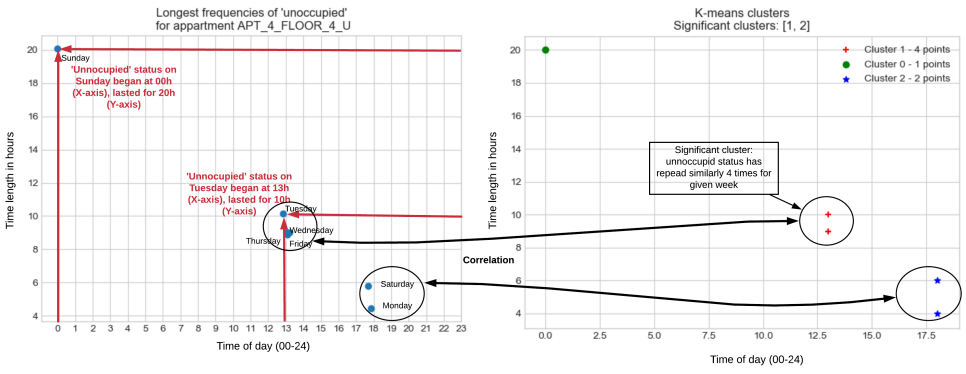



**Figure 6.18:** Detecting and clustering patterns of longest unoccupancy.

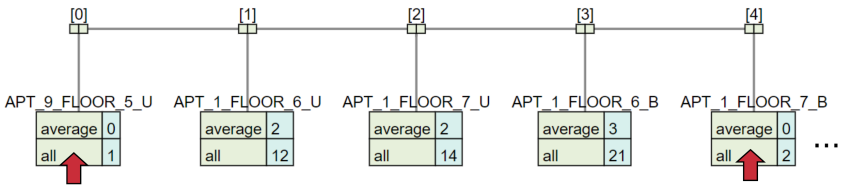



**Figure 6.19:** Average number of times tenant leaves the apartment per day.

more than two times a week). In general, one can extract clusters of patterns per apartment that are relevant for deeper inspection and other data analytics pipelines, varying what patterns are found relevant.

### Detecting Specific Patterns and Regular Expression Patterns

There are two more data analytical tasks that BLEMAT conducts, moving deeper into pattern analysis for occupancy data: finding user-defined patterns in occupancy data and finding patterns that conform to regular expressions provided. When unique patterns need to be supplied, identifying user-defined occupancy sequences for pattern searching and matching is an useful functionality. BLEMAT returns an array of indexes of all subsequences in the occupancy data for the supplied particular pattern. If the resulting array is $[0, 5, 9]$ that means that the supplied pattern repeats at hours 1, 5, and 9 (AM). As such, the *occupancy_1week_1h.json* dataset (see datasets on Zenodo [151]) was analyzed for different specific occupancy patterns. The dataset contains binary hourly occupancy for all apartments for 1 week. By providing a unique pattern *'100000'*, implying that the apartment was occupied at the starting point, but then for 5 consecutive hours it was unoccupied, all the tenants who have jobs or attend school will be identified. Similarly as above (Figure 6.18), if the pattern repeats three or more times during a week, the tenant is put into the category that has a job or attends school. Out of 149 apartments, 23 have this characteristic. Next, let's try to detect if there are apartments that are occupied most of the time during the day (but not the entire day), by supplying a simple pattern *'1111'*, and checking if the length of the list of repeats of this pattern is between 3 and 15. Everything over 15 fits the category of a static device. In the chosen dataset, 49 tenants are inside a lot of time during a day, while 100 are not, including common rooms. Twelve apartments were classified to have static devices with this analysis. The regular expression can be supplied as patterns in a form containing numbers 0 or 1. For example, let us analyze the data for pattern *'1+0'*. This pattern represents a sequence of apartment occupancy that was cut off at one point (tenant left the apartment). Figure 6.19, shows this information aggregated for all apartments, on the *occupancy_1week_1h.json* dataset. This particular pattern

can be used to infer the number of times the tenant leaves the apartment per week ('all'), as well as the average number of such events per week ('average'). Aggregated for all apartments, data in this form can also show if there are apartments that are seldom unoccupied (*APT_9_FLOOR_5_U* and *APT_1_FLOOR_7_B*) – inferring that occupancy data is potentially extrapolated from a static device (see Figure 6.19).

**Autonomous Pattern Searching Without Provided Patterns (STUMPY)**

STUMPY is a data analytics library for analyzing time-series, created through the joint efforts of University of California, University of Sao Paulo, and University of New Mexico researchers [99]. The STUMPY library efficiently computes a matrix profile based on input data. A matrix profile is a vector that stores the z-normalized Euclidean distance between any subsequence within a time-series and its nearest neighbor. The matrix profile can be used for a variety of time-series data mining tasks such as anomaly/novelty (discord) discovery, pattern/motif discovery, density estimation, etc. In order to help finding related subsequences within the time series that can be used to assess both recurrent trends and outliers, BLEMAT has incorporated STUMPY into its Data Analytics Engine. Two pattern analysis mechanisms are provided by the BLEMAT Data Analytics Engine: apartment-level and building-level.

Before STUMPY analysis was carried out on the data, a new dataset was constructed to extract the average daily occupancy per apartment (in percent) for 4 weeks. Apartment-level analysis for apartment *APT_4_FLOOR_4_B* with STUMPY was executed with *occupancy_4week_1h.json* dataset, and the results are displayed in Figure 6.20.

Figure 6.20 displays STUMPY's detection of patterns or motifs for a moving window size of 2, 5, and 7 days. Motifs represent approximately repeated subsequences within a longer time-series. The lowest values in this graph are considered the best motifs as they represent that the corresponding subsequence window (on the X-axis) has the closest subsequence with distance $j$ (on the Y-axis). As seen in Figure 6.20 (red triangles), when the window size is 7 days the 4 lowest data points are approximately 7 days apart each, suggesting that in this dataset there is a periodicity of 7 days. From the point of view of the dataset and the pattern discovery issues, this makes sense because a week is reasonable to be the most common delimiter of how the responsibilities and behaviors of people are often seen in their everyday lives. There are no major motifs for window sizes for 2 and 5 days. The indices with the highest values on the graph reflect the uniqueness of their corresponding subsequence, indicating that no other subsequence (in terms of the euclidean distance) is relatively near to the subsequence observed. There is a peak on the plot for 7 days on June 18th, showing that on that day, relative to other occupancy segments, the apartment occupancy trend was uncommon.

From the same dataset, the average hourly occupancy of the central building was extracted (see Figure 6.21). It is noticeable that the occupancy dropped between June 13 and June 20, however between June 18 and June 19 there is a drastic drop in occupancy.

On this data, the STUMPY matrix profile was built and plotted (see Figure 6.22). As expected, STUMPY detected the same unusual subsequence on the data as in Figure 6.21. Otherwise, for none of the window sizes, insightful trends could not be extrapolated from this profile, indicating that there is no periodicity in the average hourly occupancy of the whole building. However, to identify irregular peaks and drops in average building occupancy and use this knowledge to assist in SBM decision-making, STUMPY analysis on this type of forecasted data may be used.

### 6.2.8 Discussion

Detection and forecasting of patterns and trends in indoor areas occupancy create a great opportunity for scientific contributions in terms of models, techniques, and algorithms. As an ECP framework, BLEMAT not only offers LBSs but also a set of machine learning, context learning, and data analytics models to assist in a thorough understanding of the operating context of the system. This includes, but not limited to the occupational context of the space being observed, occupancy forecasting, and insightful information retrieval of both current and forecasted occupancy data. BLEMAT is deployed in a real-world residential building with 149 apartments and 12 common rooms, and being closer to the building's SBM itself, insight can have a greater impact on efficient usage of resources and cost-saving strategies. To apply the solutions described in this work, a set of minimum re-

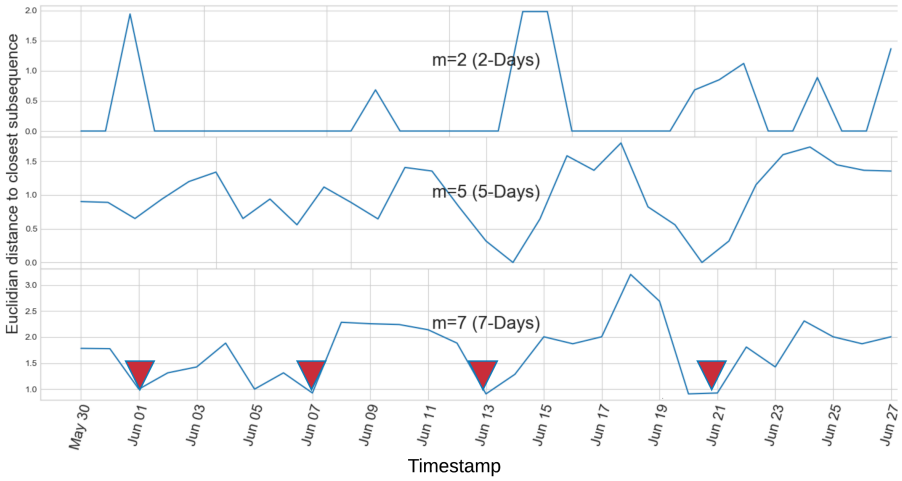



**Figure 6.20:** 4-week STUMPY profile with changing window sizes for *APT_4_FLOOR_4_B*.

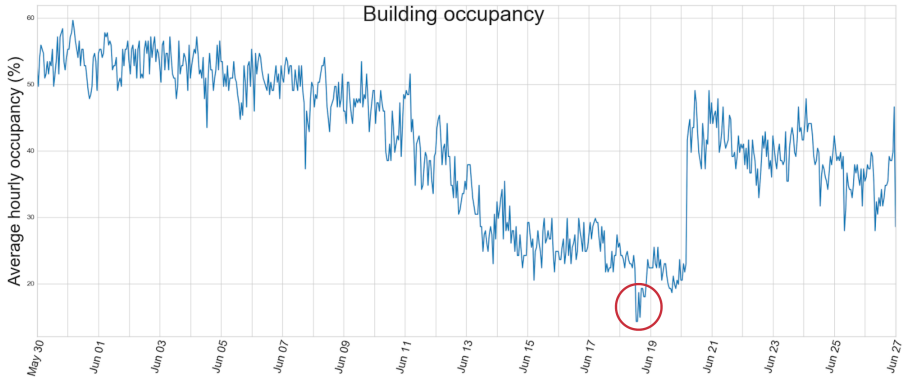



**Figure 6.21:** 4-week average hourly building occupancy.

quirements consists of having a BLEMAT-enabled infrastructure: a software stack for Wi-Fi or BLE (or both) scanning and data acquisition, and a 3D building plan modeled in code. Once correct transformations (from this work) are applied to the data obtained, the methods and models that have been presented can be applied effortlessly.

Bluetooth occupancy data was merged with Wi-Fi occupancy data in this work. It is noteworthy how these two occupancy datasets overlapped in approximately 80% of all apartments, suggesting that Bluetooth-based occupancy estimates could be considered without merging with Wi-Fi occupancy data, minimizing data merging and pre-processing times.

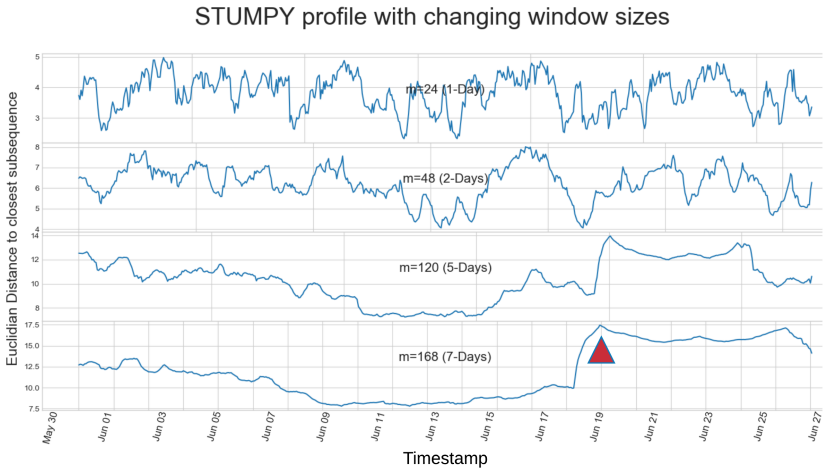



**Figure 6.22:** 4-week STUMPY analysis on average hourly building occupancy with changing window sizes.

While most SoTA research addresses the short-term and discrete occupancy forecasting problem in public and commercial buildings, in this work, the long-term binary occupancy time-series forecasting problem is solved in a large, private residential building. Firstly, it is more difficult to predict binary occupancy in residential buildings than in commercial buildings, as most tenants behave stochastically most of the time in the former type of buildings. However, if there are already existing patterns of occupancy in the behavior of tenants, forecasting will be more successful, which is normal. Second, predicting binary occupancy is a more difficult issue for deep learning than predicting discrete occupancy (number of tenants). Finally, long-term forecasting requires a good set of data and a finer calibration of the underlying RNN model.

Emergency use-cases and their relationship with occupancy detection have been tackled many times in the sections above. While occupancy detection may infer information about the individual occupancy of the apartment, static (always active) devices must be carefully filtered (may infer incorrect occupancy). Rescue decisions based on the BLEMAT Occupancy Detection should not take these devices into account. The occupancy information can, on the other hand, suggest that someone might still be inside the building in an emergency scenario. Besides, according to the expected occupancy, special, personalized evacuation emergency strategies can be developed per vertical or horizontal area.

Categorizing mobile and static user devices would make a good differentiator when inferring occupancy. A binary classification approach to this problem can be implemented to make decisions as to whether a device is mobile or static. At the moment these decisions could be made based on occupancy sequences, however, the mobility patterns of these devices (sequence of positions over some time) which BLEMAT also collects would be another relevant input. Next, using the existing context of the ground floor, which contains the physical entrance to the building, information on mobile devices can be confirmed (as static devices will not leave the building).

From a business point of view, mobility patterns and occupancy data could be used to model such business decisions as to where to place a vending machine, by finding the area with the largest average occupancy, or by collecting points (2D or 3D) that are highly repeated in different device mobility patterns.

## 6.3 Geofencing

Indoor asset tracking and positioning are an increasingly relevant feature of IoT solutions across industries. Businesses using LBSs can accurately locate, monitor, manage, and secure fleets of connected devices in real-time. Location data can be used as a qualifier for interpreting, organizing, and translating large collections of IoT data into actionable insights.

A geofence is a virtual perimeter that encapsulates a certain target area. User or object coordinates are used to assess if they are within or outside the target area, as well as whether they cross in or out of the target area. Depending on how the geofencing is configured, it can prompt mobile push notifications, trigger text messages or alerts, send targeted social media advertisements, allow vehicle fleet tracking, or deliver location-based marketing data. A retailer may place a virtual geofence around its stores to activate smartphone alerts for customers who have downloaded the smartphone app from the retailer [73]. Geofencing can be used to control and track vehicles in the shipping industry[147] or cattle in the agriculture sector[175]. Applications in e-health are also ample [64]. The safety and security of personnel is another important role of geofencing across industries [206].

Built on top of BLEMAT, the Geofencing Micro-location Asset Tracking (GEMAT) framework is a semi-unsupervised geofencing system that constantly learns about the operational context of the environment it is deployed. In an operational context like BLEMAT, geofencing capabilities are automatically enhanced by the solution's functionalities that aim to correct positioning calculations, based on the changes in the context of the managed system. By using BLEMAT's data streams and services, in the rest of this section, a context-aware, self-adapting geofencing management solution with straightforward, on-demand geofence updates is presented.

### 6.3.1 SoTA Approaches

Geofencing, especially outdoor geofencing, is not a novel subject for LBSs [140]. There have been many advancements in tourism geofencing [130], crowd-sensing and mass-gathering management [12], management of parking spaces [167], location-based marketing citemarket, etc. Although outdoor geofencing has been well researched, indoor geofencing lacks in research achievements aimed towards deployments considering efficient resource utilization and adaptation to contextual changes in the physical environment.

The capabilities of a geofencing system, similar to those of this study, have been proposed by Masafumi [138]. However, advanced methods for geofencing (geofence chains and temporal geofences) have not been studied. Besides, Shuo et al. suggest a camera-based vision-analysis geofencing system for thermal comfort and energy savings through indoor occupancy tracking [114]. It is a step forward in discussing the collaboration between IPS, geofencing, and IoT M2M notifications and actuation. The aim of this thesis is similar, with an important differentiator: to elaborate on experiments and to discuss the importance of the context-awareness of the underlying IoT system. Cardone et al. present MoST, a geofencing architecture that is a complete and complex geofencing system [13]. MosT is a resource and context-aware while providing additional geofencing capabilities (temporal ge-

ofences, geofencing relationships, etc.). With MoST, information feedback loops are not discussed – how information about the physical context needs to be transferred back to the geofencing system to improve its accuracy and operational correctness.

As another method, Namiot et al. claim that geofencing based on location could be replaced by geofencing based on network proximity [139]. By using the spectral fingerprinting RSSI for proximity detection, the geofencing areas can be set as areas with previously calculated fingerprints (per area granularity: can be $1m^2$ or less). While this is a novel approach, the contextual awareness of the system must be carefully considered [25]. If the context in which the system operates changes, the spectral fingerprint of the indoor area must also be regenerated. If not, this may lead to false positioning and proximity information. BLEMAT and GEMAT have a mechanism for refreshing spectral fingerprints to address this challenge.

It is stated in some research that the idea of a single, separate, non-temporal geofences is often inadequate to cover more complex use cases and to provide effective location-based alerts [118, 168]. These principles are taken into account in GEMAT, as it allows the generation of chains of geofences, as well as the identification of a temporal component inside a chain or an individual geofence.

Ultimately, there are not many geofencing research papers concerned with the notion of security or privacy. Since the privacy of the users is essential to LBS, Guldner et al. [61] address the challenge by proposing an LBS and geofencing system based on homomorphic encryption of the position of observed users[181]. GEMAT and BLEMAT are SoTA LBS edge computing systems, thus achieving a high level of privacy – all sensory data remain at the edge of the network. Besides, accessing location data is based on specific access-control policies, and location data is stored in an encrypted manner by leveraging Key-policy-based encryption.

### 6.3.2 Functional Architecture

GEMAT rests on three main components: GEMAT area and rules manager, GEMAT rules enforcer and GEMAT notify engine (see Figure 6.23).

The **GEMAT Area and Rules Manager** component is responsible for storing geofencing areas and rules as well as for notification or actuation. A geofencing area is dynamically determined and marked in the GEMAT as an arbitrary polygon or line on the indoor floor plan. Besides defining geofences, this aspect involves the possibility of establishing temporal connections between geofences, as well as time constraints within the geofences or in the transition between geofences. These are two very important aspects (also highlighted in [168]) required to cover sophisticated scenarios in which notification should be activated only if the user crosses multiple geofences in a given temporal order or leaves a geofence after a

certain amount of time. The geofencing rules are defined following the *'if this-then that'* logic. The first part of the rule refers to the creation of a relationship between a geofencing area and a tracked object. It also includes the creation of relationships between different rules of geofencing. The second part of the rule describes the communication behavior to the GEMAT Notify Engine to provide details on the geofencing rule that has been triggered.

The **GEMAT Rules Enforcer** represents an automated geofencing mechanism that works with the positioning data provided by the BLEMAT system and the GEMAT Area and Rules Manager to determine whether an action needs to be considered (i.e. a geofencing rule has to be triggered).

The **GEMAT Notify Engine** takes the outputs of the GEMAT Rules Enforcer and transforms them into actions – sends a notification or an actuation command to the desired IoT device. The notification mechanism delivers user notifications to designated devices (specified with the rule). Besides, in an IoT system where actuation is an integral part of system operation, the GEMAT Notify Engine can be used to send actuation commands to other IoT devices and controlled actuation systems.

GEMAT facilitates the monitoring of the entry and exit events for designed geofence areas. Geofences for GEMAT can be supplied via the BLEMAT administration dashboard, where it is possible to position arbitrary geofences on the underlying floor plan. This is suitable for use in situations where traveling through the borders of a protected area needs to be registered and/or monitored, such as: controlling access to specific parts of a larger area ( i.e. access to specific production machines/static assets), tracking of mobile assets across multiple geofences,

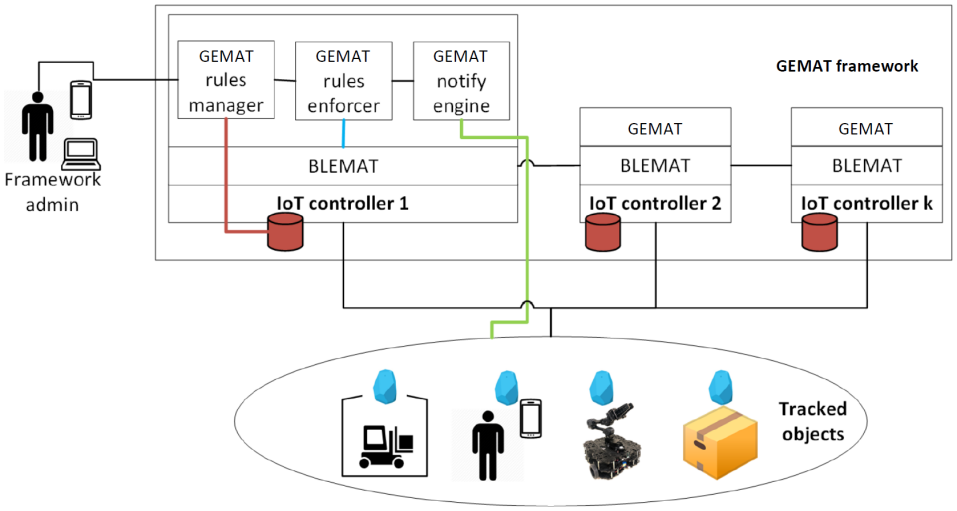



**Figure 6.23:** GEMAT workflow.

etc. The GEMAT solution, via the BLEMAT system functionalities, dynamically adapts geofencing parameters (referent signal/spectral maps, derived positions) to address changes in the operational, physical context of the system (moving barriers that worsen signal propagation, unknown interferences and the movement of people and devices through the observed indoor area). Thus, GEMAT is constantly learning contextual information from the physical environment of the underlying system.

### 6.3.3 Performance Analysis Model for Geofencing Frameworks

To measure the capability and performance of a geofencing framework, ten criteria need to be taken into consideration. The set of criteria is based on papers referencing geofencing frameworks design and implementation. The criteria are as follows:

C1. Location sampling interval;

C2. Support for temporal geofences;

C3. Geofence shape;

C4. Support for geofence relationships;

C5. Positioning accuracy;

C6. Adaptation to changes in the system context;

C7. Resource consumption;

C8. Visualized geofences specification;

C9. Support for other LBS;

C10. Timestamps deviation.

The location sampling interval should be dynamically modified to save battery life without affecting the geofencing capability. If the object being tracked is close to the geofence, the interval should be lower, and vice versa. Temporal geofences mean that an action can be specified if an object stays within a geofence for a certain amount of time. Geofence shape is relevant for more sophisticated applications. Being able to create geofence chains in the form of geofence relationships is useful in applications where it is important to track objects through multiple geofences to perform a single action. The positioning accuracy and precision are linked to the underlying positioning technique used. Dynamic adaptation to the

| Paper/Criterion | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | C9 | C10 |
|---|---|---|---|---|---|---|---|---|---|---|
| (1) GEMAT | Dynamic | ✓ | Arbitrary | ✓ | <1m | ✓ | Low | ✓ | ✓ | <2s |
| (2) Google geofencing | Inherited | ✗ | Radius | ✗ | Inherited | ✗ | Low | ✗ | ✓ | Unknown |
| (3) MoST | Dynamic | ✗ | Arbitrary | ✗ | ∼100m | ✗ | Controlled | ✓ | ✓ | Unknown |
| (4) geofencing 2.0 | Unknown | ✓ | Arbitrary | ✓ | Unknown | ✗ | Controlled | ✓ | ✓ | Unknown |
| (5) geofencing Cooling Fan System | 2s | ✗ | Object | ✗ | <1cm | ✓ | Medium | ✓ | Unknown | Unknown |

**Table 6.2:** Comparison of geofencing frameworks.

system context is critical for dynamic system environments where signal propagation changes and/or barriers are introduced dynamically. Resource consumption is relevant from the standpoint of where the geofencing system is deployed. It is important to have a simple geofences specification process ( i.e. through a visualized dashboard). The percentage of missing geofences is an important factor, although it is also related to the trade-off between positioning accuracy and location sampling. One also needs to evaluate support for LBS, other than geofencing, and how easy it is to integrate with the geofencing system. Finally, the difference between the reported and real timestamps of the triggering geofences must be reduced to a minimum, with the maximum geofencing precision and support for advanced use cases.

To the best of this author's knowledge, Table 6.2 is the first to summarize a performance analysis model for geofencing frameworks and compare them. Table 6.2 shows how GEMAT compares to other similar geofencing frameworks taking into consideration C1-C10. While the Google geofencing Library[56] provides an extension of Google LBS, it is not a standalone geofencing system thus the sampling interval and positioning accuracy are inherited from the application use-case leveraging geofencing. It does not provide temporal of geofence relationships, and geofences are specified only as circles. On the other hand, MoST is the most robust geofencing system that exists, with features to accompany each criterion. Nevertheless, it is based on the outdoors. It is important to note that in MoST, the use of resources is regulated by controlling the interval of sampling. In addition, depending on the distance between the tracked object and the nearest geofence, the sampling interval is dynamically adapted. The geofencing 2.0 [168] framework ((4) in Table 6.2) also supports managing resource consumption, and satisfies most of the criteria. GEMAT and (5) from Table 6.2 are the only two frameworks considering design towards context-awareness as a default setting. Furthermore, (5) is a camera-based framework providing sub-centimeter accuracy, but with a heavy load on resources consumption and significant costs (because of the camera and image processing) [114]. On the other hand, the shape and model of the geofence is greatly enhanced with this approach – objects that the camera sees can also be defined as geofences (i.e. a surface of the table), and are triggered when that object is used within an activity. Lastly, GEMAT is the only framework to consider the timestamps deviation metric.

In conclusion, GEMAT compares well with other similar frameworks. The summarization of GEMAT through criteria C1-C10 is presented below:

**C1. Location sampling interval** – the underlying positioning system BLEMAT handles specification of location sampling intervals. Since GEMAT is not the only LBS using positioning data from BLEMAT, sampling intervals cannot be automatically updated as in MoST.

**C2/C4. Support for temporal geofences and geofence relationships** – although support for temporal geofences and geofence relationships definition is enabled, a deeper theoretical model (i.e. a domain-specific language) for defining them (such as in geofence 2.0 [168]) and translating their specification to a programming language of choice (to be incorporated as a geofencing rule) is not yet researched and will be a part of future work.

**C3. geofence shape** – support for arbitrary shapes is enabled in GEMAT, and this is a general flexibility need for indoor geofencing frameworks.

**C5/C6. Positioning accuracy and adaptation to changes in the system context** – are inherited BLEMAT functionalities. The accuracy is sub-meter (which is enough for most of IPSs), and the process and workflows of how the IPS adapts to contextual changes are well described in a previous paper [156].

**C7. Resource consumption** – as BLEMAT was designed as a fog-computing system, without any cloud resource dependencies, and with strict resource utilization policies in mind, GEMAT is inherently resource-preserving.

**C8. Visualized geofences specification** – BLEMAT has a dashboard to track and explore positioning data for the underlying floor maps, and as an extension of the dashboard, GEMAT offers visualized geofences specification on the floorplan by drawing of arbitrary shapes. Visualized specification of temporal geofences and geofences relationships will be a part of future work on GEMAT.

**C9. Support for other LBS** – having BLEMAT as an underlying IPS opens up possibilities for implementing other LBS that leverage past or present positioning data.

**C10. Timestamps deviation** – Out of the evaluated ones, GEMAT is the only system to consider the timestamp deviation metric. This makes a step forward in defining general quality metrics for other geofencing frameworks.

In GEMAT, due to the low position sampling period, the deviation is small. Nevertheless, higher variance between recorded and actual geofence triggering timestamps is the predicted outcome for an increased position sampling interval.

### 6.3.4 Experimental Results

There were two datasets obtained for the experiments. Positioning data was collected within one week for the first dataset (DS1), every day from 8h–20h, and 300,000 position measurements (sampling interval of 1s) were collected during that time. For the second dataset (DS2), positioning data was collected within one day from 8h–20h, and during that time 7.000 position estimations (sampling interval of 10s) were collected. Six users had their phones monitored via GEMAT, and all were included in triggering scenarios for geofences monitoring activities. Geofences were activated 6,000 times over the duration of data collection, and information about the scenarios it addressed and the performance of the scenarios was collected for each triggered geofence.These datasets were collected in deployment site 1 (see Section 5.3.1). The proposed GEMAT framework and it's functionalities were tested in two experiments. Mobile assets (MA) are simulated by users (six are present in the experiments), and during the experimental phase, these users have triggered the programmed geofences (by walking) 6.000 times. The experiment incorporates Mikrotik routers (MT) as a means of increasing signal interference in some areas and serving as contextual adjustments to the system's spectral characteristics. Metal cabinets (CAB) are also introduced as signal obstructors: they block a clear path between a source and destination of the Bluetooth signal. A disruption in signal propagation is caused by the presence of reflective surfaces of people, metal artifacts, or other obstacles or radio frequencies (RF). The same may be achieved by other electrical equipment emitting heavy RFs. Since Wi-Fi also uses the 2.4 GHz bandwidth, these two signals frequently conflict with each other. The Static object ($S_1$) is a Zigbee E26 Smart Light Bulb, capable of execution of actuation commands that are received via Bluetooth, from BLEMAT-enabled IoT gateways.

GEMAT experiments were aimed at proving the following hypothesis:

(**EX1**): the GEMAT system is capable of handling both simple and sophisticated geofencing use cases (multiple geofences are tied, and there is a temporal component to them).

(**EX2**): the GEMAT achieves consistent accuracy when the operational context is changed (signal is obstructed on purpose).

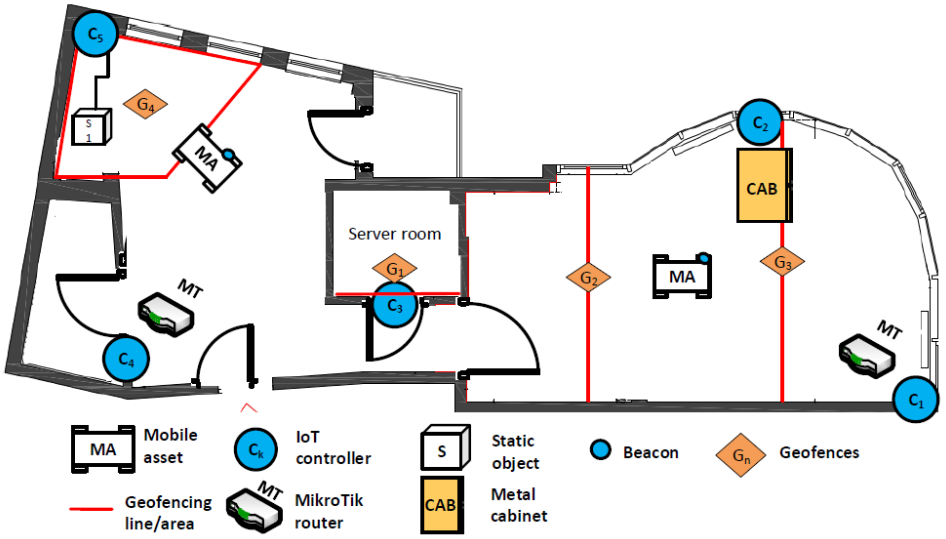



**Figure 6.24:** Experiment indoor area.

**EX1** was motivated by the IPS systems and geofencing approaches described in the geofencing SoTA Section (Section 6.3.1), to show how GEMAT can provide advanced geofencing capabilities while also being able to learn about the system's context. The motivation behind **EX2** was to explore the geofencing capabilities of GEMAT, while introducing constant signal obstruction, and thus directly impacting the system's capabilities to provide accurate positioning calculations.

A smart warehouse scenario was emulated through the observed office space, where mobile objects are tracked through BLEMAT. Based on positioning data and the geofencing rules established with GEMAT, appropriate actions are taken. Let us examine the geofences that have been programmed into the GEMAT experiments:

(**SC1**): $G_1$ represents the entrance to the server room. When a user triggers this geofence ("if this" part of the rule), this information is persisted in GEMAT's database ("then that" part of the rule).

(**SC2**): $G_2$ and $G_3$ work together to achieve the following: when a mobile asset (MA) has passed geofence $G_2$, it has under 20 seconds to pass geofence $G_3$. If this does not happen in this order and under this time constraint, then a notification is sent to the system administrator mentioning a potential problem with the mobile asset (asset has stopped moving) so that it can be inspected.

(**SC3**): $G_4$ represents a secured area, with a static object $S_1$ inside. When a mobile asset crosses the boundaries of geofence $G_4$, the GEMAT generates an actuation command towards $S_1$. In a real-world warehousing environment, $S_1$ might represent a production machine/moving product line that is triggered when an MA passes into $G_4$. In this case, to demonstrate the actuation framework and the GEMAT collaboration in the underlying system, when $G_4$ is triggered, the light is turned on ($S_1$ is a Zigbee E26 Smart Light Bulb).

During the environmental tests conducted for the **EX1** experiment, all three scenarios (**SC1**, **SC2**, **SC3**) were verified to be successful in the execution of the geofencing rules. Mobile assets were represented by human users holding Bluetooth beacons (smartphones) on them. $G_1$ was straightforward to implement and test. Also, the $G_1$ rule activates a database operation (no notification or actuation). Geofences such as $G_2$ and $G_3$ are more complex to implement. The GEMAT Rule Enforcer must continue to check both $G_2$ and $G_3$ at all times, and if there is a breach of the rules, appropriate action shall be taken (in this case notification to a particular system user). Geofence $G_4$ is an area-based geofence with a rule that triggers an actuation command to turn the lights on.

In an experimental setup with low signal disturbances and a low location sampling interval of 1s (**DS1**), GEMAT was successful in triggering $G_1$ and $G_4$ 100% of the time. Temporal and relationship geofences $G_2$ and $G_3$ were correctly triggered 97% of the time (considering both the success and failure scenarios described above). However, for **DS2** where the location sampling interval was set to 10s, the percentage of triggered geofences is only 42% for $G_1$ and $G_4$ and 37% for $G_2$ and $G_3$. With the lowest sampling range of 1s, the difference between the real and the captured timestamp is negligible. While it is not recommended to increase the sampling interval in an IPS where geofences are relatively close to each other, in this work the sampling interval was increased to 10s (**DS2**). For IPS, the sampling interval must be as low as the complexity of the geofences increases.

In the second experiment (**EX2**) signal obstruction obstacles were introduced in two ways: (a) two additional access points were added (MikroTik HAP AC routers – see Figure 6.24, labeled as MT) at the same frequency, and (b) a metal cabinet was placed in front of one of the IoT gateways (Figure 6.24, labeled as CAB). Besides, the following GEMAT geofencing capabilities have been tested: (1) when BLEMAT contextual learning is disabled and (2) when it is enabled. This means that for (1) GEMAT worked with raw, unfiltered, and uncorrected RSSI and positioning values, while for (2) the context was taken into account and the values were filtered and corrected by BLEMAT. Without BLEMAT context-learning, geofences were triggered in 62% of cases, which is below the normal operational functionality of the geofencing framework. On the other hand, where BLEMAT contextual learning is used, in 96% of cases, geofences have been triggered, which is a major improvement over the first experiment. Besides, for experimental use

with $G_2$ and $G_3$ for experiment (1), there were 17% instances where $G_3$ was correctly triggered and $G_2$ was not triggered, so notification was omitted.

Finally, it was studied how the actual timestamp of passing the geofencing boundaries relates to the timestamp registered by GEMAT. The conclusion is that the timestamps deviation is directly related to the location sampling interval, as confirmed by both **EX1** and **EX2**, confirmed with both datasets **DS1** and **DS2**. The lowest timestamp deviation in GEMAT is reached when the position sampling interval is 1s and the timestamp deviation is below 2s. However, for **DS2**, the difference between timestamps is higher than expected and ranges between 2–10s.

### 6.3.5 Discussion

It is difficult not to overstate the value of LBS and geofencing for IoT systems [18]. IoT devices, however, lack affordable and easy-to-deploy geofencing solutions that fit well with the current networking stack (Wi-Fi, Bluetooth) already deployed in indoor spaces. In addition, most of the above-mentioned indoor positioning systems with geofencing capabilities concentrate on geofencing based on proximity and ignore more advanced use cases where a geofencing is an arbitrary polygon in the space observed. In addition, there are few IoT geofencing frameworks that allow the dynamic development of geofences and their on-demand updates, as well as those that take into account high geofencing specification levels (geofence relationships and transitions from one geofencing area to another, geofencing stay length, etc.).

BLEMAT is an edge computing indoor positioning system that successfully adapts to continual updates of dynamic IoT systems (varying resource utilization, node losses, signal obstruction, etc.), while increasing the accuracy of the underlying positioning techniques [156]. While understanding context increases overall IPS accuracy, it also ameliorates machine-to-machine (M2M) communication performance, which is an essential element for the IoT [150]. Geofencing frameworks need to account for contextual changes in the underlying IPS. Conducted experiments and performance analysis show that the proposed GEMAT framework is a good candidate for solving problems in a wide range of indoor geofencing use cases.

## 6.4 Floor Plan Layout Detection

Detection of a floor plan layout refers to inferring information about the physical obstacles in the observed indoor space. Most IPS have information about the exact layout of the indoor space where they operate and this is considered as ground truth in the system that never changes. This is enough for some indoor spaces such as office buildings, residential buildings, public buildings (e.g. shopping malls,

universities) and other similar types of buildings, becuase the physical layout of such spaces will not change over time, at least not significantly enough to impact the accuracy of the underlying IPS.

On the other hand, there are indoor spaces, such as large storages, that are rather simple, with no dedicated physical layout. These types of indoor spaces do not have a lot of walls, except for the external walls, and typically have a large set of equipment, movable separating walls, machines, large racks/shelves and other similar assets that count as obstacles for signal propagation in an IPS. In these types of indoor spaces, these assets are bound to change places and 'mass': large machines (i.e. forklift) can move around, racks/shelves can be emptied and refilled, and separating walls can be displaced based on where they are needed at the time. Thus, in these indoor spaces one can not assume that a floor plan layout that was setup as the ground truth at the beginning of system operation will stay the same forever. Because of this, there has to be an efficient way to deduce how these obstacles have changed and how the signal propagation is impacted. In the few following sections, an autonomous (i.e. unsupervised) solution to approximate a floor plan layout based on gathering positioning data of the observed Bluetooth beacons (people, machines, etc.) will be presented.

### 6.4.1 SoTA Approaches

Unsupervised detection of floor plan layouts using contextual data in indoor spaces is an under-researched topic in academia [77, 209]. SmartSLAM enables unsupervised construction of floor plan layouts, using odometry tracing with inertial sensors, Wi-Fi radio maps, and Bayesian estimation [182]. It uses the history of position estimations and Wi-Fi observations to build a representation of the floor plan, achieving good results for low to average complexity floor plans. Using crowdsourced smartphone data, odometry tracing, and clustering using dynamic time warping similarity criteria Haiyong, et al. propose another approach to building estimations of floor plan layouts [121]. Mobile crowdsensing for indoor floor plan building was also research by Gao, et al. [50].

The approach from this thesis rests on contextual data as well (RSSI), but does not require any sensory data (e.g. an odometer, pedometer, etc.). Crowdsensing in this approach is carried out by observing beacons, extracting their typical paths and quantifying them, without extra hardware (such as a mobile device). Thus, the approach described here abstracts more over the operational context of the IPS than the mentioned approaches, while achieving good results.

### 6.4.2 Matrix-based Indoor Space Model

To explain how the floor plan layout detection works, first, data acquisition and preprocessing required for that task will be presented.

$$M = \begin{bmatrix} S1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & S3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & S2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & S4 & 0 \end{bmatrix}$$

**Figure 6.25:** Initial (empty) space matrix with gateways information.

BLEMAT models and keeps a matrix representation of the indoor space. The matrix model of the observed indoor space is presented as a 2D $n$ x $m$ zero-matrix, where $n$ refers to the maximum length, and $m$ to the maximum width of the space (see Fig. 6.25). Matrix fields marked with $Si$ represent the deployed BLEMAT scanners (BLEMAT-enabled IoT gateways) in the matrix. To get insight into the physical context the system operates in, BLEMAT first needs scanners to be deployed. It is important to distinguish between every scanner, as each has its characteristics – every $Si$ is characterized by its position in the observed space matrix, hardware, etc. So, at first, the only two things that are known about the observed space are the maximum width and length, and the position of deployed scanners. This is necessary to be prepared as the first step in context-building for floor plan layout detection.

Figure 6.26 shows how each element of the matrix maps to an element in the grid representation of the physical space. Every element of the matrix represents a 1m$^2$ area of the observed physical space – this means that element $(0, 0)$ corresponds to the area of the space that represents a square meter around that element. From Figure 6.25 it is clear that the position of scanner $S1$ is inside element $M[0, 0]$, meaning that, on the floor plan its position is inside a square defined with four edges: $(0, 0)$, $(0, 1)$, $(1, 0)$ and $(1, 1)$ (also visualized in Figure 6.26).

Figure 6.26 shows the fusion of the matrix representation and a 2-D floor plan representation. In a 3-D IPS, there would be a 2-D floor plan model per building floor. The floor plan representation is further used for visualization purposes and

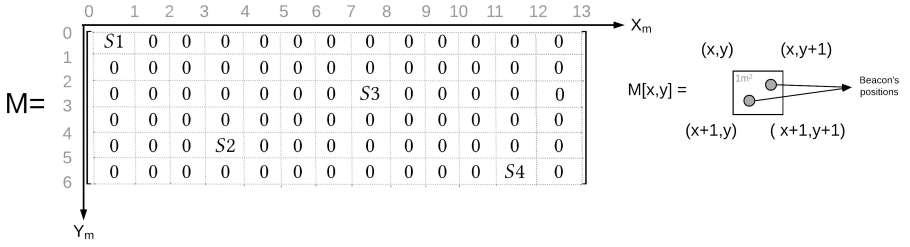



**Figure 6.26:** Matrix and floor plan models intersection.

$$M = \begin{bmatrix} S1 & 11 & 12 & 18 & 5 & 0 & 9 & 14 & 3 & 0 & 3 & 0 & 1 \\ 6 & 9 & 14 & 16 & 12 & 0 & 15 & 11 & 654 & 1312 & 1301 & 1244 & 1560 \\ 6 & 23 & 17 & 16 & 18 & 4 & 5 & 5 & 224 & 1223 & 1020 & 1311 & 1566 \\ 3 & 2 & 16 & 21 & 18 & 22 & 16 & 21 & 268 & 1100 & 3121 & 1678 & 1098 \\ 17 & 35 & 15 & S2 & 41 & 43 & 39 & 34 & 45 & 990 & 1234 & 1345 & 1312 \\ 20 & 29 & 27 & 22 & 34 & 22 & 43 & 28 & 50 & 345 & 1117 & S4 & 1087 \end{bmatrix}$$

**Figure 6.27:** Space matrix after $k$ periods.

showcasing the results of the floor plan layout detection algorithm. The matrix model is updated every time a beacon's position is captured in the system – if a beacon is captured at position $(1, 0)$, then the value $k$ at position $(1, 0)$ is increased by 1, indicating that this position has now been visited $k + 1$ times. An example of how this matrix looks like after a certain period of BLEMAT operation is displayed in Figure 6.27.

### 6.4.3 Algorithm Definition and Execution Results

In this section the algorithm used for floor plan layout detection will be specified and its execution results presented. Figure 6.28 shows how the floor plan matrix model/representation maps to the offices layout in the real, observed physical space (see Section 5.3.1). The proposed algorithm was tested on the matrix representation of the presented space (Figure 6.28). A range of Estimote BLE beacons was observed for two hours, where a position has been captured every 5 seconds resulting in 1440 position estimations. The beacons were moved to mimic movement of employees during a regular workday. An experiment for floor plan layout detection was run on the matrix defined in Figure 6.27. This matrix represents quantified movements of multiple beacons. The proposed algorithm for floor plan layout detection is presented in pseudo-code (Algorithm 1).

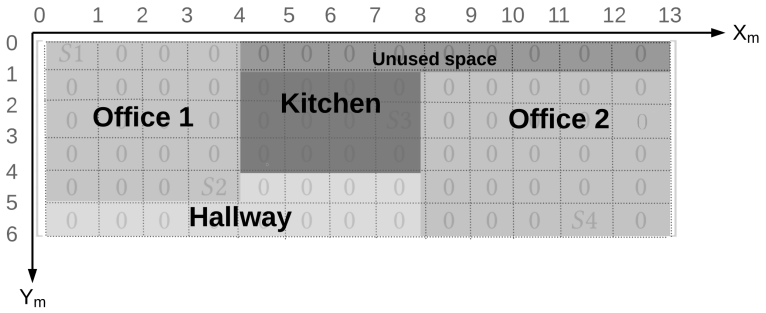



**Figure 6.28:** Deployment site 1 floor plan rooms layout.

**Algorithm 1** Floor plan layout detection algorithm.

```
1: function DETECT(M [][] )
2:     λ ← median(M)
3:     Tresholds ← [0 : 0, k : k ∗ λ, k + 1 : (k + 1) ∗ λ, ...]
4:     M_rows ← rows (M)
5:     M_columns ← cols (M)
6:     C ← M
7:     for i ← 0, M_rows do
8:         for j ← 0, M_columns do
9:             for c ∈ Tresholds do
10:                if M_[i,j] ⩾ Tresholds[k] then
11:                    C_[i,j] ← Tresholds_k
12:                end if
13:            end for
14:        end for
15:    end for
16:    return C
17: end function
```

The floor plan layout detection algorithm receives $M$ as a parameter. $M$ represents a matrix of the observed space (see Figure 6.27). The algorithm then calculates the mean of the data from $M$ and outputs a heat map of the observed space corresponding to matrix $M$. Figure 6.29 shows the heat map of the office layout where the median of empirical results was used as the only threshold, thus all elements which are larger than the median (for $M$ the median was 22) were saturated on the heat map. The heat map is a valid representation of the actual physical office space layout where all $1m^2$ areas shown in white and light gray color are indeed detected as walls and occupied space in the office. Besides obtaining empirical results 50 simulations for the same area were run. The value for each element was randomized between 0–1000. Values of elements representing walls were randomized between 0–10 representing low-frequency areas. For both empirical and simulation results it was shown that the best strategy for identifying the relevant threshold for floor plan layout detection was using the median of the observed results.

## 6.5 Graph-based Modeling of Social Networks and Relationships

From the point of their formation, through their consistency and evolution over time, social relationships can be observed and analyzed. The generation and interpretation of behavior graphs were discussed in section 6.1 and in this Section, the

emphasis is on extracting information on the consistency (continuity) of observed relationships and exploring the creation and evolution of social communities within the observed residential building. The experiments were conducted in 2019 in the months of June, November, and December.

In the case of the observed residential building (see Section 5.3.2), it is important to note that multiple tenants are living in some apartments. The number of tenants can be calculated as an educated estimate based on the number of devices active in the apartment area, but it can not be a deterministic calculation because a tenant can own multiple devices. By 2020, studies suggest that there will be 6.58 network-connected devices per person around the globe [187]. Instead, social relationships are observed at the apartment level – if two apartment vertices are connected in a tenant behavior graph, this indicates that the tenants in these apartments are interacting socially. Thus, for this deployment site, BLEMAT allows social relationship graphs to be generated, analyzed, and observed, where the social relationship is not formulated as 1:1 between two tenants, but rather as 1:1 between two apartments (and, by extension, their tenants).

### 6.5.1 Consistency of Social Relationships

Tenant behavior graphs may be extracted over a certain period. The vertices in this graph represent the visited apartments, and the edges represent the number of

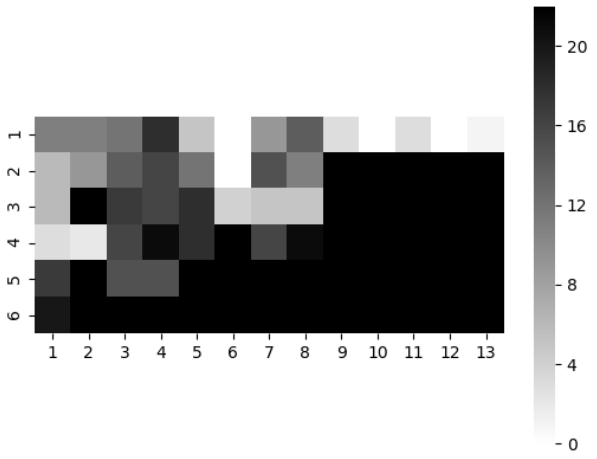



**Figure 6.29:** Deployment site 1 heat map based on matrix M (corresponds to Figure 6.28)

.

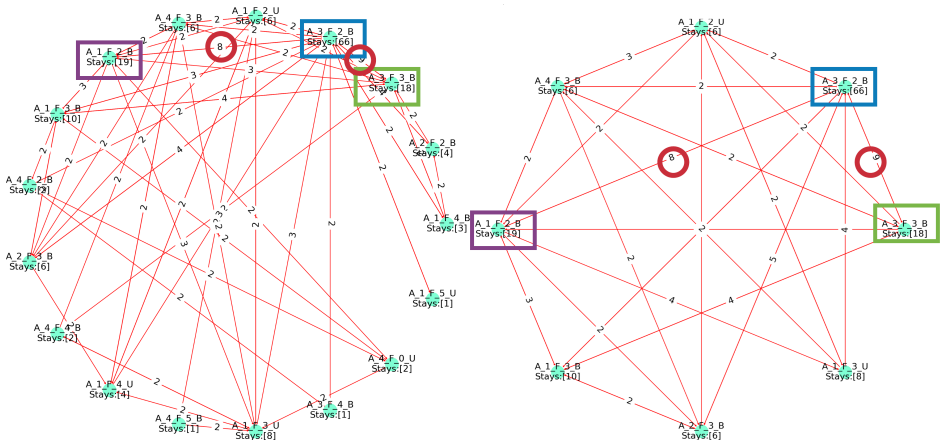


**Figure 6.30:** Two weighted tenant behavior graphs (left-full, right-reduced).

transitions made between the two apartments. The weight of the edge infers the frequency of interaction between the tenants of the apartment. Figure 6.30 shows two such graphs: on the left side, there is a full graph with all the interactions of the tenant/beacon being observed, while on the right side, there is the same graph but with the low-frequency interactions removed from the graph. The *Stays* attribute for each vertex indicates how many hours the tenant has spent in a certain apartment. The behavior graphs in Figure 6.30 correspond to the path graphs in Figure 6.3.

Tenant behavior graphs can be used to infer social behavior, frequency, consistency, and evolution of social interactions, and detection of social communities and their observation – this will be showcased in further data analysis experiments in the rest of the section.

Consistency of social relationships is confirmed by analyzing graph similarity between tenant behavior graphs for a given period. Two graph similarity metrics [93] are applied: GED and Eigenvector similarity. GED is a graph similarity measure analogous to Levenshtein distance for strings. It is defined as the minimum cost of edit path (sequence of node and edge edit operations) transforming graph $G_1$ to a graph isomorphic to $G_2$. Eigenvector similarity is explained in detail in the cited research [93, 199].

Tenant behavior graphs average similarity was extracted in the same manner as in Eq. 6.1, using the two metrics defined above, for every week of November (see Figure 6.31).

Average Eigenvector and GED show high similarities between daily tenant behavior graphs for a week, over 1 month. This indicates the consistency of social interactions over 1 month. To emphasize that the resulting Eigenvector similarity is small and implies a high similarity, graphs that are visually entirely different were

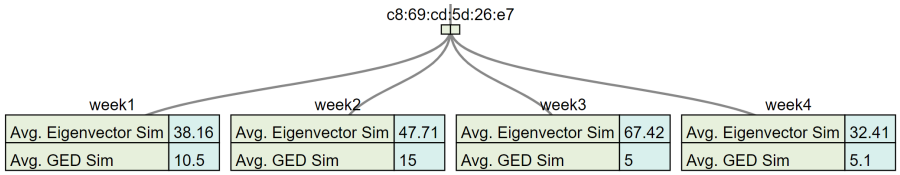



**Figure 6.31:** Tenant behavior graph weekly similarity over 1 month.

examined for comparison – the Eigenvector similarity is especially high in such cases ($>= 500$).

To further investigate the consistency of the tenant behavior, an in-depth analysis of the social interactions is performed observing the similarity of weekly tenant behavior graphs over the same month. The decision if a certain social interaction exists in the first place must be based on periodic observations and continuous repetition of analysis. Thus, BLEMAT introduces a workflow that will output a series of tenant behavior graphs for a certain apartment and its occupants to confirm the consistency of the identified social relationships. Such a series of graphs is depicted in Figure 6.32 and corresponds to the same tenant as in Figure 6.31. Figure 6.32 shows the consistency of social interactions between tenants of apartments *A_1_F_5_U* (marked orange), *A_4_F_2_B* (marked blue) and *A_1_F_0_B*

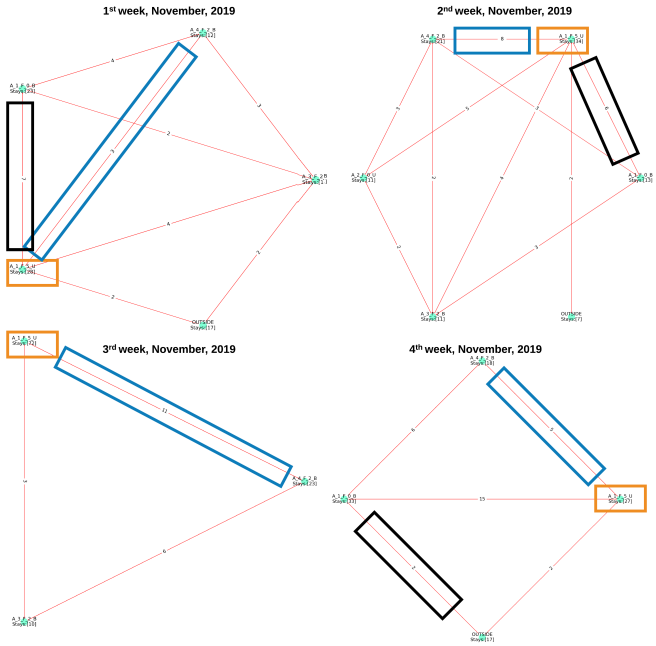



**Figure 6.32:** Consistency of social interactions over 1 month.

(marked black). From this series of behavior graphs, it is clear that the relationship between the tenants of the three apartments is consistent over 1 month.

When the consistency of social interactions is explored for the entire building, 42% of apartments (and by extension their tenants) are found to have consistent social interactions in June, 66% in November, and 63% in December (2019). This is also illustrated in Figure 6.7.

### 6.5.2 Detection of Social Communities

Community detection consists of a set of algorithms designed to identify highly interconnected groups of objects within the social network observed. These groups are called social communities. The motivation behind the discovery of the communities is diverse – in marketing, it can help a company understand the different groups of opinions about its offerings, it can help build and analyze scientific collaboration networks, etc. In the observed residential building, the objective of this thesis is to examine the current social structures of highly connected apartments (and, by extension, their tenants). This information is forwarded to BMS and can be used for tailored social group deals (meal vouchers, gym memberships, etc.) in the residential complex.

To observe social communities in a building first it is necessary to create a tenant behavior graph at the building granularity level. This graph will include a subset of detected relationships over 1 month – passive (low interactions) nodes are omitted. Such a graph is displayed in Figure 6.33 for December, and it is created for June and November as well (2019).

Algorithms for detecting social communities are run on this graph structure (Figure 6.33). Two distinctive approaches are used: Girvan-Newman (GN) algorithm and the Louvain method (LM). GN progressively removes the *most valuable* edge, i.e. the edge with the highest betweenness centrality, at each step. The community structure is exposed as the graph dissolves into smaller graphs [55]. The LM maximizes a modularity score for communities which quantifies the quality of an allocation of nodes to communities. This means evaluating how much more densely connected the nodes within a community are compared to how connected they would be in a suitably defined random network (on average) [8].

For better visualization the apartment labels were changed from strings to integers: *A_0_F_0_U = 1, A_1_F_0_U = 2*, etc. Both approaches were used on aggregated Tenant behavior graphs from June, November, and December (2019). The LM-generated communities are visible in Figure 6.34, and are labeled $J_k$, $N_k$ and $D_k$ for June, November, and December. Communities generated by the GN algorithm will be displayed in section 6.5.3, where the evolution of communities is discussed.

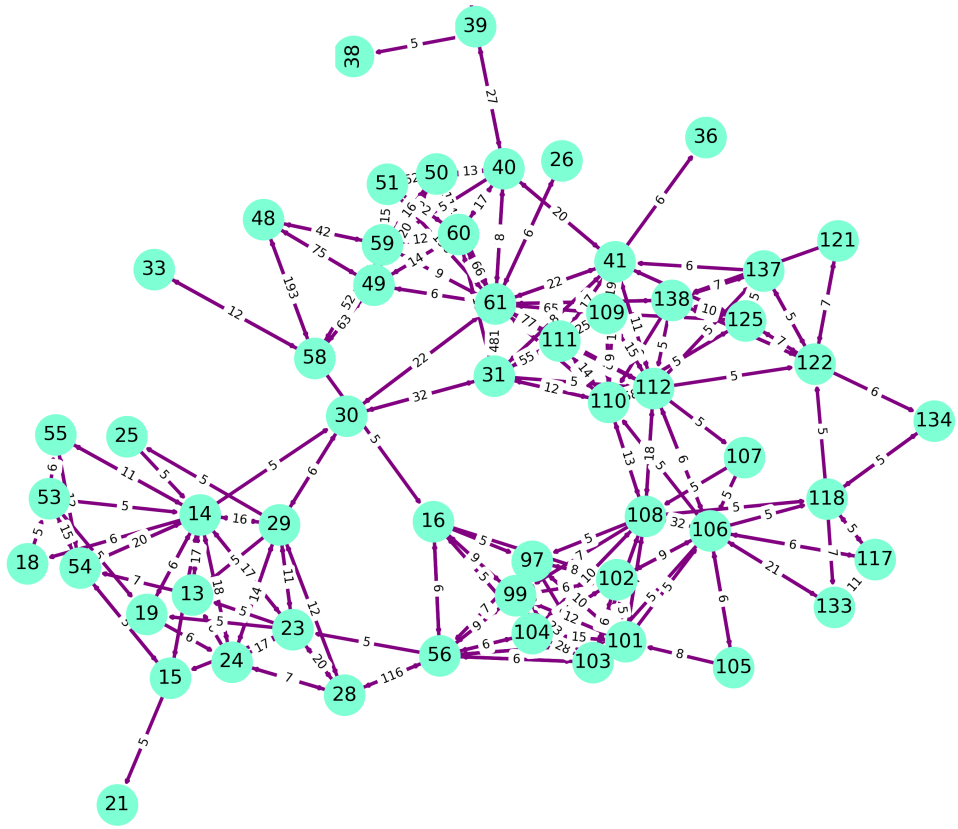


**Figure 6.33:** Social relationships in December (2019).

Running both approaches is compared in Table 6.3. Considering this is a residential building, the communities might seem large, however, that is since social relationships are observed from an apartment's perspective, and an apartment can have multiple tenants. LM has a higher modularity score for all 3 months which indicates better partitioning of nodes into communities, while GM tends to find more smaller, less dense communities. Also, LM runs significantly faster than GN – 52 times on average per graph.

| Month | GM modularity | GM communities | LM modularity | LM communities |
|---|---|---|---|---|
| June | 0.48 | 5 | 0.51 | 6 |
| November | 0.58 | 7 | 0.62 | 4 |
| December | 0.54 | 6 | 0.61 | 4 |

**Table 6.3:** Comparison of Girvan-Newman and Louvain community detection methods.

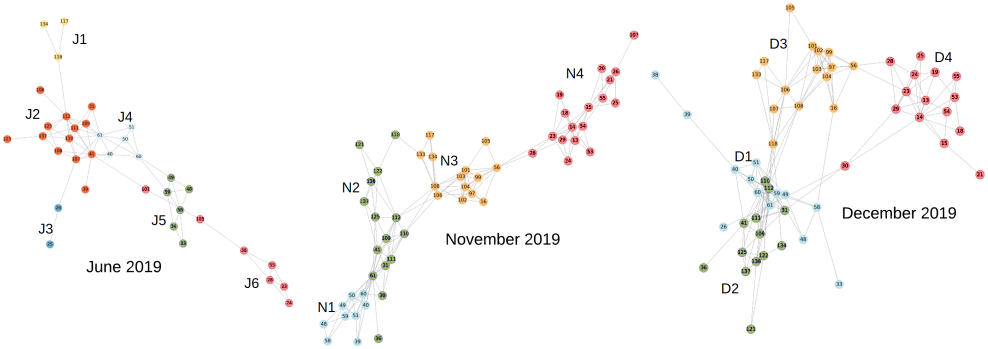



**Figure 6.34:** Social communities detected (Louvain) in June, November and December (2019).

Alongside similar community graphs, running the Girvan-Newman algorithm also outputs a dendrogram (see Figure 6.35) where each level is a partition of the graph's nodes. Communities are squared in different colors. Level 0 is the first partition, which contains the smallest communities, and the best is $len(dendrogram)-$ $-1$: the higher the level, the bigger the communities. This visual representation explains how the algorithm constructed the communities, however, detected communities can be inferred – 5 communities were detected. Looking at Figure 6.34, GN found one community less than LM since *J3-LM* joined *J2-LM*, and became the *J2-GN* community. *J5-LM* and *J4-GN*, *J1-LM* and *J1-GN*, as well as *J6-LM* and *J5-GN* overlap perfectly.

### 6.5.3 Evolution of Social Communities

Observing the life-cycle i.e evolution of a community is relevant for detecting evolution-triggering phenomena – events that cause changes in the structure of the social community networks. These events can be internal (i.e. common friend of persons A and B deactivates social media account) or external (i.e. social networking events existence). By detecting times when the observed social networks have changed, one can try tying certain events to them. In this section, only the detection of changes is explored, not the phenomena leading to them.

Tracking the evolution of social communities is not a trivial task. Challenges occur when communities naturally don't have associated identification labels (IDs). IDs may be missing because the phenomena are natural, as social relationships are. Running an algorithm for social network detection multiple times will result in different IDs for the same communities each time. One approach to solving the issue in community labeling is to assign a new ID to every detected community at each timestamp. This approach results in a large number of generated IDs while having no clear advantages for tracking communities' evolution. Additionally, there is a

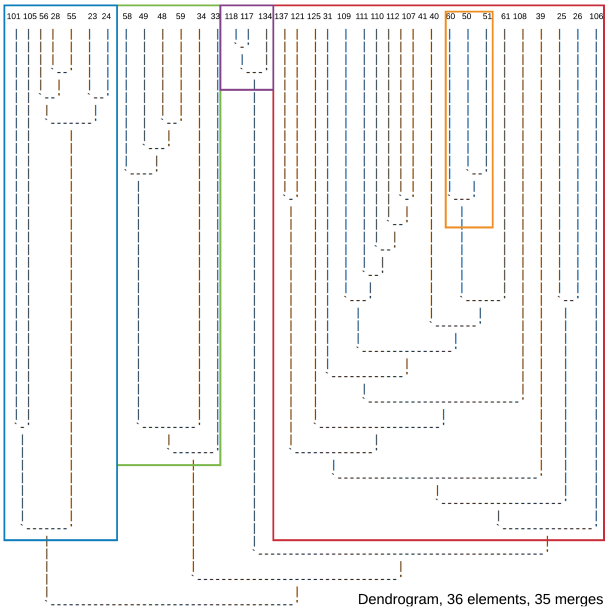



**Figure 6.35:** Dendrogram for detected communities in June (2019).

need for elegantly recording merging or splitting of communities, as those events should be included in the evolutionary history of communities. Our approach to tackling the labeling issue includes forming IDs of social communities around the vertex with the highest degree inside the community. At each time step $T_k$ *stable* vertices propagate their ID to newly joined vertices. By taking *stable* nodes a medium-to-high degree of persistence for the assignment of IDs is ensured. At this point, ID assignment for merging and dissolving communities will not be addressed and will be a part of future work on this approach.

Communities generated when applying the GN algorithm are displayed in Figure 6.36. Communities are again labeled $J_k$, $N_k$ and $D_k$ for June, November and December. Evolution of two communities (*J1* and *J5*) with highlighted most stable nodes (*J1-117, J5-24*) is visualized with a dashed line: the respective evolution paths are *J1-N4-D4* and *J5-N7-D6*. *J1* had 3 nodes, *N4* had 5 (2 joined from *J2*), and *D4* had 7 (2 more joined: *N5-105* and *N6-107*).

The evolution of all communities is displayed in Table 6.4 for the GN algorithm and in Table 6.5 for the LM algorithm. For June the table shows 5 communities and the initial number of nodes in each community. For next months, *Community* infers the label of the new community that the previous has evolved into, *Nodes* is the current number of nodes inside, *In* represents the number of nodes that have joined the community (not including nodes that were already there from previous observation steps), and *Out* represents nodes that left the community since the previous observation step. Where community is labeled as *Merged with* $I_k$ that

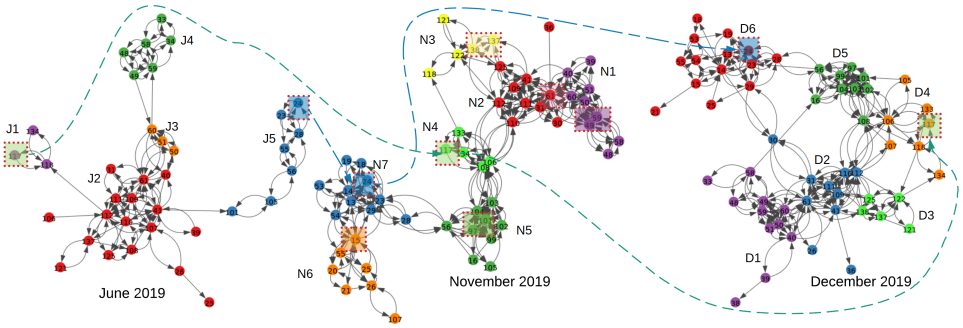



**Figure 6.36:** Social communities detected (GN) in June, November and December (2019).

infers that the community from the previous observation step has dissolved and it's nodes merged to the new community $I_k$. *Stable node* infers the most stable node of the community (refers to the first time community was detected), according to the approach of community labeling described above.

Using the information shown in these tables, respectively for GN and LM one can conclude the consistency of the evolution of social communities – if between two observation steps *In* and *Out* events are minimal, the consistency is high. In general less variation is observed when comparing November to December than June to November, due to the sheer physical time difference from one to another observation step (1 month instead of 4). Also, since the majority of tenants are students of the nearby university, a certain percent of the tenant's change during the summer break, and that is not the situation in November. Observing the evolution of communities for closer observation steps i.e. 1 month (November to December) represents a better outlook on the actual situation. Averages of *In* and *Out* events are smaller between them than June and November, indicating higher consistency. Additionally, the actual number of *In* and *Out* events is quite small per community – looking at December it varies from 1–4 for *In* and 0–1 for *Out*, for both GN and LM.

| *June* | | *November (compared to Jun)* | | | | *December (compared to Nov)* | | | | *Stable* |
|---|---|---|---|---|---|---|---|---|---|---|
| *Community* | *Nodes* | *Community* | *Nodes* | *In* | *Out* | *Community* | *Nodes* | *In* | *Out* | *node* |
| J1 | 3 | N4 | 5 | 2 | 1 | D4 | 7 | 2 | 1 | 117 |
| J2 | 17 | N2 | 10 | 3 | 10 | D2 | 10 | 1 | 1 | 61 |
| J3 | 3 | Merged to N1 | 9 | 2 | 2 | D1 | 11 | 2 | 0 | 51 |
| J4 | 6 | | | | | | | | | 49 |
| J5 | 7 | N7 | 10 | 7 | 4 | D6 | 14 | 4 | 0 | 24 |
| - | | N3 | 5 | - | - | D3 | 5 | 1 | 1 | 137 |
| - | | N5 | 9 | - | - | D5 | 9 | 1 | 1 | 97 |
| - | | N6 | 7 | - | - | Merged with D6 | | | | 15 |

**Table 6.4:** Evolution of communities generated by the Girvan-Newman algorithm.

| *June* | | *November (compared to Jun)* | | | | *December (compared to Nov)* | | | | *Stable node* |
|---|---|---|---|---|---|---|---|---|---|---|
| *Community* | *Nodes* | *Community* | *Nodes* | *In* | *Out* | *Community* | *Nodes* | *In* | *Out* | |
| J1 | 3 | Merged with N3 | 14 | 0 | 1 | D3 | 15 | 2 | 1 | 117 |
| J2 | 13 | N2 | 15 | 6 | 4 | D2 | 13 | 2 | 4 | 111 |
| J3 | 2 | Merged with N4 | 17 | 0 | 0 | D4 | 15 | 3 | 1 | 25 |
| J4 | 5 | Merged with N1 | 9 | 0 | 1 | D1 | 13 | 4 | 0 | 51 |
| J5 | 6 | | | 0 | 2 | | | | | 49 |
| J6 | 7 | Merged with N3 | 3 | 0 | 0 | - | - | - | - | 24 |
| | | Merged with N4 | 4 | 0 | 0 | | | | | |

**Table 6.5:** Evolution of communities generated by the Louvain method.

### 6.5.4 Discussion

The generation and observation of tenant behavior graphs offer a novel, multidisciplinary approach to the modeling of people's behavior in IPS, with an emphasis on social relations. Analysis of the formation, consistency, and end of such relationships is possible as an extension of IPSs. Different behavioral (and other) studies may use this data to look for patterns of social interaction and discover passive tenants. SBMs may use this information to promote social interaction between tenants within a residential building or urban area who share similar social features or are close. Besides, SBMs can learn about the social grouping of its residents through the analysis of social communities. Knowledge about the presence of these social groups may be used to improve intra-community connections, e.g. by providing targeted tickets or invitations to social events. If SBM is in charge of multiple residential buildings, approaches to the retrieval of social community information from this work can easily be extrapolated to multiple physical contexts, connecting them and inferring patterns of social behavior throughout them. Observing the lifecycle (evolution) of social communities and the detection of community-changing events can be mapped to real-world internal or external events. Both the detection and evolution of social communities in company buildings could be mapped to existing projects, deadlines, and regular business workflows.

Traditional algorithms for graph-similarity computations, such as the Graph-edit-distance are costly performance-wise, and slow. Modern research addresses this problem by offering multiple probabilistic, approximation approaches to calculating graph similarity, which are more suitable to be included as part of a service that needs to be offered in near-real-time. The Girvan-Newman algorithm is inefficient for the detection of social communities because it requires BFS, making the study of a large network a time-consuming task. The Louvain method is more suitable for practical, real-time services. Additional performance-widening capabilities for graph-related calculations and modeling should include aggregation of

graph information and embedding of graphs. Graph information aggregation can be used to reduce the size of graph datasets [205], reducing the time required for different processing workflows. Potentially, to prepare tenant path and behavior graphs as inputs to different AI-based models (consistency prediction of social relationships, prediction of social community merges), graph embedding can be explored [115].

# Chapter 7

# Conclusion and Future Work

This thesis included two major research tasks. The complexity of the Internet of Things (IoT) systems and the tasks they handle require changes in the way that resources are managed and services are delivered. Therefore, the concept of edge computing was introduced in the sphere of IoT systems to improve the scalability, reactivity, efficiency, and privacy of IoT systems. However, the edge computing concept alone is not enough to support the dynamism of a typical IoT system. Robust frameworks should be put in place to perform physical and logical resource management, decision-making processes failover and handover management and security, and system health management. The mixture of problems and requirements for creating an IoT platform with an embedded system for indoor positioning and object tracking created a solid ground for a serious research and engineering endeavor. Therefore, this thesis covers the creation of an advanced IoT platform for an IPS that includes the detection, positioning, and indoor tracking of objects based on the Bluetooth and Wi-Fi protocols.

*PART I* aimed to shed light on IoT and edge computing systems and accompanying computing and architectural paradigms, their definition, areas of application, and common use-cases, as well as operational, business, economical, social challenges and benefits. It illustrated modern needs and requests in building IoT systems and current State-of-The-Art (SoTA) approaches to designing them. It discussed the security and privacy topics of IoT and edge computing systems. As another major task, it encompassed research, design, and implementation of an MQTT-based Resource Management Framework for edge computing systems that handles: resource management, failover detection and handover administration, logical and physical workload balancing and protection, and monitoring of physical and logical system resources designed for a real-world IoT platform. It discussed modern requests for such frameworks, current SoTA approaches, and, finally, offered a solution in the form of a software framework, with minimal implementation and communication overhead.

Regarding the presented CAAVI-RICS model (see Section 2.4) the future work will include a case study where the model will be applied to a real-world problem and edge computing system, thus showcasing the broad capabilities of CAAVI-RICS security modeling.

As for the presented MQTT-based Resource Management Framework (RMF) for edge computing systems (see Chapter 3), future work will include increasing the sophistication of the RMF through three aspects: (1) inspect and reduce Manual Action Required (MAR) events, (2) refine the Workload Balance Algorithm and Workload Re-distribution Strategy Controller and Observer components, and (3) work to reduce RMF's average delay. A milestone in RMF implementation and a part of future work is integration of RMF with a commercial IoT platform created by VizLore LLC [198].

*PART II* will aim to elaborate on IPS, their definition, deployment types, commonly used positioning techniques, areas of application, and common use-cases, as well as operational, business, economic, social challenges, and benefits. It will specifically discuss designing IPS for the typical IoT infrastructure. It will offer insights to modern IPS requests, current SoTA in solving them, and underline original approaches from this thesis. It will elaborate on the research, design and authors' implementation of an IPS for the IoT – Bluetooth Low Energy Microlocation Asset Tracking (BLEMAT), including its software engines (collections of software components) for: indoor positioning, occupancy detection, visualization, patterns discovery and prediction, geofencing, movement patterns detection, visualization, discovery and prediction, social dynamics analysis, and indoor floor plan layout detection.

For this part the future work is divided by addressing BLEMAT software engines, as described in Chapter 6. When it comes to modeling and analysis of movement patterns represented by tenant path graphs (see Section 6.1), as part of the future work, probabilistic GED approaches will be researched and incorporated into BLEMAT for near real-time information retrieval capabilities. Other performance-increasing capabilities for graph-related calculations and modeling, such as graph information aggregation and graph embedding, will be explored.

When it comes to occupancy detection, prediction and data analytics (see Section 6.2), the possibilities of forecasting, data analytics and pattern searching on collected Wi-Fi utilization data will be examined. By constantly collecting and processing Wi-Fi usage data the usage of network resources can be tracked in real-time, but also forecasted to inspect utilization trends, and automatically adapt SBM strategies to trends and patterns. Lastly, lightweight pipelines for the aggregated fusion of Bluetooth positioning and Wi-Fi utilization data will be analyzed. By creating aggregated datasets for analytical tasks in BLEMAT, the techniques for occupancy detection, forecasting and pattern searching mentioned in this thesis will be inherently bettered.

To show that the conclusions from this thesis can be extrapolated to larger, and more populated indoor environments, as part of the future work GEMAT's (see Section 6.3) capabilities in large-scale deployments will be explored and experiments will be performed with longer emulation periods. Next, although support for temporal geofences and geofence relationships definition is enabled in GEMAT, a deeper theoretical model with a domain-specific language for defining sophisticated geofences and translating their specification to a programming language of choice (to be incorporated as geofencing rules) will be explored. Lastly, future work will include enhancing the current capabilities of the visualized specification of temporal geofences and geofence relationships.

When it comes to floor plan layout detection (see Section 6.4) the future work will include modeling, design and implementation of an obstacle detection framework that rests on both signal perturbation and floor plan layout detection algorithms.

Finally, as part of the future work in graph-based modeling of social networks and relationships (see Section 6.5) enhancements to the social communities' evolution tracking approach are going to be addressed, mainly community ID propagation for merging and dissolving communities, as most important events in a community's life cycle. Relevant AI-based approaches for social relationships consistency prediction and social community merges/dissolution prediction will be explored, together with graph preprocessing and transformations.

# Bibliography


[1] Aircrack-NG. Openwips-ng. `https://github.com/aircrack-ng/OpenWIPS-ng`, 2018.

[2] A. Al-Fuqaha, A. Khreishah, M. Guizani, A. Rayes, and M. Mohammadi. Toward better horizontal integration among iot services. *IEEE Communications Magazine*, 53(9):72–79, 2015.

[3] S. A. AlQahtani. Analysis and modelling of power consumption-aware priority-based scheduling for m2m data aggregation over long-term-evolution networks. *IET Communications*, 11(2):177–184, 2017.

[4] A. Alreshidi and A. Ahmad. Architecting software for the internet of thing based systems. *Future Internet*, 11(7):153, 2019.

[5] M. P. Andersen, J. Kolb, K. Chen, G. Fierro, D. E. Culler, and R. A. Popa. Wave: A decentralized authorization system for iot via blockchain smart contracts. *EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2017-234*, 2017.

[6] S. Babar, P. Mahalle, A. Stango, N. Prasad, and R. Prasad. Proposed security model and threat taxonomy for the Internet of Things (iot). In *International Conference on Network Security and Applications*, pages 420–429. Springer, 2010.

[7] A. Barki, A. Bouabdallah, S. Gharout, and J. Traore. M2m security: Challenges and solutions. *IEEE Communications Surveys & Tutorials*, 18(2):1241–1254, 2016.

[8] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[9] D. Burks. Security onion. `https://securityonionsolutions.com/`, 2012.

[10] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic. Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation computer systems*, 25(6):599–616, 2009.

[11] A. Capozzoli, M. S. Piscitelli, A. Gorrino, I. Ballarini, and V. Corrado. Data analytics for occupancy pattern learning to reduce the energy consumption of hvac systems in office buildings. *Sustainable cities and society*, 35:191–208, 2017.

[12] G. Cardone, A. Cirri, A. Corradi, L. Foschini, R. Ianniello, and R. Montanari. Crowdsensing in urban areas for city-scale mass gathering management: Geofencing and activity recognition. *IEEE Sensors Journal*, 14(12):4185–4195, 2014.

[13] G. Cardone, A. Cirri, A. Corradi, L. Foschini, R. Ianniello, and R. Montanari. Crowdsensing in urban areas for city-scale mass gathering management: Geofencing and activity recognition. *IEEE Sensors Journal*, 14(12):4185–4195, 2014.

[14] E. Carter and J. Hogue. *Intrusion prevention fundamentals*. Pearson Education India, 2006.

[15] P. Cerwall, P. Jonsson, R. Möller, S. Bävertoft, S. Carson, and I. Godor. Ericsson mobility report. *On the Pulse of the Networked Society. Hg. v. Ericsson*, 2015.

[16] W.-k. Chang and T. Hong. Statistical analysis and modeling of occupancy patterns in open-plan offices using measured lighting-switch data. In *Building Simulation*, volume 6, pages 23–32. Springer, 2013.

[17] J.-J. Chen, J.-M. Liang, and Z.-Y. Chen. Energy-efficient uplink radio resource management in lte-advanced relay networks for Internet of Things. In *2014 international wireless communications and mobile computing conference (IWCMC)*, pages 745–750. IEEE, 2014.

[18] L. Chen, S. Thombre, K. Järvinen, E. S. Lohan, A. Alén-Savikko, H. Leppäkoski, M. Z. H. Bhuiyan, S. Bu-Pasha, G. N. Ferrara, S. Honkala, et al. Robustness, security and privacy in location-based services for future iot: A survey. *IEEE Access*, 5:8956–8977, 2017.

[19] Z. Chen, Y. Chen, X. Gao, S. Wang, L. Hu, C. C. Yan, N. D. Lane, and C. Miao. Unobtrusive sensing incremental social contexts using fuzzy class incremental learning. In *Data Mining (ICDM), 2015 IEEE International Conference on*, pages 71–80. IEEE, 2015.

[20] Z. Chen et al. Bayesian filtering: From kalman filters to particle filters, and beyond. *Statistics*, 182(1):1–69, 2003.

[21] Z. Chen and Y. C. Soh. Comparing occupancy models and data mining approaches for regular occupancy prediction in commercial buildings. *Journal of Building Performance Simulation*, 10(5-6):545–553, 2017.

[22] G. Cheng, V. Peddinti, D. Povey, V. Manohar, S. Khudanpur, and Y. Yan. An exploration of dropout with lstms. In *Interspeech*, pages 1586–1590, 2017.

[23] M. Chiang and T. Zhang. Fog and IoT: An overview of research opportunities. *IEEE Internet of Things Journal*, 3(6):854–864, 2016.

[24] C. Chiţu, G. Stamatescu, I. Stamatescu, and V. Sgârciu. Wireless system for occupancy modelling and prediction in smart buildings. In *2017 25th Mediterranean Conference on Control and Automation (MED)*, pages 1094–1099. IEEE, 2017.

[25] Y.-K. Choi, S. Cho, S. Park, Y.-H. Eom, I. Kim, and B. Jeon. An extended three-dimensional geofence platform with rule-based context-awareness service for the internet of things. *Journal of Engineering Technology*, 6(1):318–328, 2018.

[26] S. Cirani, G. Ferrari, and L. Veltri. Enforcing security mechanisms in the ip-based internet of things: An algorithmic overview. *Algorithms*, 6(2):197–226, 2013.

[27] V. Cisco. Cisco visual networking index: Forecast and trends, 2017–2022. *White Paper*, 1, 2018.

[28] P. M. Comar, L. Liu, S. Saha, P.-N. Tan, and A. Nucci. Combining supervised and unsupervised learning for zero-day malware detection. In *2013 Proceedings IEEE INFOCOM*, pages 2022–2030. IEEE, 2013.

[29] S. Crook and R. Paquin. Worldwide industrial IoT platforms in energy 2019 vendor assessment. `https://www.idc.com/getdoc.jsp?containerId=US45116919`, 2019.

[30] S. Crook and R. Paquin. Worldwide industrial IoT platforms in manufacturing 2019 vendor assessment. `https://www.idc.com/getdoc.jsp?containerId=US45116819`, 2019.

[31] M. Crosby, P. Pattanayak, S. Verma, V. Kalyanaraman, et al. Blockchain technology: Beyond bitcoin. *Applied Innovation*, 2(6-10):71, 2016.

[32] David Evans. The agenda: Internet of things. https://www.politico.com/agenda/story/2015/06/internet-of-things-growth-challenges-000098, 2015.

[33] D. L. Davis. Secure boot, Aug. 1999. US Patent 5937063.

[34] G. de Blasio, A. Quesada-Arencibia, C. R. García, J. M. Molina-Gil, and C. Caballero-Gil. Study on an indoor positioning system for harsh environments based on Wi-Fi and bluetooth low energy. *Sensors*, 17(6):1299, 2017.

[35] T. Dillon, C. Wu, and E. Chang. Cloud computing: issues and challenges. In *2010 24th IEEE international conference on advanced information networking and applications*, pages 27–33. Ieee, 2010.

[36] K. Dolui and S. K. Datta. Comparison of edge computing implementations: Fog computing, cloudlet and mobile edge computing. In *2017 Global Internet of Things Summit (GIoTS)*, pages 1–6. IEEE, 2017.

[37] B. Dong and K. P. Lam. A real-time model predictive control for building heating and cooling systems based on the occupancy behavior pattern detection and local weather forecasting. In *Building Simulation*, volume 7, pages 89–106. Springer, 2014.

[38] J. Dong, C. Winstead, J. Nutaro, and T. Kuruganti. Occupancy-based HVAC control with short-term occupancy prediction algorithms for energy-efficient buildings. *Energies*, 11(9):2427, 2018.

[39] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram. Blockchain for IoT security and privacy: The case study of a smart home. In *2017 IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 618–623. IEEE, 2017.

[40] T. Ekwevugbe, N. Brown, V. Pakka, and D. Fan. Real-time building occupancy sensing using neural-network based sensor network. In *2013 7th IEEE International Conference on Digital Ecosystems and Technologies (DEST)*, pages 114–119. IEEE, 2013.

[41] E. Elnahrawy, X. Li, and R. P. Martin. The limits of localization using signal strength: A comparative study. In *IEEE Sensor and Ad Hoc Communications and Networks*, pages 406–414. IEEE, 2004.

[42] P. Esling and C. Agon. Time-series data mining. *ACM Computing Surveys (CSUR)*, 45(1):12, 2012.

[43] R. Faragher and R. Harle. An analysis of the accuracy of Bluetooth Low Energy for indoor positioning applications. In *Proceedings of the 27th International Technical Meeting of the Satellite Division of the Institute of Navigation*, pages 201–210. Institute of Navigation, 2014.

[44] R. Faragher and R. Harle. Location fingerprinting with Bluetooth Low Energy beacons. *IEEE journal on Selected Areas in Communications*, 33(11):2418–2428, 2015.

[45] E. C. Ferrer. The blockchain: a new framework for robotic swarm systems. In *Proceedings of the Future Technologies Conference*, pages 1037–1058. Springer, 2018.

[46] A. Filippoupolitis, W. Oliff, and G. Loukas. Occupancy detection for building emergency management using BLE beacons. In *International Symposium on Computer and Information Sciences*, pages 233–240. Springer, 2016.

[47] L. Fortuna, S. Graziani, A. Rizzo, and M. G. Xibilia. *Soft sensors for monitoring and control of industrial processes*. Springer Science & Business Media, 2007.

[48] E. Furey, K. Curran, and P. Mc Kevitt. Learning indoor movement habits for predictive control. *International Journal of Space-Based and Situated Computing*, 1(4):222–232, 2011.

[49] E. Ganne. *Can Blockchain revolutionize international trade?* World Trade Organization, 2018.

[50] R. Gao, M. Zhao, T. Ye, F. Ye, Y. Wang, K. Bian, T. Wang, and X. Li. Jigsaw: Indoor floor plan reconstruction via mobile crowdsensing. In *Proceedings of the 20th annual international conference on Mobile computing and networking*, pages 249–260. ACM, 2014.

[51] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández, and E. Vázquez. Anomaly-based network intrusion detection: Techniques, systems and challenges. *Computers & Security*, 28(1-2):18–28, 2009.

[52] Z. Geler, V. Kurbalija, M. Ivanović, and M. Radovanović. Weighted kNN and constrained elastic distances for time-series classification. *Expert Systems with Applications*, page 113829, 2020.

[53] Y. Geng and C. G. Cassandras. A new smart parking system infrastructure and implementation. *Social and Behavioral Sciences*, 54:1278–1287, 2012.

[54] A. Gilchrist. *Industry 4.0: the industrial internet of things*. Apress, 2016.

[55] M. Girvan and M. E. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[56] Google. User location services, 2019.

[57] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *Proceedings of the 13th ACM Conference on Computer and Communications Security*, pages 89–98, 2006.

[58] J. L. Gross and J. Yellen. *Graph theory and its applications*. CRC press, 2005.

[59] Y. Gu, Y. Chen, J. Liu, and X. Jiang. Semi-supervised deep extreme learning machine for Wi-Fi based localization. *Neurocomputing*, 166:282–293, 2015.

[60] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami. Internet of Things (iot): A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7):1645–1660, 2013.

[61] M. Guldner, T. Spieldenner, and R. Schubotz. Nexus: Using geo-fencing services without revealing your location. In *2018 Global Internet of Things Summit (GIoTS)*, pages 1–6. IEEE, 2018.

[62] A. Gulli and S. Pal. *Deep Learning with Keras*. Packt Publishing Ltd, 2017.

[63] T. Güneysu and T. Oder. Towards lightweight identity-based encryption for the post-quantum-secure Internet of Things. In *2017 18th International Symposium on Quality Electronic Design (ISQED)*, pages 319–324. IEEE, 2017.

[64] J. Helmy and A. Helmy. The Alzimio app for dementia, autism & alzheimer's: Using novel activity recognition algorithms and geofencing. In *2016 IEEE International Conference on Smart Computing (SMARTCOMP)*, pages 1–6. IEEE, 2016.

[65] S. Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116, 1998.

[66] E. Hodo, X. Bellekens, A. Hamilton, P.-L. Dubouilh, E. Iorkyase, C. Tachtatzis, and R. Atkinson. Threat analysis of IoT networks using artificial

neural network intrusion detection system. In *2016 International Symposium on Networks, Computers and Communications (ISNCC)*, pages 1–6. IEEE, 2016.

[67] M. Hoeynck and B. W. Andrews. Sensor-based occupancy and behavior prediction method for intelligently controlling energy consumption within a building, 2010. US Patent Application 12/183361.

[68] C.-H. Hong and B. Varghese. Resource management in fog/edge computing: A survey. *arXiv preprint arXiv:1810.00305*, 2018.

[69] X. Huang, Y. Xiang, A. Chonka, J. Zhou, and R. H. Deng. A generic framework for three-factor authentication: Preserving security and privacy in distributed systems. *IEEE Transactions on Parallel and Distributed Systems*, 22(8):1390–1397, 2010.

[70] R. Hummen, H. Shafagh, S. Raza, T. Voig, and K. Wehrle. Delegation-based authentication and authorization for the ip-based Internet of Things. In *2014 Eleventh Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pages 284–292. Ieee, 2014.

[71] M. H. Ibrahim. Octopus: An edge-fog mutual authentication scheme. *Network Security*, 18(6):1089–1101, 2016.

[72] C. G. C. Index. Forecast and methodology, 2016–2021 white paper. *Updated: February*, 1, 2018.

[73] F. S. Iyadurai and P. Subramanian. Smartphones and the disruptive innovation of the retail shopping experience. In *2016 International Conference on Disruptive Innovation*, 2016.

[74] C. S. Jensen, H. Lu, and B. Yang. Graph model based indoor tracking. In *Mobile Data Management: Systems, Services and Middleware*, pages 122–131. IEEE, 2009.

[75] X. Jiang, Y. Chen, J. Liu, Y. Gu, and L. Hu. FSELM: fusion semi-supervised extreme learning machine for indoor localization with Wi-Fi and Bluetooth fingerprints. *Soft Computing*, 22(11):3621–3635, 2018.

[76] Y. Jiang, W. Susilo, Y. Mu, and F. Guo. Ciphertext-policy attribute-based encryption against key-delegation abuse in fog computing. *Future Generation Computer Systems*, 78:720–729, 2018.

[77] Y. Jiang, Y. Xiang, X. Pan, K. Li, Q. Lv, R. P. Dick, L. Shang, and M. Hannigan. Hallway based automatic indoor floorplan construction using room

fingerprints. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 315–324. ACM, 2013.

[78] Z. Jianyong, L. Haiyong, C. Zili, and L. Zhaohui. Rssi based bluetooth low energy indoor positioning. In *Indoor Positioning and Indoor Navigation (IPIN), 2014 International Conference on*, pages 526–533. IEEE, 2014.

[79] M. Joye and G. Neven. *Identity-based cryptography*, volume 2. IOS press, 2009.

[80] R. Jozefowicz, W. Zaremba, and I. Sutskever. An empirical exploration of recurrent network architectures. In *International Conference on Machine Learning*, pages 2342–2350, 2015.

[81] V. Karagiannis, P. Chatzimisios, F. Vazquez-Gallego, and J. Alonso-Zarate. A survey on application layer protocols for the internet of things. *Transaction on IoT and Cloud computing*, 3(1):11–17, 2015.

[82] W. Khalid, Z. Ullah, N. Ahmed, Y. Cao, M. Khalid, M. Arshad, F. Ahmad, and H. Cruickshank. A taxonomy on misbehaving nodes in delay tolerant networks. *Computers & Security*, 77:442–471, 2018.

[83] S. Khan, S. Parkinson, and Y. Qin. Fog computing security: a review of current applications and security solutions. *Journal of Cloud Computing*, 6(1):19, 2017.

[84] K. N. Khaqqi, J. J. Sikorski, K. Hadinoto, and M. Kraft. Incorporating seller/buyer reputation-based system in blockchain-enabled emission trading application. *Applied Energy*, 209:8–19, 2018.

[85] H. Kim, E. Kang, E. A. Lee, and D. Broman. A toolkit for construction of authorization service infrastructure for the internet of things. In *Proceedings of the Second International Conference on Internet-of-Things Design and Implementation*, pages 147–158, 2017.

[86] H.-W. Kim, J. H. Park, and Y.-S. Jeong. Efficient resource management scheme for storage processing in cloud infrastructure with internet of things. *Wireless Personal Communications*, 91(4):1635–1651, 2016.

[87] S.-C. Kim, Y.-S. Jeong, and S.-O. Park. RFID-based indoor location tracking to ensure the safety of the elderly in smart home environments. *Personal and Ubiquitous Computing*, 17(8):1699–1707, 2013.

[88] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[89] M. B. Kjærgaard, M. Wirz, D. Roggen, and G. Tröster. Mobile sensing of pedestrian flocks in indoor environments using wifi signals. In *2012 IEEE International Conference on Pervasive Computing and Communications*, pages 95–102. IEEE, 2012.

[90] P. C. Kocher, P. Rohatgi, and J. M. Jaffe. Secure boot with resistance to differential power analysis and other external monitoring attacks, Feb. 2017. US Patent 9569623.

[91] J. Kolodziej, S. U. Khan, L. Wang, N. Min-Allah, S. A. Madani, N. Ghani, and H. Li. An application of markov jump process model for activity-based indoor mobility prediction in wireless networks. In *2011 Frontiers of Information Technology*, pages 51–56. IEEE, 2011.

[92] K. Korpela, J. Hallikas, and T. Dahlberg. Digital supply chain transformation toward blockchain integration. In *proceedings of the 50th Hawaii international conference on system sciences*, 2017.

[93] D. Koutra, A. Parikh, A. Ramdas, and J. Xiang. Algorithms for graph similarity and subgraph matching. In *Proc. Ecol. Inference Conf*, volume 17, 2011.

[94] D. Kozlov, J. Veijalainen, and Y. Ali. Security and privacy threats in iot architectures. In *Proceedings of the 7th International Conference on Body Area Networks*, pages 256–262, 2012.

[95] P. Kriz, F. Maly, and T. Kozel. Improving indoor localization using bluetooth low energy beacons. *Mobile Information Systems*, 2016, 2016.

[96] P. Kumar, N. Zaidi, and T. Choudhury. Fog computing: Common security issues and proposed countermeasures. In *2016 International Conference System Modeling & Advancement in Research Trends (SMART)*, pages 311–315. IEEE, 2016.

[97] V. Kurbalija, M. Radovanović, Z. Geler, and M. Ivanović. The influence of global constraints on similarity measures for time-series databases. *Knowledge-Based Systems*, 56:49–67, 2014.

[98] D. Kyriazis, T. Varvarigou, D. White, A. Rossi, and J. Cooper. Sustainable smart city iot applications: Heat and electricity management & eco-conscious cruise control for public transportation. In *2013 IEEE 14th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–5. IEEE, 2013.

[99] S. M. Law. Stumpy. `https://github.com/TDAmeritrade/stumpy`, July 2019.

[100] Y. W. Law, M. Palaniswami, G. Kounga, and A. Lo. Wake: Key management scheme for wide-area measurement systems in smart grid. *IEEE Communications Magazine*, 51(1):34–41, 2013.

[101] O. Leiba, Y. Yitzchak, R. Bitton, A. Nadler, and A. Shabtai. Incentivized delivery network of iot software updates based on trustless proof-of-distribution. In *2018 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 29–39. IEEE, 2018.

[102] J. Leveling, M. Edelbrock, and B. Otto. Big data analytics for supply chain management. In *2014 IEEE International Conference on Industrial Engineering and Engineering Management*, pages 918–922. IEEE, 2014.

[103] H. Li. Low-cost 3d Bluetooth indoor positioning with least square. *Wireless Personal Communications*, 78(2):1331–1344, 2014.

[104] N. Li, G. Calis, and B. Becerik-Gerber. Measuring and monitoring occupancy with an rfid based system for demand-driven hvac operations. *Automation in construction*, 24:89–99, 2012.

[105] S. Li, N. Zhang, S. Lin, L. Kong, A. Katangur, M. K. Khan, M. Ni, and G. Zhu. Joint admission control and resource allocation in edge computing for internet of things. *IEEE Network*, 32(1):72–79, 2018.

[106] X. Li, J. Wang, and C. Liu. A Bluetooth/PDR integration algorithm for an indoor positioning system. *Sensors*, 15(10):24862–24885, 2015.

[107] Z. Li and B. Dong. A new modeling approach for short-term prediction of occupancy in residential buildings. *Building and Environment*, 121:277–290, 2017.

[108] Z. Li and B. Dong. Short term predictions of occupancy in commercial buildings: performance analysis for stochastic models and machine learning approaches. *Energy and Buildings*, 158:268–281, 2018.

[109] Z. Li, J. Kang, R. Yu, D. Ye, Q. Deng, and Y. Zhang. Consortium blockchain for secure energy trading in industrial internet of things. *IEEE Transactions on Industrial Informatics*, 14(8):3690–3700, 2017.

[110] X. Liang, T. Hong, and G. Q. Shen. Occupancy data analytics and prediction: A case study. *Building and Environment*, 102:179–192, 2016.

[111] R. A. Light et al. Mosquitto: server and client implementation of the MQTT protocol. *J. Open Source Software*, 2(13):265, 2017.

[112] B. Liu, X. L. Yu, S. Chen, X. Xu, and L. Zhu. Blockchain based data integrity service framework for IoT data. In *2017 IEEE International Conference on Web Services (ICWS)*, pages 468–475. IEEE, 2017.

[113] H. Liu, H. Darabi, P. Banerjee, and J. Liu. Survey of wireless indoor positioning techniques and systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(6):1067–1080, 2007.

[114] S. Liu, L. Yin, W. K. Ho, K. V. Ling, and S. Schiavon. A tracking cooling fan using geofence and camera-based indoor localization. *Building and Environment*, 114:36–44, 2017.

[115] X. Liu, C. Zhuang, T. Murata, K.-S. Kim, and N. Kertkeidkachorn. How much topological structure is preserved by graph embeddings? *Computer Science and Information Systems*, 16(2):597–614, 2019.

[116] Y. Liu and Z. Yang. *Location, localization, and localizability: location-awareness technology for wireless networks*. Springer Science & Business Media, 2010.

[117] J. Lu, T. Sookoor, V. Srinivasan, G. Gao, B. Holben, J. Stankovic, E. Field, and K. Whitehouse. The smart thermostat: using occupancy sensors to save energy in homes. In *Proceedings of the 8th ACM conference on embedded networked sensor systems*, pages 211–224. ACM, 2010.

[118] P. J. Ludford, D. Frankowski, K. Reily, K. Wilms, and L. Terveen. Because i carry my cell phone anyway: functional location-based reminder applications. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 889–898. ACM, 2006.

[119] M. Lujak and S. Giordani. Centrality measures for evacuation: finding agile evacuation routes. *Future Generation Computer Systems*, 83:401–412, 2018.

[120] T. Lundqvist, A. De Blanche, and H. R. H. Andersson. Thing-to-thing electricity micro payments using blockchain technology. In *2017 Global Internet of Things Summit (GIoTS)*, pages 1–6. IEEE, 2017.

[121] H. Luo, F. Zhao, M. Jiang, H. Ma, and Y. Zhang. Constructing an indoor floor plan using crowdsourcing based on magnetic fingerprinting. *Sensors*, 17(11):2678, 2017.

[122] J. Macaulay and M. Kuckelhaus. Internet Of Things in logistics. https://discover.dhl.com/content/dam/dhl/downloads/interim/full/dhl-trend-report-internet-of-things.pdf, 2019.

[123] P. R. MacNeille, J. Wisniewski, and N. DeCia. Vehicle-to-vehicle cooperation to marshal traffic, 2018. US Patent 9928746.

[124] A. Mahdavi and F. Tahmasebi. Predicting people's presence in buildings: An empirically based model performance analysis. *Energy and Buildings*, 86:349–355, 2015.

[125] R. Mahmoud, T. Yousuf, F. Aloul, and I. Zualkernan. Internet of things (iot) security: Current status, challenges and prospective measures. In *2015 10th International Conference for Internet Technology and Secured Transactions (ICITST)*, pages 336–341. IEEE, 2015.

[126] S. Mamidi, Y.-H. Chang, and R. Maheswaran. Improving building energy efficiency with a network of sensing, learning and prediction agents. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 45–52. International Foundation for Autonomous Agents and Multiagent Systems, 2012.

[127] B. Manate, T.-F. Fortis, and V. Negru. Optimizing cloud resources allocation for an Internet of Things architecture. *Scalable Computing: Practice and Experience*, 15(4):345–355, 2014.

[128] A. Manzoor, A. Wahid, M. A. Shah, A. Akhunzada, and F. F. Qureshi. Secure login using multi-tier authentication schemes in fog computing. *EAI Endorsed Transactions on Internet of Things*, 3:1–6, 2018.

[129] Y. Mao, J. Zhang, S. Song, and K. B. Letaief. Stochastic joint radio and computational resource management for multi-user mobile-edge computing systems. *IEEE Transactions on Wireless Communications*, 16(9):5994–6009, 2017.

[130] D. Martin, A. Alzua, and C. Lamsfus. A contextual geofencing mobile tourism service. In *Information and Communication Technologies in Tourism*, pages 191–202. Springer, 2011.

[131] M. Matzner, F. Chasin, M. von Hoffen, F. Plenter, and J. Becker. Designing a peer-to-peer sharing service as fuel for the development of the electric vehicle charging infrastructure. In *49th Hawaii International Conference on System Sciences (HICSS)*, pages 1587–1595. IEEE, 2016.

[132] J. McGrath. Micropayments: Final frontier for electronic consumer payments. *Federal Reserve Bank of Philadelphia Payment Cards Center Discussion Paper*, 06, 2006.

[133] D. Minoli, K. Sohraby, and B. Occhiogrosso. Iot considerations, requirements, and architectures for smart buildings: Energy optimization and next-generation building management systems. *IEEE Internet of Things Journal*, 4(1):269–283, 2017.

[134] S. Misra, K. I. Abraham, M. S. Obaidat, and P. V. Krishna. Laid: a learning automata-based scheme for intrusion detection in wireless sensor networks. *Security and Communication Networks*, 2(2):105–115, 2009.

[135] T. Moore, M. Raya, J. Clulow, P. Papadimitratos, R. Anderson, and J.-P. Hubaux. Fast exclusion of errant devices from vehicular networks. In *2008 5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, pages 135–143. IEEE, 2008.

[136] S. Mubeen, P. Nikolaidis, A. Didic, H. Pei-Breivold, K. Sandström, and M. Behnam. Delay mitigation in offloaded cloud controllers in industrial iot. *IEEE Access*, 5:4418–4430, 2017.

[137] N. Naik. Choice of effective messaging protocols for iot systems: MQTT, CoAP, AMQP and HTTP. In *IEEE International Systems Engineering Symposium (ISSE)*, pages 1–7. IEEE, 2017.

[138] M. Nakagawa. Improvement in the geofencing service interface using indoor positioning systems and mobile sensors. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4:W4, 2013.

[139] D. Namiot and M. Sneps-Sneppe. Geofence and network proximity. In *Internet of Things, Smart Spaces, and Next Generation Networking*, pages 117–127. Springer, 2013.

[140] J. C. Navas and T. Imielinski. GeoCast: geographic addressing and routing. In *Proceedings of the 3rd Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pages 66–76. Citeseer, 1997.

[141] I. C. Ng. *Creating new markets in the digital economy*. Cambridge University Press, 2014.

[142] K. T. Nguyen, M. Laurent, and N. Oualha. Survey on secure communication protocols for the Internet of Things. *Ad Hoc Networks*, 32:17–31, 2015.

[143] T. A. Nguyen and M. Aiello. Energy intelligent buildings based on user activity: A survey. *Energy and buildings*, 56:244–257, 2013.

[144] R. Nogueras and C. Cotta. On the use of self island-based evolutionary computation methods on complex environments? *Computer Science & Information Systems*, 15(3), 2018.

[145] S. Noor, W. Yang, M. Guo, K. H. van Dam, and X. Wang. Energy demand side management within micro-grid networks enhanced by blockchain. *Applied Energy*, 228:1385–1398, 2018.

[146] D. J. Norris. *Machine Learning with the Raspberry Pi*. Springer, 2020.

[147] R. R. Oliveira, F. C. Noguez, C. A. Costa, J. L. Barbosa, and M. P. Prado. Swtrack: An intelligent model for cargo tracking based on off-the-shelf mobile devices. *Expert Systems with Applications*, 40(6):2023–2031, 2013.

[148] J. G. Ortega, L. Han, N. Whittacker, and N. Bowring. A machine-learning based approach to model user occupancy and activity patterns for energy saving in buildings. In *Science and Information Conference (SAI)*, pages 474–482. IEEE, 2015.

[149] R. Pass et al. Micropayments for decentralized currencies. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 207–218. ACM, 2015.

[150] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos. Context aware computing for the internet of things: A survey. *IEEE Sommunications Surveys & Tutorials*, 16(1):414–454, 2014.

[151] S. Pešić. BLEMAT positioning datasets. `https://doi.org/10.5281/zenodo.3339374`, July 2019.

[152] S. Pešić, M. Ivanović, M. Radovanović, and C. Bădică. CAAVI-RICS model for observing the security of distributed iot and edge computing systems. *Simulation Modelling Practice and Theory*, page 102125, 2020.

[153] S. Pešić, M. Radovanović, M. Ivanović, C. Badica, M. Tošić, O. Iković, and D. Bošković. CAAVI-RICS model for analyzing the security of fog computing systems. In *International Symposium on Intelligent and Distributed Computing*, pages 23–34. Springer, 2019.

[154] S. Pesic, M. Radovanović, M. Ivanović, C. Badica, M. Tošić, O. Iković, and D. Bošković. CAAVI-RICS model for analyzing the security of fog computing systems: Authentication. In *2019 20th International Conference on*

*Parallel and Distributed Computing, Applications and Technologies (PDCAT)*, pages 226–231. IEEE, 2019.

[155] S. Pešić, M. Tošić, O. Iković, M. Ivanović, M. Radovanović, and D. Bošković. Context aware resource and service provisioning management in fog computing systems. In *International Symposium on Intelligent and Distributed Computing*, pages 213–223. Springer, 2017.

[156] S. Pešić, M. Tošić, O. Iković, M. Radovanović, M. Ivanović, and D. Bošković. Bluetooth low energy microlocation asset tracking (blemat) in a context-aware fog computing system. In *Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics*, page 23. ACM, 2018.

[157] S. Pešić, M. Tošić, O. Iković, M. Radovanović, M. Ivanović, and D. Bošković. BLEMAT: Data analytics and machine learning for smart building occupancy detection and prediction. *International Journal on Artificial Intelligence Tools*, 28(06):1960005, 2019.

[158] S. Pešić, M. Tošić, O. Iković, M. Radovanović, M. Ivanović, and D. Bošković. Conceptualizing a collaboration framework between blockchain technology and the Internet of Things. In *Proceedings of the 20th International Conference on Computer Systems and Technologies (CompSysTech '19), June 21–22, 2019, Ruse, Bulgaria*. ACM, 2019.

[159] S. Pešić, M. Tošić, O. Iković, M. Radovanović, M. Ivanović, and D. Bošković. Gemat - Internet of Things solution for indoor security geofencing. In *Proceedings of the 9th Balkan Conference in Informatics (BCI 2019), September 26-28, 2019, Sofia, Bulgaria*. ACM, 2019.

[160] S. Pešić, M. Tošić, O. Iković, M. Radovanović, M. Ivanović, and D. Bošković. Hyperledger fabric blockchain as a service for the Internet of Things: Proof of concept. In *Proceedings of the 9th International Conference on Model and Data Engineering (MEDI 2019), October 28-31, 2019, Toulouse, France*. Springer, 2019.

[161] J. S. Preden, K. Tammemäe, A. Jantsch, M. Leier, A. Riid, and E. Calis. The benefits of self-awareness and attention in fog and mist computing. *Computer*, 48(7):37–45, 2015.

[162] B. Qolomany, A. Al-Fuqaha, D. Benhaddou, and A. Gupta. Role of deep lstm neural networks and Wi-Fi networks in support of occupancy prediction in smart buildings. In *2017 IEEE 19th International Conference on High Performance Computing and Communications; IEEE 15th International Conference on Smart City; IEEE 3rd International Conference on*

*Data Science and Systems (HPCC/SmartCity/DSS)*, pages 50–57. IEEE, 2017.

[163] L. Radaelli, D. Sabonis, H. Lu, and C. S. Jensen. Identifying typical movements among indoor objects–concepts and empirical study. In *2013 IEEE 14th International Conference on Mobile Data Management*, volume 1, pages 197–206. IEEE, 2013.

[164] W. Razouk, D. Sgandurra, and K. Sakurai. A new security middleware architecture based on fog computing and cloud to support IoT constrained devices. In *Proceedings of the 1st International Conference on Internet of Things and Machine Learning*, page 35. ACM, 2017.

[165] T. Renner, M. Meldau, and A. Kliem. Towards container-based resource management for the internet of things. In *2016 International Conference on Software Networking (ICSN)*, pages 1–5. IEEE, 2016.

[166] A. Reyna, C. Martín, J. Chen, E. Soler, and M. Díaz. On blockchain and its integration with iot. challenges and opportunities. *Future Generation Computer Systems*, 88:173–190, 2018.

[167] M. Rinne, S. Törmä, and D. Kratinov. Mobile crowdsensing of parking space using geofencing and activity recognition. In *10th ITS European Congress, Helsinki, Finland*, pages 16–19, 2014.

[168] S. Rodriguez Garzon and B. Deva. Geofencing 2.0: taking location-based notifications to the next level. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 921–932. ACM, 2014.

[169] M. Roesch et al. Snort: Lightweight intrusion detection for networks. In *Lisa*, volume 99, pages 229–238, 1999.

[170] K. Rosenfeld, E. Gavas, and R. Karri. Sensor physical unclonable functions. In *2010 IEEE International Symposium on Hardware-oriented Security and Trust (HOST)*, pages 112–117. IEEE, 2010.

[171] J. Roux, E. Alata, G. Auriol, V. Nicomette, and M. Kaâniche. Toward an intrusion detection approach for IoT based on radio communications profiling. In *2017 13th European Dependable Computing Conference (EDCC)*, pages 147–150. IEEE, 2017.

[172] S. H. Ryu and H. J. Moon. Development of an occupancy prediction model using indoor environmental data based on machine learning techniques. *Building and Environment*, 107:1–9, 2016.

[173] K. Sadouskaya. Adoption of blockchain technologyin supply chain and logistics, 2017. Bachelor's thesis.

[174] F. C. Sangogboye, K. Imamovic, and M. B. Kjærgaard. Improving occupancy presence prediction via multi-label classification. In *2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, pages 1–6. IEEE, 2016.

[175] L. A. Schmidt and L. Riegger. Moving geofence for machine tracking in agriculture, June 2015. US Patent 9066464.

[176] P. Schulz, M. Matthe, H. Klessig, M. Simsek, G. Fettweis, J. Ansari, S. A. Ashraf, B. Almeroth, J. Voigt, I. Riedel, et al. Latency critical iot applications in 5g: Perspective on the design of radio interface and network architecture. *IEEE Communications Magazine*, 55(2):70–78, 2017.

[177] H. Sedjelmaci, S. M. Senouci, and M. Al-Bahri. A lightweight anomaly detection technique for low-resource IoT devices: a game-theoretic methodology. In *2016 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2016.

[178] A. Sehgal, V. Perelman, S. Kuryla, and J. Schonwalder. Management of resource constrained devices in the internet of things. *IEEE Communications Magazine*, 50(12):144–149, 2012.

[179] L. Seitz, G. Selander, E. Wahlstroem, S. Erdtman, and H. Tschofenig. Authorization for the Internet of Things using oauth 2.0. *Internet Engineering Task Force (IETF): Fremont, CA, USA*, 2015.

[180] M. Senožetnik, L. Bradeško, T. Šubic, Z. Herga, J. Urbančič, P. Škraba, and D. Mladenić. Estimating point-of-interest rating based on visitors geospatial behaviour. *Computer Science and Information Systems*, 16(1):131–154, 2019.

[181] E. Shi, H. Chan, E. Rieffel, R. Chow, and D. Song. Privacy-preserving aggregation of time-series data. In *Annual Network & Distributed System Security Symposium (NDSS)*. Internet Society., 2011.

[182] H. Shin, Y. Chon, and H. Cha. Unsupervised construction of an indoor floor plan using a smartphone. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 42(6):889–898, 2012.

[183] M. Singh, M. Rajan, V. Shivraj, and P. Balamuralidhar. Secure mqtt for internet of things (iot). In *2015 Fifth International Conference on Communication Systems and Network Technologies*, pages 746–751. IEEE, 2015.

[184] P. K. Singh, R. Singh, S. K. Nandi, and S. Nandi. Smart contract based decentralized parking management in its. In *International Conference on Innovations for fCommunity Services*, pages 66–77. Springer, 2019.

[185] S. Singh, P. K. Sharma, S. Y. Moon, and J. H. Park. Advanced lightweight encryption algorithms for IoT devices: survey, challenges and solutions. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–18, 2017.

[186] M. Srivatsa, L. Liu, and A. Iyengar. Eventguard: A system architecture for securing publish-subscribe networks. *ACM Transactions on Computer Systems (TOCS)*, 29(4):1–40, 2011.

[187] Statista. Number of network connected devices per person around the world 2003-2020. `https://www.statista.com/statistics/678739/forecast-on-connected-devices-per-person/`, 2019.

[188] D. Strugar, R. Hussain, M. Mazzara, V. Rivera, J. Y. Lee, and R. Mustafin. On m2m micropayments: a case study of electric autonomous vehicles. In *2018 IEEE International Conference on Internet of Things (iThings)*, pages 1697–1700. IEEE, 2018.

[189] F. Subhan, H. Hasbullah, A. Rozyyev, and S. T. Bakhsh. Indoor positioning in bluetooth networks using fingerprinting and lateration approach. In *Information Science and Applications (ICISA*, pages 1–9. IEEE, 2011.

[190] Y. Sun, Z. Han, and K. R. Liu. Defense of trust management vulnerabilities in distributed networks. *IEEE Communications Magazine*, 46(2):112–119, 2008.

[191] M. Torchia, M. Kumar, and V. Turner. Worldwide semiannual internet of things spending guide. *IDC (International Data Corporation)*, 2017.

[192] M. Tosic, O. Ikovic, and D. Boskovic. Soft sensors in wireless networking as enablers for sdn based management of content delivery. In *2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 559–564. IEEE, 2016.

[193] C.-F. Tsai, Y.-F. Hsu, C.-Y. Lin, and W.-Y. Lin. Intrusion detection by machine learning: A review. *Expert Systems with Applications*, 36(10):11994–12000, 2009.

[194] C.-W. Tsai, C.-F. Lai, M.-C. Chiang, and L. T. Yang. Data mining for internet of things: A survey. *IEEE Communications Surveys & Tutorials*, 16(1):77–97, 2013.

[195] M. Turkanović, B. Brumen, and M. Hölbl. A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the Internet of Things notion. *Ad Hoc Networks*, 20:96–112, 2014.

[196] M. Valenta and P. Sandner. Comparison of ethereum, hyperledger fabric and corda. *[ebook] Frankfurt School, Blockchain Center*, 2017.

[197] P. Vigna and M. J. Casey. *The age of cryptocurrency: how bitcoin and the blockchain are challenging the global economic order*. Macmillan, 2016.

[198] VizLore LLC. VizLore IoT Platform. `https://vizlore.com/iot`, 2020.

[199] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[200] Q. Wang, R. Sun, X. Zhang, Y. Sun, and X. Lu. Bluetooth positioning based on weighted k-nearest neighbors and adaptive bandwidth mean shift. *International Journal of Distributed Sensor Networks*, 13(5):1550147717706681, 2017.

[201] W. Wang, J. Chen, G. Huang, and Y. Lu. Energy efficient hvac control for an ips-enabled large space in commercial buildings through dynamic spatial occupancy distribution. *Applied Energy*, 207:305–323, 2017.

[202] X. Wang and R. Karri. Reusing hardware performance counters to detect and identify kernel control-flow modifying rootkits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 35(3):485–498, 2016.

[203] X. Wang, C. Konstantinou, M. Maniatakos, and R. Karri. Confirm: Detecting firmware modifications in embedded systems using hardware performance counters. In *Proceedings of the IEEE/ACM international conference on computer-aided design*, pages 544–551. IEEE Press, 2015.

[204] Y. Wang and L. Shao. Understanding occupancy pattern and improving building energy efficiency through Wi-Fi based indoor positioning. *Building and Environment*, 114:106–117, 2017.

[205] Z. Wang, T. Han, and H. Yu. Research of mdcop mining based on time aggregated graph for large spatio-temproal data sets. *Computer Science and Information Systems*, 16(3):891–914, 2019.

[206] A. Wempe and R. Keefe. Characterizing rigging crew proximity to hazards on cable logging operations using GNSS-RF: Effect of GNSS positioning error on worker safety status. *Forests*, 8(10):357, 2017.

[207] X. Wen, L. Shao, Y. Xue, and W. Fang. A rapid learning algorithm for vehicle classification. *Information Sciences*, 295:395–406, 2015.

[208] M. Werner and M. Kessel. *Organisation of Indoor Navigation data from a data query perspective*. IEEE, 2010.

[209] C. Wu, Z. Yang, Y. Liu, and W. Xi. Will: Wireless indoor localization without site survey. *IEEE Transactions on Parallel and Distributed Systems*, 24(4):839–848, 2013.

[210] F. Wu, L. Xu, S. Kumari, X. Li, J. Shen, K.-K. R. Choo, M. Wazid, and A. K. Das. An efficient authentication and key agreement scheme for multi-gateway wireless sensor networks in IoT deployment. *Journal of Network and Computer Applications*, 89:72–85, 2017.

[211] A. Yaeli, P. Bak, G. Feigenblat, S. Nadler, H. Roitman, G. Saadoun, H. J. Ship, D. Cohen, O. Fuchs, S. Ofek-Koifman, et al. Understanding customer behavior using indoor location analysis and visualization. *IBM Journal of Research and Development*, 58(5/6):3–1, 2014.

[212] D. Yan, W. O'Brien, T. Hong, X. Feng, H. B. Gunay, F. Tahmasebi, and A. Mahdavi. Occupant behavior modeling for building performance simulation: Current state and future challenges. *Energy and Buildings*, 107:264–278, 2015.

[213] Z. Yang, C. Wu, and Y. Liu. Locating in fingerprint space: wireless indoor localization with little human intervention. In *Proceedings of the 18th annual international conference on Mobile computing and networking*, pages 269–280. ACM, 2012.

[214] N. Ye, Y. Zhu, R.-c. Wang, R. Malekian, and L. Qiao-Min. An efficient authentication and access control scheme for perception layer of internet of things. *Applied Mathematics & Information Sciences*, 8(4):1617, 2014.

[215] S. Yi, Z. Qin, and Q. Li. Security and privacy issues of fog computing: A survey. In *International conference on wireless algorithms, systems, and applications*, pages 685–695. Springer, 2015.

[216] M. K. Yogi, K. Chandrasekhar, and G. V. Kumar. Mist computing: Principles, trends and future direction. *arXiv preprint arXiv:1709.06927*, 2017.

[217] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi. Internet of things for smart cities. *IEEE Internet of Things journal*, 1(1):22–32, 2014.

[218] T. Zanni. KPMG: The changing landscape of disruptive technologies, 2018.

[219] M. Zhou, Z. Tian, K. Xu, X. Yu, X. Hong, and H. Wu. Scanme: location tracking system in large-scale campus Wi-Fi environment using unlabeled mobility map. *Expert Systems with Applications*, 41(7):3429–3443, 2014.

# Prošireni Izvod

## Predmet istraživanja

Internet stvari (eng. Internet of Things, skraćeno IoT) opisuje ogroman spektar standarda, arhitektura, protokola, aplikacija, tehnologija i tehnika za prikupljanje i analizu podataka. Pomoću njih, uređaji opremljeni senzorima i aktuatorima, kroz posebno dizajnirani softver, kao i ostali elektronski i digitalni sistemi koji su povezani na Internet i druge mreže komuniciraju u cilju ostvarivanje nekog društvenog, industrijskog ili poslovnog cilja. Kao konglomerati takvih uređaja javljaju se IoT platforme, koje imaju za cilj da njihovim udruživanjem postignu ostvarivanje specifične funkcije ili slučaja korišćenja: kao što je IoT sistem za unutrašnje pozicioniranje i praćenje objekata. Dakle, IoT sistemi su dinamički sistemi koji funkcionišu u određenom fizičkom kontekstu. Taj kontekst podložan je stalnim promenama koje su prouzrokovane brojnim faktorima kao što su broj i tip učesnika sistema u određenom trenutku, učestalost i vrsta komunikacije između uređaja, spektralna slika (slika rasporeda i kretanja radio talasa) unutrašnjeg prostora, fizičke prepreke koje unose smetnje, zahtevi za efikasnošću sistema i kontrola trošenja fizičkih ili logičkih resursa. Ove karakteristike otežavaju kreiranje robusnih sistema za upravljanje raspodelom logičkih resursa tj. ravnomernu distribuciju upravljačkih algoritama u samom sistemu. Stoga je predlog algoritma za upravljanje resursima (ravnomernu distribuciju) visoko distribuiranih IoT sistema (koji su svesni konteksta u kom operišu) prvi istraživački zadatak ovog doktorata.

Sistemi za unutrašnje pozicioniranje objekata ispunjavaju zadatak lociranja objekata u unutrašnjem prostoru koristeći radio talase, zvučne/video izvore, ili druge senzorske uređaje i podatke (identifikacija putem radio frekvencije, eng. Radio frequency identification (RFID), infracrveni senzori, senzori za detekciju pokreta i sl.). Ovi sistemi mogu se podeliti u odnosu na tehnologije koja se koristi za lociranje (tip radio talasa ili senzora), njihove primarne svrhe (detekcija, lociranje, praćenje, ili njihova kombinacija), zahteva za preciznost tj. greške (izraženog u metrima, npr. prosečna greška u pozicioniranju mora biti ispod 1 metra), i tipa uređaja koji vrše kalkulacije pozicija (mobilni ili statički). Detekcija pozicije i praćenje objekata u unutrašnjem prostoru je kompleksan problem jer u mnogome

zavisi od konteksta u okviru kog sistem za unutrašnje praćenje radi – tip i raspored unutrašnjeg prostora, dinamičke i statičke fizičke prepreke, fluktuacija signala i smetnje predstavljaju ozbiljne izazove u ostvarivanju zadovoljavajuće preciznosti kod ovakvih sistema. Pored toga, za IoT sisteme jedan od osnovnih zahteva je minimalna iskorišćenost resursa (skladište, memorija, processor) kao i obrada podataka što bliže njihovom izvoru, sve u cilju optimizacije procesa donošenja odluka. Osim toga, IoT sistem za unutrašnje praćenje mora se dinamički prilagođavati fizičkom kontekstu u kom radi u cilju pobošljanja funkcionalnosti sistema, a i zbog karakteristike dinamičnosti samih IoT sistema – IoT sistem čine uređaji koji se frekventno priključuju/napuštaju IoT sistem, imaju različite fizičke i logičke karakteristike, i učestalo komuniciraju sa drugim uređajima koji su/nisu deo IoT sistema.

Postojeće tehnologije za spoljašnje geografsko pozicioniranje, kao što je GPS (globalni pozicioni sistem), nisu primenljive u sistemima za unutrašnje praćenje kretanja zbog potencijalno velike greške u proceni pozicije (5–50 metara). Sistemi za unutrašnje pozicioniranje zahtevaju veću preciznost i manju granularnost pozicioniranja. Veća preciznost označava smanjenje greške u proceni do ispod 1m, dok manja granularnost podrazumeva da je u većini slučajeva potrebno znati apsolutnu poziciju, a potom i poziciju na nivou prostorije. Zbog nepogodnosti GPS-a za sisteme za unutrašnje pozicioniranje, oni moraju da se baziraju na manje robusnim tehnologijama (Wi-Fi, Bluetooth, video nadzor, senzori pokreta, itd.) i zbog toga se susreću sa nizom izazova fizičkog konteksta unutrašnjih prostora. Prvi problem su konstantne fluktuacije posmatranih signala zbog fizičkih prepreka i drugih signala. Ovakve fluktuacije moraju se pažljivo prikupljati, analizirati, filtrirati, i informacije o njima moraju da se uključe u sisteme i algoritme odgovorne za procenu pozicije. Drugi problem su stalne promene u fizičkom kontekstu funkcionisanja sistema, što podrazumeva promenu izgleda fizičkog prostora (pomeranje metalnih objekata, mašina, itd.). Te promene dodatno negativno utiču na preciznost sistema za unutrašnje pozicioniranje. Dodatno sistemi za unutrašnje pozicioniranje, da bi bili efektivni, treba da pružaju što više informacija o fizičkom kontekstu u kom rade. Takvi dodatni servisi treba da uključuju računanje zauzetosti prostorija, ekstrakciju šema/paterna njihove zauzetosti, kao i dugoročno predviđanje zauzetosti, ekstrakciju šema/paterna kretanja posmatranih objekata, i slično. Stoga, druga celina koja će biti obrađena u okviru ovog doktorata nudi kreiranje kompleksnog softverskog okvira za unutrašnje pozicioniranje koji uzima u obzir sve gorenavedene okolnosti.

Mešavina problema i zahteva za kreiranje IoT platforme sa ugrađenim sistemom za unutrašnje pozicioniranje i praćenje objekata daje dobru osnovu za ozbiljan istraživački i inženjerski poduhvat. Stoga je, u okviru ovog doktorata, pokriveno stvaranje napredne IoT platforme za unutrašnji sistem pozicioniranja koja obuhvata detekciju, pozicioniranje i unutrašnje praćenje objekata bazirano na Bluetooth protokolu. Zahtev za preciznost ovog sistema je greška u pozicioniranju ispod 1m,

dok je tip uređaja koji vrše kalkulacije statički. Istraživanje u okviru ovog doktorata podeljeno je na dva dela:

A Kreiranje softverske arhitekture za automatizovano upravljanje distribucijom logičkih resursa u IoT platformi u cilju kreiranja IoT platforme sposobne za samostalno rukovođenje svojim resursima u odnosu na definisana pravila ponašanja;

B Kreiranje naprede softverske arhitekture za unutrašnje pozicioniranje, praćenje objekata i kompleksnu analitiku nad vremenskim serijama podataka o pozicijama.

Istraživački zadaci (A) i (B) rezultuju implementacijom koja je postavljena i aktivna na dve lokacije: u poslovnom prostoru fondacije VizLore Labs u Srbiji (pogledati sekciju 5.3.1), i u stambenoj zgradi u USA 5.3.2 (zbog zaštite privatnosti stanara tačna lokacija neće biti otkrivena). U nastavku teksta stavke (A) i (B) biće detaljno predstavljane a istraživački zadaci i izazovi unutar njih rasvetljeni i ilustrovani primerima.

## Istraživački izazovi

Internet stvari (IoT) opisuje ogroman spektar standarda, arhitektura, protokola, aplikacija, tehnologija i tehnika za prikupljanje i analizu podataka. Kroz ove funkcionalnosti, uređaji opremljeni senzorima i aktuatorima komuniciraju kroz posebno dizajnirani softver radi ostvarivanja nekog društvenog, industrijskog ili poslovnog cilja. IoT sistemi su dinamički sistemi koji operišu u određenom fizičkom kontekstu podložnom konstantnim promenama koje se odnose na broj i tip učesnika sistema, učestalost i tip komunikacije između uređaja, spektralnu sliku i fizičke prepreke, itd. Edge computing sistemi predstavljaju jednu paradigmu u modelovanju IoT sistema, gde se teži tome da se svi podaci obrađuju što bliže mestu gde nastaju, tj. na fizičkim uređajima koji prikupljaju podatke (ako je moguće) i koji se nalaze na samoj ivici mreže. Stoga, IoT i edge computing softverske arhitekture zahtevaju napredna rešenja za upravljanje resursima, praćenje promena u kontekstu funkcionisanja sistema, zaštitu, nadzor operativnog statusa uređaja sistema, održavanje sistemskih funkcija i balansiranje radnog opterećenja između uređaja.

Za edge computing sisteme, potrebni su robusni i modularni podsistemi koji podržavaju niz gorenavedenih funkcionalnosti. Robusnost podrazumeva da je neophodno da se ovakvi podsistemi efektivno skaliraju sa porastom broja uređaja u posmatranom edge computing sistemu. Modularnost je bitna zbog toga što, idealno, ovakvi podsistemi treba da budu primenljivi na širok spektar različitih hardverskih i softverskih infrastruktura. U dosadašnjem naučnom i industrijskom

razvoju, mahom su ove funkcionalnosti bile posmatrane odvojeno, dok je potreba za njihovim univerzalnim ujedinjavanjem posmatrana sa rezervom – zbog nedostatka standarda, interoperabilnosti i odgovarajuće infrastrukture.

Sistemi za unutrašnje pozicioniranje objekata ispunjavaju zadatak lociranja objekata u unutrašnjem prostoru koristeći radio talase, zvučne/video izvore, ili druge senzorske uređaje i podatke (identifikacija putem radio frekvencije, eng. radio frequency identification (RFID), infracrveni senzori, senzori za detekciju pokreta i sl.). U istraživanjima u domenu sistema za unutrašnje pozicioniranje ostvaren je značajan napredak u preciznosti koristeći napredne tehnike za filtriranje vremenskih serija o pozicijama, mašinsko učenje, i probabilističke metode, koje se odnose posebno na sisteme koji funkcionišu u IoT okruženju. Međutim, sistemi za unutrašnje praćenje još uvek su pogođeni nizom problema. Prvi problem je što su modeli za merenje distanci u odnosu na jačine signala neprecizni, zbog prirode signala koji se posmatraju (fluktuacije, promene). Drugi problem ogleda se u osobinama aktuelnih algoritama mašinskog učenja koji su najpopularniji kod ovakvih sistema: fingerprinting. Fingerprinting je tehnika za prepoznavanja pozicije objekta na osnovu spektralne slike signala u neposrednoj okolini, koristeći različite algoritme mašinskog učenja. Ove tehnike, koje se uzimaju kao polazna tačka za unutrašnje lociranje bazirano na mašinskom učenju, podležu dugotrajnim procesima prikupljanja i mapiranja podataka za treniranje modela. To ih čini neprimenljivim u visoko-dinamičnim sistemima kakvi su IoT i edge computing sistemi. Treći problem, ujedno i problem koji povezuje dve celine kojima se bavi ovaj doktorat, je taj što sistemi unutar kojih funkcionišu komponente za unutrašnje pozicioniranje i sam sistem za unutrašnje pozicioniranje često se posmatraju kao dve odvojene celine, dok je neophodno da uče jedna od druge.

Da bi bili uspešni u procenama pozicija ali i korisni u smislu da se drugi servisi bazirani na tačnoj lokaciji mogu nad njima graditi, neophodno je kontekstualno znanje i kontinualno učenje o unutrašnjem prostoru koji se posmatra. Praćenje devijacije i filtriranje signala i drugih kontekstualnih podataka znatno utiče na donošenje odluka u skoro realnom vremenu. Poželjno je da sistem za unutrašnje pozicioniranje podržava i detekciju i praćenje pravilnosti u kretanju ljudi i objekata i zauzetosti prostorija, jer se nad takvim podacima mogu izvlačiti zaključci važni za upravljanje resursima samog unutrašnjeg prostora (npr. efikasno osvetljenje, upotreba sistema za hlađenje/grejanje).

## Ciljevi istraživanja

Prva hipoteza istraživanja je da se efikasnost algoritama za distribuciju upravljačkih algoritama IoT/edge computing sistema povećava sa količinom prikupljenih i obrađenih podataka o fizičkom kontekstu u kom sistem radi. Dodatno, koristeći postojeće komunikacione standarde za IoT/edge computing sisteme, integracija

rešenja za distribuciju uprljačkih algoritama u postojeće IoT/edge computing sisteme je jednostavnija i brža.

Druga originalna hipoteza od koje polazi istraživanje je da sistem za unutrašnje pozicioniranje koji figuriše u okviru nekog IoT sistema mora da konstantno uči o fizičkom kontekstu u kom funkcioniše. Na taj način, povećava se preciznost sistema, i stiču se osnove za kreiranje različitih servisa na bazi lokacija. Takav sistem mora da pored procena pozicije nudi različite servise koji poboljšavaju sam sistem, ali i fizički kontekst u kom se sistem nalazi (poslovna zgrada, privatna zgrada) kroz prikupljanje, analizu i zaključke neophodne za povećanje efikasnosti ili komfora korisnika fizičkog prostora. Sprovođenje istraživanja i eksperimenata u okviru ovog doktorata, kako bi se ilustrovale i potvrdile gorenavedene hipoteze, objediniće sledeće specifične rezultate:

1. Dizajn softverskog okvira i algoritama za detekciju isključivanja uređaja ili upravljačkih procesa sa mreže i njihovu uravnoteženu redistribuciju;

2. Kreiranje modela za pregled aspekata sigurnosti IoT/edge computing sistema, sa naglaskom na distribuirana i decentralizovana sigurnosna rešenja;

3. Kreiranje algoritama, dizajn tokova podataka, implementacija i validacija softverskog sistema za 2D i 3D pozicioniranje bazirano na Bluetooth i Wi-Fi protokolima, uz funkcije za filtriranje pozicija i signala;

4. Uvođenje novih pristupa pri modeliranju unutrašnjih prostora i kontekstualnih metapodataka za sisteme za unutrašnje pozicioniranje kroz korišćenje numeričkih matrica i grafova;

5. Kreiranje algoritama i tokova podataka (implementacija i validacija) za određivanje prisutnosti objekata u unutrašnjem prostoru koristeći samo lokalnu mrežnu infrastrukturu uz modele i algoritme za dinamičko, automatizovano otkrivanje šema ponašanja i kretanja objekata/ljudi u zatvorenim prostorima;

6. Uvođenje novih primena algoritama za mašinsko učenje i to: za klasifikaciju fizičkih delova unutrašnjih prostora i za dugoročno predviđanje zauzetosti unutrašnjih prostorija, uz poređenje sa aktuelnim pristupima;

7. Razvoj softverskog sistema za dinamičko kreiranje virtuelnih granica i stvaranje relacija između virtuelnih granica u unutrašnjem prostoru.

## Rezultati istraživanja

Rezultati istraživanja biće predstavljeni kroz dve podsekcije, kako je i napravljena podela doktorata. U okviru obe podsekcije naći će se sumirana objašnjenja

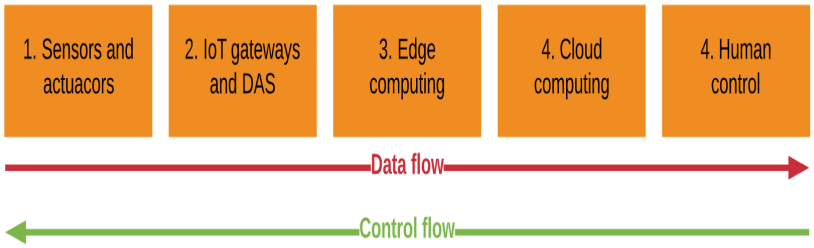



**Figure 1.1:** 5 gradivnih blokova IoT arhitektura.

rezultata, dok se čitalac može detaljno upoznati sa postignutim rezultatima u referenciranim poglavljima i sekcijama.

## Deo I

**Deo I** ima za cilj da objasni IoT i edge computing računarske sisteme i prateće arhitektonske paradigme. **Poglavlje 1** usredsređeno je na IoT i edge computing računarske sisteme. Njihove definicije istražene su u **Sekciji 1.1**, fokusirajući se na aktuelna istraživanja i kombinujući te definicije u jedinstvenu definiciju edge computing-a. U **Sekciji 1.2** diskutovani su tipični IoT i edge computing tipovi dizajna arhitektura, njihove komponente, nabrajajući pet faza dizajniranja IoT i edge computing arhitektura. Ove faze, ili gradivni blokovi su redom: senzori i aktuatori, IoT kontroleri, edge computing, računanje u oblaku (eng. cloud computing) i ljudska kontrola. Senzori mogu transformisati informacije dobijene iz spoljnog sveta u podatke za dalju obradu. IoT kontroleri su tipična polazišta za obradu informacija, nakon što je senzor prikupio informacije, ili pre nego što treba da se dalje procesuiraju. Bilo koja laka ili teška obrada podataka izvršena na ivici mreže, tj. na uređajima koji se nalaze u istom fizičkom kontekstu kao senzori, aktuatori i kontroleri ( dva prethodna sloja) smatra se edge computing-om. Za IoT sisteme koji zahtevaju strogu ljudsku kontrolu ili na neki drugi način zahtevaju kontrolisane korisničke unose, poslednja faza uključuje omogućavanje kontrole nad komponentama IoT arhitekture za administratore sistema i/ili krajnje korisnike. Svi gradivni blokovi prikazani su i na Slici 1.1. **Sekcija 1.3** pruža sumiranje perskektivnih prilika, problema i uticaja IoT i edge computing arhitektura u oblastima industrije (Podsekcija 1.3.1) i globalne ekonomije (Podsekcija 1.3.2), naglašavajući područja primene i uobičajene slučajeve korišćenja, kao i operativne, poslovne, ekonomske i socijalne izazove i mogućnosti. Glavne industrije koje imaju dobiti/pogodnosti od IoT i edge computing arhitektura uključuju proizvodnju i upravljanje proizvodnim procesima, transport i energetske sisteme. Ovo su takođe i najveće industrije sa stanovišta potrošnje u IoT-u (prema izveštaju IDC-a [30]). I na kraju, kako su servisi koji zahtevaju podatke o lokaciji važna tema ove teze, o njima se eksplic-

itno diskutuje u istoj sekciji. Kao primer, primena servisa koji zahtevaju podatke o lokacijama raznih entiteta ima potencijal da poboljša gotovo svaki aspekt infrastrukture pametnog grada: kontrolu saobraćaja i transporta generalno, energetsku efikasnost, planiranje životne sredine, rast tržišta nekretnina, smanjenje kriminala, upravljanje događajima itd. U okviru **Sekcije 1.3** takođe su ilustrovane i savremene potrebe i zahtevi u izgradnji IoT sistema i trenutno najsavremeniji pristupi u njihovom dizajniranju.

Sledeće poglavlje, **Poglavlje 2**, raspravlja o temama bezbednosti i privatnosti IoT i edge computing računarskih sistema. **Sekcije 2.1 i 2.2** razrađuju ovu temu, sa posebnim osvrtom na neuobičajen bezbednosni kontekst IoT i edge computing sistema, probleme i inženjerske zahteve. Osiguravanje bezbednosti IoT i edge computing sistema veoma se razlikuje od osiguravanja bezbednosti tradicionalnog servera ili centra podataka. Ove paradigme uvode veliku mogućnost za napad na sajber-bezbednost sistema – misli se na veliki broj uređaja pre svega. Mnogi IoT uređaji (npr. senzori) raspoređeni su u velikoj količini (ogroman broj senzora u jednom sistemu). Kao rezultat, potencijalni broj međusobno povezanih uređaja teško se deterministički određuje. Pored toga, mnogi od ovih uređaja će morati da komuniciraju sa drugim, spoljnim uređajima. Dok se **Sekcija 2.3** fokusira na prikaz istraživanja koji diskutuju različite preglede i kategorizacije sigurnosti u IoT i edge computing računarskim sistemima, u **Sekciji 2.4** predstavljen je jedan novi, metodološki okvir za pregled bezbednosti za IoT i edge computing sistemske arhitekture. Predstavljeni model, CAAVI-RICS, originalan je doprinos ove teze. CAAVI je akronim za verodostojnost (eng. Credibility), autentifikaciju (eng. Authentication), autorizaciju (eng. Authorization), verifikaciju (eng. Verification) i integritet (eng. Integrity). Principi CAAVI-ja detaljno su razmotreni kroz 4 aspekta (RICS). To su redom: (1) Obrazloženje ili eng. Rationale (šta je to i zašto je važno?), (2) Uticaj ili eng. Influence (kako utiče na opšte blagostanje sistema ako (ni)je pravilno implementirano), (3) problemi ili eng. Concerns (koje probleme donosi?) i (4) rešenja ili eng. Solutions (pregled aktuelnih rešenja). Diskusija ovog modela sažeta je u ovoj tezi. Detaljna razrada modela CAAVI-RICS predstavljena je u tri rada [153, 154, 152]. Vizualizovani rezime svih principa prikazan je redom navođenja ovih principa na Slikama 2.2, 2.3, 2.4, 2.5 i 2.6. U CAAVI-ju, inženjerska veza između principa je sledeća: prethodni principi omogućavaju naredne, a naredni zahtijevaju prethodne. Ovaj odnos je takođe prikazan na Slici 2.7. Dizajniranje i premošćavanje CAAVI principa je presudno za inženjering bezbednosti sistema, jer (ne)ispravan dizajn prethodnog principa može značajno da utiče na naredni.

**Poglavlje 3** dela I sastoji se od istraživanja, dizajniranja i primene Okvira za upravljanje resursima (u tezi nazvan RMF) koje je zasnovan na MQTT komunikacionom protokolu. RMF je dizajniran za edge computing arhitekture i u stanju je da rukovodi: upravljanjem resursima, detekcijom kvarova i administracijom primo-

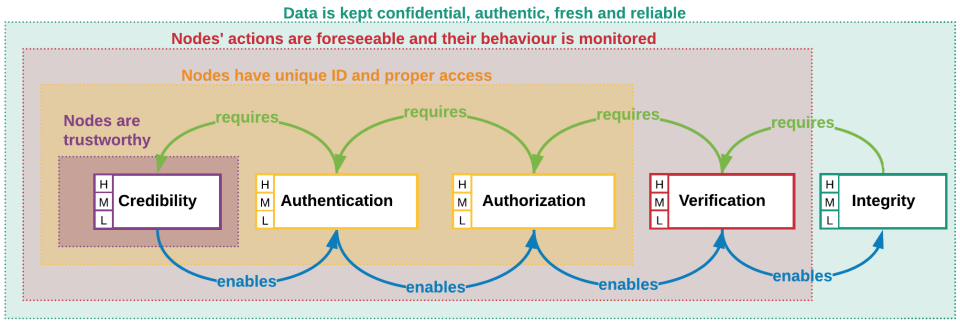



**Figure 2.7:** Dijagram zavisnosti CAAVI principa.

predaje procesa, logičkim i fizičkim balansiranjem opterećenja i praćenjem stanja fizičkih i logičkih resursa sistema. U okviru istog poglavlja takođe je diskutovano o savremenim zahtevima za takve softverske okvire (kao što je RMF), trenutnim pristupima u rešavanju datih problema i ponuđeno je jedno rešenje u vidu softverske arhitekture, sa minimalnim troškovima implementacije i komunikacije. Arhitektura na visokom nivoa predloženog okvira (RMF) za upravljanje resursima za oporavak sastavnih logičkih komponenti edge computing sistema prikazana je na Slici 3.1. RMF modelovan je po uzoru na soft-sensor princip dizajna, koji teži jakom hijerarhijskom razdvajanju funkcionalnosti od najjednostavnijih do najkompleksnijih, tako stvarajući modularnije arhitekture koje su otpornije na veće logičke kvarove.

U nastavku je dato kratko objašenjenje za RMF, a detalji objašnjenja mogu se naći u okviru **Poglavlja 3** dela I. Akteri RMF-a mogu se svrstati u fizičke i softverske komponente. Među fizičkim komponentama postoji tipični hardver za IoT sisteme: to su senzori, aktuatori i drugi uređaji. Senzori i aktuatori mogu se povezati sa više uređaja žicom i/ili bežično – npr. aktuatori 1 i 2 povezani su bežično sa uređajem 1 i povezani su žicom sa uređajem 2. Softverski paket za svaki uređaj sastoji se od dve grupe softverskih modula: (1) grupa koja je zadužena za obezbeđivanje funkcionalnosti IoT platforme (više soft-sensor komponenti, MQTT broker), i (2) grupa koja je zadužena za obezbeđivanje RMF funkcionalnosti (kontroler i SSCM modul). Svi detalji RMF softverske arhitekture i njeni akteri, detaljno su opisani u **Sekciji 3.3**. Dalje, **Sekcije 3.4 i 3.5** detaljno predstavljaju radne procese unutar RMF-a i napredne sposobnosti RMF-a kada je u pitanju broj scenarija koji može da obradi. Konačno, **Sekcija 3.6** prikazuje eksperimente, rezultate i diskusiju o prednostima i manama RMF-a, kao i o narednim koracima u istraživanju i implementaciji.

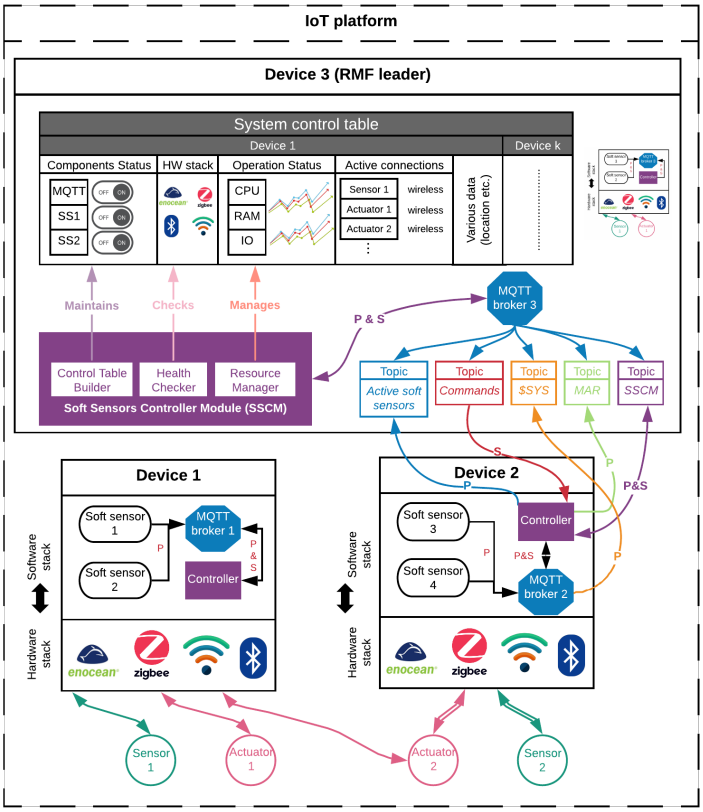



**Figure 3.1:** Radni procesi Okvira za upravljanje resursima.

## Deo II

**Deo II** fokusiran je na BLEMAT, sofisticirani i visoko funkcionalni Bluetooth sistem za unutrašnje pozicioniranje, razvijen kao originalan doprinos ove teze.

**Poglavlje 4** dela II razrađuje detalje o sistemima za unutrašnje pozicioniranje, njihovoj definiciji, vrstama primene, uobičajeno korišćenim tehnikama pozicioniranja, oblastima primene i uobičajenim slučajevima primene, kao i operativnim, poslovnim, ekonomskim i društvenim izazovima. **Sekcija 4.1** daje uobičajene definicije za sisteme za unutrašnje pozicioniranje. **Sekcija 4.2** pruža pregled najčešće korišćenih tehnika pozicioniranja u zatvorenom prostoru i analizira potrebne hardverske i softverske komponente. **Sekcija 4.3** predstavlja tipične opcije fizičkog rasporeda (eng. deployment) za takve sisteme pozicioniranja i objašnjava prednosti i nedostatke za svaki. **Sekcija 4.4** diskutuje specifične zahteve dizajna i izazove za primenu sistema unutrašnjeg pozicioniranja u edge computing računarskim okruženjima. Konačno, **Sekcija 4.5** raspravlja o povezanim istraži-

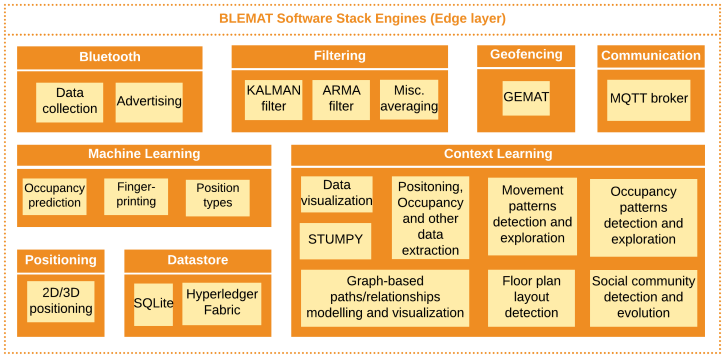



**Figure 5.1:** Funkcionalna arhitektura BLEMAT-a na visikom nivou.

vačkim nedostacima u tehnikama pozicioniranja, filtriranju podataka o položaju i mogućnostima fizičkog rasporeda za sisteme unutrašnjeg pozicioniranja.

**Poglavlje 5** dela II usmereno je na opisivanje komponenata arhitekture i operativnih tokova rada unutar BLEMAT-a. BLEMAT je prostorno-agnostički edge computing računarski sistem, koji u realnom vremenu vrši unutrašnje pozicioniranje i filtriranje, fingerprinting i otkrivanje rasporeda prostorija unutrašnjeg prostora, praćeno različitim složenim analitičkim procesima nad podacima i algoritmima za mašinsko učenje. Sistem postiže visoku tačnost i preciznost u proceni položaja uz održavanje niske upotrebe resursa. BLEMAT je kolekcija softverskih podsistema koji se mogu primeniti kao sastavni deo bilo kog IoT uređaja (npr. Raspberry Pi, Arduino itd.) sa omogućenim Bluetooth i Wi-Fi komunikacionim modulima.

**Sekcija 5.1** predstavlja funkcionalnu arhitekturu visokog nivoa sa dijagramom komponenti sistema BLEMAT. Na edge computing sloju, BLEMAT se sastoji od više IoT kontrolera koji pokreću instancu BLEMAT-a spojenih u meš topologiju mreže – svaki IoT kontroler u mreži ima vezu sa svim drugim IoT kontrolerima u toj mreži. U okviru BLEMAT softverske arhitekture postoji 8 zasebnih softverskih podsistema. Svaka od njih je detaljno objašnjena u **Sekciji 5.1**. Funkcionalna arhitektura visokog nivoa BLEMAT-a je prikazana na Slici 5.1.

**Sekcija 5.2** predstavlja radne tokove/procese u okviru BLEMAT-a koji rade na edge computing sloju (na ivici mreže), na IoT kontrolerima. Ovi procesi aktivno rade na svakom od kontrolera na edge computing sloju. BLEMAT sistem i svi procesi rade non-stop, što pomaže sistemu da uvek izvlači sveže informacije i gradi znanje o fizičkom kontekstu u kom se sistem nalazi. Nadalje, svi proračuni se izvode na edge computing sloju povećavajući sigurnost, integritet, privatnost i poverenje. BLEMAT radni tokovi su detaljno objasnjeni u **Sekciji 5.2**, a odgovarajući dijagram može se videti na Slici 5.4.

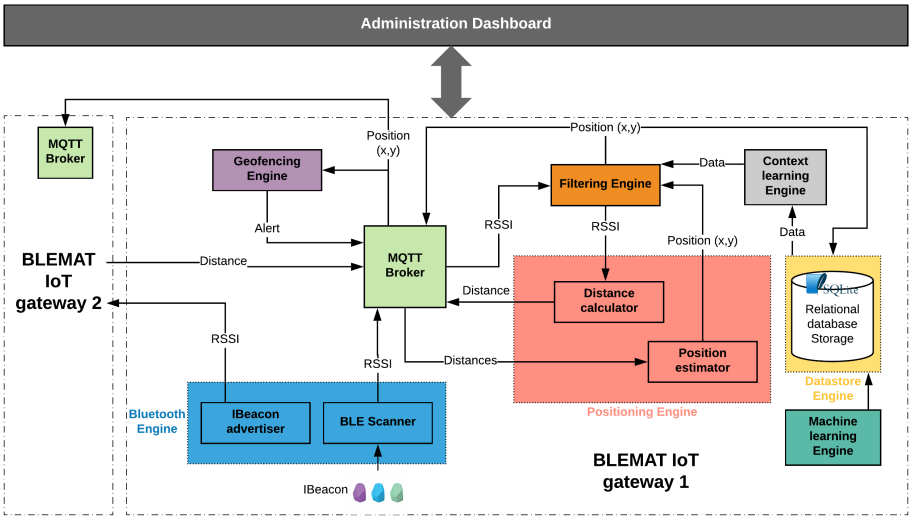


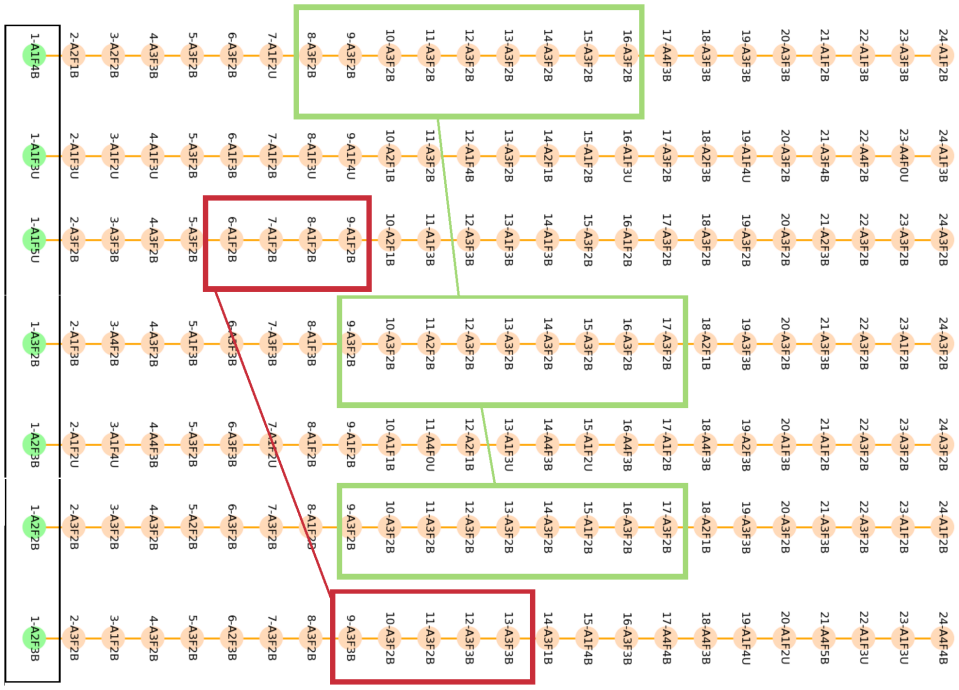

**Figure 5.4:** BLEMAT radni procesi na edge computing nivou.

Poslednja sekcija u **Poglavlju 5** dela II, **Sekcija 5.3**, predstavlja fizičke kontekste u kojima BLEMAT aktivno radi, njihov obim, lokaciju i detalje o hardveru/softveru. BLEMAT je aktivan na dve lokacije, pružajući podatke i kontekst za eksperimente o kojima će biti dalje reči u nastavku.

Unutar BLEMAT-a nalazi se niz namenskih softverskih podsistema, svaki od njih zadužen za analizu određenog aspekta unutrašnjeg prostora. Dok su neki povezani sa demistifikacijom fizičkog konteksta unutrašnjeg prostora (protok ljudi, spektralne mape, položaji prepreka, itd.), drugi su povezani sa istraživanjem ponašanja i socijalne dinamike i interakcije posmatranih ljudi i utvrđivanjem obrasca korišćenja unutrašnjeg prostora. **Poglavlje 6** obrađuje sve podsisteme implementirane u BLEMAT-u. Ove komponente, povezana aktuelna istraživanja, originalni doprinosi ove teze, detalji implementacije, rezultati i diskusija o njima prikazani su u okviru Poglavlja 6. U ovom delu teze ukratko će biti opisani podsistemi u okviru BLEMAT-a i biće prikazane glavne funkcionalnosti.

**Sekcija 6.1** predstavlja procese unutar BLEMAT-a zadužene za modeliranje i analizu obrazaca kretanja posmatranih objekata ili ljudi. BLEMAT-ov *Context learning Engine* podržava modeliranje i analizu obrazaca kretanja za posmatrane Beacon-e i koristi grafove kao strukture podataka da bi to postigao. Upotreba grafova za modeliranje ponašanja Beacon-a u BLEMAT-u zahteva upotrebu dve karakteristične vrste grafova. S jedne strane, za predstavljanje kretanja stanara BLEMAT koristi linearni graf. Čvorovi ovog grafa su oznake stanova u posmatranom unutrašnjem prostoru gde je posmatrani Beacon/stanar detektovan u određeno

**Figure 6.3:** 7-dnevno kretanje stanara.



vreme, agregirano na sat vremena. S druge strane, za predstavljanje društvenih odnosa i ponašanja stanara kroz podatke o zauzetosti stanova, BLEMAT koristi usmereni težinski graf. Graf ponašanja stanara je skup čvorova koji predstavljaju stanove u posmatranoj zgradi koje je stanar posetio. Grafovi i njihove formalne definicije opisani su u Sekciji 6.1. Kao primer, Slika 6.3 prikazuje linearni graf putanje za jednog stanara/Beacon-a, sa vremenskom aregacijom od 1 sata, u trajanju od jedne nedelje, počevši od 1-og novembra posmatrane godine. Iz ovakvih grafova možemo doneti zaključke o najposećenijim stanovima i vizuelno otkriti obrasce kretanja. BLEMAT takođe omogućava kreiranje 3D grafove kretanja posmatranih stanara. Otkrivanje postojanja obrazaca kretanja važno je za efikasno korišćenje resursa unutrašnjeg prostora. Obrazac kretanja objekta/osobe može se koristiti za otkrivanje odstupanja u stvarnom vremenu i signaliziranje vanrednih/neobičnih situacija, ali i proaktivno korišćenje resursa (tj. uključivanje svetala, poziv lifta itd.). Kada se modeliraju kao grafovi, ovi obrasci kretanja mogu da se analiziraju sa različitim algoritmima sličnosti grafova kako bi se otkrile potencijalne podudarnosti (pogledati **Sekciju 6.1.3**) No, jednostavni oblik ovih grafova stanara čine ih pogodnim za povećanje računskih performansi njihovom transformacijom u linearni niza stringova. Pretvaranje grafova u rečenice i pokretanje funkcija sličnosti stringova štedi 2–6 sekundi po jednom paru upoređenih grafova.

**Sekcija 6.2** predstavlja dizajn i primenu specijalizovanog podsistema BLEMAT sa jednostavnim, laganim i novim pristupima za predviđanje zauzetosti prostorija, analizu podataka, vizualizaciju i procese za pretraživanje šema zauzetosti, koji zahtevaju minimalan napor za upotrebu na postojećim infrastrukturama u pametnim zgradama. Ovde je predstavljen model za predviđanje dugoročne binarne zauzetosti stanova (zauzet ili nije zauzet) za nedelju dana unapred. Model je zasnovan isključivo na podacima o upotrebi Bluetooth i Wi-Fi iz pametne zgrade a ti podaci dobijaju se na osnovu aktivnog skeniranja korisničkih uređaja koji koriste lokalnu mrežu. Da bi radio, ovaj podsistem ne zahteva dodatne senzore koji indiciraju zauzetost prostora ili bilo koje druge senzore. Na kraju, prikazuju se procesi za autonomno otkrivanje šema zauzetosti unutrašnjeg prostora po željenom području granularnosti (zgrada, sprat ili stan). Za predviđanje koriste se Long Short Term Memory (LSTM) neuronske mreže, pogodne za obradu velike količine istorijskih podataka. Na početku, način na koji se prikupljaju podaci o zauzetosti detaljno je objašnjen u Podsekciji 6.2.2. Kao rezultat obrade različitih podataka dobijaju se konačni podaci o zauzetosti, prikazani u obliku kao na Slici 6.12. 1 označava zauzetost stana, analogno 0 označava da je stan prazan. U Podsekciji 6.2.4 predstavljeni su detalji oko arhitekture i podešavanja parametara za treniranje LSTM neuronske mreže. Evaluacija rezultata pokazala je da je podatke potrebno razdvojiti na radne dane i vikende, i da posmatranje granularnosti zauzetosti na svakih 10 minuta ostvaruje najbolje rezultate. To znači da je potrebno trenirati dve neuronske mreže za svaki stan, no preciznost predikcije značajno raste. Za detalje pogledati podsekciju 6.2.5. U Podsekciji 6.2.7 predstavljena je analitika nad podacima i mehanizmi koji omogućavaju otkrivanje obrazaca zauzetosti stanova u posmatranoj zgradi. Prvo se, u koraku (1) koristeći algoritam za nalaženje najdužih perioda zauzetosti ili odsutnosti stanara određenog stana, na dnevnom nivou donose zaključci o obrascima zauzetosti. Nakon toga se, u koraku (2), koristeći klastering pristup na nedeljnom nivou, rezultati iz koraka (1) analiziraju i izjednačavaju da bi se donela odluka o tome dali postoji obrazac zauzetosti na nedeljnom nivou. U okviru detekcije zauzetosti, moguće je i pronaći i detektovani korisnički-definisane obrasce zauzetosti koji se mogu definisati kao regularni izrazi od brojeva 1 i 0 (1-stan je zauzet, 0-obratno). Za samostalno, nesupervizirano otkrivanje obrazaca

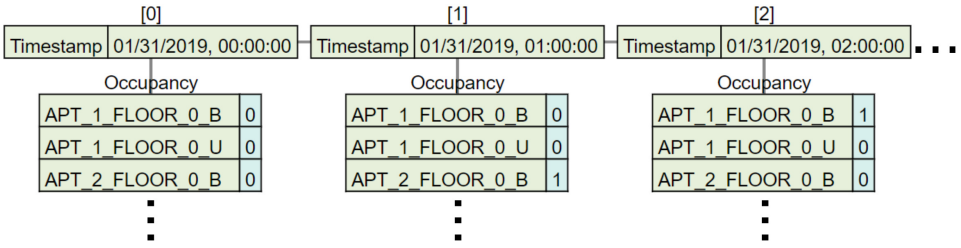



**Figure 6.12:** Konačan skup podataka o zauzetosti prostorija zgrade.

zauzetosti BLEMAT koristi STUMPY biblioteku, i u okviru proračuna baziranih na njoj donosi zaključak o tome koja je periodičnost obrazaca zauzetosti. U okviru posmatranog skupa podataka, potvrđuje se da je periodičnost 7 dana, što je i normalno u skladu sa tim kako ljudi organizuju svoje obaveze, posao, školu, itd. Konačno, doprinos ovog dela doktorata ogleda se u dugoročnoj predikciji, u korišćenju skupa podataka nad privatnim, stambenim zgradama, i u predstavljanju autonomnih mehanizama za otkrivanje obrazaca zauzetosti stanova.

**Sekcija 6.3** uvodi podsistem koji nosi naziv GEMAT. Izgrađen da radi u sklopu BLEMAT-a, okvir Geofencing Micro-Location Asset Tracking (GEMAT) podsistem je polu-nesupervizirani sistem za kreiranje virtuelnih granica koji stalno uči o operativnom kontekstu okruženja u kom se fizički nalazi. Uz operativni kontekst kao što je BLEMAT-ov, mogućnosti kreiranja virtuelnih granica automatski se poboljšavaju funkcionalnostima BLEMAT-a koje imaju za cilj da isprave proračune pozicioniranja na osnovu promena u kontekstu upravljanog sistema.

Virtuelna granica (eng. geofence) je virtualni okvir koji obuhvata određeno ciljno područje unutrašnjeg prostora. Lokacija (2D/3D koordinate) korisnika ili objekta koristi se za procenu da li su unutar ili izvan ciljnog područja, kao i da li oni prelaze ili ne prelaze ciljno područje. U zavisnosti od načina konfigurisanja geofencing-a, ovakav sistem na osnovu tih informacija može signalizirati mobilne push notifikacije, slati tekstualne poruke ili upozorenja, slati ciljane reklame na društvenim mrežama, omogućavati praćenje voznog parka, itd. Prodavac može postaviti virtuelnu granicu oko svojih prodavnica kako bi aktivirao upozorenja o sniženjima koja se prikazuju kao notifikacije na pametnim telefonima kupaca. Geofencing se može koristiti za kontrolu i praćenje vozila u brodskoj industriji [147] ili stoke u sektoru poljoprivrede [175].

GEMAT podsistem počiva na tri glavne komponente (navedene na engleskom): GEMAT Area and Rules Manager, GEMAT Rules Enforcer i GEMAT Notify Engine. Za konkretnu arhitekrutu i način na koji sarađuju ove komponente pogledati Sliku 6.23). Komponenta **GEMAT Area and Rules Manager** odgovorna je za skladištenje virtuelnih granica tj. područja i geofencing pravila, kao i za prosleđivanje upozorenja dalje ako je neko pravilo ispunjeno (virtuelna granica zakačena). Područje virtuelne granice dinamički je određeno i označeno u GEMAT-u kao proizvoljni poligon ili linija na planu unutrašnjeg prostora. **GEMAT Rules Enforcer** predstavlja automatizovani mehanizam za geofencing koji radi sa podacima o pozicioniranju iz BLEMAT-a i GEMAT Area and Rules Manager komponentom kako bi utvrdio da li je potrebno preduzeti neku akciju (tj. neko geofencing pravilo je ispunjeno). **GEMAT Notify Engine** preuzima izlaze GEMAT Rules Enforcer komponente i pretvara ih u akcije – šalje obaveštenje ili naredbu aktuacije željenom IoT uređaju. U Podsekciji 6.3.3 navedeno je deset kriterijuma koje je potrebno uzeti u obzir za merenje sposobnosti i performansi geofencing sistema. Skup kriterijuma zasnovan je na naučnim radovima koji se odnose na dizajn i im-

plementaciju ovakvih sistema. GEMAT prikazuje dobre osobine u poređenju sa drugim sličnim sistemima za geofencing, gde je poređenje bazirano na 10 navedenih kriterijuma. Sumiranje GEMAT-a kroz 10 kriterijuma takođe je pokriveno i detaljno objašnjeno u Podsekciji 6.3.3.

Eksperimenti sprovedeni u okviru GEMAT podsistema izvodili su se u kancelarijskom prostoru opisanom u Sekciji 5.3.1. Prikupljeno je više skupova podataka, a virtuelne granice aktivirane su 6.000 puta tokom perioda prikupljanja podataka. Dodatno, za svako aktiviranje prikupljene su informacije o eksperimentalnom scenariju i uspehu razrešavanja scenarija. Verifikacija scenarija i uspešnost njihovog razrešavanja, postavljanje eksperimenata i opis skupova podataka detaljno su opisani u Podsekciji 6.3.4. Sprovedeni eksperimenti i analiza performansi pokazuju da je predloženi GEMAT podsistem dobar kandidat za rešavanje problema u širokom spektru slučajeva primene geofencing-a u unutrašnjem prostoru.

**Sekcija 6.4** predstavlja podsistem BLEMAT-a za otkrivanje izgleda plana tj. arhitektonske osnove posmatranog unutrašnjeg prostora. Otkrivanje rasporeda prostorija unutrašnjeg prostora odnosi se na otkrivanje informacija o fizičkim preprekama u posmatranom unutrašnjem prostoru. Većina sistema za unutrašnje pozicioniranje ima informacije o tačnom rasporedu unutrašnjeg prostora u kome rade i to se smatra osnovnom istinom (eng. ground thruth) koja se ne menja. To je dovoljno za neke unutrašnje prostore kao što su poslovne zgrade, stambene zgrade, javne zgrade (npr. tržni centri, univerziteti) i druge slične vrste zgrada, jer se fizički

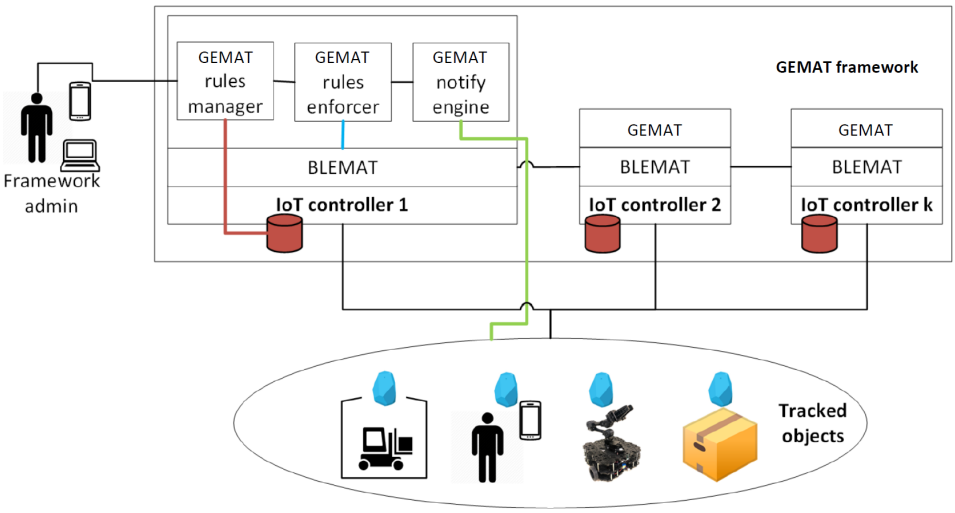



**Figure 6.23:** Radni procesi u okviru GEMAT podsistema.

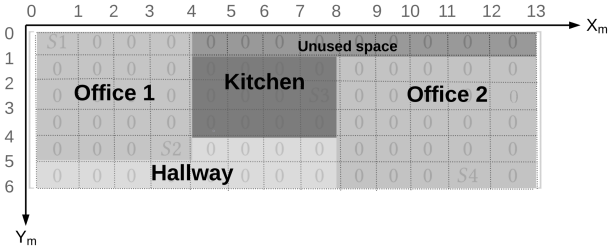



**Figure 6.28:** Plan sprata u kancelarijskom prostoru.

raspored takvih prostora neće promeniti tokom vremena, bar ne dovoljno da može značajno uticati na tačnost sistema za unutrašnje pozicioniranje.

S druge strane, postoje unutrašnji prostori, poput velikih skladišta, koji su prilično jednostavni, bez namenskog fizičkog rasporeda. Ove vrste unutrašnjih prostora nemaju puno zidova, osim spoljnih zidova, i obično imaju puno opreme, pokretne pregradne zidove, mašine, velike regale/police itd. koje se računaju kao prepreke za propagaciju signala u sistemu za unutrašnje pozicioniranje. U ovim tipovima unutrašnjih prostora, prekreke mogu da menjaju mesto i 'masu': velike mašine (npr. viljuškar) mogu se kretati, regali/police mogu se isprazniti i ponovo napuniti, a pregradni zidovi mogu se premestiti na drugo mesta gde su potrebni. Dakle, u ovim unutrašnjim prostorima ne možemo pretpostaviti da će raspored osnove prostora koji je na početku rada sistema postavljen kao osnovna istina ostati zauvek isti. Zbog toga mora postojati efikasan način da se zaključi kako su se

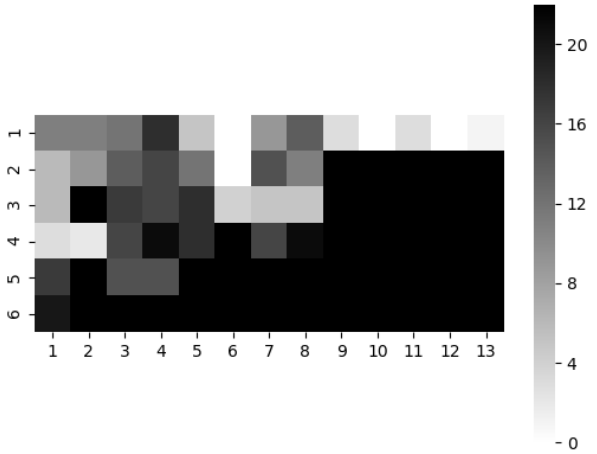



**Figure 6.29:** Heat mapa plana sprata u kancelarijskom prostoru bazirana na Matrici M (odgovara Slici 6.28).

prepreke promenile i kako to utiče na širenje signala. U **Sekciji 6.4**, predstavljeno je autonomno (tj. nesupervizirano) rešenje aproksimacije osnove prostora zasnovano na prikupljanju podataka o pozicioniranju posmatranih Bluetooth beacon-a (ljudi, mašina, itd.), sa osvrtom na obradu podataka, objašnjenje koraka, definisanje algoritma i isticanje rezultata i stope uspešnosti apriksimacije. Kao primer tih rezultata, Slika 6.28 prikazuje originalnu osnovu prostora korišćenu za eksperimente, dok je na Slici 6.29 prikazana aproksimacija generisana predstavljenim algoritmom. Za više detalja i objašnjenja pogledajte Sekciju 6.4.

Društveni odnosi se mogu posmatrati i analizirati sa stanovišta njihovog formiranja, kroz konzistentnost i evoluciju tokom vremena. Generisanje i interpretacija grafova ponašanja diskutovana je u Sekciji 6.1, a u **Sekciji 6.5**, naglasak je na ekstrakciji informacija o konzistentnosti (kontinuitetu) posmatranih odnosa i istraživanju stvaranja i evolucije društvenih zajednica unutar posmatrane stambene zgrade. Eksperimenti su sprovedeni 2019. godine, u junu, novembru i decembru u stambenoj zgradi u kojoj radi BLEMAT (pogledati Sekciju 5.3.2). Društveni odnosi se posmatraju na nivou stana: ako su dva čvora (stana) povezana u grafu ponašanja stanara, to ukazuje da stanari u tim stanovima imaju neku vrstu društvenog odnosa.

Grafovi ponašanja stanara mogu se koristiti za posmatranje društvenog ponašanja, učestalosti, konzistentnosi i evolucije društvenih interakcija i detekciju društvenih zajednica i njihovog posmatranja. Detaljan prikaz BLEMAT-ovih funkcionalnosti koje omogućavaju ove servise prikazan je u **Sekciji 6.5**.Prvo, rešava se problem utvrđivanja konzistentnosti društvenih odnosa. Konzistentnost društvenih odnosa potvrđuje se analizom sličnosti serije grafova kada se posmatraju grafovi ponašanja stanara za određeni period. Primjenjuju se dve metrike sličnosti grafova [93]: GED i Eigenvector sličnost. Ove metrike sa odgovarajućim jednačinama opisane su u Podsekciji 6.5.1. Odluka o tome da li postoje određene društvene interakcije mora se zasnivati na periodičnim zapažanjima i neprekidnom ponavljanju analize grafova ponašanja. Zato je u eksperimentima prosečna sličnost grafova ponašanja stanara posmatrana i agregirana tokom 1 meseca. Kao primer, Slika 6.32 prikazuje otkrivene društvene odnose i njihovu konzistentnost sa stanovišta jednog stana. Slika 6.32 prikazuje konzistentnost društvenih interakcija između stanara stanova *A_1_F_5_U* (označeno narandžastom bojom), *A_4_F_2_B* (označeno plavom bojom) i *A_1_F_0_B* (označeno crnom bojom). Iz ove serije grafova ponašanja jasno je da je društvena interakcije stanara ova tri stana konzistentna tokom jednog meseca.

Kada se istraži konzistentnost socijalnih interakcija za celu zgradu, za 42% stanova (i njihovih stanara) se može utvrditi da imaju konzistentne socijalne interakcije u junu, 66% u novembru, a 63% u decembru (2019). To je takođe prikazano na Slici 6.7.

Podsekcija 6.5.2 se bavi detekcijom društvenih zajednica u posmatranom stambenom objektu i njihovom evolucijom. Detekcija zajednica sastoji se od skupa algoritama koji su dizajnirani da identifikuju povezane grupe čvorova unutar posmatrane mreže. Ove grupe se nazivaju društvene zajednice. Motivacija koja stoji iza otkrića zajednica je raznolika. U marketingu može pomoći kompaniji da razume različite grupe mišljenja o svojim ponudama. Detekcija zajecnica može pomoći u izgradnji i analizi naučne saradnje itd. U posmatranom stambenom objektu, cilj ova teza je ispitivanje trenutnih društvenih struktura visoko povezanih čvorova tj. stanova (i njihovih stanara). Ove informacije se prosleđuju sistemu za upravljanje pametne zgrade i mogu se koristiti za prilagođene ponude društvenim zajednicama (vaučeri za obrok, zajedničnо članstvo u teretani itd.) u stambenom kompleksu gde stanari žive. Za posmatranje društvenih zajednica u zgradi prvo je potrebno kreirati graf ponašanja stanara na nivou čitave zgrade. Ovaj graf će uključivati podskup otkrivenih odnosa u toku jednog meseca – pasivni čvorovi (slabe interakcije) su izostavljeni. Takav graf je prikazan na Slici 6.33 za decembar 2019. godine, a može isto tako da se kreira se i za jun i novembar (2019).

Algoritmi za otkrivanje društvenih zajednica se pokreću na ovoj strukturi grafa (Slika 6.33). Koriste se dva različita pristupa: algoritam Girvan-Nevman (GN) i Louvain metoda (LM). Oba su detaljno objašnjena u Podsekciji 6.5.2. Važno je napomenuti da LM radi znatno brže od GN – 52 puta u proseku po grafu. Zajednice

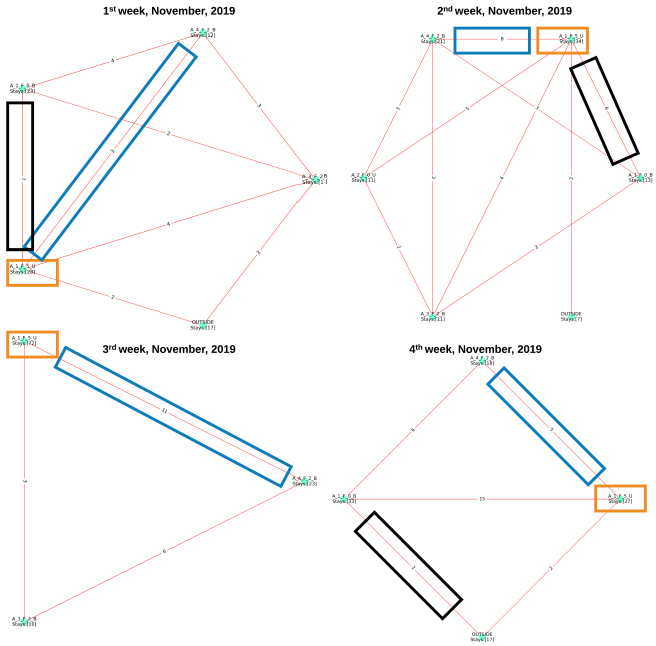



**Figure 6.32:** Konzistentnost društvenih odnosa u toku 1 meseca.

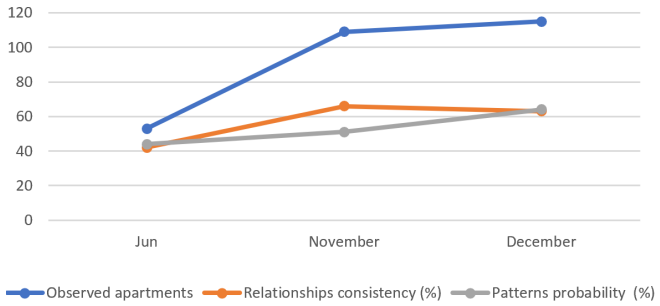



**Figure 6.7:** Karakteristike ponašanja stanara u datom periodu.

koje su generisane primenom GN algoritma prikazane su na Slici 6.36. Zajednice su označene sa $J_k$, $N_k$ i $D_k$ za jun, novembar i decembar. Evolucija dve zajednice (*J1* i *J5*) sa istaknutim najstabilnijim čvorovima (*J1-117, J5-24*) prikazana je isprekidanom linijom: odgovarajuće evolucione staze su *J1-N4-D4* i *J5-N7-D6*. Na takvom grafu (pogledati Sliku 6.36) pokrenut je algoritam za praćenje evolucije društvenih zajednica. Posmatranje životnog ciklusa, tj. evolucija društvene zajednice je relevantna tema za otkrivanje pojava koje pokreću evoluciju – događaja koji izazivaju promene u strukturi mreža društvenih zajednica. Ovi događaji mogu biti interni (zajednički prijatelj osoba A i B deaktivira nalog na društvenoj mreži) ili eksterni (postojanje društevnog događaja). Otkrivanjem vremena kada su se posma-

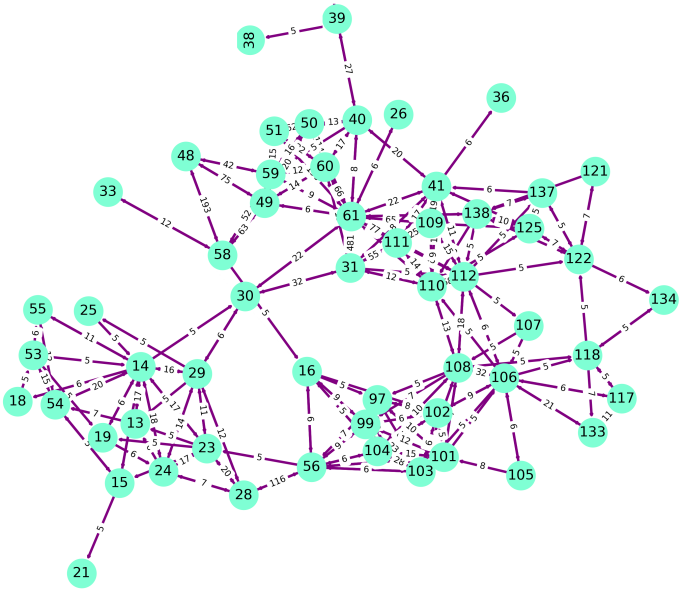


**Figure 6.33:** Društvene zajednice stanara u decembru (2019).

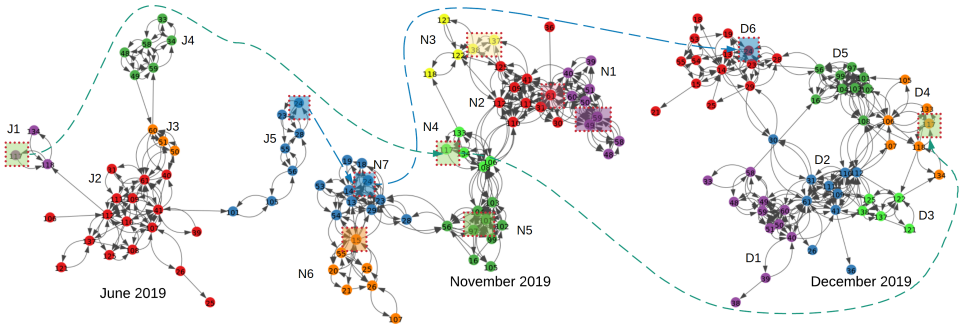



**Figure 6.36:** Društvene zajednice detektovane GN algoritmom u junu, novembru i decembru (2019).

trane društvene zajednice promenile, možemo pokušati da ih vežemo za određene događaje.

Evolucija zajednica generisanih algoritmom Girvan-Nevman, kao rezultat algoritma za praćenje evolucije detaljno je prikazana u Tabeli 6.4. Takođe se istražuju i objašnjavaju evolucione staze. Ista struktura grafa i tabela su prikazane i za LM (pogledati Sliku 6.34 i Tabelu 6.5).

Sistemi za upravljanje u pametnim zgradama mogu koristiti ove informacije za promociju socijalne interakcije između stanara u stambenoj zgradi ili gradskom području koji imaju slične socijalne karakteristike ili su im fizički blizu. Pored toga, ovakvi sistemi mogu dobiti saznanja o društvenoj strukturi svojih stanovnika kroz analizu društvenih zajednica. Znanje o prisustvu društvenih zajednica može se koristiti za poboljšanje veza unutar zajednica, npr. pružajući ulaznice ili pozivnice za društvene događaje ciljanim stanarima u okviru zajednica. Ako je sistem za upravljanje u pametnim zgradama zadužen za više stambenih zgrada, pomenuti pristupi za pronalaženje društvenih zajednica mogu se lako ekstrapolirati u više fizičkih konteksta, povezujući ih i zaključujući obrasce društvenog ponašanja u njima. Posmatranje životnog ciklusa (evolucije) društvenih zajednica i otkrivanje događaja koji ih menjaju mogu se preslikati na stvarne interne ili eksterne događaje. I otkrivanje, i evolucija društvenih zajednica u poslovnim zgradama mogu se mapirati na postojeće projekte, rokove i redovne poslovne procese.

## Zaključak

Ova teza obuhvatala je dva glavna istraživačka zadatka. Složenost sistema Interneta stvari (IoT) i zadataka s kojima se bave zahtevaju promene u načinu upravljanja resursima i pružanju usluga. Stoga je koncept edge computing-a uveden u sferu IoT sistema radi poboljšanja skalabilnosti, reaktivnosti, efikasnosti i privatnosti IoT sistema. Međutim, sam koncept edge computing-a nije dovoljan da podrži dinamiku tipičnog IoT sistema. Treba uspostaviti robusne okvire za: obavljanje

funkcije upravljanja fizičkim i logičkim resursima, efikasan transport upravljačkih procesa koji su prestali sa radom, kao i upravljanje 'zdravljem' sistema. Dalje, mešavina problema i zahteva za stvaranje IoT platforme sa ugrađenim sistemom za unutrašnje pozicioniranje i praćenje objekata stvorilo je čvrst teren za ozbiljno istraživanje i inženjerski poduhvat. Stoga, ova teza pokriva i stvaranje napredne IoT platforme za unutrašnje pozicioniranje koja uključuje računanje lokacija i praćenje objekata na osnovu Bluetooth i Wi-Fi protokola.

*DEO I* imao je za cilj da rasvetli IoT i edge computing računarske sisteme i prateće računarske paradigme softverskih arhitektura, njihovu definiciju, područja primene i slučajeve uobičajene upotrebe, kao i operativne, poslovne, ekonomske, i socijalne izazove i koristi. Ilustrovane su savremene potrebe i zahtevi u izgradnji IoT sistema i najsavremeniji pristupi u njihovom dizajniranju. Raspravljalo se o temama bezbednosti i privatnosti u IoT i edge computing računarskim sistemima. Kao još jedan glavni zadatak, teza je obuhvatila istraživanje, dizajn i implementaciju softverske arhitekture za upravljanje resursima zasnovanim na MQTT komunikacionom protokolu za edge computing računarske sisteme koja se bavi: upravljanjem resursima, detekcijom prestanka rada upravljačkih algoritama i administracijom primopredaje tj. transporta upravljačkih algoritama, i logičkim i fizičkim balansiranjem i zaštitom radnog opterećenja sistema. Diskutovani su savremeni zahtevi za takve softverske arhitekture, trenutni pristupi. Na kraju, prikazano je rešenje sa minimalnim troškovima implementacije i komunikacije.

Što se tiče predstavljenog CAAVI-RICS modela (pogledati sekciju 2.4) budući rad će uključivati studiju slučaja gde će se model primeniti na stvarni problem u okviru nekog edge computing računarskog sistemima, pokazujući tako široke mogućnosti CAAVI-RICS modeliranja sigurnosti. Što se tiče predstavljenog softverskog okvira za upravljanje resursima (pogledati Poglavlje 3), budući rad će uključivati povećanje sofisticiranosti okvira kroz tri aspekta: (1) smanjivanje događaja koji zajtevaju manuelnu tj. ljudsku intervenciju, (2) poboljšavanje algoritma za balansiranje radnog opterećenja i strategije raspodele opterećenja, i (3) smanjenje prosečnog kašnjenja kod komunikacije između komponenti. Prekretnica u implementaciji ovog softverskog okvira, i deo budućeg rada, je njegova integracija sa komercijalnom IoT platformom koju je stvorio VizLore LLC [198].

*DEO II* imao je za cilj da objasni sisteme za unutrašnje pozicioniranje, njihovu definiciju, vrste primene, najčešće korišćene tehnike pozicioniranja, područja primene i uobičajene slučajeve upotrebe, kao i operativne, poslovne, ekonomske, i socijalne izazove i koristi. Posebno se diskutovalo o dizajniranju ovakvih sistema za tipičnu IoT infrastrukturu. Ponuđen je uvid u savremene zahteve sisteme za unutrašnje pozicioniranje, trenutnih pristupa u rešavanju istih, i naglašeni su originalni pristupe iz ove teze. Dalje je fokus bio na istraživanju, dizajniranju i implementaciji sistema za unutrašnje pozicioniranje (BLEMAT), uključujući njegove softverske podsisteme (kolekcije softverskih komponenti) za: pozicioniranje

u zatvorenom prostoru, detekciju zauzeća prostorija, vizualizaciju, otkrivanje i predviđanje obrazaca kretanja, geofencing, vizualizaciju i analizu društvene dinamike i detekciju rasporeda prostorija unutrašnjeg prostora.

Kada je u pitanju modeliranje i analiza obrazaca kretanja predstavljenih grafovima (pogledati Sekciju 6.1), kao deo budućeg rada, istražiće se probabilistički GED pristupi i ugraditi u BLEMAT za mogućnost obrade podataka u skoro realnom vremenu. Istražiće se i druge mogućnosti povećanja performansi za izračunavanje sličnosti i modeliranje u vezi s grafovima, kao što su agregacija i kompresija grafova.

Kada je u pitanju otkrivanje zauzetosti prostorija unutrašnjeg prostora, predviđanje i analitika nad podacima o zauzetosti (pogledati Sekciju 6.2), proučiće se dodatno mogućnosti predviđanja i pretraživanja obrazaca zauzetosti na prikupljenim Wi-Fi podacima. Stalnim prikupljanjem i obrađivanjem podataka o upotrebi Wi-Fi mreže može se pratiti u realnom vremenu kakavi su trendov korišćenja, te automatski prilagođavati strategije na nivou pametne zgrade tim trendovima i obrascima. Konačno, analiziraće se računski manje zahtevni (eng. lightweight) procesi za agregiranje podataka o Bluetooth pozicioniranju i korišćenju Wi-Fi mreže. Stvaranjem agregiranih skupova podataka za analitičke zadatke u BLEMAT-u, tehnike detekcije zauzetosti, predviđanja i pretraživanja obrazaca pomenute u ovoj tezi će biti svojstveno bolje.

Da bi se pokazalo da zaključci iz ove teze mogu biti ekstrapolirani u većem unutrašnjem prostoru, kao deo budućeg rada na podsistemu GEMAT (pogledati Sekciju 6.3) biće vršeni eksperimenti sa dužim periodima emulacije. Dalje, iako je podrška za definisanje vremenskih virtuelnih granica veza između virtuelnih granica omogućena u GEMAT-u, analiziraće se dublji teorijski model koji bi predstavio jezik specifičan za geofencing domen (eng. Domain Specific Language – DSL) za definisanje sofisticiranih virtuelnih granica i programske šeme zadužene za prevođenje specifikacija virtuelnih granica iz DSL u programski jezik po izboru. I na kraju, budući rad uključivaće poboljšanje trenutnih mogućnosti vizuelizne specifikacije vremenskih virtielnih granica i veza između njih.

Kada je u pitanju otkrivanje rasporeda prostorija unutrašnjeg prostora (pogledati Sekciju 6.4) budući rad će uključivati modeliranje, dizajniranje i implementaciju okvira za otkrivanje prepreka koji počiva i na algoritamima za otkrivanje smetnji signala i na aproksimaciji rasporeda prostorija.

Konačno, kao deo budućeg rada na grafovskom modeliranju društvenih mreža i odnosa (pogledati Sekciju 6.5) fokus će biti na poboljšanju pristupa za praćenje evolucije društvenih zajednica. Konkretno radiće se na načinima za propagaciju identiteta zajednice kroz događaje širenja, spajanje i raspuštanja zajednica, kao najvažnijih događaja u životnom ciklusu zajednica. Istražiće se relevantni pristupi, utemeljeni na veštačkoj inteligenciji, za predviđanje konzistentnosti društvenih

odnosa i predviđanje spajanja/raspuštanja društvenih zajednica, zajedno sa transformacijama grafova.

## Kratka Biografija

Saša Pešić rođen je 11.12.1992. godine u Novom Sadu. Osnovne akademske studije Informatike, smer Informacione Tehnologije, upisuje 2011. godine, na Departmanu za Matematiku i Informatiku, Prirodno-matematičkog fakulteta, Univerziteta u Novom Sadu. Diplomski rad koji nosi naziv „Nerelacione baze podataka, manipulacija podacima i generisanje izveštaja“ brani 2015. godine, a studije završava sa prosekom 9.71/10. Master akademske studije Informatike, smer Softversko Inženjerstvo, upisuje 2015. godine, na istoj visokoškolskoj instituciji. Master rad koji nosi naziv „Uporedna analiza objektno-orijentisanog i funkcionalnog mapiranja“ brani 2016. godine, a studije završava sa prosekom 9.87/10. Doktorske akademske studije upisuje 2016. godine na istoj visokoškolskoj instituciji. Prvu i drugu godinu doktorskih studija završava sa prosekom 10/10. Trenutno je student treće godine doktorskih studija. Saša Pešić je izabran u zvanje asistenta na Departmanu za Matematiku i Informatiku, Prirodno-matematičkog fakulteta, Univerziteta u Novom Sadu, za užu naučnu oblast Računarske Nauke 2017. godine. Kao direktor razvoja za softver radi u VizLore Labs fondaciji u Novom Sadu i kompaniji VizLore LLC u Scottsdale, Arizona, SAD. Saša Pešić je gostujući istraživač i predavač na dva univerziteta: Arizona State University, School of Computing, Informatics, and Decision Systems Engineering, u Blockchain Research Lab (Tempe, Arizona, SAD) i Khalifa University, Department of Industrial and Systems Engineering, u Supply Chain and Operations Research Lab (Abu Dabi, UAE). U svom načno-istraživačkom radu bavi se visokodistribuiranim Internet of Things i edge computing sistemima, analizi njihove robustnosti, bezbednosti i stabilnosti. U okviru svoje doktorske teze, bavi se dizajnom, istraživanjem i implementacijom naprednog sistema za unutrašnje pozicioniranje pod nazivom BLEMAT. Poslednje, bavi se DLT tehnologijama i njihovom interdiscplinarnom primenom u domenima upravljanja električnim mrežama (električne mreže, mikrogridovi), bezbednosti IoT sistema, i osiguranja. Trenutno aktivno radi na dva Horizon2020 istraživačka projekta. U protekle dve godine radio je još i na: PhasmaFOOD, Vicinity, AgileIoT i SymbIote Horizon2020 istraživačkim projektima. Autor/koautor je 13 naučnih radova od kojih je 5 objavljeno u uglednim naučnim časopisima.

## Short Biography

Saša Pešić (December 11, 1992) is a PhD student at the Department of Mathematics and Informatics, Faculty of Sciences, University of Novi Sad, Serbia (generation 2016). He has acquired both Bachelor and Masters degrees at the same academic institutions, graduating with highest honors. He has been a Teaching Assistant at the Department of Mathematics and Informatics, Faculty of Science, since 2017. Since then he has conducted exercises in following courses: Introduction to programming, Object-oriented programming, Data structures and algorithms, Machine Learning and Computer hardware organization. Saša is the director of software development for VizLore Labs Foundation (Novi Sad, Serbia) and Vizlore LLC (Scottsdale, Arizona, USA). Furthermore,

Saša Pešić is a visiting researcher/consultant at two universities: Arizona State University, School of Computing, Informatics, and Decision Systems Engineering, at the Blockchain Research Lab (Tempe, Arizona, USA) and Khalifa University, Supply Chain and Operations Research Lab (Abu Dhabi, UAE). In his research work, he deals with highly distributed indoor positioning Internet of Things and edge computing systems, analyzing their robustness, security, operating capacity and stability. His research interests also include distributed ledger technologies and their interdisciplinary application in the domains of energy, finance, security of IoT systems, and peer-to-peer insurance. He is the author/co-author of 13 scientific papers (5 in respective scientific journals). He is actively working on two Horizon 2020 research projects, and in the past two years he has worked on Vicinity, AgileIoT, SymbIote and PhasmaFood Horizon 2020 projects.

**Univerzitet u Novom Sadu**
**Prirodno-matematički fakultet**
**Ključna dokumentacijska informacija**

| | |
|---|---|
| RBR<br>Identifikacioni broj:<br>IBR | |
| Tip dokumentacije:<br>TD | Monografska dokumentacija |
| Tip zapisa:<br>TZ | Tekstualni štampani materijal |
| Vrsta rada:<br>VR | Doktorska disertacija |
| Autor:<br>AU | Saša Pešić |
| Mentor:<br>MN | dr Miloš Radovanović |
| Naslov rada:<br>NR | Napredna (edge computing) Softverska Arhitektura za Upravljanje Resursima i Unutrašnje Pozicioniranje |
| Jezik publikacije:<br>JP | engleski |
| Jezik izvoda:<br>JI | srpski/engleski |
| Zemlja publikovanja:<br>ZP | Srbija |
| Uže geografsko područje:<br>UGP | Vojvodina |
| Godina:<br>GO | 2020 |
| Izdavač:<br>IZ | autorski reprint |
| Mesto i adresa:<br>MA | Novi Sad, Trg D. Obradovića 4 |
| Fizički opis rada:<br>(broj poglavlja/strana/lit. citata/tabela/slika/grafika/priloga)<br>FO | 7/198/219/6/55/0/0 |

Naučna oblast: Računarske nauke
NO
Naučna disciplina: Internet Stvari
ND Unutrašnje pozicioniranje
Predmetna odrednica/
Ključne reči: Internet stvari,
računanje na ivici mreže,
unutrašnje pozicioniranje,
mašinsko učenje

PO
UDK
Čuva se:
ČU
Važna napomena:
VN

Izvod: Deo I teze ima je za cilj da rasvetli IoT i edge computing računarske sisteme i prateće računarske paradigme softverskih arhitektura, njihovu definiciju, područja primene i slučajeve uobičajene upotrebe, kao i operativne, poslovne, ekonomske, i socijalne izazove i koristi. Teza ilustruje savremene potrebe i zahtevi u izgradnji IoT sistema i najsavremeniji pristupi u njihovom dizajniranju. Raspravlja se o temama bezbednosti i privatnosti u IoT i edge computing računarskim sistemima. Kao još jedan glavni zadatak, teza je obuhvata istraživanje, dizajn i implementaciju softverske arhitekture za upravljanje resursima zasnovanim na MQTT komunikacionom protokolu za edge computing računarske sisteme koja se bavi: upravljanjem resursima, detekcijom prestanka rada upravljačkih algoritama i administracijom primopredaje tj. transporta upravljačkih algoritama, i logičkim i fizičkim balansiranjem i zaštitom radnog opterećenja sistema. Diskutuju su savremeni zahtevi za takve softverske arhitekture, trenutni pristupi. Na kraju, prikazuje se rešenje sa minimalnim troškovima implementacije i komunikacije.

Deo II teze ima za cilj da objasni sisteme za unutrašnje pozicioniranje, njihovu definiciju, vrste primene, najčešće korišćene tehnike pozicioniranja, područja primene i uobičajene slučajeve upotrebe, kao i operativne, poslovne, ekonomske, i socijalne izazove i koristi. Posebno se diskutuje o dizajniranju ovakvih sistema za tipičnu IoT infrastrukturu. Nudi se uvid u savremene zahteve sisteme za unutrašnje pozicioniranje, trenutne pristupe u rešavanju istih, i naglašeni su originalni pristupe iz ove teze.
Dalje je fokus na istraživanju, dizajniranju i implementaciji sistema za unutrašnje pozicioniranje (BLEMAT), uključujući njegove softverske podsisteme (kolekcije softverskih komponenti) za: pozicioniranje u zatvorenom prostoru, detekciju zauzeća prostorija, vizualizaciju, otkrivanje i predviđanje obrazaca kretanja, geofencing, vizualizaciju i analizu društvene dinamike i detekciju rasporeda prostorija unutrašnjeg prostora.

IZ
Datum prihvatanja teme od strane NN veća: 20. jul. 2020
Datum prihvatanja teme od strane Senata: 20. sep. 2020
DP
Datum odbrane:
DO
Članovi komisije:
(Naučni stepen/ime i prezime/zvanje/fakultet)
KO
Predsednik: dr Mirjana Ivanović, redovni profesor, Prirodno-matematički fakultet, Univerzitet u Novom Sadu
Mentor: dr Miloš Radovanović, vanredni profesor, Prirodno-matematički fakultet, Univerzitet u Novom Sadu
Član: dr Vladimir Kurbalija, vanredni profesor, Prirodno-matematički fakultet, Univerzitet u Novom Sadu
Član: dr Miloš Savić, docent, Prirodno-matematički fakultet, Univerzitet u Novom Sadu
Član: dr Dragan Bošković, research professor, Arizona State University, Tempe, Arizona, USA

**University of Novi Sad**
**Faculty of Science**
**Key Words Documentation**

| | |
|---|---|
| Accession number:<br>NO | |
| Identification number:<br>INO | |
| Document type:<br>DT | Monograph documentation |
| Type of record:<br>TR | Textual printed material |
| Contents code:<br>CC | Doctoral dissertation |
| Author:<br>AU | Saša Pešić |
| Advisor:<br>MN | Dr. Miloš Radovanović |
| Title:<br>TI | An advanced resource management and indoor positioning edge computing architecture |
| Language of text:<br>LT | English |
| Language of abstract<br>LA | Serbian/English |
| Country of publication:<br>CP | Serbia |
| Locality of publication:<br>LP | Vojvodina |
| Publication year:<br>PY | 2020 |
| Publisher:<br>PU | Author's reprint |
| Publ. place:<br>PP | Novi Sad, Trg D. Obradovića 4 |
| Physical description:<br>(no. of chapters/pages/bib. refs/tables/figures/graphs/appendices) | 7/198/219/6/55/0/0 |

PO
Scientific field: Computer Science
SF
Scientific discipline: Internet of Things, Indoor Positioning
SD
Subject/Key words: Internet of Things, edge computing, indoor positioning, machine learning
SKW
UC
Holding data:
HD
Note:
N

Abstract: In Part I, this thesis aims to shed light on IoT and edge computing systems and accompanying computing and architectural paradigms, their definition, areas of application, and common use-cases, as well as operational, business, economical, social challenges and benefits. It illustrates modern needs and requests in building IoT systems and current State-of-The-Art (SoTA) approaches to designing them. Additionally, it discusses the security and privacy topics of IoT and edge computing systems. It also encompasses research, design, and implementation of an MQTT-based Resource Management Framework for Edge Computing systems that handle: resource management, failover detection and handover administration, logical and physical workload balancing and protection, and monitoring of physical and logical system resources designed for a real-world IoT platform. The thesis offers insights into modern requests for such frameworks, current SoTA approaches, and offer a solution in the form of a software framework, with minimal implementation and communication overhead.

In Part II, the thesis elaborates on IPS, their definition, deployment types, commonly used positioning techniques, areas of application, and common use-cases, as well as operational, business, economic, social challenges, and benefits. It specifically discusses designing IPS for the typical IoT infrastructure. It offers insights to modern IPS requests, current SoTA in solving them, and underline original approaches from this thesis.

It elaborates on the research, design and authors' implementation of an IPS for the IoT – Bluetooth Low Energy Microlocation Asset Tracking (BLEMAT), including its software engines (collections of software components) for: indoor positioning, occupancy detection, visualization, pattern discovery and prediction, geofencing, movement pattern detection, visualization, discovery and prediction, social dynamics analysis, and indoor floor plan layout detection.

AB

Accepted by Scientific Board on: 20. jul. 2020
Accepted by Senate on: 20. sep. 2020

AS

Defended:

DE

Dissertation Defense Board:

(Degree/first and last name/title/faculty)

DB

| | |
|---|---|
| President: | Dr. Mirjana Ivanovic, full professor, Faculty of Science, University of Novi Sad |
| Advisor: | Dr. Milos Radovanovic, associate professor, Faculty of Science, University of Novi Sad |
| Member: | Dr. Vladimir Kurbalija, associate professor, Faculty of Science, University of Novi Sad |
| Member: | Dr. Milos Savic, assistant professor, Faculty of Science, University of Novi Sad |
| Member: | Dr. Dragan Boskovic, research professor, Arizona State University, Tempe, Arizona, USA |

План третмана података

| **Назив пројекта/истраживања** |
| --- |
| Докторска дисертација: An advanced resource management and indoor positioning edge computing architecture |
| **Назив институције/институција у оквиру којих се спроводи истраживање** |
| а) Универзитет у Новом Саду, Природно-математички факултет<br>б) VizLore Labs Фондација, Нови Сад |
| **Назив програма у оквиру ког се реализује истраживање** |
| Докторске академске студије (област: информатика) |
| **1. Опис података** |

*1.1* Врста студије

*Укратко описати тип студије у оквиру које се подаци прикупљају*

Тип студије у оквиру које су се прикупљали подаци је докторска дисертација.

1.2 Врсте података

**а) квантитативни**

**б) квалитативни**

1.3. Начин прикупљања података

а) анкете, упитници, тестови

б) клиничке процене, медицински записи, електронски здравствени записи

в) генотипови: навести врсту _____

г) административни подаци: навести врсту _____

д) узорци ткива: навести врсту_____

ђ) снимци, фотографије: навести врсту_____

е) текст, навести врсту _____

ж) мапа, навести врсту _____

**з) остало:** прикупљани су подаци о локацији људи и електронских уређаја на две физичке локације. Подаци су прикупљани на рачунарима и другим електронским компонентама. Подаци су измењени пре коришћења у истраживању (идентитет људи сакривен, MAC адресе уређаја промењене).

1.3 Формат података, употребљене скале, количина података

Временске серије текстуалних типова података: *Java Script Object Notation (JSON), CSV* и *plain text* података о локацијама. Приликом истраживања прикупљено је ~2 милиона дигиталних инстанци података о локацијама.

1.3.1 Употребљени софтвер и формат датотеке:

а) Excel фајл, датотека _____

b) SPSS фајл, датотека _____

c) PDF фајл, датотека _____

**d) Текст фајл, серија датотека у горе наведеном формату**

e) JPG фајл, датотека _____

f) Остало, датотека _____

1.3.2. Број записа (код квантитативних података)

а) број варијабли: ~2 милиона

б) број мерења (испитаника, процена, снимака и сл.): 5 мерења

1.3.3. Поновљена мерења

а) да

**б) не**

Уколико је одговор да, одговорити на следећа питања:

а) временски размак између поновљених мера је _____

б) варијабле које се више пута мере односе се на _____

в) нове верзије фајлова који садрже поновљена мерења су именоване као _____

Напомене: Подаци су прикупљани и анализирани као део континуалне временске серије. Стога, поновна мерења нису била предвиђена нити неопходна.

*Да ли формати и софтвер омогућавају дељење и дугорочну валидност података?*

***а) Да***

*б) Не*

*Ако је одговор не, образложити* _____

## 2. Прикупљање података

2.1 Методологија за прикупљање/генерисање података

2.1.1. У оквиру ког истраживачког нацрта су подаци прикупљени?

а) експеримент, навести тип _____

б) корелационо истраживање, навести тип _____

ц) анализа текста, навести тип _____

**д) остало**: Подаци су прикупљани алгоритамским анализирањем спектралне слике Bluetooth и WiFi сигнала у посматраним физичким локацијама.

*2.1.2 Навести врсте мерних инструмената или стандарде података специфичних за одређену научну дисциплину (ако постоје).*

Специфични стандарди и мерни инструменти нису били укључени у процес прикупљања података.

2.2 Квалитет података и стандарди

2.2.1. Третман недостајућих података

а) Да ли матрица садржи недостајуће податке? Да **Не**

Ако је одговор да, одговорити на следећа питања:

а) Колики је број недостајућих података? _____

б) Да ли се кориснику матрице препоручује замена недостајућих података? Да Не

в) Ако је одговор да, навести сугестије за третман замене недостајућих података

_____

2.2.2. На који начин је контролисан квалитет података? Описати

Пре анализе подаци су очишћени и инстанце које су по правилима прикупљања неправилне (прикупљене вредности су ван дозвољених граница) елиминисане су пре пристању научно-истраживачком процесу.

2.2.3. На који начин је извршена контрола уноса података у матрицу?

Контрола уноса била је регулисана програмским кодом у оквиру рачунарских компоненти које су била задужене за прикупљање података. Подаци нису захтевали специфичну контролу уноса током прикупљања.

## 3. Третман података и пратећа документација

3.1. Третман и чување података

*3.1.*1. Подаци ће бити депоновани у **Zenodo online** репозиторијум.

3.1.2. URL адреса: https://zenodo.org/record/3635271

3.1.3. DOI: http://doi.org/10.5281/zenodo.3635271

3.1.4. Да ли ће подаци бити у отвореном приступу?

**а) Да**

б) Да, али после ембарга који ће трајати до _____

в) Не

Ако је одговор не, навести разлог _____

3.1.5. Подаци неће бити депоновани у репозиторијум, али ће бити чувани.

Није примењиво.

3.2 Метаподаци и документација података

3.2.1. Који стандард за метаподатке ће бити примењен?

Стандард за метаподатке који је примењен наметнут је од стране платформе на којој су подаци објављени – Zenodo[1]. Zenodo метаподаци усклађени су са DataCite[2] минималном схемом метаподатака и препорученим условима, уз неколико додатних обогаћења.

[1] https://about.zenodo.org/principles/
[2] https://schema.datacite.org/

3.2.2. Навести метаподатке на основу којих су подаци депоновани у репозиторијум.

<?xml version='1.0' encoding='utf-8'?>

<resource xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://datacite.org/schema/kernel-4" xsi:schemaLocation="http://datacite.org/schema/kernel-4 http://schema.datacite.org/meta/kernel-4.1/metadata.xsd">

<identifier identifierType="DOI">10.5281/zenodo.3635271</identifier>

<creators>

<creator>

<creatorName>pesicsasa</creatorName>

</creator>

</creators>

<titles>

<title>pesicsasa/BLEMAT_public: BLEMAT positioning datasets - Social relationships</title>

</titles>

<publisher>Zenodo</publisher>

<publicationYear>2020</publicationYear>

<dates>

<date dateType="Issued">2020-02-04</date>

</dates>

<resourceType resourceTypeGeneral="Software"/>

<alternateIdentifiers>

<alternateIdentifier alternateIdentifierType="url">https://zenodo.org/record/3635271</alternateIdentifier>

</alternateIdentifiers>

<relatedIdentifiers>

<relatedIdentifier relatedIdentifierType="URL" relationType="IsSupplementTo">https://github.com/pesicsasa/BLEMAT_public/tree/v2.0.0</relatedIdentifier>

<relatedIdentifier relatedIdentifierType="DOI" relationType="IsVersionOf">10.5281/zenodo.3339373</relatedIdentifier>

</relatedIdentifiers>

<version>v2.0.0</version>

<rightsList>

<rights rightsURI="info:eu-repo/semantics/openAccess">Open Access</rights>

</rightsList>

<descriptions>

<description descriptionType="Abstract"><p>The new positioning datasets are published as a support for the paper filled for submission in the Simulation Modelling Practice and Theory journal.</p></description>

</descriptions>

</resource>

3.3 Стратегија и стандарди за чување података

3.3.1. До ког периода ће подаци бити чувани у репозиторијуму?

Подаци ће се чувати заувек

3.3.2. Да ли ће подаци бити депоновани под шифром? Да **Не**

3.3.3. Да ли ће шифра бити доступна одређеном кругу истраживача? Да **Не**

3.3.4. Да ли се подаци морају уклонити из отвореног приступа после извесног времена? Да **Не**

## 4. Безбедност података и заштита поверљивих информација

Овај одељак МОРА бити попуњен ако ваши подаци укључују личне податке који се односе на учеснике у истраживању. За друга истраживања треба такође размотрити заштиту и сигурност података.

4.1 Формални стандарди за сигурност информација/података

Истраживачи који спроводе испитивања с људима морају да се придржавају Закона о заштити података о личности (*https://www.paragraf.rs/propisi/zakon_o_zastiti_podataka_o_licnosti.html*) и одговарајућег институционалног кодекса о академском интегритету.

4.1.2. Да ли је истраживање одобрено од стране етичке комисије? Да **Не**

4.1.2. Да ли подаци укључују личне податке учесника у истраживању? **Да** Не

Ако је одговор да, наведите на који начин сте осигурали поверљивост и сигурност информација везаних за испитанике:

а) Подаци нису у отвореном приступу

**б) Подаци су анонимизирани**

ц) Остало, навести шта

Лични подаци односе се пре свега на историјске податке о 3Д локацији људи/електронских уређаја у оквиру одређеног физичког простора (стамбене или пословне јединице). Лични подаци о људима нису прикупљани нити навођени. Физички идентификатори електронских уређаја су измењени. Локација физичких простора где су прикупљани подаци никада нису обајвљени нити навођени у докторстој дисертацији.

## 5. Доступност података

5.1. Подаци ће бити

**а) јавно доступни**

б) доступни само уском кругу истраживача у одређеној научној области

ц) затворени

5.4. Навести лиценцу под којом ће прикупљени подаци бити архивирани.

Ауторство – делити под истим условима.

## 6. Улоге и одговорност

*6.1. Навести име и презиме и мејл адресу власника (аутора) података*

Саша Пешић, sasa.pesic@dmi.uns.ac.rs

*6.2. Навести име и презиме и мејл адресу особе која одржава матрицу с подацима*

Саша Пешић, sasa.pesic@dmi.uns.ac.rs

*6.3. Навести име и презиме и мејл адресу особе која омогућује приступ подацима другим истраживачима*

Подаци су јавно доступни.

У Новом Саду

17.10.2020

Саша Пешић