CORA Cork Open Research Archive
Cartlann Taighde Oscailte Chorcaí

| | |
|---|---|
| Title | Detecting targeted interference in the Internet of Things |
| Authors | Morillo, Gabriela |
| Publication date | 2024 |
| Original Citation | Morillo Fuetala, D. G. 2024. Detecting targeted interference in the Internet of Things. PhD Thesis, University College Cork. |
| Type of publication | Doctoral thesis |
| Rights | © 2024, Diana Gabriela Morillo Fueltala. - https://creativecommons.org/licenses/by/4.0/ |
| Download date | 2026-11-06 06:10:47 |
| Item downloaded from | https://hdl.handle.net/10468/16391 |



OLLSCOIL NA HÉIREANN, CORCAIGH

**NATIONAL UNIVERSITY OF IRELAND, CORK**




# DETECTING TARGETED INTERFERENCE IN THE INTERNET OF THINGS

THESIS PRESENTED BY

DIANA GABRIELA MORILLO FUELTALA, MSC.

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

UNIVERSITY COLLEGE CORK

COLLEGE OF SCIENCE, ENGINEERING AND FOOD SCIENCE

SCHOOL OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

SFI CENTRE FOR RESEARCH TRAINING IN ADVANCED NETWORKS FOR SUSTAINABLE SOCIETIES, ADVANCE CRT

HEAD OF DEPARTMENT: PROFESSOR UTZ ROEDIG

SUPERVISORS: PROFESSOR UTZ ROEDIG
PROFESSOR DIRK PESCH

JULY, 2024

# Contents

**List of Figures** vii

**List of Tables** viii

**Declaration** ix

**Abstract** x

**Acknowledgements** xiii

**1 Introduction** 1

1.1 Motivation . . . . . 1

1.2 Contributions . . . . . 4

1.3 Publications . . . . . 8

1.4 Thesis Structure . . . . . 8

**2 Background** 12

2.1 Security . . . . . 12

2.1.1 Security for the Internet of Things . . . . . 13

2.1.1.1 IoT Security Threats and Vulnerabilities . . . . . 13

2.1.1.2 Security Challenges Unique to IoT . . . . . 15

2.1.1.3 Security Techniques for IoT Networks . . . . . 17

2.2 IoT Communications . . . . . 22

2.2.1 Overview of IoT Technologies . . . . . 23

2.2.2 Narrowband-Internet of Things (NB-IoT) . . . . . 27

2.2.2.1 NB-IoT Design Principles . . . . . 27

2.2.2.2 NB-IoT Operation Mode . . . . . 29

2.2.2.3 NB-IoT Physical Layer . . . . . 31

2.2.2.4 Physical Channels and Signals . . . 33
2.2.2.5 NB-IoT Communication . . . 34
2.2.3 IEEE 802.15.4 . . . 36
2.2.3.1 IEEE 802.15.4 Architecture . . . 37
2.2.3.2 IEEE 802.15.4 Operation Modes . . . 38
2.2.3.3 Guaranteed Time Slot (GTS) . . . 39
2.2.3.4 IEEE 802.15.4 GTS Communication . . . 41
2.3 Interference . . . 42
2.3.1 General . . . 42
2.3.2 Jamming Interference . . . 46
2.3.2.1 Friendly Jamming . . . 47
2.3.2.2 Malicious Jamming . . . 48
**3 IoT Monitoring to Improve Security and Resilience 52**
3.1 Introduction . . . 52
3.2 The Need for IoT Monitoring . . . 55
3.3 Intrusion Detection System (IDS) . . . 57
3.3.1 Architecture Type . . . 58
3.3.2 Analysis Method . . . 59
3.3.3 Monitoring Attacks . . . 60
3.3.4 Performance Metric . . . 61
3.3.5 IoT IDS Examples . . . 61
3.4 Localisation . . . 67
3.4.1 Architecture Type . . . 68
3.4.2 Measurement . . . 70
3.4.3 Analysis Method . . . 71
3.4.4 Monitoring Attacks . . . 72
3.4.5 Performance Metric . . . 73
3.4.6 IoT Localisation Examples . . . 74
3.5 Authentication . . . 78

3.5.1 Architecture Type . . . 78
3.5.2 Measurements . . . 80
3.5.3 Analysis Method . . . 81
3.5.4 Monitoring Attacks . . . 81
3.5.5 Performance Metric . . . 82
3.5.6 IoT Authentication Examples . . . 82
3.6 Interference Monitoring . . . 87
3.6.1 Architecture Type . . . 88
3.6.2 Measurements . . . 90
3.6.3 Analysis Method . . . 91
3.6.4 Performance Metric . . . 92
3.6.5 IoT Interference Examples . . . 93
3.7 Performance Monitoring . . . 98
3.7.1 Architecture Type . . . 99
3.7.2 Measurements . . . 100
3.7.3 Analysis Method . . . 101
3.7.4 Performance Metric . . . 102
3.7.5 Performance Monitoring Examples . . . 102
3.8 Related Work . . . 106
3.9 Discussion . . . 109
3.9.1 Lessons Learned . . . 109
**4 Jamming of NB-IoT Synchronisation Signals 112**
4.1 Introduction . . . 112
4.2 The UE and eNodeB Synchronisation Process . . . 114
4.3 Jamming the NB-IoT Synchronisation Process . . . 115
4.4 Jamming Evaluation . . . 116
4.5 Conclusions . . . 118
**5 Detecting Targeted Interference in NB-IoT 119**
5.1 Introduction . . . 119

5.2 Related Work . . . 121
5.3 Background . . . 124
5.3.1 NB-IoT PHY Layer . . . 124
5.3.2 Threat Model and Targeted Interference . . . 125
5.4 Targeted Interference Detection . . . 127
5.4.1 Detector Design . . . 128
5.4.2 Loss Rate Adjustment . . . 129
5.5 Evaluation Setup . . . 129
5.5.1 Simulation Environment . . . 129
5.5.2 Communication Scenario . . . 131
5.6 Results . . . 132
5.6.1 Loss Rate Adjustment . . . 132
5.6.2 Detection Capability . . . 133
5.7 Conclusion . . . 137
**6 Technology Independent Interference Detector 138**
6.1 Introduction . . . 138
6.2 Related Work . . . 140
6.3 Malicious Interference . . . 142
6.3.1 IoT Communications . . . 142
6.3.2 Threat Model . . . 144
6.4 IoT Case Studies and Jamming Attacks . . . 145
6.4.1 Narrowband-Internet of Things (NB-IoT) . . . 145
6.4.2 IEEE 802.15.4 . . . 146
6.5 Technology Independent Targeted Interference Detection . . . 148
6.5.1 Concept . . . 148
6.5.2 Design Principles . . . 149
6.5.3 Detector Architecture . . . 151
6.5.3.1 Adapter . . . 151
6.5.3.2 Detector . . . 151

6.5.3.3 Comprehensive Analysis for the Detector . . . 153
6.6 Evaluation Setup . . . 157
6.6.1 Simulation Environment and Scenarios . . . 157
6.6.1.1 Narrowband-Internet of Things (NB-IoT) . . . 157
6.6.1.2 IEEE 802.15.4 . . . 158
6.6.2 Noise and Attacks . . . 158
6.7 Results . . . 159
6.7.1 Loss Rate Detector . . . 160
6.7.2 Pattern Detector . . . 161
6.7.3 Fused Detector . . . 163
6.7.3.1 Fused Detector Logic . . . 163
6.7.3.2 Results . . . 164
6.8 Conclusion . . . 165
**7 Conclusion** **166**
7.1 Conclusions . . . 166
7.2 Future Work . . . 169
**References** . . . 171
**Acronyms** . . . i

# List of Figures


2.1 IoT Architecture . . . . . 15
2.2 IoT General Classification . . . . . 25
2.3 NB-IoT Standalone Deployment using Refarmed GSM Spectrum. . . . . 29
2.4 NB-IoT Inband and Guardband Operation Modes . . . . . 30
2.5 NB-IoT Frame Structure . . . . . 31
2.6 Physical Resource Block in NB-IoT Downlink . . . . . 32
2.7 Time-Multiplexing of Downlink Physical Channels . . . . . 33
2.8 NB-IoT Random Access Procedure. Initial steps of a User Equipment (UE) to establish communication with an eNodeB. . . . . 36
2.9 LR-WPAN device architecture [1] . . . . . 37
2.10 Superframe Structure . . . . . 38
2.11 Exchange Messages in IEEE 802.15.4 GTS Allocation . . . . . 42

3.1 Intrusion Detection System Scheme for IoT . . . . . 58
3.2 Localisation Scheme for IoT . . . . . 69
3.3 Authentication Scheme for IoT . . . . . 79
3.4 Interference Scheme for IoT . . . . . 89
3.5 Performance Monitoring Scheme for IoT . . . . . 99

4.1 NB-IoT Random Access Procedure. Initial steps of a UE to establish communication with an eNodeB. . . . . 115
4.2 Jamming Attack on NB-IoT: (a) Standard Transmission (b) Transmission with Jamming Signal. . . . . 118

5.1 Time-multiplexing of downlink physical channels on NB-IoT . . . . . 125
5.2 Communication Chain at the receiver side of the NB-IoT . . . . . 130
5.3 Loss rate $L$ for continuous background noise (BN). . . . . 132
5.4 Adjusted loss rate $L'$ for continuous background noise (BN). . . . . 133
5.5 Loss rate $L'_0$ and $\widetilde{L'}$ for BA and TA interference (Attack is present). . . . . 134
5.6 Loss rate $L'_0$ and $\widetilde{L'}$ for BN and BT interference (No attack is present). . . 135
5.7 Receiver Operating Characteristic (ROC) curve . . . . . 136

6.1 NB-IoT Selective MIB Attack. . . . . 146
6.2 IEEE 802.15.4 Superframe Structure. . . . . 147
6.3 Detector Design: The sequence $S_n$ is received every $T$ seconds describing PDU loss/reception. The detector decides if a targeted attack is present. . . . . 149
6.4 Receiver Operating Characteristic (ROC) curves for NB-IoT and IEEE 802.15.4 GTS Jamming Attacks using Loss Rate Detector with different $HT$ and $LT$. . . . . 159
6.5 Receiver Operating Characteristic (ROC) curves for NB-IoT Jamming Attacks using Pattern Detector with different $HT$ and $LT$. . . . . 162
6.6 Confusion Matrix. . . . . 164

# List of Tables

2.1 NB-IoT Frame Structure . . . . . 31

5.1 Physical Channels and Signals in NB-IoT . . . . . 126

# Declaration

This is to certify that the work I am submitting is my own and has not been submitted for another degree, either at University College Cork or elsewhere. All external references and sources are clearly acknowledged and identified within the contents. I have read and understood the regulations of University College Cork concerning plagiarism and intellectual property.

Signed: _____ Date: _____

Gabriela Morillo F.

# Abstract

This thesis investigates targeted jamming interference detection to enhance security in the Internet of Things (IoT) infrastructures. The study starts by assessing the critical role of IoT system monitoring in securing large networks, emphasising the need for automated solutions to detect and mitigate threats, ensuring continuous and reliable operations. This provided insight into how interference monitoring solutions should be implemented. The development of this kind of detector is important as naturally occurring interference requires a different response than targeted interference attacks.

A significant portion of the thesis is dedicated to addressing vulnerabilities in Narrowband-Internet of Things (NB-IoT), a Low Power Wide Area Network (LPWAN) radio technology required for large-scale IoT deployments. Initially, it looks specifically into how interference with NB-IoT synchronisation signals can lead to Denial of Service (DoS) attacks, highlighting the need to prevent and mitigate such vulnerabilities. A novel attack on the initial communication steps is provided in this investigation.

To address these challenges, this work introduces a novel method for detecting targeted interference at the User Equipment (UE) level in NB-IoT networks. Our solution utilises network performance data and subframe loss rates to differentiate between targeted attacks and naturally occurring interference, which is critical as they require different responses.

The costs associated with designing dedicated detectors for each technology, includ-

ing established and upcoming ones, are high. Therefore, we propose a technology-independent approach to detect targeted interference across various IoT networks. This solution, designed to function on resource-constrained IoT devices, analyses packet loss rates and patterns to detect the presence of targeted attacks. This detection technique has been proven through comprehensive assessments using several IoT technologies, including NB-IoT and IEEE 802.15.4 GTS, demonstrating its effectiveness in distinguishing targeted interference from natural interference.

This work advances the state of the art in detecting malicious interference in IoT environments by introducing a technology-independent targeted interference detection method capable of operating on resource-constrained IoT devices. Unlike prior research, which has primarily focused on machine learning IDS or including additional hardware for their solutions, our approach monitors packet loss rates and patterns across different wireless communication technologies (e.g. Narrowband Internet of Things and IEEE 802.15.4) to perform statistical anomaly detection. This is the first research to propose and validate a comprehensive, technology-independent framework that effectively distinguishes between targeted attacks and natural interference, thereby significantly enhancing the security and resilience of heterogeneous IoT deployments.

Overall, our research emphasises the importance of robust monitoring systems and innovative defence mechanisms to safeguard IoT infrastructures against evolving and emerging threats while also contributing valuable insights and tools to enhance the resilience of critical IoT applications.

Con amor, para mis queridos padres,

Fanny y Ernesto

# Acknowledgements

Firstly, I would like to express my gratitude to my supervisor, Professor Utz Roedig, for his unwavering support and encouragement throughout my research and the different academic activities that have arisen over the past years. His guidance was fundamental to the successful completion of this research project. I am taking with me countless lessons learned from each of our meetings. I appreciate the time he devoted to our enlightening discussions, during which I benefited immensely from his vast experience, knowledge, and insights. His passion, dedication, and work ethic are exceptional —all qualities that have inspired and motivated me. Professor Roedig's guidance has been invaluable in steering the direction of my research.

Also, I sincerely thank Professor Dirk Pesch and Professor Jack Stankovic at the University of Virginia. Undoubtedly, their feedback and challenging questions were extremely helpful in ensuring the accuracy and completeness of my research. The numerous learning experiences I have taken from working together are priceless.

I also wish to express my sincere gratitude to my research centre, ADVANCE CRT - SFI Centre for Research Training in Advanced Networks for Sustainable Societies. From day one in 2020 until now, I have had their constant support at every stage of my PhD, definitely a core factor in successfully meeting each milestone of this journey. Special thanks to Luis Gomez de Membrillera and James Duggan for their accurate responses, professionalism and kindness.

My colleagues at ADVANCE CRT and the Cybersecurity Research Group have been a significant part of this journey. Thank them for sharing exciting discussions,

meetings, travels, and incredible experiences in Ireland.

My personal acknowledgements must start with family. I dedicate my work to my beloved parents, Fanny and Ernesto, who have been my greatest inspiration and for their endless love day after day. I am grateful to my dear siblings, nephews, and in-laws for their constant love and support, especially during these past four years. I want to thank my friends in Ecuador, my best friends, for everything they have done for me during this time, words cannot express my gratitude for their support. I feel very fortunate and blessed to have my beautiful family and friends by my side. I can't wait to see you all again in my beloved Ecuador.

Dedico mi trabajo a mis amados padres, Fanny y Ernesto, por ser mi más grande inspiración y por su amor infinito día tras día. Quiero agradecer a mis queridos hermanos, sobrinos y cuñados por tanto amor y el respaldo de siempre, especialmente durante estos últimos cuatro años. Quiero agradecer a mis amigos en Ecuador, mis mejores amigos, no hay palabras para agradecer todo lo que hicieron por mí durante este tiempo. Me siento muy afortunada y bendecida por tener a mi Hermosa Familia y a mis amigos a mi lado, no puedo esperar a verlos de regreso en mi amado Ecuador.

This work has emanated from research conducted with the financial support of the Science Foundation Ireland under Grant numbers 18/CRT/6222 and 13/RC/2077_P2.

# 1 Introduction

## 1.1 Motivation

In recent years, the introduction of wireless sensor networks, particularly within the Internet of Things (IoT) domain, has significantly revolutionised our interaction and communication with the surrounding world. These networks have introduced a wide range of applications across multiple domains, including security and surveillance, environmental monitoring, and control systems. Their remarkable versatility has enabled real-time data collection and remote operation across diverse environments, allowing unprecedented connectivity and automation. However, this technological advancement is accompanied by significant security challenges. Due to their reliance on wireless communications, these networks are intrinsically more vulnerable to cyber attacks and unauthorised access than wired networks. The broadcast nature of wireless communication means that signals can be more easily intercepted or disrupted, exposing the confidentiality, integrity, and availability of transmitted data. To address these vulnerabilities, it is critical to implement robust security measures. Advanced encryption techniques, strong authentication mechanisms, and effective intrusion detection systems stand at the forefront of safeguarding wireless communications. Furthermore, regular security updates and network segmentation are critical for keeping systems resilient to evolving cyber threats. Using machine learning and artificial intelligence (AI) to improve security measures has also demonstrated potential. These solutions offer extensive analytics capabilities, allowing for the detection of unusual patterns that indicate security breaches as well as automatic risk mitigation responses.

Specifically, the integration of IoT devices into our daily lives and critical infrastructure implementations needs a comprehensive and dynamic security strategy. The IoT ecosystem, characterised by its heterogeneity and massive scale, presents unique security challenges. The diversified and distributed nature of IoT devices, along with resource constraints, challenges the use of standard or traditional security mechanisms. IoT devices, which frequently operate with limited computing capacity and energy supply, cannot directly employ advanced encryption techniques and security protocols used in more resource-capable systems. Furthermore, the vast size and complexity of IoT networks increase the number of potential points of failure, with each connected device serving as a potential entry point for security breaches. The integrity and confidentiality of the massive amounts of sensitive data transmitted across these networks require the implementation of a strong security mechanism tailored to the IoT's unique and specific requirements. In the following paragraphs, we explain why securing the IoT systems requires a multidimensional approach that takes into account the unique characteristics and challenges of IoT and how, by integrating advanced security technologies with innovative approaches, we can safeguard these networks from present and future threats, guaranteeing IoT systems' safe and reliable operation.

The constant growth of IoT applications and their pervasive integration into every aspect of our daily lives has led to an era where the security of wireless IoT communication systems is paramount. Within this context, detecting and mitigating jamming attacks in IoT communication networks have emerged as critical challenges. Our motivation derives from a strong desire to understand and prevent one of the most insidious types of these attacks: targeted malicious interference, which, if successful, can disrupt entire IoT communication networks. Unlike more straightforward jamming efforts that aim to flood the spectrum and disrupt communication through brute force, the attacks we are concerned with are more intelligent, attempting to undermine the IoT communication protocol itself, necessitating a sophisticated approach to detection and defence.

Our research is based on identifying and characterising these intelligent interfer-

ence attacks, particularly those that deliberately seek to prevent communication at various phases of the communication process. This interest is driven by the observation that, whereas blanket jamming is well-known and relatively easy to detect due to its explicit nature, more selective attacks provide a threat that requires a more refined approach to detection and mitigation. In our research, we study a variety of interference attacks, ranging from those that prevent the establishment of communication channels to those that enable communication to begin but then degrade its quality stealthily.

In order to address this challenge, we first investigate existing monitoring systems designed for a variety of purposes, such as performance, authentication, localisation and others, to gain insights that can be used to develop a monitoring system capable of detecting these malicious interference attacks. This investigation is motivated by critical concerns regarding identifying the best location for monitoring activities (whether at the base station or on the end device itself) and the type of data that should be collected to enable effective detection. After conducting a thorough review of the trade-offs involved in various monitoring systems, we decided to focus the detection system on the end device and use packet loss as a sufficient metric of interference.

Motivated by the need to create a platform-independent system that can identify malicious interference in a way that is not only effective but also feasible and practical to apply in a range of technical scenarios, our research firstly contributes significantly to the area by providing a thorough threat model that describes the precise nature of the interference attempts that we want to identify. This model serves as the foundation for our interference detection system, which is inspired by existing monitoring frameworks but addresses the particular issues given by targeted jamming attacks.

Following this line of investigation, our results confirm the feasibility of building a technology-independent interference detector, thereby marking a significant advancement in the ongoing effort to protect communication networks from malicious interference.

Our research not only sheds light on the types of attacks that threaten the integrity of wireless IoT networks but also gives a framework for designing generic, adaptable solutions to counteract them. In doing so, we contribute to the vital discussion about safeguarding the security and dependability of wireless IoT communication systems in the face of growing threats, particularly within the increasingly interconnected world of the Internet of Things.

## 1.2 Contributions

In this section, we list the novel aspects and contributions of this research and how they relate to the chapters of this thesis.

**An IoT Monitoring Taxonomy to Improve Security and Resilience**
We introduce a comprehensive IoT monitoring taxonomy to identify the commonalities in five different IoT monitoring systems (Chapter 3).

We propose a new *high-level standard classification scheme* for IoT monitoring systems to identify common elements across several systems deployed for different application purposes. Our approach categorises IoT monitoring apps into four levels.

The commonalities are based on four common criteria applied to five monitoring systems. We include architecture type, measurements, analysis methods, and performance metrics. The new classification scheme can be applied initially for Intrusion Detection System (Section 3.3), Localisation Monitoring (Section 3.4), Authentication Monitoring (Section 3.5), Interference Monitoring (Section 3.6) and Performance Monitoring (Section 3.7) systems in IoT. However, future works might consider a proposal for expansion to other applications and services.

**A Jamming Attack of NB-IoT Synchronisation Signals**
We present an effective jamming interference attack on the NB-IoT synchronisation signals - the initial communication steps – that can be used to implement an effective DoS attack (Chapter 4).

Interference with the synchronisation prevents communication and may also allow an attacker to force a device to connect with a specific base station. We consider an intelligent adversary using a jamming device to disrupt NB-IoT communication that targets the initial communication steps of an NB-IoT network to have a maximum impact. Specifically, we describe and investigate how a jammer can interfere with the Narrowband Primary Synchronisation Signal (NPSS) and Narrowband Secondary Synchronisation Signal (NSSS), which are used to initiate the NB-IoT contention-based random-access procedure.

We have shown that it is possible to interfere with the synchronisation signal used by NB-IoT devices (the UE) to establish communication with the base station. Thus, a simple selective jamming device can prevent communication of NB-IoT devices.

Furthermore, our experiments indicate that careful design of the interference signal might enable an attacker to force the UE to recognise a specific Narrowband Physical Cell ID (NB-PCID). Most of this content is published in *P*.1. - [2].

**A Detector for Targeted Interference in NB-IoT Networks**

We propose a novel method of detecting targeted interference in an NB-IoT network at the UE, e.g. a sensor node. We consider an adversary using a jamming device to interfere with NB-IoT communication. Specifically, we assume the attacker observes and interferes with the downlink channel to achieve maximum impact. Network performance data collected at the UE is used to explain the current interference situation. Subframe loss rates within the downlink channel are monitored and used as input for a statistical anomaly detector at the UE (Chapter 5). The specific contributions of our work are:

- *Targeted Interference Detection:* A UE based detection method using subframe loss rates observed within the downlink channel. Loss rates can be provided by the UE decoding pipeline without modification at the UE.
- *Evaluation of Targeted Interference Detection:* We evaluate the performance of the

targeted interference detection mechanism using a Matlab simulation framework. We show that the detector is able to distinguish targeted and non-targeted interference reliably.

- *Optimal Targeted Interference Detection:* We describe the optimal configuration of the detector to balance true positive and false negative detection.

We focus our research on NB-IoT downlink transmission due to potential security vulnerabilities regarding the limited encryption and insecure over-the-air broadcast transmission. For example, the first messages exchanged between the eNodeB and the UE during the initial communication establishment process are in clear text. This practice, while enabling efficient network accessibility and operation, creates significant security risks. The unencrypted nature of these transmissions, along with the broadcast characteristic of wireless communication, provide attackers with valuable network insights and a vast attack surface. This is discussed in more detail in Section 4.4.

Most of these contributions are published in *P*.2. - [3].

**A Technology-Independent Jamming Interference Detector**

In Chapter 6, we introduce a technology-independent method for detecting targeted interference in wireless IoT deployments capable of operating on resource-constrained IoT devices. Packet loss rates and packet loss patterns independent of the specific physical layer technology are analysed to determine if a targeted attack is present. The proposed approach is validated through comprehensive evaluations of two example technologies, Narrowband-Internet of Things (NB-IoT) and IEEE 802.15.4 Guarantee Time Slot (GTS), demonstrating its effectiveness in detecting attacks. Our evaluation shows that the detector can distinguish between targeted interference attacks and the impact of naturally occurring interference. The specific contributions of our work are:

- *Technology Independent Targeted Interference Detector:* We provide a design of a targeted interference detector monitoring *loss rates* and/or *loss patterns* of protocol

elements in a technology-independent way.

- *Example Detection Scenarios:* We show how the detector can be used in two scenarios: NB-IoT and 802.15.4 Guarantee Time Slot (GTS). For both scenarios we describe example attacks and how monitoring is performed.
- *Evaluation:* We provide a comprehensive simulation evaluation of the detector for the two use cases NB-IoT and 802.15.4 GTS.

We use MATLAB to implement our solution. MATLAB stands out as an accurate tool for simulating IoT networks. Its advanced signal processing capabilities provide physical layer built-in functions for modulation schemes (e.g. QPSK, BPSK), channel coding (e.g. turbo codes for NB-IoT), and error correction techniques to enable accurate representation of the radio interface. Also, MATLAB allows a flexible implementation of the entire IoT protocols, from the physical to the application layers. For NB-IoT, this includes the ability to model the Non-Access Stratum (NAS), the Random Access (RA) process, and the Radio Resource Control (RRC) protocols. For IEEE 802.15.4, MATLAB can accurately simulate CSMA-CA mechanisms and superframe structures. Furthermore, MATLAB provides comprehensive toolboxes with functions for designing, analysing, and testing communications systems, including channel models specific to IoT deployments. For large-scale network simulations, MATLAB offers parallelisation and GPU acceleration, enabling capabilities that allow efficient simulation of multiple nodes and complex channel conditions. Hardware integration capabilities support hardware-in-the-loop testing with SDR platforms, bridging the gap between simulation and real-world deployment. Lastly, MATLAB's support for standardised channel models ensures simulations align with industry-accepted standards (e.g. 3GPP standard), further enhancing the reliability and relevance of the simulation results.

Most of this content is published in *P*.3.

## 1.3 Publications

The following publications have been produced as a result of the research outlined in this thesis.

**P.1.** Gabriela Morillo and Utz Roedig: *Jamming of NB-IoT Synchronisation Signals*, in Proceedings of the 26th European Symposium on Research in Computer Security (ESORICS 2021), pp. 1-5. 2021. https://doi.org/10.1007/978-3-030-88428-4.

**P.2.** Gabriela Morillo, Utz Roedig, and Dirk Pesch: *Detecting Targeted Interference in NB-IoT*, in Proceeding of the 2023 IEEE 19th International Conference on Distributed Computing in Smart Systems and the Internet of Things (IEEE DCOSS-IoT 2023), pp. 475-482, 2023, https://doi.org/10.1109/DCOSS-IoT58021.2023.00080.

**P.3.** Gabriela Morillo, Utz Roedig, and John Stankovic: *Technology Independent Targeted Interference Detection for Wireless IoT*, in Proceedings of the 2024 IEEE 4th International Mediterranean Conference on Communications and Networking (IEEE MEDITCOM 2024), July 8 - 11, 2024.

Additional Publication:

**P.4.** Gabriela Morillo and John Cosmas: *LTE SFBC MIMO Transmitter Modelling and Performance Evaluation*, in Proceedings of the Advances in Information and Communication, FICC 2024, Lecture Notes in Networks and Systems, vol 919. Cham: Springer Nature Switzerland, pp. 268-291. Available at: https://doi.org/10.1007/978-3-031-53960-2_18.

## 1.4 Thesis Structure

The framework of this thesis is to provide a comprehensive and coherent explanation of the security aspects applied to Wireless IoT. The remainder of this thesis is organised as follows:

- *Chapter 1: Introduction* – The first part of this chapter presents the motivation behind the proposed research topic. It highlights the importance and relevance of detecting threats as a security mechanism in the context of IoT communications. Furthermore, it emphasizes the security gap identified in our research and outlines the potential impact of the proposed detection solution to provide secure IoT networks. This chapter addresses the contributions of the study, and it concludes with a summary of the thesis structure, providing guidance through the upcoming chapters.

- *Chapter 2: Background* - This chapter describes the background of the study. The first part of this chapter addresses the security aspects relevant to wireless communications with a specific emphasis on IoT communications, clearly states the IoT security threats and vulnerabilities, the challenges unique to IoT, and the security technique usually implemented in IoT networks. The second part of this chapter presents the design of the Narrowband Internet of Things (NB-IoT) as one of the technologies chosen to implement our solution. Initially, it describes the specifications and the design principles of the technology. Later, it focuses on the physical layer details and provides a complete description of the downlink and uplink transmission schemes, channels, and signals. The third part of this chapter covers the IEEE 802.15.4 technology. The chapter concludes with a description of interference in terms of jamming attacks and detection.

- *Chapter 3: IoT Monitoring to Improve Security and Resilience* – This chapter categorises the different IoT monitoring systems to improve security and resilience in IoT communications. It outlines recent implementation examples in different underlying systems and discusses their effectiveness. The analysed systems include intrusion detection systems, localisation, authentication, interference monitoring and performance monitoring. The chapter outlines differences and commonalities amongst existing approaches and identifies research gaps and opportunities.

- *Chapter 4: Jamming of NB-IoT Synchronisation Signals* – This chapter introduces an effective attack to the initial communication steps in an IoT network to generate a maximum impact on it. It describes how interference on the NB-IoT synchronisation signals can be used to implement an effective DoS attack. The chapter clearly describes how interference with the synchronisation prevents communication and may also allow an attacker to force a device to connect with a specific base station.

- *Chapter 5: Detecting Targeted Interference in Narrowband-Internet of Things (NB-IoT)* – The chapter introduces a novel method of detecting targeted interference in a Narrowband-Internet of Things (NB-IoT) network at the User Equipment (UE). It describes how network performance data collected at the UE is used to reason about the current interference situation and how subframe loss rates within the downlink channel are monitored and used as input for a statistical anomaly detector. The chapter includes the evaluation setup, which shows that the detector is able to distinguish targeted interference attacks from the impact of naturally occurring interference in IoT networks.

- *Chapter 6: Technology Independent Interference Detector for Wireless Internet of Things (IoT)* - This chapter extends our previous effort to construct an interference detector for the NB-IoT protocol based on monitoring of loss rates [3]. We describe a novel method for targeted interference detection that can execute on resource-constrained IoT devices and can be used across different wireless technologies. The interference detection is independent from the lower-layer (physical and MAC) specifics of the wireless technology. The detector monitors loss rates and loss sequence of different protocol elements. A targeted interferer will cause loss rates and/or patterns to specific protocol elements while natural interference will impact on all protocol elements the same way. To perform targeted interference detection for a wireless protocol, protocol elements need to be defined, which are then observed technology-independent by the detector. In this work, we show how IoT nodes using two example technologies, NB-IoT and 802.15.4

can perform targeted interference detection.

- *Chapter 7: Conclusion* - The final chapter summarises the research, highlighting the key findings and their implications. It also discusses the limitations of the research and suggests areas for future study.

# 2 | Background

## 2.1 Security

Wireless sensor networks, particularly within the context of the Internet of Things (IoT), have transformed the way we communicate and interact with the world. These networks offer a plethora of applications, ranging from security and surveillance systems to control systems and environmental monitoring. Their versatility is remarkable, enabling real-time data collection and remote operation in diverse settings. Nevertheless, this technological leap has also brought forth significant security challenges. Wireless networks inherently possess greater vulnerability to cyber-attacks and unauthorised access compared to the wired networks. This susceptibility stems from the nature of wireless communication, which relies on airwaves that are more easily intercepted or disrupted. Moreover, IoT nodes are often resource-constrained devices with limited computing power, memory, and energy capacity. These limitations impose significant challenges to implementing computationally intensive security measures, which are typically required to prevent sophisticated cyber threats. As a result, the resource constraints of IoT devices need the development of effective and efficient security solutions without overwhelming these devices' limited capabilities.

To address these vulnerabilities, it is critical to implement robust security mechanisms and techniques. These can include advanced encryption methods to protect data transmission, stringent authentication protocols to verify device and user identities, and intrusion detection systems to promptly identify and respond to potential

threats. Additionally, regular security updates and patches are required to handle emerging threats and keep the system resilient against new forms of attacks. Network segmentation can also play an important role in containing and preventing the spread of breaches by isolating essential areas of the network. Implementing machine learning and AI-driven analytics can also improve threat identification and neutralisation capabilities. These tools can analyse patterns and behaviours, identify anomalies that may suggest a security breach, and automate responses to mitigate risks more effectively.

In summary, wireless IoT brings about remarkable opportunities but also necessitates a heightened focus on security. We can exploit the full potential of wireless sensor networks in the IoT area while protecting against the hazards they bring by taking a comprehensive and dynamic approach to security that incorporates both technological and procedural measures.

### 2.1.1 Security for the Internet of Things

This section aims to discuss various aspects of IoT security, including the common security threats and vulnerabilities, unique security challenges specific to IoT technology, significant security standards and protocols for IoT, existing and emerging security solutions, future trends and challenges in IoT security, and research gaps and opportunities.

#### 2.1.1.1 IoT Security Threats and Vulnerabilities

Regarding the *common security threats and vulnerabilities*, several taxonomies of security attacks in IoT have been proposed. In [4], the authors analyse the communication architecture of an IoT system defined in [5] in order to identify the vulnerabilities and the possible attacks at the different levels of the IoT communication stack (See Fig 2.1). In summary, they consider the following layers: i) the edge layer, which provides the physical (PHY) and media access control (MAC) functionalities for local communications, ii) the access layer, which establishes the connection between the

IoT devices and the standard internet, and iii) the application layer, which handles the data communication at the service level.

- At the **Edge Layer**, IoT devices face significant security threats, particularly channel attacks. These attacks aim to extract sensitive information by capturing valuable information from the analysis of the signal, such as communication timing, power consumption of devices, and statistical analysis for a larger number of traces to extract additional encryption or sensitive information. At this layer, IoT devices are vulnerable to Denial of Services attacks that make resources unavailable for legitimate users, battery draining, jamming the radio communication, and tampering to extract cryptography keys or modify software to camouflage a malicious node as a legitimate one.

- At the **Access Layer**, several attacks can be identified. For example, i) sniffing attacks involving unauthorised interception and eavesdropping on data packets in a network, attackers can gather sensitive information without the knowledge or consent of the parties involved in the communication. ii) Injection of fraudulent packets, where attackers insert harmful or deceptive data packets into the IoT network. These packets can potentially transport malicious payloads, initiate unauthorised actions, or interfere with regular network processes. iii) Routing attacks aim to manipulate the routing of data packets within the network. Adversaries can spoof, redirect, misdirect, or even drop data packets. This can lead to transmitting information to unintended recipients, deliberately manipulating communication routes, or losing data.

- At the **Application Layer**, attacks directly target the software running on IoT devices rather than the communication technology itself. Here are a few examples: i) Integrity manipulation refers to deliberate tampering with the algorithms used in IoT deployments to compromise the integrity of the service application. ii) Tampering attacks, such as those targeting machine learning algorithms, include manipulating the training data or learning process to induce malicious

behaviour. Attackers might force the algorithm to act wrongly or unpredictably This could result in erroneous decision-making, data misclassification, or algorithm exploitation for malicious purposes.

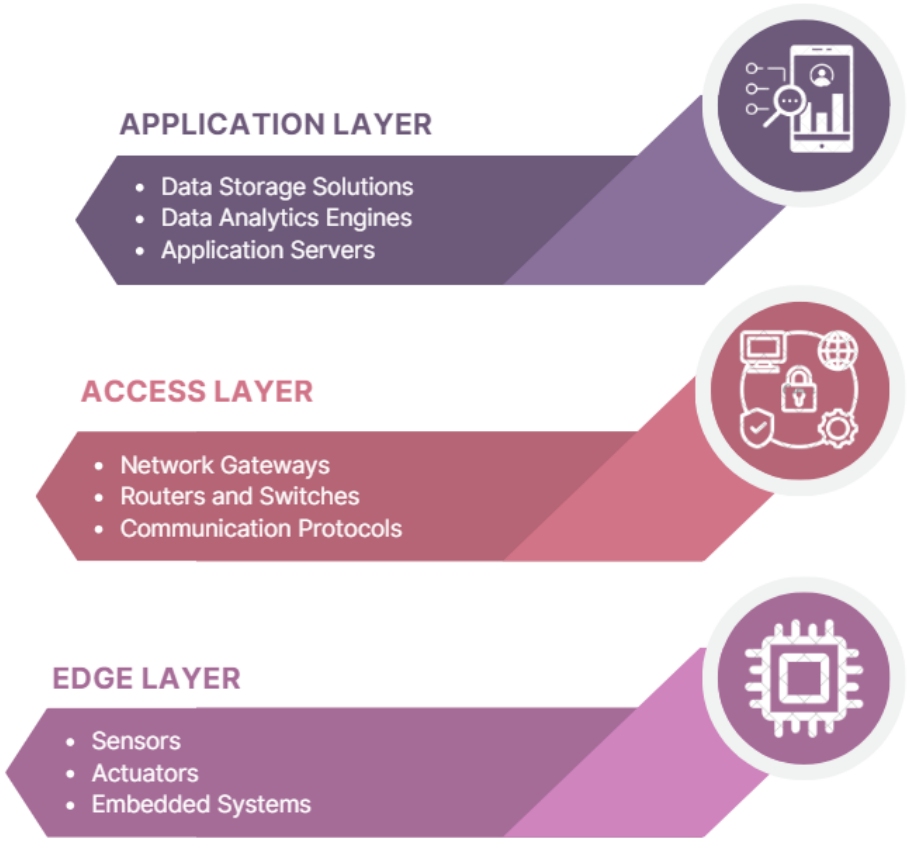



Figure 2.1: IoT Architecture

#### 2.1.1.2 Security Challenges Unique to IoT

The unique challenges IoT devices face are directly linked to the intrinsic factors of the nature and the architecture of the IoT networks. Some of these criteria include:

- *Diverse and Distributed Nature of the IoT Devices.* Since the IoT ecosystem is highly heterogeneous, an IoT system can consist of numerous devices with varying capabilities distributed that are spread out across a large geographical area. This might be translated as creating new, multifaceted, and complex challenges to protect these devices.

- *Resource Constraints in IoT Devices:* the limited processing power, memory, and energy supply, prevent the direct use of traditional security mechanisms that

rely on computationally costly encryption and complex security algorithms [6].

- *Scale and Complexity:* An IoT network might involve a vast number of interconnected devices, with each device acting as a potential point of failure in terms of security. Each individual node can be exploited as an entry point for security breaches. This can be translated into more complex network management and data analysis [7].

- *Data Privacy and Integrity:* The diverse nature of data gathered and transmitted in an IoT network may contain sensitive personal information used in critical infrastructure or personal health monitors. IoT systems must incorporate security mechanisms to guarantee the integrity and confidentiality of this data. Due to this diversity, it is challenging to implement a uniform security privacy and data protection standard [8].

- *Authentication and Access Control:* Traditional authentication techniques are often unsuitable for IoT networks due to the constant growth in the number and diversity of devices. Each device must be securely authenticated using robust and scalable access control mechanisms, as well as innovative and lightweight authentication approaches.

- *Lifecycle Management:* IoT devices cannot often receive remote updates, making it difficult to ensure their security throughout their lifespan. This is a significant difficulty as the firmware and software updates are susceptible to emerging threats over time [9].

- *Physical Security Risks:* IoT devices placed in environments lacking monitoring and physical security are vulnerable to tampering, theft, and physical damage, representing significant physical security risks. [10].

We will expand this explanation in Section 3.2.

#### 2.1.1.3 Security Techniques for IoT Networks

To protect IoT devices different security solutions have to be employed. In most cases, it is insufficient to use one solution and a combination of multiple techniques is necessary. Security techniques used in the internet can often not be used directly for IoT devices (see previous section) and adaptation is necessary. Next we provide a list of current techniques considered in the IoT, one of which is intrusion detection, specifically detecting interferers, which is the focus of this thesis.

- ***Cryptography:*** Cryptography applied to IoT devices must be addressed from a different perspective. Solutions like lightweight, quantum-enhanced, and hybrid cryptography may be the ideal approaches for devices with limited computational resources. For example, lightweight cryptography primitives such as block cyphers, stream cyphers, hash functions, and elliptic curve cryptography (ECC) provide security, confidentiality and data integrity while using minimal computational resources [11], [12]. Likewise, quantum computing enables faster processing and real-time data analysis capabilities. For IoT systems, vast amounts of data can be processed at high speeds and used to solve optimisation and logistics problems efficiently. This can be translated into enhancing decision-making processes in critical IoT applications and optimising energy consumption in IoT networks. For example, quantum computing applied in public-key cryptography presents a promising solution to enhance security [13], [14] and maximise data accuracy in a real-time environment of the IoT application [15]. Furthermore, Hybrid Cryptography proposes the combination of different cryptographic methods to provide comprehensive security solutions tailored for specific IoT applications. It leverages both symmetric and asymmetric methods to enhance security while optimizing performance. For example, in [16], the authors propose a new hybrid cryptography algorithm based on Rivest cypher (RC4), Elliptic-Curve Cryptography (ECC), and Secure Hash Algorithm (SHA-256) to protect sensitive information in IoT-based smart irrigation systems.

Within the cryptography techniques, *Encryption* is also included. Encryption techniques are critical in ensuring that data remains confidential, preventing unauthorised access from adversary users attempting to modify the information, and ensuring that the integrity of data transmitted between devices has not been compromised. There are various challenges to using encryption algorithms as a security solution in the IoT setting. For example, because most IoT devices have limited resources, they may be unable to use complicated encryption algorithms that require a lot of computational requirements. The balance between performance and security based on encryption methods must take into account the fact that the security algorithms do not overburden the device's performance, as well as the cost analysis of regular encryption algorithm updates to protect against emerging vulnerabilities. Several solutions have been presented to design lightweight encryption algorithms appropriate for IoT IoT [11, 17, 18], as well as to develop more efficient key management systems [19, 20], and lightweight encryption solutions [21].

- ***Secure Communication Protocols:*** The IoT nature restricts the use of complex security communication protocols. Therefore, developing more sophisticated and resource-efficient security protocols remains a key area of research and innovation. IoT devices frequently transfer sensitive data over potentially insecure networks. Several IoT communication protocols have been designed to address core security concerns like confidentiality, integrity, authenticity and availability. In general, these protocols can be classified as: i) adaption of standard protocols, where protocols like Internet Protocol Security (IPSec) and Secure Sockets Layer (SSL)/Transport Layer Security (TLS) are adapted for use in IoT contexts to accommodate the limitations of these devices [22], and ii) specialised IoT protocols, designed to ensure security device-to-device communication. These protocols are typically designed to be lightweight while providing required security mechanisms such as encryption and authentication, such as Constrained Application Protocol (CoAP) and Message Queuing Telemetry Transport (MQTT) [23].

- ***Authentication:*** Authentication is a key requirement for ensuring that devices connected within a network are verified and trusted before they communicate and exchange data [24]. Also, preventing unauthorised access and ensuring data integrity and privacy. This procedure starts with establishing a strong identification for each device, guaranteeing each device has a unique, verifiable identity. In resource-constrained environments, methods such as lightweight public key encryption schemes [25], two-factor authentication with secret keys, or Physical Unclonable Functions (PUF), which exploit random variation at the physical microscopic structure introduced by the chip manufacturer, where each PUF is unique [26], might be used to mitigate potential risks. Moreover, IoT ecosystems often include three mechanisms for authentication mechanisms to identify and respond to potential security threats in real-time: i) cryptography for low-power networks to face two principal issues, overhead in transmitting and distributing keys and vulnerability to high-power adversaries, ii) Device fingerprinting in wireless networks to exploit technology-specific features of signals (rate switching, active scanning, packet arrival time, clock skew, channel response) to identify them uniquely, and iii) Deep Learning algorithms, such as LSTM and Neural Networks applied to low-power wireless signals, that have correlation across long time duration, allow to overcome unique challenges from multipath, noise, and high power adversaries [27]. Furthermore, *Zero Trust Security Models* are also implemented as authentication techniques. They assume that no IoT device is trusted by default from inside or outside the network. Verification and validation at every step of data access communication are required [28].

- ***Hardware-based Schemes:*** These schemes employ the use of hardware physical characteristics to provide security to IoT devices. Among the leading solutions, we could note Random Number Generators (RNG) and Hardware Accelerators. Hardware-based RNGs are used in IoT to generate secure cryptographic keys for various authentication protocols, ensuring that the keys are difficult to guess or replicate [29]. True Random Number Generators (TRNG)s are based on physi-

cal processes and inherent unpredictability in specific physical phenomena (e.g., electrical noise or radioactive decay). TRNG are popular in security-critical applications because they generate truly random numbers that are difficult to predict or reproduce. Pseudorandom Number Generators (PRNG) are a popular technique for IoT devices because of their low computational capabilities. They generate numbers that appear random and are generated from an initial value known as a seed. While not truly random, PRNGs can be sufficient for many applications if the initial seed is kept secret and is random enough [30].

Hardware accelerators are specialised circuits designed to improve the efficiency and speed of computing tasks. These accelerators handle particular types of operations; for example, in the context of IoT devices, they are focused on cryptographic operations, enhancing the performance and security of the processes. Cryptographic Accelerators are dedicated hardware components that handle specific cryptographic algorithms like AES (Advanced Encryption Standard), RSA (Rivest-Shamir-Adleman), and ECC (Elliptic Curve Cryptography). They improve processing times and reduce computational load on the main processor, which is especially important in power-constrained IoT devices. Secure Elements (SEs) and TPMs (Trusted Platform Modules) are specialised chips capable of securely storing cryptographic keys and performing cryptographic operations. They offer robust protection against physical and logical attacks and are employed in situations where high security is critical [31].

*Secure Boot and Firmware Integrity* are also implemented to verify the integrity of device firmware at startup, and secure boot mechanism in order to prevent the execution of unauthorised or tampered firmware [32].

- ***Network Segmentation:*** It is a popular defensive mitigation strategy against attacks that seek to limit the damage an attacker can cause by partitioning a network into sections and restricting communication between them. The network is divided into separate segments to isolate IoT devices from other net-

work resources to reduce potential attacks. It aims to isolate specific types of traffic (attacks) to particular areas of the network. Some solutions consider a smart segmentation system that focuses on preventing malware spread across IoT networks by controlling network access through segment units configured with various device features and network information [33]. An appropriate segmentation for a given network environment needs to be selected to implement this technique, in [34], a heuristic search and agent-based simulation for supporting this decision-making process.

A proposed smart segmentation framework focuses on preventing malware spread within IoT networks by controlling network access through segment units configured with various device attributes and network information [35]

- ***IoT Software Security:*** It includes secure coding practices, IoT devices authentication and authorisation, software testing and auditing, regular software updates and patch management. Specifically, software/firmware update operations are vulnerable to middle attacks, or malicious updates can be carried out when legitimate security updates are performed. Then, the IoT device's security updates must be conducted safely and securely. Moreover, some software systems and applications for IoT are developed as open-source systems where developers may employ vulnerable source code in programming languages, which might lead to security threats due to the vulnerabilities in the code. Using secure programming standards that allow the design and develop secure software systems for IoT would significantly minimise potential future attacks. *Middleware* is software with an intermediary function to communicate between applications and the used operating system. Providing security to the middleware software in IoT is imperative as it deals with format, structure and information encoding that is exchanged among layers, sensors, applications, devices and operating systems [36].

- ***Intrusion Detection:*** Intrusion Detection System (IDS) monitors host or network

operations and alerts the system administrator when it discovers security violations. In the IoT context, we need to examine the processing and storage capacity of network nodes that host IDS agents, as it represents a significant issue in terms of capacity. Intrusion detection is the activity of detecting actions that intruders carry out against the network [35]. Different approaches can be taken into consideration while implementing an IDS system, for example, the placement strategy (e.g. distributed, centralised or hybrid); the detection method such as signature-base, anomaly-based, specification-based, or hybrid; the security threat that can include conventional attacks, routing attacks, man-in-the-middle attacks, denial of services and others [37], [38].

*Detecting targeted jamming interference* is a component of an Intrusion Detection System (IDS). Jamming attacks aim to disrupt communications by overwhelming the channel with interference (e.g. malicious interference), thereby interrupting device functionality and compromising network integrity. An IDS designed to identify such attacks can monitor for unusual levels of noise or signal disruption indicative of jamming, alerting to potential security breaches in real time. This capability not only preserves the operational continuity and reliability of IoT environments but also ensures that measures can be taken quickly to reduce the impact of an attack [39], [40].

We will expand this explanation in section 3.3.

## 2.2 IoT Communications

The Internet of Things enables the embedding of electronic devices into physical world objects, resulting in smart objects/things that allow humans to interact with the physical world through sensing and actuation. IoT enables networking among smart objects/things, applications, and servers [41].

### 2.2.1 Overview of IoT Technologies

The current Internet of Things (IoT) landscape includes both proprietary and standardised solutions that try to satisfy the IoT application requirements of wide-coverage, high-throughput, low-power, and low-latency [42]. Thus, we can classify the Internet of Things (IoT) communications based on this massive diversity of long-range, short-range, low-power, ultra-low-power and personal area networks and technologies for the design of IoT applications [43]. Figure 2.2 illustrates a general classification according to the aforementioned criteria.

Here is an overview of some of the most relevant IoT technologies that are shaping the way devices communicate and perform in various industries:

**LoRa:** Long Range (LoRa) is a long-range, high robustness, multipath resistance, Dopler resistance and low power proprietary spread spectrum modulation technique developed by Semtech. It is a physical layer specification based on Chip Spread Spectrum (CSS) with integrated Forward Error Correction (FEC). LoRa operates in the lower ISM bands (e.g. 868/433 MHz in the EU and 915/433 MHz in the USA). LoRa is capable of transmitting over very long distances and supports a star topology. LoRa transmissions use a wide band to mitigate interference and handle frequency offsets from low-cost crystals. LoRa radio allows the configuration of four parameters to optimise energy consumption, transmission range and noise resilience: carrier frequency, spreading factor, bandwidth and coding rate.

LoRa defines three types of nodes: i) Class A nodes, the node transmits to the gateway when needed. The node opens a receive window to retrieve queued messages from the gateway after transmission; ii) Class B nodes have scheduled receive slots. The gateway sends beacons periodically to establish synchronisation with the end devices, and iii) Class C nodes that are continuously listening, making them unsuitable for battery power operations [44].

**LTE-M:** Long-Term Evolution for Machine-Type Communications (LTE-M) specifications and design principles were introduced for the Third Generation Partnership Project (3GPP) Release 13. LTE-M technology was designed to fulfil the IoT needs of low device cost, deep coverage and long battery lifetime while supporting a massive number of IoT devices, adequate peak rates and mobility support. Some radio access design principles in LTE-M are low device complexity and cost, coverage enhancement, long device battery lifetime, support of a massive number of devices, deployment flexibility, and coexistence with Long-Term Evolution (LTE). The downlink and uplink transmission schemes principles are the same as in LTE, which means that Orthogonal Frequency Division Multiplexing (OFDM) is used in the downlink and Single-Carrier Frequency Division Multiple Access (SC-FDMA) is used in the uplink, with 15kHz subcarrier spacing in both downlink and uplink.

LTE-M devices rely on the LTE synchronisation signal (PSS and SSS) for cell carrier frequency acquisition, frame timing, duplex mode and physical cell identity. Similarly, the *Physical Broadcast Channel* (PBCH) is used to deliver the Master Information Block (MIB), which carries essential information for the device to operate in the network [41].

**Zigbee:** It is a low-power, low-data rate wireless communication technology developed by the Zigbee Alliance. Zigbee operates on the IEEE 802.15.4 standard for personal area networks. Zibgee design principles focus on providing reliable data transfer, efficient power usage, and secure communications. It supports mesh networking, which allows each node to transmit and receive data. It operates at 868 MHz in Europe, 915 MHz in North America and Australia, and 2.4 GHz ISM band worldwide, with data rates of 20 kbps, 40 kbps, and 250 kbps, respectively. Zigbee networks can support up to 653356 devices, with a distance between them of 50 meters.

Regarding the physical device types, Zigbee can support two types: Full-Function Device (FFD), which perform the role of coordinator, router or end device and Reduced-Function Device (RFD) devices that perform limited tasks; they do not route

packets and must be associated with an FFD. Furthermore, Zigbee protocol has two ways of multi-access: beacon and non-beacon. In the beacon enable mode, Zigbee nodes can only transmit in predetermined time slots. In the non-beacon enabled network, if the channel is free, every node in the network can send data [45].

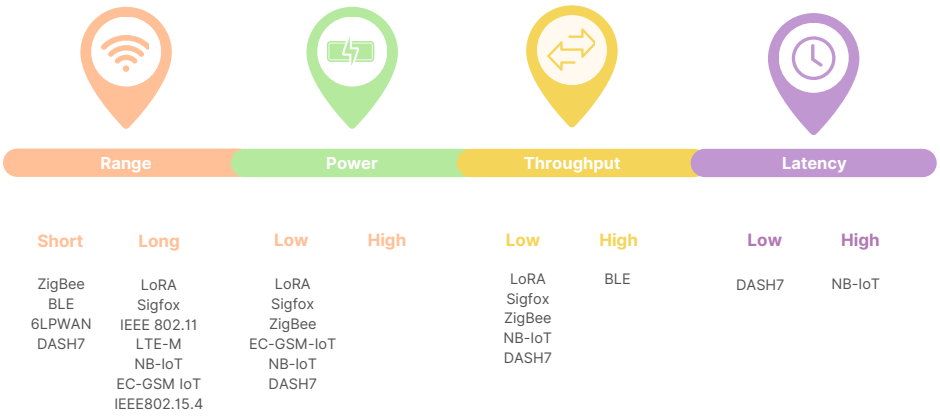



Figure 2.2: IoT General Classification

In terms of security, LoRa implements end-to-end security measures tailored for IoT applications. It utilises AES-128 encryption for application payloads, ensuring data confidentiality. Each device employs unique 128-bit keys: a Network Session Key (NwkSKey) for network-level security and an Application Session Key (AppSKey) for end-to-end encryption of application data. These keys provide network-level security and end-to-end encryption of application payloads, respectively. Device authentication is achieved through either Over-The-Air Activation (OTAA) or Activation By Personalization (ABP). Message integrity is maintained using Message Integrity Codes (MIC), while frame counters prevent replay attacks. It uses AES-CMAC for message integrity checks and AES-128 in CTR mode for payload encryption. This security architecture, combined with the long-range capabilities of LoRa, makes it well-suited for large-scale, low-power IoT deployments requiring robust security [46].

LTE-M inherits LTE's comprehensive Evolved Packet System (EPS) security archi-

tecture. It uses the AKA protocol for mutual authentication between devices and the network and deriving a 256-bit master key (KASME) for subsequent key generation. Data protection is ensured through 128-bit encryption keys for user and control planes, supporting algorithms like SNOW 3G, AES, and ZUC. The protocol maintains integrity protection for control plane messages and utilizes SIM-based security for key storage and cryptographic operations. LTE-M implements identity privacy measures to protect subscribers' permanent identities through GUTI and IMSI concealment and supports secure inter-cell handovers using Next Hop chaining. The protocol secures both non-access stratum (NAS) and access stratum (AS) layers, protecting signalling and user data. This robust security framework, coupled with cellular network integration, positions LTE-M as a highly secure option for IoT applications requiring wide-area coverage and mobility [41].

Zigbee enhances the IEEE 802.15.4 standard by implementing additional network and application layer security features. It employs a Network Key shared by all devices for broadcast communications and supports unique Link Keys for point-to-point device communication. Key transport is secured from the Trust Centre to devices, it manages security, including key distribution and device join management. Zigbee uses AES-128-CCM* for frame encryption and authentication. Devices must authenticate before joining the network. The protocol supports periodic Network Key rotation for long-term security enhancement. Zigbee also provides security services at the application layer, enabling end-to-end security. These comprehensive security measures make Zigbee well-suited for secure IoT deployments in mesh network environments [47].

Our research experiments focused on two technologies: i) Narrowband-Internet of Things (NB-IoT) and ii) IEEE 802.15.4. We provide in-depth information about both technologies in the following sections.

### 2.2.2 Narrowband-Internet of Things (NB-IoT)

This section introduces the design principles of the Narrowband-Internet of Things (NB-IoT) technology. Later, we will look at the physical channels and signals, both downlink and uplink, to understand how these channels were designed to achieve the NB-IoT goals of extended coverage, ultra-low device cost, long battery lifetime, and the ability to support a large number of devices in one cell. We focused on the downlink and uplink transmission schemes, particularly how they are mapped to radio resources in terms of frequency and time dimensions. Subsequently, we discuss the NB-IoT radio signal exchange during the initial steps of a UE to establish communication with an eNodeB.

#### 2.2.2.1 NB-IoT Design Principles

The Third Generation Partnership Project (3GPP) completed the standardisation of the core specifications for NB-IoT in June 2016, through the 3GPP Release 13 standard. The 3GPP research, *Cellular System Support for Ultra-low Complexity and Low Throughput Internet of Things* [48], introduced a new LPWAN technology with improved features in terms of coverage, battery lifetime, and capacity. 3GPP aimed to standardise a technique that would use software upgrades to exploit the already existing well-established cellular network to meet all the IoT network's performance requirements. As a result, 3GPP created the standard for integrating IoT functionalities into existing Global System for Mobile communication (GSM) networks via software upgrades.

Historically, at that point, 3GPP considered the study of none-GSM backward compatible technologies, also known as *clean-state solutions*, as network operators considered refarming their GSM spectrum for two purposes: i) to deploy the LTE technology and ii) to deploy LPWAN dedicated for IoT services. Because NB-IoT is part of the 3GPP LTE specifications, it may be implemented by a software update to current LTE networks.

In summary, the NB-IoT design principles discussed in the next section enable

NB-IoT to be deployed in both the refarmed GSM spectrum and within an LTE carrier. Further 3GPP releases examine improvements in terms of energy efficiency, system performance, and compatibility with Massive Machine-Type Communication (mMTC) in 5G networks [49].

**Radio Access Design Principles** According to Liberg et al., [41], the radio access design principle in NB-IoT takes into account five key criteria:

- *Low device complexity and cost:* NB-IoT devices simplify the synchronisation during the initial cell selection process, as well as the decoding and demodulation process during the data reception. To establish time and frequency synchronisation with the network, an NB-IoT node only needs to look for one synchronisation sequence. For the demodulation procedure, NB-IoT limits the downlink Transport Block Size (TBS) to 680 bits. LTE Tail-Biting Convolutional Code (TBCC) is a simple convolutional code used by NB-IoT devices for downlink channel coding. Regarding the RF requirements, NB-IoT devices only require one transmit and receive antenna operating in a half-duplex mode.

- *Coverage enhancement:* NB-IoT uses repetitions to guarantee reliable communications in locations with limited coverage, even at a reduced data transmission rates.

- *Long device battery lifetime:* 3GPP Release 12 and 13 introduce two new capabilities: i) extended Discontinuous Reception (eDRX), and ii) Power Saving Mode (PSM), to optimise the power consumption in NB-IoT devices. As IoT applications require infrequent transmission of small packets, NB-IoT nodes spend the majority of their lifetime in idle mode. During idle mode, the NB-IoT device needs to monitor paging and perform mobility measurements. As a result, significant energy can be saved by increasing the periodicity between paging occasions (POs) or not requiring the device to monitor paging at all during idle mode.

- *Support of a massive number of devices:* The capacity per single NB-IoT carrier can support a large number of devices because of the introduction of a spectral efficient transmission scheme in the uplink for devices in extreme coverage-limited situations.
- *Deployment flexibility:* NB-IoT supports three modes of operation to provide greater flexibility in implementation due to the refarming scenarios for its adoption. We will expand on this in the following section.

#### 2.2.2.2 NB-IoT Operation Mode

**Standalone Operation Mode** A standalone operation mode consists of an NB-IoT network deployed in a standalone carrier using any available spectrum larger than 180 kHz. For the specific case, when GSM operators refarm their GSM spectrum, the coexistence requirement between the GSM carrier and the NB-IoT carrier indicates that: i) a 200 kHz guard-band is recommended, which means that a GSM carrier should be left empty on one side of an NB-IoT carrier between two operators, and ii) a 100 kHz guard-band is recommended when the same operator is implementing both GSM and NB-IoT. Operators must refarm at least two successive GSM carriers for NB-IoT deployment, as shown in Figure 2.3.

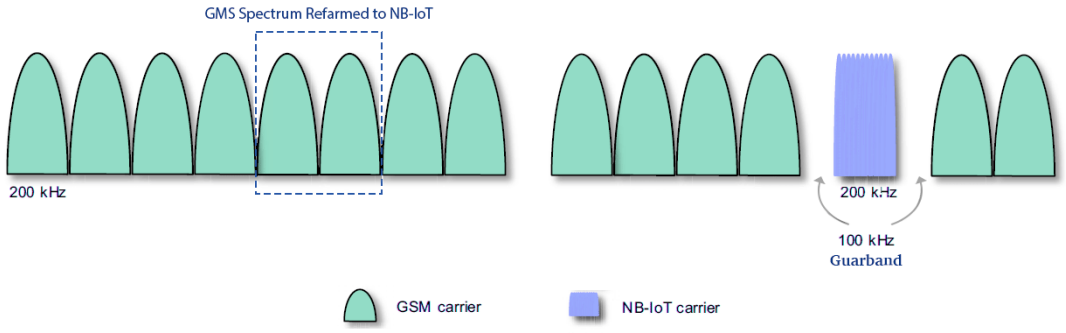



Figure 2.3: NB-IoT Standalone Deployment using Refarmed GSM Spectrum.

**Inband and Guardband Operation Modes** NB-IoT deployment within an LTE carrier can be accomplished by utilising one of the LTE physical resource blocks (PRBs) for the Inband operation mode or the LTE guard-band for the Guardband operation mode, as illustrated in Figure 2.3. Guardband deployment allows for the placement of an NB-IoT carrier in the LTE guard-band because an LTE signal occupies around 90% of the channel bandwidth [50], leaving 5% of the LTE channel capacity on either side, known as the guard-band.

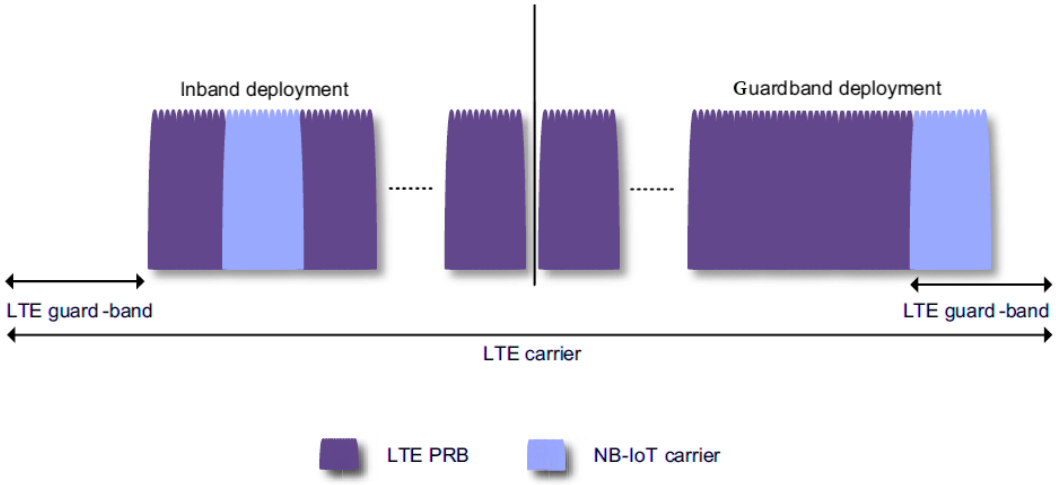



Figure 2.4: NB-IoT Inband and Guardband Operation Modes

*Efficient Spectrum Use:* The three NB-IoT operation modes discussed previously provide network operators significant flexibility regarding spectrum migration. Network operators can refarm a non-significant part of the GSM spectrum to enable NB-IoT for the Standalone deployment. While this NB-IoT carrier is already in the GSM network, it can become an NB-IoT Inband or Guardband after the entire GSM spectrum will be migrated to LTE. Similarly, the same strategy can be applied when the LTE is eventually refarmed to 5G as the superior coexistence performance design implemented between NR and NB-IoT. This flexibility was envisioned to avoid future spectrum fragmentation [41].

#### 2.2.2.3 NB-IoT Physical Layer

**NB-IoT Frame Structure** NB-IoT frame structure considers four levels: hyperframe, frame, subframe and slot. Figure 2.5 illustrates the frame structure for 15kHz subcarrier spacing. One hyperframe cycle contains 1024 hyperframes. On the second level, one hyperframe has 1024 frames, each comprising ten subframes. Each subframe is divided into two slots of 0.5 ms. Table 2.1 shows the label identifier for each level, the range, which is the number of structures present at each level, and their duration.

The NB-IoT design allows for a subcarrier spacing of 15 kHz in both the downlink and uplink, with 20 slots per frame, as shown in Figure 2.5. While, in the uplink, NB-IoT technology allows for 3.75 kHz of increased subcarrier spacing, here each frame is directly divided into five of 2 ms slots.

Table 2.1: NB-IoT Frame Structure

| Frame Structure | Label | Range | Time |
|---|---|---|---|
| Hyperframe | Hyper System Frame Number (H-SFN) | 0 - 1023 | 10.24 s |
| Frame | System Frame Number (SFN) | 0 - 1023 | 10 ms |
| Subframe | Subframe Number (SN) | 0 - 9 | 1 ms |

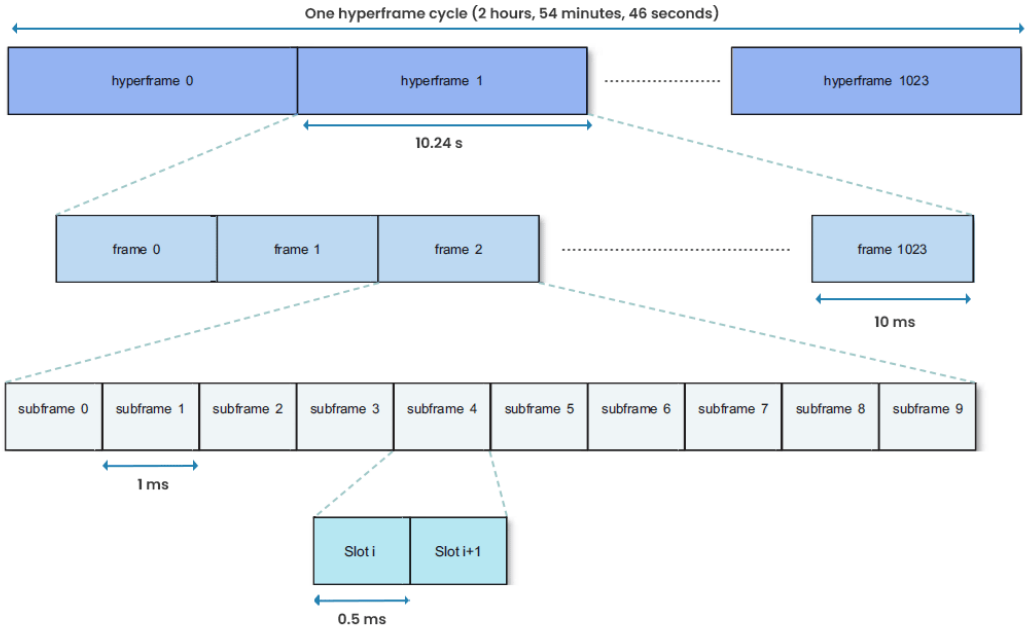



Figure 2.5: NB-IoT Frame Structure

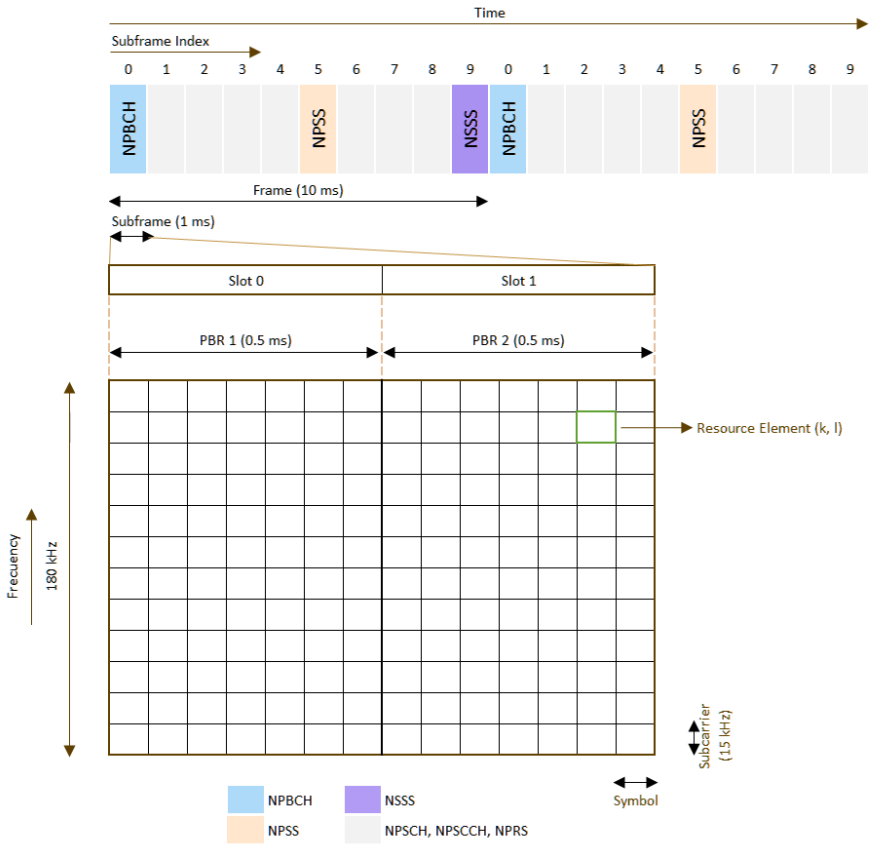



Figure 2.6: Physical Resource Block in NB-IoT Downlink

**NB-IoT Physical Resource Block** Now, it is worth to define the concepts of Physical Resource Blocks (PRB) and Resource Elements (RE). In the downlink channel, a RE is defined as the smallest physical channel unit within the PRB. A PRB pair is defined as the smallest schedulable unit that fits into two consecutives slots. A PRB uses 12 subcarriers across 7 OFDM symbols, resulting in 84 REs. It can be concluded that a PRB is defined to mapping the physical channels and signals on the RE. Figure 2.6 depicts the PRB pair in the NB-IoT downlink.

In contrast, the resource unit (RU) is defined in the uplink to specify the mapping of the uplink physical channels onto REs. The RU setting will be determined by the subcarrier spacing configuration and the amount of subcarriers allocated.

**NB-IoT Transmission Schemes** NB-IoT Release 15 support two transmission schemes Frequency Division Duplex (FDD) and Time Division Duplex (TDD). In the downlink, NB-IoT employs Orthogonal Frequency Division Multiplexing (OFDM),

using the same numerologies as LTE in terms of subcarrier spacing, slot, subframe, and frame durations.

#### 2.2.2.4 Physical Channels and Signals

**Downlink Physical Channels and Signals** For the downlink, the following channels and signals are used for communication.

- Narrowband Reference Signal (NRS).
- Narrowband Primary Synchronisation Signal (NPSS).
- Narrowband Secondary Synchronisation Signal (NSSS).
- Narrowband Broadcast Channel (NPBCH).
- Narrowband Physical Downlink Shared Channel (NPDSCH).
- Narrowband Physical Downlink Control Channel (NPDCCH).

We can analyse the physical channels and signals in the downlink channel considering the time-multiplexing on an FDD NB-IoT anchor carrier in a 20 ms period, as the same pattern is repeated in succeeding periods. Figure 2.7 depicts the allocation of the downlink physical channels and signals.

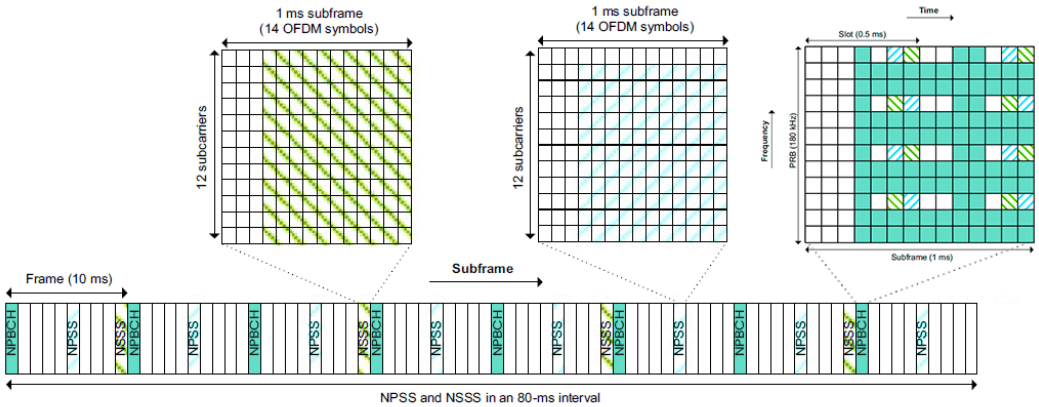



Figure 2.7: Time-Multiplexing of Downlink Physical Channels

**Uplink Physical Channels and Signals** The set of supported NB-IoT uplink physical channels and signals are listed below:

- Narrowband Physical Random Access Channel (NPRACH).
- Narrowband Physical Uplink Shared Channel (NPUSCH).
- Demodulation Reference Signal (DMRS).

#### 2.2.2.5 NB-IoT Communication

The NB-IoT standard specifies that NPSS and NSSS are transmitted in subframes with an 80 ms repetition interval. Both signals enable devices to employ a common synchronisation technique during initial acquisition, regardless of the NB-IoT operation mode [41]. To prevent collisions with LTE subframes, subframe 5 is used for the NPSS and subframe 9 for the NSSS. To prevent a collision with the LTE Physical Downlink Control Channel (PDCCH), subframes carrying NPSS or NSSS do not employ the first three OFDM symbols. This leaves only 11 OFDM symbols per subframe for NPSS and NSSS. The NPSS detection by the UE necessitates the signal being detected even with a significant frequency offset. Therefore, all cells in a NB-IoT network employ the same NPSS sequence. As a result, a device only needs to look for one unique known NPSS sequence.

The decoding mechanism for NPBCH in NB-IoT is detailed in 3GPP TS 36.211 V15.7.0 [51], [52], [53], and [54]. We will expand the NPBCH details in 5.5.

**Idle and Connected Mode Procedures** The idle mode used in NB-IoT consists of initial cell selection, System Information (SI) acquisition, cell reselection, paging procedures and PSM operation. To move from idle to connected mode, a NB-IoT node uses random access, access control, connection establishment, and multicast procedures. In contrast, the connected mode operation includes scheduling, retransmission, power control and positioning procedures.

The *Random Access Procedure* in idle mode is generally the same as in LTE. Figure

2.8 depicts the Random Access Procedure illustrating the initial steps of a UE to establish communication with an eNodeB.

In general, the NB-IoT node searches for the NPSS signal to identify the presence of an NB-IoT cell. It synchronises in time and frequency to the NPSS in order to determine the carrier frequency and the frame structure within a frame. Then, the NB-IoT node uses the NSSS to identify the Physical Cell ID (PCID) and the three least significant bits (LSB) of the System Frame Number (SFN) for subframe identification purposes and to identify whether the cell is an FDD or TDD cell. Each NPBCH subframe carries a Code Sub-blocks (CSB). A CSB is repeated in eight consecutive NPBCH subframes.

When the UE acquires the PCID, it knows the NRS placement within a resource block. With this information, it can demodulate and decode the NPBCH, which contains the Master Information Block (MIB). NPBCH is trasmitted in subframe #0 in an FDD cell. The MIB acquisition allows the identification of the operation mode, the four Most Significant Bits (MSB) of the SFN, and the two LSBs of Hyper System Frame Number (H-SFN) to resolve the frequency raster offset. Also, the Narrowband System Information Block Type 1 (SIB1-NB) scheduling and transmission mode (anchor/non-anchor carrier) is provided by the MIB. For the specific case of FDD, the SIB1-NB is always transmitted on the anchor carrier. From MIB, the SI value tag and the Access Barring (AB) information is also specified. Furthermore, from the SIB1-NB, the UE will be able to identify the complete H-SFN, the Public Land Mobile Network (PLMN), tracking area, and cell identity and to prepare for verification of the cell suitability. After acquiring SIB1-NB, the IoT device has achieved complete synchronisation to the frame structure. Thus, SIB1-NB broadcast the information regarding the minimum required signal strength and signal quality threshold parameters, and the UE will be able to evaluate the suitability of the cell against these parameters.

After synchronising with the network and ensuring that access is not denied, the device transmits a random access preamble via NPRACH (Msg 1). The UE must deter-

mine an appropriate NPRACH configuration according to its coverage class estimation. Then, a random access response (Msg 2) is sent from the eNodeB on the NPDSCH containing the timing parameters to correct time offsets, uplink resource grants, and temporary Cell Radio Network Temporary Identifier (C-RNTI). After receiving the Random Access Response, the UE uses the NPUSCH to send its connection setup request (Msg3), which includes its identity and any necessary information for network registration. The eNodeB completes the random access procedure (Msg4) by delivering a contention resolution message via NPDSCH, validating the UE's identity and allocating resources for normal data communication. Following successful random access, the UE and eNodeB continue to exchange data via NPDSCH and NPUSCH.

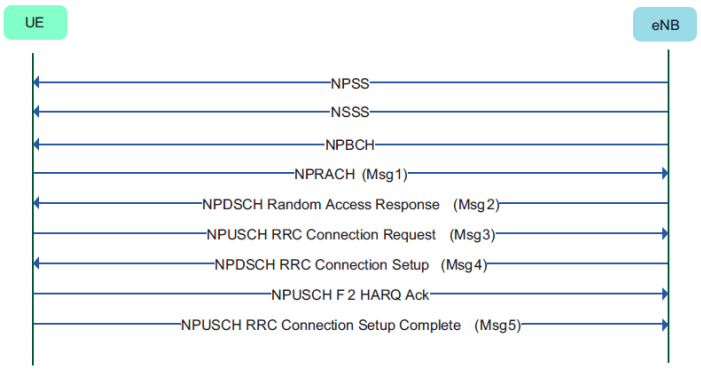



Figure 2.8: NB-IoT Random Access Procedure. Initial steps of a UE to establish communication with an eNodeB.

### 2.2.3 IEEE 802.15.4

IEEE 802.15.4 is a wireless standard developed explicitly for the purpose of establishing Low-Rate Wireless Personal Area Network (LR-WPAN) that operate at low data rates. A LR-WPAN is a low-cost communication network designed to achieve reliable data transfer, extremely low cost and reasonable battery life [1].

IEEE 802.15.4 defines two types of devices:

- **Full-Function Device (FFD)** – it is a device that can served as either a Personal Area Network (PAN) coordinator or a coordinator.
- **Reduced-Function Device (RFD)** – it is a device designed for basic and ex-

tremely simple applications; it does not send large volumes of data. It can only be associated with a single FFD at a time and requires minimal resources and memory capacity. However, it cannot function as a PAN coordinator or coordinator.

#### 2.2.3.1 IEEE 802.15.4 Architecture

According to IEEE standard for Low-Rate Wireless Personal Area Network (LR-WPAN) [1], IEEE 802.15.4 devices must include a PHY layer containing the radio frequency transceiver and its low-level control mechanism, and a MAC sublayer allowing access to the physical channel. Generally, on the upper side of the architecture, the next higher layers are located as depicted in Figure 2.9.

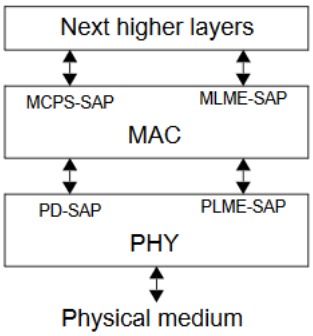



Figure 2.9: LR-WPAN device architecture [1]

- **PHY Layer:** The IEEE 802.15.4 physical layer is responsible for data transmission and reception across a particular radio channel using a specific modulation and spreading technique.
- **MAC Layer:** The IEEE 802.15.4 MAC layer is basically similar to the CSMA/CA system, with the addition of a customisable time slot structure and security functionality, and it serves as an interface between the physical layer and the higher layer protocols of LR-WPAN. The MAC protocol supports two operational modes: beacon-enabled mode and non-beacon-enabled mode, which are explained in detail in the section 2.2.3.2.

An IEEE 802.15.4 LR-WPAN can operate in one of two topologies, depending on

the application requirements. The first is a peer-to-peer topology, which allows nodes to communicate directly with one another. The second is a star topology, in which all communications, including those between devices, must be routed through the PAN coordinator [55].

#### 2.2.3.2 IEEE 802.15.4 Operation Modes

The IEEE 802.15.4 MAC [1] specifies two MAC modes: (i) non-beacon enabled and (ii) beacon-enabled. The non-beacon-enabled MAC, similar to the IEEE 802.11 MAC, operates as an asynchronous carrier-sense multiple access (CSMA) MAC. The MAC with beacon has two separate MAC periods: (i) a synchronized-CSMA MAC period, and (ii) a time-slotted, contention-free MAC period.

**Beacon-enabled MAC:** The beacon-enabled approach adopts a slotted Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) mechanism. Here, the network coordinator periodically broadcasts beacon frames to announce the presence of the network and deliver timing and control information. These beacons divide the channel into superframes of the same direction. The superframe structure is depicted in Figure 2.10.

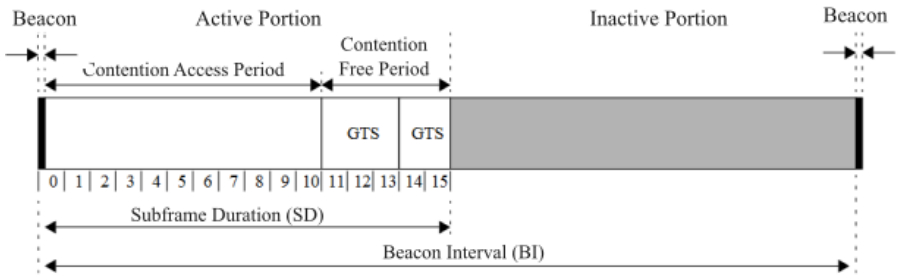



Figure 2.10: Superframe Structure

A superframe is partitioned into two distinct intervals: a mandatory active period and an optional inactive period. The active period comprises both the Contention Access Period (CAP) and the Contention-Free Period (CFP). The CAP is a specific segment within the superframe where devices can transmit data using a contention-based access method. During this period, devices that need to communicate attempt

to transmit their data using a Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) mechanism. CSMA/CA employs a mechanism to prevent collisions by checking the channel's availability before initiating data transmissions. The CFP period is a designated segment of the superframe exclusively reserved for contention-free communication. During this period, specific devices, typically associated with the network coordinator, have the capability to transport data without the requirement for contention. The coordinator allocates time slots to individual devices, allowing them to transmit data in a coordinated and collision-free manner. The active period is equally divided into 16 slots.

**Non-Beacon-enabled MAC** The non-beacon-enabled approach adopts unslotted Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). When a device tries to send a MAC frame, it first conducts the Clear Channel Assessment (CCA) to assess the condition of the channel. The device immediately transmits the MAC frame when the channel is idle. Otherwise, the device remains in a state of waiting for a period and tries again. The backoff time is defined by an exponential backoff algorithm.

The IEEE 802.15.4 standard states that under the beacon-enable MAC operating mode, a device can use a Guarantee Time Slot (GTS) to operate on a specific interval of the superframe that is exclusively allocated to that device on the Personal Area Network (PAN).

#### 2.2.3.3 Guaranteed Time Slot (GTS)

Guarantee Time Slot (GTS) is a feature of the MAC sublayer that allows a device to operate on a channel within a dedicated section of the subframe that will be used exclusively for that device. GTSs are part of the CFP period. The CFP must begin on a slot boundary immediately following the CAP period and conclude before the superframe's active section is completed. If the PAN coordinator has allocated GTSs, they must be placed within the CFP and occupy contiguous slots. As a result, the

duration of the CFP varies with the overall time of all of the GTSs combined. Some GTS general requirements are listed below:

- The PAN coordinator is responsible for the GTS allocation and is used only for the communication between the PAN coordinator and a device associated with the PAN via the PAN coordinator.
- The PAN coordinator can allocate up to seven GTSs simultaneously, as long as the superframe has the capacity.
- A single GTS may span one or more superframe slots
- The GTS Request command is sent from the device to the PAN coordinator. The PAN coordinator must be allocated the GTS before use. Therefore, it decides whether to do so based on the GTS Request command criteria and the superframe's current available capacity.
- GTSs will be assigned on a first-come, first-served basis, with all GTSs positioned adjacently at the end of the superframe and after the CAP.
- If the GTS is no longer needed, it must be deallocated. Deallocation can happen any time at the discretion of the PAN coordinator or the device that originally requested the GTS.
- The PAN coordinator must be able to shote the information to manage seven GTSs, such as the starting slot, length, direction, and associated device address. The GTS direction specifies the data flow from the devices that own the GTS; it is defined as transmit or receive.

**GTS Allocation** Once the PAN coordinator determines that capacity is available for the requested GTS, it will generate a GTS descriptor, which contains the requested specifications and the short address of the requesting device. The starting slot in the superframe at which the GTS begins and the length of the GTS are set up by the PAN coordinator in the GTS descriptor when the GTS was successfully allocated. After-

wards, when the device receives a Beacon frame with the GTS descriptor, and if the GTS allocation request is successful, the next higher layers must be notified about the updated GTS by the PAN coordinator.

**GTS Usage** If a valid GTS is found by the MAC sublayer of a device which is not the PAN coordinator, the MAC sublayer must transmit the data during the GTS, which means between its starting slot and its starting slot plus its length. At this point, the MAC sublayer must send the MAC Protocol Data Unit (MPDU), which is the Protocol Data Unit (PDU) containing the user data, without using CSMA-CA until the requested transaction can be completed before the end of the GTS. If the requested transaction cannot be completed before the end of the current GTS, the MAC sublayer will defer the transmission in the next superframes.

**GTS Deallocation** The deallocation of an existing GTS will proceed through the GTS Request command to the PAN coordinator. The Characteristics Type field on the request must be set to zero, which means, GTS deallocation, the length and direction fields must be set according to the characteristics of the GTS to be deallocated. Once receiving a GTS Request command with the Characteristics Type field set to zero, the PAN coordinator must attempt to deallocate the GTS and notify the next higher layers of the change.

#### 2.2.3.4 IEEE 802.15.4 GTS Communication

Figure 2.11 describes the communication messages exchanged between an IEEE 802.15.4 node and the PAN Coordinator during an IEEE 802.15.4 GTS allocation process. The device sniffs to collect beacon frames from PAN Coordinator to choose the PAN with which it would like to associate and to request and allocate a GTS slot through the GTS Allocation Request message sent to the PAN Coordinator. The GTS allocation policy follows a first-come-first-serve approach, whereby GTS slots are located after the CAP [56]. A GTS must be allocated only by the PAN coordinator. A single GTS can extend over one or more superframe slots. The PAN coordinator must

store the starting slot, length, direction, and associated device address for each GTS. If the PAN coordinator identifies available slots in the current superframe, new GTS information is included in the following beacon through the generation of a GTS descriptor with the requested specifications. The device is able to receive the beacon, process the descriptor and extract the GTS transmission time. When a valid GTS is identified, theMedium Access Control (MAC) sublayer transmits the data during the allocated GTS. This transmission occurs between the starting slot of the GTS plus its starting slot and the length of the GTS.

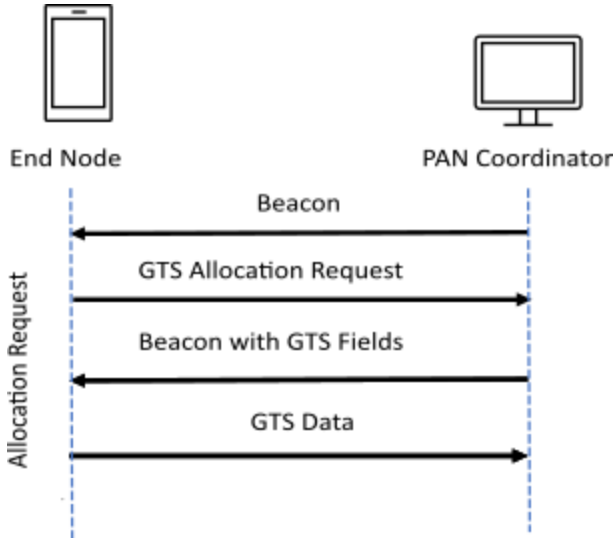



Figure 2.11: Exchange Messages in IEEE 802.15.4 GTS Allocation

The IEEE 802.15.4 security considerations will be explained in detail in Chapter 6.

## 2.3 Interference

### 2.3.1 General

Interference in the context of IoT is a major concern as it can have a great impact on the performance and reliability of the networks. Interference arises due to multiple factors: i) the scarcity of the spectrum suitable for wireless transmissions. As many technologies share the same frequency bands, many existing and emerging communication systems need to coexist with each other. So, they incur in complex interference

scenarios [57]. ii) Devices sharing a sensing and actuation utility across multiple and simultaneous running applications [58]. iii) External interference sources [59].

Hence, the study of interference in IoT is worthy of a comprehensive examination and should be effectively addressed.

This section focuses on the potential categorization of interference in IoT networks, with a particular emphasis on interference caused by malicious adversaries.

- **Radio Frequency Interference (RFI):** It usually came from devices that emitted radio frequencies in the same frequency band as the IoT devices under analysis. Thus, unwanted radio frequency signals disrupt the normal operation of IoT devices. The impact on the communication can include packet loss, increasing latency, and reduction of the data throughput. Some mitigation techniques can consist of frequency hopping, the use of spread spectrum technologies, dynamic frequency selection, and implementing robust error correction protocols [60]. Several research studies have investigated the impact of radio frequency interference (RFI) in the context of the Internet of Things (IoT). Wang et al. [61] highlight the challenges of RFI detection and coexistence in industrial IoT applications. Their research proposes a novel and efficient method for detecting Microwave Oven (MWO) interference. To achieve coexistence between IEEE 802.11 WIFI and IEEE 802.15.4-based sensor networks, they present an enhanced spectral matching method to detect 802.15.4 packets, WIFI packets, and the hopping frequency of 802.15.4 packets. Dionisio et al. [62] outlined the challenges and open issues that must be solved in order to reduce interference and make industrial wireless systems compatible. Using appropriate propagation models, the authors present a methodology for assessing radio interference between various industrial IoT systems on the factory floor. Hwang et al. [63] propose a framework forRFI-aware design. The framework can be used to systematically model radio frequency interference, resulting in the development of mitigation methods.

- **Co-Channel Interference (CCI):** CCI arises when multiple devices transmit simultaneously on the same frequency band, resulting in signal interference. This might potentially result in a decrease in signal quality, reduced data rates, and possibly a complete loss of connectivity. Some mitigation techniques include the implementation of advanced modulation schemes, such as MIMO, to enhance signal resilience and robustness. A range of studies have addressed the Co-Channel Interference (CCI) in IoT; for example, [64] introduced a novel channel equalisation algorithm to facilitate the coexistence of NB-IoT and LTE networks. Shahid et al. [65] propose an interference cancellation technique in LoRa networks that allows simultaneous decoding of multiple colliding LoRa messages. Zheng et al. [66] propose a copula-based interference model for resource allocation and network design for the coexistence of IoT networks operating in unlicensed frequency bands.

- **Multipath Interference:** It occurs when the signal reaches the receiving antenna by two or more paths. These can be originated due to reflection, diffraction, or scattering. Multipath interference can be generated due to radio signals reflecting off surfaces like buildings or mountains, allowing numerous signals to reach the receiving antenna. Several solutions have been proposed to minimise this type of interference. For instance, Kumar et al. [67] address the issue of multipath interference in heterogeneous WSN networks. They introduce a new clustering multipath routing protocol (IMCMRP) to reduce multipath interference while maintaining energy efficiency and reliable data transmission. Mohanoor et al. [68] present a solution focused on discovering multiple paths with interference between them but still achieves the highest possible. The nature of path sets with non-destructive interference is examined and simulated in the context of WSN. Lv et al. [69] analyse the multipath propagation, which can result in inter-symbol interference, that significantly degrades the system's performance. They propose a first-path detection algorithm based on NB-IoT networks' constant amplitude zero autocorrelation (CAZAC) sequences. This method focuses

on detecting the first arrival path in multipath fading channels, which is crucial for avoiding inter-symbol interference.

- **Electromagnetic Interference (EMI):** EMI refers to the disruption caused by an external source that affects an electrical circuit through electromagnetic induction, electrostatic coupling, or conduction. This interference can disrupt, degrade, or even stop the communication of IoT devices, leading to data loss, signal degradation, and system failures. EMI can be originated from natural sources, such as lightning, or human-made sources, such as electrical machinery, power lines, and digital devices. Electromagnetic interference is a common cause of network failure, demanding methods to reduce susceptibility and increase robustness [70]. In [71], a stochastic EMI noised model is proposed for IoT networks. The authors argue because electromagnetic pollution is prevalent in the environment in wireless IoT applications, a quantitative measure of electromagnetic pollution is required. They used the EMI environment temperature as the sum of waves emitted by various wireless device antennas and a stochastic EMI radiation resistance depending on the positioning and layout of system components for the detection. The model accurately predicts electromagnetic compatibility (EMC)/EMI. Mynster et al. [72] addressed the challenges and solutions related to EMI interference for IoT devices, focusing on performance monitoring and regulatory requirements.

- **Jamming Interference:** Jamming interference refers to the intentional disruption of communication caused by a malicious node that emits radio frequency signals, thereby interfering with the communication channel. Interference caused by jamming can lead to a total or partial interruption of the communication channel [73].

  In the context of wireless IoT technologies, we carefully study jamming interference monitoring systems in Section 3.6, jamming attacks in Chapter 4, and jamming detection in Chapters 5 and 6.

### 2.3.2 Jamming Interference

Jamming is a subset of Denial of Service (DoS) attack in which an attacker blocks or jams legitimate transmissions by inserting malicious messages or packets into the wireless channel on which the devices operate. Jammers disrupt wireless communication by creating high-power noise near the sending and/or receiving nodes over the whole bandwidth of the transmission channel. Therefore, identifying jamming attacks is critical for implementing countermeasures and avoiding system damage [74]. Jamming attacks can be classified as constant, reactive, intelligent, and random.

- *Constant jamming* involves continuously broadcasting malicious signals on specific frequencies in order to interfere with the target's communication. The purpose is to saturate the frequency band used by the adversary node, making it difficult or impossible for legitimate nodes to send or receive information. Constant jamming is simple but might be energy-intensive and betray the jammer's position.
- *Reactive jamming* includes listening for transmissions from the target node and only sending jamming interference signals in response. This approach is more energy-efficient and difficult to detect because the jammer remains silent until it has to jam a detected signal.
- *Intelligent jamming* involves the use of modern techniques and algorithms to identify, prioritise, and jam the most important adversary communications signals. It can adjust to changing conditions, such as shifting frequencies or modulation techniques, to remain effective.
- *Random jamming* involves intermittent jamming across a wide range of frequencies or at random times. The enemy cannot foresee when or where jamming will occur, complicating their efforts to detect and minimise its effects.

Research on intentional and malicious radio interference or jamming attacks in Wireless IoT networks has significantly progressed in recent years. However, jamming

has also been proposed as a valuable security primitive. This will be discussed more in the following section.

#### 2.3.2.1 Friendly Jamming

Frequency jamming is a powerful attack technique that aims to disrupt wireless communication. Its malicious characteristics have received significant attention in the research literature. However, numerous recent studies propose to alter the approach and envision positive use cases of jamming, so-called friendly jamming, which presents beneficial advantages for wireless networks. Recent proposals for friendly jamming applications include masking communication channels, defending against injection attacks, and access control.

According to Berger et al., [75], in realistic scenarios where jamming is employed to benefit network operations, it is necessary to implement different criteria for the jamming technique known as *minimal evasiveness*. To coexist with other networks, only specific transmissions should be targeted, with minimal impact on other transmissions. They refer to friendly jamming in cases where it is used for the good and is intended to be minimally invasive. Friendly jamming can be implemented to i) block unauthorised communications [76, 77, 78, 79, 80], or ii) to improve secrecy of communications [81, 82, 83, 84].

For example, jamming-based protection can be used to prevent the processing of unsolicited messages in smart home automation systems without impairing legitimate transmissions in neighbouring houses [76]. Wilhelm et al. [78] implement jamming to build a firewall for IEEE 802.15.4 networks, their solution called *WiFire* relies on reactive jamming to selectively block undesired communication without interfering with desired communication. It detects and analyses packets during their transmission, checking their content against a set of rules. Martinovic et al. [80] propose a *Jamming for Good* where jamming is used to provide a new authentication mechanism for IEEE 802.15.4 networks. They integrate jamming into the communication protocol to provide authentication. Authentication is performed by multiple nodes using signal

signatures, and jamming is used to prevent reception of unauthorised transmissions. Gollakota et al. [77] use jamming to build a solution capable of providing confidentiality for the communication of implantable medical devices (IMD). Their radio design allows them to jam IMDs, preventing eavesdroppers from decoding them while being able to decode them themselves.

In the improvement of secrecy of the communication domain, Vilela et al. [81] state that signal interference may be very helpful in providing security. They propose a practical jamming protocol based on the request-to-send/clear-to-send (RTS/CTS) handshake of the IEEE 802.11 standard as a signalling scheme. They demonstrate how packet collisions generated by jamming nodes may be effectively leveraged to achieve new levels of secrecy in multiterminal wireless networks. Jorgensen et al. [82] propose a physical–layer security scheme for wireless networks with known interference, which aims to achieve communication secrecy by making the eavesdropper incapable of decoding the secret wireless message.

Several studies have been conducted on IoT applications, including a friendly jamming method. Li et al. [85] propose a friendly jamming scheme to secure the Internet of Medical Things (IoMT). Their solution protects the confidential medical data of patients collected by medical sensors from being eavesdropped while having no significant influence on legitimate transmission. Wang et al. [86] introduce an anti-eavesdropping scheme using multiple unmanned aerial vehicles (UAVs) that generate jamming signals to disturb eavesdropping activities to secure industrial Internet of Things. UAV-enabled jammers emit artificial noise to mitigate the signal-to-interference plus noise ratio (SINR) at eavesdroppers, consequently reducing the eavesdropping probability. Their method can significantly reduce the eavesdropping effect while having little impact on legitimate communications.

#### 2.3.2.2 Malicious Jamming

Malicious jamming interference refers to the deliberate use of radio frequency (RF) signals to disrupt the normal operation of wireless communications. This type of in-

terference is intended to degrade, obstruct, or entirely prevent the reception and transmission of signals in wireless networks, such as Wi-Fi, cellular, or other radio-based communication systems.

In general, research related to malicious jamming in wireless communication networks, including IoT networks, can be broadly categorized into three areas: i) generation of jamming attacks, ii) detection of such attacks, and iii) countermeasures to mitigate their impact.

Understanding how an *attack* is generated helps test and strengthen the resilience of wireless networks. The tools and frameworks developed in this field are often used in controlled environments to simulate attacks and prepare defences. Many studies regarding jamming attacks have been proposed [73, 87, 88]. Li et al. [87] propose an attack where the jammer adjusts the probability of jamming and the transmission range to cause the most harm to the network in terms of corrupted communication links. The jammer action ends when the network detects it, and a notification message is transferred out of the jammed region. Tang et al. [73] use reactive jamming, where the attacker smartly launches jamming attacks only when it detects legitimate transmission. The study analyses the anti-reactive-jamming transmission strategy for IoT by exploiting the inherent weakness of the jammer via a hierarchical security game approach.

Likewise, *detecting jamming attacks* involves techniques (statistical models, machine learning approaches) to analyse signal anomalies, changes in network traffic, or unexpected packet losses [74, 89, 90, 91]. Strasser et al. [92] offer a jamming detection technique that identifies the cause of bit errors in packets by analysing the received signal intensity during bit reception. The identification of bit errors is based on predetermined knowledge, error-correcting codes, and limited node wiring. Upadhyaya et al. in [93] proposed a passive, non-node-centric, low-overhead network jamming detection mechanism using a machine learning-based approach. They employed dedicated nodes known as anchor nodes to collect network information for jamming detection

rather than using the current in-use IoT nodes and incurring overhead and other communication costs on those nodes. The detection algorithms for simulated and real network data, using radio signal strength information for jamming detection, present an accuracy of 98% and 89.7%, respectively, with no overhead communication cost involved between the nodes. While using multi-path profile information for jamming detection, the accuracy is 99.01% with simulated data from five anchor nodes.The authors implement a machine learning algorithm using binary classification at several anchor nodes distributed across the network in order to determine the presence or absence of a jammer in the network based on the received signal profile.

Regarding *countermeasures*, these are techniques that aim to mitigate or avoid the effects of jamming. They can be developed with a proactive or reactive perspective. Fadele et al. [91] present a novel countermeasure detection and consistency algorithm (CDCA) to counteract reactive jamming attacks on IoT networks. The algorithm employs channel signal strength to ensure packet consistency by detecting whether the data transmission value contradicts the threshold value. The node that sends the threshold value is periodically checked, and the threshold value is compared with the current value after data transmission to determine whether an attack has happened in the network. Taieb et al. [94] presents a comprehensive survey on different jamming attacks based on MAC layer and existence countermeasures for IoT. Li et al. [95] examines the vulnerability of smart jamming strategies and propose a deceptively adversarial attack (DAA)-based proactive intelligent anti-jamming algorithm to assure reliable communication for wireless communication networks.

In summary, malicious jamming attacks and unintended radio interference represent two of the most significant challenges to the reliability of wireless IoT communications, critically compromising the deployment of pervasive applications reliant on such networks. While jamming interference has garnered significant attention in the research community for its disruptive impacts, this thesis presents a novel approach by leveraging malicious interference selectively and natural interference occurrence to devise a lightweight detector aimed at mitigating such threats effectively, explicitly

designed for implementation on the end-user side, catering to the resource constraints typical of IoT networks. This unique approach broadens the understanding of how interference can be utilised and contributes a practical solution to enhance security in communication systems where jamming is a prevalent concern. Thus, this research not only adds to the existing body of knowledge but also sets the stage for innovative defensive strategies against jamming attacks and underscores the importance of developing lightweight, user-centric solutions in the face of evolving technological malicious jamming attacks.

# 3 | IoT Monitoring to Improve Security and Resilience

In this chapter, we investigate the important role of IoT monitoring in enhancing the security and resilience of critical infrastructures such as buildings, smart cities, factories, power plants, and transport systems. IoT systems are integral to the operational security of these applications, making their monitoring not just beneficial but essential. The expansion of IoT networks, especially in large-scale deployments (e.g. smart cities) with thousands of interconnected devices, demands automated monitoring solutions. These systems are designed to detect and mitigate security threats while providing continuous and reliable operation. Through this discussion, we aim to explore various IoT monitoring applications, focusing on their design, implementation, and impact on providing systems resilience. We will also compare different approaches, highlight the commonalities and differences, and identify potential areas for future research. This analysis seeks to provide a comprehensive overview of how IoT monitoring can be strategically implemented to safeguard critical systems against emerging threats and vulnerabilities.

## 3.1 Introduction

As our primary goal is to develop a system capable of detecting whether an IoT network is under attack, thereby ensuring operational integrity and trustworthiness, we intend to thoroughly examine existing IoT monitoring systems in order to discern the

most effective practices and identify potential areas for improvement. For example, we try to answer core open questions such as what kind of IoT monitoring is already done, how people monitor IoT networks, what practices are good or not in terms of efficient use of IoT resources, and if there is something that can be taken while we build our own solution to detect targeted interference in IoT.

Moreover, by exploring various monitoring approaches, we aim to gain a comprehensive understanding of how different configurations affect the overall security and efficiency of IoT networks. This investigation will provide valuable insights into designing systems that are robust in detecting threats and adaptable to the evolving landscape of IoT deployments and the specific challenges they face.

We present existing monitoring practices to evaluate their effectiveness and adaptability in the face of emerging threats in the context of IoT networks. To this end, four pivotal categories are examined: system architecture, analysis methods, performance metrics, and monitoring attacks. Each category is chosen for its relevance to developing a robust and resilient monitoring system capable of counteracting targeted attacks.

- **Architecture Type:** The monitoring system's architecture determines whether the monitoring should occur within the IoT network. The choice impacts the system's resilience and its ability to detect and respond to the attack (e.g. interference) swiftly and accurately. Moreover, it impacts the solution's development costs as it might imply the use of additional hardware or software, which results in a budget increase.

  A centralized architecture might involve gathering data at a single point, such as a base station, which can be effective for comprehensive analysis but may introduce a single point of failure. In contrast, a decentralized architecture employing monitoring at the end devices might enhance fault tolerance and reduce latency in threat detection.

- **Analysis Methods:** This category focuses on the techniques and algorithms used to analyse data collected by the IoT monitoring system. The chosen analysis

methods must be robust enough to handle the vast data volumes generated by IoT devices while remaining sensitive enough to detect even the most sophisticated attacks and disruptions.

Effective analysis methods will distinguish normal operational data from anomalies indicative of targeted attacks. Machine learning models, statistical analysis, and pattern recognition techniques can be employed to identify subtle signs of interference or other attacks that might elude traditional detection methods.

The selection of an accurate method will directly impact the generation of high/low computational costs; therefore, it will influence the rapid and efficient decision-making resulting from accurately addressing real-time data processing.

- **Performance Metrics:** They are considered to evaluate the effectiveness of the monitoring system, ensuring it meets the high standards required for IoT applications. These metrics evaluate the accuracy, efficiency, response time, and reliability of the system. These metrics provide objective criteria to assess whether the monitoring system meets the necessary standards for security and can guide further refinement to enhance its performance.
- **Monitoring Attacks:** Finally, understanding the attacks themselves is crucial to identifying and mitigating potential vulnerabilities within the IoT monitoring system, as a compromised monitor can severely undermine network security. This involves detecting active attacks and anticipating potential threats by analysing attack patterns and strategies commonly employed against similar systems. This proactive approach ensures that the monitoring system remains one step ahead of attackers, ready to prevent any security attempts before they can cause significant damage.

By examining existing IoT monitoring systems, we aim to identify effective practices that enhance security and efficiency while minimising energy consumption. Ensuring that our solution is lightweight in terms of energy will allow it to function

effectively within the strict resource constraints found in IoT environments.

## 3.2 The Need for IoT Monitoring

In this section, we discuss why monitoring systems are deployed in IoT installations in general. Thereafter, we discuss the needs that lead to the deployment of a monitoring system. Different monitoring systems might be deployed in parallel in one IoT installation to fulfil different monitoring needs, even though the different systems share many system aspects.

IoT systems require more than many other IT systems continuous monitoring as they are very dynamic. Most system parameters evolve significantly over the IoT deployment duration. In order to verify if the system is still operating as intended it is necessary to monitor it continuously. In general, IoT systems are very large scale, and any monitoring of such installation cannot be human-driven, and automation is required. The main reasons why IoT systems exhibit a very dynamic behaviour are: *Lifespan*, *Stakeholders*, *Heterogeneity*, *Mobility*, *Coverage*, *Environment*.

**Lifespan:** To be financially viable, an IoT system is deployed over a very long time span. For example, an IoT smart city deployment will be deployed over many decades. During the deployment time, the usage of the IoT system usually changes dramatically. New devices are deployed to measure new parameters and/or to control new environmental aspects. New services are designed to make use of the deployed IoT. These changes usually lead to dramatic changes in system parameters, and it is essential to monitor them. Verification of correct functionality at initial deployment is insufficient; monitoring is required.

**Stakeholders:** IoT infrastructures are increasingly deployed, operated and used by a set of different stakeholders. For example, different companies may provide services that make use of one IoT smart city infrastructure. As multiple stakeholders interact with and extend the system, it is subject to a high rate of change.

**Heterogeneity:** IoT systems are characterised by the integration of a wide range of devices, technologies, protocols, and software into a single network. This diversity leads to complicated interactions and dependencies between diverse system components, such as sensors, actuators, and other devices with varying capabilities, operating systems, and communication standards. This heterogeneity presents significant challenges in terms of security, interoperability, data management, and system maintenance, and traditional monitoring solutions are frequently insufficient, necessitating specialised techniques adapted to the IoT ecosystem's unique mix of technologies and devices. It is critical for the overall effectiveness and reliability of the IoT system that these different components perform seamlessly together. Heterogeneity extends to the various data kinds and processing requirements, necessitating flexible and adaptable ways of data analysis and decision-making processes. Monitoring in such heterogeneous systems must ensure that the system's many components can successfully collaborate without clashes or bottlenecks.

**Mobility:** IoT deployments exhibit slow mobility. Nodes are not necessary mobile (i.e. attached to a vehicle or person) in the classical sense. However, over long deployment duration nodes are moved from time to time which modifies the network structure. For example, a node might be attached to a lamp post and due to road construction the lamp is moved by a few meters. Such continuous small movements lead to changes in network structure which requires monitoring to verify correct functioning. Also, movement of nodes makes asset management more challenging and monitoring is employed to track these assets.

**Coverage:** IoT deployments cover a large deployment area. For example, a smart city deployment extends over many square kilometers of highly built up environment. Such areas are subject to change (i.e. construction, change of use of areas, ...) which has an impact on network performance. Also, as the network is covering a large area where access to devices can not be restricted security challenges are introduced.

**Environment:** IoT devices can be found in a wide range of environments, including industrial settings and urban landscapes, as well as rural areas and personal spaces. Each of these contexts brings with it its own set of security difficulties and requirements. For instance, IoT devices in industrial environments must survive severe temperatures, humidity, and chemical exposure. In contrast, IoT devices used in personal or home environments need to be user-friendly, unobtrusive, and secure. Environmental factors significantly impact the design, deployment, maintenance and monitoring of IoT devices. For example, devices used outdoors require robustness against energy sources for these devices, while those deployed in urban environments must handle high levels of wireless signal interference.

## 3.3 Intrusion Detection System (IDS)

A common security application is to build an IDS system. We would like to find out if there is currently an attack unfolding.

IDS is a mechanism used to prevent unauthorised access and to detect attacks from intruders against a system, inside or outside the network, by analysing the activity in the system itself. According to Hodo et al. [96], an IDS solution is strategically placed on a network to detect threats and monitor packets. The most common design for an IDS system includes three components: sensors, an analysing engine, and a reporting system. Sensors are located at different network places, and their primary function is to collect the data. The analysis engine receives the collected data, examine it and detect intrusions. The reporting system generates an alert to the network administrator if an intrusion is detected.

Several common elements can be taken into consideration to classify an IDS system. In this section, an overview of the most relevant and recent approaches is briefly presented, we have considered the following attributes: Architecture Type, Analysis Method, Monitoring Attack and Performance Metric. Figure 1 shows the proposed IDS scheme.

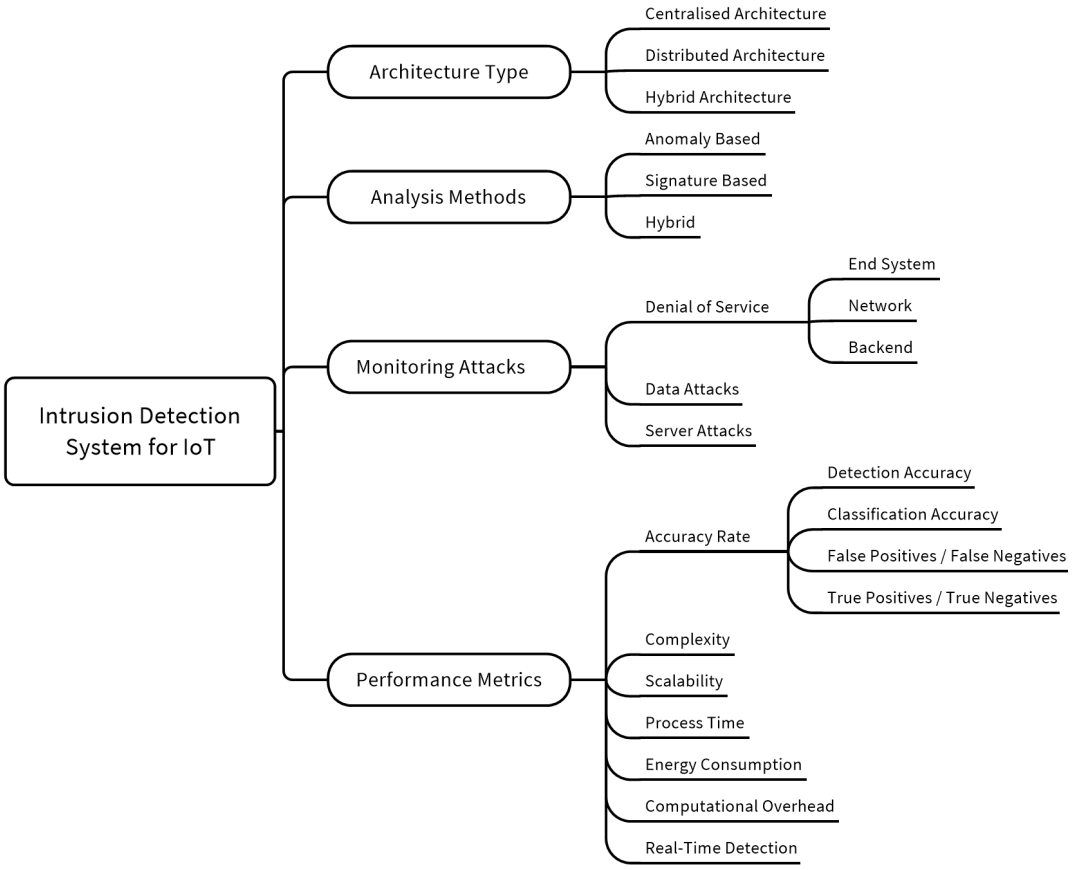



Figure 3.1: Intrusion Detection System Scheme for IoT

### 3.3.1 Architecture Type

IDS systems may be classified according to the IDS agent location into the network. This placement strategy considers three categories: distributed, centralised and hybrid [35]. Each one will differ from each other due to how the intrusion notification data will be broadcasted to the different network elements, and hence, how it will generate a significant impact on the resource performance.

In the Distributed scheme, the IDS agent is located in each physical device that sense and act over the physical environment, the message generated when an attack is detected is sent to the other nodes to isolate the attacker. In this scheme, the nodes may be able to monitor their neighbours, and the IDS deployed in each node must be optimised to guarantee the proper operation of the resources constrained nodes.

In the Centralised scheme, the IDS agent is placed in a centralised element which could be the border router or a dedicated host, the border router analyses all collected and received data. This scheme, although it makes better use of network resources since the nodes will not be processing and sending information continuously, it represents a single point of failure and vulnerability.

The Hybrid scheme combines above approaches, here clusters or regions are defined. The IDS agent is placed in the central node specified for each cluster or region; when an event is detected, the notification alert is sent to the IDS module in the border router of the specific cluster.

### 3.3.2 Analysis Method

Regarding the Analysis Method for detecting intruders in the network, four categories have been compiled: anomaly-based, signature-based, specification-based and hybrid [35].

In the signature-based approaches, IDS have an internal database where patterns or signatures attacks are stored. The Intrusion Detection System detects an attack when a system activity or network behaviour matches with one of these stored signatures, and then the IDS will trigger the alert. This scheme is very accurate at detecting known threats. However, it is not effective to detect new attacks or new variants of known attacks, because all its reliability and accuracy in detecting attacks is based on the information of the signatures stored in the database, so a matching signature for these attacks is still unknown. The mechanism to learn new threats is easy to understand.

The anomaly-based IDS approach defines a normal behaviour profile of the network, typically employing statistical techniques or machine learning algorithms. The IDS compare the activities of a system at an instant against this normal behaviour profile, and the alert is generating whenever a deviation from normal behaviour exceeds a threshold. This approach is efficient to detect new attacks; however, this method

usually has high false-positive rates because anything that does not match to normal behaviour is considered an intrusion. Its mechanism to learn the entire scope of normal behaviour is not a simple task.

The specification-based IDS approach defines a set of rules and thresholds that determine the expected behaviour for network components such as nodes, protocols, routing tables and so on. The IDS detects an intruder when network behaviour deviates from previously specification definitions. This scheme is similar to anomaly-based detection; both aims to identify deviations from normal behaviour. However, there are substantial differences between these two methods, for instance, in specification-based approaches in order to obtain lower false-positive rates than in the anomaly-based detection, a human expert should manually define the rules of each specification. Furthermore, specification-based detection systems can start working immediately after the specification setup, so it does not need a training phase. Manually defined specifications may be time-consuming and error-prone [97], and it may not adapt to different environments.

Finally, a hybrid approach combines the advantages and minimise the disadvantages of any of the methods mentioned above.

### 3.3.3 Monitoring Attacks

In the IoT vulnerability domain, there is a large number of security issues and threats. We classify the monitoring attacks in three main categories: denial of service, data attacks and server attacks. As well as diverse attack types have been addressed in the IDS for IoT, several IoT solutions which involve different technologies, services, and standards have been developed to respond to these security and privacy requirements.

IoT devices may be exposed to environmental hazards due to they are placed at different physical locations. Besides, while data is exchanged, it can face different network vulnerabilities, for instance, it can be shut down because of network floods

or malicious gateway access. Furthermore, an IDS can be exposed to unauthorised access, and various connectivity vulnerabilities regarding data integrity violation, bad QoS, etc.

### 3.3.4 Performance Metric

Performance metric refers to the criteria considered to evaluate an IDS solution. The parameters considered are accuracy rate, complexity, scalability, process time, energy consumption, computational overhead and real-time detection [98].

Accuracy rate involves several parameters such as detection accuracy, classification accuracy, false-positive, false-negative, true-positive and true-negative rates. Complexity metric represents the number of resources required for an efficient operating of an IDS system. Scalability is the system capability to stay efficient when security challenges, data traffic or attacks increases. Network delay includes some transmission times such as propagation delay, processing delay, transmission delay, queuing delay, computational overhead, real-time detection time. Energy consumption indicates the total energy consumed by all nodes during the attack detect process

### 3.3.5 IoT IDS Examples

In this section, we will outline recent implementation examples in the different categories that were presented above to improve security and resilience and discuss their effectiveness. The Intrusion Detection Systems exposed respond to the questions regarding if an IDS is an optimal solution to guarantee that the traffic between the IoT device and the network has not tampered as well as the issue regarding how can we be certain that the device is under attack. We present three recent examples that illustrate the different types of IDS architectures; they manage distinct detection methods and that have been applied and evaluated using diverse security threats and performance metrics.

**Anthi et al. [39]** propose a novel three-layer intrusion detection system (IDS) for

smart home IoT devices. The proposed architecture can automatically distinguish between IoT devices connected to the network, identifies anomalies in the network traffic and therefore detects whether the activity is benign or malicious. Then, if a malicious attack is detected, the system will be able to recognise the type of attack that is happening. The system uses a supervised approach to detect a range of four network-based cyber-attacks such as DoS, man-in-the-middle/spoofing, reconnaissance, and replay. Here is a brief explanation for these attacks: i) Denial of Service (DoS) attacks are malicious attempts to disrupt normal traffic. DoS aims to make an IoT device, network or application inaccessible to its intended users by overwhelming the network, servers, or applications with excessive requests or a flood of traffic. This can lead to service unavailability and disruptions in the functionality of IoT devices and services. ii) A Man-in-the-Middle (MITM) attack involves an unauthorised party (attacker) intercepting and potentially altering communication between two IoT devices without the knowledge of either party. This attack can lead to unauthorised data access and manipulation. iii) In a ppoofing attack, the adversary impersonates a legitimate node, where the retransmission might appear from a valid node. iv) Reconnaissance attacks involve gathering information about an IoT network to identify vulnerabilities that can be exploited in subsequent attacks. This includes scanning for open ports, identifying devices, and mapping network topology. v) Replay attacks involve intercepting and recording legitimate data and then replaying them to trick the receiving device into performing unauthorised actions. The attacker captures the data packets and retransmits them to deceive the receiver into thinking it is a legitimate communication, compromising the integrity and security of IoT communications [99], [100].

They used a centralised architecture where a border router captures all the inbound and outbound traffic from the smart devices and then stored them on a server. Regarding the detection type method, they use an anomaly-based approach, where the system determines a set of principal features from network traffic packets in order to classify the type and profile the normal behaviour of each IoT devices, and therefore identify the malicious packets used in the attack.

Furthermore, they define four automated multilevel security threats. First, the reconnaissance attack, where the attacker performs several network scanning to determine available hosts and vulnerabilities. Second, DoS attack through a quick scan of the network and six DoS attacks perform to a random IP address identified it into the predefined network. Third, MITM attack using ARP spoofing between the router and one random IP address in the system. Four, a complete and an end to end automation attack with IoT-Toolkit.

Moreover, the system conducted the validation using an empirical method as it constitutes a systematic experimental gathering of data from operational settings [35]. The authors developed unique smart home testbeds utilising a combination of eight specific IoT commercially available devices components: the Belkin NetCam camera, TP-Link NC200 Camera, TP-Link Smart Plug, Samsung Smart Things hub, Amazon Echo Dot, British Gas Hive Raspberry Pi, Contiki and sensors.

The IoT scenario and the machine learning technique are Home and Supervised respectively. The main parameters to consider for the development of the machine learning IDS are the IP address and the PCAP files. The IP address is used to detect the device type and the benign/malicious packet behaviour classification. PCAP files contain the smallest piece of the network packet information, and they allow access to all the packet attribute that are used as features to investigate which one can be used to define attack type classification. Supervised machine learning requires labelled training data of the malicious packets.

Finally, to evaluate the performance of the IDS, the system takes into consideration the accuracy rate metric, which includes four outputs: True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). They used de F-measure, which calculates the harmonic mean of precision (P) and recall (R) in terms of the four outputs mentioned above. The results for the three functions are 96.2%, 90.0% and 98% respectively, which demonstrate that the proposed architecture has a reasonable accuracy rate to distinguish how well classification algorithms can learn to profile IoT

devices on the network, to detect if an activity is benign or malicious and to classify the attacks.

**Ferdowsi and Saad [101]** propose a generative adversarial network (GAN) to provide a fully distributed IDS for the IoT to detect anomalous behaviour without reliance on any centralised controller.

The architecture type implemented considers a distributed approach, where every IoT device can monitor is own data as well as their IoT neighbours devices to detect internal and external attacks with minimum dependence on a central unit. The implemented distribute IDS do not share the datasets among IoT devices, due to preserving privacy, these are implemented in IoT applications.

The detection type method used is an unsupervised anomaly-based approach. The model considers an artificial neural network (ANN) that usually consists of artificial neurons and activation functions. For every IoT device, they define two main ANNs called Generator and Discriminator, which are used to generate data points near to the normal state and to discriminate between the anomalous and normal data points respectively. Thus, an intrusion in the system is any activity that causes an IoT device communicates data points that do not follow its data distribution.

Regarding the handled security threats, the proposed distributed IDS is able to detect any attack that manipulates the integrity of the IoT device, for instance, bad data injection attacks. In the practical model, they consider ten scenarios with different attack-to-signal power ratios. Also, they include internals and externals attacks where IoT devices check their own data and its neighbour IoT device data.

The IoT scenario is applied in a health use case, and the validation strategy is conducted using simulations. They use a daily activity recognition dataset from 30 subjects with different genders, ages, heights, and weights using a smartphone. The collected data is for 12 activities as follows: walking forward, walking left, walking right, walking upstairs, walking downstairs, running forward, jumping, sitting, standing, sleeping, elevator up, and elevator down. The data set includes 2,365 recordings with

561 frequency and time domain features each one. For evaluation purposes, the research considers three IDSs: a standalone GAN that has access only to its own dataset, a centralised GAN that has access to all datasets, and the proposed distributed GAN.

To evaluate the performance of the IDS solution, they consider several metrics such as accuracy, precision and false-positive rate. The accuracy rate is calculate based on four outputs: TP, FP, TN and FN; they conclude that the distributed IDS outperforms both of the central and standalone IDSs by up to 15% and 20%, respectively, for both internal and external attacks. The precision calculated indicates that distributed IDS has up to 25% and 20% higher precision compared to the standalone IDS and the central IDS, respectively. The false-positive rate is the number of normal state data points wrongly categorised as an intrusion; the evaluation indicates that the proposed distributed IDS has significantly less false positive rate than the standalone and central IDSs. In summary, the system evaluation shows that the proposed distributed GAN-based IDS has up to 20% higher accuracy, 25% higher precision, and 60% lower false positive rate compared to a standalone IDS.

**Midi et al. [102]** introduce Kalis, a system for Knowledge-driven Adaptable Intrusion Detection for the Internet of Things. The proposed solution collects knowledge about features and entities of the monitored network, and it leverages it to dynamically configure the most effective set of detection techniques. It can detect attacks in real-time across IoT systems that are developed using different communication protocols. Kali was designed to adapt the detection strategy to the specific network features; it does not target an individual protocol or application enabling collaborative security scenarios.

They used a hybrid architecture type, in Kalis any network feature-specific or attack-specific functionality is implemented as an independent module. Its high-level architecture defines a unique and centralised Module Manager which is the component that coordinates, actives and deactivates all the modules; also, it routes new packet events and collects alerts of detected incidents. Furthermore, Kalis includes

two types of distributed modules for sensing and detection. Sensing modules are the core of the autonomous knowledge-discovery approach of the system. They are responsible for capturing traffic to reconstruct the local topology, collecting statistics about the traffic load, and leveraging the signal strength to detect whether the network is static or mobile dynamically.

Regarding the intrusion detection type method, they use a hybrid approach which means the system is either signature-based or anomaly-based. Using the detection modules that analyse the captured traffic and detects anomalies and security incidents, the system is capable of including specialised modules on a specific attack and therefore combine both kinds, to increase the accuracy in detecting known attacks while at the same time be able to react to an unknown attack.

Moreover, the system handles the validation using an empirical method as they conduct the gathering of data through a small WSN which have 6 TelosB nodes, a Nest Thermostat, an August SmartLock, a Lifx smart lightbulb, an Arlo security system, and an Amazon Dash Button. The IDS node is placed near the heterogeneous network.

Furthermore, they choose several security threats to identify two classes of attacks. The first type includes different attacks that have similar or identical symptoms, and the second type comprises attacks with different network features. In the first scenario, they consider two possible attacks as security threats: ICMP flood and Smurf attack, which for an external observer, these two attacks have similar behaviour due to the victim node is receiving an unusually high amount of ICMP Echo Reply messages. The system could reconstruct the topology in the monitoring network and determined that it is a single-hope network. Therefore, accurate detection can be performed, knowing that the Smurf attack is not possible in a single-hop network. Thus, Kalis activates the detection technique for ICMP flood attack. For the second scenario, they evaluate the solution over a wide range of security threats that includes: TCP SYN flood, selective forwarding, wormhole, HELLO jamming, data modification and replication.

To evaluate the IDS system, they used several metrics as detection rate, accuracy and CPU usage, the performance of the system results shows 91%, 100% and 0.19% respectively.

**Commercially IoT IDS solutions.-** Several commercial IoT IDS solutions are available. For example, Palo Alto Networks offers a robust IoT security solution that integrates IDS capabilities into their broader security platform; this solution automatically identifies and classifies IoT devices, uses signatures and behavioural analysis to detect and prevent threats, and enforces zero trust security based on device identity and behaviour [103]. Fortinet's IoT security solution integrates IDS capabilities within its larger Integrated Security Fabric. This solution combines network, endpoint, and cloud security to offer advanced threat protection through signature-based detection, behavioural analysis, and machine learning [104]. Ericsson IoT Accelerator provides a comprehensive IoT security solution that includes Intrusion Detection System (IDS) capabilities to protect IoT deployments. This solution leverages advanced threat detection and real-time monitoring to identify and react to security threats across connected devices. Ericsson's IDS focuses on anomaly detection, which helps in recognising deviations from typical device behaviour, thereby identifying potential security breaches [105].

## 3.4 Localisation

In this section, we discuss why we do monitoring in IoT localisation systems. While looking at different IoT monitoring systems similar in nature to our detection solution, we find that positioning metrics are one of the key pieces of information that might be gathered from devices to perform monitoring. Localisation determines the physical position of devices in a network, providing critical contextual information. Furthermore, this spatial awareness helps understand the context of network events, detect anomalies, and identify potential intrusions more accurately. For example, knowing whether a usually stationary device has moved unexpectedly can indicate unautho-

rised activity or data spoofing.

An IoT localisation architecture could be very dynamic and evolves significantly over time in terms of the number of devices, mobility and environmental conditions where they are located. It is essential to conduct monitoring for localization purposes, where signals and communication parameters are observed to accurately determine node position. A localisation monitoring solution is required for several reasons, the expansion of the systems, the changes in the network infrastructure and more precisely for the operation and maintenance of the IoT nodes that constitute the system. These IoT devices are subject to long periods of useful life and numerous changes in the environment, in some cases very hostile.

An accuracy localisation solution will allow us to efficiently act when we want to find out where the nodes are located in a specific time instant required.

A localisation system can be classified according to the following attributes: Architecture Type, Measurements, Analysis Method, Monitoring Attacks and Performance Metric. Figure 3 shows the proposed Localisation scheme.

### 3.4.1 Architecture Type

Localisation systems might be classified according to how the nodes estimate the distances and exchange the neighbourhood information to derive their position. It can be distinguished four types of approaches: centralised, decentralised, cooperative and non-cooperative.

In the centralised scheme, the sensor nodes send information to a central processing unit. Centralised algorithms accept very high communication costs, which increase directly as the size of the network. Usually, these algorithms are more complex because they avoid the issue of constraint resources IoT node, sending the information to a powerful central unit where all calculations are done into it, instead of in each node itself. Some of these algorithms can calculate the shortest path, which could be the smallest number of hopes in order to determine the coordinates of each node.

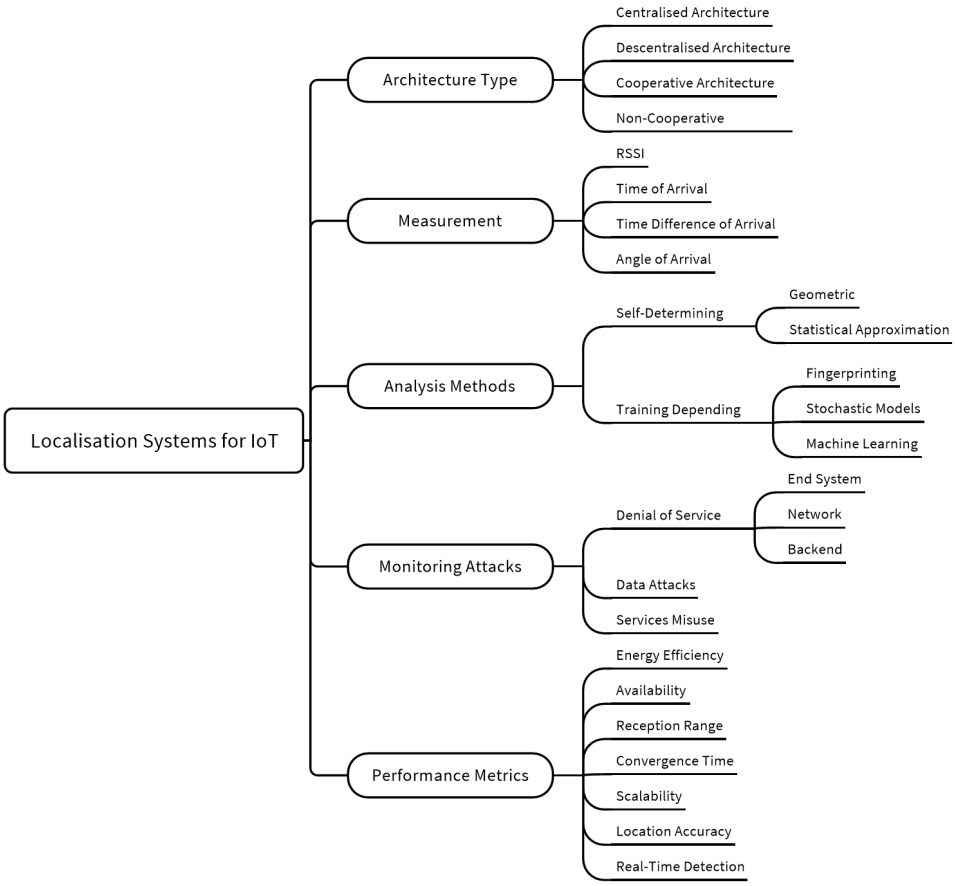



Figure 3.2: Localisation Scheme for IoT

In the decentralised scheme, all nodes communicate with each other in order to shared information such as signalling, neighbours or location parameters, etc. The algorithm in each node should be able to determine its own position as well as their neighbours' position without any central unit. For large-scale IoT networks is recommended to use the distributed location systems, because the algorithms are not complex and require less communication costs in terms of routing information to exchange.

In the cooperative scheme, the nodes could be programmed in a way that they transmit special packets or data that help with localisation purpose, and data could be not only its own information but they could transmit signals of other users. On the other hand, a non-cooperative scheme is when the nodes do not help with any measurement transmissions to estimate localisation coordinates.

These criteria might be combined. There may be centralised cooperative or non-cooperative model, as well as, decentralised cooperative or non-cooperatives model.

### 3.4.2 Measurement

During the monitoring of Localisation systems, there are several parameters that we want to look at and follow up. Localisation algorithms estimate the coordinates of the sensor nodes using the following parameters information: Received Signal Strength Indicator (RSSI), Time of Arrival (ToA), Time Difference of Arrival (TDoA) and Angle of Arrival (AoA).

RRSI measurement represents the relative quality of a received signal on an IoT device; it indicates the power level received after any possible loss at the antenna or physical level. RSSI is based on a signal attenuation, the principle of its operation is the signal strength measurement, and it does not require special hardware for operation. The higher the RSSI value, the stronger the signal.

ToA is the absolute time instant when a radio signal originating from an IoT node transmitter reaches a remote IoT node receiver. The distance from the reference IoT node transmitter can be calculated using a straightforward equation and knowing the inputs parameters which are the exact time that a signal was sent from the target and the exact time the signal arrives at a reference point.

TDoA does not require the time that the signal was sent from the IoT node transmitter to calculate the distance, only the time the signal was received at two references points.

AoA is the method for determining the direction of propagation of a radio-frequency wave; it represents the direction from which the transmitted signal is received.

### 3.4.3 Analysis Method

The analysis method tries to define the techniques used to determine the location of one IoT node. The training technique is the crucial element to classify the analysis methods hierarchically. Two main categories have been compiled: a self-determining method based on a No Offline approach and a Training Depending method based on an Offline approach [106]. Additionally, we have included Measurement techniques [107].

In the offline training models, the artificial agent learns to complete a given task by analysing a fixed dataset, they do not actively interact with a virtual environment, and they do not allow agents to learn actively in real-time. While in the no offline models, the agent that was never exposed to the task, it is designed to complete learning by interacting with an online virtual environment.

Self-Determining methods might consider two different approaches. First, the Geometric approach that exploits the geometric architecture of the network to optimise accuracy and energy consumption in IoT nodes localisation. Here, geometric information such as multilateration, centroid, angulation, energy attenuation, region overlap or connectivity among sensor nodes is used to estimate the location. The second approach considers the Statistical Approximation techniques, which uses standard approximation methods to estimate location such as: multidimensional scaling, least-square estimation, semi-definite programming, maximum likelihood estimation, hyperbola-based localisation, area-based localisation scheme, collaborative localisation, multi-stage localisation, large-scale localisation scheme, underwater sensor positioning, etc. The accuracy of each method will depend on the range of measurement.

Regarding the Training-Depending methods, they are performed into two phases, first the offline training phase before the online localisation phase. They might consider three approaches: Fingerprinting, Stochastic Models and Machine Learning Approaches. Fingerprinting methods use a signature and match it with a set of geotagged signatures to find the position of the device. Several Fingerprinting techniques

can be implemented, such as signals patterns, collaborative patterns, visual analysis, motion analysis, etc. The selection of a specific algorithm will depend on the power capability and computing power of the IoT device; therefore, the accuracy and complexity of the algorithm will depend on these two factors as well. Stochastic Models through modelling with Markov Decision Process are a novel optimisation method for decision making in scenarios where we want to model the interaction of the nodes with the environment. For localisation purposes, several different extensions of finite time and infinite time Markov models can be used: Partially Observable Markov Decision Process (POMDP), Markov Decision Process (MDP), Hierarchical Markov Decision Process (HMDP) and Constrained Markov Decision Processes (CMDP). They constitute complex algorithms; however, the POMDP technique is suitable for applications like multiple object tracking, health and body monitoring due to its low complexity and centralised applicability.

Several artificial intelligence techniques promised better estimation of position. There is plenty of variations of artificial neural networks (ANN) which uses a distance measurement matrix to improve the IoT device localisation.

### 3.4.4 Monitoring Attacks

Even though LBS can be very useful, these benefits come at the cost of users' privacy and security. We classify the monitoring attacks in three main categories: denial of service, data attacks and service misuse.

Localisation can be affected by Denial of Service attacks when an attacker can easily interfere with a physical channel to disable the localisation process through jamming or attacking critical nodes such as anchor nodes which will be compromised or destroyed the localisation process.

Regarding Data Attacks, these might target both range-based and range-free schemes. In the range-based localisation schemes, through distances or angles inputs manipulation, an attacker can make a node appear closer to or farther away from an-

other node. For instance, in ToA/TDoAbased localisation techniques, a compromised node can delay the reply to appear farther away from the sender, or send the response before receiving other nodes' beacon messages to appear closer. Furthermore, based on ToA/TDoA or signal strength parameters, a node could get an imprecise propagation time and signal strength, causing imprecise distance estimation. In the range-free localisation scheme, in order to change the network topology, a malicious node can adjust its transmission power that results in an impressive hop counting and proximity. Services Misuses, refers to the attacks to the backend localisation monitoring systems.

### 3.4.5 Performance Metric

Localisation systems might be performance evaluated from the perspective of energy efficiency, availability, reception range, latency, scalability, and location accuracy [108]. Localisation systems that consume a lot of the energy of the user devices and drain its battery might be not widely used. The energy consumption in the localisation system should be minimised without compromise the latency, periodicity for reference signal transmission and the computational complexity of the algorithms. To achieve a wide-scale adoption of the systems, it is essential that the solution will be readily available on the final user devices without requiring property hardware.

Reception range should have a reasonable value which means that the number of anchor points required is small, and the interference and performance degradation that will increase with the distance between transmitter and receiver should be a guarantee to be avoided. Localisation solutions are real-time systems which means that they should be able to locate a device with the small number of reference signals and perform complex operations with milliseconds granularity. Scalability will depend on the application and the scale of deployment, and it is essential to limit the impact of multipath effects and other environmental noises to obtain highly location accurate systems.

### 3.4.6 IoT Localisation Examples

In this section, we will outline recent implementation examples in the different categories presented above. We summary three recent examples which address the target localization problem illustrating different types of monitoring localisation architectures; they manage distinct detection methods and that have been applied and evaluated using diverse security threats and performance metrics.

**Choi et al. [109]** propose *LoRadar*, a passive packet sniffing framework for LoRaWAN, the Medium Access Control–MAC protocol in LoRa. LoRa is widely implemented due to its use of unlicensed frequency band, low cost and ease of deployment. Furthermore, the packet sniffing solution help to identify a significant amount of security and privacy vulnerabilities.

The architecture type implemented in LoRadar considers a centralised scheme as well as a cooperative approach. On the top of the LoRa physical layer, LoRaWAN implements a star network topology for the communication between gateways and sensors. LoRadar is a special application of an offline gateway that listens to the messages without relaying them, and the network server forwards the relayed messages to the application server. The offline LoRaWAN gateway is able to log all the messages sent by the nearby LoRa sensor, and capture packets send by the LoRa sensors in the neighbourhood of LoRadar, facilitating the extraction of the key information of the captured packets. The solution is unable to read the packet payload because data is encrypted. However, it can read all the information in the packet header that allows extracting a significant amount of wireless link quality-related parameters and deployment statistics. Furthermore, due to LoRa nodes broadcast messages at the gateway in the range listened and in the same frequency band that is able to pick them, so it could be stated that the solution considers a cooperative scheme.

Regarding the Measurements and Analysis Method implemented, for the localisation purpose, LoRadar assume a scenario where it is required to estimate the location of a malfunctioning device. An RSSI based localisation method is used to solve this

issue. In order to measure the distances from one node to the beacons, they use the propagation loss model, where at least three beacons are received at different locations to use the trilateration method to estimate the location of the node.

In addition to providing statistical information, LoRadar allows a deeper analysis and several use cases with the captured data, one of them is the Radio Jamming Detection. Radio jamming aims to disrupt the information flow through the transmission of high-power radio signals which decrease the SNR in the targeted frequency of communications. The authors deliberately generate a situation that interferes with the LoRa data transmission; in order to identify the presence of a triggered jamming a particular device's trace should pass through several filters: mean RSSI comparison, channel distribution inspection, drop-rate comparison and high transmission interval standard deviation identification.

The performance metric used is location accuracy; the model results show that the node location is estimated with around 10 meters of error.

**Bissett et al. [110]** implements a novel localisation algorithm for LoRaWAN networks called *MLD-Multilateral Dissection* algorithm, which is based on the TDoA of LoRa transmissions. The proposed localisation algorithm was tested over a large geographical urban area. Bissett states that in order to guarantee the accuracy in the synchronisation of gateways, it is considered that the best option is to use a proprietary and nation-wide coverage LoRa network. The solution uses a cooperative scheme due to each node is transmitting data to the LoRa gateways; the model used multiple nodes gathering data from many locations.

Regarding the measurement that the solution used to gather data, TDoA parameter is considered. TDoA does not require the transmitter to be synchronised with the receivers. It does not require the exact distance from a device to each anchor point, instead only the differences in distance from each gateway to the devices are required, it means that the model needs the differences between the timestamps of a transmission. When a LoRa signal is transmitted from a device to several gateways, these gateways

are considered the anchor points because its locations are known. Each gateway received the LoRa transmission at a different instance in time, this because each gateway will have a slightly different distance to the device. It will generate one measurement for each possible pair of gateways. Furthermore, the model considers that the position of the transmitter can be calculated using the time difference from all the possible gateway pairs and with the signal received by at least three gateways.

Bissette developed a weighted-centroid localisation algorithm that uses timestamps for weighting. He stated that the main contribution of the research is the novel Localisation technique that they created, which is called Multilateral Dissection. MLD is an area-based technique that uses TDoA measurement.

The model has been evaluated in a large area enough to experience the effects of long-range communications. The performance evaluation is doing with several adaptions of the model; they used the asymptotes of a TDoA hyperbola and the non-linear Least Mean Squares with Stochastic Gradient Descent. The location accuracy results show that MLD has 80% of cases were located to within 925m of the node's location.

**Janssen et al. [111]** propose a real-world RSS-based outdoor NB-IoT localisation system. They faced the challenges of working with real measurements in a city-scale environment. In their work, they investigated the accuracy of NB-IoT localisation, using the signal strength to a single NB-IoT antenna. The model is evaluated in three outdoor RSS-based algorithms: proximity, ranging and optimised fingerprinting.

The architecture type implemented used a Cooperative approach. One of the model's major issues is defining the amount of participating base stations during the real-world RSS measurements because the number and placement of them significantly impact the location accuracy. Hence, they evaluated different NB-IoT chipsets' behavior from manufacturers as Altair, Fibocom, HiSilicon, Huawei, Intel, Mediatek, Qualcomm, Quectel, Sequans, U-Blox, and more, in an outdoor localisation accuracy based on the received signal strength to a single NB-IoT base station. They were able

to collect 1307 messages from the NB-IoT devices with cell ID and the RSS information.

The measurement monitored is the RRSI; they developed three outdoor RSS-based algorithms. The proximity algorithm considers that the estimated location of the transmitter is the location of the eNodeB with the highest signal strength. The ranging algorithm translates the RSS of a specific base station to a distance employing a path loss model, and the authors used four path loss models: the Hata model, the COST-231-Hata model, the 3GPP Macro-cell model, and the 3GPP Pico-cell model. The optimised fingerprinting algorithm is better explained in the next section of the analysis method.

Regarding the analysis method for the optimised fingerprinting algorithm, they used a hybrid approach between Self-Determining and Training Depending approach. In the fingerprinted algorithm, a training base is created during the offline phase, and its entries consist of a vector of signal strengths to nearby base stations. While in the online phase, an evaluation fingerprint is being matched to the fingerprints stored in the training database. This matching procedure calculates the distance between the evaluation sample and each training sample in signal space. Then, in order to estimate the device location, the centroid of the ground truth locations of the k nearest neighbours is chosen. Regarding the Machine Learning techniques and to optimised the distance and RSS, they implemented an optimal parameter set in a k Nearest Neighbors (kNN) algorithm. They trained and optimised a machine learning algorithm by collecting RSS fingerprints and configuring the most optimal parameters for our outdoor city-scale test environment, for this scenario they used the combination of a lineal RSS representation, the Pearson X2 distance and k = 2.

The performance metric used is the location accuracy. Three developed RSS-based algorithms were used to evaluate the performance of large-scale localisation using NB-IoT. The proximity algorithm achieves a mean location estimation error of 340 meters and a median of 294 meters in an urban environment. For the ranging algorithm with

the four urban path loss models, the results show that the best location estimation error achieves with the 3GPP AH-macro model, the smallest mean and median localisation error of 320 meters and 259 meters, respectively. In the third scenario, for the optimised machine learning algorithm, the location estimation error results in a mean and median location accuracy of 204 meters and 132 meters, respectively.

## 3.5 Authentication

Systems that monitor individual node characteristics in order to add a layer of authentication (e.g. RF fingerprinting is a variant of this). In IoT architecture, devices process and exchange data without human intervention. Hence, IoT nodes have full autonomy, so in order to to ensure the integrity of their exchanged data, they need to recognize and authenticate each other [112]. Monitoring authentication process and systems avoid that they will be the target of malicious users and malicious use.

The even growing size of the IoT networks and its wide adoption in critical areas such as health care, military and smart cities demand that the new developments monitor and protect the data integrity, confidentiality and availability. Security issues in IoT become critical because the devices are exposed to privacy attacks which might allow severe implications if the sensitive information is revealed. Do monitoring in authentication process is an essential factor because it will guarantee trusted communication and a secure access to the IoT devices. An authentication system can be classified according to the following attributes: Architecture Type, Measurements, Analysis Method, Monitoring Attack and Performance Metric. Figure 3 shows a schematic overview of the proposed Authentication model.

### 3.5.1 Architecture Type

Authentication refers to the process that identifies an entity, it could be users or devices, in the networks [113]. Based on the architecture type that has been developed, authentications systems might be classified according to how the entities authenti-

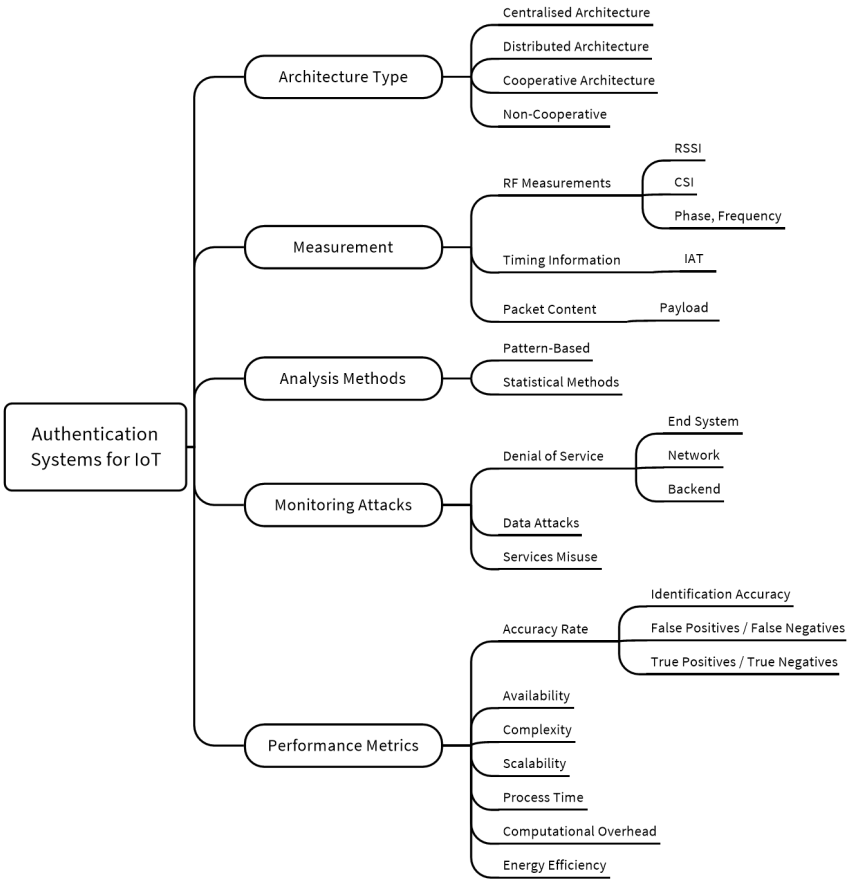



Figure 3.3: Authentication Scheme for IoT

cated one with each other. It can be categorised into four domains: centralised, distributed, cooperative and non-cooperative schemes.

In the distributed scheme, two communicated entities are authenticated using straight authentication.

In the centralised scheme, the communicating entities used a third trusted centralised entity which manages the credentials between the entities for the authentication procedure.

In the cooperative scheme, nodes share or transmit special packets with information that help to the authentication purpose. While in the non-cooperative scheme, the IoT devices do not send any measurement to identify an authorised device that wishes to communicate to it.

### 3.5.2 Measurements

Authentication requires processing data that can be described by a set of features.

In this section, we want to explain how a monitoring system can exploit the variations in the RF chain of radios to identify and to authenticated transmitters IoT uniquely. The unique imperfections generated in digital-analogue converters and power amplifiers are utilised to authenticate legitimate nodes. An RF Measurement is extracting by sampling a radio signal from an emitting device, usually called the Data Acquisition phase. The second phase is the Signal Processing to eliminate noise and other variances of the signal. The third phase is the Feature Extraction, where the features used to distinguish the signal are extracted from the radio signal. Finally, a Dimensional Reduction phase is performed in order to classify the specifical attributes of the radio signal [114].

Authentication algorithms consider several parameters that need to be monitored; these can be divided into three categories: Radio Frequency Measurements, Timing Information and Packet Content.

RF Measurements considers several parameters such as the Received Signal Strength Indicator (RSSI), the Channel Frequency Offset (CFO) which is extracted from Channel State Information (CSI), Phase and Frequency.

Timing information refers to the timing features or parameters of the RF signal analysing in the exact moment when they have been transmitted. For instance, it could be measured the time between two transmissions, and this constitutes a unique measurement per node. Several parameters have been considered in this category such as Packet inter-arrival time (IAT), Permutation entropy, Time interval error (TIE), Permutation Entropy and Clock Offset. Packet IAT tries to analyse the clock skew. IAT is defined as the amount of time that elapses after the receipt of a packet until the next packet arrives, it cannot be easily obfuscated or manipulated. Time interval error (TIE) is the time deviation from the ideal clock measured from a reference point.

Regarding the Permutation Entropy, it is a signal analysis method based on nonlinear theory; it can extract and amplify tiny changes in time series, it can fast detect the dynamic mutation in complex systems. Clock Offset is a delay of a given clock source, and signals have been used to see what is the clock drift.

Furthermore, the Packet Content of the transmission can be captured in order to find data like sensor data from any device which are attached to the IoT nodes. Temperature, pressure, flow level, and others can be captured to try to authenticate a node based on what it has seen in the payload rather what it has seen in the behaviour of the node itself.

### 3.5.3 Analysis Method

The analysis method defines the techniques implemented to process the selected measurement in order to authenticate an IoT device. Two categories have been considered: Pattern-based and Statistical Methods.

One way to authenticate an IoT node is by seeing a particular pattern in the data that is observed or monitoring, information that could be considered into this category is signal patterns, timing information or sensor data. Here, the IoT node entity proves its identity using physical characteristics or some behaviour pattern.

In the statistical methods, once the signals are received, it can be implemented Machine Learning or any other training technique for identifying legitimate users.

### 3.5.4 Monitoring Attacks

In this section, we describe the attacks to authentication systems from the monitoring perspective. Hence, the most probable attacks have been classified into three categories: Denial of Service, Data Attacks and Services Misuse.

In IoT networks, communications between the devices may face the attacks like Sybil attack, wormhole attack and Denial of Service attack. In contrast, the monitoring

system may face attacks that affect the backend systems such as impersonation attacks, reply attacks and others.

### 3.5.5 Performance Metric

Performance metric s are required to determine the adequacy of the authentication system. The following performance metrics can be used: accuracy rate, availability, complexity, scalability, process time and energy efficiency.

Regarding the accuracy rate, it describes the number of nodes that have been classified in specific classes; four evaluation metrics can be computed: False positive (FP) and False negative (FN) which represent the number of authenticated users rejected and legitimated respectively, but they were predicted as legitimate for FP and as an impostor for FN. Furthermore, it has been defined True Positive (TP) and True Negative (TN), which represent the amount of authenticated users that should be legitimated and rejected respectively, and they were predicted as such. Other indicators can be calculated from these four such as False rejection rate (FRR), False acceptance rate (FAR), Equal Error Rate (EER), True rejection rate (TRR), True acceptance rate (TAR), Recall (or Sensitivity) and Precision (or Confidence) [115] .

The performance metrics related to availability, complexity, scalability, process time and energy efficiency refer to the performance of the algorithm executed in each authentication solution.

### 3.5.6 IoT Authentication Examples

In this section, we summary three recent examples which address the target authentication problem illustrating different types of monitoring authentication architectures, analysis methods and performance metrics.

**Yu et al. [116]** propose a novel and practical CSI-based and device-to-device continuous authentication scheme for the Internet of Things. The research explores a lightweight device fingerprint extraction and verification algorithm using Channel

State Information (CSI). Continuous authentication can repeatedly authenticate the legitimacy of a user or a device in a transmission session. The solution considers a multi-feature CSI-based device fingerprint, which is difficult to spoof, and it does not require any additional hardware. Furthermore, they propose a novel fingerprint verification algorithm based on multi-feature CSI-based fingerprint; the algorithm is able to complete the fingerprinting extraction and verification within one microsecond, which means that the solution can authenticate the legitimacy of both devices without no effect on the real-time nature of the sensed data.

The architecture types implemented in the model are Cooperative and Centralised schemes. The first, because it is supposed that sensors nodes and gateways can achieve CSI from raw data frames. Besides, the model considers the gateways as central units to manage the fingerprint tuple. In the initialisation phase, the model artificially transmits large numbers of data frames between each pair of gateways and sensor nodes. The gateways apply the fingerprint extraction algorithm and stores the tuples extracted from the sensor nodes in its database. In the same way, the gateways transmit their own fingerprint tuples to the sensors node that they manage. Therefore, every gateway knows the fingerprint of the nodes that it manages, and every sensor node knows the fingerprint of the gateway that manages it.

Regarding the Measurement used, the solution considered a multi-feature CSI-based device fingerprint that is composed of three values (A, d, B), which constitutes a fingerprint triple. A is the average slope of all stripes, d is the average value of distances between every pair of adjacent stripes, and B is the intercept of the initial stripe with the smallest positive value. Besides, after extracting the fingerprint with the CSI-based algorithm and get the tuple (A, d, B), the model calculates their inter-arrival time and phase difference in order to determine if the fingerprint verification is successful or not.

About the Analysis Method implemented, the model used is a hybrid approach between the Pattern-based and the Statistical Methods. Since they used the multi-

feature CSI-based device fingerprint, they implemented a fingerprint pattern-based extraction algorithm to extract this fingerprint from Channel State Information (CSI). Furthermore, they developed an algorithm which is divided into four phases: the selection of a highly dense region, the extraction of stripes point sets, the fitting of a linear function and the calculation of the fingerprinting tuple. For the first phase, one of the major concerns is that the inter-arrival times of every pair of adjacent frame changes all the time; however when the network environment is relatively stable, the network transmission rate will change in a fixed range. Thus this generates specific regions that contain more points and clearer stripes than other regions, due to inter-arrival times are in a certain range. They used a sliding window algorithm to select the high and then extract the fingerprint tuple in it. The second phase considers the extraction of more accurate point sets of stripes, filtering the noise points. The Density-based Spatial Clustering of Applications with Noise (DBSCAN) [23] is used to achieve this. The third step is to fit the corresponding linear function of each stripe. To achieve this, the model used the least-squares method to find a satisfactory linear function. Finally, several equations are implemented in the algorithm in order to extract the fingerprint tuple (A, d, B). The algorithm only needs two adjacent frames to complete the verification process

For the Monitoring Attack, in the proposed scheme, the authors focused in the identity-based attack, where the adversary is able to collect the legitimate identity of the IoT node and then masquerade in order to pass the authentication process as the legitimate device. In this scenario, the attacker can obtain legal permissions and can perform subsequent attacks, making this type of attacks very harmful.

Regarding the evaluation performance of the model, two metrics have been considered: time consumption and accuracy rate. The time consumption refers to the average of the consumed time for extracting CSI from raw frames which costs 1.2316 ms; calculating inter-arrival time and phase difference costs 0.0759 ms, and the verification of the fingerprints costs 0.0114 ms. About the accuracy rate, two reference points have been considered: the probability of success in the IoT sensor nodes and in

the adversary devices, which results in 92,98

**Jiang et al. [117]** propose an identification method for LoRa devices based on physical layer fingerprinting using constellations trace figure. Constellations are established from the radio frequency fingerprint features of LoRa devices. The model used a classification method based on Euclidean distance to analyse the differential constellation trace figure. The performance results show that the solution can recognise the different LoRa devices accurately. The trace of the constellation trace figure reflects the unique physical characteristics of the IoT device.

The model architecture type used a Cooperative approach. The solution follows the general LoRaWAN network architecture which consists of an end device, a gateway, a network server, an application server and a joining server. The messages are transmitted from the IoT nodes to the network server through the gateway.

RF Fingerprint is considered in the proposed model. The features extracted from the received signals for identification are divided into two categories the transient part and the modulation part. The transient part measures the turn on/off of the signal or the transmitting signal variations. In contrast, the modulation part measures the frequency, phase, amplitude and I/Q samples of the entire signal.

Regarding the analysis method implemented, the model considers a hybrid approach between pattern-based and statistical schemes. Pattern-based due to the model pre-process the collected data, where for each data sample, the valid signal in it need to be extracted and then do normalisation. Then another pre-processing operation is performed in order to follow with the RF fingerprint transformation execution to extract the device-related features. The Statistical schemes are implemented by analysing the characteristics of the differential constellations trace figures with the image recognition algorithm and classification method based on Euclidean distance of clustering centre of LoRa signal.

Several monitoring attacks try to be tackled with this method. In the physical layer, the authentication process is considered simple, and due to it does not require any

information interaction, it reduces the risk of protocol attacks. Therefore, some vulnerabilities are caused because LoRa technology uses the unlicensed frequency bands at the public protocol. So, LoRa devices are usually configured with the pre-allocation key that might cause some attacks such as eavesdrop attack on the address of the legal terminal that could generate forged packets to the gateway which can produce congestion. In the same way, if the root key is stolen, the communication can be completely cracked.

The identification accuracy is evaluated in this research. The experimental results indicate that six LoRa transmission modules were recognised accurately from the differential constellation trace figures, even in a low SNR environment. The proposed method has a certain resistance to channel noise interference. The results show that the identification accuracy of the six LoRa module is 100% even when the SNR environmental condition is higher than – 10 dB. The model shows that the system identification accuracy begins to decline when SNR decreases.

**Baldini et al [118]** propose a model that authenticated IoT devices through their RF fingerprinting, specifically the entropy parameter based on two statistical features: the permutation entropy and the dispersion entropy. The model demonstrated that the IoT devices were classified and authenticated with great accuracy.

The architecture type considers a Cooperative approach because the design of the model considered nine IoT nodes configured to transmit a fixed payload to MySensor network, which is an open-source software framework for IoT.

The primary Measurements used are the Permutation Entropy (PE) and the Dispersion Entropy (DE). However, the research defined another basic set of features for performance evaluation purpose that considers measurements such as variance, skewness, kurtosis, Shannon entropy, and log entropy. The real-valued signals were sampled, synchronised and normalised offline to extract the burst of the traffic of each payload. The new/basic set of statistical features were applied to each payload in order to generate the fingerprints from the RF signals.

Regarding the Analysis Method applied, the research considers the Statistical methods. Initially, the statistical features were used to create a training matrix, which is constituted for ten statistical features, 900 payloads and nine IoT devices, resulting in 81000 values. Then, several machine learning algorithms were applied to the defined matrix, using the different set of features to authenticate the IoT nodes. The Machine Learning techniques considered are KNN, SVM and Decision Tree.

Regarding the Performance metric, the model evaluated the authentication accuracy in 27 different scenarios considering the combination of the application of different nine sets of features and three different machine learning algorithms. The scenario that considered the basic set of features with the SVM algorithm shows an authentication accuracy of 49%, which means is a random choice. Furthermore, the results show that the use of PE and DE significantly improves the authentication accuracy in contrast with Shannon and log entropy features, the results are 81.9%, 82.1% and 81,3% for KNN, SVM and Decision Tree algorithm respectively; which means 24%-30% higher than the basic set. Finally, it is stated that an outstanding improvement of accuracy was obtained using PE alone.

## 3.6 Interference Monitoring

A major problem in IoT deployments is interference. Interference may origin from other networks deployed in the same space, electrical machinery or even from a nefarious actor that aims to disturb communications. When an IoT deployment uses an unlicensed frequency band other devices may communicate in this space as well. Consequently interference will occur.

In a simple scenario another IoT network of the same type is deployed in the same physical space. In this case, two networks managed by different providers have to share one frequency space. It is not always the case that these two networks are deployed in a cooperative fashion to minimise interference. An example here would be two independent LoRa networks deployed in the same physical space by two differ-

ent providers. Therefore, monitoring of the interference situation might be necessary as a network provider must be aware of the interference situation in order to tune network parameters to deal with it.

In many cases the competing devices use an entire different communication protocol and thus, are not even designed for coexistence. An example here is an 802.15.4 network which may have to coexist with WiFi networks in the deployment space. The communication protocols used for 802.15.4 are usually not designed to coexist with WiFi and similarly, WiFi is not designed to coexist with an 802.15.4 network. A network provider may want to monitor the interference situation in order to take appropriate action when this becomes an issue.

IoT networks are often used for applications that require strict performance guarantees. For example IoT networks may be used in factory automation and control. Interference by machinery in the deployment area can have significant impact on performance. Hence, interference in these situations is often monitored such that appropriate action can be taken to meet performance requirements.

Finally, a malicious person might use interference to attack a system and prevent IoT operations. For such scenarios it may also be necessary to monitor interference in order to locate and remove a threat.

To handle the above situations, interference monitoring systems have been used. An Interference monitoring system can be classified according to the following attributes: architecture type, measurement type and timing, analysis method, and performance metric. Figure 3.4 illustrates a schematic overview of the proposed Interference monitoring model.

### 3.6.1 Architecture Type

To measure interference it is necessary to use a number of measurement probes. It is either possible to use dedicated hardware to perform interference measurements or to use the communication devices to also collect data for interference monitoring. There

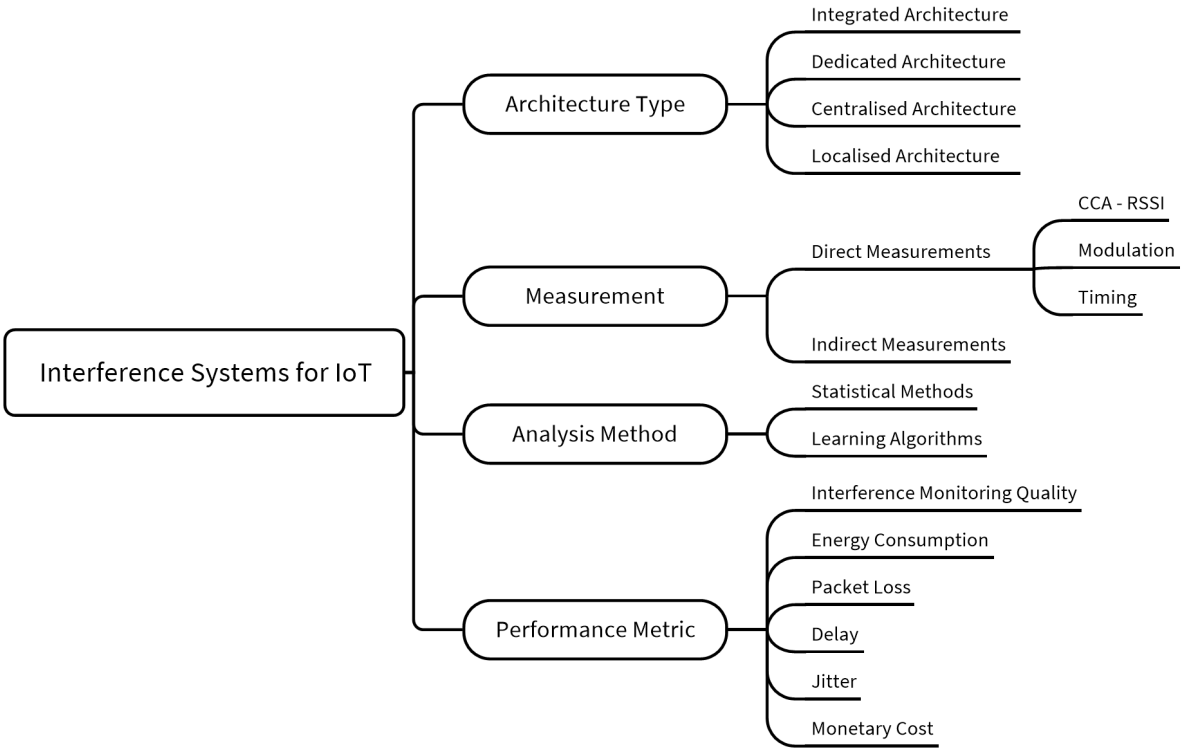



Figure 3.4: Interference Scheme for IoT

are advantages and disadvantages with both architectural choices.

With the *integrated* approach the communication transceivers of IoT nodes and base stations can be used to collect interference measurements. This saves considerable cost in terms of hardware and also reduces deployment cost. However, if a device is carrying out an interference measurement it is not possible to communicate. Thus, interleaving of interference measurement and communication tasks is challenging. Also, it is not possible to measure interference during a transmission (Which is usually the time when a measurement is of particular interest).

With *dedicated* measurement hardware it is possible to simplify interference measurements as communication and interference measurement do not have to be scheduled together. However, this leads to an increased system cost (for hardware and deployment). In addition, the device to measure interference is not located at the ex-

act position of the communication device which leads to measurement inaccuracies.

A further architectural choice is to carry out interference monitoring in a centralised or localised fashion. In a *centralised* architecture all measurement data is transported to a central point where it is analysed and decisions are made, for example on how to change network configuration to deal with the observed interference pattern. The disadvantage of this approach is that all interference data has to be transported to the central point. This means that either a dedicated communication channel for interference data is necessary or if this data is transported over the same communication link as network data a scheduling mechanism is necessary. The advantage is that a complete picture of interference in the network can be composed at a central point.

In a *localised* architecture the interference data is analysed locally and adjustments to communication devices are made directly. This simplifies the architecture as measurement data does not have to be transported over longer distances. However, the disadvantage here is that no complete interference picture of the network is available.

### 3.6.2 Measurements

To measure the level of interference directly (*direct measurement*) the distribution of energy over frequency and time is measured. As previously discussed, interference measurements are often carried out with the communication device itself. This limits the information that can be collected as devices are designed for communication and not specifically for interference measurement. Usually a device can measure energy in the channel it is using to transmit. This feature is normally used to evaluate if the channel is busy or not (Clear Channel Assessment (CCA)). A CCA provides an Received Signal Strength (RSSI) measurement that represents energy in the transmission channel at the time. The exact method of performing this signal strength measurement depends on the specifics of the transceiver hardware.

This measurement method provides information on how a frequency or frequency band is used over time. However, this often does not provide sufficient information

to determine the nature of the interfering system. However, it is often beneficial to have this information as it can provide information on how to co-exist best with the interfering system. For example, when interference is detected a node may delay a re-transmission. The back-off time should take into account the behaviour of the interfering system; for example, an interfering WiFi network will behave different than an interfering Zigbee network.

To obtain more detailed information it is possible to analyse an interfering signal more detailed. One common way is to analyse the *modulation* scheme used.

When using the communication device itself for this type of measurement it is usually only possible to asses if the interference is of the same type as the measurement device or not. Thus, to evaluate the type of the interfering system other mechanisms are required. One used approach is to observe *timing* and structure of the interfering signal and to deduce from these observations the nature of the interferes.

Instead of direct interference measurements it is also possible to use an *indirect measurement* of interference. For example, it is often assumed that lost transmissions are due to interference and in this case the pattern of loss allows us to indirectly measure interference. While such method is not as accurate as a direct measurement it is simpler to perform and often provides sufficient information on the interference situation.

### 3.6.3 Analysis Method

Data analysis can be carried out using two main approaches: *Statistical Methods* and *Machine Learning*.

Interference measurement traces are generally used to instantiate a statistical model describing interference expected at a specific location and time (i.e. in a specific communication channel at a certain time). Depending on the aim of the interference monitoring, models providing different levels of detail can be used. For example, a model may describe the expected interference strength over time while a more simple

model may only describe the expected duration of an interference signal (assuming interference above a relevant threshold).

Machine learning approaches have been specifically used in situations where the interference type is determined using the observed timing and pattern of interference. Known interference measurements of a specific type are used to for training. Then an interference measurement trace is used to determine with the trained model what type of interference is currently encountered.

### 3.6.4 Performance Metric

The performance of an interference monitoring system can be judged by evaluating the quality of the interference measurement it can provide and by the resources that are required to implement the system.

The *monitoring quality* can be judged by determining how accurately interference can be captured (in time and space domain). However, accuracy here is only required up to a level necessary for the analysis method chosen.

The resources relevant to judge performance depend on the chosen architecture type. With an *integrated* approach where the IoT infrastructure is used for interference measurements it is necessary to minimise the impact of interference measurement on the normal functionality of IoT infrastructure. For example, for battery powered IoT devices the use of energy necessary for interference monitoring should be minimised. The impact on normal data communication by the interference monitoring should also be minimized. This impact may be observed as increased *packet loss*, *delay* or *jitter* which can be used as metric to judge on the performance of the monitoring system. With a *dedicated* measurement system the resources required to deploy and operate this additional system can be used as performance metric.

### 3.6.5 IoT Interference Examples

In this section, we will outline recent implementation examples in the different categories presented above.

**Voigt et al. [119]**propose a model to mitigate inter-network interference in LoRa Networks. They investigated the use of directional antennae and multiple base stations as methods of dealing with inter-network interference. The great popularity of LoRa networks makes that multiple independent LoRa networks are deployed in close proximity, generating interference one with each other. The directional antennas increase the RSS without increase the transmission energy cost.

Regarding the Architecture type, the solution employs a non-cooperative approach. The signal parameters are reading by LoRaSim, which allows us to place N LoRa nodes and M LoRa base stations in a 2-dimensional space. Furthermore, they deploy a variable number of interfering networks around the network of interest, and this interfered network was deployed in the same way as the main network.

LoRa deployments usually used a default static setting to handle inter-network interference. However, the authors propose an additional mechanism for dealing with it, which consists of decoding message in the presence of interference. They analysis Receive Signal Strength through the capture effect. The capture effect occurs when two signals reach the receiver, and the stronger signal suppresses the weaker signal. Furthermore, to determine whether the receiver can decode packet or not, they use several conditions such as Carrier Frequency (CF), Spreading Factor (SF), power and timing.

The solution uses a pattern-based analysis method to detect interference, though the signal strength at the receiver, it implements two practical alternative scenarios. The first scenario considers directional antennae to improve receiver signal strength without increasing transmission energy cost. The second scenario uses multiple base stations to enhance the probability of decoding a message at least at one receiver.

To evaluate the system performance, they use a metric called Data Extraction Rate (DER), which is defined as the ratio of received messages to transmitted messages over a period of time. The experiments results show that the use of multiple base stations outperforms the use of directional antennae. In the scenario using directional antennae, the directional transmission improves DER, specifically when the number of base stations in high, here the signal strength of the nodes of the interfered network increases; on average DER increases 0.04 in this scenario. While the experiments using additional base stations show that for 600 sensor nodes, the DER increases from 0.35 to 0.53 for distances between the base stations of 50m and 67m. The solution demonstrates that adding more base stations rather than equipping nodes with directional antennae is more efficient when mitigating LoRa network interference.

**Caglar et al. [120]** propose iSensitive, an intelligent, performance interference-aware resource management scheme for IoT cloud backends. iSensitive is developed in a VM environment, where the virtual machines are classified based on their historic mean CPU, memory, and network usage features. In an initial phase, the model extract patterns to document the lowest performance interference level on a specified host machines. With this learned information, the solution classifies the VM by employing machine learning techniques. Furthermore, iSensitive demonstrate how application performance and resource utilisation are impacted by interference. As the authors state, the iSensitive solution is premised on the hypothesis that resource overbooking causes performance interference and degrades application performance. As a precedent, the solution defines three different setups: Base, Non-Overbooked, and Overbooked. The Base setup provides the best application performance because no other VM is contending for the resources. The Non-Overbooked setup is between the two scenarios respect the number of VM considered. In the Overbooked setup, more VMs are competing for the resources causing performance interference between VM.

Regarding the architecture type implemented for this solution, they use both cooperative and distributed approach. The model presents a predictive approach to address performance interference issues at runtime. Therefore, the predictive approach

requires to obtain and to learn real-world historical resource data, which implies the resources usage data for each virtual machine and the host machines in the cloud. The iSensitive distributed system can be integrated with the cloud infrastructure software. It comprises a Virtual Machine Manager (V-Man) for each VM residing on a physical host, a Host Manager (HMan) for each physical host, and a Cloud Manager (C-Man) to orchestrate the cluster of host machines in the cloud data centre.

To create an accurate model, high quality and fine-grained historical data is crucial to gain a sense of the resource usage patterns and how they change over time. Processor, memory and network usage features are used to shape the patterns.

iSensitive considers a hybrid approach between offline and online training; for this, the solution defines two modules: Interference Model Learning Module and Interference Model Execution and Monitoring Module, respectively. The offline phase outlines a Workload Accumulator component to collect and store the obtained raw data; it is fed from a phyton-based synthetic workload generator. Then, a Virtual Machine Classifier component is defined, it is responsible for generating clusters with the VMs that have a similar set of objects by employing the k-means algorithm to classify the VMs in different classes; then the silhouette method is used to decide the best number of clusters. All this is using the previous raw training data created. Next, the Model Learning via Artificial Neural Network component is used to capture the non-linear relationships between performance interference and the VMs through the propagation-based artificial neural network (ANN) technique. In the online phase, a Decision Maker component is employed to find the appropriately suited host machine with minimal performance interference to place each VM and utilise the trained model. Finally, an Interference Monitoring component compares the actual performance interference value and its predicted value to detect interference to track the system's error rate.

iSensitive evaluates the performance interference through different resources metrics, such as CPU usage, memory usage, network I/O usage, internal scheduler met-

rics, hardware and kernel-level performance counters for VMs and physical host as input to the system. Furthermore, the performance degradation in the cloud-hosted IoT application is evaluated. The experiments show that the performance improvement due to iSensitive's placement decision is 39% better for mean throughput than the commonly used strategy of first-fit heuristics. Besides, it was demonstrated that the iSensitive approach finds the aptly suited host machine with minimum performance interference level at the host-level and provides better performance to the applications running in the VM being migrated.

The authors state that their approach is generic enough and can be applied by cloud service providers in their data centres for their expected workloads and hardware platform.

**Baidya et al. [121]** introduce the notion of content-oriented cognitive interference control to current data streams in heterogeneous local access networks for urban IoT. Their work focuses on city monitoring applications with real-time traffic (video data streams from cameras systems). They demonstrate that the interference from other current applications may severely impact the performance of algorithms processing the video streaming in real-time. The traffic generated by sensing and processing systems that overlap the networks could be aggravated by the multiple scales involved.

Regarding the Architecture type, the solution uses a cooperative and distributed mode. The work implements a network scenario where two communication technologies are considered: Device-to-Device and Long-Term Evolution (LTE). It is stated that LTE with D2D communication perfectly meets the requirements of urban IoT. Therefore, having this kind of scenario might challenge to control mutual interference in such heterogeneous networks. The model chooses an architecture where an end host is transmitting real-time data on the uplink of LTE. Also, two mobile devices in proximity are connected with network assistance for D2D communication.

As Measurements, their approach considers only the long-term statistical knowledge of the wireless channel to efficiently shape the transmission process of the D2D

link; the objective is to maximise the throughput while generating an interference pattern compatible with the specific coexisting application. Specifically, they consider the LTE control messages, such as preamble parameters which indicates the type of data. Furthermore, the SINR can be measured by control signals such as Sounding Reference Signal (SRS) or data signal like Physical Uplink Shared Channel (PUSCH) at eNodeB MAC layer. Also, the solution calculates a Channel Quality Index (CQI) to report to the scheduler. The scheduler is responsible for choosing a suitable Modulation and Coding Scheme (MCS) and notifies to the UE MAC through downlink control information. If interference is detected between the communication of D2D nodes and the LTE uplink, the CQI degrades, and the LTE receiver will more likely fail to decode the PDU. An appropriate value of CQI and MCS is established to control interference.

Regarding the Analysis Method, the proposed technique uses statistical approaches to analyse the network channel coefficients' knowledge to establish a strategy that differentiates between groups of packets in the LTE video stream to improve the D2D throughput and object detection algorithms. In the novel cognitive transmission technique, the transmission power and channel access scheme of the D2D link is based on the video data stream structure. The transmission strategy is based on the observation that damage to groups of LTE packets, the transmission activity by the D2D link within a temporal period, generate a more considerable effective interference to the video quality regarding the same activity performed during differential frame transmission.

The solution measures e object detection probability of received video with respect to the reference video, which is the received signal without D2D interference. The numerical results show 20% to 30% improvement of the object detection probability in low-speed fading case, and almost 25% to 35% improvement in the high-speed fading case. Moreover, another measurement considered is thee efficiency of D2D operations with respect to the degradation caused to the video stream; results show that the model grants a significant throughput increase to applications coexisting with the video stream.

## 3.7 Performance Monitoring

Many systems look at the network traffic to spot bottlenecks and performance issues (which is a resilience issue, may also be an attack). Slightly different to an IDS but there are some similarities.

IoT allows us to simplify our daily lives and be more productive. From cars, health, military applications and smart homes and cities, the Internet of Things has a presence in everyday objects. With this deep-rooted presence, a few questions come to our minds. What will be the impact of a continually growing network for specific critical applications such as medical ones? How can the network administrator measure a particular application's correct operation in real-time to prevent or react on time in front of any failure?

An IoT infrastructure generates higher volumes of network traffic due to their size, complexity, the volume of connected devices and the new technologies with new protocols and security requirements. For these reasons, routine network management and capacity planning tasks are a complete challenge.

An IoT Performance Monitoring solution enables advanced diagnostics and data correlation capabilities in real-time. An efficient IoT Performance Monitoring system which delivers value in real-time represents a quick and efficient location of the source of the problems, the correlation of the data and the swift application of the fix solution immediately, instead of taking several hours to identify performance issues.

In this section, we want to describe several approaches for performance monitoring applications. According to the following attributes, Performance Monitoring Systems can be classified into four categories: Architecture Type, Measurements, Analysis Method and Performance Metric. Figure 5 illustrates a schematic overview of the proposed performance monitoring model.

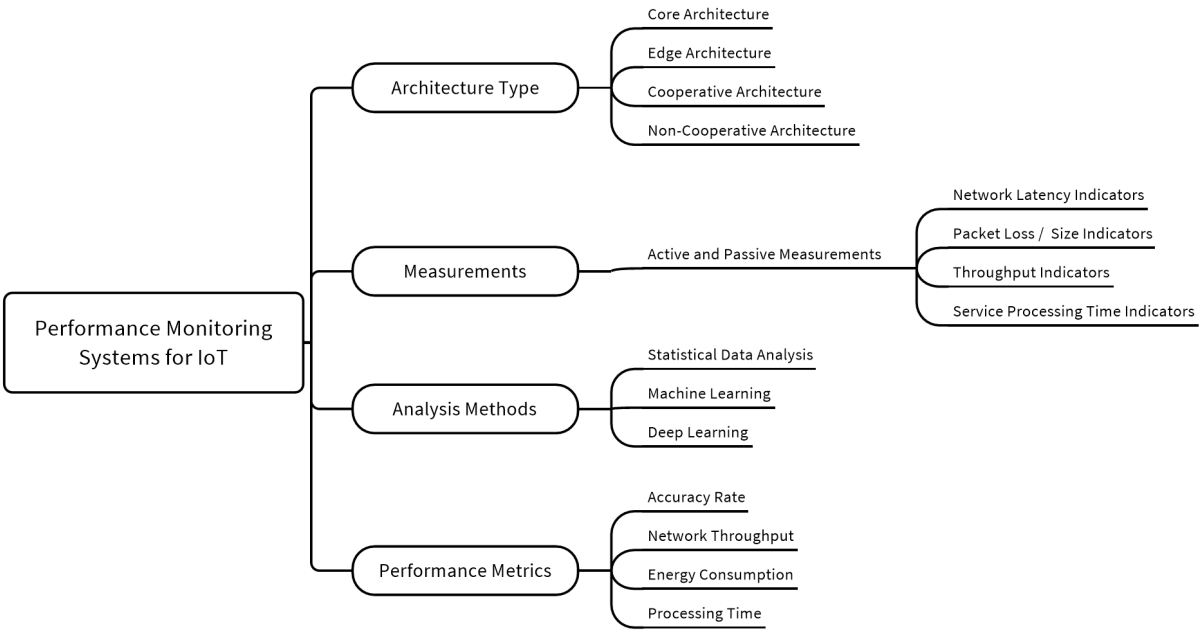



Figure 3.5: Performance Monitoring Scheme for IoT

### 3.7.1 Architecture Type

To design a sensitive IoT performance management and since one of the biggest challenges addressed for them is the scalability in data collection. Performance monitoring systems can be categorised based on the network monitor placement into the following categories: edge and core scheme. Additionally, they can be divided based on how the communication is carried out between the different network elements; here, we can consider a cooperative and non-cooperative approach.

In the *edge* architecture, the monitoring system is placed at the edge of the network, as an individual end system to capture incoming and outgoing traffic. Here, measurements are easier to undertake because IoT devices are operated and maintained by a single system itself.

In the *core* architecture, measurements are more complicated in terms of collecting traffic logs, due to the administrative, privacy and property issues; these are useful for services and applications that demand the communications through an ISP between them.

In the *cooperative* approach, IoT devices send information inputs or groups of datasets to perform the monitoring task.

In the *non-cooperative* scheme, there is additional software or hardware responsible for reading this information.

### 3.7.2 Measurements

This section aims to identify the measurements that allow us to analyse the network performance. To define specific measurements for performance monitoring is not a simple task because of several reasons that include [122]: the number of applications and services is enormous, continuously increasing and updating, these different type of service has to be identifying using and observing little information of the network, and this small information remains visible in packets with are typically encrypted. The primary goal is to achieve data acquisition at high-speeds by reducing the data volume at the collection points without impact the data quality. To achieve a correct measurement performance classification, we have been considered two categories: active and passive measurements.

The *passive* measurements are built on the continuous observation of the traffic, and their extraction is in real-time. This type of measurement introduces no additional traffic; hence, they do not perturb the network's state.

In the *active* measurements, data is injected into the network and then sees the network response. It is usually used for troubleshooting [Alconzo].

The following classification of metrics to measure performance monitoring systems is independent of how they are obtained (passive or active mode). The indicators

are as follows: network latency indicators, packet loss/size indicators, throughput indicators and service processing time indicators.

### 3.7.3 Analysis Method

Now, we want to address how to process the IoT sensors measurements for performance purposes. We aim to generate knowledgeable insight into data patterns for rapid decision-making. We have included three main categories that consider the statistical data analytics, Machine Learning and Deep Learning techniques.

It has to be noticed that a large amount of useless data might generate high computational costs which are critical for constrained IoT networks. We need to consider the IoT sensor data's main aspects and characteristics to choose the more accurate analysis method. One of the most significant property that can be addressed is real-time processing. The IoT performance monitoring solution needs to guarantee raw data transformation into more valuable and insightful information in real-time. Second, they should handle the exponential growth of IoT sensors and actuators without compromised the quality of services of a specific service. Furthermore, these methods also should be adapted to handle different data sources.

*Statistical Data Analytics* models include several techniques that will depend on the type of data required to analyse and the insights that services want to uncover. We name the more relevant for performance monitoring systems: Gaussian mixture model, Markov chains, Bayesian methods and Probabilistic matrixes.

*Machine Learning* methods can be customised to achieve an efficient data structure analysis to handle the features in real-time. They are carried out for data training in order to classify the data for a specific IoT application or service.

*Deep Learning* methods are computationally expensive regarding its execution, considering the resource-constrained IoT sensor nodes. However, they provide an excellent feature learning taking into consideration the unclear or overloaded useless information required to be made accurate through real-time.

Neural Networks are used for data processing and analysis. Artificial neural networks (ANN) are well suited to pattern recognition using supervised learning techniques. Convolutional neural networks (CNN) are suited for analysing image sensors data through a set of filters that filter the high-dimensional data to a lower dimension to extract a specific feature from the input image. The Recurrent neural networks (RNN) are able to learn from sequences or time-series data with variable length, and they are a good alternative for predictive the time series sensor data. Generative adversarial networks (GAN) are typically used for image processing applications.

### 3.7.4 Performance Metric

In this section, we want to briefly describe several performance monitoring metrics, which allow us to evaluate a system efficiently. We have considered several metrics such as accuracy rate, network throughput, energy consumption and processing time.

The *accuracy rate* metrics indicate the system's effectiveness in identifying whether performance falls within the normal ranges necessary for proper operation. Similar to the scenarios analysed in previous sections, true positives, true negatives, false positives and false negatives may result from this indicator.

The *network throughput, energy consumption and processing time* achieved by the system during a specific transmission will evaluate the system's performance while this is operating.

### 3.7.5 Performance Monitoring Examples

In this section, we present the most relevant and recent papers summarising performance monitoring for IoT research. We have explored the literature for performance monitoring surveys, restricting our focus to the last three years.

**Maldonado et al. [123]** provide an Analytical Performance Evaluation Framework for NB-IoT. The analysis evaluation includes the limitations due to the channel estimation and SNR realistic conditions. The model shows the impact on the coverage

extension in the NB-IoT user UE performance in terms of two indicators described in the *Measurements* section. The model simulation results regarding the battery lifetime for MCL of 164 dB show that it can reduce 90% in the realistic CE, compared with the ideal CE.

Concerning the *Architecture type* implemented, they use a cooperative approach. The solution is based on NB-IoT in-band on an LTE cell, where the NB-IoT UE is camping on the cell and is transferring data to the network periodically. After the end of the communication with the base station or eNB, it is considered a period of NPDCCH monitoring before the UE stays in PSM to save battery. To uplink data, the IoT node needs to re-establish the Radio Resource Control (RRC) connection between the UE and the eNB. It is essential to highlight that the research is focused only on the study of the uplink transmission.

The *Measurement* used is the data transmission rate and SNR. The combination of the following parameters determines the data rate: Modulation and Coding Scheme (MCS), Resource Units (RU) allocated to the UE, size of the data packet in bits, CRC is the size of the cyclic redundancy check code in bits, and T which is the duration in seconds of an RU. The time duration of each RU depends on the bandwidth allocated to the UE. SNR estimation, the ideal or realistic Channel Estimation (CE), and the cross-subframe CE technique are used as inputs to configure the link adaptation of the signalling packet transfers required before the uplink data transmission.

In terms of *Analysis Methods*, the model considers a Statistical Data Analytics, based on Shannon capacity fitting for NB-IoT and the estimation of the dependency of the Channel Estimation (CE error) and the SNR. The analytical evaluation framework is divided into SNR estimation, link adaptation, and NB-IoT energy and delay estimation. For the energy consumption estimation, the model uses a Markov chain and four power levels (i.e., transmission, reception, inactive, and standby) to describe the UE power consumption.

The *Performance Metric* of NB-IoT is evaluated in terms of uplink packet transmis-

sion latency and UE's battery lifetime. Using the realistic Channel Estimation (CE) compared to ideal CE, it is demonstrated that the SNR gain's performance when doubling repetitions is significantly affected when assuming realistic CE compared to ideal CE. Regarding UE's battery lifetime, the experiments' results show that for a maximum coupling loss (MCL) of 164 dB, realistic CE evaluations obtain a battery lifetime reduction of approximately 90% compared to ideal CE. For lower MCL such as 154 dB, realistic CE with cross-subframe shows a battery lifetime reduction of a 50% and 18% compared to ideal CE for IATs of 2 h and 24 h, respectively.

**Sarr et al. [124]** present a Performance Analysis of a Smart Street Lighting application using LoRaWan. The solution is a realistic implementation of a large-scale smart street lighting control system that uses solar lamppost to manage street lights dynamically. The city lampposts are remote-controlled using LoRa technology. Results simulations show a high packet loss rate due to the collision in a one-gateway scenario. Lower energy consumption is achieved when a high number of gateways are used.

Regarding the *Architecture type*, they use a cooperative approach as the model considers a LoRa-based solution. LoRaWAN is defined as a star topology, composed of three elements: end-devices, network server, and gateways. The communication is carried out through the LoRa gateways, responsible for listening all message from IoT sensor nodes.

The implemented *Measurements* include channel variability (packet loss, collision occurrence and energy consumption) and the number of gateways.

The used *Analysis Method* involves Statistical Data Analytics using the log-distance path loss model, which is ideal for modelling deployments in built-up and densely populated areas. They implement the Street Lighting System using the European regional parameters for LoRa physical layer. The deployment uses a simulation scenario based on OMNET++. Furthermore, the Adaptive Data Rate (ADR) mechanism allows setting the spreading factor and transmission power. SDR is based on SNR to set the

suitable SF. The higher the number of gateways, the small SF that nodes should use. Therefore, multiple gateways balance the load and reduce the noise.

The considered *Performance Metric* are Data Extraction Rate (DER), energy consumption, and collisions for the network. DER is defined as the ratio, over a period, of a received message to transmitted message. Network Energy Consumption (NEC) is the necessary energy to send a packet. Simulation results show that increasing the number of gateways from 1 to 2 improves DER from 20% to 11%. Moreover, energy consumption is reduced from 47% to 59% for moderate and typical channel variability. In the same scenario, the collision reduction rate decreases from 20% to 49% for moderate and typical channel variability, respectively. For more than three gateways, the improvement is not significant. Furthermore, SF equals 10 gives better results than other data rates, even contrary to the recommended LoRa parameter (SF equals 12).

**Ayele et al. [125]** provide an in-depth performance evaluation of LoRa for indoor IoT applications. They build a prototyped LoRa radio-enabled network to test network performance.

Regarding the *Architecture type*, the model uses a cooperative approach. The LoRa gateway is the network's received node designed to listen for all incoming messages.

Regarding the *Measurements*, and to test the performance of LoRa, several physical layer parameters that might impact the performance of the radio link are considered. The three physical layer parameters which influence the effective bit rate of the modulation, its resistance to noise and interference and its ease of decoding are: spreading factor (SF), bandwidth (BW) and coding rate (CR). The solution measures the expected receiver sensitivity based on SF and BW. As higher values in channel bandwidth, more decreasing of receiver sensitive. Furthermore, a higher value in coding rate results in an improvement in packet delivery ratio.

Regarding the *Analysis Methods*, the model uses a statistical data analysis approach, based on the Data Rate Optimisation feature (DE). The simulations consist of varying the three primary measurements (SF, BW, CR) in the implemented prototype to com-

pare them with the theoretical values.

The *Performance Metrics* considered in the set of experiments evaluate the coding rate and the transmission power on LoRa performance. The transmissions use packets transmitted at a fixed SF. The results show that when the data rate is decreased, the packet loss decreases due to the increase in the spreading factor.

## 3.8 Related Work

In this section, we present a brief comparison of the taxonomies introduced in this chapter with other state-of-the-art taxonomies.

Regarding Intrusion Detection Systems, Zarpelao et al. [35] present an IDS taxonomy that classifies IDS systems for IoT according to four main attributes: detection method, IDS placement strategy, security threat, and validation strategy. The detection methods include signature-based, anomaly-based, specification-based, and hybrid approaches, each with its own strengths and weaknesses. The IDS placement strategy is categorised into distributed, centralised, and hybrid placements, addressing different deployment scenarios within IoT networks. Security threats are identified and categorised to align with common IoT vulnerabilities, such as DoS attacks and routing attacks. Finally, validation strategies emphasise the importance of verifying the effectiveness and efficiency of IDS solutions through realistic simulations and testing. Although there are categories similar to our taxonomy (e.g., placement strategy and detection method), the study differs from our research due to the approach used to define the attacks; the authors describe the category *Security Threat*, emphasising the intrusion attacks that IoT devices are susceptible to, whereas our research focuses on the types of attacks that the IDS monitoring system is vulnerable to. Additionally, they do not include the category *Performance Metrics*, which is essential for evaluating the performance of the IDS monitoring system. Jamalipour et al. [126] propose an IDS taxonomy which categorises intrusion detection systems for IoT into three primary types based on intrusion data sources, detection techniques, and placement strategies. In-

trusion data sources include host-based IDS and network-based IDS, each examining different data sources to identify attacks. Detection techniques encompass signature-based, anomaly-based, specification-based, and hybrid-based IDSs, utilising various methodologies to detect and mitigate threats. Placement strategies classify IDS into centralised, distributed, and hybrid systems, determining the physical or logical deployment of IDS components within the network to optimise detection efficiency and resource utilisation. This work differs from our research as despite the authors defining a new category, *Data Source*, we consider these definitions within our *Architecture Type* category. Compared to our approach, the proposed taxonomy covers a narrow spectrum, as this study's analysis excludes security attacks and performance measurements for IDS systems.

El-hajj et al. [127] present a taxonomy for IoT authentication schemes using several criteria: authentication factor, use of tokens, authentication procedure, architecture, IoT layer, and hardware-based authentication. Authentication factors can be identity-based or context-based (physical or behavioural). Tokens may or may not be used. Procedures include one-way, two-way (mutual), and three-way authentication. Architectures can be centralised or distributed and further classified as hierarchical or flat. Authentication can occur at different IoT layers (perception, network, application). Finally, hardware-based schemes use implicit methods like Physical Unclonable Functions (PUF) or explicit methods such as Trusted Platform Modules (TPM). This study differs from our investigation as it defines a multi-criteria classification framework providing a structured approach that may limit its implementation. In contrast, important aspects such as the security attacks the monitoring system will face or the performance metrics to evaluate the solution are not addressed. Alsaaed et al. [128] proposed an IoMT authentication taxonomy that categorises and evaluates authentication methods within the Internet of Medical Things (IoMT) systems. It includes multiple perspectives: authentication levels (device-level, user-level, and network-level), authentication architectures (centralised and decentralised), and types of credentials (knowledge-based, possession-based, inherence-based, and behaviour-

based). It also details authentication procedures, distinguishing between static and continuous authentication, and various authentication schemes, including basic, key agreement, certificate-based, and cryptography-based methods, focusing on hybrid schemes that enhance security by combining multiple approaches. Our research takes a different approach than this study as our proposal considers a general taxonomy for IoT, while their work is dedicated to IoMT. Consequently, more specific categories have been included (e.g., type of credential and authentication schemes), but some essential aspects, such as the attacks and performance metrics, are not considered.

Regarding the Localisation monitoring systems for IoT, Ghorpade et al. [129] define three levels of classification: localisation approach (centralised, distributed, and iterative), measurement technique, and localisation technique. These approaches are further divided into range-based techniques, which rely on distance or angle measurements (e.g., AoA, ToA, TDoA, RSSI), and range-free techniques, which use connectivity information (e.g., centroid-based, hop count-based). Device-based methods utilise specific hardware, while device-free methods depend on signal transmission interactions without particular devices. This work presents some similarities to our work when defining the architecture type and some measurements; however, the author's approach is limited as they do not consider monitoring attacks and performance metrics for evaluating the localisation solutions. Chandra et al. [130] propose a taxonomy of localisation methods in IoT infrastructure; they categorise techniques into two main groups: self-determining and training-dependent approaches. Self-determining methods are divided into geometric methods, mobility models, path planning, and statistical approximations. These methods involve techniques such as trilateration, multilateration, and various forms of geometric calculations that do not require prior training. On the other hand, training-dependent methods include fingerprinting, stochastic models, and machine learning approaches, which utilise a training phase to improve localisation accuracy. Although the analysis methods presented in this study are similar to ours, the authors do not address other essential criteria regarding the architecture, attacks or performance metrics.

To the best of our knowledge, no specific taxonomies for interference and performance IoT monitoring systems have been proposed. The frameworks proposed in this study result from the collection of parameterisations used in different independent studies on interference [131], [132], [133], [134], [119], and performance [135], [136], respectively.

## 3.9 Discussion

### 3.9.1 Lessons Learned

The comprehensive study of five distinct IoT monitoring systems—focusing on intrusion detection, localisation, interference, performance, and authentication—provides key insights based on a critical evaluation of system architecture, analysis methods, performance metrics, and attack monitoring.

Several key outcomes emerge from this investigation, which determined how we designed our targeted interference detection system and advanced to the subsequent chapters on jamming attacks and targeted interference detection. The survey findings primarily support the decision to deploy interference detection directly on IoT devices. This approach is based on the observation that interference usually occurs at the device level, making it practical to monitor and analyse the interference data exactly where it appears. This decision is supported by the challenges associated with data transmission to a central element for analysis, which includes increased bandwidth usage and potential delays, thereby impacting the timeliness and reliability of interference detection, and creating a single point of failure for attacks.

Moreover, the study highlights a preference for simpler, more direct monitoring metrics on devices, such as signal loss, which can be used as effective proxies for detecting interference. This insight leads to a shift in the analysis methods, emphasising local statistical techniques over more complex AI-driven solutions that are better suited for base station analysis. These methods are more resource-efficient and reduce

latency in threat detection and mitigation.

Synthesising these findings, we learn that an effective IoT security system must be both adaptable to the specific challenges presented by the environment in which it operates and resilient against the diverse tactics employed by adversaries. The lessons learned from evaluating the four categories across different monitoring systems provide a solid framework for developing a targeted interference detector sensitive to IoT environment requirements and lays the groundwork for future discussions making sure that the strategies developed are practical and effective in real-world situations.

Finally, it is noteworthy that we conducted a study of existing IoT monitoring solutions to achieve two main objectives: understanding current monitoring practices and identifying security gaps in IoT systems. Our findings revealed several key points: monitoring systems are often evaluated based on common categories, with certain systems (e.g. authentication) and technologies, like LoRa, receiving more research attention than others. Specifically, unlicensed IoT technologies have been the focus of more research on interference security issues, whereas such attacks in IoT-cellular network-based devices have been less studied. Our survey exposes the critical nature of jamming interference in IoT networks, noting that while numerous studies have addressed general interference in wireless networks, fewer have focused on jamming interference attacks and even fewer on jamming detection. From our investigation, we can also point out that although jamming interference attacks are straightforward to execute (e.g. constant jamming), their effects can be potentially devastating, and detecting and mitigating such attacks remains complex and requires advanced techniques for ensuring network resilience. Based on these factors and in order to address these security gaps, we formulated a solution that involves the development of a smart jamming interference attack for IoT networks and developing, initially, a detection mechanism for targeted malicious interference in NB-IoT. Subsequently, we extend this solution to a technology-independent targeted interference detector for wireless IoT. Considering our survey elements, our solution is implemented on each user equipment (UE) side, using packet losses and pattern losses as evaluation met-

rics, and employs statistical anomaly detection algorithms to assess the effectiveness against constant, selective, and combined jammer attacks.

# 4 | Jamming of NB-IoT Synchronisation Signals

NB-IoT is a relatively novel LPWAN radio technology used to deploy IoT infrastructures at scale. It is important that such deployments are resilient to attacks. In this work we describe how interference on the NB-IoT synchronisation signals - the initial communication steps - can be used to implement an effective DoS attack. Interference with the synchronisation prevents communication and may also allow an attacker to force a device to connect with a specific base station.

## 4.1 Introduction

Narrowband-Internet of Things (NB-IoT) is a Low Power Wide Area Network (LPWAN) radio technology defined by the 3GPP standard, Release-13 [137]. NB-IoT aims to support a large number of low-cost, low energy consumption, and low data rate devices operated in a large enhanced coverage area. NB-IoT is increasingly seen as the preferred future IoT technology as it is deployed as part of the existing LTE infrastructure and uses a licensed band that enables reliable and future-proof deployments. NB-IoT provides several security mechanisms based on established mechanisms defined for LTE [138]. However, privacy and security in NB-IoT have yet received little research attention.

In this work we consider an adversary using a jamming device to disrupt NB-IoT communication. We consider an intelligent jammer that targets the initial communi-

cation steps of NB-IoT communication to have a maximum impact. Specifically we describe and investigate how a jammer can interfere with the NPSS and NSSS which are used to initiate the NB-IoT contention-based random-access procedure. NPSS and NSSS perform time and frequency synchronisation, cell identity detection and acquisition of frame infrastructure information. NPSS is used to obtain symbol timing and Carrier Frequency Offset (CFO), while NSSS is used to obtain the NB-PCID. Interference with NPSS and NSSS can prevent an NB-IoT device to initiate communication with a base station (eNodeB). Furthermore, our experiments indicate that careful design of the interference signal could allow an attacker to force the UE to recognise a specific NB-PCID.

Existing work has focused on LTE specific attacks. For example, Eygi et al. [139] describe a countermeasure against smart jamming attacks on LTE synchronisation signals. The authors point out the vulnerability of LTE to jamming attacks due to the broadcast nature of the channel. Their work is focused specifically on the LTE downlink synchronisation signals. As countermeasure a jamming detection is proposed based on the Neyman-Pearson theorem. Simulation results show lower jamming success and better cell id detection. This work differs from our investigation as it considers jamming of synchronisation in LTE, proposing a countermeasure technique to mitigate jamming attacks on PSS and SSS by signal relocation within OFDM symbols, using Error Vector Magnitude (EVM) and Cell ID correction percentage. Although the authors describe a method for reducing the impact of jamming on the correct detection of cell IDs in LTE, a Denial-of-Service attack and the detection of jamming in a more general way on the downlink channel are not considered. Labib et al. [140] describe a mechanism to enhance the immunity of LTE Systems against spoofing. The work analyses spoofing of the LTE synchronisation signal where standard-compliant primary and secondary synchronisation sequences are transmitted by a fake cell. Several mitigation techniques are proposed. The simulation results show that LTE control channel spoofing is an effective DoS attack during the cell selection process. Although this work uses synchronisation signals to cause a DoS, it employs a different

approach for the attack. It explores various levels of RF spoofing attacks on LTE systems. It highlights that attackers can set up fake LTE base stations to transmit primary and secondary synchronisation signals and control messages, thereby deceiving user equipment (UE) and causing it to camp on the fake cell.

To the best of our knowledge, there has not yet been an investigation of smart jamming attacks on the synchronisation signals in NB-IoT.

## 4.2 The UE and eNodeB Synchronisation Process

The full specification of the NB-IoT protocol was completed in June 2016 and is described in [141]. Here we describe only the synchronisation procedure.

When an NB-IoT device (the UE) becomes active, it synchronises with a base station (eNodeB) to establish a communication link. First, the UE needs to identify a suitable cell to attach to, and for this purpose, parameters such as symbol, subframe, and frame timing and carrier frequency synchronisation must be obtained.

The NB-IoT random access procedure is illustrated in Figure 4.1. The process starts with the transmission of the NPSS and NSSS which are used by the UE to perform time and frequency synchronisation, cell identity detection and acquisition frame infrastructure information. The NPSS is used to obtain symbol timing and Carrier Frequency Offset (CFO), while the NSSS is used to obtain the NB-PCID.

After this step, the UE acquires the MIB carried by the NPBCH which is transmitted in subframe 0 in every transmitted frame. The MIB provides scheduling information for the uplink and downlink data channels.

Then, the random-access procedure initiates when the UE sends a random access preamble through NPRACH. Several messages are exchanged between UE and eNodeB through the NPDSCH and NPUSCH to obtain scheduling information and identity parameters that allow the connection establishment [41].

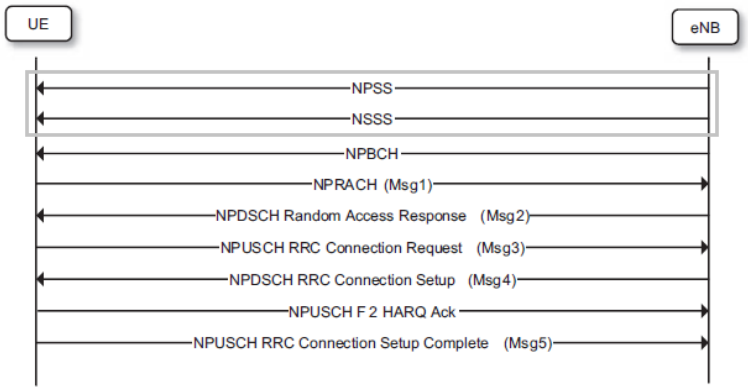



Figure 4.1: NB-IoT Random Access Procedure. Initial steps of a UE to establish communication with an eNodeB.

## 4.3 Jamming the NB-IoT Synchronisation Process

The NB-IoT standard defines that NPSS and NSSS are transmitted in specific subframes on an 80 ms repetition interval. Both signals are designed to allow a device to use a unified synchronisation algorithm during initial acquisition without knowing the specific NB-IoT operation mode [41]. Therefore, to avoid collisions with LTE subframes, subframe 5 is used for the NPSS and subframe 9 is used for the NSSS. Also, in order to avoid a potential collision with the LTE PDCCH, the subframes that carry NPSS or NSSS do not use the first three OFDM symbols. Thus, it leaves only 11 OFDM symbols per subframe available for NPSS and NSSS. The NPSS detection by the UE requires that the signal is detectable even with a very large frequency offset. Hence, all cells in an NB-IoT network use the same NPSS sequence. Consequently, a device only needs to search for one specific known NPSS sequence. Each of the 11 OFDM symbols in an NPSS subframe carries a copy of the base sequence. NB-IoT supports 504 unique NB-PCIDs indicated by the NSSS. Four different NSSS sequences are transmitted in an 80 ms repetition interval.

The NB-PCIDs is determined by a correlation algorithm using NPSS and NSSS which defines the peak correlation magnitude PEAK as the sum of the peak correlation magnitudes from time-domain NPSS detection and frequency-domain NSSS

detection. PEAK is a scalar indicating the peak magnitude of the correlation used to detect a cell.

Adding a jamming signal to the NPSS and NSSS signal may have the effect that the correlation algorithm is still able to produce a result but the determined NB-PCIDs is incorrect. Jamming can be applied only to the NPSS and NSSS signal without the need to continuously jam the entire transmission channel.

## 4.4 Jamming Evaluation

To test the aforementioned jamming attack on the synchronisation signal, we use Matlab [142] with the LTE-Toolbox. The simulation fully synchronises, demodulates and decodes an NB-IoT downlink signal.

We focus our jamming attacks on the downlink because of its potential security vulnerabilities, such as limited encryption and insecure over-the-air broadcast transmission. A wide range of critical messages are transmitted unencrypted, including synchronisation signals (NPSS and NSSS), the Narrowband Master Information Block (MIB-NB), which is sent in the Narrowband Physical Broadcast Channel (NPBCH), system information blocks (SIB1-NB, and other SIBs), and paging messages. These unencrypted transmissions, carrying essential network information, synchronisation data, control information, and initial connection parameters, provide attackers with valuable insights and clear attack vectors. The broadcast nature of these transmissions creates a large attack surface. Disrupting the downlink, particularly these unencrypted signals, can severely impact network functionality, preventing devices from connecting or maintaining communication. By exploiting these vulnerabilities in the downlink channel, an attacker could effectively disrupt total or partial NB-IoT network operations.

A time-domain waveform of a Reference Measurement Channel (RMC) is generated for an NPDSCH; the Matlab class used for this purpose is *NBIoTDownlinkWaveformGenerator*. Then, we introduce a jamming signal generated using a downlink 4G

RMC waveform where we can adjust the signal power. The resulting signal received by the UE is the sum of the generated signal obtained from the base station plus the jamming signal.

Following the example, the frequency offset estimation and correction are performed through the *lteFrequencyOffset* and *lteFrequencyCorrect* methods. Afterwards, OFDM demodulation and channel estimation are executed. Finally, to decode the MIB, the RE corresponding to the NPBCH from the first subframe across all receive antennas and channel estimates are extracted.

In our simulation, if the power of the interference signal is above 13dB a successful modification of the NB-PCIDs is observed. From the experiment results, it is observed that if the power on the jammer is insufficient, the communication with the specific eNodeB is inhibited because the UE is not able to decode the MIB and get the scheduling information.

Figure 4.2(a) shows the transmission signal without a jamming signal. Here, the cell ID is accurately detected [NNCellID: 120], and the MIB is decoded correctly. At this point, the MIB parameters are extracted (NNCellID, NB Reference Signal, Number of Sufbrame, HyperSuframe, Operation Mode, Additional Transmission SIB1).

In contrast, scenario Figure 4.2(b) shows the signal with jammer, where it is observed that it is not possible to decode the MIB, and a wrong cell ID is displayed [NNCellID: 371].

We can now evaluate the impact of different transmitter power levels on the effectiveness of jamming the Narrowband Internet of Things (NB-IoT) synchronisation signals. Our study identifies a power threshold for an effective jamming attack. If the interference signal power exceeds this threshold, it effectively modifies the Narrowband Primary Cell IDs (NB-PCIDs). This indicates that a sufficiently powerful jamming signal can alter the identification of the cell, potentially redirecting the User Equipment (UE) to a different base station (eNodeB). When the jammer's power is above the necessary threshold, the UE also fails to decode the Master Information Block (MIB). This

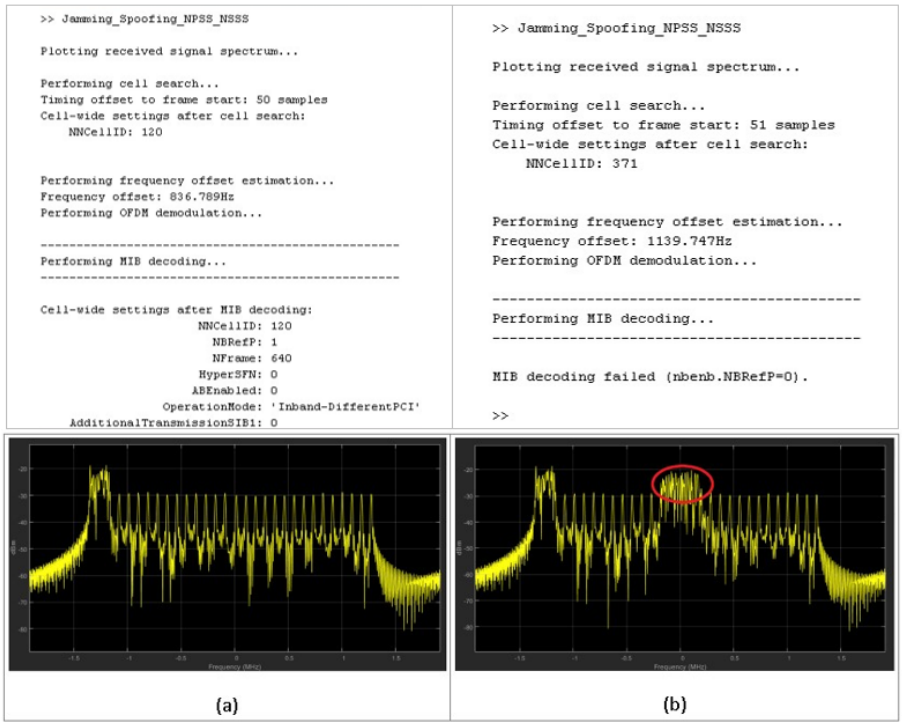



Figure 4.2: Jamming Attack on NB-IoT: (a) Standard Transmission (b) Transmission with Jamming Signal.

failure prevents the UE from obtaining crucial scheduling information from the eNodeB, thereby inhibiting communication. Understanding this threshold is vital for developing countermeasures. By ensuring the robustness of the synchronisation process against interference below this power threshold, communication can be safeguarded.

## 4.5 Conclusions

We have shown that it is possible to interfere with the synchronisation signal used by NB-IoT devices (the UE) to establish communication with the base station. Thus, a simple selective jamming device can prevent communication of NB-IoT devices. Furthermore, our experiments indicate that careful design of the interference signal might enable an attacker to force the UE to recognise a specific NB-PCID. In our next steps, we will analyse how the jamming signal should be designed for this purpose. We also plan to investigate methods to detect the jamming of synchronisation signals and methods to make the detection of the NB-PCID more robust.

# 5 | Detecting Targeted Interference in NB-IoT

Many IoT applications are considered critical systems, and it is important to guarantee that such deployments are resilient to attacks. An attacker may use radio interference selectively to disrupt communication while minimising the risk of their detection. It is essential to identify such attacks in order to remove the threat. In this work, we propose a novel method of detecting targeted interference in a Narrowband-Internet of Things (NB-IoT) network at the User Equipment (UE). NB-IoT is a recent LPWAN radio technology used to deploy IoT infrastructures at scale. Network performance data collected at the UE is used to reason about the current interference situation. Subframe loss rates within the downlink channel are monitored and used as input for a statistical anomaly detector. Our evaluation shows that the detector is able to distinguish targeted interference attacks from the impact of naturally occurring interference in cellular networks. This is important as naturally occurring interference requires a different response than targeted interference attacks.

## 5.1 Introduction

Narrowband-Internet of Things (NB-IoT) is an LPWAN radio technology defined in 3GPP standard Release-13 [137]. NB-IoT aims to support a large number of low-cost, low energy consuming, and low data rate devices operated in a large coverage area. NB-IoT is increasingly seen as one of the key future IoT technologies, as it is deployed

as part of the existing LTE infrastructure and uses a licensed band that enables reliable and future-proof deployments. NB-IoT provides several security mechanisms based on established mechanisms defined for LTE [138].

NB-IoT is used to construct IoT applications in domains such as smart cities, transport systems or smart grids. Many of these IoT applications are considered critical systems, and their secure and reliable operation is paramount. A challenge in IoT deployments is radio interference. Interference may originate naturally within cellular networks such as NB-IoT, from other networks deployed in the same frequency space, electrical machinery or a nefarious actor that aims to disturb communications. While IoT installations and protocols are designed with interference in mind, they usually do not consider deliberate interference from an adversary.

It has been shown that an attacker can use targeted interference to have a significant impact on an NB-IoT deployment [2, 143, 144]. For example, an attacker can use interference to perform a battery depletion attack [143, 144], to prevent synchronisation [2] or to carry out a DoS attack. Such targeted interference attacks are difficult to detect, as communication is not prevented entirely.

It is necessary to determine if a network is subject to such targeted interference attack, as the response to an interference attack is usually different compared to the response to natural interference. For example, in case of natural interference, a communication protocol may use redundant transmissions to deal with a situation. In case of a targeted interference attack, this may not be the appropriate reaction; instead, it might be better not to change communication behaviour, but instead alert the network operator to identify and remove the attacker.

In this work, we propose a novel method of detecting targeted interference in an NB-IoT network at the UE, e.g. a sensor node. We consider an adversary using a jamming device to interfere with NB-IoT communication. Specifically, we assume that the attacker observes and interferes with the downlink channel to achieve maximum impact. Network performance data collected at the UE is used to reason about the

current interference situation. Subframe loss rates within the downlink channel are monitored and used as input for a statistical anomaly detector at the UE. The specific contributions of our work are:

- *Targeted Interference Detection:* A UE based detection method using subframe loss rates observed within the downlink channel. Loss rates can be provided by the UE decoding pipeline without modification at the UE.
- *Evaluation of Targeted Interference Detection:* We evaluate the performance of the targeted interference detection mechanism using a Matlab simulation framework. We show that the detector is able to distinguish targeted and non-targeted interference reliably.
- *Optimal Targeted Interference Detection:* We describe the optimal configuration of the detector to balance true positive and false negative detection.

In Section 5.2, we describe the existing related work. In Section 5.3 we provide an overview of NB-IoT technology and describe our threat model. Section 5.4 presents the proposed NB-IoT targeted interference detection model. In Section 5.5, we present the experimental evaluation of the model and discuss results. Section 5.7 concludes the paper.

## 5.2 Related Work

In this section, we describe previous works that analyse jamming interference in cellular networks, emphasising those works that addressed jamming interference in cellular IoT networks. Multiple studies have been conducted to describe, model and mitigate interference in wireless communication networks. A more limited number of studies have focused on jamming interference attacks, and even fewer works have concentrated on jamming detection. Jamming detection in LTE and, specifically, NB-IoT has received little attention.

Jamming attacks, which exploit the use of interference generated by an adversary,

have been widely conducted in cellular networks such as LTE [145, 146, 147] or the recently deployed 5G [148, 149, 150]. IoT networks are also subject to a range of potential security attacks. Multiple works have focused on the performance analysis and evaluation of jamming in IoT networks [73, 151, 152, 153] as well as its detection and mitigation [154, 155, 156, 157]. Similarly, several works have studied jamming interference in LPWANs specifically, which include LoRa [158, 159], Sigfox [148] and NB-IoT. LPWANs are one of the dominant IoT technologies for managing critical infrastructure due to their low-power and long-range capabilities, leading to long-life deployments, e.g. many years up to a decade. A broad literature has focused on IoT security vulnerabilities and signals jamming attacks in LoRaWAN [160, 161], with one of the primary areas being the development of jamming attacks on the LoRa physical layer [162, 163], smart jammers [164, 165], and their mitigation techniques [119, 166].

To understand jamming vulnerabilities in LTE, several works exist that study the impact of the various types of jamming on physical channels, signals, frame structure, and decoding protocols as part of security in the physical layer of LTE [167, 168, 169, 170]. From the attack perspective, various authors studied multiple attacks like partial or total DoS and loss-of-service attacks generated by smart jammers [171, 172]. There is also work that proposes mitigation techniques [173, 174, 175] and countermeasures [139, 140] for jamming attacks in LTE. However, less work has focused on jamming detection for LTE or NB-IoT. In the following, we describe the most relevant related work.

**Topal et al. [176]** introduce a system to identify smart jammers in LTE. The model considers two steps, (i) signal pre-processing and (ii) classification. The authors use a wavelet-based pre-processing step to represent the signal as an image. In the classification step, a Deep Convolutional Network (DCNN) and Support Vector Machine (SVM) architecture is considered. The study considers three different jamming attacks: barrage jamming, synchronisation signal jamming and reference signal jamming. The classification performance of the proposed approach demonstrates that the DCNN technique has superior performance compared to SVM. The principal dif-

ference between this work and our solution is that it focuses on LTE, not NB-IoT and considers the jamming of synchronisation information and not data transmissions. Furthermore, the proposed detector uses Machine Learning (ML), is heavyweight, requires training and is therefore unsuitable for executing on a low-power NB-IoT UE. Additional equipment co-located with the Evolved Node B (eNodeB) would usually be required.

**Hachimi et al. [177]** present a jamming attack detection method for 5G Cloud Radio Access Networks (C-RAN) using a multi-stage machine learning-based intrusion detector (ML-IDS). The authors used a combination of deep learning and kernelised support vector machine (SVM) to detect four types of jamming attacks: constant jamming, random jamming, deceptive jamming, and reactive jamming. The detection method consists of two stages feature extraction and attack classification. This work differs from ours as it examines the detection of jamming attacks in 5G C-RAN networks using a multi-stage machine learning approach.

**Martinez et al. [159]** provide a comprehensive analysis of the effects of jamming attacks on the performance of a LoRaWAN network. The authors conducted experiments to model channel-aware and channel-oblivious jammers in LoRaWAN networks. In the former, statistical methods were used to analyse the data collected from the experiments using four performance metrics (packet loss probability, collision probability, packet rejection probability and network throughput). For the latter, two metrics are considered, the gateway (GW) occupancy, defined as the percentage of time the GW is busy processing packets, and the packet forwarding rate, which is the number of packets per minute the GW can process. Although this work focuses on the study of the impact of jamming attacks using statistical methods to evaluate performance metrics, it differs from our research as it focuses on another technology, LoRaWAN networks.

**Ionescu et al. [143, 144]** describe energy depletion attacks on NB-IoT devices using targeted interference. The work considers a jammer that aims to disrupt the process-

ing of specific information (MIB-NB, SIB1-NB or SIB2-NB information) delivered in the downlink channel from the base station (called eNodeB) to the UE. The UE's energy consumption increases as it aims to receive redundant messages. The results show that this type of jamming attack can impact the lifespan of a NB-IoT UE, e.g. lifetime reduces from 17 years to approximately four months. This work is complementary to our work as it describes similar attacks to those we consider, but it does not address detection.

## 5.3 Background

NB-IoT is designed to coexist with legacy GSM and LTE technologies [178]. NB-IoT operates in the licensed spectrum and supports three modes of operation: stand-alone, in-band and guard-band. A stand-alone deployment uses any available spectrum with a bandwidth larger than 180kHz. In this case, parts of the GSM spectrum might be used [41]. An NB-IoT in-band deployment uses the existing LTE network, occupying one of the LTE PRBs. NB-IoT in guard-band mode, is deployed using the unused bandwidth in the guard-band of LTE networks.

### 5.3.1 NB-IoT PHY Layer

NB-IoT supports FDD and TDD operations. The NB-IoT frame structure with 15kHz subcarrier spacing has 1024 hyper-frames, each hyper-frame has 1024 frames, and each frame consists of 10 subframes and each subframe is split into two slots of 0.5ms duration. For subcarrier spacing of 15kHz, supported in the uplink and downlink, each frame has 20 slots, while with the additional subcarrier spacing of 3,75 kHz, which is used only in the uplink, each frame contains five slots of 2ms each [41]. PRBs are defined to map the physical channels and signals in NB-IoT. A PRB consists of 12 subcarriers over 7 OFDM symbols, generating 84 RE, the smallest physical channel unit in the downlink [179], as shown in Figure 5.1. Table 5.1 summarizes the NB-IoT channels and signals in the uplink and downlink that are multiplexed across sub-

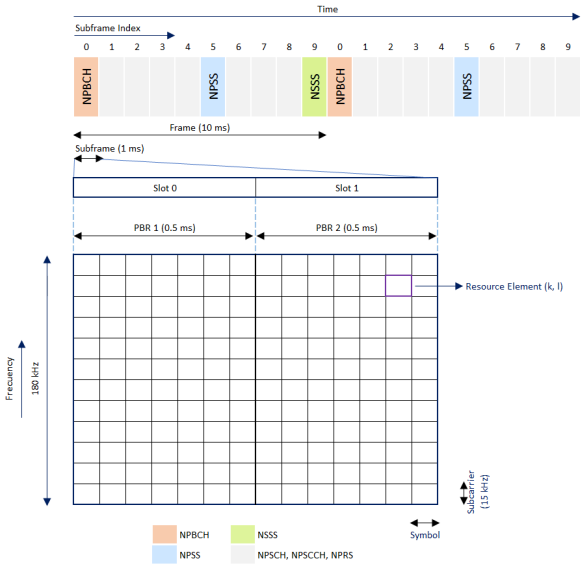



Figure 5.1: Time-multiplexing of downlink physical channels on NB-IoT

frames and REs.

### 5.3.2 Threat Model and Targeted Interference

We assume that an attacker is equipped with an NB-IoT transmitter. The attacker uses the device to emit a jamming/interference signal. The attacker will apply the interference signal for the shortest duration possible. This approach is used to (i) reduce energy usage (the attacker may operate on a battery powered device), and (ii) reduce the likelihood of detection (the attacker does not want to be found). Thus, the attacker will aim interference at specific protocol elements to achieve maximum impact with a minimum interference effort (a targeted attack).

Similarly, signal strength might impact the effectiveness of an attacker. The attacker must maintain a signal strength sufficient to jam the communications. Typically, the attacker operates within a distance range similar to User Equipment (UE) communicating with the base station (eNodeB). However, factors such as physical obstructions, materials, and line-of-sight issues can affect its impact. Thus, the attacker must carefully calibrate their signal strength to cause interference without being easily detected or unnecessarily depleting their energy resources.

Table 5.1: Physical Channels and Signals in NB-IoT

| Link | Physical Channel/Signal | Description |
|---|---|---|
| Down | Narrowband Primary Synchronisation Signal (NPSS), Narrowband Secondary Synchronisation Signal (NSSS) | Time and frequency synchronization. |
| Down | Narrowband Reference Signal (NRS) | Reference Time and Frequency. |
| Down | Narrowband Broadcast Channel (NPBCH) | Master information for system access. |
| Down | Narrowband Physical Downlink Control Channel (NPDCCH) | Uplink and downlink scheduling information. |
| Down | Narrowband Physical Downlink Shared Channel (NPDSCH) | Downlink dedicated and common data. |
| Up | Narrowband Physical Uplink Shared Channel (NPUSCH) | Uplink dedicated data. |
| Up | Narrowband Physical Random Access Channel (NPRACH) | Random access signalling in the uplink. |

We assume further that the attacker aims to disrupt the downlink channel with the attack. The attacker can monitor communication between UE and eNodeB. The attacker performs a targeted attack by submitting a jamming signal to disrupt (parts) of a specific physical channel, e.g. NPBCH. For example, the attacker may only target NPBCH to perform a battery depletion attack as described in [144] or NPDSCH to disrupt data transmission between a UE and eNodeB. As a result, the attacker will prevent successful reception of specific subframes (see Figure 5.1) at the UE. For example, by attacking the NPBCH, most subframes of index 0 are not received while most other subframes are received correctly.

The attacker may not be able to determine the content of all protocol elements, as some are transmitted encrypted. However, not all downlink elements use encryption (e.g. MIB) and interference can also be applied to encrypted signal parts.

## 5.4 Targeted Interference Detection

Our aim is to detect targeted interference as described in Section 5.3.2. Detection should be performed by the UE without hardware modification. Thus, only information already available via the NB-IoT transceiver hardware can be used; specific spectrum measurements are not possible. In addition, detection should be integrated with normal UE operations to avoid excessive additional energy costs. Detection should be a lightweight implementation to suit an embedded NB-IoT UE.

An NB-IoT UE wakes up periodically from deep sleep, synchronises with the eNodeB, transmits data and then returns to sleep. When waking up, the UE will receive a sequence of subframes via the downlink channel as shown in Figure 5.1. The UE will decode the subframes and use the information provided to synchronise and attach to the network. A targeted interferer will want to prevent decoding of specific subframes to disrupt communication. Of course, there may also be some subframe decoding failures due to natural interference.

To detect a targeted interference attack, we monitor the subframe loss rate during the wake-up period of the UE. The decoding chain of the UE transceiver can provide this information. The assumption is that a targeted interference attack will lead to a marked increase in subframe losses in the targeted areas compared to loss rates observed in non-targeted areas. We expect natural interference to cause equal loss rates for all subframes, while a targeted attack will cause an unequal distribution of lost subframes. Thus, it is possible to distinguish targeted interference from natural interference.

Natural interference typically spans a broad range of frequencies in the spectrum and is characterised by its random and uniform distribution across communication channels. This type of interference might include background noise from other electronic devices or general environmental electromagnetic noise, which does not target specific frequencies but affects the entire band used by the NB-IoT network. As a re-

sult, it causes a roughly equal loss rate across all subframes. In contrast, a targeted attack is more strategic and localised within the frequency spectrum. An attacker can use a jamming device to emit signals on specific frequencies or channels used by the NB-IoT communication, aiming to disrupt particular subframes. This focused interference can result in certain subframes experiencing higher loss rates, creating a noticeable disparity. Analysing the frequency spectrum makes it possible to identify an unequal distribution of lost, with concentrated spikes or anomalies at frequencies used by the targeted subframes, which are indicative of a targeted interference attack.

### 5.4.1 Detector Design

The detector monitors decoding success of a sequence of $S$ subframes in the downlink channel during the active period of the UE (a sequence of subframes as depicted in Figure 5.1). For each of the subframes $\eta$ ($\forall \eta \in 0, 1, 2, 3, 4, 6, 7, 8, 9$) within a frame, the loss rate $L_\eta$ is calculated. We do not consider subframes $\eta = 5$ as these contain only synchronisation information. Thus, a set of loss rates $L$ containing 8 values is collected. As we assume that loss is caused by either a targeted attack or natural interference, $L$ provides a picture of the interference environment that the UE is currently exposed to.

Next, values in the set $L$ have to be adjusted, resulting in the adjusted loss rate set $L'$. Adjustment is necessary as not all subframes react to interference similarly. Specifically, subframe 0 carries the NPBCH, which uses a different encoding mechanism than the other subframes. Subframe 0 experiences more loss than the other subframes for the same level of interference. To make loss rates comparable across all subframes, we have to adjust the loss rate before performing detection. We discuss this adjustment in detail in the next subsection.

The detector is used to check if a specific subframe $\eta$ is under targeted attack. For this purpose, the adjusted loss rate $L'_\eta$ is compared with the average adjusted loss rate $\widetilde{L'}$ computed over all other subframes. If the adjusted loss rate in the investigated subframe $L'_\eta$ is above the average adjusted loss rate $\widetilde{L'}$, a targeted attack is present.

We use a $\varphi$ to ensure that an attack is only detected if $L'_\eta$ is significantly above the observed average in order to reduce false alarms. We define $\varphi$ as a safety margin.

### 5.4.2 Loss Rate Adjustment

Subframe 0 experiences more loss than the other subframes for the same level of interference. The loss rate must therefore be adjusted to make it comparable with loss rates in other subframes. We use a compensation factor $\alpha$ as follows:

$$L'_0 = L_0 \cdot \alpha(\widetilde{L}) \tag{1}$$

$\widetilde{L}$ is the average loss rate in all other subframes except subframe 0. $\alpha(\widetilde{L})$ is a function, and the necessary adjustment depends on the observed interference level. As we show in the experimental evaluation, $\alpha(\widetilde{L})$ can be described as an exponential function:

$$\alpha(\widetilde{L}) = \delta \cdot e^{\lambda.\widetilde{L}} \tag{2}$$

It has to be taken into account that subframes might be under targeted attack when $\alpha(\widetilde{L})$ is computed. In this case, the adjustment may be incorrect. To account for this issue, it is possible to approximate $\alpha(\widetilde{L})$ by a static $\alpha$.

## 5.5 Evaluation Setup

We use the Matlab simulation environment to evaluate our proposed targeted interference detection method.

### 5.5.1 Simulation Environment

Our simulation environment was developed with Matlab using its LTE Toolbox, implementing the full transmitter and receiver signal processing chains. The receiver

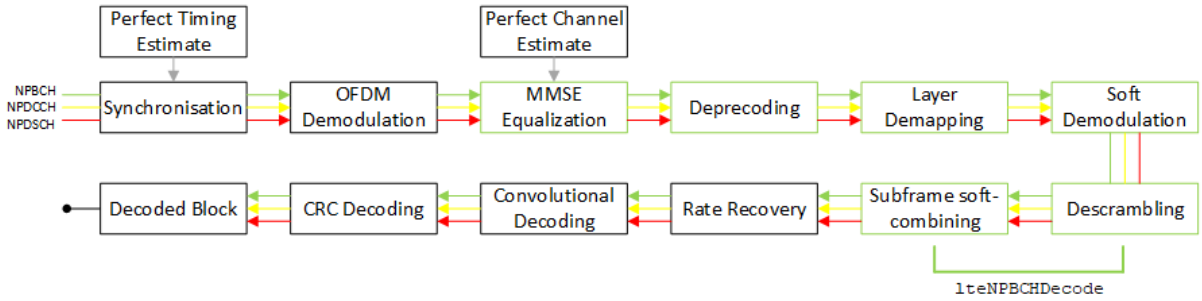



Figure 5.2: Communication Chain at the receiver side of the NB-IoT

chain fully synchronizes, demodulates and decodes an NB-IoT downlink signal sent by a transmitter. The downlink signal can be subjected to interference, and we can detect if subframes are received correctly. In order to do so, we use the decode block defined per each channel at the UE side (the receiver). For this, we use the *lteNPBCHDecode* class and *lteNDLSCHDecode* class in the Matlab LTE Toolbox. The outcome for each channel is a structure with the decoding state indicating 1 if the decoding was successful, and 0 if the decoding failed.

The decoding process for NPBCH in NB-IoT is described in detail in 3GPP TS 36.211 V15.7.0 [51]. We have analysed the overall transmission chain of the NB-IoT eNB as defined in 3GPP specifications [52], [53], [54]. Briefly summarised, the decoding process for NPBCH in NB-IoT starts once the signal has been received by the NB-IoT device. The first step in the process is demodulation, where the received signal is converted from its radio frequency representation to a digital signal using quadrature phase-shift keying (QPSK) de-modulation. The next step is channel de-coding, where the demodulated signal is decoded using a rate-1/3 Viterbi decoder to correct any errors introduced during transmission. This is followed by de-interleaving, which further corrects any errors introduced by the channel. The signal is then de-scrambled to obtain the uncoded bits, and an error check is performed on the uncoded bits to detect any errors in the received system information.

Figure 5.2 depicts in detail the communication chain at the receiver side [180]. We show in the figure the blocks executed for NPBCH, NPDCCH and NPDSCH decoding in Matlab. The decoding procedure is performed for every subframe that arrives

within the NB-IoT communication chain at the receiver side. For the NPBCH, a CRC can be evaluated to detect if the subframe was damaged (each NPBCH subframe is self-decodable). For the NPDCCH and NPDSCH, transmitted information (called transport block) may span several subframes and a CRC is only available for the entire transport block. Thus, it is only possible to determine if a subframe used for NPDCCH and NPDSCH is damaged using a CRC if transport blocks fit in a subframe. We make in our simulation this assumption. However, it has to be noted that it is also possible to estimate which subframe was damaged in case a transport block spans multiple subframes if NRS is used for interference estimation.

### 5.5.2 Communication Scenario

We consider communication between an NB-IoT node A and an eNodeB B (base station). B is transmitting subframes in the downlink channel to A and this communication can be subject to interference. We consider three types of interference that can be present:

- *Background Noise (BN):* Additive White Gaussian Noise (AWGN) that is present at all time while the downlink channel is active.
- *Background Traffic (BT):* Noise (interference) that is only present during certain periods of time (on-off noise). The duration of these noise periods and the gaps in between follow a Poisson process. This simulates interference from another communication network deployed in the NB-IoT space.
- *Targeted Attack (TA):* Interference signal that is present only in subframe 0, simulating a targeted attack on the Narrowband Broadcast Channel (NPBCH).

Each interference type can be present at different SNR levels. Different interference types can be present at the same time. The detector must be able to identify situations in which a targeted attack is present and must avoid false alarms, falsely identifying BN/BT as an attack.

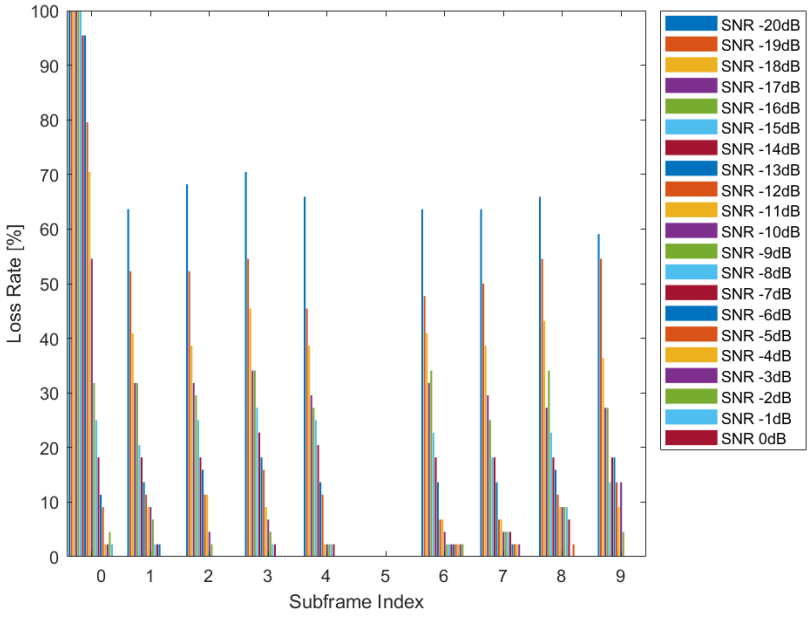



Figure 5.3: Loss rate $L$ for continuous background noise (BN).

## 5.6 Results

In this section, we describe first an experiment that was used to determine the loss rate adjustment. Thereafter we present an evaluation of the detector capability.

### 5.6.1 Loss Rate Adjustment

In this experiment, we subjected the downlink channel to continuous background noise (BN) at different SNR levels. We then determine the subframe loss rate $L$. Figure 5.3 depicts the observed loss rate for various SNR values.

It can be clearly seen that for the same noise level, the loss rate $L_0$ in subframe 0 is significantly larger than the loss rate in all other subframes. For example, for an $SNR = -19dB$ $L_0 = 100\%$ while $L_1 = 40.9\%$. Also, the exponential trend of loss rate depending on the SNR is clearly visible. As discussed previously, subframe 5 carries only timing information and is excluded.

Using this experimental data, we determine the parameters $\delta$ and $\lambda$ in Equation 2. $\delta = 0.25$ and $\lambda = 0.007$ provide a good fit, and we used these settings to adjust $L_0$.

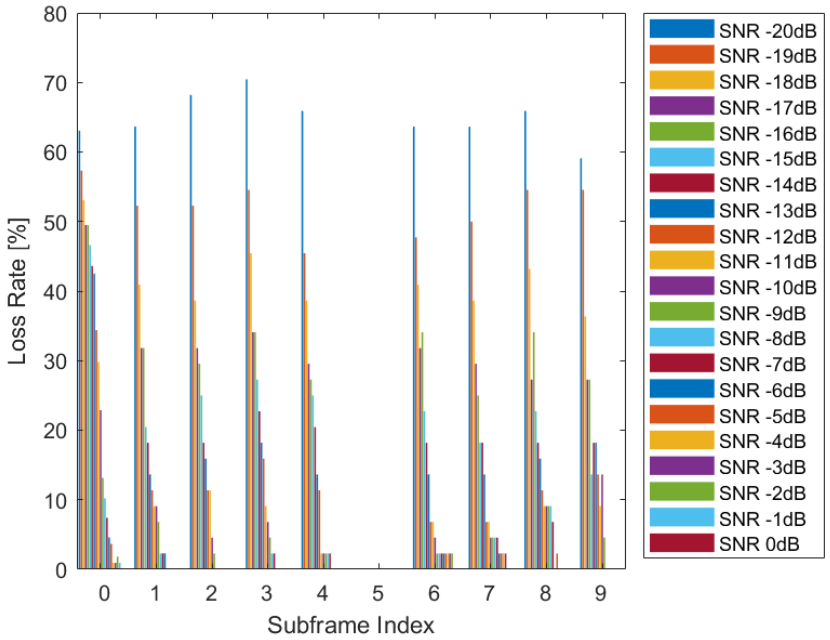



Figure 5.4: Adjusted loss rate $L'$ for continuous background noise (BN).

We assume that the values of $\delta$ and $\lambda$ can be generalised to other settings because they are derived from fundamental observations of subframe loss rates under different levels of interference. Our experiments demonstrate that these parameters accurately reflect the adjustment function behaviour across most of the tested scenarios. Importantly, these values are not tied to a specific experimental setup; rather, they can be broadly applied to other NB-IoT environments to achieve similar detection performance.

The adjusted figure for $L'$ is shown in Figure 5.4. Using the adjustment, loss rates in all subframes are now comparable; similar levels of interference cause the same loss rate.

### 5.6.2 Detection Capability

In this experiment, we create a number of scenarios with and without targeted interference attacks. We then test if the detector is able to determine that an attack is happening (True Positive) or if the detector falsely claims an attack is present (False Positive).

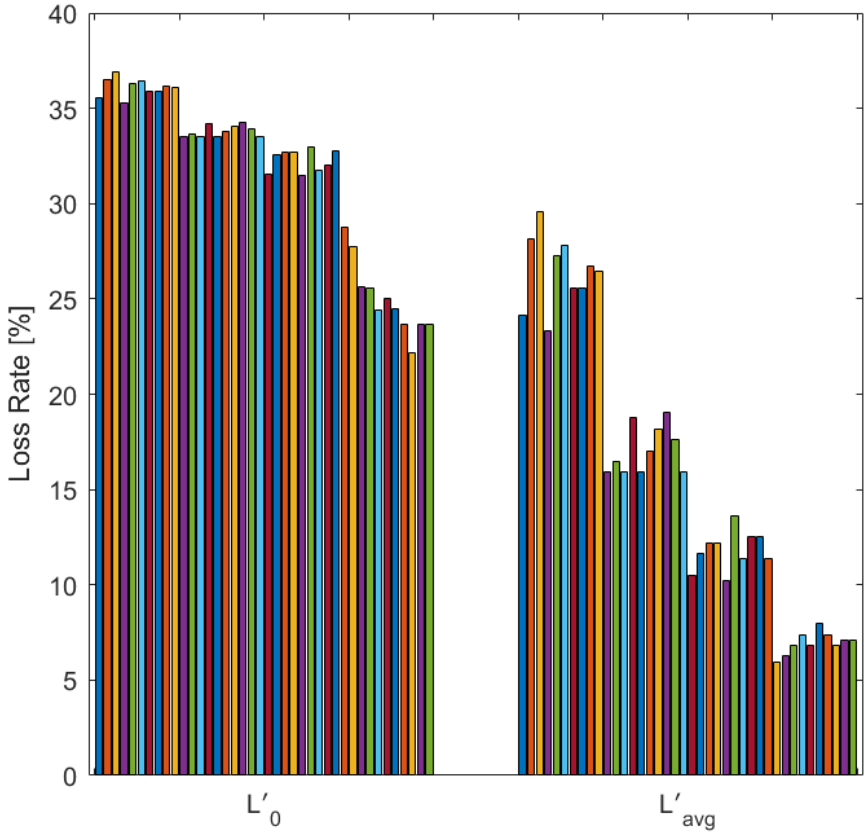



Figure 5.5: Loss rate $L'_0$ and $\widetilde{L'}$ for BA and TA interference (Attack is present).

We combine background noise (BN) and targeted attack (TA) noise to create realistic scenarios; in practice, even when a targeted attack is present, some background noise can be present too. For this purpose, we create interference signals combining BN and TA at different SNR levels. We create 40 attack scenarios with BN based SNR ranging from -10dB to -16dB (steps of 2dB) and TA based SNR ranging from -19.5dB to 0dB (steps of 0.5dB). The interference signal is in most cases stronger than the background noise; a scenario in which the targeted attack is buried in background noise is not a realistic setup.

We then also create 40 scenarios in which no attack is present. We use background noise (BN) based SNR ranging from -10db to -19.5dB (steps of 0.5dB) and background traffic (BT) based SNR ranging from -10db to -19.5dB.

Figure 5.5 shows the relation between the loss rate in subframe 0 $L'_0$ and the loss

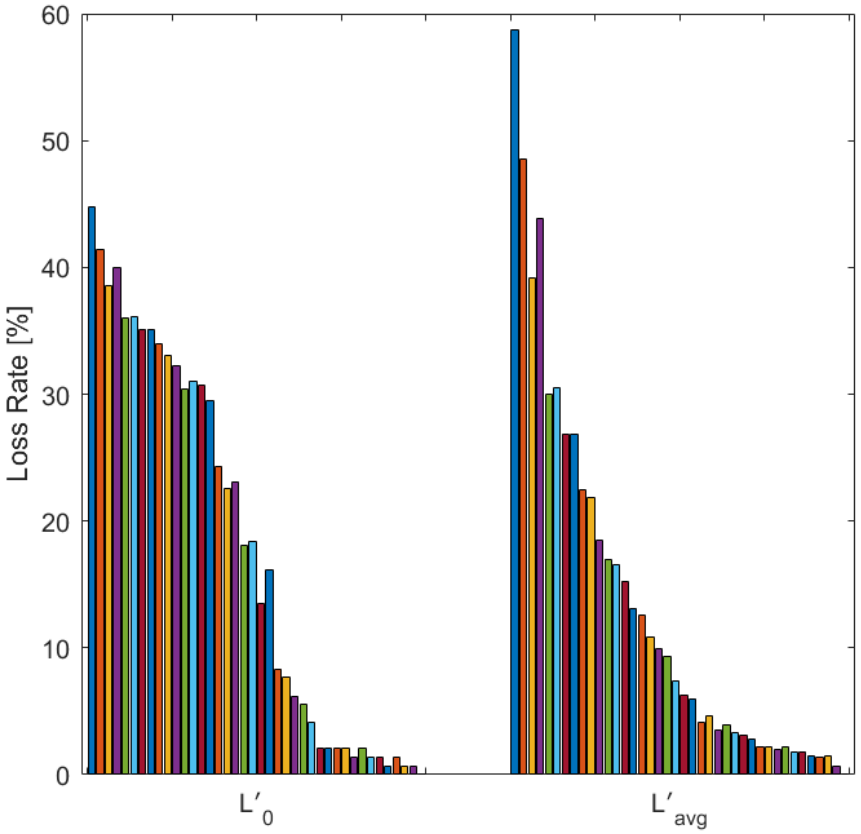



Figure 5.6: Loss rate $L'_0$ and $\widetilde{L'}$ for BN and BT interference (No attack is present).

rate average for the other subframes $\widetilde{L'}$ for background noise combined with the targeted attack in the NPBCH channel (Our 40 attack scenarios). It can be seen that with the same interference scenario, the loss rate $L_0$ in subframe 0 is generally larger than the average loss rate in all other subframes. For example, for an $SNR_{TA} = -17dB$ and $SNR_{BA} = -16dB$ $L_0 = 36.45\%$ while $\widetilde{L'} = 27.84\%$.

Figure 5.6 shows the comparison between the loss rate in subframe 0 $L'_0$ and the loss rate average for the other subframes $\widetilde{L'}$ for the background noise (BN) and the Background Traffic (BT) (Our 40 scenarios without attack). For the same interference scenario, the loss rate $L_0$ in subframe 0 is generally similar to the loss rate in all other subframes. For example, for an $SNR = -18dB$ $L_0 = 39,98\%$ while $\widetilde{L'} = 43,83\%$.

We use the detector as described in Section 5.4.1 and evaluate all possible safety margins $\varphi$ with the aforementioned 80 interference scenarios. Using a large $\varphi$ reduces

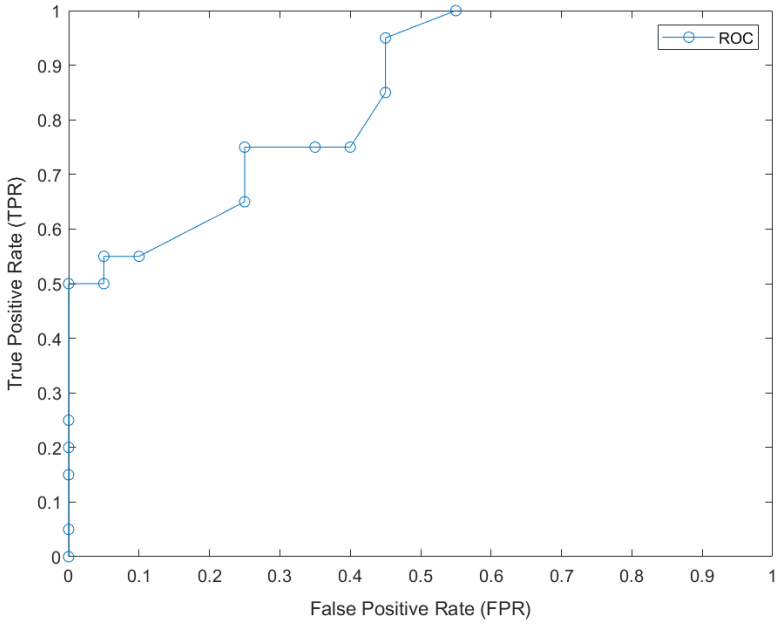



Figure 5.7: Receiver Operating Characteristic (ROC) curve

the amount of false positives (classifying an interference scenario without targeted attack as a targeted attack being present). At the same time, an increase in $\varphi$ reduces the true positives (classifying a present targeted interference as a targeted interference). Figure 5.7 shows the result of this experiment in form of a ROC curve.

As it can be seen in Figure 5.7 that it is possible to tune the detector such that no false positives occur while still 50% of targeted interference cases are detected ($\varphi = 10$ in this case). This setting would be used if it is necessary to eliminate false alarms, while it is acceptable to reduce detection reliability. It is also possible to tune the detector such that all targeted interference cases are detected; however, in this case, a high false positive rate of 55% is observed. False positive and false negative rates are equal for an EER=25% (for $\varphi = 5$).

In summary, it is possible to distinguish targeted attacks from other interference scenarios using the proposed methods. It would be acceptable in practice to tune the system to not create false positives and therefore miss some attacks. Over time (multiple wake-up periods) an attacker will eventually be detected.

## 5.7 Conclusion

We have demonstrated that it is possible to construct a statistical anomaly detector capable of detecting targeted interference on NB-IoT devices by utilising the available data in the UE as an indirect measure of interference. The detector has an EER=25% and it is possible to detect 50% of attacks without any false positives. For future work, we aim to broaden the capabilities of the detector. At present, the detector is able to detect targeted attacks on NPBCH and NPDSCH but not on NPSS, NSSS and NRS. To include detection for these attacks, additional mechanisms are required.

# 6 | Technology Independent Targeted Interference Detection for Wireless IoT

The expansion of IoT applications demands developing effective measures against targeted radio interference attacks. To remove such threats, it is essential to identify attacks and to distinguish them from natural occurring interference. Furthermore, it is desirable to develop a communication technology-independent detection method as IoT deployments such as smart cities or factories are usually heterogeneous. This research introduces a technology-independent method for detecting targeted interference in IoT deployments capable of operating on resource-constrained IoT devices. Packet loss rates and packet loss patterns independent of the specific physical layer technology are analysed to determine if a targeted attack is present. The proposed approach is validated through comprehensive evaluations of two example technologies, NB-IoT and IEEE 802.15.4 GTS, demonstrating its effectiveness in detecting attacks. Our evaluation shows that the detector can distinguish between targeted interference attacks and the impact of naturally occurring interference.

## 6.1 Introduction

It is important to guarantee IoT deployments are resilient to cyber-attacks. An attacker may use radio interference to disrupt communication while minimising the risk of his

detection. The attacker will selectively interfere with specific elements of the communication protocol in order to maximise disruption while minimising interference duration. Instead of a simple continuous and overpowering jamming signal an attacker will use a highly selective jamming signal with just sufficient power to cause disruption.

It is essential to identify such an attack in order to remove the threat. A particular challenge in this context is to distinguish naturally occurring interference from a deliberate attack. It is important to distinguish these two interference scenarios as they require different solutions. In case of natural interference, a response might be to step up transmission effort. In case of a targeted attack it might be better to stop transmission temporarily and put effort on identifying and removing the attacker. There is a need to devise methods for targeted interference detection.

IoT deployments may use a variety of wireless communication technologies. Depending on the specific use case a different wireless technologies may be beneficial. In some cases a number of wireless technologies might be used together in a heterogeneous setup. It is therefore desirable to devise a targeted interference detection that can be useful to a large set of wireless technologies.

Targeted interference detection may be executed at different points of a network and with the help of different tools and equipment. For example, it may be possible to execute interference detection centrally at a base station with specialist equipment. On the other hand, interference detection may be carried out on IoT devices only using information available from the existing communication transceiver. Depending on the approach, different cost and coverage will be associated with interference detection.

This chapter extends our previous effort to construct an interference detector for the NB-IoT protocol based on monitoring of loss rates [3]. We describe a novel method for targeted interference detection that can execute on resource-constrained IoT devices and can be used across different wireless technologies. The interference detection is independent from the lower-layer (physical and MAC) specifics of the wireless

technology. The detector monitors loss rates and loss sequence of different protocol elements. A targeted interferer will cause loss rates and/or patterns to specific protocol elements while natural interference will impact on all protocol elements the same way. To perform targeted interference detection for a wireless protocol, protocol elements need to be defined, which are then observed technology-independent by the detector. In this work, we show how IoT nodes using two example technologies, NB-IoT and 802.15.4 can perform targeted interference detection. The specific contributions of our work are:

- *Technology Independent Targeted Interference Detector:* We provide a design of a targeted interference detector monitoring *loss rates* and/or *loss patterns* of protocol elements in a technology-independent way.
- *Example Detection Scenarios:* We show how the detector can be used in two scenarios: NB-IoT and 802.15.4 GTS. For both scenarios we describe example attacks and how monitoring is performed.
- *Evaluation:* We provide a comprehensive simulation evaluation of the detector for the two use cases NB-IoT and 802.15.4 GTS.

## 6.2 Related Work

This section provides an overview of previous works that have examined the issue of jamming interference in Wireless IoT networks.

IDS in wireless IoT is subject of numerous research efforts, and some work aims at developing protocol-independent IDS [181, 182, 183]. These works differ from ours as they aim to detect general attacks and do not focus on targeted interference (jamming).

In an effort to characterise, model, and mitigate interference in wireless IoT communication networks, numerous studies have been undertaken. Jamming interference attacks have been the subject of a comparatively small number of studies, with even fewer works giving their attention to jamming attack detection. To the best of our

knowledge, no prior work has proposed a technology-independent jamming attack detection method. Thus, we focus in the next paragraphs on work in the IEEE 802.15.4 and LPWAN space as we use these technologies as examples for our work.

For IEEE 802.15.4 MAC layer jamming attacks, extensive research works have been proposed [56, 184, 185, 186, 187]. Likewise, a range of studies have addressed the issue of detecting jamming in IEEE 802.15.4 networks. [188, 189, 190],

Wood et al. [191] propose DEEJAM, a MAC-layer protocol for defeating energy-efficient jamming in networks based on IEEE 802.15.4. Combining four defensive mechanisms together, the protocol defeats the effectiveness of jamming from attackers with the same capabilities as other IEEE 802.15.4 network nodes. The authors demonstrate the performance of the protocol against four jamming attack classes: interrupt jamming, activity jamming, scan jamming, and pulse jamming. The proposed solution guarantees uninterrupted operation of network nodes even when a jammer is present. The evaluation was conducted using the MICAZ mote-embedded platform. Additionally, the model takes into account four complementary solutions to reduce the probability of a successful jamming attack: frame masking, channel hopping, packet fragmentation, and redundant encoding. The results indicate that despite a pulse jammer corrupting an entire channel, DEEJAM maintains a packet delivery ratio of 88%. This work focuses on IEEE 802.15.4 jamming attacks and it differs from our research as it focuses on the techniques to reduce/eliminate most of the impact of jamming instead of detecting ongoing attacks.

Mahony et al. [192] propose an IDS for ZigBee networks that is based on machine learning. The model formulates an interference detection focused entirely on analyzing received in-phase and quadrature-phase samples from the received WSN signal. The authors define a set of distinct features from the probability distribution function (PDF) and statistical analysis of the received samples. The results indicate that with a maximal feature depth of five and fifty-five decision trees, the lowest error level for the four-class case is approximately 5.87%. In contrast to our research, this study em-

ploys a machine learning approach to detect interference which would be difficult to execute on resource-constrained devices.

In a similar fashion, jamming interference in LPWANs has been the subject of numerous studies, including those on LoRa [158, 159], Sigfox [148] and NB-IoT [2, 3, 143, 144].

Additionally, the development of IDS in wireless IoT is subject of numerous studies [181, 182, 183].

Roux et al. [193] proposed RadIoT, a radio communication intrusion detection for IoT based on the monitoring and profiling radio communications at the physical layer using machine learning techniques. The implemented approach is designed to be independent of the wireless communications protocols, and it is proposed as a solution that analyses radio communications patterns to detect abnormalities that may indicate malicious activity. By employing a likelihood ratio test and a statistical model approach, the system is capable of identifying malicious gateways that compromise communications between access points and IoT devices. In order to enhance the precision of intrusion detection for IoT networks, the system integrates machine learning and data mining techniques. The PCA algorithm is employed to select and reduce features, while the FCM algorithm is utilised for clustering. The results show that RadIoT has an 88.46% precision and recall. This work differs from ours as they implement the solution at the physical layer, and it is limited to a smart home environment using machine learning techniques.

## 6.3 Malicious Interference

### 6.3.1 IoT Communications

A significant challenge in IoT deployments is interference. In a simple scenario, a second IoT network of the same type is deployed in the same physical space. In this case, two networks managed by different providers must allocate and share one frequency

space. The deployment of these two networks in a cooperative fashion to minimise interference is not necessarily the prevailing approach. An example here would be two independent LoRa networks deployed in the same physical space by two different providers. Therefore, monitoring the interference situation might be necessary as a network provider must be aware of the interference situation to tune network parameters to deal with it.

In many cases, the competing devices use a different communication protocol and, thus, are not even designed for coexistence. An example is an 802.15.4 network that may have to coexist with WiFi networks in the same deployment space. The communication protocols used for 802.15.4 are usually not designed to coexist with WiFi; similarly, WiFi is not designed to coexist with an 802.15.4 network. A network provider may want to monitor the interference situation to take appropriate action when this becomes an issue.

Interference from other technologies sharing the same unlicensed bands significantly impacts communication in the Internet of Things in various ways. The unlicensed frequency bands, such as the 2.4 GHz and 5 GHz bands, are commonly used by numerous wireless technologies, including Wi-Fi, Bluetooth, Zigbee, and others. With multiple devices operating in the same frequency band, the channels can become congested, and signals from different devices can interfere with each other, causing data packets to be corrupted or lost. As data packet collisions can occur more frequently, more retransmissions are necessary, which further reduces effective data rates. When the devices need to resend data packets that were lost or corrupted due to interference, it leads to higher energy consumption. Additionally, unlicensed bands are more susceptible to jamming attacks, and the shared nature of these bands makes it easier for unauthorised devices to eavesdrop on communications. In our work, we consider this type of interference under the *Background Traffic (BT)* (See Section 6.6.2), to replicate the impact of interference generated by other communication networks.

IoT networks are often used for applications that require strict performance guar-

antees. Interference by machinery in the deployment area can significantly impact performance. Hence, interference in these situations is often monitored so appropriate action can be taken to meet performance requirements. On the other hand, a malicious adversary might use interference to attack a system and prevent IoT operations. For such scenarios, monitoring interference may also be necessary to locate and remove a threat.

As the domain of IoT applications and services evolves, jamming interference attacks become increasingly diverse. Therefore, each UE must use a built-in mechanism for detecting, classifying, and characterising interference. We propose a solution that uses the PDU loss rates within the channels, which will be monitored and used as input for a statistical anomaly detector at the UE. The solution must be general enough to be applied to a variety of IoT devices using different standards such as LTE-M, NB-IoT, IEEE802.15.4 or LoRA. It should be flexible enough to manage various types of jamming interference scenarios and should be able to detect the attack with appropriate accuracy.

### 6.3.2 Threat Model

We assume an attacker is equipped with an IoT device with similar capabilities to the other devices in the network. The attacker uses the device to emit a jamming interference signal, and it is able to monitor the communication between the transmitter node and the target node. The attacker knows the protocol and aims to use the jamming signal such that communication between the nodes is still possible to evade detection. The attacker attempts to use interference for the shortest duration possible to reduce energy usage (the attacker may operate on a battery-powered device) and to reduce the likelihood of detection as the attacker does not want to be found. Thus, the attacker will aim interference at specific protocol elements to achieve maximum impact with a minimum interference effort. We called that a targeted attack. We assume further that the attacker aims to disrupt the communication channels with the attack. The attacker performs a targeted attack by emitting a jamming signal to disrupt (parts) of

a specific physical channel. As a result, the attacker will prevent the successful reception of specific PDU at the end device. The attacker may not be able to determine the content of all protocol elements, as some are transmitted encrypted. However, not all channel elements use encryption (e.g. MIB in NB-IoT), and interference can also be applied to encrypted signal parts.

## 6.4 IoT Case Studies and Jamming Attacks

### 6.4.1 Narrowband-Internet of Things (NB-IoT)

NB-IoT is a standard introduced by the 3GPP, is based on LTE and is designed for LPWAN. This allows reuse of the existing LTE infrastructure in the licensed spectrum, resulting in a more predictable and stable performance compared to other unlicensed solutions [194].

For the specific case of NB-IoT, we assume that the attacker aims to disrupt the downlink channel with jamming interference. The attacker might monitor communication between UE and eNodeB. The attacker can perform a targeted attack by submitting a jamming signal to disrupt parts of a specific physical channel, e.g. NPBCH [3], NPSS and NSSS [2], System Information Block (SIB) [143]. The attacker may only target NPBCH to perform a battery depletion attack or NPDSCH to disrupt data transmission between a UE and eNodeB. The attacker prevents the successful reception of specific subframes at the UE.

In this study, we consider two selective jamming interference attacks (referred to as *Constant MIB* and *Selective MIB* attack) on the downstream NB-IoT channel based on attacks described in [143]. Both attacks target the NPBCH transporting the MIB in subframe 0. The MIB is transported in eight CSB which are each repeated eight times. The *Constant MIB* attack assumes that the interferer targets subframe 0 (of 10 subframes present in the downlink channel). The *Selective MIB* attack assumes that only 16 consecutive CSB are subject to interference as this is the minimum level of

interference required to prevent successful decoding of the MIB. Figure 6.1 illustrates the Transmission of the NPBCH-CSB with a Transmission Time Interval (TTI) of 640ms and the Selective MIB targeted attack to only 16 consecutive subframes.

*Constant MIB* and *Selective MIB* attack are examples of a targeted interference attack that can be used to prevent efficient communication without preventing all communication (downgrade).

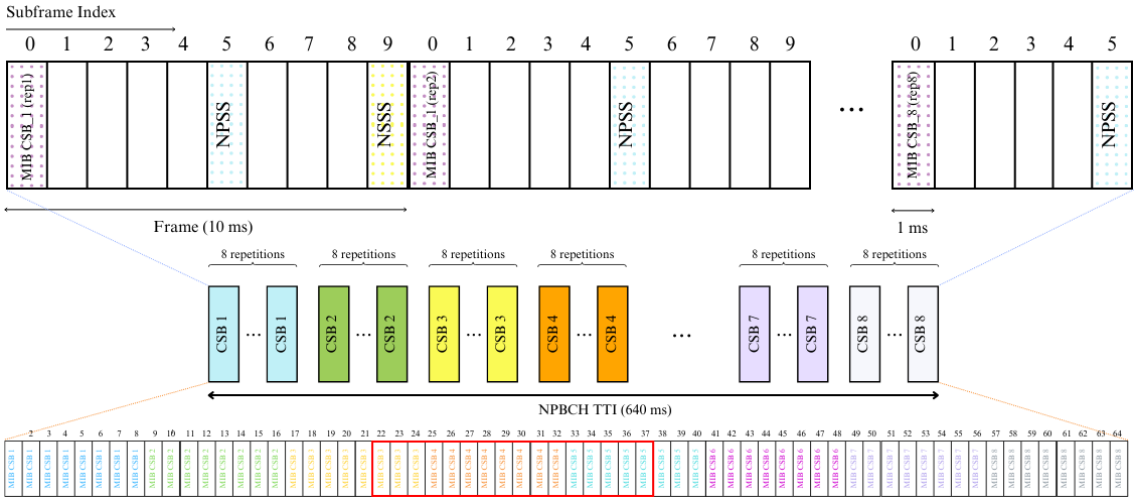



Figure 6.1: NB-IoT Selective MIB Attack.

### 6.4.2 IEEE 802.15.4

The IEEE 802.15.4 MAC [1] defines two fundamental MAC modes: (i) non-beacon enabled and (ii) beacon-enabled.

The beacon-enabled approach adopts a slotted CSMA/CA. Here, the network coordinator periodically broadcasts beacon frames to announce the presence of the network and provide timing and control information. These beacons divide the channel into superframes of the same direction. The superframe structure is depicted in Figure 6.2.

A superframe is divided into two periods: a mandatory active period and an optional inactive period. The active period includes CAP and CFP. CAP is a part of the superframe during which devices can transmit data using a contention-based access

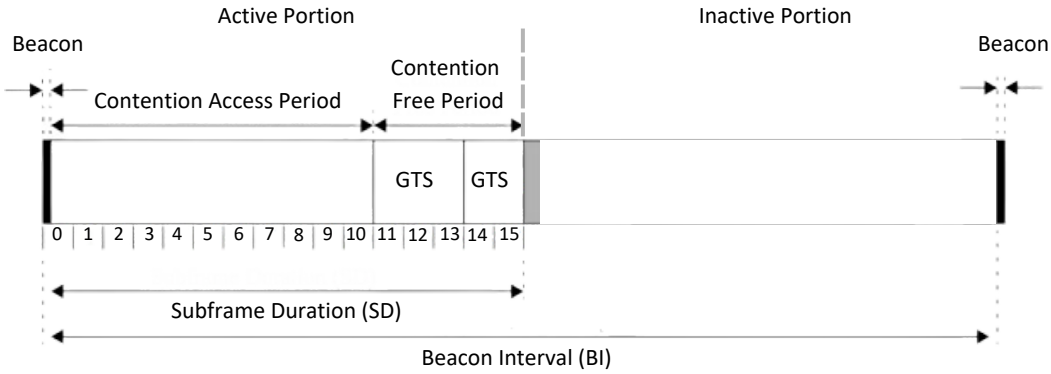



Figure 6.2: IEEE 802.15.4 Superframe Structure.

method. In this period, devices that need to communicate attempt to transmit their data using a Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) mechanism. CSMA/CA helps avoid collisions by checking the channel's availability before initiating data transmissions. The CFP period is a portion of the superframe reserved for contention-free communication. During this period, specific devices, typically associated with the network coordinator, can transmit data without the need for contention. The coordinator allocates time slots to individual devices, allowing them to transmit data in a coordinated and collision-free manner. The active period is equally divided into 16 slots.

In this study we consider an attack scenario using the IEEE 802.15.4 GTS in which an adversary device synchronises with the PAN coordinator through the reception of the beacon messages. The beacon frame contains the GTS descriptor, which indicates the GTS starting slot, length, direction, and associated device address. The adversary node can learn the GTS times from the coordinator by extracting the GTS descriptor from the beacon. Once the attacker obtains the allocated GTS times, it can create jamming interference in any of these dedicated slots, causing a denial of service as these slots are assumed to provide collision-free communication [195], [56]. We refer to this attack as *Constant GTS*.

## 6.5 Technology Independent Targeted Interference Detection

### 6.5.1 Concept

We consider an adversary performing targeted interference against an IoT device (see Section 6.3.2). We aim to detect such targeted interference on the victim IoT device. It is important for the victim to understand that he is subject to an interference attack as it requires a different response than a response to a noisy environment (i.e. stop transmissions and send alert vs. increase transmit repetitions). As the detector is included in a resource-constrained IoT device, the detector should be lightweight and only use information already available via the communication transceiver. Furthermore, the detector should be independent from the lower-layer (physical and MAC) specifics such that the design can be used across multiple different wireless technologies.

We monitor on the device the reception success of Protocol Data Units (PDUs) over time. A PDU is a protocol element received periodically by the IoT device (e.g. a beacon, a data frame, a routing message). Not all possible PDUs need to be tracked for targeted interference monitoring. The assumption is that a device is able to tell if a PDU has been lost or has been received successfully.

For targeted interference detection, the sequence of reception success of the monitored PDUs is analysed. This task can be performed independent of the specifics of a wireless protocol and device once PDU monitoring specifics are defined.

The assumption is that an interferer will target a specific PDU to achieve his goal. Such an attack can be distinguished from naturally occurring interference. Targeted interference will cause an increased loss rate for one specific PDU. In addition, loss patterns for a PDU will exhibit periodicity in case of a targeted attack while natural interference will cause a random loss patterns.

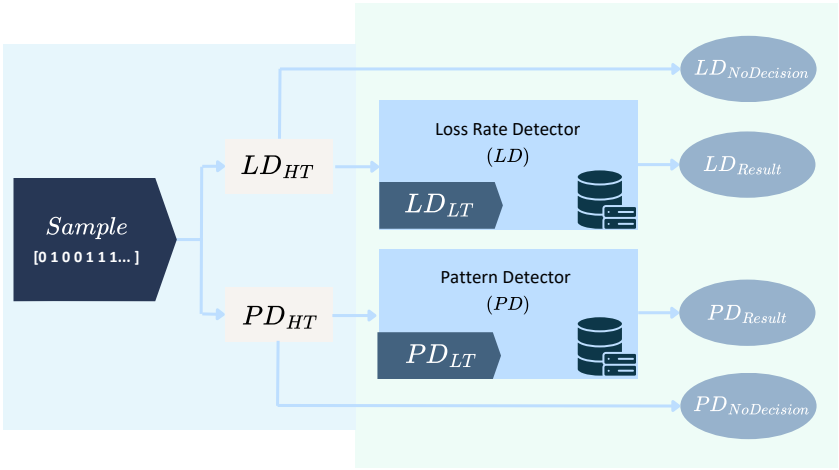



Figure 6.3: Detector Design: The sequence $S_n$ is received every $T$ seconds describing PDU loss/reception. The detector decides if a targeted attack is present.

### 6.5.2 Design Principles

In the development of a generic targeted interference detector, particularly when integrating new devices from different IoT setups (e.g. LoRa, NB-IoT, Sigfox, ZigBee, etc.), understanding and defining Protocol Data Unit (PDU) is pivotal for effective data transmission and interference detection. A PDU is a block of data transferred over a network, carrying information used for critical functions directly linked to the successful establishment of communication within the network. These PDUs encapsulate both control information and user data necessary for transmission.

To annotate PDUs effectively, one must first identify them within the data flow. For instance, in a scenario involving an IoT transmitter communicating with an IoT node, begin by examining the transmission patterns—note any recurring packets that appear periodically. These packets likely represent the PDUs, which are necessary for the operational requirements of the protocol and the detector. Analyzing these packets involves assessing their content to determine what specific information they carry and how they are structured. This includes identifying important bits that provide error control information.

Once PDUs are identified, the next step is to instruct the system on processing

these units. This involves configuring the detector handling software to recognize the structure of these PDUs—understanding where they start and end, and how to extract particular data. To extract these important bits, the system must be aware of any error-control techniques specific to these PDUs, such as reliability parameters (e.g. checksum/CRC to check if packets are lost), latency requirements, and handling of corrupted packets.

Thus, when using a new IoT protocol or adapting an existing one to our generic detector, the identification and annotation of PDUs play a foundational role. They determine how data integrity, reliability, and efficiency are maintained in a network, dictating the robustness of the detection system. This process involves precise steps and definitions that must be tailored to the specifics of the protocol and the operational environment:

- **Identifying PDUs -** A PDU is selected based on its relevance to the ongoing communication (e.g. PDU containing control or synchronisation bits) and its characteristics within the transmission environment (e.g. non-encrypted PDU). The first step requires a comprehensive analysis of the network's data flow to determine which PDUs are most relevant for detection purposes. This involves capturing a broad set of data packets transmitted over the network and analysing their frequency, size, and content.
- **Categorising PDUs -** Once potential PDUs are identified, they need to be categorised (e.g., PDU1, PDU2, etc.). This categorisation helps in structuring the detection process and assigning specific attributes to each type of PDU based on their roles and functions within the protocol.
- **PDU Requirements -** A PDU should exhibit some degree of periodicity, enhancing detectability and reliability in interference assessment. This periodic nature allows the detector to anticipate and verify the presence of PDUs in a predictable pattern, facilitating accurate monitoring. However, strict periodic might not be required.

- **PDU Selection Criteria -** When selecting a PDU, it is essential to ensure that the detector can identify and abstract transmission sequences to recognize patterns of success or loss, which are indicative of potential interference.

### 6.5.3 Detector Architecture

The targeted interference detector is split into a small adapter interface layer and a detector. The adapter is specific to the wireless protocol and device considered, while the detector functions are protocol and device-independent, and is therefore reusable in different IoT settings.

#### 6.5.3.1 Adapter

The adapter requires the ability to monitor the reception success of $\eta$ PDUs over a time period $T$. The adaptor may use a configuration syntax to specify what constitutes each PDU (e.g. mapping of message type or slot number to PDU). During $T$ the adapter collects $\eta$ sequences $S_n$ of 0s and 1s denoting $PDU_\eta$ reception failure or success. In a TDMA protocol where a PDU is mapped to a specific slot, a sequence $S_n$ has implicit timing information; it is known when a reception succeeded or failed. In a contention-based protocol, it is not possible to reconstruct from the sequence $S_n$ when each reception failure or success occurred within $T$. Every period $T$, the sequences $S_n$ are passed to the detector.

#### 6.5.3.2 Detector

The detector (shown in Figure 6.3) receives the sequences $S_n$ every $T$ seconds and must make a decision if the device is exposed to a targeted interference attack or not. The detector consists of a *Loss Rate Detector* and a *Loss Pattern Detector*. Each detector may be used on their own or the outputs are fused to obtain an overall decision.

A High Threshold ($HT$) (named $LD_{HT}$ and $PD_{HT}$ for the two detectors) is implemented to filter the sequences $S_n$ to exclude sequences that exhibit a very high packet loss. In such cases, it is impossible for any detector to make a decision; it cannot be

decided if a targeted attack exists, but it is obvious that the entire communication link is non-functional.

A Low Threshold (*LT*) (named $LD_{LT}$ and $PD_{LT}$ for the two detectors) is implemented within the detectors decision logic. If the observed losses are below this threshold an attack on the specific PDU will be inefficient and we count an ineffective attack as a non-attack case.

It has to be noted that multiple instances of *Loss Rate Detector* and a *Loss Pattern Detector* may be executed for each PDU to be monitored on an IoT device.

**Loss Rate Detector (LD)** The Loss Rate Detector evaluates the packet loss rate of one PDU in relation to the loss rates of all other PDUs. For each monitored $PDU_\eta$ ($\forall \eta \in 0, 1, 2, 3, 4, ..., \eta$), we calculate the loss rate $L_\eta$. As a result, a set of loss rates $L$ containing $\eta$ values is collected. $L$ indicates the interference environment to which the device is exposed, assuming that it is caused by either a targeted attack or natural interference. Under a targeted attack, the investigated PDU will exhibit a higher loss rate than other PDUs. As PDUs may differ in packet size or may use different coding schemes, it may be required to adjust loss rates accordingly, resulting in the adjusted loss rate set $L'$. A targeted attack is present if the adjusted loss rate in the investigated PDU $L'_\eta$ is above the average loss rate $\widetilde{L'}$. To mitigate false alarms, we define $\varphi$ as a safety margin to ensure that an attack is detected only if $L'_\eta$ exhibits a significant deviation from the average $\widetilde{L'}$. Algorithm 1 is defined to test the *Loss Rate Detector (LD)*.

**Algorithm 1** Loss Rate Detection Algorithm

**Input:** $L'$, $\eta$
**Output:** {0, 1}
1: $\widetilde{L'}$ = AVERAGE($L'$, $\eta$)
2: **if** $L'_\eta > \widetilde{L'} \cdot \varphi$ **then**
3: **return** 1 {$PDU_\eta$ is under attack}
4: **else**
5: **return** 0 {$PDU_\eta$ is not attacked}
6: **end if**

**Loss Pattern Detector (PD)** The Loss Pattern Detector analyses the loss pattern of a single PDU. For the specific monitored $PDU_\eta$ ($\forall \eta \in 0, 1, 2, 3, 4, ..., \eta$), the sequence $S_n$ is evaluated. Thus, a set of loss sequences $S_n$ is collected per a specific PDU. The loss sequence $S_n$ is processed using the autocorrelation with autocovariance coefficients ($corrcoef$) and then passing the result through an $FFT$ function. If clear peaks are visible in the result a pattern exists. Under a targeted attack, the pattern detector will be able to determine if periodicity of losses is present; for natural interference no clear pattern will be visible. A targeted attack is present if a significant peak in the autocovariance function is identified. To mitigate false alarms, we define $\gamma$ as a safety margin determining the significance of a present peak.

Algorithm 2 is defined to test *Loss Pattern Detector (PD)*

**Algorithm 2** Loss Pattern Detection Algorithm

**Input:** $L$, $\eta$
**Output:** {0, 1}
1: $\widetilde{L}$ = corrcoef($L$, $\eta$)
2: $\widetilde{L}'$ =findpeaks( FFT($\widetilde{L}$, $\eta$))
3: **if** $L_\eta > \widetilde{L}' \cdot \gamma$ **then**
4: **return** 1 {$PDU_\eta$ is under attack}
5: **else**
6: **return** 0 {$PDU_\eta$ is not attacked}
7: **end if**

The Fused Detector integrates the outputs of the $LD$ and the $PD$ detectors. This integration involves processing the individual outcomes through two logical operations considering the union and the intersection of the results. The final detection decision is flagged by the Fused Detector. This dual-decision approach facilitates a comprehensive analysis of the detection effectiveness, particularly when the nature of the network attack is unknown.

#### 6.5.3.3 Comprehensive Analysis for the Detector

**Energy Analysis.-** Our solution analyses binary sequences from the adapter module and processes them in the detector module in order to complete the targeted inter-

ference detection process. The proposed detector has a minimal energy impact on the device due to its reliance on operations (e.g. obtaining the sequences) already performed by the transceiver during normal transmission and reception processes. When implementing the detector, the device remains active during transmission, eliminating the need for additional wake-up energy to process 0s and 1s. Although there is a small energy requirement to process this data, it is relatively insignificant compared with the energy required to process and execute the entire communication chain with the different transmission blocks (See Fig. 5.2). The main computational task of the detector is to calculate an average/autocovariance over a sequence, which is far less complex than the operations involved in the transmission chain.

Moreover, in NB-IoT, for every subframe transmission or reception, each NB-IoT device must execute the entire transmitter or receiver chain, which involves a significant number of calculations. For example, recording a transmission of 640 subframes (such as the MIB) requires the transceiver chain to run 640 times, while our algorithm only runs once at the end of these transmissions. Therefore, the computational overhead introduced by our detector is negligible compared to the transceiver's processing requirements.

In practical terms, the additional instructions needed for the detector executed once are minimal when compared to the extensive calculations of the transceiver chain executed multiple times while receiving the complete bits for completing the transmission. We can assume this as the transceiver already performs the necessary computations to generate the 0s and 1s, and our detector only needs to calculate a simple average/autocorrelation algorithm.

This approach contrasts sharply with more complex methods, such as machine learning, which would require running extensive models that require similar computation that the transmission chain and significantly increase energy consumption. By opting for simple statistical methods, we ensure minimal additional energy impact. In an ideal scenario, if we measure the energy consumption during the transmission and

reception phases before and after adding our detector code, we expect no noticeable difference, underscoring the negligible energy impact of our proposed solution.

**Approach Analysis.-** Our proposed solution relies on statistical methods, which fundamentally involve collecting and examining data at the UE side to identify loss rates and/or loss patterns of protocol elements in a technology-independent way. This approach utilises established mathematical formulas (e.g. autocorrelation, Fast Fourier Transform, average) to analyse data, making it relatively straightforward and interpretable. It is possible to replace these two detectors with one that can feed data from our adapter module into a machine learning system; our detector architecture has the adaptability to allow this.

In the context of IoT communications, one appropriate approach could be the use of solutions based on tiny machine learning (TinyML) mechanisms, which involve implementing machine learning models on devices with limited computational resources. TinyML focuses on deploying lightweight algorithms that can perform classification, regression, and anomaly detection tasks with minimal power consumption and memory usage.

While both approaches have advantages and limitations, we present now some core differences. For example, regarding complexity and interpretability, our statistical anomaly detector is, in general, simpler and more interpretable, providing clear insights based on well-understood IoT network behaviour; while TinyML potentially could be more complex as training data models are required. Regarding the resource requirements, our statistical-based solution requires less computational power and is easier to implement on constraint IoT devices; despite TinyML being designed to optimise machine learning capabilities for devices with limited resources, the complexity of using an ML model can be substantial; this is particularly true for complex models, which require significantly more computational effort compared to our straightforward calculations. Regarding adaptability, TinyML solutions can continuously learn and adapt to new data, making them ideal for applications where conditions change

over time; our solution was tested to perform in a similar manner, as we showed that our detector is able to detect combined attacks which simulates a dynamic and real attack without any recalibration as new data becomes available at the adapter module.

Overall, it might be the case that certain attacks can be detected better by machine learning, but this comes with increased complexity. Our research demonstrates that the use of statistical methods can produce accurate detection results.

**Location Analysis.-** The UE-based detection approach focuses on identifying interference directly at the UE, as this is where the actual impact of a targeted interference attack is experienced. This method ensures that the UE detects when it is not receiving expected data. This approach allows for real-time detection and response at the UE level, providing a more immediate and localised assessment of the targeted interference attack.

Conversely, a detection solution based on statistical or machine learning analysis at the base station or cloud involves transporting measurement data from the UE to the base station or cloud. This approach requires transmitting sequences of data (e.g., 0s and 1s) to the base station/cloud for analysis from the UE or the channel. While this method allows for the utilisation of advanced machine learning techniques without burdening the UE, it introduces latency and the need for significant data transport. It also implies a need for reliable and continuous data transmission from the UE to the base station, which might be resource intensive. Additionally, the base station might not capture the same granularity of interference details experienced at the UE, making it less effective in detecting UE-specific attacks. Furthermore, if the base station itself is under attack, our proposed anomaly detector can be applied to detect such scenarios combined with machine learning approaches while incurring a relatively low cost for the base station.

In summary, while statistical/ML-based analysis at the base station or cloud can leverage powerful computation resources and advanced techniques, the UE-based detection approach offers more immediate, precise, and detailed insights into interfer-

ence, particularly when the attack is localised at the UE.

## 6.6 Evaluation Setup

### 6.6.1 Simulation Environment and Scenarios

All simulations were carried out using Matlab; we used the LTE Toolbox for NB-IoT and the Communications Toolbox Library for IEEE 802.15.4.

#### 6.6.1.1 Narrowband-Internet of Things (NB-IoT)

We simulate downlink communication between a base station (eNodeB) and a device (UE). The UE should determine if it is under attack using the detector. The complete transmitter and receiver signal processing chains of an NB-IoT system is simulated. The receiver chain completes the synchronisation, demodulation, and decoding of the NB-IoT downlink signal transmitted by the eNodeB. In our detector configuration, the adaption layer defines the NB-IoT downstream subframes as the PDUs of interest. The downlink signal may be subjected to interference, and it is possible to determine the accurate reception of subframes; 1 if decoding was successful, and 0 if unsuccessful, creating our sequences $S_n$.

As each NPBCH subframe is capable of self-decoding, a CRC can be assessed for the NPBCH channel to determine if the subframe has been damaged. In the case of the NPDCCH and NPDSCH, a CRC is only available for an entire transport block. Therefore, the damage of a subframe used for NPDCCH and NPDSCH can only be determined by a CRC if the transport blocks fit in a subframe. Nevertheless, it is important to acknowledge that when a transport block covers numerous subframes, it is feasible to determine the corruption of a specific subframe by employing NRS for interference estimation [3].

#### 6.6.1.2 IEEE 802.15.4

We simulate a PAN coordinator and a node receiving transmissions from the PAN coordinator via a GTS slot. The node should determine if it is under attack using the detector. We simulate an IEEE 802.15.4 beacon-enable MAC with GTS transmission. In our model, the node synchronises with the PAN Coordinator through the successful reception of a beacon. After that, the coordinator can communicate with the device using a GTS slot.

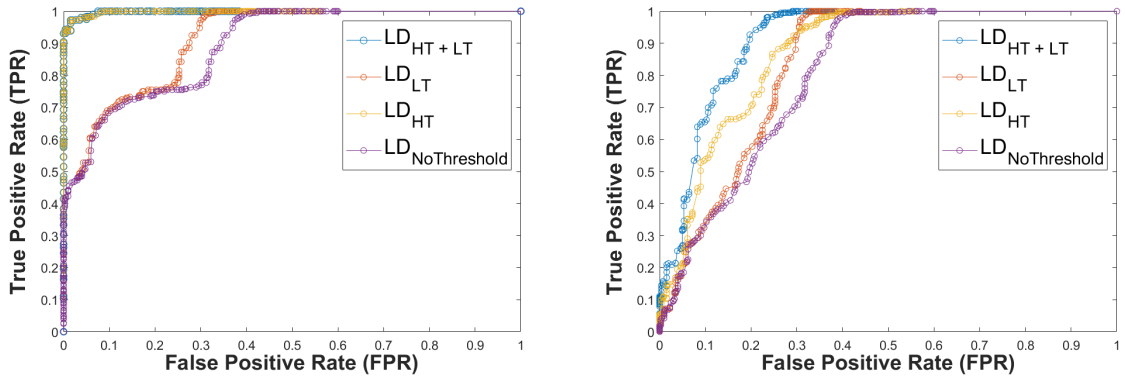
In our detector setup, the adaption layer defines the beacon messages and the specific GTS slot as two PDUs of interest. Receiving a message (either beacon or GTS transmission) is recorded a 1, a loss is recorded as 0 creating our sequences $S_n$ for the two PDUs.

### 6.6.2 Noise and Attacks

The communication between the transmitter device (eNodeB or PAN coordinator) may be susceptible to interference. Three types of interference are considered.

- *Background Noise (BN):* refers to the presence of Additive White Gaussian Noise (AWGN) that remains constant while the transmission channel is active.
- *Background Traffic (BT):* refers to the presence of noise (interference) limited to specific time intervals. The duration of these noise periods and the intervals between them adhere to a Poisson process. This noise interference replicates the impact of interference caused by an additional communication network that is implemented in the currently analysed network space.
- *Targeted Attack (TA):* Interference signal that is present only in specific Protocol Data Unit (PDU)s, simulating a targeted attack on particular protocol elements.

Each interference type can be present at different Signal-to-noise Ratio (SNR) levels. The coexistence of different types of interference is possible.




(a) Constant MIB

(b) Selective MIB

(c) Combined MIB

(d) Constant GTS

Figure 6.4: Receiver Operating Characteristic (ROC) curves for NB-IoT and IEEE 802.15.4 GTS Jamming Attacks using Loss Rate Detector with different *HT* and *LT*.

## 6.7 Results

In this section, we describe the results of several experiments that were used to assess the detector's capability against different types of targeted jamming interference attacks.

In these experiments, we create several scenarios with and without targeted interference attacks; we test the constant jamming attack for NB-IoT and IEEE 802.15.4 GTS, as well as the selective jamming attack and a third scenario that combines constant and selective approaches to simulate a more realistic attack when the node does not know the nature of the attack for NB-IoT. Then, we evaluate our loss rate and pattern detector's detection capacity by determining whether it can detect an attack is happening (True Positive) or if it incorrectly asserts an attack is present (False Posi-

tive).

Considering that in a realistic scenario, even when a targeted attack is present, some background noise may also be present, we combine background noise (BN) and targeted attack (TA) noise to accomplish this. We create interference signals combining BN and TA at different SNR levels.

For NB-IoT, we generate three experiment sets with 800 transmission sequences each. i) 400 sequences in each set are attacks (400 Constant MIB, 400 Selective MIB and 200 Constant MIB and 200 Selective MIB in the third set) with TA-based SNR ranging from -9.75dB to 0B (steps of 0.25dB) and BN-based SNR ranging from -16dB to -10dB (steps of 2dB). This SNR setup guarantees that we are considering successful attacks. ii) 400 sequences in each set have no attack. We use here background noise (BN) based SNR ranging from 0dB to -4.75dB (steps of 0.25dB) and background traffic (BT) based SNR ranging from -5dB to -9.75dB(steps of 0.25dB).

For IEEE 802.15.4, we generate one set with 560 sequences, distributed as follows: i) 280 attacks with TA-based SNR ranging from 0.0125dB to 0.510dB (steps of 0.0125dB), and BN-based SNR ranging from 0.26dB to 0.45dB (steps of 0.01dB). ii) 280 sequences in which no attack is present. We use background noise (BN) based SNR ranging from 0.275dB to 0.525dB (steps of 0.0125dB) and background traffic (BT) based SNR ranging from 0.0125dB to 0.250dB(steps of 0.0125dB).

### 6.7.1 Loss Rate Detector

Figure 6.4 shows the result of our experiments in the form of a Receiver Operating Characteristic (ROC) curves for Narrowband-Internet of Things (NB-IoT) and IEEE 802.15.4 GTS. We use the Area Under the Curve (AUC) as one of the metrics to evaluate the detector design capabilities.

Figures 6.4a and 6.4b illustrate the ROC curves for the Loss Rate Detector (LD) applied to NB-IoT MIB Constant and Selective attack while Figure 6.4c shows the result if both attacks are present. Using our experimental data, we determined the

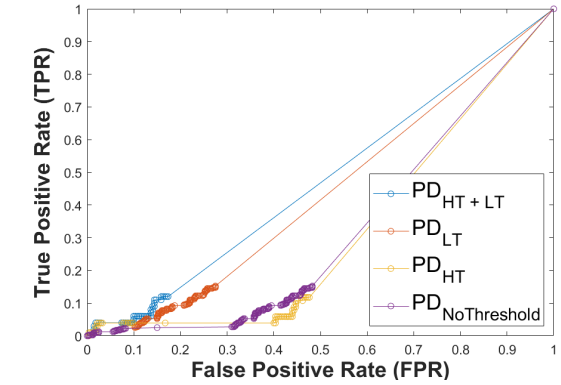
parameters $LT = 12.5$ and $HT = 24$ provide a good fit, and we used these settings for our LD Detector in NB-IoT. For the Constant jamming attack, it can be observed that the $LD_{HT+LT}$ setup performs better than the other setups. The detector can identify 93.04% of targeted interference cases with no false positives (an ideal setting as in a practical deployment false alarms should be avoided). For this scenario we obtain AUC = 99.79%).

For the Selective and Combined jamming attack, it is shown that the $LD_{HT+LT}$ detector performs better than the ones that consider only one of the thresholds ($LD_{HT}$ or $LD_{LT}$) or no threshold at all. The detector identifies 11.11% and 63.64% of targeted interference cases with no false positives, respectively. We obtain an AUC = 91.32% for the Selective attack and the AUC = 96.01% for the Combined attack.

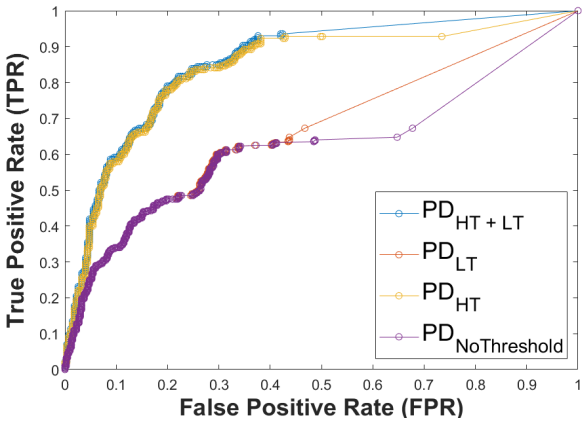
Figure 6.4d shows the Loss Rate Detector (LD) performance detection for IEEE 802.15.4 GTS attack. Using our experimental data, we determined the parameters $LT = 7.5$ and $HT = 55$ provide a good fit. It can be seen that the $LD_{HT+LT}$ detector performs better than the ones that consider one of the thresholds or does not threshold at all. For this optimal scenario, the detector is able to identify 55.17% of targeted interference cases with no false positives, with an AUC=97.44%.

### 6.7.2 Pattern Detector

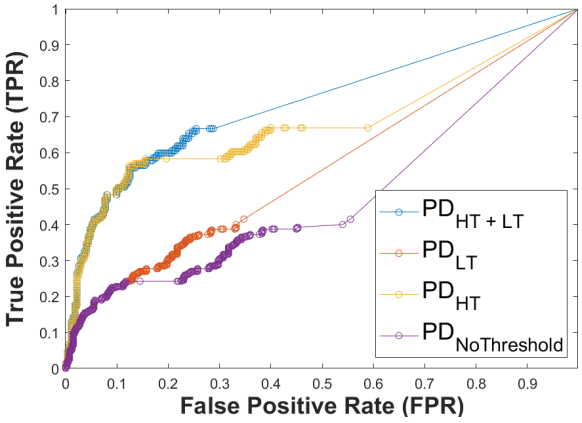
Figures 6.5a, 6.5b and 6.5c show the ROC curves for the Pattern Detector (PD) for NB-IoT MIB Constant, Selective and Combined targeted interference attack, respectively. Using our experimental data, we determined the parameters $LT = 7.5$ and $HT = 75$ provide a good fit. As expected, the detector underperforms against constant jamming attacks as it is unable to identify any pattern in the attack sequences. In this attack all MIB frames are destroyed resulting in a sequence $S_n$ of (mostly) 1. A good performance is observed for the Selective attack. In this scenario, when the detector applies both thresholds ($PD_{HT+LT}$), it results in an $AUC = 90.18\%$. For the combined attack, an acceptable performance is also observed with an $AUC = 95.41\%$.




(a) Constant MIB




(b) Selective MIB




(c) Combined

Figure 6.5: Receiver Operating Characteristic (ROC) curves for NB-IoT Jamming Attacks using Pattern Detector with different *HT* and *LT*.

### 6.7.3 Fused Detector

The *Fused Detector* concept is based on the notion that different jamming scenarios (e.g. constant and selective) might impact the performance of detection models differently. *Constant jamming* is typically characterised by a continuous high interference level, resulting in numerous packet losses, whereas *selective jamming* is more strategic, targeting specific packets based on content or timing, which may result in a pattern that can be statistically discerned.

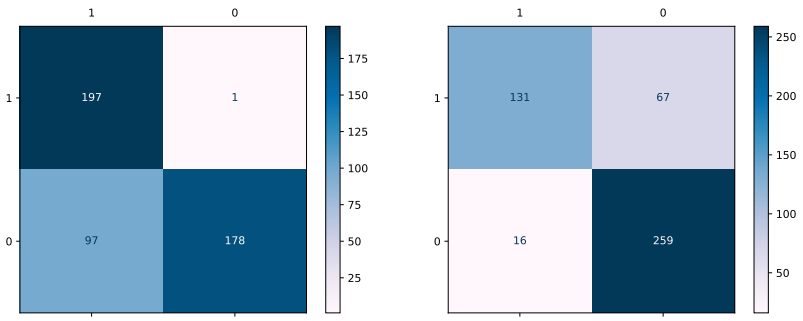
The primary objective of the *Fused Detector* is to leverage the strengths of both the *Loss Rate Detector* $(LD)$ and the *Pattern Detector* $(PD)$ in order to enhance the robustness and accuracy of jamming detection across varying scenarios.

The *Fused Detector* aims to capture a broader spectrum of jamming characteristics by integrating/merging the outputs of various models using logical operators, the fused detector aims to capture a broader spectrum of jamming characteristics. This integration also helps in overcoming the limitations of each individual detector when faced with diverse jamming strategies. The decision to use logical OR and AND gates is based on their fundamental properties: i) OR gate increases the sensitivity of the detection process, capturing any indication of jamming from either detector, whereas ii) the AND gate requires concurrent detection from both models, thus prioritizing precision over sensitivity.

#### 6.7.3.1 Fused Detector Logic

Mathematically, the outputs of the *Loss Rate Detector* and the *Pattern Detector* are sequences of binary values where 1 indicates the detection of jamming and 0 indicates no detection. Let $LD_\phi$ and $PD_\gamma$ denote the binary output from the *Loss Rate Detector* and the *Pattern Detector*, respectively. The *Fused Detector*, therefore, can be defined using the following logic gates:

**OR Gate Fused Detector:** $LD_\phi \vee PD_\gamma$




(a) OR: $\phi \cup \gamma$ (b) AND: $\phi \cap \gamma$

Figure 6.6: Confusion Matrix.

Here, the fused output is 1 if either $LD_\phi = 1$ or $PD_\gamma = 1$, indicating that any sign of jamming detected by either model results in a positive detection overall. This approach enhances the sensitivity of the detector, making it more responsive to various jamming activities.

**AND Gate Fused Detector:** $LD_\phi \wedge PD_\gamma$

Here, the fused output is 1 if only if both $LD_\phi = 1$ or $PD_\gamma = 1$, indicating a scenario where both detectors agree on the presence of jamming. This method emphasizes precision, reducing false positives but at the cost of decreased sensitivity.

#### 6.7.3.2 Results

Figure 6.6 shows the Confusion Matrix obtained for the optimal $LD_\phi$ and $PD_\gamma$ for the Combined MIB targeted jamming attack. The Equal Error Rate values for the LD and PD are EER=11.72% and EER=28.l78%, respectively. As can be seen, the confusion matrix for the union of the results of the two detectors presents a high rate of true positives and false negatives and a low rate of true negatives, which will translate into reducing false alarms. Meanwhile, a higher level of true negatives is observed for the intersection operation.

## 6.8 Conclusion

We have demonstrated that it is possible to construct a technology-independent statistical anomaly detector capable of detecting targeted interference on IoT devices by utilising the available data in the node as an indirect measure of interference. We have shown that our detectors have excellent performance in detecting known-characteristic attacks $AUC = 99.79\%$ and $AUC = 97.44\%$ for LR detector with Constant jamming attack in NB-IoT and IEEE 802.15.4 GTS, respectively; and $AUC = 90.18\%$ for PD detector with Selective jamming attack in NB-IoT. Also, we have shown that our detectors have adequate performance in detecting interference when the UE is unaware of the nature of the jamming attack; the Loss Rate detector has a AUC $= 96.01\%$ and can detect 63.64% of attacks without false positives, while the Pattern Detector has an AUC $= 95.41\%$. Finally, we have demonstrated that the threshold $HT$ has more impact than $LT$.

# 7 | Conclusion

The contributions of this thesis are focused on jamming attacks and targeted interference detection for wireless IoT. Throughout the chapters, we investigated the security issues related to jamming interference that IoT devices face, we addressed the question of how to implement a technology-independent targeted interference detector for IoT, and we explored new ways of designing anomaly detectors.

## 7.1 Conclusions

In terms of interference, this thesis addresses two fundamental challenges in wireless IoT communications: malicious jamming attacks and unintended radio interference (a.k.a natural interference). These issues critically compromise the reliability of IoT networks and delay the deployment of pervasive applications reliant on such systems. Despite significant research interest in the disruptive effects of jamming, this thesis introduces a novel paradigm that employs malicious interference selectively and natural interference occurrence to develop a lightweight detection mechanism specifically designed for implementation on the end-user side. This approach is particularly tailored to accommodate the resource constraints typical of IoT environments. The methodology presented here expands the understanding of how interference can be strategically utilized and delivers a practical solution to strengthen security against jamming a predominant threat in communication systems. This research contributes significantly to the existing body of knowledge, offering an innovative defensive strategy against malicious jamming attacks. Furthermore, it underscores the importance of

developing lightweight, user-centric solutions essential in the evolving landscape of technological threats posed by malicious jamming.

We critically evaluate five IoT monitoring systems, focusing on intrusion detection, localisation, interference, performance, and authentication. For the specific case of interference, the analysis reveals that effective interference detection should be deployed directly on IoT devices where interference typically occurs, enabling immediate and practical monitoring and analysis at the source of interference. This decision avoids the challenges associated with data transmission to a central analysis hub, such as increased bandwidth usage and potential delays or single points of failure, which can compromise detection timeliness and reliability. Our findings support the use of simple, direct monitoring metrics like signal loss, which are used by local statistical methods rather than complex AI-driven analytics suited for base stations. These local methods enhance resource efficiency and reduce threat detection and mitigation latency.

This research has demonstrated the vulnerability of NB-IoT devices to interference with their synchronisation signals, which are necessary for establishing initial communication with the base stations. Our findings show that a selectively designed jamming attack can effectively disrupt communication among these devices. Furthermore, our experiments indicated that by precisely tailoring the interference signal, it is feasible to manipulate the UE to recognize a specific NB-PCID, suggesting a sophisticated method of interference that could be exploited by attackers.

Moving further, we focused on designing jamming attacks to achieve specific disruptive outcomes and explore detection mechanisms for NPBCH and NPDSCH signals related interference. Our research has successfully demonstrated the feasibility of constructing a *statistical anomaly detector for NB-IoT devices*, utilizing data from the UE as an indirect measure of interference. The detector achieved an Equal Error Rate (EER) of 25% and could detect 50% of targeted attacks without any false positives. Currently, the detector can identify attacks on NPBCH with over 99% accuracy (Con-

stant MIB).

While extending the scope of our research by considering IEEE 802.15.4 networks, we have successfully developed a technology independent statistical anomaly detector that efficiently identifies targeted interference in IoT devices. This was accomplished by maintaining the same methodology used for the NB-IoT detector, which means leveraging the data available in each node as an indirect metric to measure interference effects. Our study provides compelling evidence of the detector's (Loss Rate Detector and Pattern Detector) robustness across various scenarios and two network protocols. Specifically, the performance of our detectors in recognizing attacks with known characteristics (constant/selective/combined attacks) was outstanding. For instance, the Loss Rate Detector (LD) detector exhibited an Area Under the Curve (AUC) of 99.79% and 97.44% in detecting constant jamming attacks on NB-IoT and IEEE 802.15.4 GTS systems, respectively. Similarly, the Pattern Detector (PD) managed an AUC of 90.18% against selective jamming attacks on NB-IoT. Moreover, our detectors proved to be effective even in scenarios where the user equipment (UE) was unaware of the nature of the jamming attack. In such cases, for NB-IoT, the Loss Rate Detector (LD) achieved an AUC of 96.01%, successfully detecting 63.64% of attacks with zero false positives. The Pattern Detector demonstrated a similar proficiency with an AUC of 95.41%. Our findings also highlight the critical role of threshold settings in the performance of our detectors, with the high threshold ($HT$) showing a greater impact than the low threshold ($LT$) on detection capabilities.

The Loss Rate detector, in particular, demonstrated exceptional performance across a range of jamming scenarios. It exhibited superior detection capabilities for constant, selective, and combined jamming attacks, achieving high detection accuracy. This detector's broad effectiveness underscores its potential as a versatile tool in securing TDMA-based IoT systems against various forms of interference. On the other hand, the Pattern detector showed a dedicated capability for detecting selective jamming attacks. While it provides valuable insights into specific interference patterns, its application is more limited compared to the broader capabilities of the Loss Rate de-

tector. Given these observations, the Loss Rate detector emerges as a highly effective, technology-independent solution for addressing targeted interference in TDMA Wireless IoT environments. Its ability to perform reliably across different types of jamming attacks suggests that it can serve as a generic tool for enhancing the resilience of IoT networks against a wide spectrum of security threats.

Overall, this research demonstrates the feasibility and efficacy of deploying statistical anomaly detectors to safeguard IoT systems against interference, providing a significant contribution to the field of cybersecurity in IoT networks. This provides a solid foundation for future research on adaptive, real-time security mechanisms for emerging IoT applications.

## 7.2 Future Work

Future work in securing IoT networks against interference could pursue several promising directions.

- Firstly, expanding the scope of interference types and IoT protocols covered by the developed anomaly detectors could further broaden and enhance the applicability of the proposed solution across various IoT applications and industries. Exploring the integration of these detectors with emerging technologies such as 6G and beyond, where interference might be more complex due to denser networks and mixed-service deployments, could also provide valuable insights.

- Moreover, there is a need to focus on reducing the energy consumption and computational requirements of the detection mechanisms to ensure they are viable on resource constrained IoT devices. This could involve optimizing algorithmic efficiency that provides both security and minimal overhead.

- Furthermore, conducting field trials of the proposed solutions in real-world IoT deployments would be beneficial to validate their effectiveness and robustness in operational environments. This would help in understanding the practical

challenges under typical usage conditions.

- The evaluation of experimental platforms in future work could involve several challenges. Firstly, conducting experiments in a real testbed requires access to an NB-IoT UE and an eNodeB with specific characteristics. For example, the NB-IoT node should have the capability to allow access to the data (e.g., binary sequence indicating PDU's success/failure) needed to implement the adapter module. Obtaining the necessary data from an NB-IoT node to implement the detector is critical; however, not all NB-IoT UEs permit access to the sequences of $0s$ and $1s$ needed for this purpose or implement our algorithms to process these sequences. Therefore, acquiring an open-source UE that facilitates the implementation of the adapter module is crucial. Although implementing the detector itself might be relatively straightforward, implementing the adapter module to access the required data poses a significant challenge.

  Experimental platforms might utilise SDR-based NB-IoT, necessitating a UE capable of handling interference. An attacker must be able to effectively interfere with the UE, considering that interference is illegal in real-world scenarios. Therefore, the experimentation must either occur within a protected frequency chamber to contain emissions or shift to a permissible frequency band, such as 2.4 GHz, that can mimic the IoT network's normal behaviour. Implementing the platform can involve multiple Software Define Radios (SDRs) functioning as the eNodeB, targeted acUE, and jammer, using the srsRAN open-source protocol stack for communication. An important consideration is the transition from licensed to unlicensed frequency bands.

  HackRF One SDRs, provided by Great Scott Gadgets, could be well-suited for this implementation [143]. These devices can transmit and receive radio waves between 1 MHz and 6 GHz, tuned for the open frequency of 868 MHz and calibrated to optimise signal quality. However, since HackRF One is a half-duplex SDR, it can only transmit or receive data at any given time. Utilising srsRAN,

which includes UE and eNodeB protocol stacks, in conjunction with third-party core network solutions, can help build comprehensive end-to-end mobile wireless networks, providing a robust foundation for evaluating experimental platforms.

# Bibliography


[1] IEEE, "IEEE Standard for Low-Rate Wireless Networks," *IEEE Std 802.15.4-2020 (Revision of IEEE Std 802.15.4-2015)*, pp. 1–800, 2020.

[2] G. Morillo and U. Roedig, "Jamming of NB-IoT Synchronisation Signals," in *Computer Security. 26th European Symposium on Research in Computer Security ESORICS 2021 International, Darmstadt, Germany, October 4–8, 2021*. Springer Int. Publishing, 2021, pp. 759–763.

[3] G. Morillo, U. Roedig, and D. Pesch, "Detecting Targeted Interference in NB-IoT," in *2023 19th International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT)*. Institute of Electrical and Electronics Engineers IEEE, 2023, pp. 475–482.

[4] F. Meneghello, M. Calore, D. Zucchetto, M. Polese, and A. Zanella, "IoT: Internet of Threats? A Survey of Practical Security Vulnerabilities in Real IoT Devices," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8182–8201, 2019.

[5] C. Pielli, D. Zucchetto, A. Andrea Zanella, L. Vangelista, and M. Zorzi, "Platforms and Protocols for the Internet of Things," *EAI Endorsed Transactions on Internet of Things*, vol. 1, no. 1, 2015.

[6] S. Ray, S. Bhunia, Y. Jin, and M. Tehranipoor, "Security validation in IoT space," in *2016 IEEE 34th VLSI Test Symposium (VTS)*, 2016, pp. 1–1.

[7] E. Vasilomanolakis, J. Daubert, M. Luthra, V. Gazis, A. Wiesmaier, and P. Kikiras,

"On the Security and Privacy of Internet of Things Architectures and Systems," in *2015 International Workshop on Secure Internet of Things (SIoT)*, 2015, pp. 49–57.

[8] R. Billure, V. M. Tayur, and M. V, "Internet of Things - a study on the security challenges," in *2015 IEEE International Advance Computing Conference (IACC)*, 2015, pp. 247–252.

[9] Rachit, S. Bhatt, and P. R. Ragiri, "Security trends in Internet of Things: a survey," *SN Applied Sciences*, vol. 3, pp. 1–14, 2021.

[10] M. N. Aman, K. C. Chua, and B. Sikdar, "Position Paper: Physical Unclonable Functions for IoT Security," in *Proceedings of the 2nd ACM International Workshop on IoT Privacy, Trust, and Security*, ser. IoTPTS '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 10–13. [Online]. Available: https://doi.org/10.1145/2899007.2899013

[11] S. S. Dhanda, B. Singh, and P. Jindal, "Lightweight Cryptography: A Solution to Secure IoT," *Wireless Personal Communications*, vol. 112, no. 3, pp. 1947–1980, Jun. 2020. [Online]. Available: https://doi.org/10.1007/s11277-020-07134-3

[12] V. A. Thakor, M. A. Razzaque, and M. R. A. Khandaker, "Lightweight Cryptography Algorithms for Resource-Constrained IoT Devices: A Review, Comparison and Research Opportunities," *IEEE Access*, vol. 9, pp. 28 177–28 193, 2021.

[13] S. Shamshad, F. Riaz, R. Riaz, S. S. Rizvi, and S. Abdulla, "A hybrid architecture for resolving Cryptographic issues in internet of things (IoT), Employing Quantum computing supremacy," in *2021 International Conference on Information and Communication Technology Convergence (ICTC)*, 2021, pp. 271–276.

[14] S. Shamsshad, F. Riaz, R. Riaz, S. S. Rizvi, and S. Abdulla, "An Enhanced Architecture to Resolve Public-Key Cryptographic Issues in the Internet of Things (IoT), Employing Quantum Computing Supremacy," *Sensors*, vol. 22, no. 21, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/21/8151

[15] M. Bhatia and S. K. Sood, "Quantum Computing-Inspired Network Optimization for IoT Applications," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5590–5598, 2020.

[16] O. García, J. Berrocal, J. M. García-Alonso, C. Canal, and J. M. Murillo, "Evaluation and application of autonomic computing techniques in a next-generation radio access network," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 1, pp. 1129–1140, 2020.

[17] I. K. Dutta, B. Ghosh, and M. Bayoumi, "Lightweight Cryptography for Internet of Insecure Things: A Survey," in *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, 2019, pp. 0475–0481.

[18] K. Zhang, J. Long, X. Wang, H.-N. Dai, K. Liang, and M. Imran, "Lightweight Searchable Encryption Protocol for Industrial Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 6, pp. 4248–4259, 2021.

[19] S. Sciancalepore, A. Capossele, G. Piro, G. Boggia, and G. Bianchi, "Key Management Protocol with Implicit Certificates for IoT Systems," in *Proceedings of the 2015 Workshop on IoT Challenges in Mobile and Industrial Systems*, ser. IoT-Sys '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 37–42. [Online]. Available: https://doi.org/10.1145/2753476.2753477

[20] S. Sciancalepore, G. Piro, G. Boggia, and G. Bianchi, "Public Key Authentication and Key Agreement in IoT Devices With Minimal Airtime Consumption," *IEEE Embedded Systems Letters*, vol. 9, no. 1, pp. 1–4, 2017.

[21] S. Khan, W.-K. Lee, and S. O. Hwang, "Scalable and Efficient Hardware Architectures for Authenticated Encryption in IoT Applications," *IEEE Internet of Things Journal*, vol. 8, no. 14, pp. 11 260–11 275, 2021.

[22] K. T. Nguyen, M. Laurent, and N. Oualha, "Survey on secure communication protocols for the Internet of Things," *Ad Hoc Networks*, vol. 32, pp. 17–31, 2015, internet of Things security and privacy: design methods and optimization.

[23] 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*, 2015, pp. 746–751.

[24] O. Lucia, B. Isong, N. Gasela, and A. M. Abu-Mahfouz, "Device Authentication Schemes in IoT: A Review," in *2019 International Multidisciplinary Information Technology and Engineering Conference (IMITEC)*, 2019, pp. 1–6.

[25] N. Li, D. Liu, and S. Nepal, "Lightweight Mutual Authentication for IoT and Its Applications," *IEEE Transactions on Sustainable Computing*, vol. 2, no. 4, pp. 359–370, 2017.

[26] M. N. Aman, M. H. Basheer, and B. Sikdar, "Two-Factor Authentication for IoT With Location Information," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3335–3351, 2019.

[27] R. Das, A. Gadre, S. Zhang, S. Kumar, and J. M. F. Moura, "A Deep Learning Approach to IoT Authentication," in *2018 IEEE International Conference on Communications (ICC)*, 2018, pp. 1–6.

[28] T. Dimitrakos, T. Dilshener, A. Kravtsov, A. La Marra, F. Martinelli, A. Rizos, A. Rosetti, and A. Saracino, "Trust Aware Continuous Authorization for Zero Trust in Consumer Internet of Things," in *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2020, pp. 1801–1812.

[29] C. Camara, H. Martín, P. Peris-Lopez, and L. Entrena, "A True Random Number Generator Based on Gait Data for the Internet of You," *IEEE Access*, vol. 8, pp. 71 642–71 651, 2020.

[30] K. Wallace, K. Moran, E. Novak, G. Zhou, and K. Sun, "Toward Sensor-Based Random Number Generation for Mobile and IoT Devices," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 1189–1201, 2016.

[31] L. Du, Y. Du, Y. Li, J. Su, Y.-C. Kuan, C.-C. Liu, and M.-C. F. Chang, "A Reconfigurable Streaming Deep Convolutional Neural Network Accelerator for Internet of Things," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 1, pp. 198–208, 2018.

[32] C. Profentzas, M. Günes, Y. Nikolakopoulos, O. Landsiedel, and M. Almgren, "Performance of Secure Boot in Embedded Systems," in *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 2019, pp. 198–204.

[33] J. Lim, S. Sohn, and J. Kim, "Proposal of Smart Segmentation Framework for preventing threats from spreading in IoT," in *2020 International Conference on Information and Communication Technology Convergence (ICTC)*, 2020, pp. 1745–1747.

[34] N. Wagner, C. S. Sahin, M. Winterrose, J. Riordan, J. Pena, D. Hanson, and W. W. Streilein, "Towards automated cyber decision support: A case study on network segmentation for security," in *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2016, pp. 1–10.

[35] B. B. Zarpelao, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, "A Survey of Intrusion Detection in Internet of Things," *J. Netw. Comput. Appl.*, vol. 84, no. C, p. 25–37, Apr. 2017. [Online]. Available: https://doi.org/10.1016/j.jnca.2017.02.009

[36] M. Sarrab and S. M. Alnaeli, "Critical Aspects Pertaining Security of IoT Application Level Software Systems," in *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, 2018, pp. 960–964.

[37] M. Ge, X. Fu, N. Syed, Z. Baig, G. Teo, and A. Robles-Kelly, "Deep Learning-Based Intrusion Detection for IoT Networks," in *2019 IEEE 24th Pacific Rim International Symposium on Dependable Computing (PRDC)*, 2019, pp. 256–25 609.

[38] H. Qiu, T. Dong, T. Zhang, J. Lu, G. Memmi, and M. Qiu, "Adversarial Attacks Against Network Intrusion Detection in IoT Systems," *IEEE Internet of Things Journal*, vol. 8, no. 13, pp. 10 327–10 335, 2021.

[39] E. Anthi, L. Williams, M. Słowińska, G. Theodorakopoulos, and P. Burnap, "A Supervised Intrusion Detection System for Smart Home IoT Devices," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 9042–9053, 2019.

[40] A. Mudgerikar, P. Sharma, and E. Bertino, "Edge-Based Intrusion Detection for IoT devices," *ACM Trans. Manage. Inf. Syst.*, vol. 11, no. 4, oct 2020. [Online]. Available: https://doi.org/10.1145/3382159

[41] O. Liberg, M. Sundberg, Y.-P. E. Wang, J. Bergman, J. Sachs, and G. Wikstroem, *Cellular internet of things: From massive deployments to critical 5G applications*. Academic Press, an imprint of Elsevier, 2020.

[42] T. Pedersen, B. Fleury, and COST Action CA15104, IRACON, "Whitepaper on New Localization Methods for 5G Wireless Systems and the Internet-of-Things," 2018.

[43] R. Srinivasan, B. Mallikarjuna, M. Kavitha, R. Kavitha, and B. B. Naib, "A Comparative Study: Wireless Technologies in Internet of Things," in *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, 2022, pp. 675–679.

[44] M. Bor, J. Vidler, and U. Roedig, "LoRa for the Internet of Things," in *Proceedings of the 2016 International Conference on Embedded Wireless Systems and Networks*, ser. EWSN '16. USA: Junction Publishing, 2016, p. 361–366.

[45] C. M. Ramya, M. Shanmugaraj, and R. Prabakaran, "Study on ZigBee technology," in *2011 3rd International Conference on Electronics Computer Technology*, vol. 6, 2011, pp. 297–301.

[46] K.-L. Tsai, F.-Y. Leu, I. You, S.-W. Chang, S.-J. Hu, and H. Park, "Low-Power AES Data Encryption Architecture for a LoRaWAN," *IEEE Access*, vol. 7, pp. 146 348–146 357, 2019.

[47] M. J. H. Biddut, N. Islam, R. S. Sultana, A. Sarker, and M. M. Rahman, "A new approach of ZigBee MAC layer design based on security enhancement," in *2015 IEEE International Conference on Telecommunications and Photonics (ICTP)*, 2015, pp. 1–5.

[48] Third Generation Partnership Project, "Technical Report 45.820, v13.0.0. Cellular system support for ultra-low complexity and low throughput Internet of Things," 3GPP, Tech. Rep., 2016, version 13.0.0.

[49] 3GPP, "Standardization of NB-IoT completed," Online, 2016, available from: http://www.3gpp.org/news-events/3gpp-news/1785-nb_iot_complete.

[50] E. Dahlman, S. Parkvall, and J. Skold, *4G: LTE/LTE-Advanced for Mobile Broadband*. Academic Press, Mar 2011.

[51] "ETSI TS 136 211 V14.2.0, LTE Evolved Universal Terrestrial Radio Access (E-UTRA), Physical channels and modulation (3GPP TS 36.211 version 14.2.0 Release 14)," 2017. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2719

[52] G. Third Generation Partnership Project (a), "3GPP TS 36.104, "Evolved Universal Terrestrial Radio Access (E-UTRA); Base Station (BS) radio transmission and reception (Release 13)"," 2017.

[53] G. Third Generation Partnership Project (b), "3GPP TS 36.141, "Evolved Universal Terrestrial Radio Access (E-UTRA); Medium Access Control (MAC) protocol specification (Release 13)"," 2018.

[54] G. Third Generation Partnership Project (c), "3GPP TS 36.321, "Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Link Control (RLC) protocol specification (Release 13)"," 2018.

[55] S. Fan, J. Li, and H. Sun, "Study and Simulation of GTS Allocation in Beacon Enabled IEEE 802.15.4," in *2010 6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM)*, 2010, pp. 1–5.

[56] R. Sokullu, O. Dagdeviren, and I. Korkmaz, "On the IEEE 802.15.4 MAC Layer Attacks: GTS Attack," in *2008 Second International Conference on Sensor Technologies and Applications*, 2008, pp. 673–678.

[57] S. Liu, L. Xiao, Z. Han, and Y. Tang, "Eliminating NB-IoT Interference to LTE System: A Sparse Machine Learning-Based Approach," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6919–6932, 2019.

[58] J. A. Stankovic, "Research Directions for the Internet of Things," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 3–9, 2014.

[59] B. Vejlgaard, M. Lauridsen, H. Nguyen, I. Z. Kovacs, P. Mogensen, and M. Sorensen, "Interference Impact on Coverage and Capacity for Low Power Wide Area IoT Networks," in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, 2017, pp. 1–6.

[60] A. Mishra, E. Rozner, S. Banerjee, and W. Arbaugh, "Exploiting partially overlapping channels in wireless networks: turning a peril into an advantage," in *Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC '05. USA: USENIX Association, 2005, p. 29.

[61] Y. Wang, Y. Zeng, S. Sun, P. H. Tan, and E. Kurniawan, "RF Interference Detection and Signal Classification for IIoT Application," in *2018 24th Asia-Pacific Conference on Communications (APCC)*, 2018, pp. 150–155.

[62] R. Dionísio, T. Lolić, and P. Torres, "Electromagnetic Interference Analysis of Industrial IoT Networks: From Legacy Systems to 5G," in *2020 IEEE Microwave Theory and Techniques in Wireless Communications (MTTW)*, vol. 1, 2020, pp. 41–46.

[63] C. Hwang and J. Fan, "Modeling and Mitigation of Radio Frequency Interference for Wireless Devices," *IEEE Electromagnetic Compatibility Magazine*, vol. 12, no. 1, pp. 87–92, 2023.

[64] L. Zhang, A. Ijaz, P. Xiao, and R. Tafazolli, "Channel Equalization and Interference Analysis for Uplink Narrowband Internet of Things (NB-IoT)," *IEEE Communications Letters*, vol. 21, no. 10, pp. 2206–2209, 2017.

[65] M. O. Shahid, M. Philipose, K. Chintalapudi, S. Banerjee, and B. Krishnaswamy, "Concurrent interference cancellation: decoding multi-packet collisions in LoRa," in *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*, ser. SIGCOMM '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 503–515. [Online]. Available: https://doi.org/10.1145/3452296.3472931

[66] C. Zheng, M. Egan, L. Clavier, G. W. Peters, and J.-M. Gorce, "Copula-Based Interference Models for IoT Wireless Networks," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, 2019, pp. 1–6.

[67] K. A. Kumar, K. S. Chatrapati, and A. V. Krishna, "Multipath Interference Minimization in Heterogeneous Wireless Sensor Networks for Reliable Data Transfer," in *2016 International Conference on Computer and Communication Engineering (ICCCE)*, 2016, pp. 261–266.

[68] A. B. Mohanoor, S. Radhakrishnan, and V. Sarangan, "Interference aware multipath routing in wireless networks," in *2008 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems*, 2008, pp. 516–518.

[69] Y. Lv, H. Qin, Z. Liu, Y. Dong, W. Feng, and W. Xu, "An Efficient Synchronization Algorithm in OFDM-Based NB-IoT Systems," in *2019 IEEE Intl Conf on Depend-*

*able, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, 2019, pp. 459–464.

[70] J. Van Waes, J. Vankeirsbilck, D. Pissoort, and J. Boydens, "Electromagnetic interference in the Internet of Things: An automotive insight," in *2017 XXVI International Scientific Conference Electronics (ET)*, 2017, pp. 1–4.

[71] M. Mehri and A. Amini, "Stochastic EMI Noise Model of PCB Layout for Circuit-Level Analysis of System in IoT Applications," *IEEE Transactions on Microwave Theory and Techniques*, vol. 68, no. 12, pp. 5072–5081, 2020.

[72] A. P. Mynster and P. T. Jensen, "EMC for the IoT," in *2016 International Symposium on Electromagnetic Compatibility - EMC EUROPE*, 2016, pp. 144–149.

[73] X. Tang, P. Ren, and Z. Han, "Jamming Mitigation via Hierarchical Security Game for IoT Communications," *IEEE Access*, vol. 6, pp. 5766–5779, 2018.

[74] F. T. Zahra, Y. S. Bostanci, and M. Soyturk, "Real-Time Jamming Detection in Wireless IoT Networks," *IEEE Access*, vol. 11, pp. 70 425–70 442, 2023.

[75] D. S. Berger, F. Gringoli, N. Facchi, I. Martinovic, and J. Schmitt, "Gaining insight on friendly jamming in a real-world IEEE 802.11 network," in *Proceedings of the 2014 ACM Conference on Security and Privacy in Wireless & Mobile Networks*, ser. WiSec '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 105–116. [Online]. Available: https://doi.org/10.1145/2627393.2627403

[76] J. Brown, I. E. Bagci, A. King, and U. Roedig, "Defend your home! jamming unsolicited messages in the smart home," in *Proceedings of the 2nd ACM Workshop on Hot Topics on Wireless Network Security and Privacy*, ser. HotWiSec '13. New York, NY, USA: Association for Computing Machinery, 2013, p. 1–6. [Online]. Available: https://doi.org/10.1145/2463183.2463185

[77] S. Gollakota, H. Hassanieh, B. Ransford, D. Katabi, and K. Fu, "They can hear your heartbeats: non-invasive security for implantable medical devices," in *Proceedings of the ACM SIGCOMM 2011 Conference*, ser. SIGCOMM '11. New York, NY, USA: Association for Computing Machinery, 2011, p. 2–13. [Online]. Available: https://doi.org/10.1145/2018436.2018438

[78] M. Wilhelm, I. Martinovic, J. B. Schmitt, and V. Lenders, "WiFire: a firewall for wireless networks," *SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 4, p. 456–457, aug 2011. [Online]. Available: https://doi.org/10.1145/2043164.2018518

[79] W. Shen, P. Ning, X. He, and H. Dai, "Ally Friendly Jamming: How to Jam Your Enemy and Maintain Your Own Wireless Connectivity at the Same Time," in *Proceedings of the 2013 IEEE Symposium on Security and Privacy*, ser. SP '13. USA: IEEE Computer Society, 2013, p. 174–188. [Online]. Available: https://doi.org/10.1109/SP.2013.22

[80] I. Martinovic, P. Pichota, and J. B. Schmitt, "Jamming for good: a fresh approach to authentic communication in WSNs," in *WiSec 2009 ACM conference on Wireless network security*, ser. WiSec '09. New York, NY, USA: Association for Computing Machinery, 2009, p. 161–168. [Online]. Available: https://doi.org/10.1145/1514274.1514298

[81] J. P. Vilela, P. C. Pinto, and J. Barros, "Position-Based Jamming for Enhanced Wireless Secrecy," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 616–627, 2011.

[82] M. L. Jorgensen, B. R. Yanakiev, G. E. Kirkelund, P. Popovski, H. Yomo, and T. Larsen, "Shout to Secure: Physical-Layer Wireless Security with Known Interference," in *IEEE GLOBECOM 2007 - IEEE Global Telecommunications Conference*, 2007, pp. 33–38.

[83] Y. S. Kim, P. Tague, H. Lee, and H. Kim, "Carving secure wi-fi zones with defensive jamming," in *Proceedings of the 7th ACM Symposium on Information,*

*Computer and Communications Security*, ser. ASIACCS '12. New York, NY, USA: Association for Computing Machinery, 2012, p. 53–54. [Online]. Available: https://doi.org/10.1145/2414456.2414487

[84] R. Jin, K. Zeng, and K. Zhang, "A Reassessment on Friendly Jamming Efficiency," *IEEE Transactions on Mobile Computing*, vol. 20, no. 1, pp. 32–47, 2021.

[85] X. Li, H.-N. Dai, Q. Wang, M. Imran, D. Li, and M. A. Imran, "Securing Internet of Medical Things with Friendly-jamming schemes," *Computer Communications*, vol. 160, pp. 431–442, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366420310227

[86] Q. Wang, H.-N. Dai, H. Wang, G. Xu, and A. K. Sangaiah, "UAV-enabled friendly jamming scheme to secure industrial Internet of Things," *Journal of Communications and Networks*, vol. 21, no. 5, pp. 481–490, 2019.

[87] M. Li, I. Koutsopoulos, and R. Poovendran, "Optimal Jamming Attack Strategies and Network Defense Policies in Wireless Sensor Networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 8, pp. 1119–1133, 2010.

[88] N. Namvar, W. Saad, N. Bahadori, and B. Kelley, "Jamming in the Internet of Things: A Game-Theoretic Perspective," in *2016 IEEE Global Communications Conference (GLOBECOM)*, 2016, pp. 1–6.

[89] A. S. Ali, W. T. Lunardi, G. Singh, L. Bariah, M. Baddeley, M. A. Lopez, J.-P. Giacalone, and S. Muhaidat, "RF Jamming Dataset: A Wireless Spectral Scan Approach for Malicious Interference Detection," *IEEE Communications Magazine*, pp. 1–7, 2024.

[90] J. Singh, I. Woungang, S. K. Dhurandher, and K. Khalid, "A jamming attack detection technique for opportunistic networks," *Internet of Things*, vol. 17, p. 100464, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2542660521001050

[91] A. A. Fadele, M. Othman, I. A. T. Hashem, Y. Yaqoob, I. Muhammad, and S. Muhammad, "A novel countermeasure technique for reactive jamming attack in internet of things," *Multimedia Tools and Applications*, vol. 78, pp. 29 899–29 920, 2019.

[92] M. Strasser, B. Danev, and S. Čapkun, "Detection of reactive jamming in sensor networks," *ACM Trans. Sen. Netw.*, vol. 7, no. 2, sep 2010. [Online]. Available: https://doi.org/10.1145/1824766.1824772

[93] B. Upadhyaya, S. Sun, and B. Sikdar, "Machine Learning-based Jamming Detection in Wireless IoT Networks," in *2019 IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS)*, 2019, pp. 1–5.

[94] T. Hamza, G. Kaddoum, A. Meddeb, and G. Matar, "A Survey on Intelligent MAC Layer Jamming Attacks and Countermeasures in WSNs," in *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*, 2016, pp. 1–5.

[95] W. Li, J. Wang, L. Li, X. Chen, W. Huang, and S. Li, "Countermeasure for Smart Jamming Threat: A Deceptively Adversarial Attack Approach," in *ICC 2021 - IEEE International Conference on Communications*, 2021, pp. 1–6.

[96] E. Hodo, X. Bellekens, A. Hamilton, P. 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)*, 2016, pp. 1–6.

[97] L. Santos, C. Rabadao, and R. Gonçalves, "Intrusion detection systems in Internet of Things: A literature review," in *2018 13th Iberian Conference on Information Systems and Technologies (CISTI)*, 2018, pp. 1–7.

[98] O. Faraj, D. Megías, A.-M. Ahmad, and J. Garcia-Alfaro, "Taxonomy and Challenges in Machine Learning-Based Approaches to Detect Attacks in the Internet of Things," in *Proceedings of the 15th International Conference on Availability, Reliability and Security*, ser. ARES '20. New York, NY, USA:

Association for Computing Machinery, 2020. [Online]. Available: https://doi.org/10.1145/3407023.3407048

[99] S. Khanam, I. B. Ahmedy, M. Y. Idna Idris, M. H. Jaward, and A. Q. Bin Md Sabri, "A Survey of Security Challenges, Attacks Taxonomy and Advanced Countermeasures in the Internet of Things," *IEEE Access*, vol. 8, pp. 219 709–219 743, 2020.

[100] M. Nawir, A. Amir, N. Yaakob, and O. B. Lynn, "Internet of Things (IoT): Taxonomy of security attacks," in *2016 3rd International Conference on Electronic Design (ICED)*, 2016, pp. 321–326.

[101] A. Ferdowsi and W. Saad, "Generative Adversarial Networks for Distributed Intrusion Detection in the Internet of Things," in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.

[102] D. Midi, A. Rullo, A. Mudgerikar, and E. Bertino, "Kalis — A System for Knowledge-Driven Adaptable Intrusion Detection for the Internet of Things," in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, 2017, pp. 656–666.

[103] P. A. Networks, "The palo alto networks approach to intrusion prevention," https://www.paloaltonetworks.com/apps/pan/public/downloadResource?pagePath=/content/pan/en_US/resources/whitepapers/palo-alto-networks-approach-to-intrusion-prevention, accessed: 2024-07-29.

[104] Fortinet, "Iot solution for enterprise and midsize business," https://www.fortinet.com/solutions/enterprise-midsize-business/iot-solution, accessed: 2024-07-29.

[105] Ericsson, "Ericsson's iot businesses are now part of the new aeris," https://www.ericsson.com/en/news/2023/4/ericssons-iot-businesses-are-now-part-of-the-new-aeris, accessed: 2024-07-29.

[106] R. C. Shit, S. Sharma, D. Puthal, and A. Y. Zomaya, "Location of Things (LoT): A Review and Taxonomy of Sensors Localization in IoT Infrastructure," *IEEE Communications Surveys Tutorials*, vol. 20, no. 3, pp. 2028–2061, 2018.

[107] F. Khelifi, A. Bradai, A. Benslimane, P. Rawat, and M. Atri, "A Survey of Localization Systems in Internet of Things," *Mobile Networks and Applications*, vol. 24, pp. 761–785, 2019.

[108] F. Zafari, A. Gkelias, and K. K. Leung, "A Survey of Indoor Localization Systems and Technologies," *IEEE Communications Surveys Tutorials*, vol. 21, no. 3, pp. 2568–2599, 2019.

[109] K. N. Choi, H. Kolamunna, A. Uyanwatta, K. Thilakarathna, S. Seneviratne, R. Holz, M. Hassan, and A. Y. Zomaya, "LoRadar: LoRa Sensor Network Monitoring through Passive Packet Sniffing," *SIGCOMM Comput. Commun. Rev.*, vol. 50, no. 4, p. 10–24, Oct. 2020. [Online]. Available: https://doi.org/10.1145/3431832.3431835

[110] D. Bissett, "Analysing TDoA localisation in LoRa networks," *Master Thesis Delft University of Technology*, 2018.

[111] T. Janssen, M. Weyn, and R. Berkvens, "A Primer on Real-world RSS-based Outdoor NB-IoT Localization," in *2020 International Conference on Localization and GNSS (ICL-GNSS)*, 2020, pp. 1–6.

[112] M. T. Hammi, B. Hammi, P. Bellot, and A. Serhrouchni, "Bubbles of Trust: A decentralized blockchain-based authentication system for IoT," *Comput. Secur.*, vol. 78, pp. 126–142, 2018.

[113] S. Agrawal and P. Ahlawat, "A Survey on the Authentication Techniques in Internet of Things," in *2020 IEEE International Students' Conference on Electrical,Electronics and Computer Science (SCEECS)*, 2020, pp. 1–5.

[114] D. Nouichi, M. Abdelsalam, Q. Nasir, and S. Abbas, "IoT Devices Security Using RF Fingerprinting," in *2019 Advances in Science and Engineering Technology International Conferences (ASET)*, 2019, pp. 1–7.

[115] L. Gonzalez-Manzano, J. M. D. Fuentes, and A. Ribagorda, "Leveraging User-Related Internet of Things for Continuous Authentication: A Survey," *ACM Comput. Surv.*, vol. 52, no. 3, Jun. 2019. [Online]. Available: https://doi.org/10.1145/3314023

[116] B. Yu, C. Yang, and J. Ma, "Continuous Authentication for the Internet of Things Using Channel State Information," in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.

[117] Y. Jiang, L. Peng, A. Hu, S. Wang, Y. Huang, and L. Zhang, "Physical layer identification of LoRa devices using constellation trace figure," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, pp. 1–11, 2019.

[118] G. Baldini, R. Giuliani, G. Steri, and R. Neisse, "Physical layer authentication of Internet of Things wireless devices through permutation and dispersion entropy," in *2017 Global Internet of Things Summit (GIoTS)*, 2017, pp. 1–6.

[119] T. Voigt, M. Bor, U. Roedig, and J. Alonso, "Mitigating Inter-Network Interference in LoRa Networks," in *Proceedings of the 2017 International Conference on Embedded Wireless Systems and Networks*, ser. EWSN '17. USA: Junction Publishing, 2017, p. 323–328.

[120] F. Caglar, S. Shekhar, A. Gokhale, and X. Koutsoukos, "Intelligent, Performance Interference-Aware Resource Management for IoT Cloud Backends," in *2016 IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI)*, 2016, pp. 95–105.

[121] S. Baidya and M. Levorato, "Content-Based Cognitive Interference Control for City Monitoring Applications in the Urban IoT," in *2016 IEEE Global Communications Conference (GLOBECOM)*, 2016, pp. 1–6.

[122] A. D'Alconzo, I. Drago, A. Morichetta, M. Mellia, and P. Casas, "A Survey on Big Data for Network Traffic Monitoring and Analysis," *IEEE Transactions on Network and Service Management*, vol. 16, no. 3, pp. 800–813, 2019.

[123] P. Andres-Maldonado, P. Ameigeiras, J. Prados-Garzon, J. Navarro-Ortiz, and J. M. Lopez-Soler, "An Analytical Performance Evaluation Framework for NB-IoT," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 7232–7240, 2019.

[124] Y. SARR, B. GUEYE, and C. SARR, "Performance Analysis of a Smart Street Lighting Application Using LoRa Wan," in *2019 International Conference on Advanced Communication Technologies and Networking (CommNet)*, 2019, pp. 1–6.

[125] E. D. Ayele, C. Hakkenberg, J. P. Meijers, K. Zhang, N. Meratnia, and P. J. M. Havinga, "Performance analysis of LoRa radio for an indoor IoT applications," in *2017 International Conference on Internet of Things for the Global Community (IoTGC)*, 2017, pp. 1–8.

[126] A. Jamalipour and S. Murali, "A Taxonomy of Machine-Learning-Based Intrusion Detection Systems for the Internet of Things: A Survey," *IEEE Internet of Things Journal*, vol. 9, no. 12, pp. 9444–9466, 2022.

[127] M. El-hajj, A. Fadlallah, M. Chamoun, and A. Serhrouchni, "A Survey of Internet of Things (IoT) Authentication Schemes," *Sensors*, vol. 19, no. 5, 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/5/1141

[128] N. Alsaeed and F. Nadeem, "Authentication in the Internet of Medical Things: Taxonomy, Review, and Open Issues," *Applied Sciences*, vol. 12, no. 15, 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/15/7487

[129] S. Ghorpade, M. Zennaro, and B. Chaudhari, "Survey of Localization for Internet of Things Nodes: Approaches, Challenges and Open Issues," *Future Internet*, vol. 13, no. 8, 2021. [Online]. Available: https://www.mdpi.com/1999-5903/13/8/210

[130] R. C. Shit, S. Sharma, D. Puthal, and A. Y. Zomaya, "Location of Things (LoT): A Review and Taxonomy of Sensors Localization in IoT Infrastructure," *IEEE Communications Surveys and Tutorials*, vol. 20, no. 3, pp. 2028–2061, 2018.

[131] V. R. Stoynov, V. K. Poulkov, and Z. V. Valkova-Jarvis, "Interference Management in LoRaWANs — Overview and Simulation Study," in *2018 Advances in Wireless and Optical Communications (RTUWO)*, 2018, pp. 251–256.

[132] A. King, J. Brown, J. Vidler, and U. Roedig, "Estimating node lifetime in interference environments," in *2015 IEEE 40th Local Computer Networks Conference Workshops (LCN Workshops)*, 2015, pp. 796–803.

[133] J. Brown, U. Roedig, C. A. Boano, and K. Römer, "Estimating packet reception rate in noisy environments," in *39th Annual IEEE Conference on Local Computer Networks Workshops*, 2014, pp. 583–591.

[134] C. A. Boano, T. Voigt, C. Noda, K. Römer, and M. Zúñiga, "JamLab: Augmenting sensornet testbeds with realistic and controlled interference generation," in *Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks*, 2011, pp. 175–186.

[135] J. de C. Silva, P. H. M. Pereira, L. L. de Souza, C. N. M. Marins, G. A. Marcondes, and J. J. Rodrigues, "Performance Evaluation of IoT Network Management Platforms," in *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2018, pp. 259–265.

[136] A. A. Ismail, H. S. Hamza, and A. M. Kotb, "Performance Evaluation of Open Source IoT Platforms," in *2018 IEEE Global Conference on Internet of Things (GCIoT)*, 2018, pp. 1–5.

[137] T. G. P. Project, "3GPP: Cellular system support for ultra-low complexity and low throughput Internet of Things (CIoT) (Release 13)," *Technical Specification Group GSM/EDGE Radio Access Network, 3GPP TR 45.820 V13.1.0.*, vol. 45, p. 820, 2015, 3GPP TR V13.1.0.

[138] J. Cao, P. Yu, M. Ma, and W. Gao, "Fast Authentication and Data Transfer Scheme for Massive NB-IoT Devices in 3GPP 5G Network," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1561–1575, 2019.

[139] M. Eygi and G. Karabulut-Kurt, "A Countermeasure against Smart Jamming Attacks on LTE Synchronization Signals," *J. Commun.*, vol. 15, pp. 626–632, 2020.

[140] M. Labib, V. Marojevic, J. H. Reed, and A. I. Zaghloul, "How to enhance the immunity of LTE systems against RF spoofing," in *2016 International Conference on Computing, Networking and Communications (ICNC)*, 2016, pp. 1–5.

[141] G. Third Generation Partnership Project, "Third Generation Partnership Project Standardization of NB-IoT Completed," 2016. [Online]. Available: https://www.3gpp.org/news-events/3gpp-news/

[142] The MathWorks Inc. Natick, "MATLAB: 9.7.0.1190202 (R2020b)," 2020.

[143] V. Ionescu and U. Roedig, "Battery Depletion Attacks On NB-IoT Devices Using Interference," in *ADIoT ESORICS 2021*, 2021, p. 276–295.

[144] V. Ionesscu and U. Roedig, "NB-IoT Battery Depletion via Malicious Interference," in *Proceedings of the 2022 International Conference on Embedded Wireless Systems and Networks EWSN*, 2022, p. 244–249.

[145] R. Krenz and S. Brahma, "Jamming LTE signals," in *2015 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, 2015, pp. 72–76.

[146] H. Alakoca, H. B. Tugrel, G. K. Kurt, and C. Ayyildiz, "CP and pilot jamming attacks on SC-FDMA: Performance tests with software defined radios," in *2016 10th International Conference on Signal Processing and Communication Systems (ICSPCS)*, 2016, pp. 1–6.

[147] G. Romero, V. Deniau, and O. Stienne, "LTE Physical Layer Vulnerability Test to Different Types of Jamming Signals," in *2019 International Symposium on Electromagnetic Compatibility - EMC EUROPE*, 2019, pp. 1138–1143.

[148] Y. Wang, S. Jere, S. Banerjee, L. Liu, S. Shetty, and S. Dayekh, "Anonymous Jamming Detection in 5G with Bayesian Network Model Based Inference Analysis," in *2022 IEEE 23rd International Conference on High Performance Switching and Routing (HPSR)*, 2022, pp. 151–156.

[149] C. Örnek and M. Kartal, "An Efficient EVM Based Jamming Detection in 5G Networks," in *2022 4th IEEE Middle East and North Africa COMMunications Conference (MENACOMM)*, 2022, pp. 130–135.

[150] Y. Arjoune and S. Faruque, "Smart Jamming Attacks in 5G New Radio: A Review," in *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*, 2020, pp. 1010–1015.

[151] Z. Dou, G. Si, Y. Lin, and M. Wang, "An Adaptive Resource Allocation Model With Anti-Jamming in IoT Network," *IEEE Access*, vol. 7, pp. 93 250–93 258, 2019.

[152] A. S. Ali, M. Baddeley, L. Bariah, M. A. Lopez, W. T. Lunardi, J.-P. Giacalone, and S. Muhaidat, "Performance Analysis and Evaluation of RF Jamming in IoT Networks," in *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, 2022, pp. 2745–2751.

[153] A. Al Sharah, H. Abu Owida, and T. A. Edwan, "Aggressive Jamming Attack in IoT Networks," in *2022 4th IEEE Middle East and North Africa COMMunications Conference (MENACOMM)*, 2022, pp. 235–239.

[154] M. S. Abdalzaher, M. Elwekeil, T. Wang, and S. Zhang, "A Deep Autoencoder Trust Model for Mitigating Jamming Attack in IoT Assisted by Cognitive Radio," *IEEE Systems Journal*, vol. 16, no. 3, pp. 3635–3645, 2022.

[155] A. Hussain, N. Abughanam, J. Qadir, and A. Mohamed, "Jamming Detection in IoT Wireless Networks: An Edge-AI Based Approach," in *Proceedings of the 12th International Conference on the Internet of Things*, ser. IoT '22. New York, NY,

USA: Association for Computing Machinery, 2023, p. 57–64. [Online]. Available: https://doi.org/10.1145/3567445.3567456

[156] M. Reynvoet, O. Gheibi, F. Quin, and D. Weyns, "Detecting and Mitigating Jamming Attacks in IoT Networks Using Self-Adaptation," in *2022 IEEE International Conference on Autonomic Computing and Self-Organizing Systems Companion (ACSOS-C)*, 2022, pp. 7–12.

[157] S. Yu, C. Lin, X. Zhang, and L. Guo, "Defending against Cross-Technology Jamming in Heterogeneous IoT Systems," in *2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS)*, 2022, pp. 702–712.

[158] C.-Y. Huang, C.-W. Lin, R.-G. Cheng, S. J. Yang, and S.-T. Sheu, "Experimental Evaluation of Jamming Threat in LoRaWAN," in *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*, 2019, pp. 1–6.

[159] I. Martinez, P. Tanguy, and F. Nouvel, "On the performance evaluation of LoRaWAN under Jamming," in *2019 12th IFIP Wireless and Mobile Networking Conference (WMNC)*, 2019, pp. 141–145.

[160] S. E. Viswapriya, S. Dinesh, and D. R. Teja, "IOT Security Vulnerabilities and Predictive Signal Jamming Attack Analysis in LoRaWAN," *International Journal of Computer Science and Mobile Computing*, 2021.

[161] F. Hessel, L. Almon, and M. Hollick, "LoRaWAN Security: An Evolvable Survey on Vulnerabilities, Attacks and Their Systematic Mitigation," *ACM Trans. Sen. Netw.*, vol. 18, no. 4, mar 2023. [Online]. Available: https://doi.org/10.1145/3561973

[162] N. Hou, X. Xia, and Y. Zheng, "Jamming of LoRa PHY and Countermeasure," in *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, 2021, pp. 1–10.

[163] S. J. Moon and W. Lee, "Friendly Jamming in LoRa Physical Layer Using Imperfect Orthogonality of Spreading Factor," in *2022 International Conference on Information Networking (ICOIN)*, 2022, pp. 423–428.

[164] T. Perković, H. Rudeš, S. Damjanović, and A. Nakić, "Low-Cost Implementation of Reactive Jammer on LoRaWAN Network," *Electronics*, vol. 10, no. 7, 2021. [Online]. Available: https://www.mdpi.com/2079-9292/10/7/864

[165] V. Kalokidou, M. Nair, and M. A. Beach, "LoRaWAN Performance Evaluation and Resilience under Jamming Attacks," in *2022 Sensor Signal Processing for Defence Conference (SSPD)*, 2022, pp. 1–5.

[166] C. Moy, "Artificial Intelligence for Jamming Mitigation in IoT Networks: LoRaWAN Field Measurements Using IoTligent," in *2021 XXXIVth General Assembly and Scientific Symposium of the International Union of Radio Science (URSI GASS)*, 2021, pp. 1–4.

[167] P. Stenumgaard, K. Fors, and K. Wiklundh, "Interference impact on LTE from radiated emission limits," in *2015 IEEE International Symposium on Electromagnetic Compatibility (EMC)*, 2015, pp. 165–170.

[168] A. El-Keyi, O. Uereten, T. Yensen, and H. Yanikomeroglu, "Lte physical-layer identity detection in the presence of jamming," in *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, 2017, pp. 1–6.

[169] K. Fors, K. Wiklundh, and S. Linder, "Interference Impact on Two of LTE's Control Channels," in *2018 International Symposium on Electromagnetic Compatibility (EMC EUROPE)*, 2018, pp. 309–314.

[170] N. K. Narcisse, L. Damon, and G. Bossard, "Impact of Jamming on Real Life LTE Networks: Experimental Results," in *2022 International Wireless Communications and Mobile Computing (IWCMC)*, 2022, pp. 248–253.

[171] F. M. Aziz, J. S. Shamma, and G. L. Stüber, "Jammer-Type Estimation in LTE With a Smart Jammer Repeated Game," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 8, pp. 7422–7431, 2017.

[172] X. Li and Z. Wang, "A Study on Pseudo CRS Signal Jamming Attacks in LTE Network," in *2018 IEEE Symposium on Computers and Communications (ISCC)*, 2018, pp. 1–4.

[173] J. Kakar, K. McDermott, V. Garg, M. Lichtman, V. Marojevic, and J. H. Reed, "Analysis and Mitigation of Interference to the LTE Physical Control Format Indicator Channel," in *2014 IEEE Military Communications Conference*, 2014, pp. 228–234.

[174] M. Lichtman, T. Czauski, S. Ha, P. David, and J. H. Reed, "Detection and Mitigation of Uplink Control Channel Jamming in LTE," in *2014 IEEE Military Communications Conference*, 2014, pp. 1187–1194.

[175] M. Lichtman, R. P. Jover, M. Labib, R. Rao, V. Marojevic, and J. H. Reed, "LTE/LTE-A jamming, spoofing, and sniffing: threat assessment and mitigation," *IEEE Communications Magazine*, vol. 54, no. 4, pp. 54–61, 2016.

[176] O. A. Topal, S. Gecgel, E. M. Eksioglu, and G. K. Kurt, "Identification of Smart Jammers: Learning based Approaches Using Wavelet Representation," 2019.

[177] M. Hachimi, G. Kaddoum, G. Gagnon, and P. Illy, "Multi-stage Jamming Attacks Detection using Deep Learning Combined with Kernelized Support Vector Machine in 5G Cloud Radio Access Networks," in *2020 International Symposium on Networks, Computers and Communications (ISNCC)*, 2020, pp. 1–5.

[178] Y.-P. E. Wang, X. Lin, A. Adhikary, A. Grovlen, Y. Sui, Y. Blankenship, J. Bergman, and H. S. Razaghi, "A Primer on 3GPP Narrowband Internet of Things," *IEEE Communications Magazine*, vol. 55, no. 3, pp. 117–123, 2017.

[179] R. Ratasuk, N. Mangalvedhe, J. Kaikkonen, and M. Robert, "Data Channel Design and Performance for LTE Narrowband IoT," in *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*, 2016, pp. 1–5.

[180] M. Kanj, V. Savaux, and M. Le Guen, "A Tutorial on NB-IoT Physical Layer Design," *IEEE Communications Surveys and Tutorials*, vol. 22, no. 4, pp. 2408–2446, 2020.

[181] J. Roux, E. Alata, G. Auriol, V. Nicomette, and M. Kaaniche, "Toward an Intrusion Detection Approach for IoT Based on Radio Communications Profiling," in *2017 13th European Dependable Computing Conference (EDCC)*, 2017, pp. 147–150.

[182] P.-F. Gimenez, J. Roux, E. Alata, G. Auriol, M. Kaaniche, and V. Nicomette, "RIDS: Radio Intrusion Detection and Diagnosis System for Wireless Communications in Smart Environment," *ACM Trans. Cyber-Phys. Syst.*, vol. 5, no. 3, apr 2021. [Online]. Available: https://doi.org/10.1145/3441458

[183] Y. Wan, K. Xu, G. Xue, and F. Wang, "IoTArgos: A Multi-Layer Security Monitoring System for Internet-of-Things in Smart Homes," in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, 2020, pp. 874–883.

[184] M. Achour, M. Mana, and A. Rachedi, "On the issues of selective jamming in IEEE 802.15.4-based wireless body area networks," *Peer-to-Peer Networking and Applications*, vol. 14, no. 1, pp. 135–150, 2021.

[185] Y. M. Amin and A. T. Abdel-Hamid, "Classification and analysis of IEEE 802.15.4 MAC layer attacks," in *2015 11th International Conference on Innovations in Information Technology (IIT)*, 2015, pp. 74–79.

[186] S. S. Jung, M. Valero, A. Bourgeois, and R. Beyah, "Attacking Beacon-Enabled 802.15.4 Networks," in *Security and Privacy in Communication Networks*, S. Jajodia and J. Zhou, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 253–271.

[187] S. M. Sajjad and M. Yousaf, "Security analysis of IEEE 802.15.4 MAC in the context of the Internet of Things (IoT)," in *2014 Conference on Information Assurance and Cyber Security (CIACS)*, 2014, pp. 9–14.

[188] H. Pirayesh, P. Kheirkhah Sangdeh, and H. Zeng, "Securing ZigBee Communications Against Constant Jamming Attack Using Neural Network," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4957–4968, 2021.

[189] D. Han, A. Li, L. Zhang, Y. Zhang, J. Li, T. Li, T. Zhu, and Y. Zhang, "Deep Learning-Guided Jamming for Cross-Technology Wireless Networks: Attack and Defense," *IEEE/ACM Transactions on Networking*, vol. 29, no. 5, pp. 1922–1932, 2021.

[190] R. Daidone, G. Dini, and M. Tiloca, "A solution to the GTS-based selective jamming attack on IEEE 802.15.4 networks)," in *2014 Conference on Information Assurance and Cyber Security (CIACS)*, 2014.

[191] A. D. Wood, J. A. Stankovic, and G. Zhou, "DEEJAM: Defeating Energy-Efficient Jamming in IEEE 802.15.4-based Wireless Networks," in *2007 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, 2007, pp. 60–69.

[192] G. D. O'Mahony, P. J. Harris, and C. C. Murphy, "Detecting Interference in Wireless Sensor Network Received Samples: A Machine Learning Approach," in *2020 IEEE 6th World Forum on Internet of Things (WF-IoT)*, 2020, pp. 1–6.

[193] J. Roux, E. Alata, G. Auriol, M. Kaaniche, V. Nicomette, and R. Cayre, "RadIoT: Radio Communications Intrusion Detection for IoT - A Protocol Independent Approach," in *2018 IEEE 17th International Symposium on Network Computing and Applications (NCA)*, 2018, pp. 1–8.

[194] A. Sørensen, H. Wang, M. J. Remy, N. Kjettrup, R. B. Sørensen, J. J. Nielsen, P. Popovski, and G. C. Madueño, "Modeling and Experimental Validation for

Battery Lifetime Estimation in NB-IoT and LTE-M," *IEEE Internet of Things Journal*, vol. 9, no. 12, pp. 9804–9819, 2022.

[195] R. Sokullu, I. Korkmaz, O. Dagdeviren, A. Mitseva, and N. R. Prasad, "An investigation on IEEE 802.15. 4 MAC layer attacks," in *Proc. of WPMC*, vol. 41, 2007, pp. 42–92.

# Acronyms

**3GPP** Third Generation Partnership Project

**AB** Access Barring

**AUC** Area Under the Curve

**AWGN** Additive White Gaussian Noise

**C-RNTI** Cell Radio Network Temporary Identifier

**CAP** Contention Access Period

**CCA** Clear Channel Assessment

**CFO** Carrier Frequency Offset

**CFP** Contention-Free Period

**CSB** Code Sub-blocks

**CSMA/CA** Carrier Sense Multiple Access with Collision Avoidance

**CSS** Chip Spread Spectrum

**CoAP** Constrained Application Protocol

**DCNN** Deep Convolutional Network

**DMRS** Demodulation Reference Signal

**DoS** Denial of Service

**EMI** Electromagnetic Interference

**EVM** Error Vector Magnitude

**FDD** Frequency Division Duplex

**FEC** Forward Error Correction

**FFD** Full-Function Device

**FN** False Negatives

**FP** False Positives

**GSM** Global System for Mobile communication

**GTS** Guarantee Time Slot

**H-SFN** Hyper System Frame Number

**IDS** Intrusion Detection System

**IPSec** Internet Protocol Security

**IoT** Internet of Things

**LPWAN** Low Power Wide Area Network

**LR-WPAN** Low-Rate Wireless Personal Area Network

**LSB** least significant bits

**LTE-M** Long-Term Evolution for Machine-Type Communications

**LTE** Long-Term Evolution

**LoRa** Long Range

**MAC** Medium Access Control

**MIB** Master Information Block

**ML** Machine Learning

**MPDU** MAC Protocol Data Unit

**MQTT** Message Queuing Telemetry Transport

**MSB** Most Significant Bits

**NB-IoT** Narrowband-Internet of Things

**NB-PCID** Narrowband Physical Cell ID

**NPBCH** Narrowband Broadcast Channel

**NPDCCH** Narrowband Physical Downlink Control Channel

**NPDSCH** Narrowband Physical Downlink Shared Channel

**NPRACH** Narrowband Physical Random Access Channel

**NPSS** Narrowband Primary Synchronisation Signal

**NPUSCH** Narrowband Physical Uplink Shared Channel

**NRS** Narrowband Reference Signal

**NSSS** Narrowband Secondary Synchronisation Signal

**OFDM** Orthogonal Frequency Division Multiplexing

**PAN** Personal Area Network

**PCID** Physical Cell ID

**PDCCH** Physical Downlink Control Channel

**PDU** Protocol Data Unit

**PLMN** Public Land Mobile Network

**PRB** Physical Resource Blocks

**PRNG** Pseudorandom Number Generators

**PSM** Power Saving Mode

**PUF** Physical Unclonable Functions

**RE** Resource Elements

**RFD** Reduced-Function Device

**RFI** Radio Frequency Interference

**RMC** Reference Measurement Channel

**RMC** Reference Measurement Channel

**RNG** Random Number Generators

**ROC** Receiver Operating Characteristic

**SC-FDMA** Single-Carrier Frequency Division Multiple Access

**SFN** System Frame Number

**SIB1-NB** Narrowband System Information Block Type 1

**SIB** System Information Block

**SI** System Information

**SNR** Signal-to-noise Ratio

**SSL** Secure Sockets Layer

**SVM** Support Vector Machine

**TBCC** Tail-Biting Convolutional Code

**TBS** Transport Block Size

**TDD** Time Division Duplex

**TLS** Transport Layer Security

**TN** True Negatives

**TP** True Positives

**TRNG** True Random Number Generators

**TTI** Transmission Time Interval

**UE** User Equipment

**eDRX** extended Discontinuous Reception

**eNodeB** Evolved Node B

**mMTC** Massive Machine-Type Communication