Attack Modelling and Detection in Distributed and Cooperative Controlled Microgrid Systems

Mingxiao Ma

UNIVERSITÉ DE LORRAINE

École doctorale IAEM Lorraine

Attack Modelling and Detection in Distributed and Cooperative Controlled Microgrid Systems

THÈSE

présentée et soutenue publiquement le 22 Avril 2021

pour l'obtention du

Doctorat de l'Université de Lorraine

(mention informatique)

par

Mingxiao MA

Composition du jury

Rapporteurs :	Mohamed Kaâniche	Directeur de recherche CNRS au LAAS, France
	Stéphane Mocanu	Maître de conférences à Université de Grenoble-Alpes, France
Examinateurs :	Ghita Mezzour	Maître de conférences à Université Internationale de Rabat, Maroc
	Abdelmadjid Bouabdallah	Professeur à Université de Technologie de Compiègne, France
	Ye-Qiong Song	Professeur à Université de Lorraine, France
Encadrants :	Isabelle Chrisment	Professeure à Télécom Nancy, Nancy , France
	Abdelkader Lahmadi	Maître de conférences à Université de Lorraine, France

Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503

Mis en page avec la classe thesul.

Acknowledgement

Research is indeed a social activity involved with discussions and exchanging ideas. This thought originates from the wonderful interactions that I want to acknowledge with the many contributors to this thesis.

As a PhD student, the initial research inspiration comes from supervisors. I am thrilled to have both Prof. Isabelle Chrisment and Dr. Abdelkader Lahmadi taking that role. Throughout my academic research journey, their guidance and professional insights have always been inspiring me.

Besides my supervisors, I would like to thank Dr. Mohamed Kaâniche and Dr. Stéphane Mocanu for being the reviewers of my thesis. Moreover, I want to thank the rest of my jury committee members: Prof. Ye-Qiong Song, Prof. Abdelmadjid Bouabdallah, and Dr. Ghita Mezzour. Likewise, I am grateful to Mme Souad Boutaguermouchet and Mme Sylvie Musilli who helped me with all the administrative tasks.

The sharing and discussion of ideas is an important aspect of research. In this sense, I am really appreciative of the wonderful visiting times spent both abroad and in France. I am thankful to Prof. Babak Nahid-Mobarakeh, Prof. Peng-Yong Kong, Dr. Lei Mo, and Dr. Qipeng Song for the great discussions we had in different academic occasions.

I would like to express my gratitude to all of my colleagues and co-authors for the excellent talks and discussions we had together. I owe a debt of gratitude to all of the promising ideas we have encountered and fought with... the ones that worked and the ones that did not, they have all been a valuable source of inspiration and knowledge. A special thanks to Pierre Marie Junges for all the fantastic ideas we exchanged and discussed, and to Thomas Lacour for all the kind help in the implementation of experimental testbeds.

I am very grateful for the wonderful atmosphere at LORIA. It is incredible how many interesting conversations and partnerships began during coffee breaks or on the way to lunch. Thank you to Prof. Ye-Qiong Song, who is also a Chinese working in LORIA, for all the kind help that adapts myself faster to the life in France and make me less homesick.

A special thanks to my former advisor and colleague in the Netherlands, Dr. André Teixeira, for helping me build my theoretical foundation of cyber-security research.

The work leading to this thesis has been funded by the French Government under grant FUI 23 PACLIDO (Protocoles et Algorithmes Cryptographiques Légers pour l'Internet Des Objets). All the related supports are greatly acknowledged.

Without the help of my family, none of this would have been possible. Special thanks to my parents Zhanping Ma and Xingying Liu and my brother Mingxu for their support and encouragement throughout my life. My dearest love to my wife, Liufang Chen, for her unflinching support and unselfish devotion.

Mingxiao Ma

Contents

List of Figures

Abstract

Modern low-voltage microgrid systems rely on distributed and cooperative control approaches to guarantee safe and reliable operational decisions of their inverter-based distributed generators (DGs). However, many sophisticated cyber-attacks can target these systems, deceive their traditional detection methods and cause a severe impact on the power infrastructure.

In this thesis, we systematically study the vulnerabilities and threats of distributed controlled microgrid systems. We design a novel attack named "measurement-as-reference" (MaR) attack and take it as a typical stealthy attack example to theoretically analyze the attack impact on the microgrid system and use numerical simulation results to verify the analysis. We provide mathematical models of possible false data injection (FDI) and denial of service (DoS) attacks in a representative distributed and cooperative controlled microgrid system. We propose a secure control framework with an attack detection module based on machine learning techniques. To validate the effectiveness of this framework, we implement two typical attacks, MaR attack and delay injection attack, on a hardware platform modeled after a microgrid system. We collect datasets from the platform and validate the performance of multiple categories of machine learning algorithms to detect such attacks. Our results show that tree-based classifiers (Decision Tree, Random Forest and AdaBoost) outperform other algorithms and achieve excellent performance in detecting normal behavior, delay injection and false data attacks.

Résumé

Les micro-réseaux électriques s'appuient sur des approches de contrôle distribuées et coopératives pour garantir des décisions opérationnelles sûres et fiables de leurs générateurs distribués (DG). Cependant, de nombreuses cyber-attaques sophistiquées peuvent viser ces systèmes, tromper leurs méthodes de détection traditionnelles et avoir des conséquences importantes sur l'infrastructure électrique.

Dans cette thèse, nous étudions les attaques ciblant les systèmes de contrôle associés à ces mircro-réseaux. Nous avons développé dans un premier temps une nouvelle attaque nommée *MaR* (Measurement as Reference) qui cible les consignes de synchronisation échangées entre les entités du système de contrôle. Nous avons analysé par simulation numérique l'impact de cette attaque sur la stabilité du micro-réseau et la convergence du système de contrôle vers une consigne commune. Nous avons également développé des modèles d'analyse des attaques de type injection de fausses données et déni de service sur ces systèmes pour étudier leurs impacts et leur détection. Ensuite, nous avons proposé un *framework* qui permet de détecter ces attaques, en se basant sur l'apprentissage automatique des caractéristiques des paquets réseau échangés entres les entités d'un système de contrôle distribué. Nous avons mis en oeuvre une plate-forme expérimentale représentative d'un micro-réseau électrique et son système de contrôle pour collecter des jeux de données et valider notre *framework*, en particulier son module de détection des attaques. Enfin, nous avons évalué les performances de différents algorithmes d'apprentissage automatique pour détecter les attaques que nous avons introduites sur la plate-forme expérimentale. Nos résultats montrent que les algorithmes basés sur les techniques d'arbres, à l'image des arbres de décision, les forêts aléatoire et AdaBoost offrent les meilleurs performance en termes de précision et de justesse pour détecter les différentes attaques et les distinguer.

Chapter 1

General Introduction

Contents

1.1 Cyber-Physical Security of Microgrids

New information and communication technologies (ICT) and various distributed generations (DGs), e.g., photovoltaic (PV) and wind turbine generators, are integrated into the power grid to build smart grids. They are essentially defined according to IEEE 2030-2011 standard, as a composition of three interoperability infrastructures [10], as outlined in Figure 1.1. This suggested interdependence has led to the problem of security becoming increasingly complex and impose additional challenges which could easily lead to damage to the fundamental security concerns as well as create new ones.

A new power generation paradigm where a mixed generation pool consists of conventional centralized plants and distributed generation units at lower voltage levels is emerging. The concept of microgrids is developed to tackle these challenges by expediting a local integration of renewable energy sources. Microgrids, composed of several DGs, energy storage elements, and controllable loads, are autonomous, and controllable small-scale power systems can operate in grid-connected and islanded modes [51, 63].

Microgrids utilize a hierarchical control structure, including primary, secondary, and tertiary control levels [51, 7]. The primary control conventionally utilizes local droop controllers on distributed energy resources (DERs) to maintain microgrid voltage stability after the islanding. The secondary control of microgrid, which regulates the voltage to synchronize the average voltage of microgrid to the nominal setpoint, can be implemented through a centralized structure. In this structure, the microgrid secondary control center is responsible for determining and sending the voltage and power setpoints to the individual DERs through a complex two-way, high-bandwidth communication network [51]. However, this control structure is vulnerable to the single-point-of-failure in the control center, which could reduce system reliability. Alternatively, distributed cooperative control of microgrid with a sparse communication network [51, 7, 4]

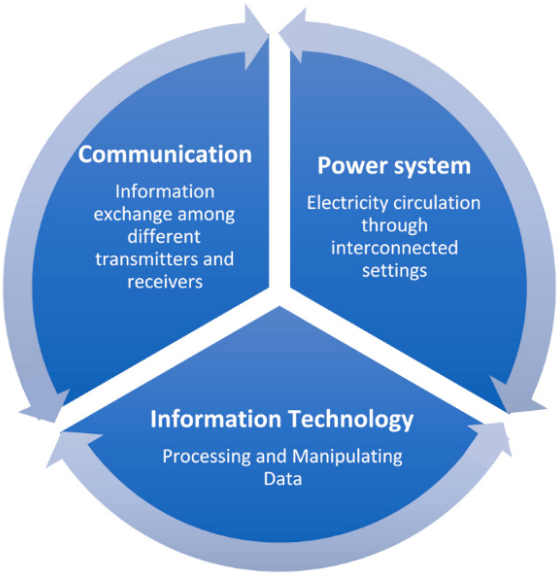

Figure 1.1: Smart grid architecture as a composition of three interoperability infrastructures in compliance with IEEE 2030 [10].

has been proposed recently to increase the reliability, scalability, and resilience of the microgrid control system.

These microgrids are evolving into cyber-physical systems (CPSs) with advanced software-based control and communication networks [4]. These control techniques map DERs to nodes on a sparse communication digraph and ensure that all nodes reach an agreement on quantities of interests issued by the leader node (known as synchronization) by adopting the locally relevant information exchanged among neighboring DERs.

Such microgrids with a distributed nature are vulnerable to malicious cyber-attacks. There is no globally centralized entity to monitor activities of all DERs, and each DER only has access to its local data and partial data from neighboring nodes, resulting in a limited global situational awareness of cyber-attacks [4]. Thus, it becomes more challenging to operate the power networks in a reliable and resilient mode in this new paradigm. To the best of our knowledge, the first known hacker-caused outage happened in 23 December 2015, hackers compromised information systems of three energy distribution companies in Ukraine and temporarily disrupted the electricity supply to consumers.

The awareness and concern over the security of CPS like microgrids have been increased. Modern SCADA systems have moved towards using standard communication technologies to enable access to remote devices and facilitate a smooth interface between devices from different vendors. Consequently, the number of possible attack points for malicious cyber agents to exploit has dramatically increased. Another common practice is the use of standard hardware and software platforms to decrease costs and improve flexibility. New vulnerabilities of these standard platforms may be discovered over their life-cycle, which significantly increases the risk of cyber threats to a large number of SCADA systems. In fact, the number of reported CPS-related security incidents has significantly increased over recent years. As shown in Figure 1.2, total vulnerability instances amount of each CPS is mapped to the years from 2000 till 2019, considering common vulnerabilities and exposures (CVE) vulnerabilities since 1999 till now [32]. Compared to master terminal units (MTUs) and remote terminal unit devices (RTUs), more

vulnerabilities of programmable logic controllers (PLCs) and HMI/SCADA have been disclosed, especially in the past 6 years [32].

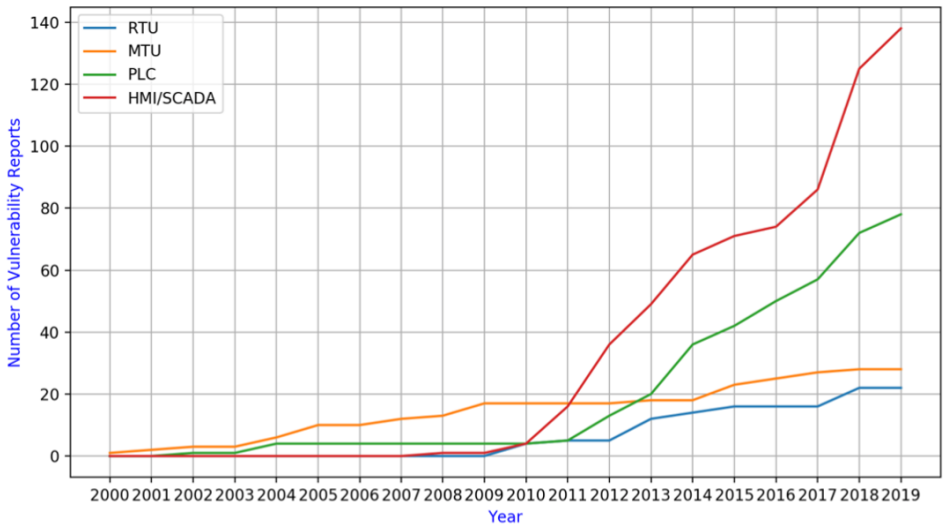

Figure 1.2: CPS vulnerability totals trend by year [32].

Analogous to the situation in CPSs susceptible to various types of cyber-attacks, attacks on microgrid systems can be roughly grouped into two classes based on attack goals:: false data-injection (FDI) attacks and denial of service (DoS) attacks. FDI attacks, which is also named deception attacks in some literature [14], target the sensors and controllers and even the communication links, which in turn corrupt the transferred data and compromise the microgrid data *integrity*. DoS attacks prevent the controllers from receiving sensor measurements or the actuators from receiving control commands and thus compromise the *availability* of microgrid system services. In this thesis, both FDI and DoS attacks are of concern.

The cyber-attacks in microgrid control systems have been investigated in [63, 41], where concrete DoS and FDI attack scenarios and impact analysis have been studied, but no attack detection and mitigation schemes are provided. To detect corrupted data in the Supervisory Control and Data Acquisition (SCADA) system of power grid, state vector estimation (SVE) is extensively used [39, 35]. The detector obtains the system state, which is estimated from the observed measurements and then computes the residual between the estimated and the observed measurements. If the residual is greater than the given threshold, a data injection attack is detected. However, the exact recovery of state vectors is a challenge for SVE-based methods in sparse networks [12, 35], where the Jacobian measurement matrix is sparse, and the performance of this approach is restricted by the sparsity of the state vectors. Besides, if the false data injected vectors reside in the column space of the Jacobian measurement matrix and satisfy some sparsity conditions, then the false data injection attacks, called unobservable attacks, cannot be detected [39, 35]. Additionally, detection methods broadly applying state estimation processes, e.g., using Kalman filter [43], sparse optimization [38], graphical methods [6] to address FDI attack detection all require centralized communication structures. The FDI attack detection in distributed control systems deploying Kullback-Leibler (KL) divergence is addressed in [51]. However, the above attack detection techniques only deal with a single type of cyber attack (FDI). A distributed state estimation method based on the alternating direction method of multipliers (ADMMs) is proposed to detect hybrid cyber-attacks (FDI and DoS attacks) in [16]. However, this method mainly benefits the large-scale power transmission systems due to its

heavy computation complexity.

The best practices and techniques from security are a sound first approach to increase the security and resilience of CPS. However, traditional IT security does not consider the interdependencies between the physical components and the cyber domain. A holistic approach is required to effectively handle the complex coupling between the physical process and the IT infrastructure.

1.2 Research Questions

This thesis addresses the problem of cybersecurity and resilience in the distributed control system of microgrids. Traditional cybersecurity does not consider the interdependencies between the physical components and the cyber systems. On the other hand, control theoretic approaches typically deal with independent disturbances and faults. Thus they are not tailored to handle cyber threats. Theory and tools to analyze and build control system resilience are, therefore, lacking and in need to be developed.

The recently coined cyber-physical system term integrates a physical infrastructure and a cyber framework to reduce the complexity and costs of traditional control systems in, e.g., industrial environments. Microgrid control systems are composed of sensors, actuators, and other field devices that interact with the physical processes. The technology evolution brings these systems towards a combination between a physical layer that encompasses the physical framework; and a cyber layer that encompasses the communication and computation framework. Figure 1.3 shows schematic and risk assessment concepts of a typical cyber-physical control system with a communication network that is vulnerable to adversaries.

The thesis focuses on the following groups of questions related to the distributed control system of microgrids:

- **Reference Architecture.** What core components should be considered in a reference architecture for cyber-secure and resilient microgrid control systems? What components could be vulnerable to cyber attacks?

- **Modeling Framework.** How can a malicious adversary be modeled from a control-theoretic perspective? What is the right level of modeling detail? What assumptions should be made to model the adversary considering system properties of microgrid control systems?

- **Impact Assessment.** What metrics could be used to assess and compare threats? What is the worst-case impact the threat could cause, and how to quantitatively measure the attack impacts? How can the metrics be used to prioritize attack scenarios and assess the effectiveness of defensive actions?

- **Attack Detection and Response.** What tools could be employed to detect the threats as effectively as possible? What tools and actions can be devised to increase the cybersecurity and resilience of microgrid control systems? Can such methods be implemented and used in real-life microgrids?

These questions are tackled throughout this thesis, contributing towards a framework to analyze, identify, and evaluate the consequences of vulnerabilities in microgrid control systems, as well as to propose and devise effective protection schemes.

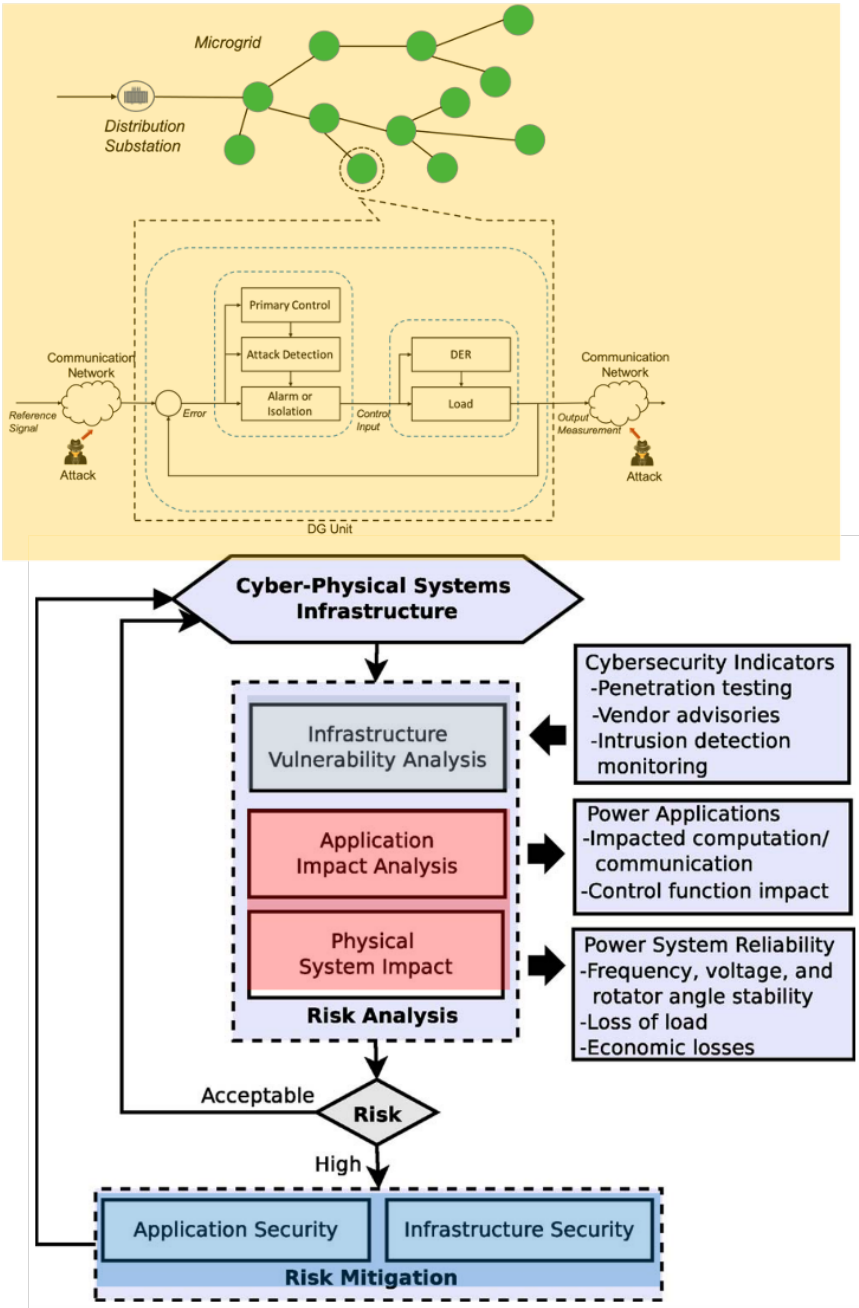

Figure 1.3: Schematic and impact assessment concepts of a typical cyber-physical control system with a communication network that is vulnerable to adversaries.

1.3 Methodology and Contributions

In this thesis, we address security issues in microgrid control systems, which are typical cyber-physical systems. We focus on the protection between the cyber and physical layers of these systems by designing a secure framework, including an attack detection mechanism.

Threat modeling for complex systems is a four-stage process that begins with defining and diagramming the system, then identifying things that can go wrong, after which mitigation

strategies can be developed. The entire process is validated to ensure accurate conclusions. In this thesis, we employ a methodology described in the Flowchart 1.4 to study the threat modeling and risk management problem of microgrid systems.

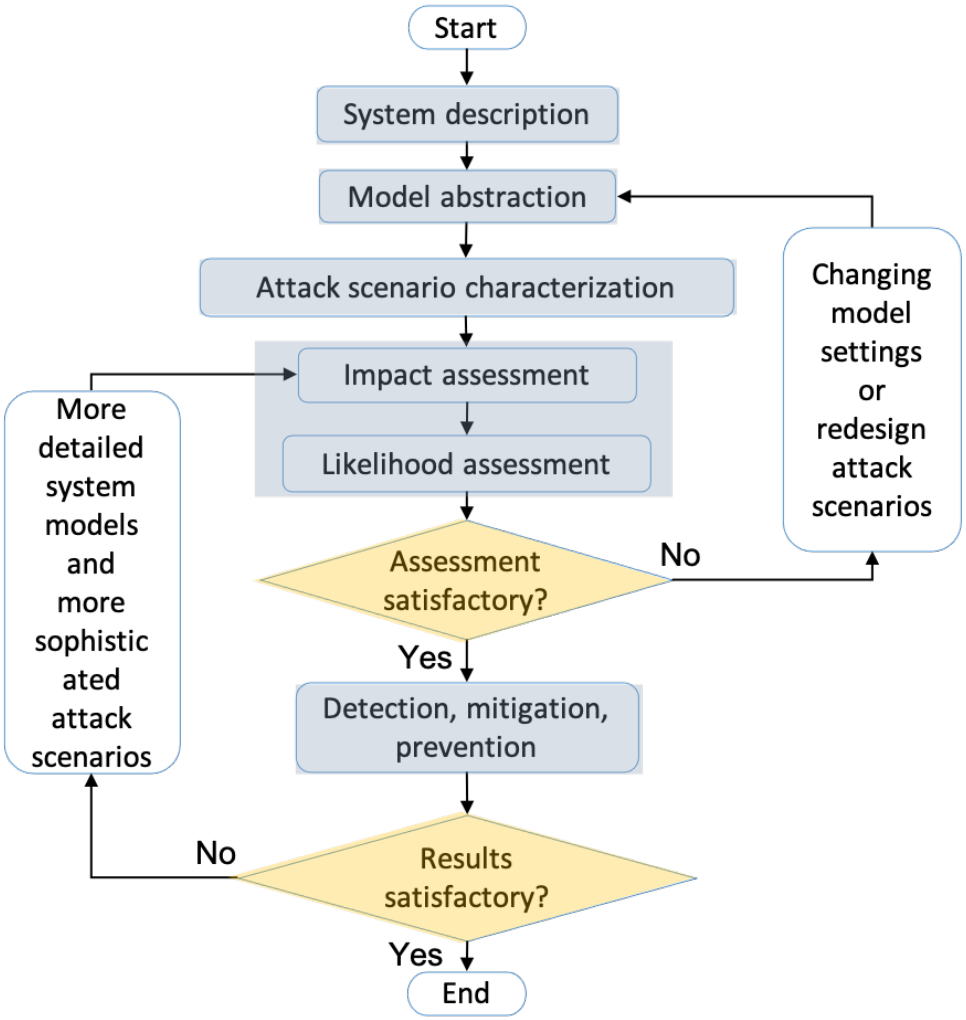

Figure 1.4: Methodology flowchart to study the threat modeling and risk management problem of microgrid control systems

We start with the security analysis of previous theoretical detection mechanisms work. We consider a more comprehensive control structure for a microgrid system, mainly a hierarchical control approach consisting of three control levels: primary, secondary and tertiary control [7]. Specifically, we design a distributed cooperative hierarchical control system consisting of droop control as primary control and cooperative control as secondary control. We believe this control scheme is more realistic and reliable than the control architecture considered in [63] and [40], because of its better support for scalable integration of local loads and high integration of distributed generation.

Based on the above control model, we consider cyber attacks that may target the communication links between DGs, and we use risk assessment methods to quantitively analyze the impact of these attacks on voltage deviation at the primary control level and reference voltage synchronization at the secondary control level. Our impact analysis results are valuable

for microgrid system designers to help them evaluate potential cyber threats targeting these systems.

We propose a resilient distributed cooperative control for microgrids that accommodates FDI and DoS attack detection. Compared to the existing attack detection methods based on state estimation [39, 35], statistical machine learning algorithms outperform them in detecting both observable and unobservable attacks [50]. The attack detection problem can be modeled as a typical classification problem to learn communication patterns at the network-level of the distributed control system and predict the class of a packet as attack or normal.

To validate the performance of machine learning and deep learning based attack detection, we give a detailed hyper-parameter tuning methodology to get the best hyper-parameters used for classification. We collect datasets from the platform and compare the performances of different machine learning and deep learning algorithms to detect measurement-as-reference (MaR) and delay injection attacks. We give a detailed analysis for the three classification problems on balanced and imbalanced dataset, respectively.

1.3.1 Thesis Outline

The thesis outline is as follows.

1) In Chapter 2, we review some of the main concepts of the microgrid systems, including the control structure and the communication networks used throughout the thesis. We present a brief summary of recent developments in the area of microgrid control systems, which are described and modeled in more detail and the cyber-physical security problems are reviewed, and various cyber-physical attacks grouped in different categories are summarized. At last, we give a comprehensive summary of the attack detection schemes of recent related work on cyber-secure control systems. The Chapter concludes with a brief description of the main applications and experimental setups considered in the thesis.

2) In Chapter 3, we consider a more comprehensive control structure for a microgrid system, mainly a hierarchical control approach consisting of three control levels: primary, secondary and tertiary control [7]. Specifically, we design a distributed cooperative hierarchical control system consisting of droop control as primary control and cooperative control as secondary control. We believe this control scheme is more realistic and reliable than the control architecture considered in [63] and [40], because of its better support for scalable integration of local loads and high integration of distributed generation. Based on the above control model, we consider cyber attacks that may target the communication links between DGs, and we use risk assessment methods to quantitively analyze the impact of these attacks on voltage deviation at the primary control level and reference voltage synchronization at the secondary control level. Our impact analysis results are valuable for microgrid system designers to help them evaluate potential cyber threats targeting these systems.

3) In Chapter 4, we model and generalize both FDI and DoS attacks in distributed and cooperative control microgrid systems. We also give a general system stability condition analysis under FDI and DoS attacks. We propose a secure control framework with attack detection and response modules relying on machine learning and deep learning algorithms. Various groups of supervised machine learning and deep learning algorithms are studied to design the attack detection module against FDI attacks and DoS attacks under well-chosen performance metrics. Considering it is difficult to find publicly available datasets, we build an

experimental platform emulating the distributed and cooperative controlled microgrid systems, and implement two typical FDI and DoS attacks hard to be detected by traditional residual comparison detectors. The two attacks are: measurement as reference (MaR) attack as typical FDI attack and delay injection attack as typical DoS attack. Balanced and imbalanced datasets are collected to under normal and attacked conditions to be further analyzed.

4) in Chapter 5, we validate the inefficiency of traditional threshold comparison method in detecting MaR and delay injection attacks. We describe how we tune the machine learning and deep learning hyper-parameters and list the best parameters of each classifier that we use for classification. At last, we show the experimental results and performance analysis for collected imbalanced and balanced dataset, respectively.

5) in Chapter 6, we give a general conclusion about this thesis and talk about the future work.

1.3.2 Publications

Early versions of the results covered in this thesis have been successfully reported in the following peer-reviewed publications.

1) Ma, Mingxiao, and Abdelkader Lahmadi. "On the Impact of Synchronization Attacks on Distributed and Cooperative Control in Microgrid Systems." 2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm). https://hal.inria.fr/hal-02980115v1

2) Ma, Mingxiao, Abdelkader Lahmadi, and Isabelle Chrisment. "Demonstration of Synchronization Attacks on Distributed and Cooperative Control in Microgrids." 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM). https://hal.inria.fr/hal-02389307v1

3) Ma, Mingxiao, Abdelkader Lahmadi, and Isabelle Chrisment. "Detecting a Stealthy Attack in Distributed Control for Microgrids using Machine Learning Algorithms." 2020 IEEE Conference on Industrial Cyberphysical Systems (ICPS). Vol. 1. https://hal.inria.fr/hal-02980115v1

Chapter 2

Related Work

Contents

2.1 Introduction

As a controlled small-scale power system, a microgrid can be operated in both islanded and grid-connected mode in a specified area in order to facilitate the provision of supplementary power and maintain a standard service [18, 28]. It usually consists of small-scale distributed generators, energy storage elements, local loads, and controllers. Because of their advantages in reliability, flexibility, scalability and efficiency, microgrids are becoming increasingly popular in power systems. Microgrids can use alternate current (AC) or direct current (DC) energy or even AC/DC hybrid topologies, in which AC and DC power supplies and loads can both be considered [18].

Resilience is the potential to sustain sufficient operating levels in the face of unusual conditions [64]. The microgrid infrastructure must have resiliency in relation to disturbances and faults, by developing stable and secure frameworks as a typical cyber-physical system. Not only

should such a framework be fault-tolerant, but it should also secure the system from malicious adversaries. While the security of IT systems is commonly known in the literature, the same is uncertain in the case of microgrid control systems.

In this Chapter, we review some of the main concepts of the microgrid systems, including the control structure and the communication networks used throughout the thesis. Firstly, we present a brief summary of recent developments in the area of microgrid control systems, which are described and modeled in more detail in Chapter 3. Secondly, the cyber-physical security problems are reviewed, and various cyber-physical attacks grouped in different categories are summarized. At last, we give a comprehensive summary of the attack detection schemes of recent related work on cyber-secure control systems. This Chapter concludes with a brief description of the main applications and experimental setups considered in the thesis.

2.2 Microgrid Control Systems

2.2.1 Overview

Microgrids are designed to run in both grid-connected and islanded working modes, depending on whether or not they are connected to the main grid. The microgrid control system is responsible for regulating voltage and frequency in various operating modes and ensuring a proper load sharing among DGs, controlling the power flow between the main grid and the microgrid and minimizing the operations costs.

In current power networks, the security and reliability of operations are known as crucial problems. Modern information and communication technology (ICT) and distributed generation technologies, including photovoltaic (PV) and wind turbines, are introduced into the power grid. Thus there is a modern trend of power production, in which a hybrid generation pool is used, including of traditional clustered plants and distributed generator (DG) units at lower voltage levels. In this new paradigm, though, it becomes more challenging to run the power grids in a stable and robust manner. To overcome these problems, the idea of microgrid is built by the rapid local deployment of renewable sources. Autonomous, small-scale controllable power systems may operate in grid-connected and islanded modes, consisting of a variety of DGs, energy stock elements and controllable loads [51, 63].

Three critical roles are typically implemented through microgrid control systems: (i) control of distributed generators, often known as distributed energy resources (DER), (ii) energy management, and (iii) microgrid protection. Although microgrids can be run at medium voltage levels, there is no doubt that the microgrid is most widely used in low-voltage delivery systems. In Figure 2.1 the hierarchical control structure is typically used to control and handle the AC-microgrid, DC-microgrid or even a hybrid-microgrid.

In a hierarchical framework, the control mechanism is split down into three major layers: central, secondary and tertiary control loops (see Figure 2.1). The droop control system is generally used for the primary control loop, simulating a synchronous machine's actions. The secondary control loop aims at repairing the secondary (e.g., voltage and frequency) variables to their nominal values. Finally, the energy management system (EMS) is commonly used for maximum operation and congestion management in a micro-microgrid in the third-level control loop. In order to control current and voltage in the input/output of a power converter mounted on the microgrid, an internal (inner) control loop is typically used at the lower stage. The above internal control is generally named zero control level. They normally use resonant controllers, controllers implemented in the synchronous rotating d-q framework, predictive controllers for realization.

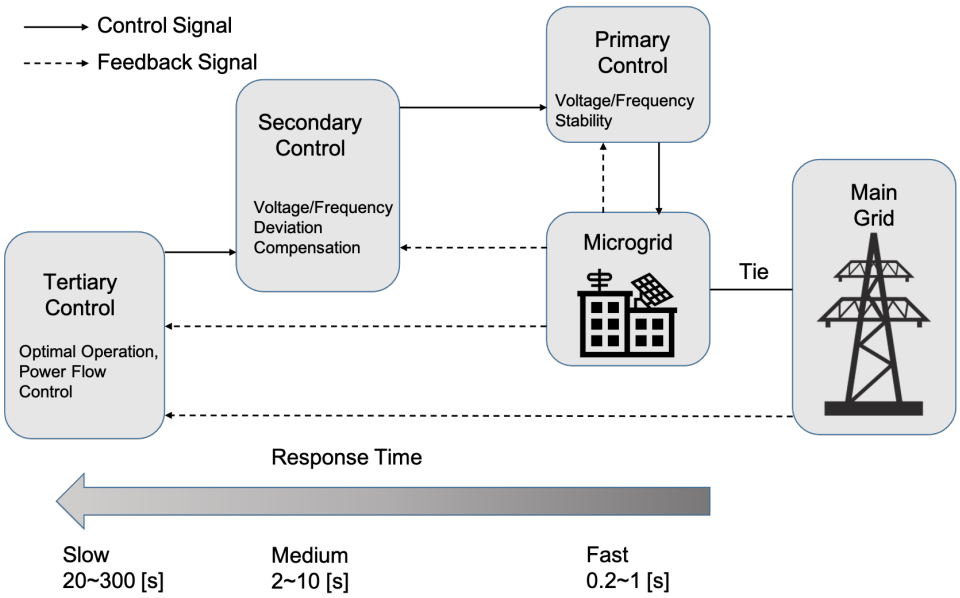

Figure 2.1: The hierarchical control structure of a microgrid consisting of primary, secondary and tertiary control levels as specified in [7].

As for each control layer's implementation, three application methodologies may be implemented for each of the microgrid topologies discussed in this thesis. These topologies involve centralized, distributed and decentralized designs. Below [18] is a short overview of each implementation methodology:

1) Centralized Control. In this case, the microgrid requires a central controller to interact with the DERs in the microgrid. All information transmitted from the other components in the microgrid must be processable by the central controller. Centralized controllers are not very robust and will be addressed further in the following section.

2) Distributed Control. In this scenario, centralized control is not needed since the control effort is distributed and unified among all the microgrid modules. The autonomous agents cooperatively attempt to obtain the global goals. Distributed control systems increase the microgrid's scalability and provided that fault-tolerant mechanisms are implemented.

3) Decentralized Control. In this situation, only local measurements are used for the control scheme of a DER unit (agent). The methodologies for control are generally based on "voltage-reactive power" ($V-Q$) and "frequency-active power" ($f-P$) droops, which subtract proportional parts of the output average reactive and active powers to the voltage amplitude and frequency of each DG unit. The microgrid load between the distributed generation (DG) units is exchanged through a physical connection by using droop controllers, according to their power capabilities. Notice that there are no communication networks in this approach, and this undoubtedly makes it impossible to implement secondary and tertiary control loops. However, in the literature such methods have been suggested, such as the use of high-pass "wash-out" filters [18].

Generally, secondary and tertiary control layers of the microgrid can be implemented utilizing centralized topologies, and it is cost-efficient to do so given that they have to measure all DERs

throughout the microgrid to reach the control target. However, these control layers can be applied with greater efficiency and protection by the new advances in distributed control theory. In addition, the distributed architecture of communication is sparse and dependent on local controls, which provides an essential factor in DER control and collaboration during operations of the microgrid in order to achieve global objectives. The critical value of distributed control systems can be outlined as follows for microgrid applications:

1) Robustness. If a single feature ceases operating well or if there is any fault, none of the overall functions of the microgrid can be affected. On the other hand, the centralized control center appears to be vulnerable to common errors and have significant teamwork concerns.

2) Scalability. This is a control method that is flexible to use. Changes to the microgrid, such as introducing more DERs, battery energy system (BESS), and more loads, are much simpler to incorporate, without disrupting the operation of other microgrid components.

3) Flexibility. It enables the DERs to operate plug-and-play functions. This is an enticing microgrid feature.

In the mid of 2000s, the implementation of microgrids coupling with distributed control algorithms was initially suggested [13], [52]. The potential implementation of agent control for the energy market was addressed in [13]. In [52], the authors designed a strategy that integrates the activities of a centralized controller and a distributed local controller. Since then, discussions have been conducted about what kind of control scheme (centralized or distributed) is more effective for microgrid applications. A detailed comparative description of characteristics of the centralized and distributed control topologies is given in [18]. We further conclude that the distributed control topology has the following advantages over the centralized control topology:

1. **Lower communication needs.** Microgrids with the centralized control scheme continuously monitor a multitude of involved DERs that are vulnerable to communication delays or packet losses. Local DER controllers in the distributed control system share data only with neighbors that are resilient in response to communication delays or packet losses and relax strict communication requirements.

2. **Lower computational burden.** Microgrids in a centralized control scheme process data from a variety of contributing distributed energy resources, which results in substantial computational burdens on microgrids. Parallel data from neighbors will be handled by local DER controllers which engage in the distributed control system based on consensus protocols with a substantial reduction of the computational burdens.

3. **Better scalability.** The system scale of the centralized control system is constrained by the communicational and computational capacities of the microgrid, the communication topology, and the usable bandwidth. DERs involved in the distributed control framework only communicate through the sparse communication networks with their neighbors, with plug-and-play characteristics and increased scalability for possible modifications and expansions.

4. **Better reliability.** When a communication connection fails in the system, the centralized control structure is highly vulnerable and unable to operate correctly. The whole power network may be shut down by the maintenance and improvement of the centralized control system. Thanks to the redundancy of the distributed control system, it can withstand a communication link failure due to the presence of alternate routes. The distributed

control system could be maintained and upgraded without the whole system being shut down.

5. **Better resilience.** The centralized control device implemented in the star communication topology has minimal controller redundancy and is prone to single-point failures, which can cause the system to crash. The distributed control system implemented in the meshed communication network has improved controller redundancy to maintain robustness against single-point failures.

Provided that different fault tolerance mechanisms are implemented, it can be concluded that distributed techniques have the following benefits compared with centralized methods: they improve robustness and reliability, allow for flexibility and scalability, including operations with plug-and-play functions.

2.2.2 Hierarchical Control Layers of Microgrid Systems

For the hierarchy control structure shown in Figure 2.1, each control level has its own control goals because of operating in different timescales. The primary control operates on a fast timescale, and is responsible for the control of transients to stabilize the voltage and frequency of the microgrid during changing of load or generation, or subsequent to an islanding event. Secondary control is designed to compensate for voltage and frequency deviations caused by primary control in terms of fault conditions. Finally, tertiary control, as the highest and slowest control level, optimizes the operations in both operating modes and manages to control the power flow between the main grid and the microgrid.

Primary Control

The primary control is locally implemented at each DG, e.g., droop control for inverter-based distributed energy resources (DER). However, the secondary control usually relies on a centralized control structure. Central controllers are designed to issue global commands requiring information gathered from the whole system and thus a complex two-way communication network is also needed. This kind of communication network makes it vulnerable and may effect the system reliability. In [7], the authors replace the existing standard centralized secondary control with an efficient distributed control structure. They consider the microgrid as a multi-agent system where each DG is an agent. A voltage source inverter (VSI) is employed to connect a DG to the microgrid. The VSIs are interconnected through the physical power network configuration.

The primary control conventionally utilizes local droop controllers on distributed energy resources (DERs) to maintain microgrid voltage stability after the islanding. Primary control responds quickly and is designed to stabilize the voltage and frequency of the microgrid. Subsequent to an islanding event, the microgrid may lose its voltage and frequency stability due to the mismatch between the power generated and consumed. Primary control offers the plug-and-play capability for DGs and properly shares the active and reactive power among them, preferably without any communication links. This control level must mitigate circulating currents that can cause the over-current phenomenon in the power electronic devices and damage the dc-link capacitor.

Primary control ensures the stability of the DG but may cause frequency and voltage deviations. Secondary control restores the microgrid voltage and frequency to their desired reference values ω_{ref} and v_{ref} and compensates for deviations caused by the primary control. In islanded mode, ω_{ref} is set to the nominal angular frequency, and v_{ref} is set to the nominal voltage of

the microgrid. In the grid-connected mode, ω_{ref} and v_{ref} are dictated by the tertiary controller. The secondary control hierarchy has a lower dynamic response than that of the primary control. This timescale separation justifies considering the dynamics of the primary and secondary control loops to be decoupled and facilitates their individual designs [7].

Primary control responds quickly and is designed to stabilize the voltage and frequency of the microgrid. After an islanding event, the microgrid may lose its voltage and frequency stability due to the mismatch between the power generated and consumed. Primary control offers the plug-and-play capability for DGs and appropriately shares the active and reactive power among them, preferably without any communication links. This control level must mitigate circulating currents that can cause an over-current phenomenon in the power electronic devices and damage the dc-link capacitor.

Primary control provides the reference points for the voltage and current control loops of the DGs. These inner control loops are commonly referred to as zero-level control, which is generally implemented in either active-reactive power (PQ) or voltage control modes. In the PQ control mode, the DG active and reactive powers are regulated on predetermined reference points. The control strategy is implemented with a current-controlled VSI (CCVSI). In the voltage control mode, the DG operates as a voltage-controlled VSI (VCVSI), where the reference voltage is determined by the primary control, conventionally via droop characteristics.

The droop control is an autonomous control that operates at each DG and does not require intercommunication links between the DGs. The conventional active power control (frequency droop characteristic) and reactive power control (voltage droop characteristic) are used for frequency and voltage control. The principles of the conventional droop techniques are discussed next, using an equivalent circuit model of a VCVSI connected to an ac bus. Neglecting the switching ripples and high-frequency harmonics, the VCVSI is modeled as an ac source, with voltage $E\angle\delta$. It is assumed that the DG is connected to a microgrid bus with the voltage $V_{\text{com}}\angle 0$, and the DG output impedance and the line impedance are lumped as a single effective line impedance of $Z\angle\theta$. The complex power delivered to the microgrid bus is calculated as

$$S = V_{\text{com}} I^* = \frac{V_{\text{com}} E\angle\theta - \delta}{Z} - \frac{V_{\text{com}}^2\angle\theta}{Z} \tag{2.1}$$

from which the real and reactive powers are achieved, respectively, as

$$\begin{cases} P = \frac{V_{\text{com}} E}{Z}\cos(\theta-\delta) - \frac{V_{\text{com}}^2}{Z}\cos(\theta), \\ Q = \frac{V_{\text{com}} E}{Z}\sin(\theta-\delta) - \frac{V_{\text{com}}^2}{Z}\sin(\theta). \end{cases} \tag{2.2}$$

If the effective line impedance, $Z\angle\theta$ is assumed to be purely inductive, $\theta = 90°$, then (2.2) can be reduced to

$$\begin{cases} P = \frac{V_{\text{com}} E}{Z}\sin\delta, \\ Q = \frac{V_{\text{com}} E\cos\delta - V_{\text{com}}^2}{Z} \end{cases} \tag{2.3}$$

If the phase difference between the converter output voltage and the common ac bus, δ, is small enough, then, $\sin\delta \approx \delta$. and $\cos\delta \approx 1$. Thus, the frequency and voltage droop characteristics can be used to fine-tune the voltage reference of the VCVSI, based on the droop control equations

$$\begin{cases} \omega = \omega^* - D_P P, \\ E = E^* - D_Q Q \end{cases} \tag{2.4}$$

where E^* and ω^* are the primary control references. The droop coefficients, D_P and D_Q, can be adjusted either heuristically or by tuning algorithms (for example, particle swarm optimization). In the former approach, D_P and D_Q are determined based on the converter power rating and the maximum allowable voltage and frequency deviations. For instance, in a microgrid with N DGs, the constraints

$$\begin{cases} D_{P1}P_{n1} = D_{P2}P_{n2} = \cdots = D_{PN}P_{nN} = \Delta\omega_{\max} \\ D_{Q1}Q_{n1} = D_{Q2}Q_{n2} = \cdots = D_{QN}Q_{nN} = \Delta E_{\max} \end{cases} \tag{2.5}$$

must be satisfied, where $\omega_{\max}$ and $E_{\max}$ are the maximum allowable angular frequency and voltage deviations, respectively, and P_{ni} and Q_{ni} are the nominal active and reactive power of the ith DG.

The droop equations (2.4) are a feedback control scheme giving the required frequency ω and voltage E of the DG depending on the current power conditions P and Q observed by DG and the prescribed primary control reference frequency $\omega^*$and voltage E^*. The primary control reference frequency ω^* and voltage E^* are set by the secondary control level, as discussed below.

During the grid-tied operation of the microgrid, the DG voltage and angular frequency, E and ω, are enforced by the grid. The DG output active and reactive power references, P^{ref} and Q^{ref} can hence be adjusted through E^* and ω^* as

$$\begin{cases} P^{\text{ref}} = \frac{\omega^* - \omega}{D_P} \\ Q^{\text{ref}} = \frac{E^* - E}{D_Q} \end{cases} \tag{2.6}$$

The conventional droop method can be implemented with no communication links, which improves reliability. However, this method has some drawbacks:

1) The conventional droop method is developed assuming highly inductive effective impedance between the VCVSI and the ac bus. However, this assumption is challenged in microgrid applications since low-voltage transmission lines are mainly resistive.

2) As opposed to the frequency, the voltage is not a global quantity in the microgrid. Thus, the reactive power control in (2.4) may adversely affect the voltage regulation for critical loads.

3) For nonlinear loads, the conventional droop method is unable to distinguish the load current harmonics from the circulating current. Moreover, the current harmonics distort the DG output voltage. The conventional droop method can be modified to reduce the total harmonic distortion of the output voltages.

These potential drawbacks are widely discussed in the literature. A comprehensive survey on primary control schemes that correct these issues is provided in [8].

Secondary Control

Primary control ensures the stability of the DG but may cause frequency and voltage deviations. Secondary control restores the microgrid voltage and frequency to their desired reference values ω_{ref} and v_{ref} and compensates for deviations caused by the primary control. In islanded mode, ω_{ref} is set to the nominal angular frequency, that is, $2\pi \times 50$ or $2\pi \times 60$ rad/s, and v_{ref} is set to the nominal voltage of microgrid, $v_{nominal}$. In the grid-connected mode, ω_{ref} and v_{ref} are dictated by the tertiary controller. Secondary control hierarchy has a lower dynamic response

than that of the primary control. This timescale separation justifies considering the dynamics of the primary and secondary control loops to be decoupled and facilitates their individual designs. The frequency of the microgrid and the terminal voltage of a given bus are compared with the corresponding reference values, ω_{ref} and v_{ref}, respectively. Then the error signals are processed by individual controllers according to

$$\begin{cases} \delta\omega = K_{P\omega}\left(\omega_{\text{ref}} - \omega\right) + K_{I\omega}\int\left(\omega_{\text{ref}} - \omega\right)dt + \Delta\omega_s, \\ \delta E = K_{PE}\left(v_{\text{ref}} - E\right) + K_{IE}\int\left(v_{\text{ref}} - E\right)dt, \end{cases} \tag{2.7}$$

where $K_{P\omega}$, $K_{I\omega}$, K_{PE} and K_{IE} are the secondary controller parameters. The resulting signals $\delta\omega$ and δE are sent to the primary controller as changes to ω^* and E^* in the droop control (2.4) of the DG primary controller to compensate for the frequency and voltage deviations [8].

An additional term, $\Delta\omega_s$, is considered in the frequency controller in (2.7) to facilitate synchronization of the microgrid to the main grid. In the islanded operating mode, this additional term is zero. However, during synchronization, a phase-locked loop (PLL) module is required to measure $\Delta\omega_s$ [8]. During grid-tied operation, the voltage and frequency of the main grid [v_{ref} and ω_{ref}, in (2.7)] are considered as the references.

This standard secondary control (2.4) is implemented by a centralized node with access to all the required information in the microgrid. This implementation requires an extensive communication network, generally with two-way communication links. As such, this method is expensive, unreliable, and vulnerable to a single point of failure. This thesis describes how to replace (2.4) by distributed control structures using cooperative control techniques, which rely on sparse, inexpensive communication networks.

The secondary control of microgrid, which regulates the voltage to synchronize the average voltage of microgrid to the nominal setpoint, can be implemented through a centralized structure. In this structure, the microgrid secondary control center is responsible for determining and sending the voltage and power setpoints to the individual DERs through a complex two-way, high-bandwidth communication network [51]. However, this control structure is vulnerable to the single-point-of-failure in the control center, which could reduce system reliability. Alternatively, distributed cooperative control of microgrid with a sparse communication network [51, 7, 4] has been proposed recently to increase the reliability, scalability, and resilience of the microgrid control system.

Tertiary Control

Tertiary control is the highest (and the slowest) control level and is meant to optimize the economic performance of the microgrid and manage the power flow between microgrid and main grid [8]. In the grid-tied mode, the power flow between microgrid and main grid can be managed by adjusting the amplitude and frequency of the DG voltages. The active and reactive output powers of the microgrid, P_G and Q_G, are compared with the corresponding reference values, P_G^{ref} and Q_G^{ref}, to obtain the frequency and voltage references, ω_{ref} and v_{ref} based on

$$\begin{cases} \omega_{\text{ref}} = K_{PP}\left(P_G^{\text{ref}} - P_G\right) + K_{IP}\int\left(P_G^{\text{ref}} - P_G\right)dt, \\ v_{\text{ref}} = K_{PQ}\left(Q_G^{\text{ref}} - Q_G\right) + K_{IQ}\int\left(Q_G^{\text{ref}} - Q_G\right)dt, \end{cases} \tag{2.8}$$

where K_{PP}, $K_{I\omega}$, K_{PE} and K_{IE} are the controller parameters. The values ω_{ref} and v_{ref} are further used as the reference values for the secondary control, as in (2.7).

2.3 Communication Networks of Microgrid Systems

2.3.1 Preliminaries of Graph Theory for Microgrid Communication Networks

The communication network of a multiagent cooperative system can be modeled by a directed graph (digraph). A generic network topology of a microgrid system is usually characterized by a directed graph (digraph) $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A_G)$ with a nonempty finite set of N nodes $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$, a set of edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, and the associated adjacency matrix A_G. The set $\mathcal{N}_i = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$ denotes the neighbor set of node i. In a microgrid, DGs are the nodes of the communication digraph. The edges of the corresponding digraph of the communication network represent the communication links.

The weight is represented by a_{ij} for the edge from node j to node i, and $a_{ij} = 0$ if there is no data transfer from node j to node i. Considering N nodes on the graph, the adjacency matrix is defined as

$$A_G = [a_{ij}] \in \mathbb{R}^{N \times N}. \tag{2.9}$$

We define the diagonal in-degree matrix $D = diag\{d_i\} \in \mathbb{R}^{N \times N}$, where d_i is the sum of communication weights from neighbors of node.

$$d_i = \sum_{j \in \mathcal{N}_i} a_{ij}. \tag{2.10}$$

The Laplacian matrix of the graph is defined as $L = D - A_G$. One property of the Laplacian matrix L is that the row sums of L are all zero, because the row sums of D and A_G are equal.

A directed path from node i to node j is a sequence of edges depicted as $\{(v_i, v_k), (v_k, v_l), \ldots, (v_m, v_j)\}$. A graph is considered to have a *spanning tree* if there exists a *root node* with a directed path from that node to every other node in this graph. A graph is *strongly connected* if there exists a directed path between every two nodes, i.e., there exists a spanning tree where every node is a root node [7] [46].

2.3.2 Sparse Communication Topology to Implement Distributed Control

To implement the distributed control for microgrid systems, sparse communication topology is employed [7, 49, 54]. The controllers designed in this thesis must be supported by a local communication network. This network can be implemented by a TCP/IP communication protocol. The sampling time of the secondary control can be chosen to be much larger than the sampling time of the communication network. Communication links contain an intrinsic delay (for example, around 1 ms in the work [73]). Since the time scale of the secondary control is large enough, the communication link delays can be safely assumed to be zero. The software and hardware infrastructures required to implement the discussed distributed control protocols are quite modest. Voltage source inverters can be accompanied by commercial digital signal processors (DSPs) to implement internal voltage and current control loops [7]. The distributed control protocols embedded on each DG would not require dedicated signal processing units and do not impose a heavy processing burden on the existing DSP. The distributed adaptive controllers can be deployed on the existing processors, with a slight software update to pre-existing codes.

The distributed controllers presented here can admit both time-invariant and time-varying communication networks. For time-invariant communication networks, the adjacency matrix

A_G is fixed. The communication topology must be a graph containing a spanning tree in which the controller of each DG only requires information about that DG and its immediate neighbors on the communication graph. Therefore, the communication requirements to implement the controller with a time-invariant communication network are rather mild. However, the controller could be susceptible to communication link failures or data loss. In time-varying communication structures, each DG can send its own information to the neighboring DGs intermittently. The communication network is piecewise constant; that is, the adjacency matrix A_G changes intermittently at prespecified time instants. Therefore, the secondary control is robust against data loss and communication link failures [49, 54, 69]. According to [69], a time-varying communication network can provide synchronization if the sequential completeness condition is satisfied. The sequential completeness condition means that, given an infinite sequence of finite intervals, the resulting graph over each finite interval must contain a spanning tree. In the following chapters of this thesis, we study time-invariant communication networks for the sake of simplicity.

2.3.3 Communication Types for Microgrid Control

Architecture for the operation and control of microgrids requires an encrypted and reliable channel for successful operation. Encrypted communication across the microelements is the desired task. However, wired communication helps in short-distance data transmission; RS 232 and RS 385 are the most used ones, but with the increase in the size of the grid, wired data transmission is not advisable as it increases the complexity of the grid. So, opting for internet communication protocol based on ISO or open system interconnection, transmission control protocol/internet protocol, and UDP/IP protocol ensures security and fast transmission of the data. Table 2.1 summarizes the various communication links available for microgrids concerning coverage distance [70, 15, 61, 65, 74].

Besides, the microgrid structure has been categorized based on the range of communication into home area network (HAN), local area network (LAN), and wide-area network (WAN). Microgrid control architecture has various power system devices embedded with Intelligent Electronic Devices (IEDs). IED receives the signals from the DER and transmits them to the microgrid controller. Then, the microgrid controller generates the reference and control signals of voltage, current, frequency, real and reactive power to IEDs, which send the control signal to DERs and the connected loads. The human-machine interface is used for monitoring, event recording, and controlling purposes.

The microgrid communication is affected by various parameters such as speed, topology, and congestion status of the network and computational capability of IEDs hardware. The second transmission method is the Power Line Carrier (PLC), in which power lines are used for the transfer of the data. It is efficient in terms of performance and cost. Some of the disadvantages of using PLC are noise, deviation, and signal attenuation risk.

Every component of the microgrid has some delay requirements. The efficiency of the system can be increased by decreasing the bottleneck, congestion, loss of data packets, etc.; to achieve the desired efficiency, selection of protocols is the primary step. If the system exceeds the specified delay range, it leads to the electric malfunctioning in the microgrid. In this regard, the Quality of Service (QoS) concept is provided, in which priority treatment is given for the system having critical delay so that information reached on proper time and reduced the loss of data. In order to achieve it, the components are layered based on the requirement as follows [70].

1) Physical layer. This layer consists of sensors, actuators, smart plugs, smart meters, mi-

Table 2.1: Comparison of various communication channels

Mode	Frequency Band	Signal Rate	Power Consumption	Expense	Application in the Microgrid	Role Index in Resiliency Enhancement
Weightless	subGHz	0.1-25mbps	Low	Low	Smart metering	Low
LoRa	subGHz	< 50kbps	Low	Moderate	Utility metering	Low
NB-IoT	subGHz	0-1mbps	Moderate	High	Facility management services	Moderate
Wireless HART	2.4GHz	250kbps	Moderate	Moderate	Encryption of data	Moderate
WiFi	subGHz, 2.4GHz	0-50mbps	Moderate	Low	Intercommunication between neighboring systems	High
Z-wave	subGHz	30kbps	Low	Moderate	Smart metering	Low
802.15.4	subGHz, 2.4GHz	40, 250kbps	Low	Low	Equipment maintenance	Moderate
SigFox	subGHz	<1kbps	Low	Low	Monitoring power generation from the grid	Moderate
ZigBee	2.4GHz	230kbps	Low	Medium	Intra and inter communication between devices	High
LTE Cat 0/1	Cellular	0-10mbps	Moderate	High	Transmission of preprocessed data across the microgrid	High
Bluetooth	2.4GHz	1-3mbps	Low	Low	Communication across the components of the microgrid	High
3G	Cellular	10-30mbps	High	High	Autonomous interaction across the components	High

crocontrollers and storages. Based on the input data given by sensors, actuators perform the desired task, smart meter monitors the data given by sensors and controls the smart plug for transferring the power from the grid elements to the appliances. The interaction between the sensors, actuators, smart meter and plug with the computer is achieved using microcontrollers, and storage units are used to store the processed data received from the sensors.

2) Network layer. It is used to transfer the data from the physical layer to the application layer. These layers can be either wired or wireless. Wired includes cabled or fiber optics, use of fiber optics prevents intervention by electromagnetic radiation. The wireless mode includes 3G, LTE, WiFi. The only disadvantage of the former concerning the latter is the loss of data; data loss might account for up to 30% of the total data.

3) Cloud layer. The encryption of the data, retrieval of the data, and data management occur in this layer. The necessity of this layer arouses due to an increase in the amount of the data, loss of the data, and invasion of the privacy of the data.

4) Application layer. It is the final layer, which the end-users see and use. This layer manages the consumption of energy and the pricing of the system. The functioning of this layer entirely depends upon the other three layers. Over the years, there has been significant development in wireless networks. Internet of Things (IoT) is one such element, which has been turning around most of the theoretical applications to possibilities of achieving. One such application where IoT can be used is communication between various components across microgrids. Using IoT, the most critical microgrid factors, namely, the efficiency of power generation and distribution is achieved. Since the components of a microgrid are mostly non-homogeneous, various protocols have been used to achieve the desired requirement.

To communicate between the layers of the communication channel, Internet Protocol base or non-Internet Protocols base is applied. Internet-based protocols include Constrained Application Protocol (CoAP), Message Queuing Telemetry Transport (MQTT), 3G, LTE, Wi-Fi, whereas non-Internet Protocols include ZigBee based, z-wave based and Bluetooth.

2.4 Cyber-Physical Security of Microgrid Control Systems

2.4.1 Cyber-Physical Security

Traditionally, security used to be achieved on two fundamental levels; short-term security that deals with the stability of the demand-supply procurers. The dynamism enables the energy system to adapt quickly to sudden changes in the grid loads. Moreover, long-term security focuses on investments that support economic and sustainable development requirements. With the arrival of smart grids, which are defined as a composition of three interoperability infrastructures (communication, power system and information technology), has led the security problem to grow in complex imposing supplementary challenges threatening to introduce easier ways of causing damage to the fundamental security concerns and creating new ones.

Consequently, recent security assessment has focused on identifying the potential vulnerabilities introduced by the cyber layer and analyzes the possible impacts on energy systems, which has given birth to a brand new research area called cyber-physical security. A cyber-physical system is co-engineered collaborating domains of physical and computational counterparts, in which the crucial system tasks are handled with its physical part. Simultaneously, informatically

enhanced processes normally referred to as cyber- are responsible for maximizing the exploration of intelligent devices and applications.

The reason why academia recently chose to add the term "physical" to the equation is to shed light on the emerging threats imposed by connecting these two fundamentally different infrastructures, which practically may lead to problems that do not particularly belong to a failure of either system. In light of these assumptions, further investigation is still needed to confirm or deny the putative relationships.

The most indispensable objectives of security requirements considerations of any data transferring communication in the IT network security are known as CIA-triad, which stands for Confidentiality, Integrity, and Availability, respectively. Smart grid security is also built upon the previous trestles, but with a difference in priority order, where availability comes on top of the requirements, followed by integrity, accountability, and confidentiality. Other referencing emphasizes accountability as additional security criteria. This sequence of importance goes back to the severity of impacts resulting from tampering with these criteria.

Attackers can penetrate the smart grid communication systems using vulnerable entry points in the logical border surrounding a network, known as the Electronic Security Perimeter (ESP). Interventions may occur with the help of numerous mediums, such as the Universal Serial Bus (USB) thumb drive, viruses, and even software patches and updates. Despite the fact that cyber intrusions on cyber-physical systems (CPSs) can be found under different terms, such as bias injection attack, zero dynamics attack, denial of service (DoS) attacks, eavesdropping attack, replay attack, stealthy attack, covert attack, and dynamic false data injection attacks. These attacks can still be classified according to the one or multiple security criteria they are jeopardizing.

Intentionally introduced faults or malicious attacks triggered by the cyber layer leaving serious impacts on not only the technical aspects, but also on economic and social correlations in power network operations, are the focus of this research. Effects range from tampering with smart meters data or manipulating the forecasted load profiles to reach equipment damage or even complete blackouts.

2.4.2 Threats to the Microgrid Systems and Associated Vulnerabilities

It is essential to enhance the ability of decision-makers to assess and manage risks associated with microgrids that may change over time and vary geographically. Therefore, clear identification of threats and vulnerabilities and their likelihood and impact is the first step in building resilient microgrids [45].

- Threats. Threats are anything that can damage, destroy, or disrupt utility grid or microgrid operation. In other words, threats are what we are trying to protect microgrid against. They are typically natural or human-induced hazards that are not within the control of the site, such as wildfire, hurricane, cyberattacks, or physical attacks. Threats are identified by reviewing climate data and state hazard assessments, and stakeholder interviews with site staff or emergency management teams from the surrounding community.

- Vulnerabilities. Vulnerabilities are weaknesses within the microgrid either in infrastructure or processes. Unlike threats, they are within the control of the site and can be modified or mitigated to prevent or reduce the impact of a disruption. Vulnerabilities are identified through stakeholder interviews with energy managers, electrical engineers, maintenance staff, emergency managers, utilities, microgrid designers, operators, and end-users. They

can also be identified by reviewing contingency response plans and after-action reports following disasters or disruptions.

- Risk. Assessing the risk essentially means finding a way to quantify the relative potential of damage that various threats in the environment can cause to the microgrid. Therefore, risk is a function of threats exploiting vulnerabilities to impact the operations and damage or destroy the assets. Thus, we define the risk factor metric based on the probability of a threat, the probability of a vulnerability's exploitation given that threat, and the impact of the vulnerability. This definition of risk factor has been extensively used in the literature. A microgrid system-level risk assessment is the first step in building resilient microgrids.

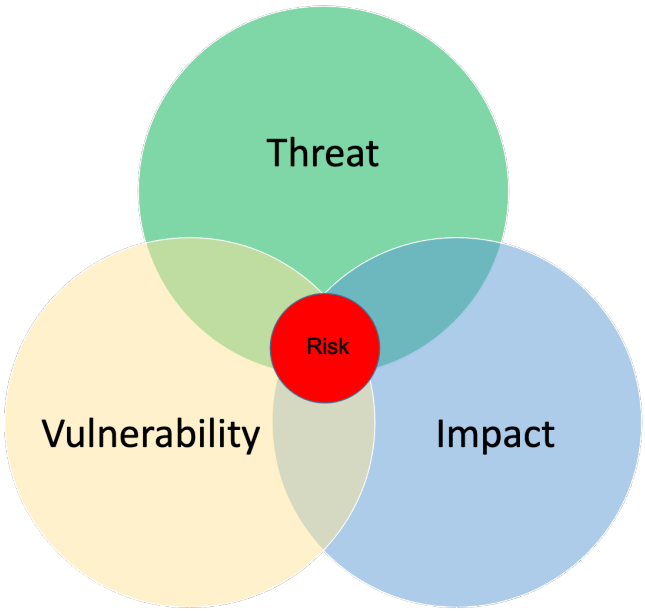

Figure 2.2: Risk Factor Venn Diagram

To evaluate the relationship between threats and vulnerabilities, the threat likelihood score is multiplied by the vulnerability probability and impact scores to create a risk score for each specific threat-vulnerability combination. Risk is created when a threat can exploit an already present vulnerability in the microgrid. The magnitude of the risk is determined by the likelihood of the threat and vulnerability, as well as the scale of damage the vulnerability could cause if exploited. Therefore, risk factor is calculated as intersection of these three factors, as shown in Figure 2.2. The risk score enables analysis and ranking of the risks to prioritize mitigation actions. Threats and hazards change over time, so the risk assessment should be updated on a regular basis.

For a complex cyber-physical system like a microgrid, there are multiple dimensions of resilience. They can be classified into three domains:

- Physical resilience. Sustaining physical infrastructure (which includes assets and electrical cables connecting the assets) during a disruptive event and continuing or restoring operations rapidly after the event is over.

- Cyber resilience. Identifying and defending against various types of cyberattacks as well as accidental failures and maintaining safe performance during such an event.

- Communications resilience. Maintaining safe and stable operational performance during communications failures from the utility grid end or internal microgrids communications.

More detailed discussions about these threats are shown as follows.

Physical Threats

Physical threats include natural hazards (wildfires, hurricanes, floods), changing climate (more extreme temperatures), and human-induced attacks (including terrorist attacks, such as shooting substations) [45].

- Natural hazards. Weather events like tropical cyclones, hurricanes, floods, earthquakes, and wildfires can cause damage to the utility grid, and if the epicenter of the impact is geographically close to the microgrid, then the impact can be of equivalent order.
- Changing climate. Increasingly extreme temperatures caused by the changing climate can put undue strain on a microgrid especially when it is operating in islanded mode. The spike in cooling and heating loads can cause harm in two potential ways: 1) excessive power demand can cause equipment and devices to operate above their rated capacity, resulting in permanent damage; and 2) increased demand can cause load-shedding in the microgrid. If load-shedding strategies are not automated, then there is an increased risk of a site-wide outage.
- Human-induced attacks. Physical attacks, such as bombings or shootings of major microgrid components like generators and storage assets, could cause serious damage and disable a microgrid. Electromagnetic pulse attacks can also take out generation assets and controllers in the microgrid. The attacks can be from one gunman shooting specific equipment to a massive bombing that can destroy the whole site.

Cyber Threats

Cyberattacks on microgrids can be classified into the following three types [67, 37]: attacks on availability, integrity, and confidentiality.

- Availability. Attacks on availability attempt to limit or delay access to data. In a microgrid, this may be extended to include availability of power in addition to data. Attacks against availability are generally referred to as Denial of Service (DoS) attacks. DoS attacks in microgrids generally include not only blocking resource access but also delaying the timing of critical message exchange. For example, jamming communications channels between various parts of the microgrid.
- Integrity. Attacks on integrity interfere with the accuracy or trustworthiness of data. Spoofing, tampering, and elevation of privilege are all integrity violations. For the microgrid, interference with the quality of power might also be considered a violation of integrity. For example, an attacker uses a Domain Name System (DNS) hijack to feed inaccurate price data to the controller, showing diesel as cheaper than natural gas. The result is that the microgrid uses diesel instead of natural gas, costing the operator money. Faking a microgrid disconnect from the grid signal compromises the integrity. Other examples of integrity attacks include sending false commands to the microgrid controller, causing unwanted or premature islanding; and changing the setting groups of protection relays in a microgrid, causing the relays to trip erroneously.

- Confidentiality. Attacks on confidentiality are attempts to gain access to privileged information. This might include operational or billing data, or any other information that should not be accessible to unauthorized entities. For microgrids participating in markets, the bidding price data can be stolen.

In a microgrid system, each interface or trust boundary represents a potential attack surface that must be analyzed for vulnerabilities. As methods for identifying specific threats are extensively well-documented elsewhere, we will focus here on those attack surfaces that are common to microgrid systems in general, rather than on specific attacks, which must necessarily be identified on a case-by-case basis. The vulnerable attacks surfaces along with the types of threats that can impact the operations of microgrids are listed as Table 2.2 [45].

For microgrids with advanced controls and analytics capabilities, maintaining cyber-reliability is a matter of concern as well. The more complex the control algorithm, the higher the risk associated with having unforeseen/blinding scenarios making it behave in an unintended manner. Moreover, the higher the sophistication of the software stack, the greater the risk of injecting unintended bugs while developing the stack further. Since these challenges are not imposed by an external entity, they are not categorized as threats against which resilience solutions should be built in the microgrid. But they do pose a challenge for maintaining the reliability of microgrid, which is a significant aspect as well.

Communications Failure Threats

The modern grid is envisioned to be highly integrated with multidirectional data exchange and information communication between all its parts. The vision is consistent with microgrids as well for deploying a more active and responsive infrastructure for faster demand response and more exhaustive energy management capabilities. It is indispensable for an intelligent microgrid to rely on communication networks. On the other hand, using sophisticated communication schema makes the microgrid vulnerable, due to the increased risk of communication failure, along with the microgrid's growing dependence on electric systems and public communication networks.

In the case of microgrids, if the communication failure happens during grid-connected operation, the microgrid's controller will lose access to grid-side information, resulting in its inability to maintain the commanded power and power-factor at the point-of-interconnection. The microgrid should be able to maintain stable operations and stop feeding the power to the grid if it has been doing demand management before the communication failure. In worst case scenarios, the microgrid needs to have the ability to island itself safely. Communication failure internal to the microgrid is more of a reliability concern, as it is not imposed by an external entity. If not addressed appropriately, it can adversely impact the operation of the microgrid and damage the assets if left unattended for longer durations especially when operating in islanded mode.

Threats Due to Interdependencies Between Various Systems

Electricity, oil, natural gas, transportation, telecommunications, and water systems are all interdependent critical infrastructure (CI) systems. The operations of power systems are in constant exchange with other CI systems. On one level, electric power systems support other CI systems in their operations, whereas, on the other level, electric power systems depend on other CI systems to be able to operate.

These tightly intertwined interdependencies between other CI systems and electric power systems increase their vulnerabilities by exposing them to second-order and third-order threats.

Table 2.2: Cybersecurity Risk Analysis

Attack Surface	Examples	Threats/Vulnerabilities	Potential Impact
Wired Links	Ethernet, Phone lines, RS-232, RS-422, RS-485, I2C (often used for sensors)	Line may be cut or unplugged, causing DoS. An attacker may splice into the wires and eavesdrop, tamper with, and inject data into the medium	Blocking resources Delaying the timing of critical message exchange
Wireless Links	IEEE 802.11 (WiFi), Cellular networks, ZigBee/XBee	The broadcast nature of the medium enables anyone to listen to communications. Signals may be jammed, resulting in DoS.	Confidentiality lost With jammed signals, SCADA system loses connection with microgrid's analysis layer, causing sub-optimal operation.
Unencrypted Communications	Most low-level serial protocols including I2C and SPI Unencapsulated TCP and UDP, including HTTP, FTP, Telnet	An attacker who can access the medium can view, modify, and send meaningful data.	Data confidentiality lost?market participation bidding strategy compromised.
Unauthenticated Communications	DNS, HTTP, most lowlevel serial protocols	Attacker can pretend to be legitimate data source or user	Can cause unwanted islanding of microgrid or unwanted change to grid-connected mode during island (can be a life-threating issue if maintenance workers are working on electrical components).
Exposed Endpoints	Sensors, HMI, Remote Servers, Anything the attacker can gain physical or virtual access to	Attacker could falsify actionable data. Attacker could damage interface, causing DoS.	Wrong price-signal or weather data will cause suboptimal and noneconomic operational dispatch strategy of the microgrid.
Human	Technicians, administrators, contractors, any user with the system or network access	Attacker gains real credentials from a legitimate user and can then access normally restricted systems. Insider threat: a legitimate user exploits their access to cause damage	Unwanted islanding Dispatch strategy causes economics losses in operations. Unwanted outages

Microgrids are also equally, if not more, vulnerable to the threats posed by such interdependencies. Microgrids operate as integrated energy systems where gas, water, distributed generation, and utility grid supplies are coordinated to supply various kinds of loads. Interruption in the supply of any of these critical utilities (electricity, natural gas, water, or heat) can negatively impact the ability of the integrated energy system to meet loads and affect the operational economy of the microgrid.

2.5 Attack Detection of Microgrid Systems

There are different methods to detect attacks for microgrid system, including traditional methods based on state estimation and also data driven methods like machine learning. We give a detailed description about these methods and elaborate their advantages and disadvantages in detection attacks targeting microgrid systems.

2.5.1 Attack Detection based on State Estimation

For smartgrid systems including microgrids, false data injection attacks are defined in the following model:

$$\mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{n}, \tag{2.11}$$

where $\mathbf{x} \in \mathbb{R}^D$ contains the voltage phase angles at the buses, $\mathbf{z} \in \mathbb{R}^N$ is the vector of measurements, $\mathbf{H} \in \mathbb{R}^{N \times D}$ is the measurement Jacobian matrix, and $\mathbf{n} \in \mathbb{R}^N$ is the measurement noise, which is assumed to have independent components. The attack detection problem is defined as that of deciding whether or not there is an attack on the measurements. If the noise is distributed normally with zero mean, then an state vector estimation (SVE) method can be employed by computing

$$\hat{\mathbf{x}} = \left(\mathbf{H}^T \Lambda \mathbf{H}\right)^{-1} \mathbf{H}^T \Lambda \mathbf{z}, \tag{2.12}$$

where Λ is a diagonal matrix whose diagonal elements are given by $\Lambda_{ii} = v_i^{-2}$ and v_i^2 is the variance of $n_i \forall i = 1, 2, \ldots, N$. The goal of the attacker is to inject a false data vector $\mathbf{a} \in \mathbb{R}^N$ into the measurements without being detected by the operator. The resulting observation model is

$$\tilde{\mathbf{z}} = \mathbf{H}\mathbf{x} + \mathbf{a} + \mathbf{n}. \tag{2.13}$$

The false data injection vector, a, is a nonzero vector, such that $\mathbf{a}_i \neq \mathbf{0}, \forall i \in \mathcal{A}$, where $\mathcal{A}$ is the set of indices of the measurement variables that will be attacked. The secure variables satisfy the constraint $\mathbf{a}_i = \mathbf{0}, \forall i \in \overline{\mathcal{A}}$, where $\overline{\mathcal{A}}$ is the set complement of $\mathcal{A}$.

To detect an attack, the measurement residual is examined in ℓ_2-norm $\rho = \|\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}\|_2^2$, where $\hat{\mathbf{x}} \in \mathbb{R}^D$ is the SVE. If $\rho > \tau$, where $\tau \in \mathbb{R}$ is an arbitrary threshold, which determines the tradeoff between the detection and the false alarm probabilities, then the network operator declares that the measurements are attacked.

Limitations of SVE-based attack detection methods

One of the challenging problems of SVE-based attack detection methods is that the exact recovery of state vectors is a challenge for SVE-based methods in sparse networks [12, 35], where the Jacobian measurement matrix is sparse, and the performance of this approach is restricted by the sparsity of the state vectors.

In addition, unobservable attacks can be constructed even if the network operator can estimate the state vector correctly [39, 35]. For instance, if $\mathbf{a} = \mathbf{H}\mathbf{c}$, where $\mathbf{c} \in \mathbb{R}^D$ is an attack vector, then the attack is unobservable using the measurement residual ρ .

Moreover, detection methods broadly applying state estimation processes, e.g., using Kalman filter [43], sparse optimization [38], graphical methods [6] to address FDI attack detection all require **centralized communication** structures, which is difficult to achieve for distributed

microgrid control systems. The FDI attack detection in distributed control systems deploying Kullback-Leibler (KL) divergence is addressed in [51]. However, all the above attack detection techniques only deal with a single type of cyber attack (FDI). A distributed state estimation method based on the alternating direction method of multipliers (ADMMs) is proposed to detect hybrid cyber-attacks (FDI and DoS attacks) in [16]. However, this method mainly benefits the large-scale power transmission systems due to its heavy computation complexity.

2.5.2 Attack Detection based on Machine Learning Techniques

Machine learning is a form of artificial intelligence that enables computers to learn and improve without being explicitly programmed. Researchers have proposed different machine learning algorithms to enable FDI attacks detection. In this subsection, we first discuss the metrics used for the performance evaluation of machine learning-based FDI attacks detection algorithms. Leveraging these metrics, we review the existing literature and compare their performance.

Performance metrics

A multitude of metrics has been adopted to evaluate the performance of the detection methods. Accuracy, precision, recall, F1 score, and receiver operating characteristic (ROC) curve are among the most common metrics. With the true label of a measurement and its predicted label, the output of a detection model can be divided into true positive (TP): indicating a correct positive prediction, true negative (TN): a correct negative prediction, false positive (FP): an incorrect positive prediction, and false negative (FN): an incorrect negative prediction.

Accuracy is the ratio of the number of correct predictions to the number of total predictions

$$\text{Accuracy } = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{2.14}$$

Accuracy is meaningful when the measurement data is balanced (when the number of positive and negative measurement samples are equal). To evaluate the performance of the detection model with imbalanced data, precision, recall, and F1 score are often considered. Precision (also called positive predictive value) describes the capability of a model to identify an attack's overall true positive predictions [111]. It is represented as the ratio of the correct positive predictions to the number of samples labelled as positive

$$\text{Precision } = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{2.15}$$

The recall (also called sensitivity) gives the model the capability to identify all attacks. Recall is described as the ratio of the number of correct positive predictions to the number of positive samples

$$\text{Recall } = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{2.16}$$

From (2.15) and (2.16) , it can be observed that precision and recall are closely related. For a given model, a decrease in FP (precision) leads to an increase in FN (recall), and vice versa. To achieve an optimal trade-off between precision and recall, the F-Measure is used to combine

these two metrics. To avoid being heavily impacted by extreme values of precision or recall, the F-Measure score is designed as the harmonic average of precision and recall, as shown below

$$\text{F1 score } = \frac{2 \times (\text{ Recall } \times \text{ Precision })}{\text{Recall } + \text{ Precision}}. \tag{2.17}$$

For a given classifier, recall and precision may vary a lot with different sets of measurements, e.g., balanced data and unbalanced data, making it hard to evaluate the performance of a classifier. To have a stable representation of the classifier performance, the ROC curve is often used. In a continuous binary classifier, the output is a continuous variable ranging from 0 to 1. Thus, a threshold is leveraged to divide the outputs into positive and negative. Different thresholds result in different TP rates (TPR, equals to recall) and FP rates (FPR = FP/(FP + TN)). ROC curve plots TPR (y-axis) against FPR (x-axis) under different thresholds. The closer the ROC curve is to the upper left corner or coordinate (0, 1) (the larger the area under the curve), the better the performance. ROC illustrates how well the detection method distinguishes between the attacked and the secured measurements

2.5.3 Different machine learning methods for attack detection

Supervised learning

Machine learning algorithms can be classified into supervised learning, semi-supervised learning, and unsupervised learning. In supervised learning, inputs and desired outputs are provided to the machine to construct a function that maps the input to the desired output.

The attack detection problem can be modeled as a binary supervised machine learning problem. Detecting FDI attacks is considered a supervised binary classification problem. The objective of the binary classifier is to decide whether the given data $\mathbf{s}$ with m features is either $\mathbf{z}$, a normal measurement vector (negative class) or $\mathbf{z}_a = \mathbf{z} + \mathbf{a}$, an attacked measurement vector (positive class) [33]. The output class labels are

$$y = \begin{cases} +1 & \text{for } a \neq 0 \\ -1 & \text{for } a = 0 \end{cases} \tag{2.18}$$

where $\mathbf{a}$ is the attack vector.

The common used supervised learning algorithms are perceptrons, support vector machines (SVMs), knearest neighbours (KNNs), and logistic regression.

- Perceptron. In perceptrons, a weight vector $w \in \mathbb{R}^{M_{\text{Tr}}}$ is trained such that the output label, y_i, of a sample s_i is predicted by the following classification function:

$$f(s_i) = \text{sign}(w \cdot s_i) = \begin{cases} 1 & \text{for } w \cdot s_i \geq 0 (a \neq 0), \\ -1 & \text{for } w \cdot s_i < 0 (a = 0). \end{cases} \tag{2.19}$$

  During the training phase, the weight vector is updated for each training sample as $w(i+1) = w(i) + \Delta w$, where $\Delta w = \gamma (y_i - f(s_i)) s_i$ and γ is the learning rate. From the classification function, we can see that the convergence of the perceptron algorithm can be guaranteed when the samples are linearly separable. Therefore, it is suitable for FDIs detection only when a hyperplane can separate the measurements.

- Support Vector Machine (SVM). In SVMs, a hyperplane is constructed to separate two different classes. The hyperplane can be represented by a weight vector w and a bias value

b. The decision boundaries for the linear separable data can be formulated as two parallel hyperplanes using (2.20)

$$\begin{cases} w^{\mathrm{T}} s_i + b = +1, & \text{if } y_i = +1 \\ w^{\mathrm{T}} s_i + b = -1, & \text{if } y_i = -1 \end{cases} \tag{2.20}$$

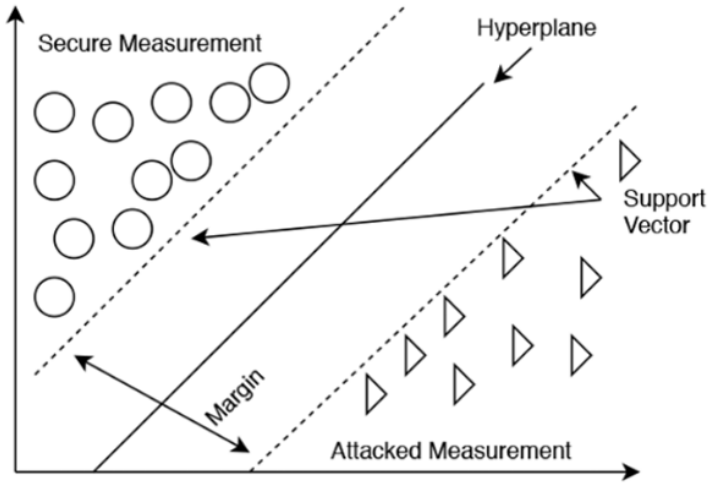

Figure 2.3: SVM concept [50, 57].

In (2.20), each line represents a support vector in Figure 2.3. Margin D is the separation area between the two support vectors and can be computed as

$$\begin{gathered} \min_{w,\xi,b} \|w\|_2^2 + \zeta \sum_{i=1}^{M^{\mathrm{Tr}}} \xi_i \\ \text{Subject to:} \quad y_i \left(w^{\mathrm{T}} \cdot s_i + b \right) - 1 + \xi_i \geq 0 \\ \xi_i \geq 0 \forall i = 1, 2, 3, \ldots, M_{\mathrm{Tr}} \end{gathered} \tag{2.21}$$

where ζ is the adjustable regularisation parameter, ξ_i is the slack variable for the non-linear separable training set, and M_{Tr} is the feature vector.

- k-Nearest Neighbor (KNN). KNN is another supervised learning algorithm that assigns labels to an unlabelled sample according to its KNNs. The Euclidean distance is used to determine the similarity between a given labeled sample, s_i, and an unlabelled sample, s_i'. The set of KNNs for a given measurement sample can be determined using the Euclidean distance as follows :

$$\begin{gathered} \left\| s_i' - s_{i(1)} \right\|_2 \leq \left\| s_i' - s_{i(2)} \right\|_2 \leq \ldots \leq \left\| s_i' - s_{i(M_{\mathrm{Tr}})} \right\|_2 \\ \aleph \left(s_i' \right) = \left\{ s_{i(1)}, s_{i(2)}, \ldots, s_{i(k)} \right\} \end{gathered} \tag{2.22}$$

Majority voting is one of the most commonly used methods for assigning labels from the set of KNNs of s_i. KNN is easy to implement but it fails to work when the size of the data sample is smaller than the dimension of the feature vector.

- Logistic Regression. The logistic regression algorithm assumes that the distribution of the label y_i of data s_i follows the following logistic function:

$$P\left(y_i \mid s_i\right) = \frac{1}{1 + \exp\left(-y_i\left(w \cdot s_i + b\right)\right)}. \tag{2.23}$$

The weight vector w is estimated by maximizing the following cost function:

$$J(w) = -\frac{1}{M_{\mathrm{Tr}}} \sum_{i=1}^{M_{\mathrm{Tr}}} \log\left(1 + \exp\left(-y_i\left(w \cdot s_i + b\right)\right)\right). \tag{2.24}$$

A brief comparison of various machine learning methods is presented in [50]. The paper is one of the first research works to utilize supervised learning algorithms for FDI detection in smartgrid systems. The authors used a hierarchical network in which the measurements are grouped as clusters, and each cluster is regarded as a sample si . The false data is directly injected into the measurements before the measurements are grouped into clusters. The detection method is based on the observations made in [38]. According to [38], the distance between samples determines the attack vector:

$$\left\|s_i - s_j\right\|_2 = \begin{cases} \left\|z_i - z_j + a_i - a_j\right\|_2, & \text{if } a_i, a_j \neq 0 \\ \left\|z_i - z_j + a_i\right\|_2, & \text{if } a_i \neq 0, a_j = 0 \\ \left\|z_i - z_j\right\|_2, & \text{if } a_i, a_j = 0 \end{cases} \tag{2.25}$$

Therefore, by looking into the distance between two samples, it is possible to detect an FDI. In their experiment, the performance of different machine learning algorithms is evaluated against FDI attacks with different sparsity k/m (the ratio of measurements that the attacker has access to). Accuracy, precision, and recall are used as performance metrics. The results proved that the machine learning algorithms perform better than any other algorithm (e.g., state vector estimation approach) in detecting FDIs. Although SVMs achieved the highest prediction accuracy, they also present some limitations, such as the selection of the kernel and sensitivity to the sparsity of the system. KNN is very sensitive to system size and performed better for small-sized systems. Despite conducting plenty of experiments, the authors of [50] did not evaluate the performance of detection algorithms for stealthy FDIs. Furthermore, only the sparsity of injected data was considered. The magnitude of the injected data could potentially impact both the operation of the system and the performance of the detection methods. Last, the lack of attack data can result in imbalanced data samples during the detector training process affecting its classification accuracy.

Considering the limitations of [50], a similar work is conducted in [71]. Both works utilize closely related system models. Two assumptions are taken into consideration when the attack vectors are created in the adversary model: (i) that the injected value a_i is greater than the noise level and (ii) that the mean of the attack vector a_i is larger than the variance of the attack vector. Attack vectors with different sparsity and variance (The variance reflects the magnitude of disturbances caused by false data.) are tested in their experiments. To solve the imbalanced data problem, they propose the extended nearest neighbor (ENN) algorithm. For each class, ENN measures the average ratio of the nearest neighbors belonging to the same class. Instead of using majority voting, the label of a sample was predicted by finding the class which presents the greatest ratio variability with the sample labeled in that class. The performance of SVMs, KNN, and ENN is then experimentally evaluated. Accuracy and F1 scores are used as performance metrics. SVMs outperformed KNN and ENN in most of the test cases. A critical range of

sparsity was observed in which the accuracy and F1 score increased significantly. However, this is reasonable since the distance increases $\|s_i - s_j\|$ when the sparsity increases, which leads to more distinct classes. The experiment was conducted on the IEEE 30 bus system. The detection performance of the algorithms in larger systems was not demonstrated.

Esmalifalak et al. [17] proposed a distributed SVM algorithm. Each substation owned a training set and stealthy FDI attacks, which could bypass BDD methods based on their corresponding residuals. Before training, PCA is applied to the training set to reduce the feature dimension. To avoid a huge volume of data exchange, each substation is trained using a local classifier, and only the locally optimized weight vectors are exchanged. Their optimization problem, (2.21), is provided below

$$\begin{gathered} \min_{w_k,\xi_k,b_k} \|w_k\|_2^2 + \zeta \sum_{k=1}^{n} \sum_{i=1}^{\mathrm{Tr}} \xi_{ki} \\ \text{Subject to: } y_{ki}\left(w_k^{\mathrm{Tr}} \cdot s_{ki} + b_k\right) - 1 + \xi_{ki} \geq 0 \\ \xi_{ki} \geq 0 \forall i = 1, 2, 3, \ldots, m; k = 1, \ldots, n \end{gathered} \tag{2.26}$$

where n is the number of substations and w_k is the local optimization parameter. The alternating direction method of multipliers is used to solve this distributed optimization problem. Experiments are performed on the IEEE 118 bus system. The authors empirically verify the convergence of distributed SVM classifiers to centralized SVMs with different numbers of substations.

To recapitulate, supervised learning methods have achieved superior performance in comparison with traditional residual-based BDD methods. Among the aforementioned algorithms, SVMs have demonstrated to achieve the highest accuracy. According to Esmalifalak et al. [17], the curse of the dimensionality problem can be solved by leveraging PCA, which significantly enhanced the efficiency of machine learning algorithms. Nevertheless, most of the attack data are often generated randomly in the experiments while a sophisticated adversary would deliberately choose attack vectors considering the system dynamics. The performance of the proposed methods against such sophisticated FDI attacks still remains unknown. Moreover, most of the prior works conducted simulation experiments. Thus, the efficiency of existing methods, if applied to real power system deployments, cannot be guaranteed.

In addition to FDI attacks, supervised machine learning techniques are also used for detecting other cyber attacks. In [44], the authors proposed DoS attack detection schemes based on SVM and Neural Network classifiers. In [29], the authors present a machine learning-based classification framework to detect the Denial of Service (DoS) attacks. The framework consists of five stages, including 1) selection of the relevant Dataset, 2) Data pre-processing, 3) Feature Selection, 4) Detection, and 5) reflection of Results.

Semi-supervised learning

In semi-supervised learning, the majority of the given data is unlabelled. Although semi-supervised learning algorithms are the least common learning approaches applied for the detection of FDI attacks, we still introduce them in this survey work for completeness. An example of a semi-supervised learning algorithm is a semisupervised SVM (S_3VM). S_3VM assumes that samples with different labels are clustered into different groups and that the diameter of each cluster is small enough to avoid sub-clusters [11] The objective function of S_3VM is defined as

$$\min_{w,b} \zeta \left[\sum_{i=1}^{M_{\mathrm{Tr}}} L^{\mathrm{Tr}}\left(s_i, y_i\right) + \sum_{i=1}^{M_{\mathrm{Ts}}} L^{\mathrm{Ts}}\left(s_i'\right) \right] + \|w\|^2 \tag{2.27}$$

where $y = w^{\mathrm{T}} s_i + b$ and ζ is the regularisation parameter, L^{Tr} and L^{Ts} are the loss function of the training and test samples, respectively.

Foroutan and Salmasi [20] investigated FDI attack detection methods by using the S_3VM based on Gaussian mixture distributions. According to Filho [19], a finite mixture distribution model, defined as a convex combination of two or more probability density functions, is capable of approximating any arbitrary distributions due to its flexibility in modeling complex data. The authors assume that all FDI attacks have the same amount of energy or c vector, where $a = H\,c$, and that they have the same mean squared error. In the adversary model, the attack vectors are designed based on the minimum energy residual attack and sparsest attack, introduced in [35, 25], respectively. In the training phase, a positive data set, i.e., a data set with attacked measurements, was used to build the Gaussian mixture model. Then, a mixture of data sets consisting of both positive and negative labels (attacked and normal measurements) determines the threshold. In the evaluation phase, the unlabelled data set used for testing and F1 score evaluates the performance of the results. PCA was applied to the data set to overcome measurement dimensionality issues. The authors demonstrate the performance of the proposed detection method on the IEEE 118 bus power system. To generate diversified datasets, different topological networks are constructed using Monte-Carlo simulations. The performance of the proposed detection method depends on the selection of a proper threshold. A high-threshold value reduces recall while a low-threshold value lowers precision. The impact of the detection algorithms is illustrated with a ROC curve. Although the proposed model demonstrates a high F1 score compared with other machine learning algorithms (e.g., SVMs and perceptrons), it performs well only when the attacked measurements and the real measurements lie in distinct regions of the feature space, i.e., the attacked data can be effortlessly isolated.

Another detection method based on the S_3VM algorithm is proposed in [50]. The input samples are integrated into the cost function forming the following optimization problem:

$$\min \| W \|_2^2 + \zeta_1 \sum_{i=1}^{M_{\mathrm{Tr}}} L^{\mathrm{Tr}} \left(S_i, y_i \right) + \zeta_2 \sum_{i=1}^{M_{\mathrm{Te}}} L^{\mathrm{Te}} \left(S_i' \right), \tag{2.28}$$

where ζ_1 and ζ_2 are the cost parameters, L^{Tr} and L^{Te} are the loss functions for the training and testing samples. In the simulation, the authors use default values for the parameters as suggested in [33]. The experiments are conducted on IEEE 9, 57, and 118 bus systems, and the measurement matrix is generated using Matlab's Matpower toolbox. Compared with supervised learning algorithms, S_3VM demonstrated improved robustness against data sparsity despite the fact that S_3VM still remains sensitive to unbalanced data samples.

Unsupervised learning

Unsupervised learning algorithms group the unlabelled samples based on the similarities and differences between samples, without any prior training. Clustering is the most popular unsupervised learning method where the measurement samples are grouped based on the distance between samples in the feature space. Different distance metrics can be chosen (e.g., Euclidean distance).

K-means (also called hard c-means) is an example of a clustering method, which divides data into k groups. k-means iteratively assigns each data point to one of the k groups, whose centroid has the minimum distance to the data point in the feature space. Each centroid in a cluster is a collection of features, which define a group. The centroids are updated at each round. Several techniques are used to validate the k-value including cross-validation and other

information criteria. Fuzzy c-means (FCM) clustering is another type of k-means clustering, which assigns data points to two or more clusters [48]. Each point belongs to a cluster based on a corresponding probability value, rather than having a binary value as is the case of k-means clustering. In FCM, the clustering problem can be solved by minimizing the following equation:

$$J = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^{m} \left\| x_i - C_j \right\|^2, \quad 1 \leq m < \infty \tag{2.29}$$

where N is the number of data points, $C = 2$ is the number of the clusters (cluster of attacks and cluster of normal measurements), x_i is the ith dimensional measured data, and C_j is the center of the jth cluster, which is determined using

$$C_j = \frac{\sum_{i=1}^{N} u_{ij}^{m} \cdot x_i}{\sum_{i=1}^{N} u_{ij}^{m}} \tag{2.30}$$

where u_{ij} is the degree of membership of the ith measurement. The updated membership u_{ij} computed by the following equation:

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\left\| x_i - C_j \right\| \right) / \left(\left\| x_i - C_k \right\| \right)^{2/(m-1)}} \tag{2.31}$$

Mohammadpourfard et al. [47] presented a visualization based on the unsupervised anomaly detection method and FCM clustering to detect and locate FDI attacks. The authors also propose a localization method that helps in identifying the attack after topology reconfigurations and the integration of different resources. When FDI attacks occur, the probability distributions of system states deviate significantly from normal states, hence enabling FDI attack detection. First, the authors normalize the data, and then various statistical measures are applied to characterize the probability distribution of each state vector. PCA is applied to the new feature set to reduce the dimensionality of data and to visualize them in a two-dimensional space where the grid operators can determine whether an attack has occurred or not (using patterns of normal and abnormal data). FCM is used to detect outliers and locate the FDI attacks. Load data from the New York Independent System Operator are used for the simulations. FDI attacks data are generated on the IEEE 9 and 14 bus system with the assumption that the adversary decreases or increases a specific state variable by at least 6% of its original value. The proposed method is applied to two different FDI attack scenarios: detecting FDI attacks with and without topology changes. Compared to supervised learning algorithms such as SVMs and KNN, the proposed model achieves higher detection accuracy [47].

Yang et al.[72] proposed three different anomaly detection approaches to detect FDI attacks: (i) local outlier factor, (ii) isolation forest, and (iii) robust covariance estimation. The local outlier factor is a density-based anomaly detection method that measures the local standard deviation of any given data point from its neighbors by comparing their local density [34]. Isolation forest is an outlier detection technique based on decision trees that does not employ any distance or density measure and can handle large, high-dimensional datasets. Robust covariance estimation is another anomaly detection method based on the elliptic envelope fitting method, which assumes that the given data is a Gaussian distribution and defines the shape of the data. An IEEE 14 bus system case is used to evaluate the mentioned detection approaches. Attack vectors generated with Gaussian distributed non-zero elements have the same mean and variance as the original measurement set. The authors use PCA to reduce the data dimension from 41 to 2, to reduce noise, and simplify the detection problem. All proposed methods achieve high accuracy for FDI attacks detection. However, these three detection methods achieve high detection rates only when the contamination rate is known and small[72] .

Deep neural network

Deep learning algorithms mimic the human brain structure, functions, and are one of the fastest developing artificial intelligence technologies. Although deep learning algorithms require time and large amounts of data for their training stage, they have been applied for FDI attacks detection achieving high-accuracy rates [22].

Multilayer perceptrons (MLPs), also called feed-forward neural networks, are deep learning models where information flows in only one direction, i.e., from the input through the hidden layers to the output, as shown in Figure 2.4. They consist of an input layer, which receives the input signals, one or more hidden layers to construct the approximation function, and an output layer that predicts the final decision based on the input and approximation function.

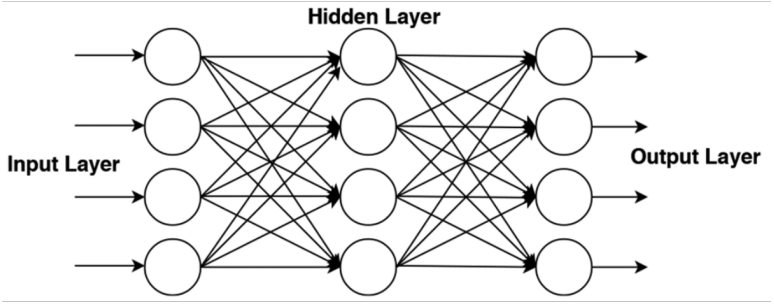

Figure 2.4: MLP concept [57].

Multiple studies where MLPs have been applied to detect FDI attacks have been reported in the literature [47, 21, 62]. In these works, the FDI attack detection problem is formulated as a supervised classification problem. In MLP-assisted binary classification, a linear combination of an input weight vector produces a single output, as shown in the following equation:

$$y = \varphi\left(\sum_{i=1}^{n} w_i \mathbf{s}_i + b\right) \tag{2.32}$$

where y is the estimated output of the activation function, w is the weight, $\mathbf{s}$ is the input vector, b is the bias, and φ is the non-linear activation function. The activation function is an essential feature of the MLP architectures. It decides whether a neuron should be fired or not by calculating the weighted sum of inputs and adding a corresponding bias to it. Sigmoid, tanh, and rectified linear unit (RELU) are examples of activation functions. RELU is the most widely used function because it is fast and less computationally expensive. MLPs use back-propagation training algorithms and the weights are updated using gradient descent to minimize the error function.

Ashrafuzzaman et al. [2] proposed different MLP structures for the detection of FDI attacks in an AC static SE system topology. The paper assumes that partial knowledge of the system, including the $\mathbf{H}$ matrix and other parameters, is known to the attacker. A standard IEEE 14 bus system is used to conduct the simulation. The Matpower toolbox is used to generate the measurement vector $\mathbf{z}$, which contains 122 measurement features (40 active and reactive power flows, 14 power injections, and 27 voltage measurements). The authors train the MLP using stochastic gradient descent (an optimization technique for the network parameters update) and tanh as the activation function. Four models with different network architectures are utilized for detection. The first model consists of one hidden layer with 100 neurons, and the second model consists of three hidden layers with 150 neurons. For the third and fourth models, the

authors use the first and second models with a regularisation value of 0.0001. Regularisation is a technique used to reduce or prevent overfitting of a neural network. The models' detection performance is compared with other machine learning algorithms. Accuracy, precision, recall, and F1 score are used to evaluate the MLP detection. The results of the four discussed models are similar with an accuracy of around 98%.

Similarly, Foroutan and Salmasi [20] applied MLPs to detect FDI attacks and compared them with common machine learning detection models. The network consists of an input layer, one hidden layer, and an output layer. Tanh is used as an activation function. Although MLP produced higher detection accuracy than the other algorithms, their training process is slow. Ganjkhani et al. [21], on the other hand, introduced a novel MLP algorithm leveraging a non-linear autoregressive exogenous (NARX) configuration, which takes into account the high correlation between power system measurements as well as the state variables. The NARX configuration is used for time series prediction and can predict step-ahead values of the states by factoring measurement values and historical data as input variables. In the experiment, NARX is constructed with an input and a hidden layer with different numbers of neurons and sigmoid linear activation functions. The historical data contained 6048 measurement vectors and state variables. The detection model is trained using 70% of the historical data and 30% for the testing and validation.

A recurrent neural network (RNN) is a sophisticated deep learning algorithm that uses internal memory or feedback loops, as shown in Figure 2.5. Unlike MLPs, RNNs use the information from past events for their predictions. A long short-term memory unit can be added to a standard RNN to solve the problem of vanishing gradient descent and store information for an extended period [27]. RNNs formulate the FDI attacks detection problem as a sequence of prediction. Results from previous time steps are used for the prediction of the current output rendering RNNs efficient in detecting manipulated measurements.

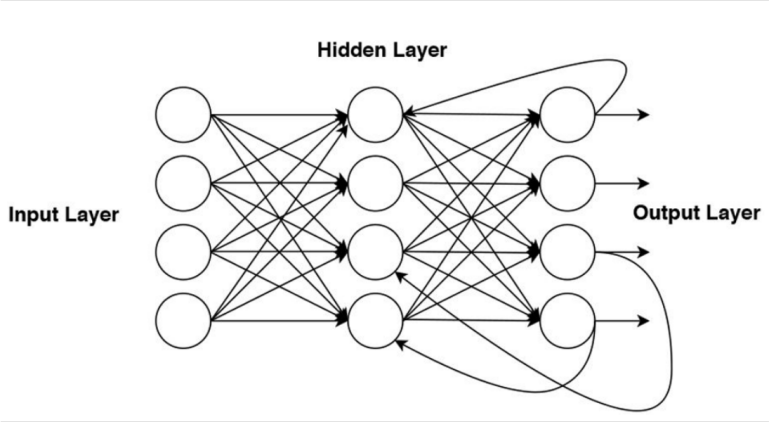

Figure 2.5: RNN concept [57].

Ayad et al. [3] utilized RNNs as a sequence classification algorithm for detecting FDI attacks in DC SE. Back-propagation through time, an extensive type of back-propagation, is applied to train the algorithm. The authors run the training algorithm multiple times to produce the optimal set of parameters that achieve the least error. Then, the optimal parameters are applied to the network for the prediction of the test data classes. Since the output ranged from 0 to 1, a threshold is set to determine the output class as either 1 or 0 (1 is compromised, and 0 is normal). The IEEE 30 bus test case is used for the experiments with 112 measurement vectors. The proposed model obtains outstanding detection results with an accuracy rate of 99%.

James et al. [31] proposed an RNN architecture for detecting FDI attacks in AC SE setups. The discrete wavelet transform (DWT) algorithm is used for the RNN model. The main goal of DWT is to extract the hidden time-frequency domain characteristics and features at every specific time. The proposed model is able to leverage dynamic temporal and spatial features for attack detection. The authors detect FDI attacks in AC SE with complete and incomplete system knowledge. For the incomplete knowledge case, the attacker only knew a few selected phase angles, power flows, and power injections for selected buses. The remaining buses' information is generated using the algebraic sum of the connecting buses. The RNN-based detection model is constructed with two types of neuron layers: gated recurrent unit and fully-connected dense layers. To tune the network hyper-parameters, a dropout approach is used. In dropout, the outputs of some layers are discarded according to predefined probabilities. Dropout solves the overfitting problem, eminent in extensive training datasets, and increases the accuracy for newly added test data. The proposed detection method's performance is assessed on the IEEE 118 bus and 300 bus test cases. Over 200k samples are generated to train the detection model, and which achieves a high detection rate of 93% [107]. However, the main challenge of this RNN type is to optimally tune the network hyper-parameters.

Deep belief networks (DBNs) consist of multiple layers of stochastic and latent variables [26]. The latent variables are generally binary variables. DBNs are compositions of simple, unsupervised networks such as restricted Boltzmann machines or autoencoders [26]. The authors in [1] utilized autoencoder networks for the detection of FDI attack leveraging temporal and spatial sensor data correlations. He et al. [24] proposed a DBN and state vector estimator (SVE) for real-time detection of FDI attacks. The proposed model utilizes an extended DBN called conditional DBN which extracts temporal features in high-dimensions. SVE calculates the ℓ_2-norm of the measurement residuals and compares them with a given threshold as follows [24]:

$$\begin{aligned} r = \|\hat{z} - H\hat{x}\|_2 > \tau, &\quad \text{Attack alarm} \\ r = \|\hat{z} - H\hat{x}\|_2 \leq \tau, &\quad \text{No attack alarm} \end{aligned} \tag{2.33}$$

The authors design the model based on the assumption that the topology of the power system does not change significantly within a small time-frame. For the simulation, the IEEE 118 bus test case is used to simulate four different attack scenarios. ROC curve is used to evaluate the detection scheme. Then, the detection results with a different number of attacked measurement k are compared with other detection algorithms such as MLPs and SVMs. The proposed model achieves the highest detection accuracy. However, training a DBN is extremely computationally expensive, since this process can take up to weeks even if specialized hardware exploiting graphics processing unit acceleration is used [56].

Wei et al. [68] proposed a different DBN-based model, where the detection process can be divided into three parts: (i) the pre-processing data stage, (ii) the training stage, and (iii) the testing stage. During the pre-processing data stage, measurement data including the attacked measurements are extracted using different IEEE standard nodes. The training process of the DBN is divided into the pretraining stage and reverse-trimming stage. In the pretraining stage, the authors use an unsupervised greedy learning algorithm from the bottom layer to the upper layer to extract the measurement features, train every layer, and share the measurement features with every layer. The Restricted Boltzmann Machine (RBM) is trained layer-by-layer and tuned using backpropagation to minimize prediction errors. After the training process, part of the measurement data is used to test and validate the performance of the detection model. The simulation results show that DBN-based detection achieves high accuracy in detecting FDI attacks (98%).

As shown in Table 2.3, the machine learning methods and related references are summarized.

Table 2.3: Summary of different references in attack detection using machine learning methods

Reference	Machine Learning Methods				Attack Type	Core Feature	Core Metrics
	Supervised	Semi-supervised	Unsupervised	Deep Learning			
[50]	Perceptron, SVM, KNN, Logistic Regression	S_3VM	K-means		FDI	Observed Measurements	Precision, Recall, Accuracy
[17]	SVM				FDI	Observed Measurements	Precision, Recall
[71]	SVM, KNN				FDI	Observed Measurements	Precision, Recall, Accuracy, F-Measure
[44]	SVM, Naive Bayes			MLP	DoS	Certain Network Parameters	Precision, Recall, Accuracy
[29]	Decision Tree, Random Forest, SVM, Naive Bayes, KNN			MLP	DoS	Certain Network Parameters	Precision, Recall, Accuracy, F-Measure, MCC
[20], [19], [35], [25]		S_3VM			FDI	Observed Measurements	Precision, Recall, Accuracy
[48]			K-means, Fuzzy c-means		FDI	Observed Measurements	Precision, Recall, F-Measure, Detection Rate
[72]			Local Outlier Factor, Isolation Forest, Robust Covariance Estimation		FDI	Observed Measurements	Precision, Accuracy
[21], [62], [2]				MLP	FDI	Observed Measurements	Precision, Accuracy
[27], [3]				RNN	FDI	Observed Measurements	Precision, Accuracy, Sensitivity, Specificity
[31]				RNN	FDI	Observed Measurements	False Alarm Rate, Detection Rate
[1], [68]	SVM			DBN	FDI	Observed Measurements	False Alarm Rate, Detection Rate
[24]				DBN	FDI	Observed Measurements	Detection Accuracy

2.6 Summary

This chapter gives a detailed review of the microgrid control systems and the related vulnerabilities and attack detection methods. Improving the cybersecurity of cyber-physical energy systems is vital for the efficient and resilient operation of the power grid. Cyber attacks can elicit severe physical and economic impacts on power systems. Researchers have thoroughly investigated FDI attacks and have proposed algorithms to detect these data integrity attacks. Among these algorithms, machine learning-based methodologies are gaining attention due to their superior detection performance.

In this chapter, we provide a comprehensive review of various cyber attack detection methods leveraging machine learning algorithms. The goal of this survey is to compare different machine learning cyber attack detectors employed in power systems. Our investigation concludes that supervised learning and deep learning methods achieve the highest detection rates. Our future work will explore how machine learning-based cyber attack detectors perform in DS which incorporate DERs (e.g. microgrids) and what modifications are essential. Also, we aim to develop detection algorithms leveraging generative adversarial networks to further improve cyber attacks detection performance against stealthy and more sophisticated attacks.

Chapter 3

Impact Analysis of Synchronization Attacks on Microgrid Systems

Contents

3.1 Introduction

A microgrid is a small-scale low-voltage electrical network, consisting of generation units, loads and storage elements, where dedicated control systems enable them to provide guaranteed power quality for local loads and have a high integration of distributed generators (DGs), including photovoltaic (PV) and wind power generation systems. The stability of the control system in microgrids is a key element to guarantee safe and reliable operations. However, the widely use of communication networks in their control systems introduces new security challenges since they are vulnerable to malicious cyber attacks. A power outage incident caused by the malware "BlackEnergy" in Ukraine during 2015 has proved that cyber attacks could cause a major damage on power quality and devices. In [63], the authors studied cyber attacks on the voltage/frequency

control systems of microgrid systems where they consider two types of attack scenario, reference signal attack and measurement routing attack, and study their impact on voltage stability and deviation in the droop-controlled DGs. Risk assessment methods to quantify the impact of measurement falsification attacks on a microgrid system are studied in [40]. To the best of our knowledge, these previous works only consider threat models targeting the primary control level of a microgrid system.

In this chapter, we consider a more comprehensive control structure for a microgrid system, mainly a hierarchical control approach consisting of three control levels: primary, secondary and tertiary control [7]. Specifically, we design a distributed cooperative hierarchical control system consisting of droop control as primary control and cooperative control as secondary control. We believe this control scheme is more realistic and reliable than the control architecture considered in [63] and [40], because of its better support for scalable integration of local loads and high integration of distributed generation.

Based on the above control model, we consider cyber attacks that may target the communication links between DGs, and we use risk assessment methods to quantitively analyze the impact of these attacks on voltage deviation at the primary control level and reference voltage synchronization at the secondary control level. Our impact analysis results are valuable for microgrid system designers to help them evaluate potential cyber threats targeting these systems.

The rest of this chapter is organized as follows. Section 3.2 describes the detailed system model of the microgrid system under study. Section 3.3 shows details about the primary and secondary controllers that are employed to the microgrid system. Section 3.4 takes a specific measurement as reference (MaR) attack as an eample, and shows how do we model the attacks and theoretically analyze the attack impacts. Section 3.5 validates the correctness of the attack impact analysis with simulation results. At last, Section 3.6 summarizes this chapter.

3.2 System Model

In this section, we describe the model of a microgrid system in terms of its hierarchical control structure and communication network.

3.2.1 Preliminaries

In this section, we review several important definitions and properties with regard to certain classes of linear time-invariant (LTI) systems that will be useful in building our system model and running further theoretical analysis. Consider a state-space represented continuous LTI system:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t). \end{cases} \tag{3.1}$$

In the LTI system (3.1), $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$ and $y(t) \in \mathbb{R}^p$ are the state vector, the input vector, and the output vector at time t, respectively. And A, B, C and D are the dynamics matrix, input matrix, output matrix and feedthrough matrix respectively. Denoting $a_{ij} = [A]_{i,j}$ as the entry of A in the i-th row and j-th column, the class of diagonally dominant matrices is defined as follows.

Definition 1 (Diagonally dominant matrices)**.** *A square matrix A is called to be row-diagonally dominant if its entries satisfy the conditions*

$$|a_{ii}| \geq \sum_{j \neq i} |a_{ij}|, \forall i \in \{1, \ldots, n\}. \tag{3.2}$$

Given Definition 1, the system (3.1) is called to be row-diagonally dominant if the dynamics matrix A is row-diagonally dominant.

Besides row-diagonally dominant systems, another important class of systems throughout this paper is that of positive systems. Next we describe the definition and properties of positive systems.

Definition 2 (Positive systems)**.** *The LTI system* (3.1) *is said to be (internally) positive if and only if its state $x(t)$ and output $y(t)$ are non-negative for every non-negative input $u(t)$ and every non-negative initial state $x(0)$.*

Lemma 1 (Positivity)**.** *The LTI system* (3.1) *is positive if and only if A is a Metzler-matrix, i.e., it has non-negative off-diagonal entries, and B, C and D are non-negative, i.e., they only have non-negative entries.*

Lemma 2 ([53])**.** *If the system* (3.1) *is positive, the following statements are equivalent:*

1) the matrix A is Hurwitz, i.e., every eigenvalue of A has strictly negative real part).

2) There exists a $\xi \in \mathbb{R}^n$ such that $\xi > 0$ and $A\xi < 0$.

3) The matrix $-A^{-1}$ exists and has nonnegative entries.

These first two statements give a scalable stability verification and relate the stability of positive systems to a set of inequality constraints, which is helpful to derive stability conditions. The third statement contributes to the voltage magnitude deviation analysis for systems under cyber attacks.

3.2.2 Microgrid System with Hierarchical Control Structure

Microgrids are designed to work in both grid-connected and islanded operating modes, depending on whether connecting or not to the main grid. The task of the microgrid control system is to regulate voltage and frequency for different operating modes, to achieve proper load sharing among DGs, to control the power flow between the main grid and the microgrid, and to optimize the cost of its operations.

Currently, a hierarchical control structure is designed to achieve the above operating goals [7]. As shown in Figure 3.1, the hierarchical control structure consists of primary, secondary and tertiary control levels. Each control level has its own control goals because of operating in different timescales. The primary control operates on a fast timescale, and is responsible for the control of transients to stabilize the voltage and frequency of the microgrid during changing of load or generation, or subsequent to an islanding event. Secondary control is designed to compensate for voltage and frequency deviations caused by primary control in terms of fault conditions. Finally, tertiary control, as the highest and slowest control level, optimizes the operations in both operating modes and manages to control the power flow between the main grid and the microgrid.

In this chapter, we mainly consider the primary and secondary control levels of microgrid with balanced loads. The primary control is locally implemented at each DG, e.g., droop control for

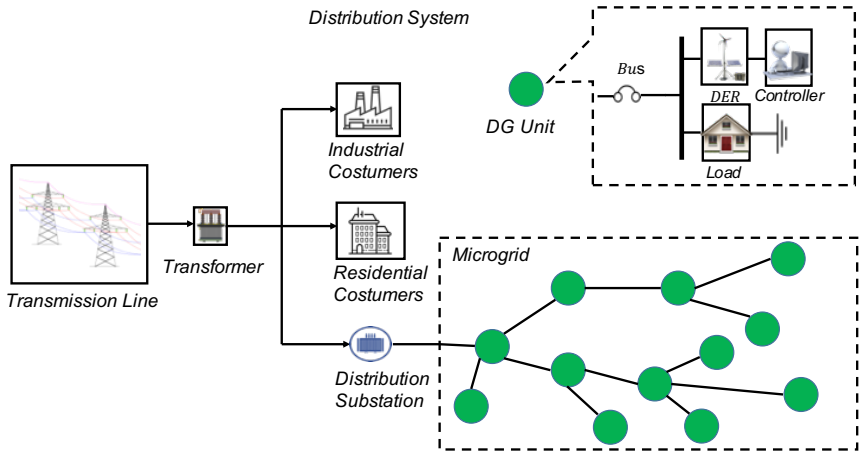

Figure 3.1: Microgrid with a communication network. As a part of the power distribution network, microgrids are at the downstream of distribution substations. Sparse communication networks are employed for data exchange between neighboring DG units, e.g., transmitting reference signals and measurements.

inverter-based distributed energy resources (DER). However, the secondary control usually relies on a centralized control structure. Central controllers are designed to issue global commands requiring information gathered from the whole system and thus a complex two-way communication network is also needed. This kind of communication network makes it vulnerable and may effect the system reliability. In [7], the authors replace the existing standard centralized secondary control with an efficient distributed control structure. They consider the microgrid as a multi-agent system where each DG is an agent. A voltage source inverter (VSI) is employed to connect a DG to the microgrid. The VSIs are interconnected through the physical power network configuration.

A sparse communication network, which overlays the physical power network, is integrated in the control system. The controllers use this communication network to only communicate in a distributed way with neighboring nodes. The secondary controller also uses cooperative multi-agent control techniques to make all agents act as a one group to a common synchronization goal and follow cooperative decisions.

In this chapter, we consider a distributed cooperative control structure similar to [7], but design our own primary and secondary control loops emulating real microgrid systems. The microgrid with N DG units is depicted in Figure 1. Each DG unit is represented by a bus. The lumped DERs and loads within one DG unit are respectively modeled as a single DER and load. We set the DG unit directly connected to main grid as DG Unit_1.

Assumption 1. *In the power distribution system under study, we make the following assumptions:*

1) The system has balanced three-phase power network, i.e., it can be represented as an equivalent single-phase system;

2) All N buses are inverter-based, and represented by V_i and θ_i for $i = 1, \ldots, N$.

Let R_{ij} and X_{ij} be resistance and reactance of the transmission line between bus i and bus j, respectively, thus under Assumption 1, the active and reactive power injections at bus i is

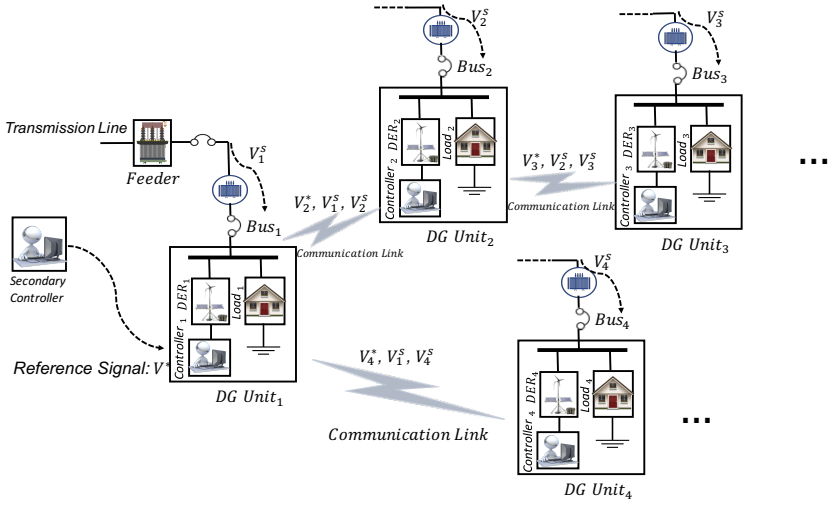

Figure 3.2: A microgrid control system with primary and secondary controllers. Sparse communication networks are employed for data exchange between neighboring DG units, e.g., transmitting reference signals and measurements (denoted by the superscript s).

given respectively by

$$\begin{aligned} P_i &= V_i^2 G_i - \sum_{j \in N_i} V_i V_j (G_{ij} \cos(\theta_{ij}) + B_{ij} \sin(\theta ij)), \\ Q_i &= -V_i^2 B_i - \sum_{j \in N_i} V_i V_j (G_{ij} \sin(\theta_{ij}) - B_{ij} \cos(\theta ij)), \end{aligned} \tag{3.3}$$

in which $G_{ij} = R_{ij}/(R_{ij}^2 + X_{ij}^2) \geq 0$ and $B_{ij} = -X_{ij}/(R_{ij}^2 + X_{ij}^2) \leq 0$ are, respectively, the conductance and susceptance of the transmission line between bus i and bus j. Additionally, we define self-conductance and self-susceptance as $G_i = G_{ii} + \sum_{j \in N_i} G_{ij} \geq 0$ and $B_i = B_{ii} + \sum_{j \in N_i} B_{ij} \leq 0$, respectively. Note that we use $\theta_{ij} = \theta_i - \theta_j$ to represent the angle difference between node i and j in the remainder of this paper.

Assumption 2. *In the power distribution system under study, we assume the transmission line impedances have the same ratio $R_{ij}/X_{ij} = -G_{ij}/B_{ij} = \rho \geq 0$ for all lines $(i,j) \in \mathcal{E}$.*

The line ratio reflects the inherent property of the power system. A small ratio $\rho > 0$ represents power systems with inductive transmission lines ($R_{ij} \ll X_{ij}$), while higher ratio ρ means the transmission lines are resistive, which is often the case for medium and low voltage distribution systems. We can draw the conclusion that, for systems with homogeneous transmission lines that have similar characteristics, Assumption 2 naturally holds because the line ratio ρ depends on the transmission line characteristics. Moreover, Assumption 2 is also commonly employed in the case of purely inductive lines ($\rho = 0$) [58, 60].

3.2.3 Communication Network

In the microgrid system, DG units are considered as the nodes of the communication graph and the corresponding communication links represent its edges. Figure 1 describes a line network, however a generic network topology of a microgrid system is usually characterized by a directed graph (digraph) $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A_G)$ with a nonempty finite set of N nodes $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$, a set

of edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, and the associated adjacency matrix A_G. The set $\mathcal{N}_i = \{j \in \mathcal{V} : (i,j) \in \mathcal{E}\}$ denotes the neighbor set of node i. The weight is represented by a_{ij} for the edge from node j to node i, and $a_{ij} = 0$ if there is no data transfer from node j to node i. Considering N nodes on the graph, the adjacency matrix is defined as $A_G = [a_{ij}] \in \mathbb{R}^{N \times N}$.

We define the diagonal in-degree matrix $D = diag\{d_i\} \in \mathbb{R}^{N \times N}$, where d_i is the sum of communication weights from neighbors of node i, $d_i = \sum_{j \in \mathcal{N}_i} a_{ij}$.

The Laplacian matrix of the graph is defined as $L = D - A_G$. One property of the Laplacian matrix L is that the row sums of L are all zero, because the row sums of D and A_G are equal.

A directed path from node i to node j is a sequence of edges depicted as $\{(v_i, v_k), (v_k, v_l), \ldots, (v_m, v_j)\}$. A graph is considered to have a *spanning tree* if there exists a *root node* with a directed path from that node to every other node in this graph. A graph is *strongly connected* if there exists a directed path between every two nodes, i.e., there exists a spanning tree where every node is a root node [7] [46].

3.3 Distributed Cooperative Control system

In this section, we rely on existing control models to propose a distributed cooperative control scheme for voltage dynamics. We give a detailed description of both the primary and secondary control of the microgrid system under study.

3.3.1 Primary Control

Figure 1 illustrates how the reference signals and measurements are available to each controller. Making use of the capabilities of inverter-based DERs, each DG unit is controlled by a droop controller. The droop controller receives reference signal through synchronization process between neighboring nodes and voltage measurements from local meters.

For each DG unit, the voltage and phase-angle dynamics can be respectively modeled by a single integrator

$$\begin{aligned} \tau_i \dot{V}_i(t) &= u_{v_i}(t), \\ \tau_{\theta_i} \dot{\theta}_i(t) &= u_{\theta_i}(t), \end{aligned} \tag{3.4}$$

where $\tau_i > 0$ and $\tau_{\theta_i} > 0$ are the inverter's time-constants and $u_{V_i(t)}$ and $u_{\theta_i(t)}$ are the control signals computed by the droop controller at time $t \geq 0$. As illustrated in Fig. 1, the measurements and reference signals are available to each controller from the architecture of the control system. Each DG unit is controlled by a droop controller based on the capabilities of the local inverter-based DERs. Each controller receives the reference signal computed remotely and measurements through the communication network. Let V_i^* be the reference voltage for the i-th bus and V_j and θ_j, be the voltage magnitude and voltage angle of the j-th bus, respectively. A suitable communication protocol is needed for the transmission of these data, e.g., IEC 61850.

In this chapter, we are mainly interested in the voltage dynamics of the power distribution system. So we consider the following assumption and neglect the phase-angle dynamics throughout the rest of the chapter.

Assumption 3. *The phase-angle difference θ_{ij} between any neighboring nodes i and j is assumed to be constant.*

Note that Assumption 3 could be employed in a local analysis when the phase-angles remain in the neighborhood of the original equilibrium point. In addition, we underline that the assumption is valid if there exists a time-scale separation between the voltage dynamics and the phase-angle.

In order to compute the control output signals, we refer to the voltage quadratic droop controller [60] described by

$$u_{V_i}(t) = -\kappa_i V_i^c(t)(V_i^c(t) - V_i^{c*}(t)) - Q_i^c(t), \tag{3.5}$$

where $\kappa_i > 0$ is the control gain of the droop controller. Additionally, $V_i^c(t)$, $V_i^{c*}(t)$ and $Q_i^c(t)$ respectively represent the voltage measurement, voltage reference signal with respect to bus i and reactive power injection measurement. They are received by the droop controller, as illustrated in Fig. 1. Under nominal operation, these signals match the corresponding physical variables and reference signals, i.e., $V_i^c(t) = V_i(t)$, $Q_i^c(t) = Q_i(t)$, and $V_i^{c*}(t) = V_i^*(t)$ ($V_i^*(t)$ is sent by a higher level controller from the substation). The closed-loop dynamics of the i-th DG unit under nominal operation are described by the differential equations

$$\begin{aligned} \tau_i \dot{V}_i &= -\kappa_i V_i(V_i - V_i^*) - Q_i \\ &= -V_i(\kappa_i V_i - \kappa_i V_i^* + \sum_{j\in\mathcal{V}} l_{ij}(\theta) V_j), \forall i = 1, \dots, N, \end{aligned} \tag{3.6}$$

with the time argument omitted. Additionally, under the Assumption 2, the parameter $l_{ij}(\theta)$ is described as

$$l_{ij}(\theta) = \begin{cases} B_{ij}(\rho \sin(\theta_{ij}) + \cos(\theta_{ij})), & i \neq j \\ -B_i, & i = j. \end{cases} \tag{3.7}$$

Denoting $V = [V_1 \dots V_N]^\top$, $\tau = [\tau_1 \dots \tau_N]^\top$, $\kappa = [\kappa_1 \dots \kappa_N]^\top$, and $[V]$ as the diagonal matrix with V_i as the i-th diagonal entry, we can get the voltage dynamics under the quadratic droop control in vector form:

$$[\tau]\dot{V} = [V]([\kappa]V^* - ([\kappa] + L(\theta))V), \tag{3.8}$$

where the matrix $L(\theta)$ is defined as $[L(\theta)]_{ij} = l_{ij}(\theta)$.

Linearization of the voltage dynamics. For convenience of theoretic analysis of system stability and attack impact, we consider the *Jacobian linearization* of the power system (3.8) around an equilibrium point $(\bar{V}, \bar{V}^{c*})$ such that $-([\kappa] + L(\theta))\bar{V} + [\kappa]\bar{V}^* = 0$. Denote $x(t) = V(t) - V$ and $u(t) = V^{c*}(t) - \bar{V}^{c*}$ as the voltage and reference deviations, respectively. The corresponding linearized system is described by

$$\dot{x}(t) = Ax(t) + Bu(t), \tag{3.9}$$

where $A = -[\bar{V}][\tau]^{-1}([\kappa] + L(\theta))$ and $B = [\bar{V}][\tau]^{-1}[\kappa]$. For the sake of simplicity, we suppose that $\bar{V} = \mathbf{1}pu$ subsequently, where $\mathbf{1}$ represents a vector with all entries equal to 1.

System properties

In this subsubsection, necessary and sufficient conditions for the linearized power system to be positive are elaborated. These conclusions will play an important role in the impact analysis for the power system under attack in subsequent sections.

Firstly, the following assumption is required to derive necessary and sufficient conditions for the linearized system (3.9) to be positive.

Assumption 4. *The maximum phase difference between any two neighboring nodes, defined as* $\Delta_\theta := \max_{(i,j)\in\mathcal{E}} |\theta_{ij}|$*, satisfies the inequality* $\Delta_\theta < \pi/2$.

For any conventional power system, the constraint $\Delta_\theta < \pi/2$ is required for the stability of the phase-angle dynamics [58]. Under the previous assumptions, we establish the following result for system positivity.

Lemma 3 ([63])**.** *Consider the power distribution system under study, having active and reactive power injections* (3.3) *at bus* i *with* $\Delta_\theta < \pi/2$*, and applying the quadratic droop controller* (3.6) *for each DG unit. Then a necessary and sufficient condition for the corresponding linearized system* (3.9) *to be positive is*

$$\rho \leq |\cot(\Delta_\theta)| \,. \tag{3.10}$$

Note that several of the properties of positive systems stated in Subsection 3.2.1 have important consequences in the context of power systems and, in particular, the voltage dynamics. Letting the input $u(t)$ be the voltage reference at one individual bus and $x(t)$ and the voltages of all buses, the closed-loop system being positive implies that an increase in the voltage reference $u(t)$ translates to an increase in all the voltages $x(t)$. Hence, there is no contradictory effect where a desired increase in voltage at one bus inadvertently decreases the voltage in other buses. Additionally, positivity of the closed-loop system also reduces the voltage overshoots in response to step changes in the voltage reference.

Remark 1. *While the latter discussion motivates positivity as a desirable system feature, Lemma 2 provides a design objective for the phase-angle controller that ensures positivity of the voltage dynamics, namely the inequality* (3.10)*. In fact, since the line ratio* ρ *is a system parameter that depends solely on the transmission lines' characteristics,* (3.10) *can be interpreted as a bound on the phase-angle differences that is parameterized by the line ratio* ρ*. Rewriting* (3.10) *as* $|\tan(\Delta_\theta)| \leq \rho^{-1}$*, we have that a resistive system with a large* ρ *yields a strict bound on the maximum phase-angle difference* Δ_θ*, while a purely inductive system with* $\rho = 0$ *does not constrain the phase-angle difference.*

Lemma 4 ([63])**.** *Suppose the linearized system* (3.9) *is positive. The system* (3.9) *is row-diagonally dominant if, and only if, the following inequality holds*

$$\kappa_i + |B_{ii}| \geq (\sqrt{\rho^2 + 1} - 1) \sum_{j\in\mathcal{N}_i} |B_{ij}| \,. \tag{3.11}$$

Theorem 2. *Consider the linearized dynamics of the power system* (3.9)*) and suppose the system is positive. Then the following statements hold:*

1) *the system is asymptotically stable if and only if there exist positive scalars* $\xi > 0$ *such that the following inequality holds for all* $i = 1, \ldots, n$*;*

$$\xi_i \, |-\kappa_i + B_i| > \sum_{j\in\mathcal{N}_i} \xi_j \, |-B_{ij}(\rho \sin(\theta_{ij}) + \cos(\theta_{ij}))| \,;$$

2) *the system is asymptotically stable if it is row-diagonally dominant, i.e., the following inequality holds for all* $i = 1, \ldots, n$*:*

$$|-\kappa_i + B_i| > \sum_{j\in\mathcal{N}_i} |-B_{ij}(\rho \sin(\theta_{ij}) + \cos(\theta_{ij}))| \,;$$

3.3.2 Secondary Control

In this work, the goal of the secondary controller is to generate the voltage reference signal for the primary controller at each DG unit. We employ the distributed cooperative control of multi-agent systems [7] to design the secondary voltage controller for our microgrid system.

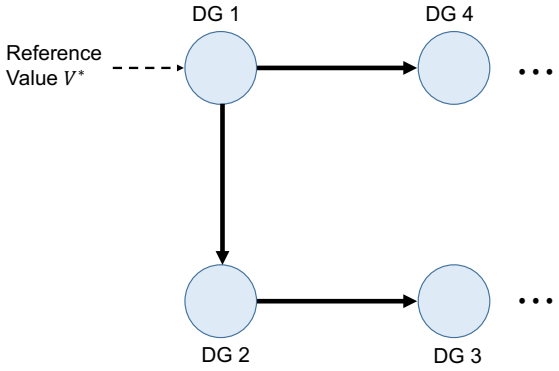

Figure 3.3: The topology of the communication graph $\mathcal{G}$ of a microgrid system.

We simplify the microgrid system to show the topology of the system communication graph $\mathcal{G}$ in Fig. 3.3.

DGs are assumed to be able to communicate with each other through the communication network $\mathcal{G}$. We assume that only one *leader* DG node i has access to the reference V^* by a weight factor called the *pinning gain* g_i. The pinning matrix G is defined to carry all the pinning gains of the graph:

$$G = diag\{g_i\} \in \mathbb{R}^{N\times N}. \tag{3.12}$$

The computation of the global reference voltage V^* is based on the load deferences between loads in this microgrid and its neighbors, as specified in [46].

The synchronization of reference values of DGs in a communication network $\mathcal{G}$ can be modeled as a *tracking problem* of multi-agent cooperative control systems [7]. The goal of a cooperative tracking problem is to have consensus for the agent variables. However, unlike the synchronization problem, the consensus value is a control objective usually provided by the leader. One can state the tracking problem as $\mathbf{x} = \mathbf{1} \cdot \mathbf{x}_{ref}$, where $\mathbf{x}_{ref}$ is the reference value determined by the leader. The leader is connected to a limited number of agents. A pinned graph is used to model the tracking problem. In Fig. 3.3, the leader node is set to be DG 1 and the consensus is the reference value V^*. To reach the consensus at the reference value, the graph should contain at least one pinned root node. Then, the control agents cooperate to hold

$$\dot{\mathbf{x}} = -(L + G)(\mathbf{x} - \mathbf{1} \cdot \mathbf{x}_{ref}) \tag{3.13}$$

at the steady state. Since all eigenvalues of $L+G$ have positive real values, equation (3.13) results in the desired tracking performance at the steady state. It is noteworthy that the reference value can also have dynamics. However, to have acceptable tracking performance for all agents, the dynamics of the tracking equation (3.13)should be much faster than those of the reference.

We assume that only one *leader* node i acting as the command generator has access to the reference V_{ref} with pinning gain g_i, and all other nodes synchronize to node i. The value

V_{ref} can be determined by keeping a critical load operating at the nominal voltage $V_{nominal}$ [9]. Hence the secondary controller computes V^* as

$$V_{ref} = \kappa_p(V_{nominal} - V_{critical}) + \kappa_i \int (V_{nominal} - V_{critical})dt \tag{3.14}$$

where $V_{critical}$ is the critical load voltage and κ_p and κ_i are the controller gains.

Note the voltage reference vector $V^* = [V_1^* \dots V_N^*]^\top$. Consider the tracking problem in equation (3.13) and we set $\mathbf{x} = V^*$ in the secondary control problem. Then the reference synchronisation process of all DG units follows

$$\dot{V}^* = -(L + G)(V^* - \mathbf{1} \cdot V_{ref}) \tag{3.15}$$

at the steady state. Equation (3.15) leads to the desired tracking performance of each DG unit at the steady state. Note the adjacency matrix of the communication network $\mathcal{G}$ is $A_G = [a_{ij}] \in \mathbb{R}^{N \times N}$. The voltage reference setpoint of each DG unit can be written as

$$\dot{V}_i^* = \sum_{j \in \mathcal{N}_i} a_{ij}(V_j^* - V_i^*) + g_i(V_{ref} - V_i^*). \tag{3.16}$$

3.4 Cyber attacks on voltage control and Impact Analysis

In this section we consider the model of the microgrid system described in section 3.2 and we study the possible adversary actions on the reference signal transmitted by the communication network at the primary and secondary control levels.

It is natural to consider a naive attack where an attacker simply falsifies the reference voltage value V_i^* for DG i. However, it is easy to be detected by simple detection algorithms [30] considering the admissible range

$$V_{min}^* \leq V_i^* \leq V_{max}^*, \tag{3.17}$$

where V_{min}^* and V_{max}^* are respectively the lower and upper limit for the reference value.

In this work, we identify a novel type of of attack scenarios: "measurement as reference " attack. Under the assumption that the attacker has knowledge about the hierarchical control structure, it targets the communication link between DG units and maliciously replaces the reference signal with the measurement of the previous node. Since the two signals have very close values and follow the same dynamic changes, this attack is "naturally stealthy", and is possible to cause a serious impact without being detected by traditional intrusion detection algorithms.

First we describe the considered attack scenario and we give its mathematical definition. Then we describe how the attack influences the voltage control systems. We characterize the attack impact by the maximum voltage magnitude deviation and inaccurate reference voltage synchronization it causes.

3.4.1 Measurement as reference attack

Without loss of generality, we assume the attacker replaces the reference signal V_{i+1}^* for node $i+1$ with the voltage measurement V_i^s of node i when defining the attack.

The goal of measurement as reference attack is set to cause voltage fluctuations and irregular regulations to harm the loads without violating the admissible range (3.17). For convenience of analysis, we assume only one communication link is under attack. The definition of measurement as reference attack is given as follows.

Definition 3 (measurement as reference attack)**.** *In a measurement as reference attack on the communication link between DG unit i and $i+1$, the attacker manipulates the exchanged data by replacing the reference signal $V^*_{i+1}(t)$ for DG unit $i+1$ with the voltage measurement $V^s_i(t)$ of DG unit i , so that*

$$V^*_{i+1}(t) = V^s_i(t), \tag{3.18}$$

Furthermore, the droop control signal at DG unit $i+1$ under attack is given by

$$\tau_i \dot{V}_{i+1}(t) = -\kappa_{i+1} V^c_{i+1}(t)(V^c_{i+1}(t) - V^s_i(t)) - Q^c_{i+1}(t) \tag{3.19}$$

3.4.2 Attack Impact on Voltage Deviation

One impact of the measurement as reference attack is voltage magnitude deviations . Consider the resulting changes to the voltage magnitude at DG j in the network, i.e., V_j. The resulting linearized system under attack can be written as

$$\begin{aligned} \dot{x}(t) &= Ax(t) + \tau^{-1}_{i+1}\kappa_{i+1} e_{i+1} u(t) \\ y_j(t) &= e_j^\top x(t) \\ u(t) &= e_i^\top x(t), \end{aligned} \tag{3.20}$$

where $A = -[\tau]^{-1}(\kappa + L(\theta))$ and $e_i \in \mathbb{R}^N$ is the i-th column of the N-dimensional identify matrix. The attack impact can be quantified as the maximum deviation of $y_j(t)$ caused by a corrupted input $u(t) = e_i^\top x(t)$, which is bounded by $|u(t)| \le \delta$, where $V^*_{min} \le \delta \le V^*_{max}$.

Computation of maximum voltage deviation. We use $\sup_{t\ge 0} |y_j(t)|$ to represent the maximum voltage magnitude of DG j under a measurement as reference attack (3.18) on the link between DG i and $i+1$. We know that the linearized system (3.20) under attack is a positive system [53] since A is a *Metzler* matrix with non-negative off-diagonal entries and $\tau^{-1}_{i+1}\kappa_{i+1}e_{i+1}$, $e_j^\top$ are non-negative [63]. Let's define the transfer function of the system (3.20) as $H(s) = Y(s)/U(s) = \mathcal{L}\{y(t)\}/\mathcal{L}\{u(t)\}$, where $\mathcal{L}$ means Laplace transform. Thus we can compute

$$H(s) = \tau^{-1}_{i+1}\kappa_{i+1} e_j^\top (sI - A)^{-1} e_{i+1}. \tag{3.21}$$

According to the properties of positive systems [53], we know that the $\mathcal{L}_\infty$-induced norm of system (3.20) is given as

$$\|H\|_{\mathcal{L}_\infty - ind} = \sup_{\|u\|_{\mathcal{L}_\infty} < \infty} \frac{\|y\|_{\mathcal{L}_\infty}}{\|u\|_{\mathcal{L}_\infty}} = H(0), \tag{3.22}$$

which characterizes the maximum amplitude of the output signal $y_j(t)$ which can be achieved by an input $u(t)$ with bounded amplitude

$$\|u\|_{\mathcal{L}_\infty} := \sup_{t\ge 0} |u(t)| \le \delta. \tag{3.23}$$

So we can compute

$$\begin{aligned} \sup_{t\ge 0} |y_j(t)| &= \|H\|_{\mathcal{L}_\infty - ind} \cdot \|u\|_{\mathcal{L}_\infty} \\ &\le H(0) \cdot \delta \\ &= -\tau^{-1}_{i+1}\kappa_{i+1}\delta e_j^\top A^{-1} e_{i+1} \\ &= \tau^{-1}_{i+1}\kappa_{i+1}\delta [-A^{-1}]_{j,i+1}, \end{aligned} \tag{3.24}$$

where $[-A^{-1}]_{j,i+1}$ is the element of row j column $i{+}1$ from matrix $-A^{-1}$. So $\tau_{i+1}^{-1}\kappa_{i+1}\delta[-A^{-1}]_{j,i+1}$ gives the upper bound of the voltage deviation caused by the attack. Since the δ value in (3.23) is determined by the system state, the upper bound in (3.24) is tight because the maximum is reachable. Therefore, we find that the impact on voltage deviation of each node depends on the structure of matrix A, which reflects the power system topology.

3.4.3 Impact on Reference Voltage Synchronization

Besides the voltage deviation, the attack also affects the reference synchronization regarding the secondary control level. Note that the reference of each DG unit will follow the differential equation (3.15) under nominal operations. So the problem of studying what is the attack impact on reference synchronization is mapped to seeking for the initial conditions of differential equation (3.15) caused by the attack.

Since the primary control is operating much faster than the secondary control, the voltage state of each DG node would become stable again very quickly and reaches a new equilibrium point when the secondary controller regulates the reference signals. To compute this equilibrium point, we let $\dot{V} = 0$ in (3.8), i.e., $[\kappa]\, V_i^* - ([\kappa] + L(\theta))\, V = 0$. According to the definition of measurement as reference attack $V_{i+1}^*(t) = V_i^s(t)$, we have

$$V_{i+1}^* = V_i = e_i^\top ([\kappa] + L(\theta))^{-1} [\kappa]\, V_{ref}. \tag{3.25}$$

So we can get the initial conditions of differential equation (3.15) under attack as

$$(e_i^\top ([\kappa] + L(\theta))^{-1} [\kappa] - e_{i+1}^\top)\, V_{ref} = 0. \tag{3.26}$$

The differential equation (3.25) with initial condition (3.26) can be written as

$$\begin{cases} \dot{V}_{ref} = -(L+G)(V_{ref} - \mathbf{1} \cdot V^*) \\ (e_i^\top ([\kappa] + L(\theta))^{-1} [\kappa] - e_{i+1}^\top)\, V_{ref} = 0. \end{cases} \tag{3.27}$$

Equation (3.27) describes the reference signal synchronization process under attack for each node. When the secondary controller regulates the set point V^*, all references of the nodes relying on communication with node $i{+}1$ for synchronization, are not able to reach the consensus V^*.

3.5 Simulation System and Validation

In this section, we use simulation tools to validate the theoretical analysis of the measurement as reference attack and its impact proposed in the previous section.

3.5.1 Simulation Settings

We use Matlab and Simulink tools to build the microgrid control system described in Figure 1 and model the measurement as reference attack provided in Definition 3.

As shown in Fig. 3.4, we build a simulink model with different modules to simulate the distributed and cooperative controlled microgrid system. The module *microgrid* contains derivative blocks to simulate the primary and secondary controllers as described in Section 3.3. The module *x_out* represents the observed state vector, i.e., the DG voltage measured by each sensor at the related DG unit. The module *y_out* represents the output vector. The signals *V_ref*,

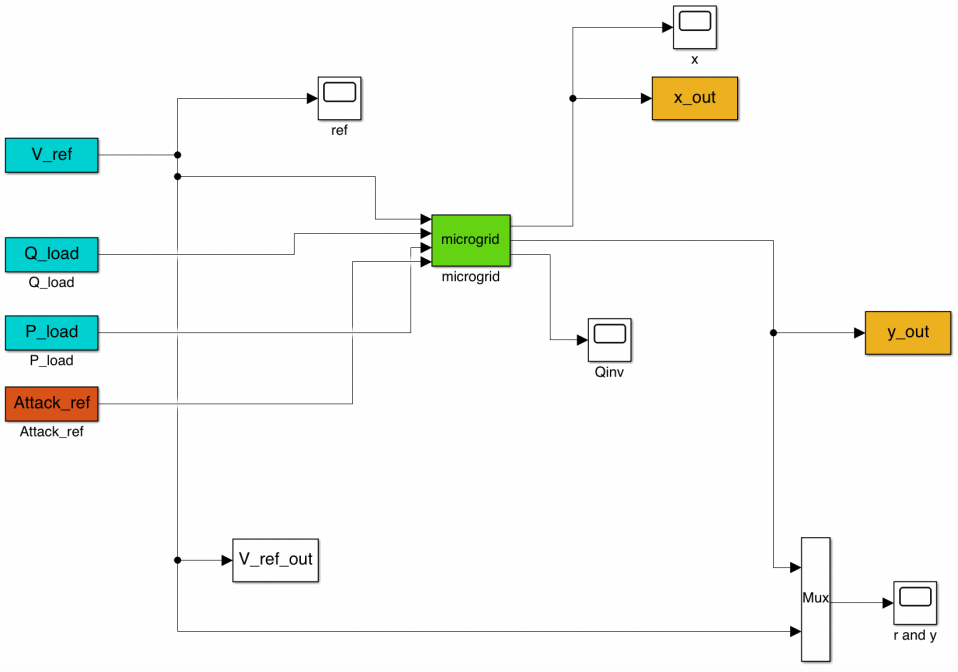

Figure 3.4: The simulink model of the distributed and cooperative controlled microgrid system.

Q_load, *P_load* and *Attack_ref* represent the voltage reference signal, injected reactive power, injected positive power and attack signal, respectively.

For the power system simulation settings, we consider a microgrid system with 4 DG units as shown in Figure 1, i.e., $N = 4$. We follow the model settings proposed in [63] and [40], where all power lines, DERs and loads are identical. The power system is characterized by the equation (3.3) with parameters: $\rho = 0.5$, $B_{ij} = -0.2$, and $G_{ij} = -\rho B_{ij}$ for all edges $(i, j) \in \mathcal{E}$ and $B_{ij} = -0.2$ and $G_{ii} = -\rho |B_{ii}|$ for all buses.

For the primary control, we set the parameters $\theta_{12} = -0.01$rad, $\theta_{23} = -0.045$rad, and $\theta_{34} = -0.01$rad, to make sure that the phase-angle differences between any neighboring nodes are constant. The quadratic droop control modeled by equations (3.8) and (3.5) is characterized by parameters $\tau_i = 10^{-4}$, $\tau_{\theta_i} = 10^{-2}$, and $\kappa_i = 0.2$ for all buses.

The voltage dynamics under primary control are defined by the nonlinear differential equations (3.8). Through Jacobian linearization, the corresponding linearized dynamics characterized by equation (3.9) is given as

$$A = 10^{-4} \cdot \begin{bmatrix} -4.01 & 1.88 & 0 & 0 \\ 2.1 & -6.01 & 2.04 & 0 \\ 0 & 1.95 & -6.01 & 1.88 \\ 0 & 0 & 2.1 & -4.01 \end{bmatrix}.$$

Clearly the above system is positive and the properties of positive systems [53] are applicable.

Finally, we choose typical parameters for the secondary control. As depicted in Figure 1, the network is a 4-node line topology and DG 1 is the only pinned root node. Considering the communication network topology and a proper synchronization rate, we set the adjacency matrix $A_G = [a_{ij}]$ carrying the communication weights of $a_{21} = 12$, $a_{32} = 11$, $a_{41} = 10$ and all other weights equal to zero, and the pinning matrix $G = diag\{g_i\}$ carrying all pinning gains of $g_1 = 10$, $g_2 = g_3 = g_4 = 0$.

3.5.2 Simulation Results

Considering the above simulation settings, the voltage of each DG is in stable state and the global reference set point $V^* = 1\text{pu}$. At time $t = 2 \times 10^{-3}\text{s}$, the attacker introduces the measurement as reference attack by replacing the reference signal V_2^* at node 2 with the voltage measurement V_1 at node 1.

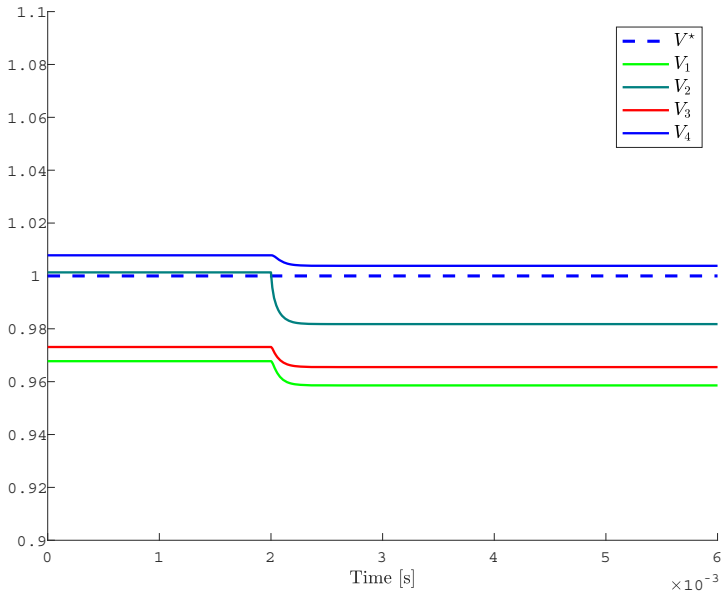

Figure 3.5: The impact of the measurement as reference attack on voltage deviation.

As shown in Figure 3.5, the attack cause voltage deviation at each DG unit, which is a short-term impact on the primary control. We can see the voltages reach a stable state again very quickly. Comparing the voltage deviation of the simulated non-linear system under attack with the linearization analysis in Section 3.4, we can find that the voltage deviation caused by the attack is within boundary computed in (3.24). Comparison result shows that the voltage deviation of the simulated non-linear system under attack is the same to the linearized system, which shows our linearization analysis in Section 3.4 is valid.

How the attack influences the reference synchronization is shown in Figure 2. At time $t = 0.1\text{s}$, the secondary controller regulates the reference set point from $V^* = 1\text{pu}$ to $V^* = 1.05\text{pu}$. After about 0.5s, the reference signals finish the synchronization process. We find that DG 2 and DG 3 no longer reach the consensus V^*, which could lead to a heavy impact on voltage regulation in a microgrid system.

3.6 Summary

In this chapter we present the design of a novel attack named Measurement as Reference (MaR) attack targeting the distributed control system of microgrid. In this attack that we instantiated on voltage control, the attacker introduces a reference voltage deviation by manipulating the synchronization data exchanged between the distributed controllers. We show the impact of this synchronisation attack by simulation and theoretical analysis. We use control-theoretic

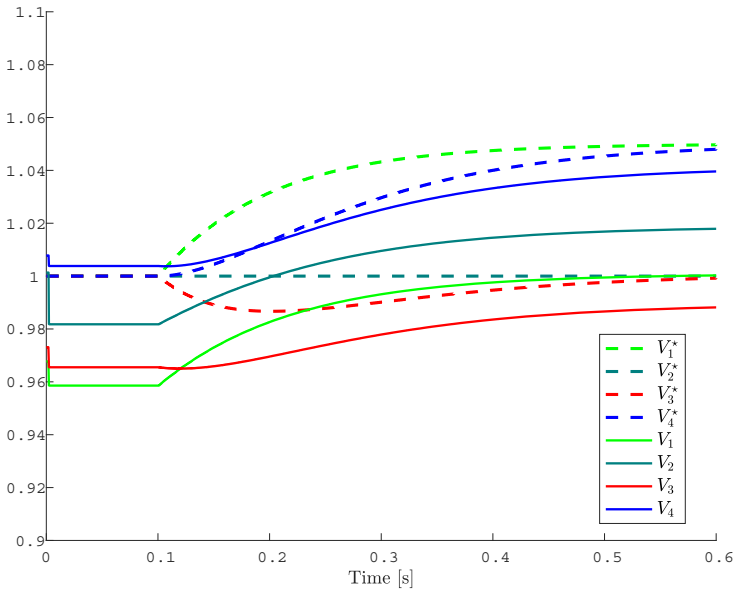

Figure 3.6: The impact of the measurement as reference attack on reference synchronization.

tools to derive the maximum voltage deviation introduced by this attack where the system will not synchronize to the correct setting point provided by the secondary controller.

However, a distributed control system is also subject to other attacks including delay injection and denial of service attacks, that also severely impact the operations of this system. The detection of the attacks is still challenging because of the stealthy nature of some of them where the deviations from nominal values are very low. This challenge is the main motivation of the following chapter that proposes a more robust method based on Machine Learning algorithms to detect these attacks including the MaR attack.

Chapter 4

A Framework for Secure Control in Microgrids

Contents

4.1 Introduction

Several existing studies highlight the requirements for a comprehensive attack resilient control system for microgrids that simultaneously: (i) expedites the proper voltage stability and regulation; (ii) handles attacks on the distributed cooperative control systems; (iii) deals with both FDI and DoS attacks; (iv) is applicable to resource-limited small-scaled distributed controllers.

To this end, this chapter proposes a resilient distributed cooperative control for microgrids that accommodates FDI and DoS attacks detection. Compared to the existing attack detection methods based on state estimation [39, 35], statistical machine learning algorithms outperform them in detecting both observable and unobservable attacks [50]. The attack detection problem can be modeled as a typical classification problem to learn communication patterns at the network-level of the distributed control system and predict the class of a packet as attack or

normal . We study various groups of supervised machine learning algorithms to design a comprehensive attack resilient control framework, in particular to detect FDI and DoS attacks in microgrids. In this chapter, we make the following contributions [42]:

- We model and generalize both FDI and DoS attacks in distributed and cooperative control microgrid systems. We also give a general system stability condition analysis under FDI and DoS attacks.
- We propose a secure control framework with attack detection and response modules relying on machine learning and deep learning algorithms. Various groups of supervised machine learning and deep learning algorithms are studied to design the attack detection module against FDI attacks and DoS attacks under well-chosen performance metrics.
- Considering it is difficult to find publicly available datasets, we build an experimental platform emulating the distributed and cooperative controlled microgrid systems, and implement two typical FDI and DoS attacks hard to be detected by traditional residual comparison detectors. The two attacks are: measurement as referent (MaR) attack as typical FDI attack and delay injection attack as typical DoS attack. Balanced and imbalanced dataset are collected to under normal and attacked conditions to be further analyzed.

The rest of this chapter is organized as follows. Section 4.2 gives a general modelling of FDI and DoS attacks on the distributed and cooperative control microgrid system and formulate the problem under study. Section 4.3 proposes a detailed secure control framework. Section 4.4 describes the hardware testbed we designed to realize the distributed and cooperative control and implement the designed attacks. At last, Section 4.5 summarizes this chapter.

4.2 Attack Modeling and Problem Formulation

In this section, we describe a generalized attack model for the proposed distributed and cooperative control system described in Chapter 3. Both FDI and DoS attack scenarios are considered.

4.2.1 Modeling of FDI and DoS Attacks

To build general models for both FDI and DoS attacks with regard to the distributed and cooperative control system with quadratic nonlinearity, we adopt *Jacobian linearization* to study the droop controlled DG units (3.8) around an equilibrium point ($\bar{V}$, $\bar{V}^*$). As described in equation (3.9), $-([\kappa] + L(\theta))\,\bar{V} + [\kappa]\,\bar{V}^* = 0$ must hold at the equilibrium point. Note that state matrix $A = -[\bar{V}][\tau]^{-1}([\kappa] + L(\theta))$ and input matrix $B = [\bar{V}][\tau]^{-1}[\kappa]$. For the sake of simplicity, let $\bar{V} = \mathbf{1}pu$ subsequently, where $\mathbf{1}$ denotes a vector with all entries equal to 1.

To better describe the system dynamics of the microgrid system, we introduce additional terms $\omega(t)$ and $y(t)$ to the linearized dynamic system (3.9):

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + \omega(t) \\ y(t) &= Cx(t) \end{aligned} \tag{4.1}$$

where $\omega(t) \in \mathbb{R}^n$ is the Gaussian process noise representing the system errors. Furthermore, $C \in \mathbb{R}^{m \times n}$ is the output matrix and the output $y(t) \in \mathbb{R}^m$ is observed by a sensor network with m sensors.

FDI and DoS attacks are the two major types of attacks in CPS systems. FDI attacks target data integrity, where the adversary forwards false data from one or more corrupted sensors

or controllers. The injected false data could cover incorrect measurements, timestamps, or sender/receiver IDs. FDI attacks can be launched by breaking the underlying cryptography system or compromising some sensors/controllers or the communication links between them. The MaR that we designed and validated through simulation in Chapter 2 is an example of this kind of attacks where the attacker modifies the transmitted reference value by the measurement value. We have shown that such attack deviates the voltage control and the system becomes unstable.

On the other hand, DoS attacks target system availability. The adversary makes the sensor measurements unavailable to the controllers or the control commands unavailable to the actuators. DoS attacks can be launched by jamming the communication links, delaying or dropping data packets, modifying the network routings, data flooding on the network, etc. A well known DoS attack in microgrids is delay injection attacks where the attacker introduces some communication delay to the travelling packets in the network to make the valid data received with time-delays. The introduced delay affects the control system behavior, which is time critical, and creates instability in the power system that may compromise its availability.

We propose a general framework to model both FDI and DoS attacks against the microgrid system by using additive changes to (4.1):

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + \omega(t) + \Gamma r(t) \\ y(t) &= Cx(t) + \Psi z(t) \end{aligned} \tag{4.2}$$

where the attack vectors $r(t)$ and $z(t)$, and their respective coefficient matrices Γ and Ψ are to be determined by the FDI/DoS attack settings. We assume that the attack starts at time $t = t_0$, and we introduce the unit step function $1_{\{t \geq t_0\}}$ to model this assumption, where $1_{\{t \geq t_0\}} = 0$ for $t < t_0$ and $1_{\{t \geq t_0\}} = 1$ for $t \geq t_0$.

For FDI/DoS attack on a subset $\mathcal{U}$ of control signals, Γ and $r(t)$ in (4.2) are determined as

$$\begin{aligned} \Gamma &= B \\ r_i(t) &= \begin{cases} (-u(t) + \alpha_i(t))1_{\{t \geq t_0\}}, & \forall i \in \mathcal{U} \\ 0, & \forall i \notin \mathcal{U} \end{cases} \end{aligned} \tag{4.3}$$

where $\alpha_i(t)$ is the corrupted control signal sent by the attacker. Note that $\alpha_i(t) \neq 0$ for FDI attack and $\alpha_i(t) = 0$ for DoS attack.

Similarly, for FDI/DoS attack on a subset $\mathcal{Y}$ of sensor nodes, Ψ and $z(t)$ in (4.2) are determined as

$$\begin{aligned} \Psi &= C \\ z_i(t) &= \begin{cases} (-x_(t) + \beta_i(t))1_{\{t \geq t_0\}}, & \forall i \in \mathcal{Y} \\ 0, & \forall i \notin \mathcal{Y} \end{cases} \end{aligned} \tag{4.4}$$

where $\beta_i(t)$ is the corrupted sensor measurements sent by the attacker. Note that $\beta_i(t) \neq 0$ for FDI attack and $\beta_i(t) = 0$ for DoS attack.

4.2.2 General Stability under Attack

Besides the general modeling of FDI and DoS attacks on the microgrid system, any concrete attack scenario can be further written as

$$\dot{x}(t) = A_a x(t) + Bu_a(t) + \omega(t) \tag{4.5}$$

where A_a is the corrupted state-transition matrix and $u_a(t)$ is the manipulated input by the attacker.

Hereby we give the necessary and sufficient condition for the stability under a concrete attack (4.5).

Theorem 1. *Consider the linearized dynamics under attack* (4.5)*, and then the system is asymptotically stable if and only if the following conditions hold:*

1) A_a *is a row-diagonally dominant Metzler matrix, i.e.,* $|\tilde{a}_{ii}| \geq \sum_{j \neq i} |\tilde{a}_{ij}|, \forall i, j \in \{1, \ldots, N\}$*, where* $\tilde{a}_{ij} = A_{ij}$*, and* $\tilde{a}_{ij} \geq 0$ *for* $i \neq j$*.*

2) There exists a $\xi \in \mathbb{R}^n$ *such that* $\xi > 0$ *and* $A_a \xi < 0$*.*

Proof. The two statements in Theorem 1 get A_a is Hurwitz, i.e., every eigenvalue of A_a has a strictly negative real part [40]. So the necessary and sufficient condition for stability under attack (4.5) follows directly from Lyapunov's first method of stability criteria. □

4.3 Secure Control Framework

In this section, we propose a comprehensive secure control framework relying on statistical machine learning algorithms to detect attacks targeting microgrid systems. We explore various machine learning algorithms to design the attack detection module against observable/stealthy FDI attacks and DoS attacks modeled as (4.3) and (4.4).

4.3.1 Architecture

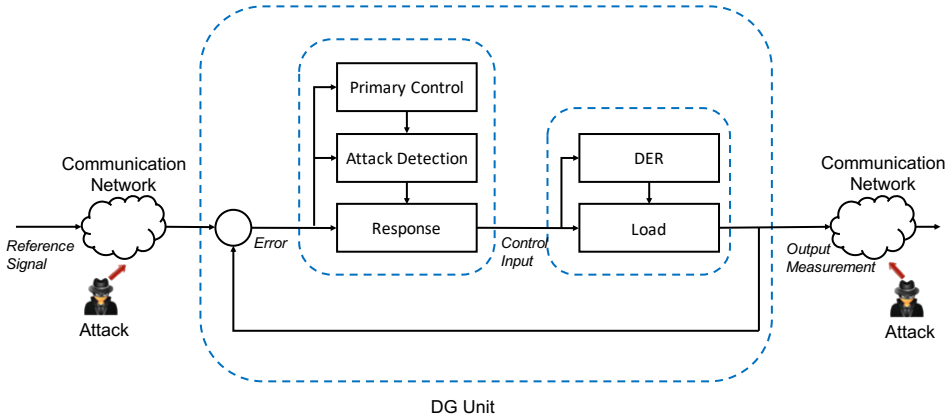

Figure 4.1: Secure control framework with attack detection and response modules.

As shown in Fig. 3, we illustrate a secure control framework with attack detection and response modules at each DG unit. Instead of building a centralized attack detection and response system, the distributed and cooperative DG units can actively defend and maintain the integrity and availability of the microgrid system.

A detailed description of the architecture and the components of the attack detection and response modules is shown in Fig. 4.2. The proposed architecture contains the following four main stages:

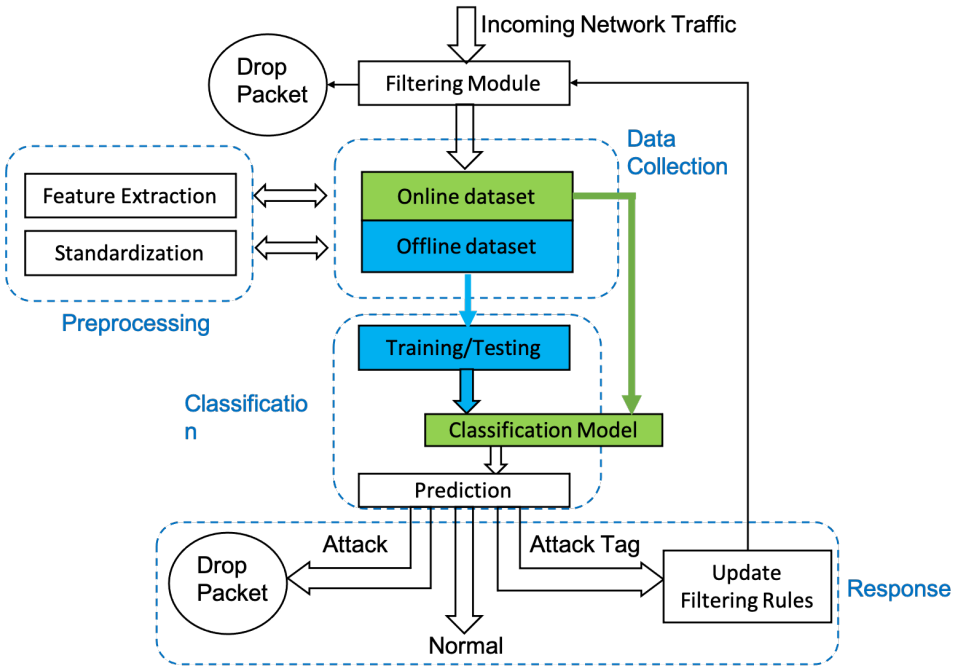

Figure 4.2: Internals of attack detection and response modules in the secure control framework.

1) *Data Collection.* Data collection is the process of gathering and measuring information on targeted variables in the system. All incoming packets are captured and stored as sets of traffic flow. Each sample is described by a set of features and expressed in the vector space model.

2) *Preprocessing.* Data preprocessing is that step in which the data gets transformed, or encoded, to bring it to such a state that is easy to parse. In other words, the features of the data can now be easily interpreted by the algorithm. It extracts information about packet connections from data and constructs new statistical features. A feature is an individual measurable property or characteristic of a phenomenon being observed. Standardization is needed if there is a necessity to eliminate the effect of scale differences of features.

3) *Classification.* Dataset of particular class is split into subsets. Different machine learning and deep learning classifiers are trained by using labeled datasets. Cross-validation is used for manipulating training data to subdivide the training data into k disjoint subsets and to reconstruct training sets by leaving out some of the subsets.The performance of the classifiers are tested on online data. Different groups of machine learning and deep learning algorithms are chosen as the candidate of the "Classifier" module. Comparing the performances of each algorithm and the algorithm with the best performance in terms of certain metrics will be determined as the final classifier. As shown in Algorithm 1, we iteratively evaluated a candidate set of known ML techniques to identify the best technique that is able to detect the running attacks with high performance.

4) *Response.* The detection system deployed in each DG unit maintains and updates a hash table of attack tags recording the IP address and port number of the suspicious nodes under attack. When a DG unit receives an attack tag, it verifies if it exists in its hash table. If not, it adds the attack tag to a blacklist and sends the updated attack tag to all cooperative nodes

Algorithm 1 Selection of the best attack detection method for Classification module.

Input: The training dataset S_1 and testing dataset S_2 with correct labels $y_i \in \Omega = \{0, 1, 2\}$; the set of various machine learning and deep learning classifiers $D = \{Ada\_Boost, KNN, 1D\_CNN, ..., etc.\}$;

Output: The best performed classifier $d^* \in D$ in attack detection;

```
Initialize F1 = 0;
for i ∈ D do
    Train classifier i with S1;
    Test classifier i with S2;
    Calculate the F-Measure F1_i;
    if F1_i > F1 then
        F1 = F1_i ;
        d* = i;
    end if
end for
return d*;
```

to update their respective filtering modules. Data packets from suspicious nodes available in this blacklist could be dropped to prevent possible damages to the microgrid system. As shown in Algorithm 2, we propose a detailed response mechanism triggered by each DG unit after receiving labeled packets steam from the previous the classification module. When the boolean value $Drop\_packet = True$, it means the DG unit will simply drop the received packet. Similarly, when the boolean value $Send\_List\_to\_neighbours = True$, it means the DG unit will spread the updated hash table to its neighbouring nodes.

Algorithm 2 The response mechanism of the DG unit after receiving labeled packets form the classification module.

Input: A packets stream $P = [P_1, P_2, P_3, ...]$, where $P_i = \{P_i\_value, P_i\_label, P_i\_ip\_address, P_i\_port\_number\}$, the hash table $List$;

Output: The updated hash table $List$, the boolean values $Drop\_packet$ and $Send\_List\_to\_neighbours$;

```
Initialize Hashtag = 0, Drop_packet = False, Send_List_to_neighbours = False;
for P_i ∈ P do
    if P_i_label = "Attack" then
        Hashtag = Hash(P_i_ip_address, P_i_port_number);
        if Hashtag ∉ List then
            List = List.add(Hashtag);
            Send_List_to_neighbours = True;
        end if
        Drop_packet = True;
    end if
end for
return List, Drop_packet, Send_List_to_neighbours;
```

4.3.2 Selected Machine Learning algorithms

The classification module that we described in the previous subsection relies mainly on a classification method to detect attack packets from the network traffic exchanged between the DGs of the microgrid. From a statistical learning perspective, attack detection can be modeled as a typical classification problem that maps each network packet transmitted between DG units to a label, including normal or attack labels and the attack type. Many machine learning algorithms have been already used for the detection of attacks in communication networks of microgrid systems. They use network traffic, control and sensing features as input of their classification methods.Therefore, there is no best classification model for all applications but some will perform better than others in our case.

Since we have full knowledge of the FDI and DoS attack models as elaborated in (4.3) and (4.4), we can implement the attacks and collect labeled datasets of them, which is usually a difficult task to realize in real life. Thus, we can explore supervised machine learning algorithms and state-of-the-art deep learning algorithms to validate their effectiveness for attack detection in microgrid systems.

Considering that it is usually tricky to find one rule (or feature set) which outperforms other rules (or features) of particular classification methods, we also test ensemble learning algorithms. Therefore, we select **seven** different machine learning classifiers grouped into **five** categories, and also one deep learning algorithm represented by convolutional neural networks, as shown in Fig. 4.3.

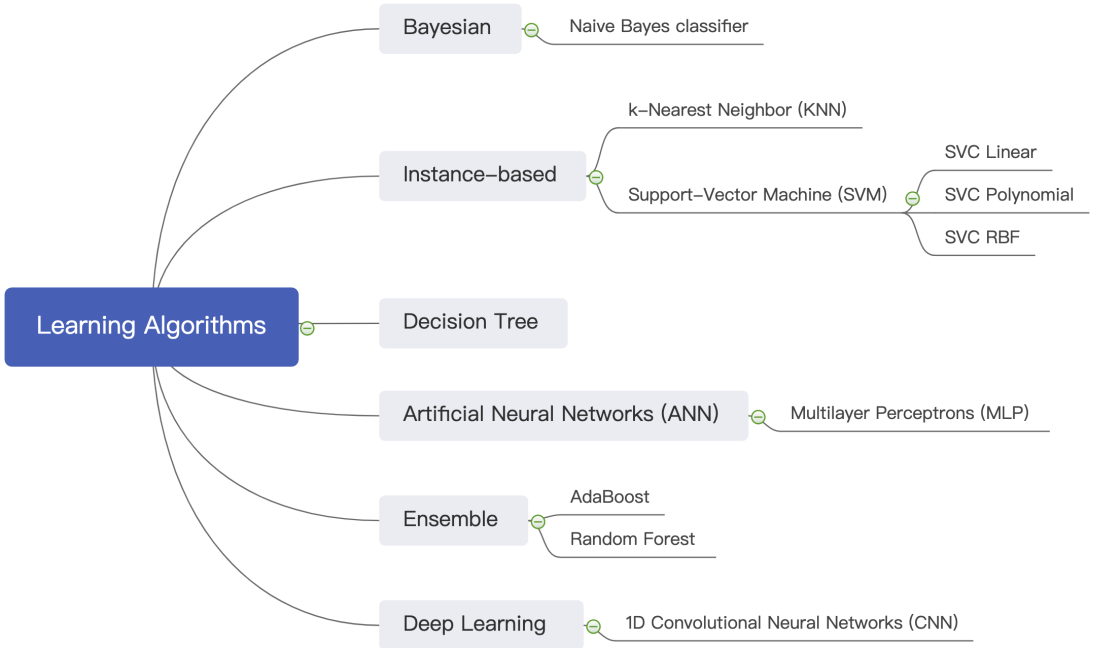

Figure 4.3: Selected learning algorithms and their respective groups.

Bayesian Algorithms

The Naive Bayes classifier is a classical probabilistic classifier with strong independence assumptions between the features. It demonstrates how generative assumptions and parameter estimations can simplify the learning process [23]. The Naive Bayes classifier requires a number

of linear parameters in the number of variables (features or predictors) in the learning problem, thus it is highly scalable. By evaluating a closed-form expression, the Naive Bayes classifier can do maximum-likelihood training in linear time, in contrast to expensive iterative approximation that is used by many other types of classifiers.

Instance-based Algorithms

This category of algorithms do not derive abstractions for the classification problem, instead they rely on the similarity of an instance to its nearest neighbours available in the training set to predict a class label. In this category, we used two algorithms: K-Nearest Neighbor (KNN), Support-Vector Machine (SVM).

- The KNN classifier finds k samples in the training data that are closest to the test sample, and labels the new sample with the most frequent label among these samples. The KNN classifier outputs a class membership. Based on the plurality vote of its neighbors, an object is classified to the most common class among its k nearest neighboring objects, where k is positive integer and typically small. When $k = 1$, then the classifier simply assigns the object to the class of its single nearest neighbour [23, 59].

- SVM is used for classification by constructing a hyperplane or set of hyperplanes in a high or infinite-dimensional space. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on the side of the gap on which they fall [5, 59]. To keep the computational load reasonable, the mappings used by SVM schemes are designed to ensure that dot products of pairs of input data vectors may be computed easily in terms of the variables in the original space, by defining them in terms of a kernel function$k(x, y)$ selected to suit the problem. The hyperplanes in the higher-dimensional space are defined as the set of points whose dot product with a vector in that space is constant, where such a set of vectors is an orthogonal (and thus minimal) set of vectors that defines a hyperplane. The vectors defining the hyperplanes can be chosen to be linear combinations with parametersα_i of images of feature vectors x_i that occur in the data base. With this choice of a hyperplane, the points x in the feature space that are mapped into the hyperplane are defined by the relation$\sum_i \alpha_i k(x_i, x) = \text{constant}$. Note that if $k(x, y)$ becomes small as y grows further away from x, each term in the sum measures the degree of closeness of the test point x to the corresponding data base point x_i. In this way, the sum of kernels above can be used to measure the relative nearness of each test point to the data points originating in one or the other of the sets to be discriminated. Note the fact that the set of points x mapped into any hyperplane can be quite convoluted as a result, allowing much more complex discrimination between sets that are not convex at all in the original space.

Decision Tree Algorithms

Decision Tree has a structure like a flowchart, where each internal node denotes a test on a data attribute, each branch denotes the outcome of the test, and each leaf node denotes a class prediction. A decision tree is a predictor that predicts the label associated with an instance x by traveling from a root node of a tree to a leaf. At each node on the root-to-leaf path, the successor child is chosen on the basis of a splitting of the input space. Usually, the splitting

is based on one of the features of x or on a predefined set of splitting rules. A leaf contains a specific label. A decision tree is built by splitting the source set, constituting the root node of the tree, into subsets. which constitute the successor children. The splitting is based on a set of splitting rules based on classification features. This process is repeated on each derived subset in a recursive manner called recursive partitioning. The recursion is completed when the subset at a node has all the same values of the target variable, or when splitting no longer adds value to the predictions [59].

Artificial Neural Network (ANN) Algorithms

Multilayer Perceptrons (MLP) a class of feedforward ANN optimizing the weights for the activation function of neurons organized in a network architecture. An MLP consists of at least three layers of nodes: an input layer, a hidden layer and an output layer. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called backpropagation for training. Back-propagation learning may be implemented in one of two basic ways, sequential mode and batch mode. In sequential mode, adjustments are made to the free parameters of the network on an example-by-example basis. The sequential mode is best suited for pattern classification. In batch mode, adjustments are made to the free parameters of the network on an epoch-by-epoch basis, where each epoch consists of the entire set of training examples. The batch mode is best suited for nonlinear regression [59]. To address overfitting and improve the neural network, L2 regularization can be applied as regularization methods for neural networks.

Ensemble Algorithms

This category of algorithms relies on a set of estimators, in particular decision tree, to build a multi-stage classifier and predict the class label according to the output of this set of estimators. These algorithms apply averaging or boosting methods to produce the final prediction. in this category, we applied two algorithms: AdaBoost and RandomForest.

- AdaBoost classifiers implement boosting to combine "weak classifiers" into a single "strong classifier". AdaBoost classifier builds a strong classifier by combining multiple poorly performing classifiers so that you will get high accuracy strong classifier. The basic concept behind AdaBoost is to set the weights of classifiers and training the data sample in each iteration such that it ensures the accurate predictions of unusual observations. Any machine learning algorithm can be used as base classifier if it accepts weights on the training set. AdaBoost classifiers should be trained interactively on various weighed training examples. In each iteration, it tries to provide an excellent fit for these examples by minimizing training error [55, 59].

- Random Forest classifiers consist of a collection of decision trees, where each tree is constructed by applying an algorithm A on the training set S and an additional random vector, θ, where θ is sampled independent and identically distributed from some distribution [59]. The prediction of the random forest is obtained by a majority vote over the predictions of the individual trees. As a meta estimator, Random Forest fits multiple decision tree classifiers on various sub-samples of the dataset and improves its predictive accuracy by averaging. Random Forest adds additional randomness to the model, while growing the trees. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. This results in a

wide diversity that generally results in a better model.Therefore, in Random Forest, only a random subset of the features is taken into consideration by the algorithm for splitting a node. We can even make trees more random by additionally using random thresholds for each feature rather than searching for the best possible thresholds (like a normal decision tree does).

Deep Learning algorithms

More recently, few work have applied [36] deep learning algorithms, in particular Convolutional Neural Networks (CNNs) , for attack detection in cyber-physical systems. CNNs are feedforward neural networks that became popular in image processing. By applying convolutions to small regions of the input, CNNs can significantly improve the efficiency of neural networks by avoiding performing matrix multiplication over the entire image at once. Unlike two-dimensional (2D) CNNs used in image processing, 1D CNNs can be successfully used for time series processing because time series have a strong 1D (time) locality which can be extracted by convolutions. The work in [36] shows that 1D CNNs can be effectively used for detecting cyber-attacks in complex multivariate ICS data. To compare with the above machine learning algorithms, we also selected 1D CNNs to evaluate its performance for detecting attacks in mircorgrids.

4.3.3 Performance metrics

The performance evaluation of the selected ML algorithms relies on a set of widely used metrics that we will explain in this subsection. The evaluation of these metrics will allow us to identify the best ML algorithms that are able to detect attack on microgrid systems, including the MaR attack that we described in Chapter 3. In binary classification, we indicate data under attack as positive, and normal data as negative. Denote tp as the number of true positives, fp as the number of false positives, tn as the number of true negatives, and fn as the number of false negatives. We measure the learning capabilities of the algorithms by three classical metrics which are: *Precision, Recall and F-Measure*. The *Precision* measures the performance in predicting positive samples, which is known as *positive predictive value (PPV)*.

$$Precision = \frac{tp}{tp + fp}$$

On the other hand, *Recall* measures the capability to identify all the positive samples, which is also called *true positive rate (TPR)*.

$$Recall = \frac{tp}{tp + fn}$$

Finally, $F - Measure$ is the harmonic mean of Precision and Recall and captures both Precision and Recall properties as a combined single measure. Note that we choose F-Measure over another standard metric Accuracy as the total classification performance metric because F-Measure gives a better measure when the class distribution is imbalanced like in the attack detection problem.

$$F - Measure = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

A classifier has a strong performance when the above measures are approaching 1. To evaluate the classification performance of each learning algorithm, we also consider receiver operating characteristic (ROC) curves by plotting the true positive rate (TPR) against the false

positive rate (FPR), where $TPR = \frac{tp}{tp+fn}$, $FPR = \frac{fp}{fp+tn}$. For each ROC curve, its area under the curve (AUC) directly represents the classification performance where a better classifier has a larger AUC value.

When running the classifiers, we employ k-fold (e.g., $k = 5$) cross-validation techniques to protect against overfitting during training. We partition the dataset into k randomly chosen subsets of roughly equal size. We use one subset to validate the performance of the classifier model trained from the remaining subsets. This process is repeated k times to guarantee that each subset is used exactly once for validation.

4.4 Experimental platform and data collection

Our proposed framework relies on a ML algorithm for the detection of attacks which require the collection of several datasets of normal and attack samples to be able to evaluate the candidate ML algorithms and select the algorithms that perform better than the others. However, it is difficult to find publicly available datasets for the two proposed attacks. For that reason, we implemented two types of attacks targeting the communication links and instantiated them on a hardware platform built for emulating the distributed and cooperative controlled microgrid system. We used this platform to collect several datasets by running it in normal and attack modes.

4.4.1 Microgrid testbed

As shown in Fig. 4.4, we built a hardware testbed modeling a simplified microgrid platform consisting of 4 DG units.

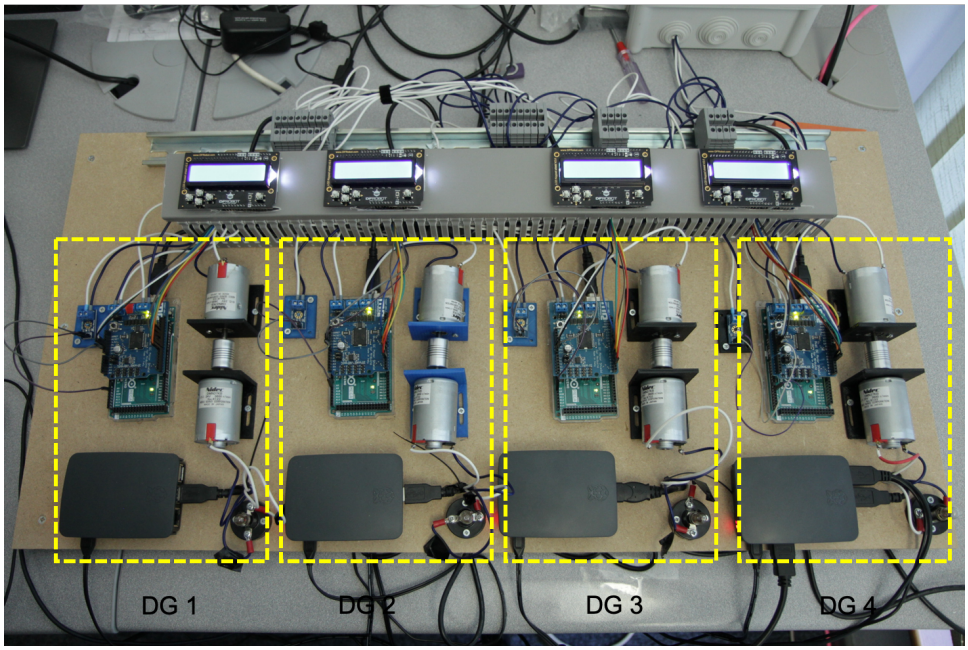

Figure 4.4: The hardware microgrid platform consisting of 4 DG units that perform distributed and cooperative control strategy.

As shown in Fig. 4.5, each DG unit contains the following hardware components:

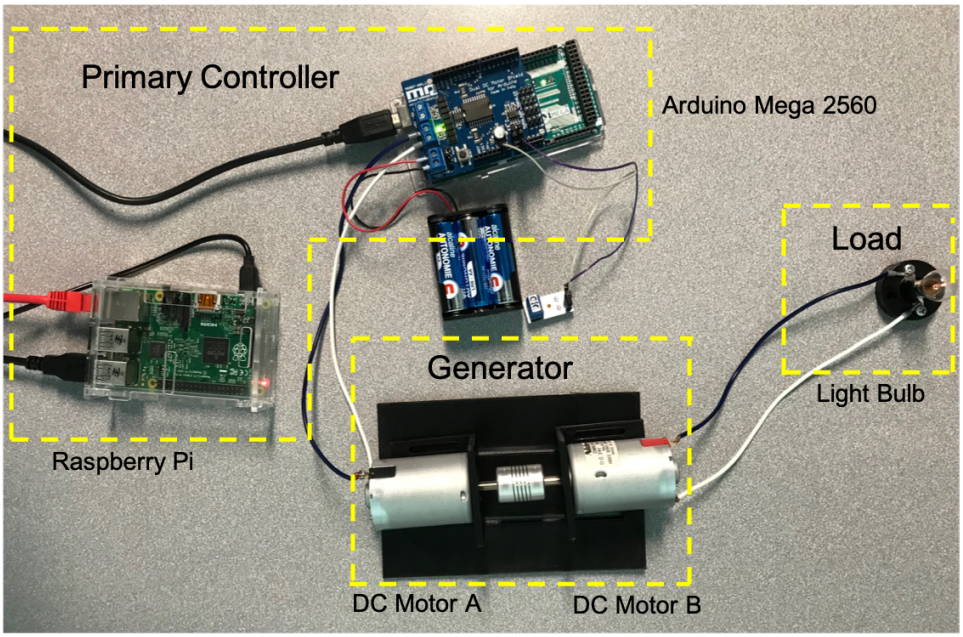

Figure 4.5: The hardware components of each DG.

- a Raspberry Pi as the main control center of the DG unit;
- an Arduino with a motor shield runing a PID control algorithm to drive the DC motor A;
- two DC motors, where DC motor A drives DC motor A to generate electrical power;
- a voltage sensor measuring the voltage at each DG unit;
- a lightbulb serving as the domestic load.

For each DG unit, one Raspberry Pi and one Arduino board act as the primary controller, two motors A and B act as a power generator, and one lightbulb acts as an electrical load. The primary control equivalent to (3.6) is realized in the Arduino board with the help of PID settings. The Raspberry Pi devices are connected through a WiFi network and only communicate with its neighbor using unencrypted messages. DG 1 acts as the leader node and has access to the secondary control voltage instructions, and the remaining DG units exchange information with neighboring nodes and synchronize their voltage value to DG unit 1.

4.4.2 The distributed control system

In order to give the control instructions from a DG to another (concretely from a Raspberry Pi to its neighbouring Raspberry Pi), it is necessary to be able to run simultaneously a server and a client on the same DG. The server part receives the control instructions (the measurement and the reference values) whereas the client gives it to the next node of the tree topology.

The server listening on a specific port, waits for clients to connect and then handles the packets it gets from them. In our case, the server just accepts the connections of the clients and receives their messages. Figure 4.6 shows the different connections between the clients and the servers running the 4 DGs to propagate the control instructions between them and synchronise to the desired voltage.

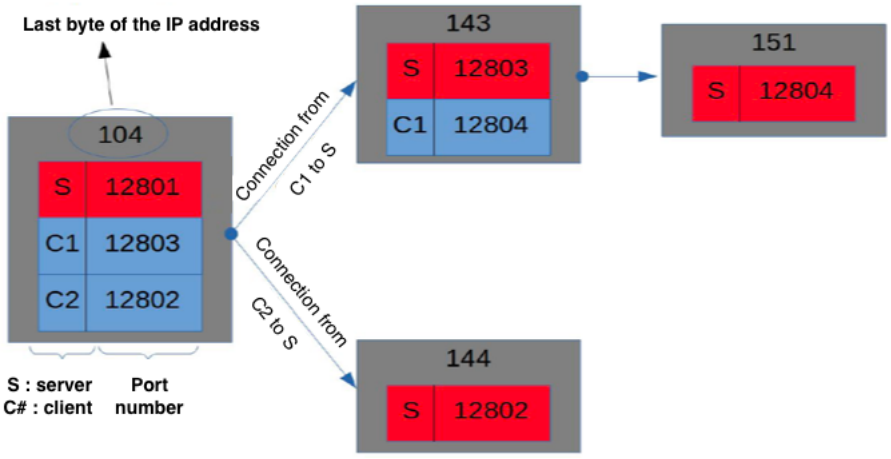

Figure 4.6: Illustration of the communication between the DGs acting as clients and server for the transmission of control values.

To inject a control instruction in the network which is in our case the desired voltage, an other kind of client is needed, which serves as the secondary controller. Its objectives are the same as the others but the main difference is that it does not wait for receiving an instruction from a previous node because it does generate the control instruction. The user is asked for the instruction value (it must be between 0 and 12V - 0 is excluded because the Arduino will not accept it). The rest of the function is similar to what the *client* function does.

4.4.3 Attacks implementation

We implemented two specific attacks on our platform to illustrate the FDI and the DoS attacks modeled in Section 4.2.1:

- Measurement-as-reference (MaR) attack: the attacker intercepts the communication link between DG unit i and unit $i+1$ to compromise the transmitted data packets, and maliciously substitutes the reference value $V^*_{i+1}(t)$ of DG unit $i+1$ with the voltage measurement $V_i(t)$ of DG unit i.

- Delay injection attack: the attacker injects some delays according to a specific statistical distribution to the packets exchanged between DG unit i and unit $i+1$ to perturb the communication. In this work, we consider a Gaussian-distributed delay injection attack with a proper mean and variance to cause loss of system availability without being observed, because it is tricky to distinguish delay injection attack from naturally generated fault delay. We use an open-source tool "Saboteur" [1] to inject delays into the communication links.

MaR attack implementation with Man-in-the-Middle technique

For MaR attack, the reference value $V^*_{i+1}(t)$ is close to the voltage measurement $V_i(t)$, and they both transmit in the same link. MaR attack is naturally stealthy in front of traditional residual comparison detectors and can cause voltage fluctuation and synchronization errors [41]. We

[1] https://github.com/tomakehurst/saboteur

implemented the MaR attack by using a Man-in-the-Middle (MitM) technique where the attacker intercepts the packets exchanged between two successive DGs and substitues the reference value with the measurement. The attack tool is running on a specific machine connected to the same network as the DGs. To implement the attack tool, we relied on the Scapy library for packet capture and their modification before injecting them into the network.

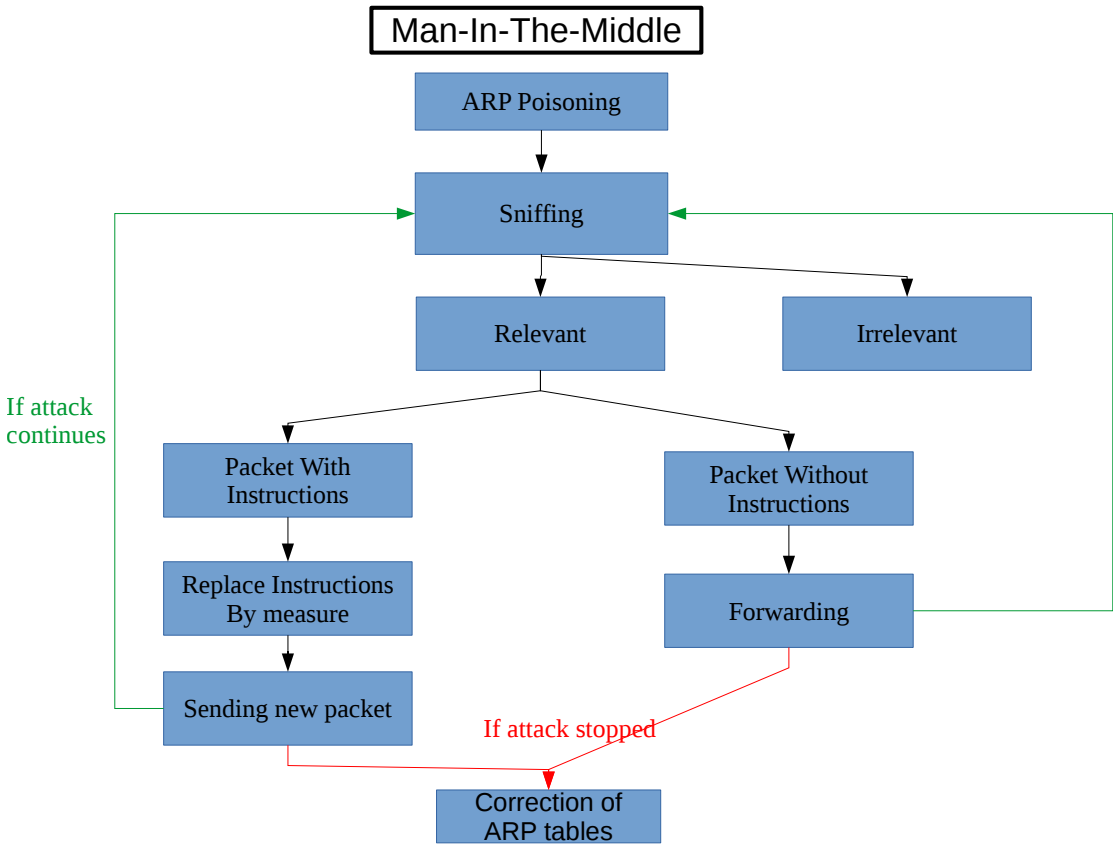

Figure 4.7: Steps of the Man-In-The-Middle technique to realize a MaR attack.

The attack tool using the MitM technique realizes the following four steps as shown in Fig. 4.7:

- *ARP Poisoning.* First, the intruder must trick the targets into thinking the one they want to talk to is him. To do so, he takes advantage of the ARP protocol to send counterfeit ARP packets to both targets : the sender and the receiver. The intruder modifies their ARP tables in order to make them send the packet to him when they want to contact their peer.

- *Packets sniffing.* Once the ARP Poisoning is launched, the intruder is in the middle of the conversation between the victims. He can observe the packets but of course alongside the packets sent through the socket, many other exist and are captured by the sniffer. Filtering the packets is important to only stop the packets we are interested in. Basically, the filter checks the IP addresses and the protocol of the packet. If these fields correspond to the socket, we examine the packet.

- *Packets inspection.* If a packet is captured, it is processed to check if it contains synchro-

nisation values. If there is an instruction, we must replace its value by the measured value also contained in the packet and the MAC addresses must also be modified because we are rerouting the packet. Eventually, we must force the recalculation of the checksum field of the TCP packet because we modified the payload. If no instruction was found in the packet, we must change the MAC addresses before sending it. These changes depend on the source and destination of the packet.

- *Stopping and restoring ARP tables.* When the intruder stops the attack, the tool modifies again the ARP tables of its victims to disturb the less possible the network when he stops playing the attack.

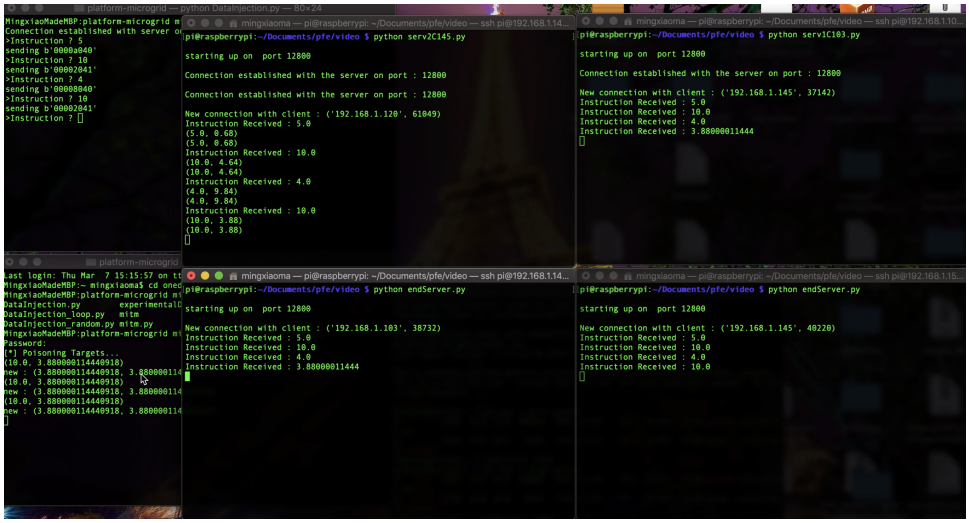

Figure 4.8: Illustration of the execution of the MaR attack in our platform.

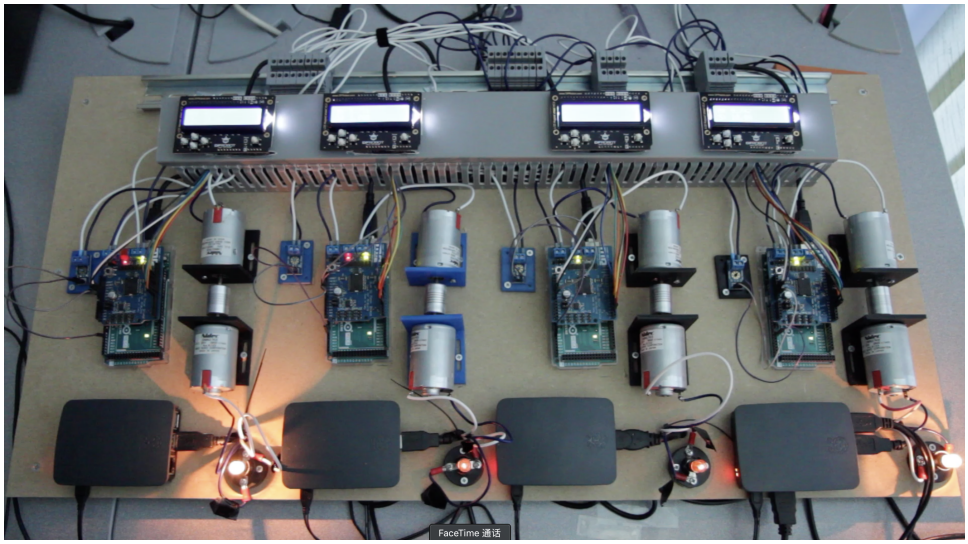

Figure 4.9: The effect of MaR attack on the microgrid testbed. We observe that the brightness of the lightbulbs in the middle are lower than the two others.

The MaR attack process can be seen from Fig. 4.8, and the MaR attack effects on the testbed

is shown in Fig. 4.9. Only one leader node (DG 1) gets access to the reference V^* sent by the secondary controller. All other nodes will synchronize their corresponding reference values with the leader node while communicating with their neighbouring nodes. The brightness differences show that the synchronization between DG units has been interrupted due to our attack.

Delay injection attack implementation

We implemented the delay injection attack by using saboteur [2] which is an open source network fault injection tool for stability testing. Its core component is an agent that accepts commands over HTTP and configures its host's network stack for various common fault scenarios: total network partition, remote service dead (not listening on the expected port), delays, packet loss and TCP connection timeout (as often happens when two systems are separated by a stateful firewall). We run the saboteur tool on one DG to delay the packets targeting the TCP port used by the control system.

Considering the delay injection attack should be stealthy/unobservable, and at the same time could cause as much damage as possible, we consider a Gaussian-distributed delay with mean $2000ms$ and variance $1500ms$.

4.4.4 Data collection and datasets description

We collected datasets under three different scenarios: normal operations without attack, MaR attack, and delay injection attack by using our experimental platform. Assuming the attacker targets the communication link between DG unit 1 and DG unit 2, we use *tcpdump* tool to capture the packets transmitted on the network and dissect them to extract five features:

- *latency*: the time difference between sending and receiving timestamps of the same packet from DG unit 1 to DG unit 2.
- ref_dg1: the reference value at DG unit 1.
- ref_dg2: the reference value at DG unit 2.
- mea_dg1: the voltage measurement value at DG unit 1.
- mea_dg2: the voltage measurement value at DG unit 2.

We run *tcpdump* on both DG1 and DG2 to capture the TCP control packets exchanged between them. Then, we extracted from the obtained PCAP files the 5 features described above. We labelled each collected dataset with three class labels: label 0 represents normal operations, label 1 represents system under MaR attack, and label 2 represents system under delay injection attack. A sample of the collected dataset is shown in Fig. 4.10.

We collected **12297** samples in total where **8736** samples are labeled as "normal," **1836** samples are labeled as "MaR attack", and 1725 samples are labeled as "delay injection attack".

This first dataset is imbalanced considering the fact that cyber-attacks happen much rarely than normal operations in real-life microgrid systems, so the collected dataset is usually with a larger proportion of "normal" samples and smaller proportions of "MaR attack" and "delay injection attack" samples. However, it is of great importance to have a balanced dataset to guarantee the performance of the classifiers based on machine learning and deep learning algorithms. Thus, for training the classifiers, we sampled the "normal" set to obtain a balanced dataset with **5552**

[2] https://github.com/tomakehurst/saboteur

	1 latency	2 ref_dg1	3 mea_dg1	4 ref_dg2	5 mea_dg2	6 class
1	0.0434	1.5513	1	1.5513	1	0
2	0.1005	11.8459	1.4200	11.8459	1.4200	0
3	0.0057	4.5594	7.4700	4.5594	7.4700	0
			...			
2845	0.2807	6.0434	5.9800	5.9800	5.9800	1
2846	0.3494	9.5572	6.2000	6.2000	6.2000	1
2847	0.1583	10.5710	9.3300	9.3300	9.3300	1
			...			
4759	0.0200	3.0665	5.7900	3.0665	5.7900	2
4760	0.0100	3.1535	3.2000	3.1535	3.2000	2
4761	2.9000	8.9683	3.0300	8.9683	3.0300	2

Figure 4.10: A set of samples extracted from the captured packets with their respective 5 feature values and class labels.

samples in total and **1991** samples labeled as "normal," **1836** samples labeled as "MaR attack", and 1725 samples labeled as "delay injection attack". Table 4.1 shows the number of samples for each scenario and for both imbalanced and balanced datasets.

-	**Imbalanced dataset**	**Balanced dataset**
Normal samples	8736	1991
MaR samples	1836	1836
Delay injection samples	1725	1725
Total	12297	5552

Table 4.1: Number of samples for normal and attack scenarios with balanced and imbalanced datasets.

We will train and test the classifiers both on the balanced and imbalanced datasets. By comparing the performance differences of the classifiers on balanced and imbalanced datasets, we can better evaluate the performances of different algorithms as attack detectors. In particular, we are interested in the evaluation of the effect of the imbalanced dataset on the performance of the classifiers algorithms.

4.5 Summary

In this chapter, we have designed and described a secure control framework for microgrid systems. We have modeled both FDI and DoS attacks with regards to the distributed and cooperative control microgrid. The secure framework mainly contains four parts: data collection, preprocessing, classification and response. With help of various machine learning and deep learning algorithms, we further detail the classification module of the secure framework. To generate datasets containing normal and attack samples, we build a microgrid testbed to implement two typical attacks, measurement as reference attack and delay injection attack.

Chapter 5

Evaluation Results of Attack Detection Methods

Contents

5.1 Introduction

In this chapter, we validate the attack detection methods to be used by our secure control framework described in Section 4.3. The two typical attacks targeting the communication links, namely measurement as referent (MaR) and delay injection attack are implemented on the hardware testbed described in Chapter 4

Before studying the machine learning and deep learning based detection algorithms, we firstly show the inefficiency of traditional threshold comparison method in attack detection for the distributed microgrid system. Experimental results show that traditional threshold comparison method performs poorly in terms of precision, recall and accuracy.

To validate the performance of machine learning and deep learning based attack detection, we give a detailed hyper-parameter tuning methodology to get the best hyper-parameters used for classification. We collect datasets from the platform and compare the performances of different machine learning and deep learning algorithms to detect MaR and delay injection attacks. The classification problem will be divided into three categories: binary classification of "normal" and "MaR attack", binary classification of "normal" and "delay injection attack", and multi-class classification of "normal" , "MaR attack" and "delay injection attack". We give a detailed analysis for the three classification problems on balanced and imbalanced dataset, respectively.

The rest of this chapter is organized as follows. Section 5.2 validates the inefficiency of traditional threshold comparison method in detecting MaR and delay injection attacks. Section 5.3 describes how we tune the machine learning and deep learning hyper-parameters and list the best parameters of each classifier that we use for classification. Section 5.4 and Section 5.5 show the experimental results and performance analysis for collected imbalanced and balanced dataset, respectively. In the end, Section 5.6 gives a comprehensive discussion and analysis about the experimental results.

5.2 Detection using threshold comparison method

In this section, we make a first experiment to mainly motivate the choice of relying on ML algorithms in our secure control framework, described in the previous chapter, to detect attacks in microgrid systems.

For that, we evaluate the performance of a well known technique for corrupted data detection in such systems that relies on state vector estimation (SVE) method. The method obtains the system state, which is estimated from the observed measurements and then computes the residual between the estimated and the observed measurements. If the residual is greater than the given threshold, a data injection attack is detected.

We believe that threshold comparison based methods applied to (SVE) cannot be deployed directly in the microgrid system described in Section 4.2, because there is no globally centralized estimation system in the distribution network of power systems. However, we can still utilize the same threshold comparison idea to design a straightforward detection algorithm of our MaR attack described in Chapter 3. Since the voltage measurement value V_i and reference value V_i^* at each DG follow the equation 3.5, we compute the residue $r = |V_i - V_i^*|$ and compare r with a certain threshold δ, where $\delta > 0$. If $r \geq \delta$, we consider that the system is under attack.

To evaluate the performance of the above threshold comparison method, we ran the microgrid testbed described in Section 4.4, under both normal operations and under MaR attack described in Section 3.4.1, to get **2097** normal and **1040** attacked samples. We tune the δ value from 0.05V to 1.20V, with a step 0.05V, to observe how the accuracy, precision and recall changes when δ changes.

As shown in Fig. 5.1, we observe that when the threshold value δ is low, the attack detection achieves high recall and low accuracy. When we increase δ, the recall decreases rapidly and becomes close to 0 when $\delta = 1.20\,V$, and the accuracy gets higher gradually. Overall, the precision of this detection algorithm is very low (lower than 0.4). Fig. 4 depicts the ROC curve of this detection method where the true positive rate and the false positive rate are computed under different thresholds. We observe that the performance of a classifier based on this method is quite bad and it is not better than random.

So this method is not a good choice for attack detection in this system.

These results show that such method, even its is simple to apply and realize, is inefficient to

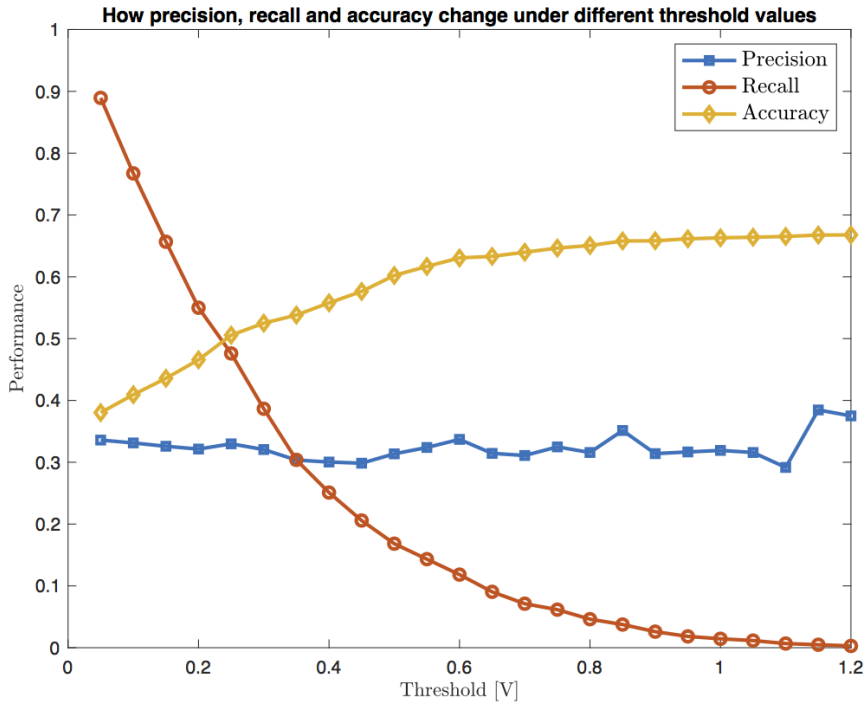

Figure 5.1: Performance metrics (Accuracy, Precision and Recall) of the threshold comparison method.

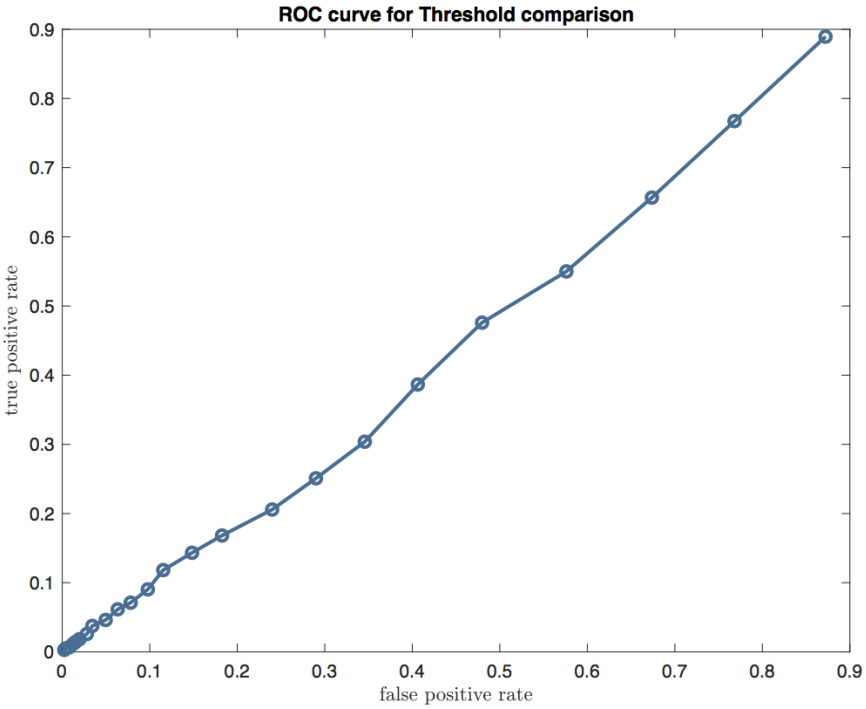

Figure 5.2: ROC curve of a classifier based on threshold comparison applied to detect the MaR attack.

detect our synchronisation attack. Therefore, it is more appropriate to apply learning techniques and evaluate their performance for such attack and others.

5.3 Hyper-parameters tuning of the selected ML algorithms

In order to validate and evaluate the set of ML algorithms, described in the previous chapter (Section 4.3.2), on the collected datatsets, we use the scikit-learn library [3] for traditional algorithms, and we use Keras and Tensorflow to build and evaluate a 1D CNN neural network. We also identified manually the optimal hyperparameters for each algorithm that provide the best performance according to the metrics that we described in subsection 4.3.3. In the next subsection, we illustrate our methodology for parameters tuning of Random Forest on a balanced dataset and then we detail the tuned parameters values of the selected ML algorithms.

5.3.1 Hyper-parameters tuning of Random Forest on a balanced dataset

To illustrate the method of hyper-parameters tuning, we consider the Random Forest classifier on binary classification of "normal" and "delay injection attack" as an example. We rely on validation curves that map different parameter values to a performance score and then visually check the potentially optimized values of the classifier model. A validation curve can be plotted on a graph to show how well a model performs with different values of a single hyper-parameter. As shown in Fig. 5.3, the three different hyper-parameters, *number of trees*, *max_depth*, and *max_features*, are plotted while varying their respective values to show how they affect the F-Measure metric of the Random Forest classifier model. We varied the *n_estimators* and *max_depth* parameters between 1 and 50 with a step of 1. We varied the *max_features* parameter between 1 and 5 with a step of 1. For each varied parameter, we plotted its F-measure scores during the training and also during the cross-validation to avoid the overfitting with a k-fold equal to 5. Through the different plots, we can easily observe that the appropriate values of these hyper-parameters are as follows: $number\ of\ trees = 30, max\_depth = 20, max\_features = 3$.

[3] https://scikit-learn.org/stable/index.html

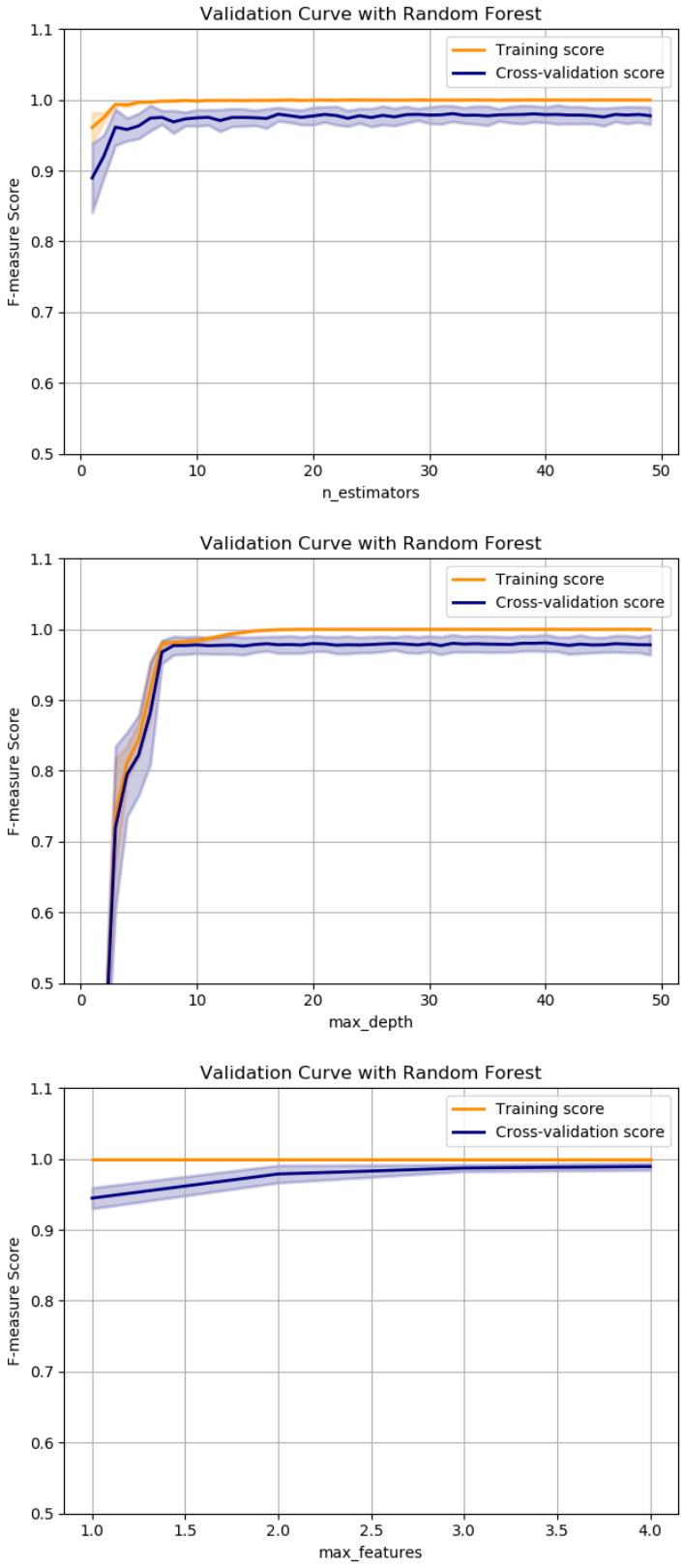

Figure 5.3: Validation curves for the hyper-parameters n_estimators (number of trees), max_depth and max_features of the Random Forest algorithm for the classification of delay attack and normal behaviour of the system.

5.3.2 Selected hyper-parameter values

The optimal parameters of the machine learning and deep learning algorithms are listed in Table 5.1. The AdaBoost classifier uses 30 decision trees with a max depth of 20 for each of them. The KNN classifier is tuned with a number of k nearest neighbours equal to 10 of each query point. The MLP classifier is tuned with 100 neurons in the hidden layer and a number of epochs equal to 1000. For the Naive Bayes classifier, we used a Gaussian model with the default values [4]. The Random Forest classifier uses 30 decision trees with a maximum depth of 20 of each of them and with a maximum number of features equal to 3 to consider for the best split. For the SVC polynomial classifier, we selected a degree of 2 and the other parameters shared with linear and RBF SVC classifiers have default values.

The 1D CNN is implemented using a single convolutional layer containing 16 filters with a kernel size equal to 2 for each of them and an activation function of type *relu*. We add also to the neural network a dropout layer with a value of 0.5, a Max polling layer of size 2 and a dense layer with a *relu* activation function. The last layer used for the classification is relying on a sigmoid function. The model is using an Adam optimizer with a binary cross entropy loss function for binary classification and categorical cross entropy loss function for the multi class classification. The accuracy metric is used for model evaluation during training and testing.

Table 5.1: Parameters of the selected machine learning algorithms.

Classifier	Parameters
Ada Boost	DecisionTree estimator with max_depth=20, number of estimators=30, learning rate=0.1
Decision Tree	max_depth=20
KNN	number of neighbors=10, metric='cosine'
MLP	number of neurons=100, L2 penalty=1, max_iter=1000
Naive Bayes	GaussianNB with default parameters
Random Forest	number of trees=30, max_depth=20, max_features=3
SVC Linear	kernel='linear', regularization parameter=1.0
SVC polynomial	kernel='poly', degree=2, gamma='auto', regularization parameter=1.0
SVC RBF	kernel='rbf', gamma='auto', regularization parameter=1.0
1D CNN	filters=16, kernel_size=2, activation='relu'

5.4 Evaluation of ML algorithms with imbalanced dataset

In this section, we use the imbalanced dataset to train and test the selected machine learning and deep learning algorithms. Imbalanced dataset has more "normal" data samples compared to data samples of "MaR attack" and "delay injection attack". We have to note that this imbalanced dataset is more close to real-life datasets compared to balanced dataset, since the number of attack packets and samples is always lower than those generated during normal operations. Our

[4] https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html

goal is evaluate the selected ML algorithm with such imbalanced dataset described in Table 4.1 in chapter 4 and identify the effect of such imbalance on their performance. Therefore, we evaluated the attack detection performances into three different classification problems: binary classification of normal and MaR attack, binary classification of normal and delay injection attack, multi-class classification of the three labels (normal, delay inject and MaR).

5.4.1 Classification of "normal" and "MaR attack" with imbalanced dataset

The ROC curves of the selected classifiers for binary classification with "normal" and "MaR attack" are shown in Fig. 5.4. We can find that many classifiers perform well with a high $AUC = 0.99$. These classifiers are KNN, MLP, Random Forest, SVC Linear, SVC polynomial and SVC RBF. It means that these models can distinguish samples labeled as "normal" and "MaR attack" with a probability of 99%. Note that 1D CNN classifier has the worst $AUC = 0.87$, which shows that traditional machine learning outperforms deep learning in this specific classification problem.

Table 5.2: Performances of different classifiers for binary classification with "normal" and "MaR attack" on imbalanced dataset.

Classifier	Precision	Recall	F-Measure
Ada Boost	0.93 (+/- 0.23)	0.97 (+/- 0.01)	0.95 (+/- 0.13))
Decision Tree	0.93 (+/- 0.23)	0.97 (+/- 0.01)	0.95 (+/- 0.13)
KNN	0.96 (+/- 0.14)	0.91 (+/- 0.05)	0.93 (+/- 0.05)
MLP	1.00 (+/- 0.00)	0.89 (+/- 0.06)	0.94 (+/- 0.03)
Naive Bayes	0.89 (+/- 0.35)	0.98 (+/- 0.02)	0.92 (+/- 0.22)
Random Forest	0.93 (+/- 0.23)	0.98 (+/- 0.01)	0.95 (+/- 0.13)
SVC Linear	0.91 (+/- 0.37)	0.92 (+/- 0.02)	0.90 (+/- 0.22)
SVC polynomial	0.92 (+/- 0.31)	0.95 (+/- 0.02)	0.93 (+/- 0.18)
SVC RBF	1.00 (+/- 0.01)	0.87 (+/- 0.05)	0.93 (+/- 0.02)
1D CNN	0.92 (+/- 0.00)	0.70 (+/- 0.00)	0.80 (+/- 0.00)

The evaluation using the different metrics measuring the performances of the selected algorithms are listed in Table 5.2. We can see that three tree based algorithms, Ada Boost, Decision Tree and Random Forest have the best performance in terms of the overall metric $F - Measure = 0.95$. These three algorithms also have good performances in Precision and Recall, especially for Random Forest, which has the highest $Recall = 0.98$. We can draw a conclusion that Tree based machine learning algorithms, especially for Random Forest, are the best classifiers in binary classification with "normal" and "MaR attack" on imbalanced dataset. Even though MLP and SVC RBF have the highest $Precision = 1$, the relatively lower Recall proves that they are not capable of detecting all possible MaR attacks. However, we observe also that 1D CNN has lower performance than the others with a F-Measure value of 0.8.

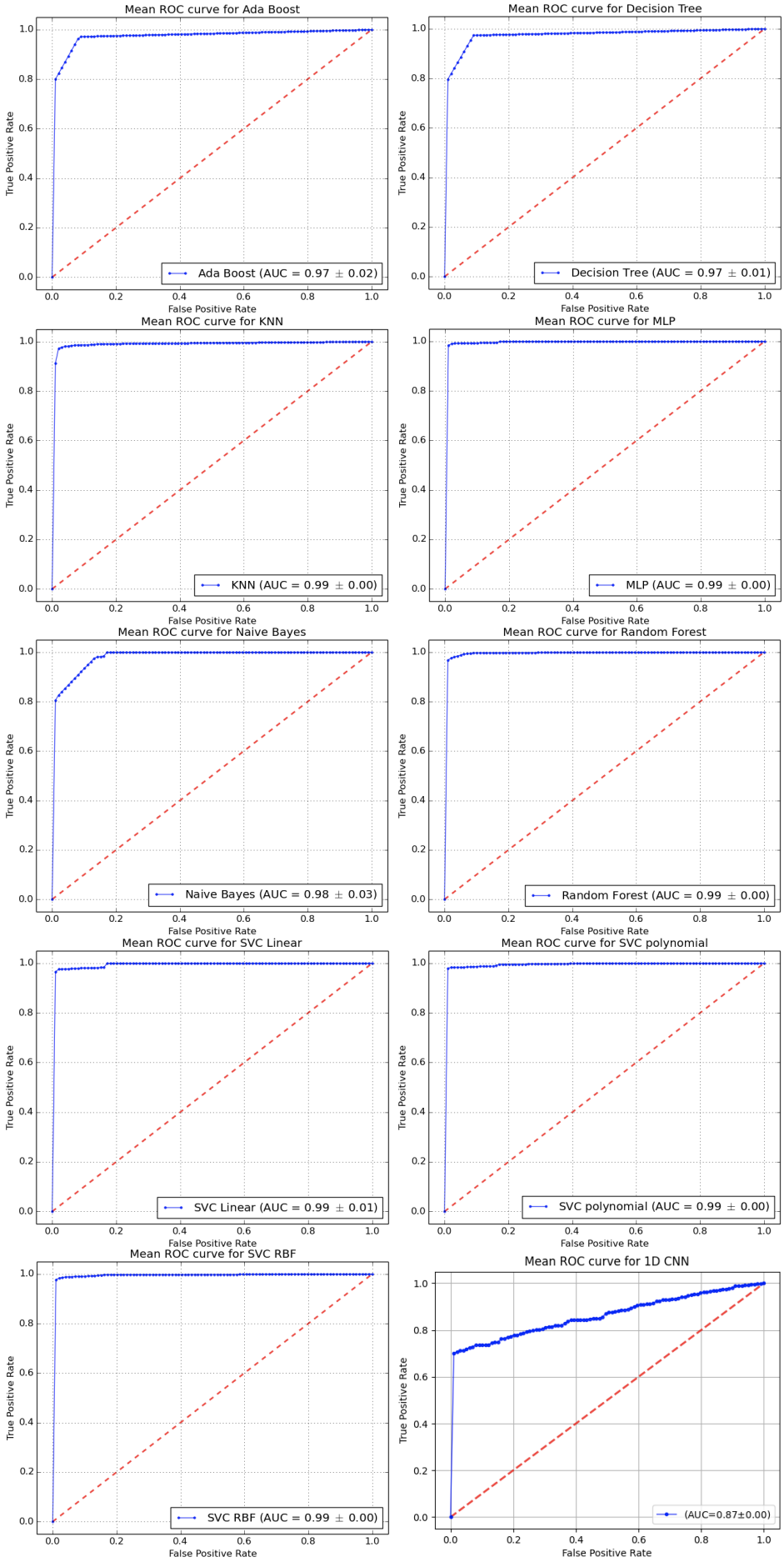

Figure 5.4: ROC curves of the selected classifiers for binary classification with "normal" and "MaR attack" on imbalanced dataset.

5.4.2 Classification of "normal" and "delay injection attack" with imbalanced dataset

The ROC curves of the selected classifiers for binary classification of "normal" and "delay injection attack" are shown in Fig. 5.5. We can find that three classifiers perform well with a high $AUC = 0.99$, which are Ada Boost, Decision Tree and Random Forest. It means that these three tree based models can distinguish samples labeled as "normal" and "MaR attack" with a probability of 99% on the imbalanced dataset. Note that all other classifiers including 1D CNN classifier have much lower AUC values compared to the previous classification problem (Normal and MaR attack) and also compared to the tree based methods.

Table 5.3: Performances of different classifiers for binary classification with "normal" and "delay injection attack" on imbalanced dataset

Classifier	Precision	Recall	F-Measure
Ada Boost	0.99 (+/- 0.02)	1.00 (+/- 0.01)	0.99 (+/- 0.01)
Decision Tree	0.98 (+/- 0.02)	1.00 (+/- 0.01)	0.99 (+/- 0.01)
KNN	0.96 (+/- 0.02)	0.21 (+/- 0.08)	0.34 (+/- 0.11)
MLP	1.00 (+/- 0.00)	0.21 (+/- 0.07)	0.34 (+/- 0.10)
Naive Bayes	0.77 (+/- 0.64)	0.22 (+/- 0.08)	0.33 (+/- 0.19)
Random Forest	0.99 (+/- 0.02)	0.98 (+/- 0.02)	0.99 (+/- 0.01)
SVC Linear	1.00 (+/- 0.00)	0.21 (+/- 0.08)	0.35 (+/- 0.10)
SVC polynomial	1.00 (+/- 0.00)	0.21 (+/- 0.08)	0.35 (+/- 0.10)
SVC RBF	1.00 (+/- 0.00)	0.21 (+/- 0.07)	0.34 (+/- 0.10)
1D CNN	1.00 (+/- 0.00)	0.01 (+/- 0.00)	0.02 (+/- 0.00)

The evaluation of the metrics measuring the performances of different algorithms are listed in Table 5.3. We can see that three tree based algorithms, Ada Boost, Decision Tree and Random Forest still have the best performance in terms of the overall metric $F - Measure = 0.99$. These three algorithms also have good performances in Precision and Recall. And all other classifiers have much lower performances. Our finding that Tree based machine learning algorithms are the best classifiers in binary classification with "normal" and "delay injection attack" on imbalanced dataset.

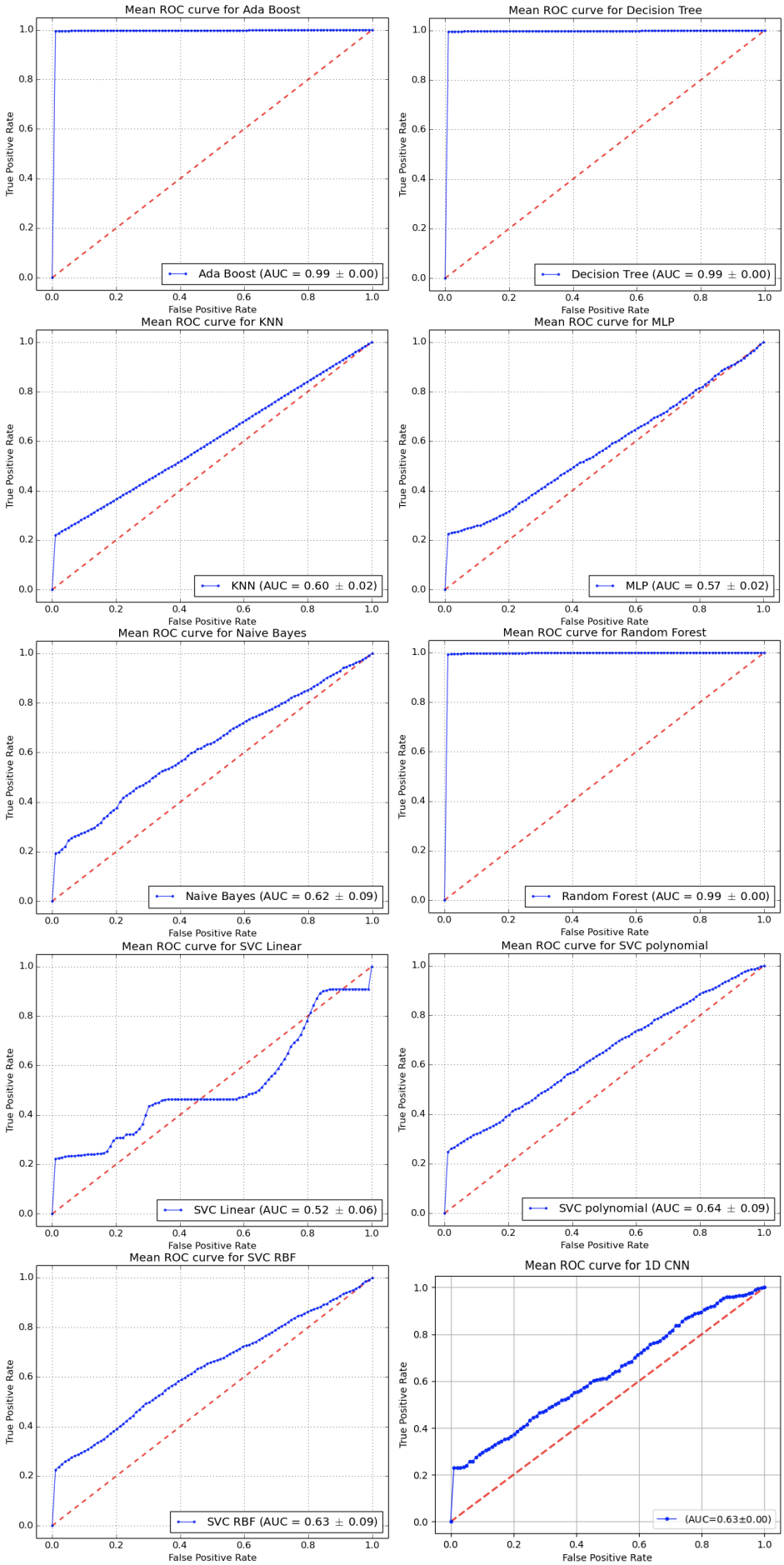

Figure 5.5: ROC curves of the selected classifiers for binary classification with "normal" and "delay injection attack" on imbalanced dataset.

5.4.3 Classification of "normal", "MaR attack" and "delay injection attack" with imbalanced dataset

In a multi-class classification problem, we compute the micro, macro, and weighted averages of the three metrics (Precision, Recall, and F-Measure) to evaluate the performances of each classifier [66]. To be more clear:

- *Micro-averaged:* all samples equally contribute to the final averaged metric.
- *Macro-averaged:* all samples equally contribute to the final averaged metric.
- *Weighted-averaged.:* each classes's contribution to the average is weighted by its size.

A high macro average indicates that the classifier performs well for each class, while a high micro average indicates that it performs well overall. Weighted average balances the effects of micro and macro averages.

The ROC curves of the selected classifiers for three-class classification with "normal", "MaR attack", and "delay injection attack" on imbalanced dataset are shown in Fig. 5.6. We find that the micro-average and macro-average ROC curves of the Ada Boost, Decision Tree and Random Forest classifiers have larger AUCs than other classifiers, especially for Random Forest, having the AUC for both micro-average and macro-average ROC curves, verifying that Random Forest has the best performance in detecting all three classes.

Table 5.4: Performances of different classifiers for multiclass classification with "normal", "MaR attack" and "delay injection attack" on imbalanced dataset

Classifier	Average Precision			Average Recall			Average F-Measure		
	Mirco	Macro	Weighted	Mirco	Macro	Weighted	Mirco	Macro	Weighted
Ada Boost	0.97 (+/- 0.05)	0.95 (+/- 0.07)	0.97 (+/- 0.03)	0.97 (+/- 0.05)	0.96 (+/- 0.02)	0.97 (+/- 0.05)	0.97 (+/- 0.05)	0.95 (+/- 0.05)	0.97 (+/- 0.04)
Decision Tree	0.97 (+/- 0.05)	0.95 (+/- 0.07)	0.97 (+/- 0.03)	0.97 (+/- 0.05)	0.96 (+/- 0.02)	0.97 (+/- 0.05)	0.97 (+/- 0.05)	0.95 (+/- 0.05)	0.97 (+/- 0.04)
KNN	0.86 (+/- 0.03)	0.85 (+/- 0.06)	0.85 (+/- 0.03)	0.86 (+/- 0.03)	0.69 (+/- 0.03)	0.86 (+/- 0.03)	0.86 (+/- 0.03)	0.71 (+/- 0.06)	0.83 (+/- 0.04)
MLP	0.87 (+/- 0.01)	0.92 (+/- 0.02)	0.88 (+/- 0.01)	0.87 (+/- 0.01)	0.70 (+/- 0.03)	0.87 (+/- 0.01)	0.87 (+/- 0.01)	0.72 (+/- 0.03)	0.84 (+/- 0.02)
Naive Bayes	0.85 (+/- 0.10)	0.87 (+/- 0.12)	0.86 (+/- 0.07)	0.85 (+/- 0.10)	0.70 (+/- 0.06)	0.85 (+/- 0.10)	0.85 (+/- 0.10)	0.70 (+/- 0.12)	0.82 (+/- 0.10)
Random Forest	0.97 (+/- 0.05)	0.96 (+/- 0.07)	0.98 (+/- 0.03)	0.97 (+/- 0.05)	0.96 (+/- 0.02)	0.97 (+/- 0.05)	0.97 (+/- 0.05)	0.96 (+/- 0.06)	0.97 (+/- 0.05)
SVC Linear	0.84 (+/- 0.10)	0.88 (+/- 0.11)	0.86 (+/- 0.07)	0.84 (+/- 0.10)	0.67 (+/- 0.05)	0.84 (+/- 0.10)	0.84 (+/- 0.10)	0.67 (+/- 0.11)	0.80 (+/- 0.09)
SVC polynomial	0.82 (+/- 0.08)	0.75 (+/- 0.10)	0.81 (+/- 0.05)	0.82 (+/- 0.08)	0.62 (+/- 0.07)	0.82 (+/- 0.08)	0.82 (+/- 0.08)	0.66 (+/- 0.06)	0.80 (+/- 0.06)
SVC RBF	0.86 (+/- 0.00)	0.91 (+/- 0.02)	0.88 (+/- 0.01)	0.86 (+/- 0.00)	0.68 (+/- 0.01)	0.86 (+/- 0.00)	0.86 (+/- 0.00)	0.71 (+/- 0.02)	0.83 (+/- 0.01)
1D CNN	0.88 (+/- 0.00)	0.92 (+/- 0.01)	0.89 (+/- 0.00)	0.88 (+/- 0.00)	0.71 (+/- 0.00)	0.88 (+/- 0.00)	0.88 (+/- 0.00)	0.74 (+/- 0.00)	0.85 (+/- 0.00)

The evaluation metrics measuring the performances of different machine learning algorithms are listed in Table 5.4 where for each metric we provide its micro, macro and weighted values. Besides Random Forest, Decision Tree and AdaBoost also outperform the rest of the classifiers in all three metrics. The rest of the classifiers have poor performances in both average Precision and average Recall, which means they can not only detect most of the positive samples under attack but also generate lots of false alarms.

5.5 Evaluation of ML algorithms with balanced dataset

In this section, we use the balanced dataset to train and test the selected machine learning and deep learning algorithms. In the balanced dataset the number of "normal" samples is close to those of the samples of "MaR attack" and "delay injection attack". Such balanced dataset is usually used to get more accurate models compared to imbalanced dataset, because when one set of classes dominate over another set of classes, the machine learning model will be more biased towards the majority class. Thus it causes poor classification of the minority classes.

We evaluated the ML algorithms using three classification problems to detect separately normal from MaR samples, normal from delay injection attack, and jointly the three behaviours in a single dataset. The evaluation results are provided the following subsections.

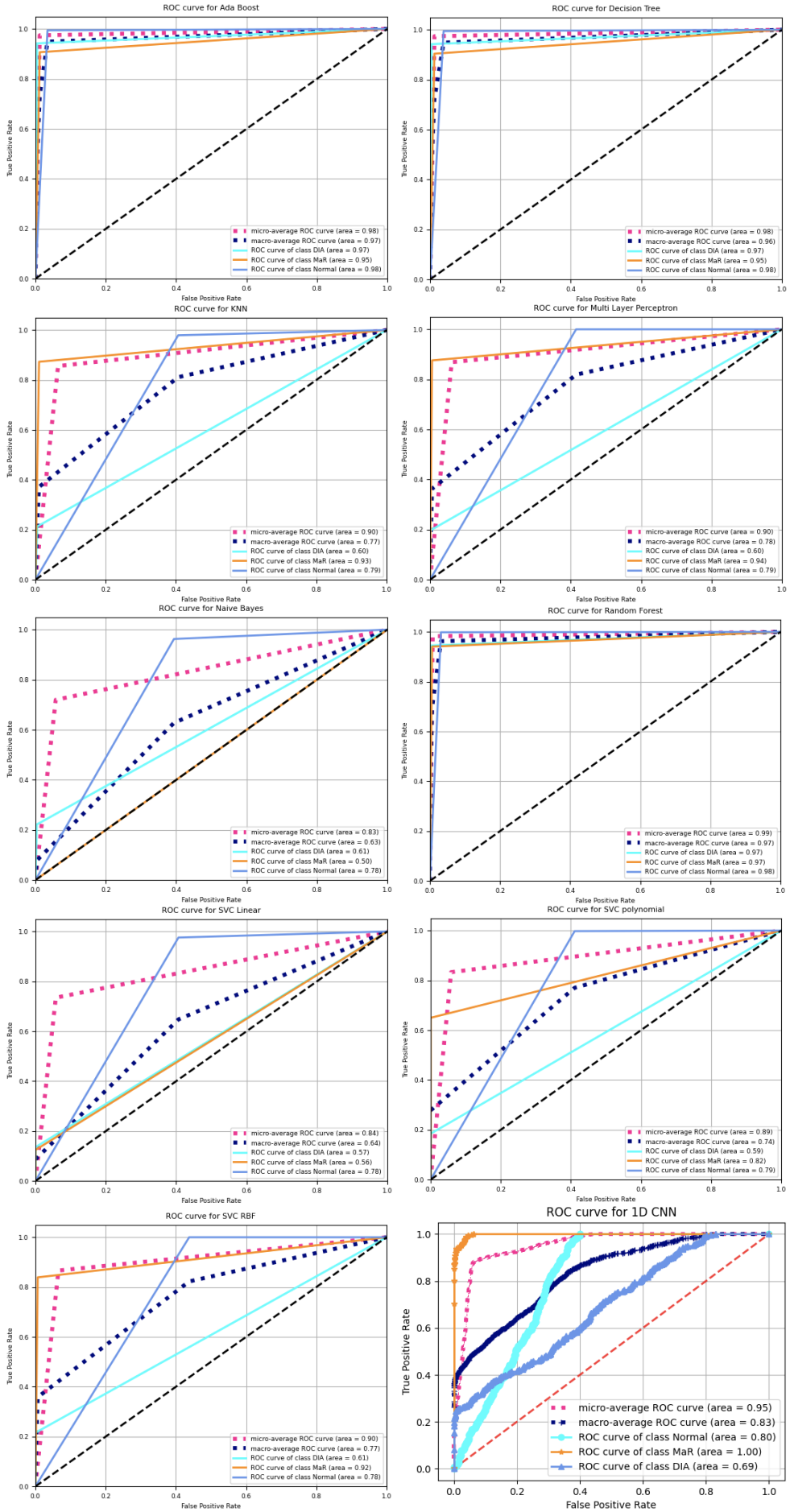

Figure 5.6: ROC curves of the selected classifiers for multiclass classification with "normal", "MaR attack" and "delay injection attack" on imbalanced dataset.

5.5.1 Classification of "normal" and "MaR attack" with balanced dataset

The ROC curves of the selected classifiers for binary classification with "normal" and "MaR attack" are shown in Fig. 5.7. We can find that many classifiers perform well with a high $AUC = 0.99$. These classifiers are MLP, Naive Bayes, Random Forest, SVC Linear, SVC polynomial, SVC RBF and 1D CNN. It means that these models can distinguish samples labeled as "normal" and "MaR attack" with a probability of 99%. Note that all other classifiers have also have high AUC values greater than 97%, which means all selected classifiers can handle the classification problem with good accuracy.

Table 5.5: Performances of different classifiers for binary classification with "normal" and "MaR attack" on balanced dataset.

Classifier	Precision	Recall	F-Measure
Ada Boost	0.97 (+/- 0.04)	0.97 (+/- 0.02)	0.97 (+/- 0.03)
Decision Tree	0.97 (+/- 0.04)	0.97 (+/- 0.02)	0.97 (+/- 0.03)
KNN	0.97 (+/- 0.04)	0.91 (+/- 0.02)	0.94 (+/- 0.03)
MLP	1.00 (+/- 0.01)	0.94 (+/- 0.03)	0.97 (+/- 0.02)
Naive Bayes	0.95 (+/- 0.08)	0.94 (+/- 0.05)	0.95 (+/- 0.06)
Random Forest	0.98 (+/- 0.04)	0.97 (+/- 0.01)	0.98 (+/- 0.02)
SVC Linear	0.99 (+/- 0.03)	0.92 (+/- 0.09)	0.95 (+/- 0.05)
SVC polynomial	0.99 (+/- 0.01)	0.95 (+/- 0.02)	0.97 (+/- 0.01)
SVC RBF	1.00 (+/- 0.00)	0.89 (+/- 0.02)	0.94 (+/- 0.01)
1D CNN	0.97 (+/- 0.00)	0.75 (+/- 0.00)	0.85 (+/- 0.00)

The evaluation metrics measuring the performances of different algorithms are listed in Table 5.5. We can see that Random Forest has the best performance in terms of the overall metric $F - Measure = 0.98$. All traditional machine learning classifiers have also good performances in Precision, Recall and F-Measure. Note that 1D CNN has the lowest $F - Measure = 0.85$ and lowest $Recall = 0.75$, which means that traditional machine learning outperforms deep learning in binary classification with "normal" and "MaR attack" on balanced dataset.

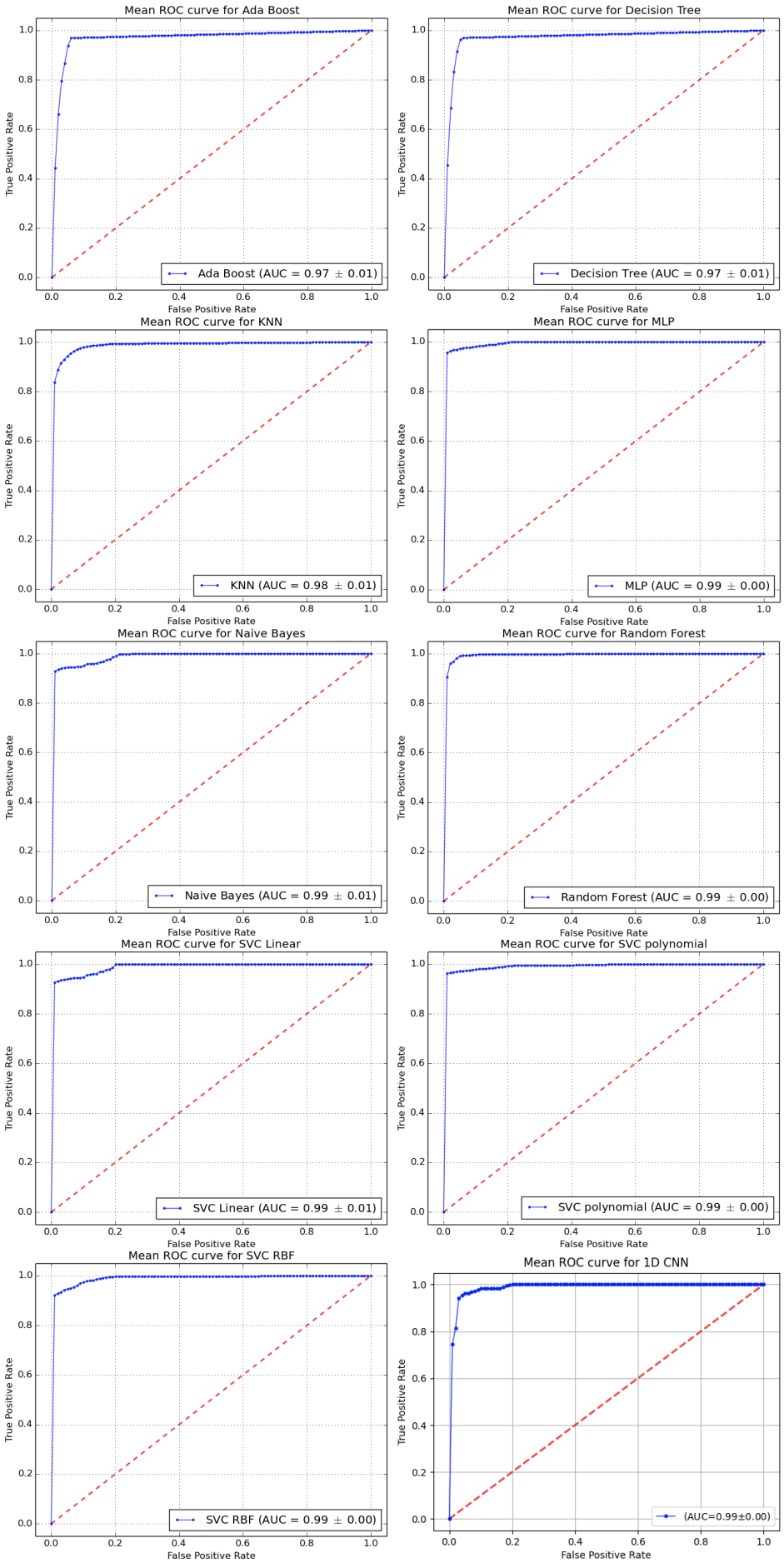

Figure 5.7: ROC curves of the selected classifiers for binary classification with "normal" and "MaR attack" on balanced dataset.

5.5.2 Classification of "normal" and "delay injection attack" on balanced dataset

The ROC curves of the selected classifiers for binary classification with "normal" and "delay injection attack" are shown in Fig. 5.8. We can find that three classifiers perform well with a high $AUC = 0.99$, which are Ada Boost, Decision Tree and Random Forest. It means that these three tree based models can distinguish samples labeled as "normal" and "delay injection attack" with a probability of 99% on the balanced dataset. Note that all other classifiers including 1D CNN classifier have much lower AUC values.

Table 5.6: Performances of different classifiers for binary classification with "normal" and "delay injection attack" on balanced dataset

Classifier	Precision	Recall	F-Measure
Ada Boost	0.99 (+/- 0.02)	1.00 (+/- 0.00)	0.99 (+/- 0.01)
Decision Tree	0.99 (+/- 0.02)	0.99 (+/- 0.01)	0.99 (+/- 0.01)
KNN	0.62 (+/- 0.06)	0.40 (+/- 0.07)	0.49 (+/- 0.05)
MLP	0.99 (+/- 0.03)	0.22 (+/- 0.08)	0.36 (+/- 0.10)
Naive Bayes	0.98 (+/- 0.06)	0.22 (+/- 0.08)	0.36 (+/- 0.11)
Random Forest	0.99 (+/- 0.01)	0.98 (+/- 0.03)	0.98 (+/- 0.02)
SVC Linear	1.00 (+/- 0.00)	0.21 (+/- 0.07)	0.35 (+/- 0.10)
SVC polynomial	1.00 (+/- 0.00)	0.21 (+/- 0.08)	0.35 (+/- 0.10)
SVC RBF	0.88 (+/- 0.16)	0.23 (+/- 0.09)	0.36 (+/- 0.11)
1D CNN	1.00 (+/- 0.00)	0.20 (+/- 0.00)	0.33 (+/- 0.00)

The evaluation metrics measuring the performances of different algorithms are listed in Table 5.6. We can see that three tree based algorithms, Ada Boost, Decision Tree and Random Forest have the best performance in terms of the overall metric $F - Measure = 0.99$. These three algorithms also have good performances in Precision and Recall. And all other classifiers have much poorer performances. We can draw a conclusion that Tree based machine learning algorithms are the best classifiers in binary classification with "normal" and "delay injection attack" on balanced dataset.

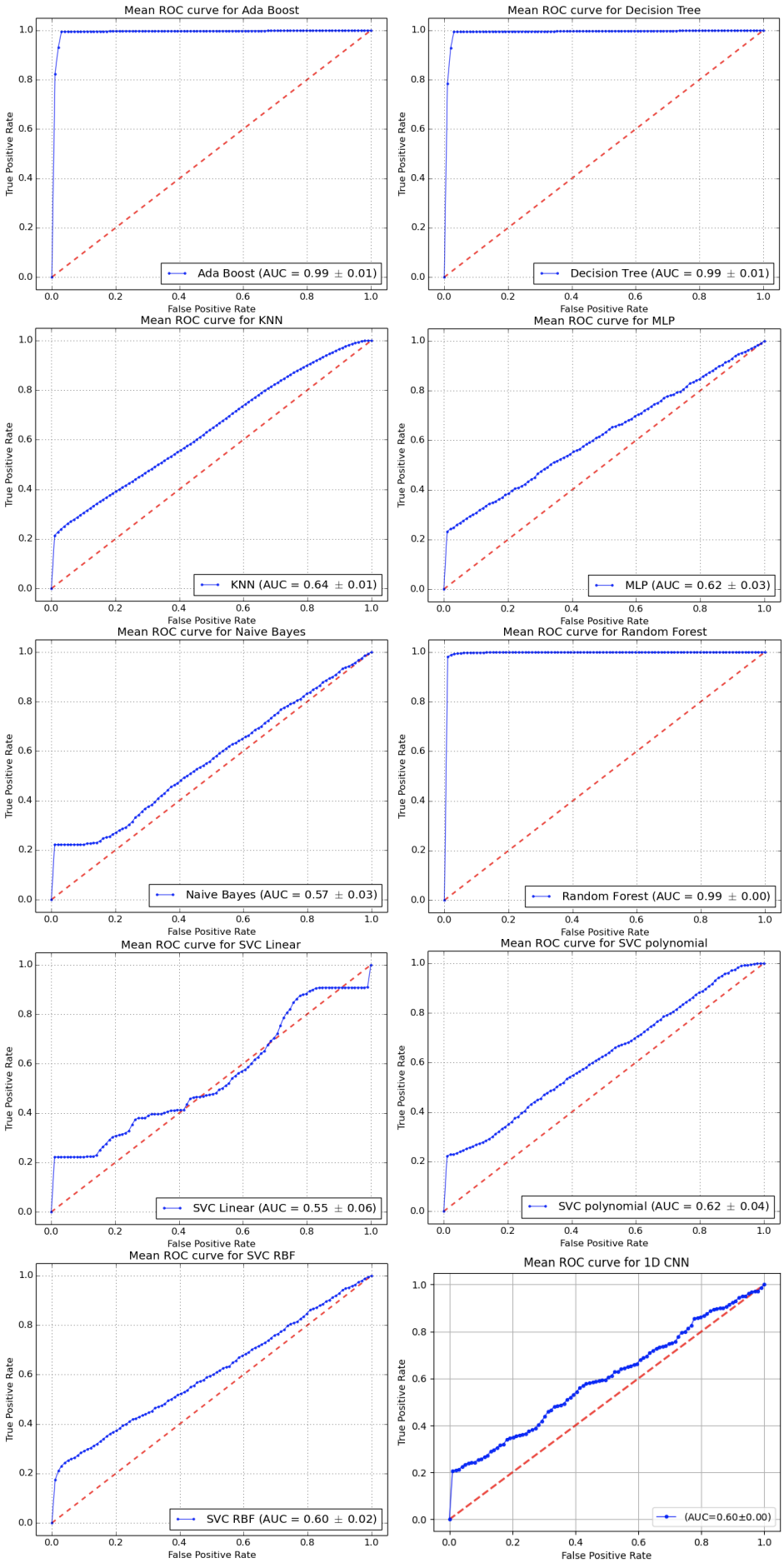

Figure 5.8: ROC curves of the selected classifiers for binary classification with "normal" and "delay injection attack" on balanced dataset.

5.5.3 Classification of "normal", "MaR attack" and "delay injection attack" on balanced dataset

The ROC curves of the selected classifiers for three-class classification with "normal", "MaR attack", and "delay injection attack" on balanced dataset are shown in Fig. 5.9. We can find that the micro-average and macro-average ROC curves of the Ada Boost, Decision Tree and Random Forest classifiers have larger AUCs than other classifiers, especially for Random Forest, having the AUC for both micro-average and macro-average ROC curves, verifying that Random Forest has the best performance in detecting all three classes.

Table 5.7: Performances of different classifiers for multiclass classification with "normal", "MaR attack" and "delay injection attack" on balanced dataset

Classifier	Average Precision			Average Recall			Average F-Measure		
	Mirco	Macro	Weighted	Mirco	Macro	Weighted	Mirco	Macro	Weighted
Ada Boost	0.95 (+/- 0.02)	0.95 (+/- 0.02)	0.95 (+/- 0.02)	0.95 (+/- 0.02)	0.95 (+/- 0.02)	0.95 (+/- 0.02)	0.95 (+/- 0.02)	0.95 (+/- 0.02)	0.95 (+/- 0.02)
Decision Tree	0.95 (+/- 0.01)	0.95 (+/- 0.01)	0.95 (+/- 0.01)	0.95 (+/- 0.01)	0.95 (+/- 0.01)	0.95 (+/- 0.01)	0.95 (+/- 0.01)	0.95 (+/- 0.01)	0.95 (+/- 0.01)
KNN	0.67 (+/- 0.02)	0.67 (+/- 0.03)	0.67 (+/- 0.03)	0.67 (+/- 0.02)	0.67 (+/- 0.02)	0.67 (+/- 0.02)	0.67 (+/- 0.02)	0.67 (+/- 0.03)	0.67 (+/- 0.03)
MLP	0.73 (+/- 0.02)	0.82 (+/- 0.07)	0.81 (+/- 0.06)	0.73 (+/- 0.02)	0.71 (+/- 0.02)	0.73 (+/- 0.02)	0.73 (+/- 0.02)	0.67 (+/- 0.05)	0.68 (+/- 0.05)
Naive Bayes	0.70 (+/- 0.06)	0.79 (+/- 0.05)	0.79 (+/- 0.05)	0.70 (+/- 0.06)	0.69 (+/- 0.06)	0.70 (+/- 0.06)	0.70 (+/- 0.06)	0.64 (+/- 0.07)	0.65 (+/- 0.06)
Random Forest	0.96 (+/- 0.02)	0.96 (+/- 0.02)	0.96 (+/- 0.02)	0.96 (+/- 0.02)	0.96 (+/- 0.02)	0.96 (+/- 0.02)	0.96 (+/- 0.02)	0.96 (+/- 0.02)	0.96 (+/- 0.02)
SVC Linear	0.70 (+/- 0.04)	0.82 (+/- 0.02)	0.81 (+/- 0.02)	0.70 (+/- 0.04)	0.68 (+/- 0.04)	0.70 (+/- 0.04)	0.70 (+/- 0.04)	0.62 (+/- 0.05)	.63 (+/- 0.05)
SVC polynomial	0.65 (+/- 0.05)	0.67 (+/- 0.06)	0.67 (+/- 0.06)	0.65 (+/- 0.05)	0.63 (+/- 0.05)	0.65 (+/- 0.05)	0.65 (+/- 0.05)	0.60 (+/- 0.04)	0.61 (+/- 0.04)
SVC RBF	0.70 (+/- 0.02)	0.77 (+/- 0.06)	0.77 (+/- 0.06)	0.70 (+/- 0.02)	0.68 (+/- 0.02)	0.70 (+/- 0.02)	0.70 (+/- 0.02)	0.65 (+/- 0.03)	0.65 (+/- 0.03)
1D CNN	0.74 (+/- 0.00)	0.82 (+/- 0.02)	0.81 (+/- 0.02)	0.74 (+/- 0.00)	0.72 (+/- 0.00)	0.74 (+/- 0.00)	0.74 (+/- 0.00)	0.69 (+/- 0.00)	0.70 (+/- 0.00)

The evaluation metrics measuring the performances of different machine learning algorithms are listed in Table 5.7. Besides Random Forest, Decision Tree and AdaBoost also outperform the rest of the classifiers in all three metrics. The rest of the classifiers have poor performances in both average Precision and average Recall, which means they can not only detect most of the positive samples under attack but also generate lots of false alarms.

5.6 Discussion and Analysis

As shown in the previous sections, we find that Decision Tree, Random Forest (bagged trees), and Adaboost (boosted trees) outperform the other selected classification models. These algorithms can not only detect attack data packets but can also distinguish different attack types with high performance. We find also that the deep learning algorithm 1D CNN does not provide good performance and it is less performant than the others. Regarding the effect of balanced and imbalanced datasets, all the algorithms provide close performances for both of them. A summary of the best performing algorithms for each classification problem and for both balanced and imbalanced datasets is shown in Tables 5.9 and 5.8.

Besides, we evaluated the running time of different learning and testing tasks for each algorithm as shown in Table 5.10 and Table 5.11. Note that we measured the running time on a MacBook Pro PC with 2.9 GHz Dual-Core Intel Core i5 CPU and 8 GB 2133 MHz LPDDR3 RAM. Table 5.10 and Table 5.11 show that imbalanced dataset requires more time to train and test than balanced dataset. We can easily observe that the SVC polynomial classifier has larger running time in all cases, which means it is not a good choice if the DG units are resource limited in the microgrid system. SVC RBF has the least time cost in all three SVM based classifiers. The three tree based classifiers, Decision Tree, AdaBoost, and Random Forest have very fast training and testing speeds compared to other classifiers, where Decision Tree and AdaBoost are slightly faster than Random Forest.

The obtained results show that the learning models of the three algorithms: Decision Tree, Random Forest and AdaBoost provide good performance when training and validating either

	Classification problem		
	Normal and MaR	Normal and Delay Injection	Three classes
Ada Boost	✓ (0.95)	✓ (0.99)	✓ (0.95)
Decision Tree	✓ (0.95)	✓ (0.99)	✓ (0.95)
KNN			
MLP			
Naive Bayes			
Random Forest	✓ (0.95)	✓ (0.99)	✓ (0.96)
SVC Linear			
SVC polynomial			
SVC RBF			
1D CNN			

Table 5.8: Summary of best performing ML algorithms based on the F-Measure for each classification problem with imbalanced dataset.

	Classification problem		
	Normal and MaR	Normal and Delay Injection	Three classes
Ada Boost	✓ (0.97)	✓ (0.99)	✓ (0.95)
Decision Tree	✓ (0.97)	✓ (0.99)	✓ (0.95)
KNN			
MLP	✓ (0.97)		
Naive Bayes			
Random Forest	✓ (0.98)	✓ (0.98)	✓ (0.96)
SVC Linear			
SVC polynomial	✓ (0.97)		
SVC RBF			
1D CNN			

Table 5.9: Summary of best performing ML algorithms based on the F-Measure for each classification problem with balanced dataset.

	Running time (seconds) of training and testing on balanced dataset		
	Normal and MaR	Normal and Delay Injection	Three classes
Ada Boost	0.415	0.571	0.081
Decision Tree	0.509	0.253	0.098
KNN	1.239	0.747	0.334
MLP	20.442	11.763	5.815
Naive Bayes	0.485	0.232	0.047
Random Forest	2.508	1.614	0.617
SVC Linear	4.940	164.892	135.249
SVC polynomial	12.489	3632.050	1621.737
SVC RBF	6.177	14.180	6.581
1D CNN	7.861	6.848	11.257

Table 5.10: Summary of the running time of training and testing for each classifier on balanced dataset.

with balanced or imbalanced datasets, and have also low running time. In a last experiment, we evaluated the robustness of these algorithms when training them with a balanced dataset and testing them with an imbalanced dataset. Such situation is more realistic, since the data packets transmitted in the real-life microgrid systems are imbalanced, i.e., there are many more

	Running time (seconds) of training and testing on imbalanced dataset (s)		
	Normal and MaR	Normal and Delay Injection	Three classes
Ada Boost	0.576	0.574	0.167
Decision Tree	0.410	0.386	0.141
KNN	5.566	4.481	1.726
MLP	41.014	28.703	16.448
Naive Bayes	0.287	0.252	0.070
Random Forest	3.243	3.791	1.298
SVC Linear	10.639	1054.497	596.080
SVC polynomial	22.829	21820.063	4384.425
SVC RBF	19.768	155.710	26.540
1D CNN	16.920	16.455	24.912

Table 5.11: Summary of the running time of training and testing for each classifier on imbalanced dataset.

normal data packets than data packets under attacks to be processed by DGs. The results of this experiment are shown in Table 5.12. We find that these three tree based algorithms still have the largest macro average F-Measures in all three classification problems, especially for the binary classification of "normal" and "delay injection" and multiclass classification. It proves that these three classifiers trained from a balanced dataset can successfully deal with imbalanced data in real-life microgrid systems. The training models are close enough to the behavior of the microgrid system and the studied attacks.

	Macro average F-Measure		
	Normal and MaR	Normal and Delay Injection	Three classes
Ada Boost	1.00	0.99	0.99
Decision Tree	1.00	0.99	0.99
KNN	0.97	0.61	0.71
MLP	0.98	0.65	0.73
Naive Bayes	0.97	0.64	0.71
Random Forest	1.00	0.99	0.99
SVC Linear	0.97	0.64	0.69
SVC polynomial	0.98	0.64	0.67
SVC RBF	0.97	0.63	0.72
1D CNN	0.76	0.35	0.84

Table 5.12: Summary of the Macro average F-Measure values of different classifiers for training on balanced dataset and testing on imbalanced dataset.

To summarize the experimental results, we consider that the three tree based algorithms, Decision Tree, Ada Boost and Random Forest as the most suitable machine learning algorithms to design the attack detection module in Fig. 4.2 among all selected classifiers because of the following reasons:

- Their classification results do not make a large difference between imbalanced and balanced dataset, and classifiers trained from a balanced dataset can successfully deal with imbalanced data in real-life microgrid systems, which proves its training model is close enough to the behaviour of the microgrid system and also the studied attacks.

- They have better classification performances than the other machine learning and deep learning algorithms in both binary and multi-class classification results by having the largest Precision, Recall, F-Measure and AUC values. Among the three classifiers, Random Forest slightly outperforms Decision Tree and Ada Boost.
- They can handle various types of features and need very little preprocessing. This property ensures the attack detector to be capable of handling sophisticated attack scenarios involving various types of features.
- They have fast prediction/training speed, and prediction speed is significantly faster than training speed because generated classifiers are saved for future uses. It is a crucial property for attack detection when DG units have limited resources. From Table 5.10 and Table 5.11, Decision Tree and Ada Boost are faster than Random Forest in training and testing. However, in real-life cases, Random Forest could also be faster to train than Decision Tree because only a subset of features in this model is utilized, so it works quickly with high dimensional data. Random Forest is also parallelizable, so the training process can be split into multiple machines to run, resulting in faster computation time than sequential AdaBoost.

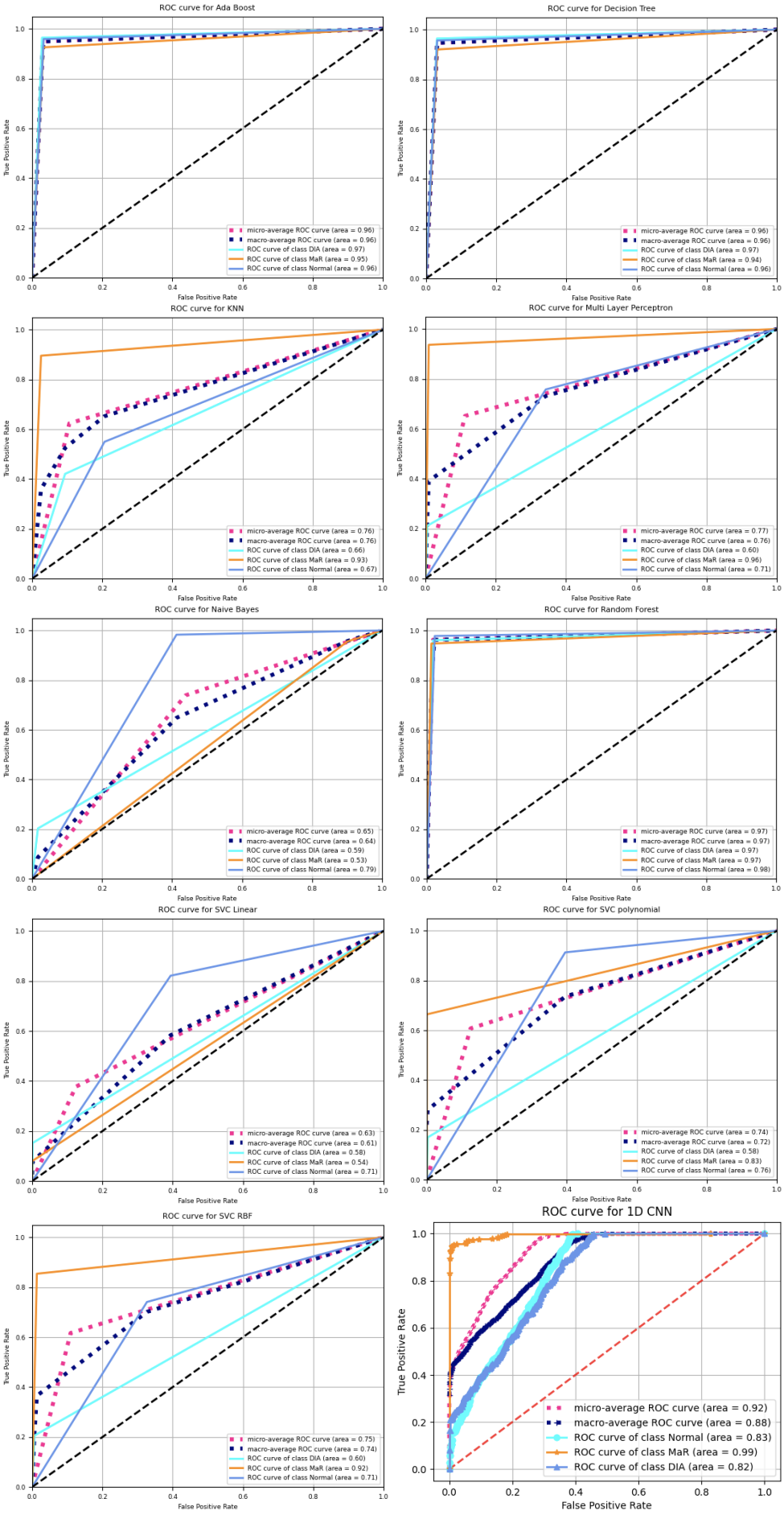

Figure 5.9: ROC curves of the selected classifiers for multiclass classification with "normal", "MaR attack" and "delay injection attack" on balanced dataset.

Chapter 6

Conclusions and Future Work

Contents

This thesis considered the security and attack detection of distributed controlled microgrid systems. Modern low-voltage microgrid systems are relying on distributed and cooperative control approaches to guarantee safe and reliable operations. However, many sophisticated cyber-attacks can target these systems, deceive their traditional detection methods and cause a severe impact on the power infrastructure. In this thesis, novel distributed control models of microgrids are presented. Attack models targeting the microgrid control system and their related impact analysis are discussed. A secure control framework with an attack detection module based on machine learning techniques is proposed. To validate the effectiveness of this framework, datasets considering false data injection (FDI) and denial of service (DoS) attacks are collected from the designed hardware platform that models the microgrid system. Different machine learning and deep learning classifiers are compared and analyzed based on the datasets collected.

A brief summary of the thesis contributions and possible future research directions are discussed below.

6.1 Conclusions

This thesis addressed several topics concerning security and attack detection of distributed controlled microgrids. The main contributions are as follows.

- **Bridging Microgrid Control and Cyber Security.** In this study, we study some of the key principles of the microgrid structures, including the control mechanism and the

communication networks used in the research. Next, we include a short overview of recent developments in the area of microgrid control systems. Secondly, we study cyber-physical protection topics, and summarize a number of cyber-physical attacks in different categories. Finally, we offer an extensive overview of the recent relevant studies on cyber-secure control systems for attack detection schemes.

- **Attack Modelling and Impact Analysis.** In this study, we present a new attack, Measurement as Reference (MaR), directed at the distributed microgrid control system. The assailant introduces a reference voltage variance by manipulating the synchronization data shared between the distributed controllers in this attack, which we instantiated on voltage control. By simulation and theoretical analysis, we demonstrate the result of this synchronization attack. If the system does not synchronize to the proper location given by the secondary control system, the control theory tools extract the maximum voltage deviation introduced by this attack. However, other threats, such as delaying injections and denial of service attacks, also affect the operations of this system, are also triggered by a distributed controlled system. The detection of the attacks is still challenging because of the stealthy nature of some of them, where the deviations from nominal values are very low. This challenge is the inspiration behind this chapter, which proposes a better way of detecting such attacks, including the MaR attack, based on machine learning algorithms.

- **Design of a Secure Framework with Attack Detection.** We study various groups of supervised machine learning algorithms to design a comprehensive attack resilient control framework, in particular to detect FDI and DoS attacks in microgrids. We model and generalize both FDI and DoS attacks in distributed and cooperative control microgrid systems. We also give a general system stability condition analysis under FDI and DoS attacks. We propose a secure control framework with attack detection and response modules relying on machine learning and deep learning algorithms. Various groups of supervised machine learning and deep learning algorithms are studied to design the attack detection module against FDI attacks and DoS attacks under well-chosen performance metrics. Considering it is difficult to find publicly available datasets, we build an experimental platform emulating the distributed and cooperative controlled microgrid systems, and implement two typical FDI and DoS attacks hard to be detected by traditional residual comparison detectors. The two attacks are: measurement as referent (MaR) attack as typical FDI attack and delay injection attack as typical DoS attack. Balanced and imbalanced dataset are collected to under normal and attacked conditions to be further analyzed.

- **Verification and Analysis of Various Attack Detection Algorithms.** We validate the attack detection methods to be used by our secure control framework. The two typical attacks targeting the communication links, namely measurement as referent (MaR) and delay injection attack are implemented on the hardware testbed described in Chapter 4. Before studying the machine learning and deep learning based detection algorithms, we firstly show the incompetence of traditional threshold comparison method in attack detection for the distributed microgrid system. Experimental results show that traditional threshold comparison method performs poorly in terms of precision, recall and accuracy. To validate the performance of machine learning and deep learning-based attack detection, we give a detailed hyper-parameter tuning methodology to get the best hyper-parameters used for classification. We collect datasets from the platform and compare the performances of different machine learning and deep learning algorithms to detect MaR and delay injection attacks. The classification problem will be divided into three categories:

binary classification of "normal" and "MaR attack", binary classification of "normal" and "delay injection attack", and multi-class classification of "normal" , "MaR attack" and "delay injection attack". We give a detailed analyzation for the three classification problems on balanced and imbalanced dataset, respectively. By collecting datasets from an experimental microgrid platform running these attacks, we find that tree-based machine learning models, especially Random Forest, have the best performance in binary and three-class classification problems. In fact, Random Forest is quite a general machine learning algorithm to design the attack detection module proposed in our approach. It can handle various types of features and need very little preprocessing and has quick prediction/-training speed. Compared to single Decision Tree and Adaboost, it is robust to outliers and nonlinear data; it can also handle imbalanced datasets, has low bias and moderate variance, and limits overfitting without substantially increasing error due to bias.

6.2 Future Work

Several steps remain to be taken concerning the opportunities for future study (due to the work begun in this thesis. It is worth noting that cyber-physical systems have a wide variety of issues to overcome. This thesis dealt with some of the security issues in the subject with restricted scope, giving specific attention to the identification of malicious actions concealed or mixed with faults and accidents. However, cyber-physical networks include many other aspects that need to be tackled together in order to enhance their resistance to threats and misuse.

There are also a variety of steps to be taken with respect to possible research perspectives (as a result of the study begun in this thesis). It should be remembered that cyber-physical devices face many obstacles. In this thesis, certain security issues have been dealt with within a restricted spectrum, with the identification of disruptive acts concealed or in conjunction with errors and injuries being a specific concern. Nonetheless, there are also other cyber-physical devices that need to be managed together to enhance their ability to strike and exploit.

As future works, we will study the scalability and transferability of our methodology for cyber-security problems in more complex and close-to-reality cyber-physical systems, like smart-home, smart-city, etc. More advanced machine learning techniques like deep learning and more sophisticated attack scenarios will also be explored.

Moreover, a bigger scope of security issues in terms of cyber-physical system could be investigated, e.g., the privacy of data transmitted among the communication links. The threat scenarios discussed in the study consisted mainly of data deception and denial of service attacks. In these cases, the attacker was attempting to interrupt the device by tampering with the data of the sensor and actuator. In addition to such situations, disclosure attacks that collect private information from the plant and control algorithms are also important. Methodologies for resolving privacy while maintaining an appropriate degree of monitoring and evaluation of the success of disclosure attacks are needed.

Résumé de la thèse en français

Modélisation et détection des attaques dans les systèmes de contrôle distribué et coopératif de micro-réseaux électriques

1 Introduction

La sécurité et la fiabilité des opérations sont considérées comme des questions critiques dans les systèmes de réseaux électriques modernes. Les nouvelles technologies de l'information et de la communication (TIC) et diverses générations réparties (DG), par exemple les générateurs photovoltaïques (PV) et les éoliennes, sont intégrées dans le réseau électrique. Ainsi, un nouveau paradigme de production d'électricité, dans lequel un parc de production mixte se compose de centrales conventionnelles centralisées et d'unités de production distribuées à des niveaux de tension inférieurs, est en train d'émerger. Cependant, il devient plus difficile d'exploiter les réseaux électriques de manière fiable et résiliente dans ce nouveau paradigme. Le concept de micro-réseaux est développé pour relever ces défis en accélérant l'intégration locale des sources d'énergie renouvelables. Les micro-réseaux utilisent une structure de contrôle hiérarchique pour contrôler et réguler, en particulier la tension électrique, y compris les niveaux primaire et secondaire de la structure de contrôle. Le contrôle primaire utilise conventionnellement des régulateurs de statisme locaux sur les ressources énergétiques distribuées (DER) pour maintenir la stabilité de la tension du micro-réseau après l'îlotage. Le contrôle secondaire du micro-réseau, qui régule la tension pour synchroniser la tension moyenne du micro-réseau avec le point de consigne nominal, peut être mis en œuvre par une structure centralisée. Ces micro-réseaux évoluent vers des systèmes cyber physiques (CPS) avec des réseaux de contrôle et de communication basés sur des solutions logicielles [4]. En revanche, à cause de cette intégration de ces solutions logicielles et de communication, ces réseaux sont sujets à de multiples cyber-attaques, en particulier sur leurs système de contrôle [63].

Dans cette thèse, notre contribution porte sur l'analyse et la modélisation de ces attaques dans des environnements de contrôle distribués associés à ces mircro-réseaux électriques. Dans la section 2, nous étudions l'impact d'une cyber attaque ciblant un réseau de contrôle pour altérer sa synchronisation. Dans le section 3, nous présentons une solution pour détecter et remédier à ces attaques, en particulier celles d'injection de fausses données et de déni de service. Dans la section 4, nous évaluons plusieurs algorithmes d'apprentissage automatique pour détecter ces attaques d'une façon précise. Enfin, une conclusion générale et les perspectives de recherche sont présentées.

2 Analyse de l'impact des attaques de synchronisation

Nous considérons, un système de contrôle distribué basé sur trois niveaux : primaire, secondaire et contrôle tertiaire. Nous estimons que ce schéma de contrôle est plus réaliste et plus fiable que l'architecture de contrôle centralisée, en raison de son meilleur support pour l'intégration évolutive des charges locales et des l'intégration de la production distribuée. Sur la base de ce modèle de contrôle, nous considérons des cyber attaques qui peuvent viser la communication entre les entités distribuées de génération (DG), et nous utilisons des méthodes d'évaluation des risques pour analyser quantitativement les l'impact de ces attaques sur la déviation de la tension au niveau du contrôle primaire et de la tension de référence de la synchronisation au niveau du contrôle secondaire. La Figure 1 présente le modèle du système considéré pour analyse l'impact d'une attaque de synchronisation sur le réseau de communication reliant les différentes DGs.

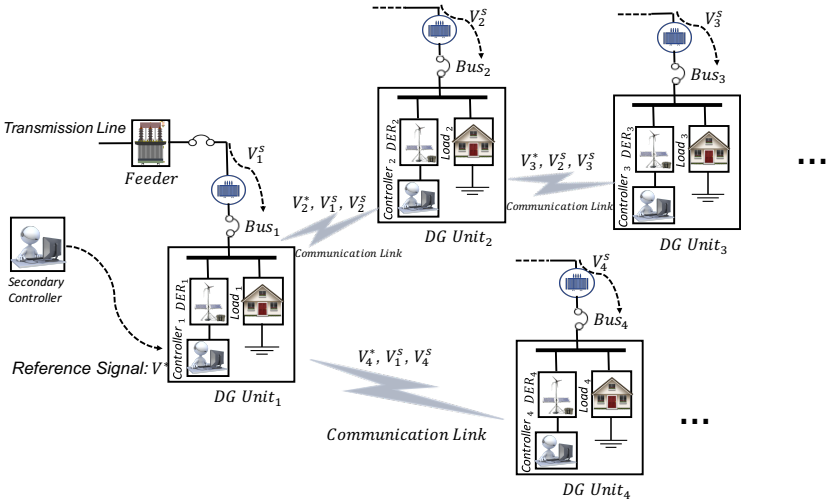

Figure 1: Un système de contrôle de micro-réseau avec des contrôleurs primaires et secondaires. Des réseaux de communication clairsemés sont utilisés pour l'échange de données entre les unités de production d'électricité voisines, par exemple pour transmettre des consignes de référence et des mesures (indiqués par l'exposant s).

Nous définissons une nouvelle attaque de synchronisation, nommée *Measurement as Reference* (*MaR*), ciblant ce système qui permet à un attaquant de remplacer la consigne par une mesure, en particulier celles pour réguler la tension électrique. Dans cette situation, l'attaquant remplace une consigne V_{i+1}^* d'un noeud $i+1$ avec une valeur de mesure V_i^s d'un noeud i. L'objectif de cette attaque est de provoquer des fluctuations de tension et des régulations irrégulières pour endommager les charges sans violer la plage admissible.

Nos simulations ont montré que cette attaque introduit des déviations de la consigne au niveau de chaque DG comme le montre la Figure2.

3 Un framework pour un contrôle distribué plus sécurisé

L'objectif de cette section est de proposer une méthode pour sécuriser un système de contrôle distribué d'un micro-réseau. Nous proposons un framework pour la détection des attaques ciblant ces micro-réseaux, en particulier celles d'injection de fausses de données, à l'image que l'attaque *MaR* présentée dans la section 2 et également des attaques de déni de service. La

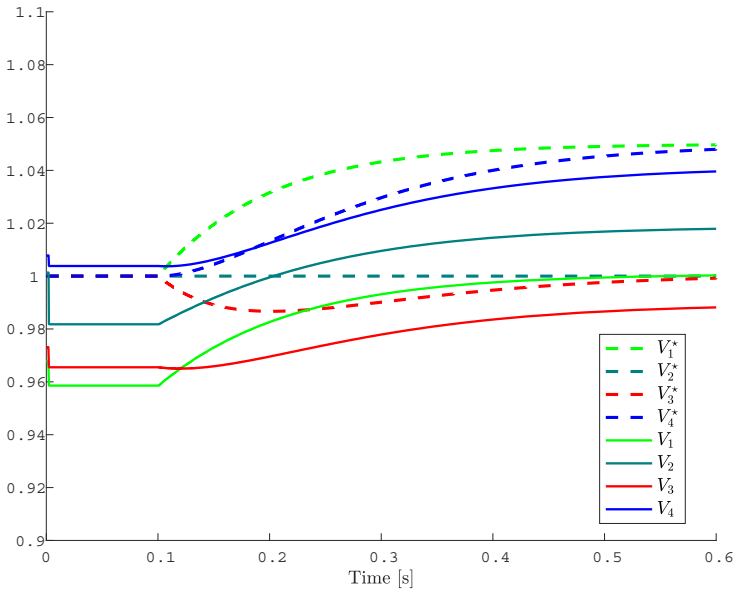

Figure 2: L'impact de l'attaque *MaR* sur la synchronisation d'une tension de référence entre 4 DGs.

Figure 3 présente l'architecture de notre framework pour à la fois détecter et répondre à ces attaques.

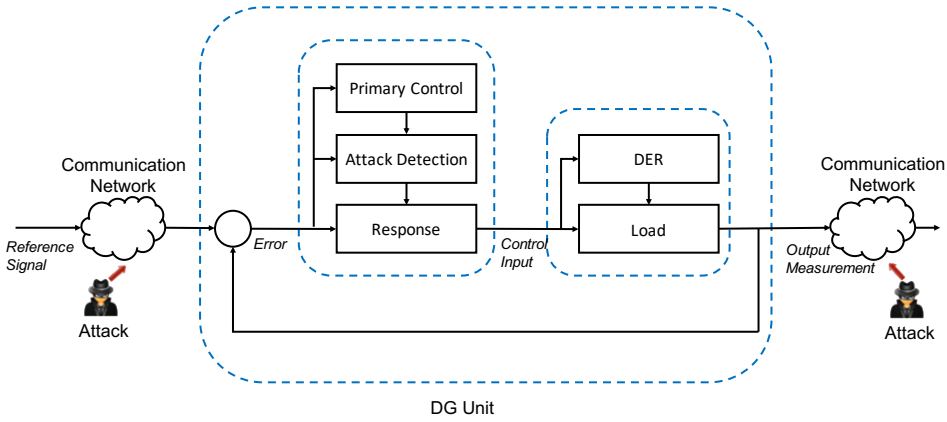

Figure 3: Un framework de contrôle sécurisé avec des modules de détection et de réaction aux attaques.

Ce framework comporte deux modules implantés dans chaque DG pour détecter d'une façon distribuée les attaques et en utilisant les données échangées sur le réseau de communication. Le module de détection s'appuie, notamment, sur des modèles de détection par apprentissage automatique, construits par des techniques supervisées. Nous avons considérés différents algorithmes d'apprentissage regroupés en 6 familles : les algorithmes Bayésiens, les algorithmes à base des instances (KNN, SVM), les algorithmes d'arbres de décision, les réseaux de neurones artificiels, les algorithmes d'ensemblistes (AdaBoost, les forêts aléatoires) et les algorithmes

d'apprentissage profond (1D-CNN). Pour évaluer ces algorithmes, nous avons mis en oeuvre une plate-forme expérimentale d'un micro-réseau pour collecter des jeux de données représentatifs de son fonctionnement normal et sous deux scénarios d'attaques : *MaR* et injection de délais. Nous avons avons considérés deux types de jeux de données : équilibrés et non équilibrés en termes de tailles des échantillons pour chaque classe afin d'analyser les performances de ces algorithmes dans les deux situations.

4 Méthodes de détection par apprentissage automatique

Cette section détaille les résultats de l'évaluation de performances des différents algorithmes d'apprentissage présentés dans la section précédente pour valider le module de détection de notre framework de contrôle sécurisé. Tout d'abord, nous avons montré que des algorithmes de détection basés sur des seuils sont inefficaces pour détecter les attaques introduites dans notre plate-forme expérimentale. La figure 4 montre la courbe ROC d'un classifieur basé sur la comparaison de seuils pour détecter l'attaque *MaR*. Nous observons que les performances de ce classifieur sont faibles pour détecter cette attaque.

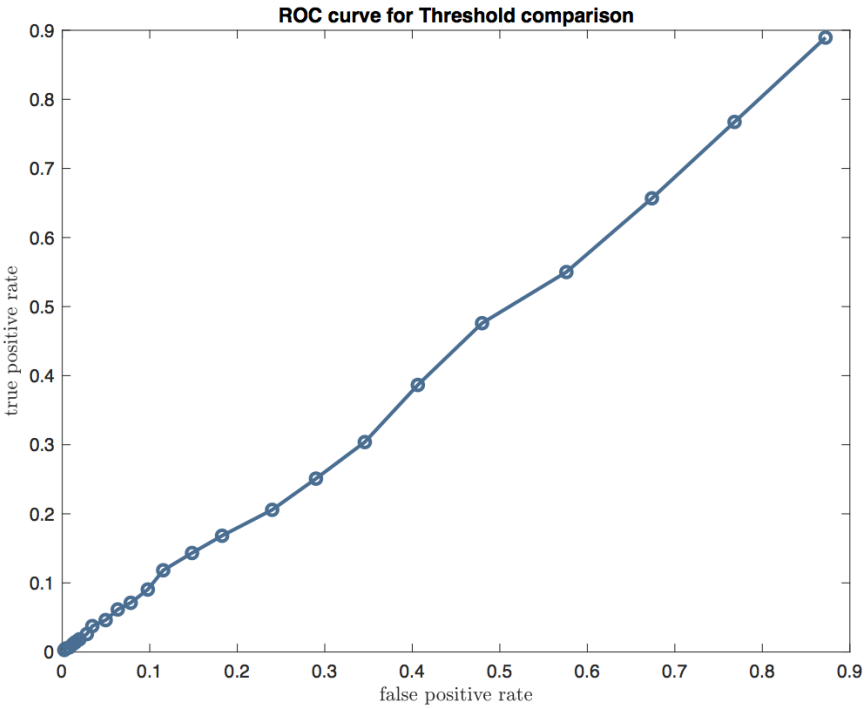

Figure 4: Courbe ROC d'un classifieur basé sur la comparaison de seuils pour détecter l'attaque *MaR*.

Nous nous sommes intéressés ensuite aux performances des algorithmes d'apprentissage que nous avons identifiés. Dans une première étape, nous avons estimé pour chaque algorithme ces hyper-paramètres optimaux. A titre d'exemple, nous avons identifié un nombre d'arbres égal à 30 avec une profondeur de 20 et un nombre maximal de caractéristiques (features) ègal à 3 pour l'algorithme de forêts aléatoires.

Nous avons évalué ces algorithmes en utilisant les deux ensemble de jeux de données collectés avec notre plate-forme expérimentale. Chaque ensemble contient 3 scénarios de classification :

classifier un comportement normal et une attaque *MaR*, classifier un comportement et une attaque d'injection de délais, classifier un comportement normal, une attaque *MaR* et une attaque d'injection de délais.

Tout d'abord, nous avons étudié, le premier ensemble de jeux de données qui est déséquilibré en terme de nombre d'échantillons par classe. Nos résultas sur cet ensemble de jeux de données, quelque soit le scénario de classification, montrent que les algorithmes à base d'arbres (arbres de décision, forêts aléatoires et AdaBoost) présentent des meilleurs performances en comparaison aux autres algorithmes. Par exemple, on obtient une valeur de F-measure d'environ 0.99 pour ces trois algorithmes pour classifier un comportement et une attaque d'injection de délais.

Deuxièmement, nous avons étudié le deuxième ensemble de jeux de données qui est équilibré en terme de nombre d'échantillons par classe. Nos résultas sur cet ensemble de jeux de données, quelque soit le scénario de classification, montrent que les algorithmes à base d'arbres (arbres de décision, forêts aléatoires et AdaBoost) présentent également des meilleurs performances en comparaison aux autres algorithmes. Par exemple, on obtient une valeur de F-measure d'environ 0.99 pour ces trois algorithmes pour classifier un comportement et une attaque d'injection de délais, en utilisant cet ensemble de jeux de données.

Nous avons également évalué le temps d'entrainement et de tests de ces différentes algorithmes. Nos résultats montrent que les 3 algorithmes basés sur les arbres présentent des durées acceptables pour ces réaliser ces opérations. A titre d'exemple, l'algorithme des arbres de décision nécessite environ 0.141 secondes construire un modèle et le tester sur un jeu de données de taille 12297 échantillons.

Nos différents résultats confirment que les algorithmes à base d'arbres (arbres de décision, forêts aléatoires et AdaBoost) sont adaptés à notre module de détection pour détecter d'une façon précise les différentes attaques étudiées.

5 Conclusion générale

5.1 Travail réalisé

L'intégration des système de contrôle distribués et coopératifs dans les micro-réseaux a introduit plusieurs problématiques de cyber sécurité, notamment, des attaques ciblant leurs réseaux de communication pour falsifier les données ou introduire des délais. Dans cette thèse, nous traitons ce problème en 3 étapes. Tout d'abord, nous avons étudié l'impact d'une attaque de synchronisation sur la régulation d'un micro-réseau. Cette attaque permet à une attaquant de remplacer la consigne par une mesure lors de leurs échanges entre deux entités de contrôle. Nous avons simulé cette attaques et nous avons démontré son effet sur la régulation d'une consigne de la tension du réseau électrique. Deuxièmement, nous avons proposé un framework pour offrir un contrôle distribué et sécurisé pour ces micro-réseaux. Notre framework s'appuie notamment sur la détection des attaques de falsification de données et l'injection des délais. Nous avons mis en oeuvre et développé une plate-forme expérimentale pour étudier ces attaques et collecter des jeux de données représentatifs de leurs modes opératoires. Troisièmement, nous avons évaluer différents algorithmes d'apprentissage pour classifier les paquets échangés sur le système de contrôle et détecter ces attaques. Nous avons identifié que les algorithmes basés sur les arbres offrent les meilleures performances en terme de précision et temps de détection.

5.2 Perspectives de recherche

Cette thèse ouvre de nombreuses perspectives visant à améliorer notre approche. Tout d'abord, il faudrait étudier la détection d'autres attaques et d'autres anomalies dues à des dysfonctionnement et développer des modèles capables de les distinguer. À un autre niveau, notre solution est seulement appliquée aux micro-réseaux et leurs système de contrôle, il est également intéressant d'adapter notre approche aux autres environnements à l'image de smart-home, smart-city, etc. Enfin, la méthode de réponse aux attaques, mérite d'être améliorée en reconfigurant par exemple le réseau de communication d'une façon dynamique, ce qui permet de réduire l'impact des attaques et réduire leurs surfaces.

Bibliography

[1] Mariam MN Aboelwafa, Karim G Seddik, Mohamed H Eldefrawy, Yasser Gadallah, and Mikael Gidlund. A machine learning-based technique for false data injection attacks detection in industrial iot. *IEEE Internet of Things Journal*, 2020.

[2] Mohammad Ashrafuzzaman, Yacine Chakhchoukh, Ananth A Jillepalli, Predrag T Tosic, Daniel Conte de Leon, Frederick T Sheldon, and Brian K Johnson. Detecting stealthy false data injection attacks in power grids using deep learning. In *2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*, pages 219–225. IEEE, 2018.

[3] Abdelrahman Ayad, Hany EZ Farag, Amr Youssef, and Ehab F El-Saadany. Detection of false data injection attacks in smart grids using recurrent neural networks. In *2018 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, pages 1–5. IEEE, 2018.

[4] Omar Ali Beg, Taylor T Johnson, and Ali Davoudi. Detection of false-data injection attacks in cyber-physical dc microgrids. *IEEE Transactions on industrial informatics*, 13(5):2693–2703, 2017.

[5] Asa Ben-Hur, David Horn, Hava T Siegelmann, and Vladimir Vapnik. Support vector clustering. *Journal of machine learning research*, 2(Dec):125–137, 2001.

[6] Suzhi Bi and Ying Jun Zhang. Graphical methods for defense against false-data injection attacks on power system state estimation. *IEEE Transactions on Smart Grid*, 5(3):1216–1227, 2014.

[7] A. Bidram, F. Lewis, and A. Davoudi. Distributed control systems for small-scale power networks: Using multiagent cooperative control theory. *IEEE Control Systems*, 34(6):56–77, 2014.

[8] Ali Bidram and Ali Davoudi. Hierarchical structure of microgrids control system. *IEEE Transactions on Smart Grid*, 3(4):1963–1976, 2012.

[9] Ali Bidram, Ali Davoudi, Frank L Lewis, and Shuzhi Sam Ge. Distributed adaptive voltage control of inverter-based microgrids. *IEEE Transactions on Energy Conversion*, 29(4):862–872, 2014.

[10] Bushra Canaan, Bruno Colicchio, and Djaffar Ould Abdeslam. Microgrid cyber-security: Review and challenges toward resilience. *Applied Sciences*, 10(16):5649, 2020.

[11] Olivier Chapelle, Vikas Sindhwani, and Sathiya S Keerthi. Optimization techniques for semi-supervised support vector machines. *Journal of Machine Learning Research*, 9(Feb):203–233, 2008.

[12] Eduardo Cotilla-Sanchez, Paul DH Hines, Clayton Barrows, and Seth Blumsack. Comparing the topological and electrical structure of the north american electric power infrastructure. *IEEE Systems Journal*, 6(4):616–626, 2012.

[13] Aris Dimeas and Nikos Hatziargyriou. A multi-agent system for microgrids. In *Hellenic Conference on Artificial Intelligence*, pages 447–455. Springer, 2004.

[14] Derui Ding, Qing-Long Han, Yang Xiang, Xiaohua Ge, and Xian-Ming Zhang. security control and attack detection for industrial cyber-physical systems. *Neurocomputing*, 275:1674–1683, 2018.

[15] Peter Dondi, Deia Bayoumi, Christoph Haederli, Danny Julian, and Marco Suter. Network integration of distributed power generation. *Journal of power sources*, 106(1-2):1–9, 2002.

[16] Dajun Du, Xue Li, Wenting Li, Rui Chen, Minrui Fei, and Lei Wu. Admm-based distributed state estimation of smart grid under data deception and denial of service attacks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49(8):1698–1711, 2019.

[17] Mohammad Esmalifalak, Lanchao Liu, Nam Nguyen, Rong Zheng, and Zhu Han. Detecting stealthy false data injection using machine learning in smart grid. *IEEE Systems Journal*, 11(3):1644–1652, 2014.

[18] Enrique Espina, Jacqueline Llanos, Claudio Burgos-Mellado, Roberto Cárdenas-Dobson, Manuel Martínez-Gómez, and Doris Sáez. Distributed control strategies for microgrids: An overview. *IEEE Access*, 2020.

[19] Ivan Gesteira Costa Filho. Mixture models for the analysis of gene expression. 2008.

[20] S Armina Foroutan and Farzad R Salmasi. Detection of false data injection attacks against state estimation in smart grids based on a mixture gaussian distribution learning method. *IET Cyber-Physical Systems: Theory & Applications*, 2(4):161–171, 2017.

[21] Mehdi Ganjkhani, Seyedeh Narjes Fallah, Sobhan Badakhshan, Shahaboddin Shamshirband, and Kwok-wing Chau. A novel detection algorithm to identify false data injection attacks on power system state estimation. *Energies*, 12(11):2209, 2019.

[22] Demis Hassabis, Dharshan Kumaran, Christopher Summerfield, and Matthew Botvinick. Neuroscience-inspired artificial intelligence. *Neuron*, 95(2):245–258, 2017.

[23] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[24] Youbiao He, Gihan J Mendis, and Jin Wei. Real-time detection of false data injection attacks in smart grid: A deep learning-based intelligent mechanism. *IEEE Transactions on Smart Grid*, 8(5):2505–2516, 2017.

[25] Julien M Hendrickx, Karl Henrik Johansson, Raphael M Jungers, Henrik Sandberg, and Kin Cheong Sou. Efficient computations of a security index for false data attacks in power networks. *IEEE Transactions on Automatic Control*, 59(12):3194–3208, 2014.

[26] Geoffrey E Hinton. Deep belief networks. *Scholarpedia*, 4(5):5947, 2009.

[27] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[28] Md Alamgir Hossain, Hemanshu Roy Pota, Md Jahangir Hossain, and Frede Blaabjerg. Evolution of microgrids with converter-interfaced generations: Challenges and opportunities. *International Journal of Electrical Power & Energy Systems*, 109:160–186, 2019.

[29] Ahmed Iqbal, Shabib Aftab, Israr Ullah, Muhammad Anwaar Saeed, and Arif Husen. A classification framework to detect dos attacks. *International Journal of Computer Network & Information Security*, 11(9), 2019.

[30] Y. Isozaki, S. Yoshizawa, Y. Fujimoto, H. Ishii, I. Ono, T. Onoda, and Y. Hayashi. On detection of cyber attacks against voltage control in distribution power grids. In *IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 842–847, 2014.

[31] JQ James, Yunhe Hou, and Victor OK Li. Online false data injection attack detection with wavelet transform and deep neural networks. *IEEE Transactions on Industrial Informatics*, 14(7):3271–3280, 2018.

[32] Yuning Jiang, Yacine Atif, and Jianguo Ding. Cyber-physical systems security based on a cross-linked and correlated vulnerability database. In *International Conference on Critical Information Infrastructures Security*, pages 71–82. Springer, 2019.

[33] Thorsten Joachims. Making large-scale svm learning practical. Technical report, Technical Report, 1998.

[34] Charalambos Konstantinou and Michail Maniatakos. A data-based detection method against false data injection attacks. *IEEE Design & Test*, 2019.

[35] Oliver Kosut, Liyan Jia, Robert J Thomas, and Lang Tong. Malicious data attacks on the smart grid. *IEEE Transactions on Smart Grid*, 2(4):645–658, 2011.

[36] Moshe Kravchik and Asaf Shabtai. Detecting cyber attacks in industrial control systems using convolutional neural networks. In *Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and PrivaCy*, pages 72–83, 2018.

[37] Zhiyi Li, Mohammad Shahidehpour, and Farrokh Aminifar. Cybersecurity in distributed power systems. *Proceedings of the IEEE*, 105(7):1367–1388, 2017.

[38] Lanchao Liu, Mohammad Esmalifalak, Qifeng Ding, Valentine A Emesih, and Zhu Han. Detecting false data injection attacks on power grid by sparse optimization. *IEEE Transactions on Smart Grid*, 5(2):612–621, 2014.

[39] Yao Liu, Peng Ning, and Michael K Reiter. False data injection attacks against state estimation in electric power grids. *ACM Transactions on Information and System Security (TISSEC)*, 14(1):13, 2011.

[40] M. Ma, A. Teixeira, J. van den Berg, and P. Palensky. Voltage control in distributed generation under measurement falsification attacks. *IFAC-PapersOnLine*, 50(1):8379–8384, 2017.

[41] Mingxiao Ma and Abdelkader Lahmadi. On the impact of synchronization attacks on distributed and cooperative control in microgrid systems. In *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pages 1–6. IEEE, 2018.

[42] Mingxiao Ma, Abdelkader Lahmadi, and Isabelle Chrisment. Detecting a stealthy attack in distributed control for microgrids using machine learning algorithms. In *2020 IEEE International Conference on Industrial Cyber Physical Systems (ICPS)*. IEEE, 2020.

[43] Kebina Manandhar, Xiaojun Cao, Fei Hu, and Yao Liu. Detection of faults and attacks including false data injection attack in smart grid using kalman filter. *IEEE transactions on control of network systems*, 1(4):370–379, 2014.

[44] Nisharani Meti, DG Narayan, and VP Baligar. Detection of distributed denial of service attacks using machine learning algorithms in software defined networks. In *2017 international conference on advances in computing, communications and informatics (ICACCI)*, pages 1366–1371. IEEE, 2017.

[45] Sakshi Mishra, Kate Anderson, Brian Miller, Kyle Boyer, and Adam Warren. Microgrid resilience: A holistic approach for assessing threats, identifying vulnerabilities, and designing corresponding mitigation strategies. *Applied Energy*, 264:114726, 2020.

[46] S. Moayedi and A. Davoudi. Distributed tertiary control of dc microgrid clusters. *IEEE Transactions on Power Electronics*, 31(2):1717–1733, 2016.

[47] Mostafa Mohammadpourfard, Ashkan Sami, and Ali Reza Seifi. A statistical unsupervised method against false data injection attacks: A visualization-based approach. *Expert Systems with Applications*, 84:242–261, 2017.

[48] Mostafa Mohammadpourfard, Ashkan Sami, and Yang Weng. Identification of false data injection attacks with considering the impact of wind generation and topology reconfigurations. *IEEE Transactions on Sustainable Energy*, 9(3):1349–1364, 2017.

[49] Reza Olfati-Saber and Richard M Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on automatic control*, 49(9):1520–1533, 2004.

[50] Mete Ozay, Inaki Esnaola, Fatos Tunay Yarman Vural, Sanjeev R Kulkarni, and H Vincent Poor. Machine learning methods for attack detection in the smart grid. *IEEE transactions on neural networks and learning systems*, 27(8):1773–1786, 2015.

[51] Binod P Poudel, Aquib Mustafa, Ali Bidram, and Hamidreza Modares. Detection and mitigation of cyber-threats in the dc microgrid distributed control system. *International Journal of Electrical Power & Energy Systems*, 120:105968, 2020.

[52] Milan Prodanovic and Timothy C Green. High-quality power generation through distributed control of a power park microgrid. *IEEE Transactions on Industrial Electronics*, 53(5):1471–1482, 2006.

[53] A. Rantzer. Scalable control of positive systems. *European Journal of Control*, 24:72–80, 2015.

[54] Wei Ren and Randal W Beard. Consensus seeking in multiagent systems under dynamically changing interaction topologies. *IEEE Transactions on automatic control*, 50(5):655–661, 2005.

[55] Raúl Rojas et al. Adaboost and the super bowl of classifiers a tutorial introduction to adaptive boosting. *Freie University, Berlin, Tech. Rep*, 2009.

[56] Ruhi Sarikaya, Geoffrey E Hinton, and Anoop Deoras. Application of deep belief networks for natural language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(4):778–784, 2014.

[57] Ali Sayghe, Yaodan Hu, Ioannis Zografopoulos, XiaoRui Liu, Raj Gautam Dutta, Yier Jin, and Charalambos Konstantinou. Survey of machine learning methods for detecting false data injection attacks in power systems. *IET Smart Grid*, 2020.

[58] J. Schiffer, R. Ortega, A. Astolfi, J. Raisch, and T. Sezi. Conditions for stability of droop-controlled inverter-based microgrids. *Automatica*, 50(10):2457–2469, 2014.

[59] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[60] J. Simpson-Porco, F. Dörfler, and F. Bullo. Synchronization and power sharing for droop-controlled inverters in islanded microgrids. *Automatica*, 49(9):2603–2611, 2013.

[61] Devender Singh, Rakesh Kumar Misra, and Deependra Singh. Effect of load models in distributed generation planning. *IEEE Transactions on Power Systems*, 22(4):2204–2212, 2007.

[62] Adel Tabakhpour and Morad MA Abdelaziz. Neural network model for false data detection in power system state estimation. In *2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE)*, pages 1–5. IEEE, 2019.

[63] A. Teixeira, K. Paridari, H. Sandberg, and K. Johansson. Voltage control for interconnected microgrids under adversarial actions. In *IEEE Emerging Technologies & Factory Automation (ETFA)*, pages 1–8, 2015.

[64] André Teixeira. *Toward cyber-secure and resilient networked control systems*. PhD thesis, KTH Royal Institute of Technology, 2014.

[65] Jen-Hao Teng, Yi-Hwa Liu, Chia-Yen Chen, and Chi-Fa Chen. Value-based distributed generator placements for service quality improvements. *International Journal of Electrical Power & Energy Systems*, 29(3):268–274, 2007.

[66] Vincent Van Asch. Macro-and micro-averaged evaluation measures [[basic draft]]. *Belgium: CLiPS*, 49, 2013.

[67] W. Wang and Z. Lu. Cyber security in the smart grid: Survey and challenges. *Computer Networks*, 57(5):1344–1371, 2013.

[68] Lei Wei, Donghuai Gao, and Cheng Luo. False data injection attacks detection with deep belief networks in smart grid. In *2018 Chinese Automation Congress (CAC)*, pages 2621–2625. IEEE, 2018.

[69] Huanhai Xin, Zhihua Qu, John Seuss, and Ali Maknouninejad. A self-organizing strategy for power flow control of photovoltaic generators in a distribution network. *IEEE Transactions on Power Systems*, 26(3):1462–1473, 2010.

[70] Monika Yadav, Nitai Pal, and Devender Kumar Saini. Microgrid control, storage, and communication strategies to enhance resiliency for survival of critical load. *IEEE Access*, 8:169047–169069, 2020.

[71] Jun Yan, Bo Tang, and Haibo He. Detection of false data attacks in smart grid with supervised learning. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 1395–1402. IEEE, 2016.

[72] Can Yang, Yong Wang, Yuhao Zhou, Jiongming Ruan, Wei Liu, et al. False data injection attacks detection in power system using machine learning method. *Journal of Computer and Communications*, 6(11):276, 2018.

[73] Qiang Yang, Javier A Barria, and Tim C Green. Communication infrastructures for distributed control of power distribution networks. *IEEE Transactions on Industrial Informatics*, 7(2):316–327, 2011.

[74] Ahmad Zahedi. A review of drivers, benefits, and challenges in integrating renewable energy sources into electricity grid. *Renewable and Sustainable Energy Reviews*, 15(9):4775–4779, 2011.