Attack detection in PV-integrated power distribution systems using Machine Learning and Deep Learning methods

Masoud Ahmadzadeh

A Thesis

in

The Department

of

Concordia Institute for Information Systems Engineering (CIISE)

Presented in Partial Fulfillment of the Requirements

for the Degree of

Master of Applied Science (Information Systems Security) at

Concordia University

Montréal, Québec, Canada

April 2023

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Masoud Ahmadzadeh**

Entitled: **Attack detection in PV-integrated power distribution systems using Machine Learning and Deep Learning methods**

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Information Systems Security)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

______________________ Chair
Dr. Suryadipta Majumdar

______________________ External Examiner
Dr. Farnoosh Naderkhani

______________________ Examiner
Dr. Suryadipta Majumdar

______________________ Supervisor
Dr. Mohsen Ghafouri

Approved by ______________________
Abdessamad Ben Hamza, Chair
Department of Concordia Institute for Information Systems Engineering (CIISE)

__________ 2023

____
Mourad Debbabi, Dean
Faculty of Engineering and Computer Science

Abstract

Attack detection in PV-integrated power distribution systems using Machine Learning and Deep Learning methods

Masoud Ahmadzadeh

This master's thesis presents a comprehensive study on false data injection (FDI) attacks in photovoltaic (PV)-integrated power distribution systems (PDSs) and proposes two data-driven detection frameworks to identify such attacks against voltage regulation in both steady-state and transient modes. With the deployment of information and communication technologies (ICTs) for voltage regulation, PDSs are exposed to cyber threats, including FDI attacks.

The first proposed framework employs a supervised machine learning (ML) approach based on a support vector machine (SVM) to detect FDI attacks during the transmission of data from the centralized controller to PVs in a transient state. The framework collects a dataset of different operating points, such as loading conditions, to train the SVM. The performance of the trained framework for attack detection is compared with other supervised and unsupervised ML-based techniques. The results demonstrate the superior performance of the proposed framework in detecting FDI attacks against modified IEEE 33-bus PDS.

The second proposed framework is a convolutional neural network (CNN) approach to detect FDI attacks against voltage regulation in steady-state. The framework creates a comprehensive and realistic dataset that covers all normal conditions and unpredictable changes of a PDS during two years, including features such as voltage measurements, season, weekdays/weekends, load conditions, and PV generation power. The CNN is trained to distinguish normal grid changes from FDI attacks. The performance of the trained framework has been compared with other supervised ML-based and deep-learning techniques for FDI attacks against a modified IEEE 33-bus and 141-bus PDS to show the scalability of the proposed method, and the results demonstrate the superior performance of the proposed framework in detecting FDI attacks.

Overall, this study aims to enhance the cyber-security of PV-integrated PDSs by proposing two effective and reliable detection frameworks against FDI attacks in both steady-state and transient modes. By utilizing ML and deep learning techniques, the proposed frameworks are capable of accurately detecting FDI attacks, thereby improving the overall resilience of PDSs against cyber threats.

Acknowledgments

I would like to take this opportunity to express my sincere gratitude and appreciation to several individuals who have supported and encouraged me throughout my research and study journey. Firstly, I would like to extend my heartfelt thanks to my supervisor, Dr. Mohsen Ghafouri, for his invaluable guidance, expert knowledge, and unwavering support. His encouragement, feedback, and insightful advice have been instrumental in shaping the direction of my research, and his mentorship has been truly inspiring.

I am grateful to my family, especially my father, for their unwavering support, encouragement, and inspiration during my academic journey. His trust in my abilities has been a constant source of strength and motivation, and I cannot express my gratitude enough for his unconditional love and support. I would also like to express my love and appreciation to my mother for bringing color and joy into my life. Furthermore, I would like to acknowledge my dear nieces, Rozha and Roysa, and my beloved nephews, Shahan and Ilia, for the happiness they have brought into my life.

I would also like to acknowledge my dear friends, Arash, Fatemeh, Sahel, Amir, Mohsen, Younes, Mamraz, Ahmad, Ashkan, Negar, and Assef, for their unrelenting support, encouragement, and empathy throughout my research. Their compassion, motivation, and willingness to lend an ear have been invaluable to me, and I am deeply grateful for their presence in my life.

Contents

List of Figures

List of Tables

List of Acronyms

AC Alternating Current

AGC Automatic Generation Control

AMI Advanced Metering Infrastructure

API Application Programming Interface

BDD Bad Data Detection

CNN Convolutional Neural Network

DC Direct Current

DL Deep Learning

DT Decision Tree

ENN Extended Nearest Neighbor

FDI False Data Injection

FN False Negative

FP False Positive

ICT Information and Communication Technologies

kNN k-Nearest Neighbor

LR Logistic Regression

MITM Man-in-the-middle

ML Machine Learning

MLP Multilayer Perceptron

MLSTM Multi-layer Long Short-Term Memory Network

MPP Maximum Power Point

MPPT Maximum Power Point Tracking

OLTC On-load tap changer

PCC Point of Common Coupling

PDS Power Distribution System

PI Proportional-integral

PLL Phase-Locked Loop

PSO Particle Swarm Optimization

PV Photovoltaic System

RBF Radial Basis Function

ReLU Rectified Linear Unit

RF Random Forest

RTUs Remote terminal units

SNR Signal to Noise Ratio

SVM Support Vector Machine

SVR Static Voltage Regulator

TN True Negative

TP True Positive

WANs Wide-area networks

Chapter 1

Introduction

The power grid is a highly complex and essential system that facilitates the delivery of electricity to end users [54]. One of its most significant roles is to maintain a stable and reliable power supply by regulating the balance between power generation and demand. The power grid is typically divided into three main sections, namely, generation, transmission, and distribution [48]. Generation refers to the process of producing electricity at power plants through various methods, such as burning fossil fuels [19], nuclear reactions [65], and renewable sources like solar, wind, hydro, and geothermal energy [33]. The electricity generated by power plants is in the form of high-voltage alternating current (AC) which is then transmitted to distant locations via power lines [66]. Transmission involves the movement of electricity over long distances from the power plants to substations and local distribution centers. To minimize losses during transmission, high-voltage power lines are used, typically ranging from 69,000 to 765,000 volts, which are stepped down at substations to lower voltages before distribution [63]. Finally, distribution is the last step in delivering electricity to customers. It refers to the process of delivering electricity to homes, businesses, and other end-users via local distribution networks [73]. This is done through transformers which step down the voltage to levels appropriate for use in homes and businesses. The distribution network also includes devices such as circuit breakers and fuses to protect the grid and consumers from faults, such as power outages and electrical fires [38].

The incorporation of renewable energy sources, particularly solar and wind, into the grid has been a crucial strategy in enhancing the efficiency of power distribution systems (PDSs). Photovoltaic (PV) systems, in particular, have been widely recognized for their efficacy, low maintenance requirements, and practicality as a source of renewable energy [60]. PV systems have the potential to bring a number of benefits to voltage regulation systems in PDS. One benefit of PV systems is that they can help reduce line losses, which occur when electricity is transmitted over long distances. By generating power closer to where it is being consumed, PV systems can reduce the need for long-distance transmission, which can be expensive and inefficient [24]. These systems can also be used to regulate the grid parameters, such as voltage. The fluctuating nature of power consumption in distribution systems often causes voltage disturbances, which could result in equipment failure and system blackouts [12]; thus, PV systems can help regulate voltage levels within acceptable ranges by injecting or absorbing reactive power as needed to regulate the voltage [35]. Also, a key benefit of PV systems for voltage regulation is that they can respond quickly to changes in power output, helping the operation of the grid in real time. In contrast, traditional voltage regulation systems, such as tap-changing transformers, tend to be slower and less responsive to changes in power output [79].

Information and Communication Technologies (ICTs) have been suggested as a solution to improve the efficiency and effectiveness of voltage regulation and data transfer between the central controller and PV systems [6]. The use of ICTs enables the collection of real-time data from the PV system and helps to make informed decisions for the optimal operation of the PV system. By utilizing ICTs, the central controller can have access to precise data, which in turn, enables accurate monitoring and management of the PV system. Additionally, ICTs can facilitate communication and control between the central controller and PV systems, allowing for efficient and effective management of the entire PDS [62]. However, these same technologies, that have brought these benefits, also have introduced new vulnerabilities to the power grid, making it susceptible to various cyber-attacks such as False Data Injection (FDI) attacks [5]. FDI attacks are a type of cyber attack that has become a significant threat to the security and reliability of PDSs. Adversaries carry out these attacks by manipulating the measurements of a power system to create unfavorable conditions, such as unacceptable voltage levels [15]. PV systems, which are integrated with PDS, are a common target of FDI attacks, as adversaries seek to create unacceptable voltage levels at various buses and trigger the unnecessary operation of protection systems [29]. These attacks can be detrimental to the smooth functioning of PDS, leading to costly consequences such as damage to equipment, power outages, regulatory penalties, etc. [75].

FDI attacks are relatively simple to execute but can cause severe disruptions in the real-time control systems used in power grids, making them challenging to detect and mitigate in a timely manner [41, 70, 23]. As such, it is crucial to implement robust and adaptive security measures when integrating control systems in the industrial IoT to prevent FDI attacks in PV-integrated power grids [22]. This requires a comprehensive security framework that can monitor and detect any anomalous behavior, quickly identify and respond to security threats, and mitigate their impact on the PDS. Such measures can include the use of encryption [3], multi-factor authentication [69], and intrusion detection systems [44] to protect the power grid's critical infrastructure and data against malicious cyber-attacks. Furthermore, to ensure the sustainability and resilience of PV-integrated power grids, it is essential to continually monitor, evaluate, and enhance their cybersecurity posture by implementing up-to-date security protocols, threat intelligence, and response strategies. Thus, it is necessary to develop detection mechanisms that can efficiently and effectively identify these attacks.

In this study, we focus on identifying FDI attacks in a PV-integrated PDS voltage control system. The voltage readings and loading data are sent to a centralized controller to manage the voltage at a desired location, such as the point of common coupling (PCC). The controller then calculates the control signals that are sent back to the PVs, where those signals are added to the local control loops of the PV converters. When an attacker manipulates the measurements in this scheme, an FDI attack is launched. This attack causes an inappropriate voltage profile and can trip the relays in the protection systems, leading to potential system failure.

To tackle this problem, we propose two different frameworks for detecting FDI attacks in transient and steady-state conditions.

- In the transient condition, we develop a machine learning (ML) framework based on support vector machines (SVM). A dataset of different operational points, such as loading conditions and voltage measurements, is obtained to train this supervised framework. The modified IEEE 33-bus PDS is used to show the attack implications and evaluate the effectiveness of the tuned

SVM detection framework. Moreover, the comparison with several existing supervised and unsupervised ML-based approaches demonstrated the better performance of the developed framework.

- In the steady-state condition, we propose a deep learning-based detection strategy that utilizes a convolutional neural network (CNN). We develop the framework using a dataset that includes various features, such as voltage measurements, loading conditions, time-based factors (e.g., season and weekdays/weekends), and PV power generation. The framework is evaluated using a modified IEEE 33-bus PDS and tested on the IEEE 141-bus PDS to demonstrate scalability. The proposed method is compared to existing supervised and deep learning-based approaches and is shown to perform superiorly. Furthermore, in order to improve the credibility and practicality of the proposed approach in actual real-world situations, the research exhibits the ability of the method to withstand and function effectively in the presence of noise.

Therefore, our contributions in this study are two-fold: first, we develop an SVM-based detection framework for transient conditions, and second, we propose a novel deep learning-based framework utilizing CNN for FDI attack detection in steady-state conditions.

The thesis is structured into eight chapters, each addressing a specific aspect of the research. The second chapter reviews the related works in the field of FDI attacks and detection techniques. In the third chapter, the background of PDSs and PVs is explained, including their functioning and control methods. The fourth chapter presents the system model used in the study, including the PDS architecture, voltage regulation system, and power flow analysis system. The fifth chapter describes the threat model and different types of FDI attacks that can affect the PDS. In the sixth chapter, two different detection models, namely SVM and CNN, are proposed to detect FDI attacks in the transient and steady state, respectively. The seventh chapter presents the simulations and results obtained from the proposed detection models on the modified IEEE 33-bus PDS and 141-bus PDS. Finally, the eighth chapter concludes the thesis by summarizing the findings, discussing the limitations and future directions, and suggesting practical recommendations for power system operators to enhance the security of PV-integrated PDSs against FDI attacks.

Chapter 2

Literature Review

Generally speaking, cyber-attack detection algorithms can be broadly categorized into two categories: model-based and data-driven techniques [27, 75].

In model-based techniques, a mathematical model is developed based on the knowledge of the system being monitored. The model is used to simulate the behavior of the system under normal operating conditions, and any deviation from the expected behavior is considered an anomaly [49]. Data-driven techniques, on the other hand, rely on ML algorithms to detect anomalies in the data generated by the system being monitored. In data-driven techniques, the algorithms are trained on historical data to identify patterns and regularities in the data. Once the model is trained, it is used to detect anomalies in real-time data streams. Data-driven techniques are typically more flexible and can adapt to changes in the system being monitored [50].

2.1 Model-based Techniques

Model-based methods employ mathematical models to detect cyber-attacks, such as the state estimation approach presented in the literature [45, 80, 43, 30] for the detection of FDI attacks in distribution systems.

Reference [42] highlighted the vulnerability of real-time topology to cyber-attacks, and emphasized the need for effective protection and detection strategies. Also, [7] presents an anomaly detection approach to identify abnormal power flow patterns in the network caused by accidental or intentional changes to the database. The method aims to identify any deviations from the normal power flow behavior and alerts system operators to potential issues. Moreover, [57] introduces an algorithm that can identify and describe cyberattacks on network parameter data in the context of optimal power flow problems. In addition, [67] presents a framework for ensuring that power systems are resilient to cyberattacks by proposing a model-based anomaly detection and attack mitigation algorithm for Automatic Generation Control (AGC). The paper highlights the impact of data integrity attacks on power system frequency and electricity market operation, and evaluates the detection capability of the proposed algorithm through simulation studies. The manipulation of data collected by smart meters and the corresponding methods for identifying such manipulation have been extensively examined in previous literature, as evidenced by studies such as [10, 32]. In previous studies, load redistribution attacks and the algorithms used to detect them were discussed and outlined in detail, as presented in references [76, 36, 13]. The article [10] suggests a two-tier attack detection system that operates in real-time and can recognize organized data falsification in decentralized microgrids, even under complex threat models. In [32], the current status of Advanced Metering Infrastructure (AMI) energy theft detection techniques are summarized and divided into three categories: classification-based, state estimation-based, and game theory-based methods, and they are extensively compared and discussed. The study [53] investigates that how adversaries can deceive the electrical grid by manipulating AMI systems. Moreover, reference [55] presents a detection mechanism based on a model that uses Extended Kalman filter interval state estimation (ISE) to eliminate inaccurate measurements resulting from meter malfunctions or external attacks in power systems. In [58], a decentralized and fast method for detecting cyber attacks in power grids is proposed. This method employs a maximum likelihood estimation approach, taking advantage of the near chordal sparsity of power grids. Meanwhile, [46] aims to propose a model-based approach for detecting FDI attacks based on median filtering, which uses information from neighboring nodes.

Furthermore, [77] proposes a short-term state forecasting-based method, considering nodal state temporal correlations, for detecting FDIAs in smart grids. [8] provides a comparative analysis of previous research on FDI attacks in smart power grids. The study proposes novel attack and countermeasure classifications based on factors such as the targeted subsystem, and the physical and economic impact of the attack. [68] evaluates the feasibility of a malicious attack on the measurements utilized by the integrated volt-var control (VVC) mechanism in smart PDS. In the study by Ibrahim et al. [28], a model-based approach is proposed for detecting cyberattacks on sensors in grid-tied PV systems. The authors highlight various cyber vulnerabilities in the control scheme of these systems and propose a defense mechanism that involves adding a watermarking signal to the control inputs to improve security. Additionally, Isozaki et al. [30] investigates the effects of cyber attacks on voltage regulation within PDS that incorporate a significant number of PV systems. The paper proposes a detection algorithm that can identify false sensor measurements, but it is limited in scenarios where a large number of attacks are executed, which can cause voltage violations at specific nodes. Despite being proficient, model-based techniques have certain drawbacks, such as the need for accurate system parameter knowledge, which may not be accessible to distribution grid operators, and limited operational scope, resulting in suboptimal performance when the system is operating at different points [59, 56]. Additionally, FDI attacks can deceive model-based bad data detection systems by injecting false information into measurements without detection, which affects the state estimation process. To address this issue, the use of data-driven machine learning techniques for identifying malicious sensor data manipulation has gained popularity due to their speed and accuracy. These techniques have been proven to be effective in detecting such attacks, even when a significant number of sensors are compromised [64].

2.2 Data-driven Techniques

Recent progress in data processing technology has sparked increased attention towards data-driven methods for detecting cyber attacks in smart grid systems [39]. To address the issue of cyber-attacks, several studies have investigated the use of machine learning (ML) methods in the detection of FDI attacks in smart grids[17, 71, 25, 9, 64]. For instance, [61, 25, 9, 64] have explored different ML algorithms, including real-time detection, supervised, semi-supervised, ensemble-based, and online learning algorithms, for FDI attack detection in smart grids. In addition, in reference [74], the authors evaluated the effectiveness of three machine learning (ML)-based models—SVM, K-Nearest Neighbor (kNN), and Extended Nearest Neighbor (ENN)—in detecting cyber attacks in power systems. Meanwhile, reference [31] proposed a time-series algorithm based on a neural network, specifically a discrete-time nonlinear autoregressive neural network with exogenous inputs (NARX), for detecting and mitigating FDI attacks on advanced metering infrastructure. This algorithm is designed to identify FDI attacks and mitigate them accordingly. Additionally, reference [40] suggested a deep learning-based framework, called Multi-layer Long Short-Term Memory Network (MLSTM), for detecting cyber attacks in active PDS using voltage and current measurements. Also, authors in [39] presented a high-dimensional data-driven approach called HCADI to identify and detect cyber attacks in power systems. Additionally, according to [11], a two-stage method that combines ML techniques, namely Random Forest (RF) and Logistic Regression (LR), is proposed to identify and locate cyber attacks on voltage control systems in distributed generator systems. The first stage uses RF for predicting the current-voltage levels using past voltage measurements and weather data, while the second stage uses LR for comparing the predicted voltage levels to actual measurements to identify and locate the attack in real-time. However, due to the unpredictability of weather conditions and other uncertainties, the accuracy of the voltage level prediction is compromised, which can lead to inaccuracies when comparing predicted and actual measurements.

2.3 Research Gap

Data-driven techniques have emerged as a promising approach to address the challenge of detecting cyber attacks in smart grid systems. One of the advantages of data-driven techniques is their ability to effectively identify the presence of malicious data injection by analyzing the behavior and patterns of the grid data. This is because these techniques rely on large amounts of historical data to detect changes in the normal patterns of the grid data. By using machine learning algorithms, these techniques can also adapt to changes in the system and improve their accuracy over time. Compared to model-based methods that require accurate system parameter knowledge and have a restricted operational scope, data-driven techniques are not limited by these constraints. Model-based methods rely on precise knowledge of the system parameters, which can be difficult to obtain in practice. Additionally, model-based methods often have a limited operational scope, meaning that they may not be able to detect cyber attacks that occur outside of their scope. In contrast, data-driven techniques can use a wide range of data sources and do not require precise knowledge of the system parameters. For example, these techniques can use data from smart meters, weather sensors, and other sources to detect changes in the grid data. By combining data from multiple sources, these techniques can provide a more complete view of the system and improve their ability to detect cyber attacks.

The data-driven techniques mentioned earlier have not been utilized to detect cyber attacks aimed at the voltage regulation system of PV-integrated PDS in the presence of uncertainties such as load variations and weather conditions in both steady-state and transient conditions. The nature of these uncertainties and the complexity of PV-integrated PDSs makes it challenging to utilize ML approaches for detecting FDI attacks, and this has not been extensively researched in the literature. In other words, the use of data-driven techniques in detecting cyber attacks on voltage regulation systems in PV-integrated PDS has not been widely studied due to the inherent complexity of these systems and the presence of uncertainties such as load variations and weather conditions that make it difficult to use machine learning approaches for FDI attack detection.

Chapter 3

System model

This chapter provides an in-depth explanation of the under study system in this project. In Section 3.1, the background of the power flow analysis used to evaluate the system is discussed. The focus is on understanding the principles and methods of power flow analysis, which will provide a solid foundation for subsequent sections. In Section 3.2, the distribution systems modeled in this project are explained. This section discusses the different aspects of the distribution systems, such as their design and operation, and provides a detailed overview of the modeling process. Section 3.3 of this chapter is dedicated to explaining the PV system implemented in this project. The section covers various aspects of the PV system, including its components, characteristics, and operation. Additionally, the section also provides a detailed explanation of the modeling and simulation of the PV system. Finally, in Section 3.4, the voltage regulation system is explained. This section provides a detailed overview of the voltage regulation system implemented in the study, including its design, operation, and control.

3.1 Power Flow

In this section, we briefly discuss the required background for the readers to understand the power flow mechanisms of the power system. Understanding this mechanism is necessary for obtaining the operating point in transient operation and the system parameters in the steady state.

Power flow analysis is a simulation method employed to model the electric power flow within a power system network. The key goal of power flow analysis is to determine the steady-state behavior of the system, including power flow and voltages at each bus, under normal operating conditions or various contingency situations. To solve power flow problems, the Newton-Raphson method is a numerical approach commonly employed for solving non-linear algebraic equations based on the Newton-Raphson iteration algorithm that is employed to calculate the voltage magnitude and phase angles at each bus in a power system. The method is widely used due to its ability to converge rapidly and accurately to the solution, providing highly accurate results with a relatively low computational expense [52].

The behavior of electric power within a power system network can be modeled using power flow analysis. The power flow equations are a set of non-linear equations that describe the relationship between the voltage magnitude, phase angle, and real and reactive power injections at each bus in a power system. The aim of power flow analysis is to determine the voltage magnitude and phase angle at each bus that satisfies these equations. To begin the Newton-Raphson method, an initial estimation of the voltage magnitude and phase angles at each bus is made. This estimation is then utilized to calculate the real and reactive power injections at each bus using the power flow equations. The deviation between the computed and the actual power injections is then used to build the Jacobian matrix, which is then used to update the voltage magnitude and phase angles in the next iterations. This procedure is repeated until the difference between the computed and the actual power injections is within an acceptable tolerance level.

In a network containing N nodes, a linear equation can be used to express the relationship between the current of a particular node, which is denoted by I_k, and the voltage of that node with respect to the reference voltage, which is denoted by E_k. The admittance matrix of the network has an element given by Y_{km}, and the use of a bar notation indicates that the values are complex.

$$\bar{I}_k = \sum_{m=1}^{N} \bar{Y}_{km} \bar{E}_{km} \tag{1}$$

The mathematical expression for the complex power at node k is represented by the following equation:

$$(P_k + jQ_k) = \bar{E}_k \sum_{m=1}^{N} \bar{Y}_{km} * \bar{E}_m \tag{2}$$

In order to solve the power flow problem, we must determine the real and reactive power (P_k and Q_k) entering node k, which involves using the imaginary unit j = $\sqrt{-1}$ and a complex conjugate. The power flow problem consists of a set of $(N-1)$ nonlinear equations from equation (1), which must be solved iteratively. Newton's method can be utilized if the Jacobian matrix can be calculated accurately.

The iterative algorithm for solving an unadjusted power flow problem using Newton's method involves five stages. The first stage involves obtaining an initial approximation for the voltage solution by setting the slack node voltage and assigning magnitude values to the given voltages. A per-unit system is used, and the angles are initialized as the slack node angle. It is assumed that the initial approximation of the voltages are 1 p.u. and angles are 0 rad.

In the second step, one iteration of the successive displacement method is performed to ensure a good starting point without over-correction. In the third step, the residuals are augmented with the Jacobian matrix.

In the fourth stage, the correction for the node voltages is determined by applying Gaussian elimination and back-substitution methods to solve the Jacobian matrix, which becomes an upper triangular matrix. The residuals are organized into a polar column that contains corrections in angle and magnitude. These calculated correction values are then added to the node voltages.

The last stage involves evaluating the residuals ΔP_k and ΔQ_k. If they are found to be small enough, then the problem is considered solved. Otherwise, the algorithm is repeated from stage three.

To sum up, the iterative algorithm based on Newton's method is a structured approach for solving the unadjusted power flow problem. It involves a sequence of steps that include successive displacement, formation of the Jacobian matrix, computation of voltage corrections, and assessment of residuals to approach the voltage solution iteratively.

The iterative algorithm of the Newton-Raphson method is explained in Algorithm 1.

```
Algorithm 1: Power Flow Iteration Algorithm
   Input: Power flow equations, tolerance $\epsilon$
   Output: Voltage magnitude and phase angle at each bus
1  Initialize voltage magnitude and phase angle at each bus as $V_0$ and $\theta_0$;
2  Set iteration counter $k = 0$;
3  Calculate real and reactive power injections at each bus using power flow equations;
4  Compute mismatch vector $\mathbf{F}(V_k, \theta_k)$ and Jacobian matrix $\mathbf{J}(V_k, \theta_k)$;
5  Solve for voltage correction $\Delta V_k$ using $\mathbf{J}(V_k, \theta_k)\Delta V_k = -\mathbf{F}(V_k, \theta_k)$;
6  Update voltage magnitude and phase angle at each bus as $V_{k+1} = V_k + \Delta V_k$ and
    $\theta_{k+1} = \theta_k + \Delta\theta_k$;
7  Calculate real and reactive power injections at each bus using updated voltage magnitude
    and phase angle;
8  Compute new mismatch vector $\mathbf{F}(V_{k+1}, \theta_{k+1})$ and check if $||\mathbf{F}(V_{k+1}, \theta_{k+1})|| < \epsilon$;
9  while $||\mathbf{F}(V_{k+1}, \theta_{k+1})|| \geq \epsilon$ do
10     Increment the iteration counter and repeat from step 4;
11 end
```

Note that this algorithm assumes the power flow equations are already formulated and available for use. Also, $\Delta\theta_k$ is the angle correction vector, which can be computed from ΔV_k and $\mathbf{J}(V_k, \theta_k)$, but is omitted from the algorithm for simplicity.

3.2 PDS architecture

In this study, a modified version of the IEEE 33-bus and 141-bus PDS as a test system is used, which includes the addition of PV panels and a voltage regulation system. The IEEE 33-bus test system in the transient condition is simulated and a power flow analysis of both the IEEE 33-bus and 141-bus systems in the steady-state condition is performed. By using this methodology, the power flow characteristics of the system and regulation of the voltage with the help of the integrated PV panels is elaborated. This approach can help to identify the effect of uncertainties such as weather conditions and load variations on the voltage regulation system of a PV-integrated PDS, and can

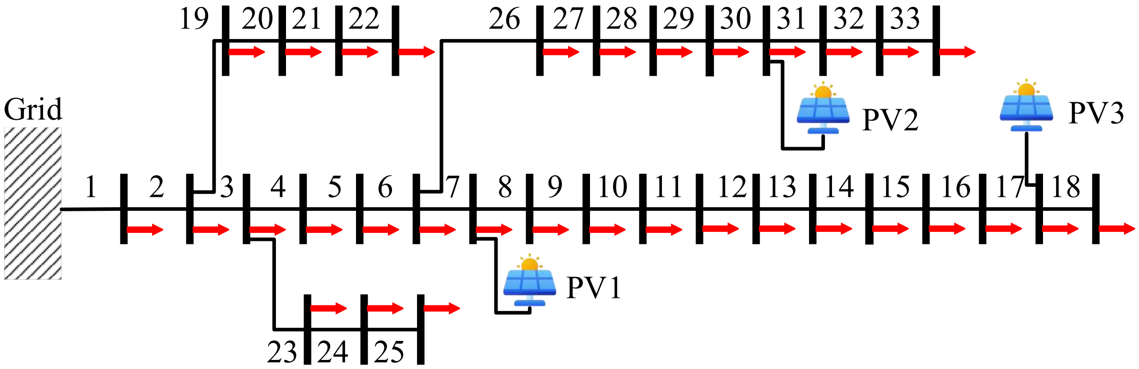


Figure 3.1: IEEE 33-bus PDS architecture

provide valuable insights into using ML approaches for FDI attack detection in such systems.

This study focuses on PDS which acts as the last stage in the delivery of electricity to consumers by transferring energy from transmission lines to various end-users. The main objective of this system is to maintain acceptable operating voltages while supplying energy to consumers. For this study, the IEEE 33-bus and 141-bus system were chosen as the test system. Modifications were made on IEEE 33-bus by adding PV panels at buses 7, 17, and 30 [47]. These PV panels were incorporated into the system to generate electricity from solar energy and provide sustainable energy solutions to consumers. The addition of PV panels can affect the voltage regulation system of the PDS, which is why this study focuses on detecting cyber attacks targeting the voltage regulation system of PV-integrated PDSs. The IEEE 33-bus system illustrated in Fig. 3.1 comprises one high-voltage feeder, 33 buses, and three PV panels. Its nominal operating voltage is 12.66 kV, with maximum active and reactive powers of 3.715 MW and 2.3 MVAR, respectively. There are 37 branches in the system, and it experiences losses of 0.09 MW and 0.06 MVAr of reactive power. The system's minimum voltage magnitude is 0.91 p.u. at bus 18, and its maximum voltage magnitude is 1.000 p.u. at bus 1. The minimum voltage angle is -0.29 degrees at bus 18, and the maximum voltage angle is 0.34 degrees at bus 30. For more details on this network, please refer to [16, 51].

In addition, the 141-bus PDS referred to here is an electrical power system that consists of 141 buses, 140 branches, 1 connection to the transmission grid, and 83 loads. The purpose of this system is to distribute electrical power to consumers, and it is depicted in Fig. 3.2. To study the impact of PV panels on this system, 4 PV panels were added to buses 52, 77, 106, and 111. According to [21],

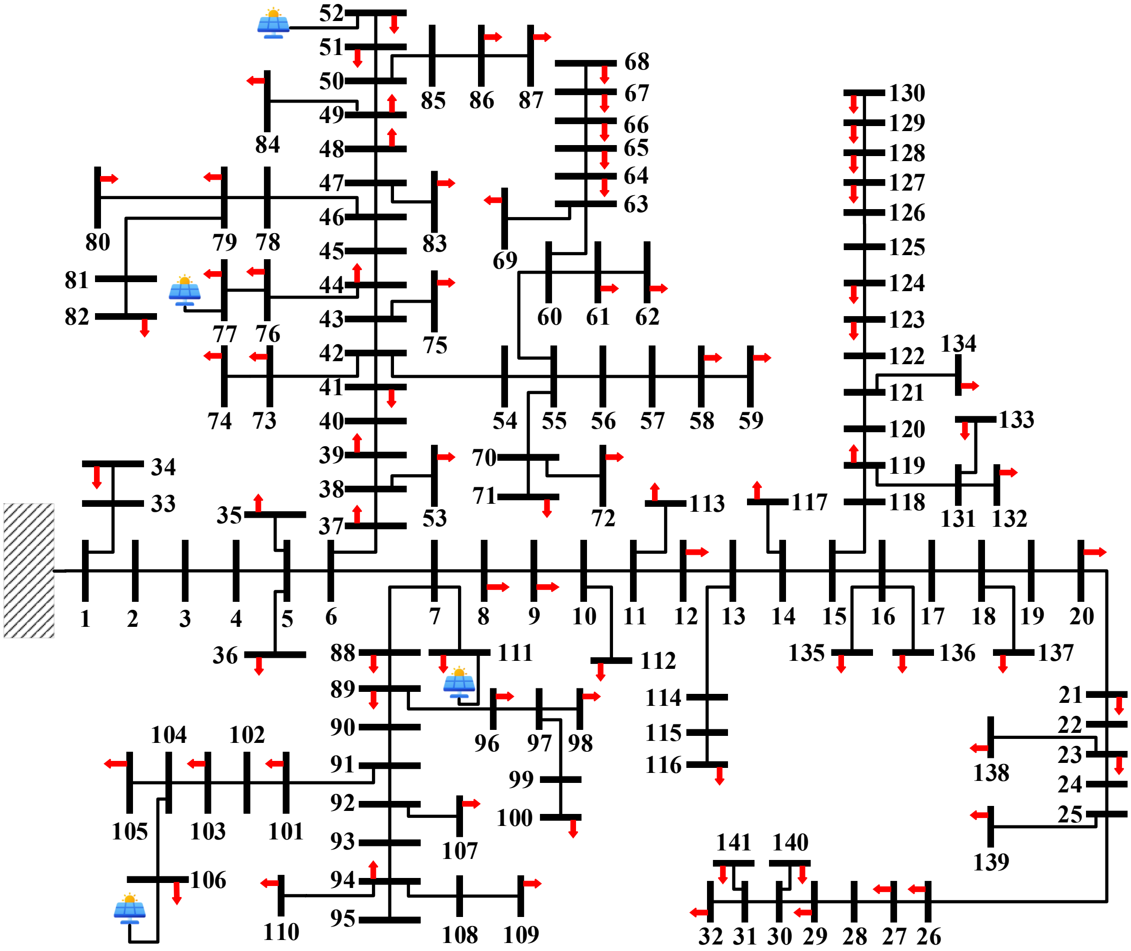


Figure 3.2: 141-bus PDS architecture

the total capacity of the system is 100 MW, which is provided by the connection to the transmission system. The total load demand in the system is 8.2 MW and 5.1 MVAr of reactive power. In this benchmark, all loads within the PDS are often assumed to be fixed. However, the present study modifies the loads to be variable, meaning they can fluctuate according to the system's needs and the details of the load variation are explained in Subsections 3.4.1 and 3.4.2 for transient and steady-state conditions respectively. The voltage magnitude in the system with fixed loads ranges from 0.89 per unit (p.u.) at bus 52, 87 to 1 p.u. at bus 1, while the voltage angle ranges from -0.19 degrees at bus 141 to 0.00 degrees at bus 1.

3.3 PV System

A PV cell is a semiconductor device that converts sunlight into electrical energy. When sunlight hits the PV cell, it excites the electrons in the semiconductor material, causing them to flow and generate a direct current (DC) which means that the current flows in only one direction. The amount of power generated by the PV system depends on various factors such as the amount of sunlight falling on the panels, the temperature of the panels, the angle at which the panels are installed, and so on. The PV module's efficiency decreases as the temperature increases, resulting in reduced power output. The decrease in efficiency is due to the increase in the number of free electrons and the decreased mobility of these electrons in the semiconductor material, resulting in lower voltage output.

According to the architecture of a grid-tied PV system, depicted in Fig. 3.3, to generate power, solar panels produce DC electricity which varies based on the intensity of sunlight and temperature. In order to maintain efficiency, two DC-DC converters are required to regulate and increase the DC voltage to a specific level using Maximum Power Point Tracking (MPPT). The DC voltage output must then be converted into alternating current (AC) to be utilized in the PDS, and inverters play a crucial role in this conversion process. This architecture is simulated in Matlab/Simulink as depicted in Fig.3.4.

MPPT technology is used in PVs. MPPT works by continuously tracking the maximum power point (MPP) of the solar panel, which is the point at which the panel produces the maximum possible power for a given level of sunlight and temperature. The MPP is typically found at a specific voltage and current combination that maximizes the power output of the panel. The MPPT controller uses an algorithm to constantly adjust the load on the panel to keep it operating at the MPP [18]. The MPPT controller accomplishes this by measuring the voltage and current of the panel and then computing the power output. This process is repeated continuously, ensuring that the solar panel operates at the MPP under all conditions. Figure. 3.5 illustrates the MPPT curve for a constant temperature, showing how the maximum power output of the PV is affected by adjusting voltage and current. For example, at a temperature of 25 C°, the maximum power generated by the PV is 100 kW.

The MPPT algorithm used in this project is Perturb & Observe algorithm. This algorithm is

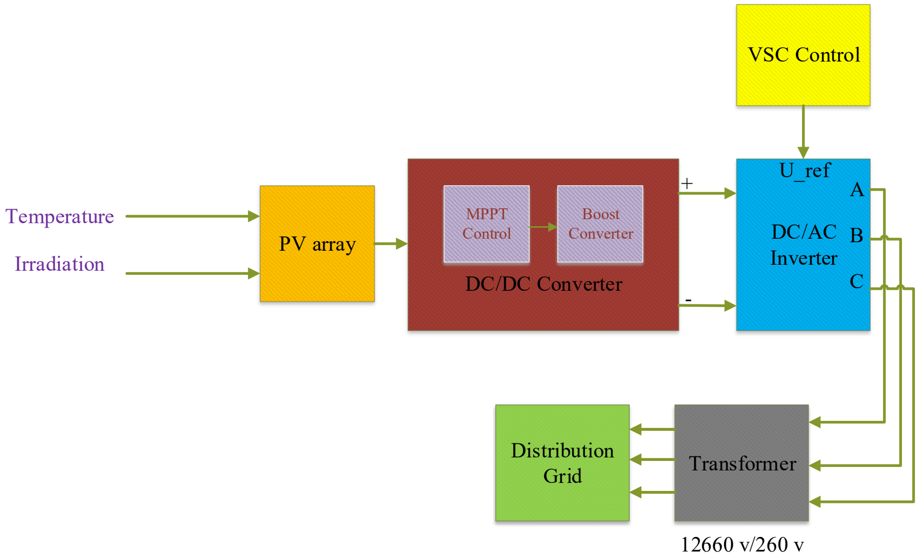


Figure 3.3: Grid-tied PV system

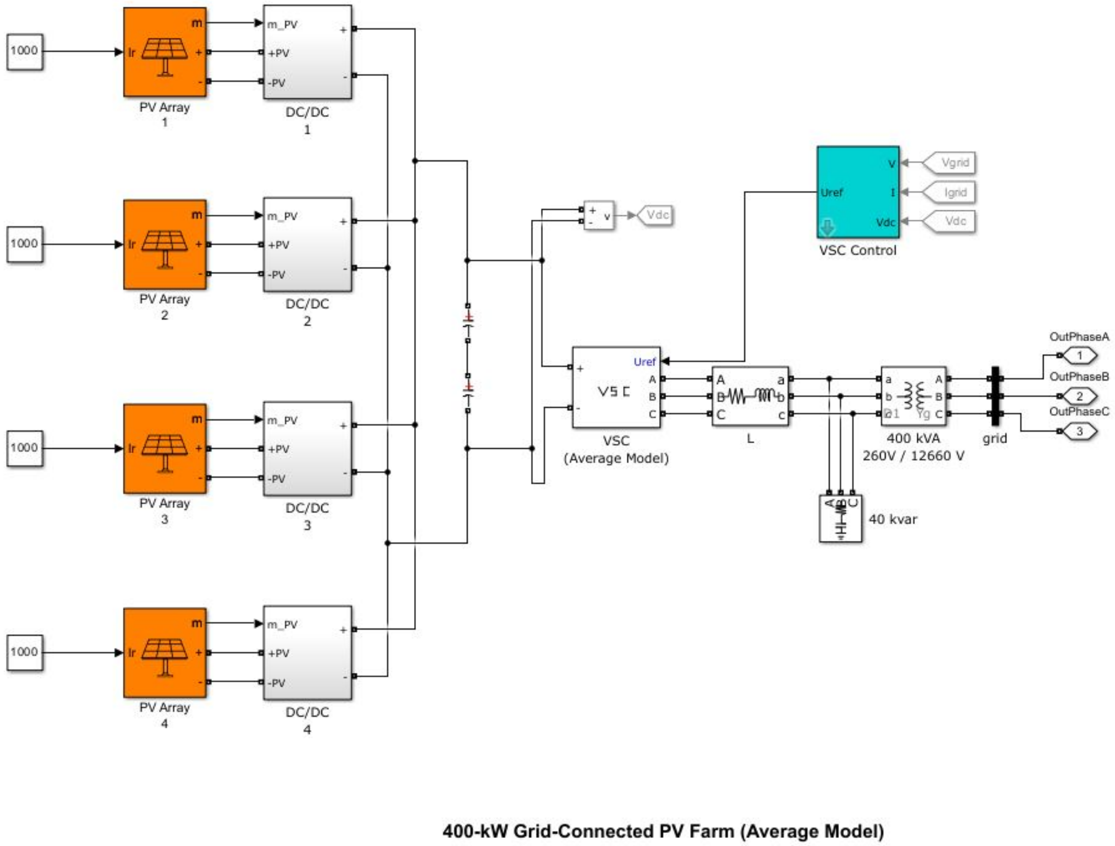


Figure 3.4: Grid-tied PV system model simulated in MATLAB/Simulink

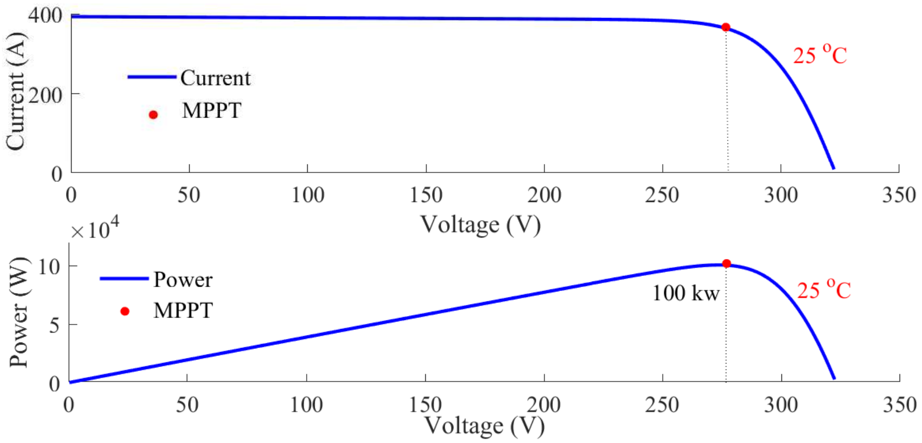


Figure 3.5: Current-Voltage and Power-Voltage curve at constant temperatures

a simple and widely used MPPT algorithm that works by perturbing (i.e., changing) the operating voltage of the solar panel slightly and observing the effect on the output power; thus, controlling the operating point of the system. If the output power increases, the voltage is perturbed further in the same direction. If the output power decreases, the voltage is perturbed in the opposite direction. This process is repeated until the MPP is reached, and the operating voltage is then fixed at this value. The Perturb & Observe algorithm can be enhanced by introducing a control loop that adjusts the perturbation step size according to the rate of change of the power output, so that the system can converge to the MPP quickly and accurately [14, 20]. The MPPT control Perturb & Observe algorithm is widely used in PV systems because of its simplicity, effectiveness, and low cost. However, it has some limitations, such as susceptibility to oscillations around the MPP in rapidly changing conditions and reduced efficiency at low light levels. Modeled DC to DC converter used in this project is depicted in Fig. 3.6. MPPT Control block is implemented Perturb & Observe algorithm, and the Boost block is boost converter. A boost converter is a type of DC to DC converter that steps up the input voltage to a higher output voltage. Additionally, there is an LC filter in the figure because The output of a PV panel is usually a DC voltage that contains some ripple due to factors such as the fluctuation of solar irradiance, the temperature variation of the panel, and the switching of power electronics. This ripple can cause unwanted effects on the power grid and the loads connected to it, such as electromagnetic interference and voltage/current distortion. By using an LC filter, which consists of an inductor (L) and a capacitor (C) connected in series, the ripple voltage

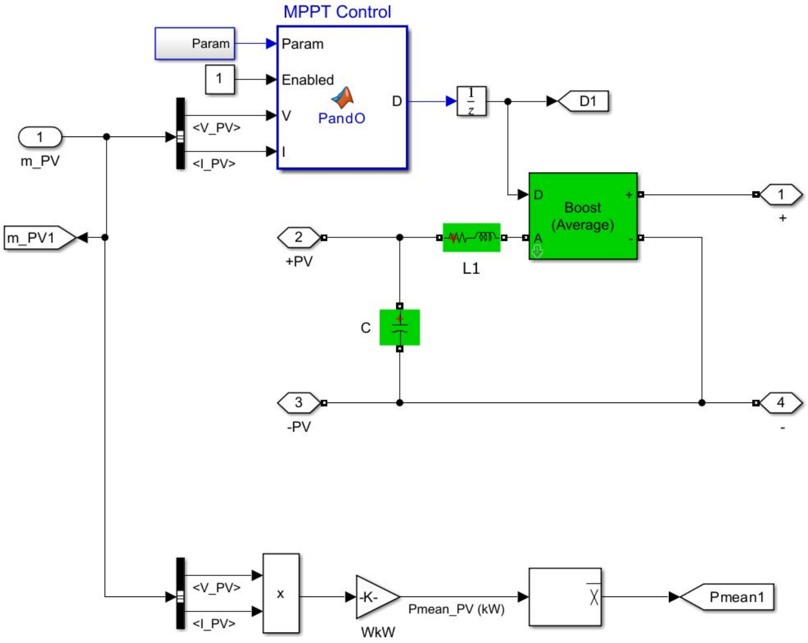


Figure 3.6: DC to DC block

and current can be smoothed out by selectively filtering out the high-frequency components of the output waveform.

The output of the PV system is usually at a voltage and current level that is not suitable for most applications. In order to make the power usable, it needs to be converted into a different form. This is done by an inverter, which converts the DC power into AC power that can be used by appliances and devices.

In this thesis, an average model is used for simulating the inverter. An average model inverter is a type of inverter that uses a simplified mathematical model to convert DC power from a PV system to AC power. This model uses average values of the input and output signals to simulate the inverter's behavior. The DC power is first converted to a high-frequency AC signal, which is then filtered and transformed to a low-frequency AC signal that is suitable for use by appliances and devices.

The considered inverter model, which is implemented in MATLAB/Simulink, consists of a total of four distinct blocks,i.e., Phase-locked loop (PLL), VDC regulator, current regulator, and U_ref block that are shown in Fig. 3.7. The PLL block within the inverter model is a crucial component that performs two distinct functions. First, it contains a PLL control system, which is responsible

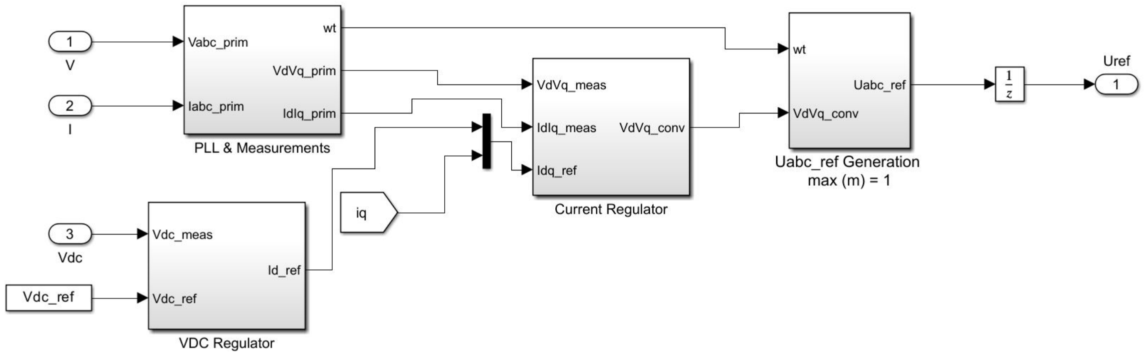


Figure 3.7: VSC control system model simulated in MATLAB/Simulink

for generating a highly accurate and stable frequency reference signal for the inverter. It works by comparing the frequency of the inverter's output waveform with the frequency of a stable reference signal and then adjusting the inverter's output frequency accordingly to eliminate any discrepancies. This process ensures that the inverter produces a highly stable output waveform that is synchronized with the frequency of the reference signal. Second, it incorporates a dq0 transformation module that converts the 3-phase measurements obtained by the inverter into 2-dimensional dq measurements, which can be easily used by the PV proportional-integral (PI) controllers. The PI controller compares the measured dq currents with a set of reference values and adjusts the inverter's output accordingly to maintain the desired current levels. This transformation enables the PI controller to accurately regulate the current flowing through the inverter's components, ensuring that it remains within safe limits and operates efficiently.

The VDC block is a vital component of the inverter model that serves the purpose of regulating the voltage of the direct current V_{dc} supply. It achieves this by generating a reference signal for the inverter's direct current component (I_d) based on the measured V_{dc} and the desired V_{dc} reference value. This reference signal is then used by a PI controller to maintain the V_dc at the desired level.

The current regulator block is a fundamental component of the inverter model that plays a critical role in regulating the electrical current flowing through the system. The block takes various input measurements, including (I_d) and (I_q) measurements, (I_d) and (I_q) reference values, and (V_d) and (V_q) measurements. It then uses a combination of proportional-integral (PI) and feedforward controllers to generate converted (V_d) and (V_q) values, which are used to control the inverter's output. The PI controller in the current regulator block processes the difference between the measured and desired (I_d) and (I_q) values to generate an appropriate control signal. The feedforward controller takes the measured (V_d) and (V_q) values and feeds them into the system's control algorithm. This controller is responsible for predicting the required control signal based on the previous input measurements, improving the system's overall response time and stability. The output of the current regulator block is the converted (V_d) and (V_q) values, which are used to control the inverter's output. These values are fed into the inverter's dq0 transformation module, where they are transformed into 3-phase values to generate the inverter's output waveform. The current regulator block's primary function is to ensure that the electrical current flowing through the inverter is maintained within safe and acceptable limits. By precisely controlling the converted (V_d) and (V_q) values, the block regulates the electrical current flowing through the inverter's components, ensuring that they operate efficiently and safely.

The output of the inverter is a critical component that determines the characteristics of the electrical signal generated by the system. It is generated by the U_{ref} block, which takes two input signals: the angle frequency (wt) generated by the PLL block, and the converted voltage signals (V_d and V_q) generated by the current regulator block. The U_{ref} block combines these input signals to generate an output signal that is used to adjust the inverter's output voltage and current to the desired levels. The block applies a set of control algorithms that take into account the desired reference values for voltage and current, as well as any external disturbances or variations in the system's operating conditions. The resulting output signal is fed back into the inverter's electrical circuit, where it is used to control the power delivered to the load or to the electrical grid.

3.4 Voltage Regulation Scheme

Voltage regulation schemes are a critical component of PDS as they maintain voltage magnitudes within specific limits and ensure the proper functioning of the system. Voltage levels can vary due to changes in load and generation, and it is essential to keep them within the acceptable range to maintain a stable and secure power supply. Voltage regulation schemes achieve this by controlling the voltage at specific points, such as the point of common coupling (PCC), by monitoring and adjusting the voltage levels. Control systems adjust the reactive power output of generators and regulate the voltage to ensure that voltage magnitudes remain within predefined tolerances. It is important to note that the acceptable voltage range in a PDS can vary depending on the standards set by the operator. For example, the ANSI C84.1 voltage standard ranges from 0.9 per unit (pu) to 1.05 pu according to [2], while it ranges from 0.95 pu to 1.05 pu according to [34]. For the purposes of this study, the desired voltage range is assumed to be within 0.9 pu to 1.05 pu.

Regulatory agencies and industry standards typically set the acceptable range of voltage magnitudes. Voltage regulation schemes must take into account various factors that can affect the voltage levels, such as changes in load and generation. For instance, when there is an increase in load demand, the voltage levels tend to decrease. In contrast, an increase in generation causes the voltage levels to increase. Voltage regulation schemes must balance these factors and ensure that the voltage levels remain within the acceptable range.

There are different types of voltage regulation schemes, such as on-load tap changer (OLTC), static voltage regulator (SVR), and shunt reactors, that use different methods to regulate voltage. OLTCs adjust the voltage levels by changing the tap positions of transformers. SVRs use electronic switches to control the voltage levels, while shunt reactors absorb excess reactive power to maintain voltage levels. The choice of voltage regulation scheme depends on the specific requirements of the PDS. It should be noted that the implementation of all these devices incurs additional costs to the grid, and hence is not the best solution from the operator's perspective. On the other hand, the use of PVs for voltage generation is almost cost-free for the operator since these devices are often used for active power generation, and regulating voltage is an ancillary service that they can provide.

In order to ensure the efficient and reliable functioning of a PV system, there are the following constraints that must be adhered to during its operation:

1- The operation of the i-th PV panel depends on the capacity constraint, which enforces that the total active (P_i) and reactive power (Q_i) output of the converter must not exceed a predefined value of apparent power (S_i) [72]. It is assumed that the active power generated by each individual PV unit is 100 kW and its apparent power capacity is 500 kVA. This constraint is particularly important because exceeding the apparent power limit could result in system instability, voltage fluctuations, and other power quality issues. In other words, the PV units are designed to operate within a specific range of power output, and exceeding this range could result in system failure. Additionally, this constraint may also be translated into a reactive power constraint.

$$\begin{gathered} Q^2 \leq S^2 - P^2 \Longrightarrow Q^2 \leq 500^2 - 100^2 \\ Q^2 \leq 240 \, kVAR \end{gathered} \tag{3}$$

2- The acceptable range for normal operation is predefined, and it is essential to ensure that the PV system operates within this range. Operating outside the acceptable voltage range can cause equipment damage, system failure, and a reduction in system efficiency. To comply with this constraint, the study proposes a voltage regulation system, which regulates the voltage levels within the acceptable range by adjusting the reactive power output of the PV system.

The upcoming two subsections, namely Subsection 3.4.1 and Subsection 3.4.2, provide information on the voltage regulation scheme that has been implemented in both transient and steady-state conditions.

3.4.1 Transient conditions

In transient conditions, voltage regulation schemes aim to stabilize voltage levels after a disturbance or change in system operating conditions. This can occur due to various factors, such as sudden changes in load demand or the loss of a generator. Voltage regulation systems typically use feedback control to rapidly respond to these changes and adjust the system's reactive power output to restore voltage levels to within acceptable limits. For example, in the event of a sudden increase in load demand, the voltage regulation system may increase the reactive power output of the generators to compensate and stabilize the voltage levels.

During transient conditions, the load is sinusoidally varying within a range of -1 to +1. In each step, the fixed load value is multiplied by the sinusoidal value. It is assumed that the load in transient conditions can fluctuate up and down by 100% to account for all possibilities and demonstrate comprehensive variations.Additionally, inputs of the PV panels (irradiation and temperature) are considered to be fixed because the time step used in the analysis is very small (in milliseconds). Thus, any small changes in the weather conditions that may affect the output power of the PV

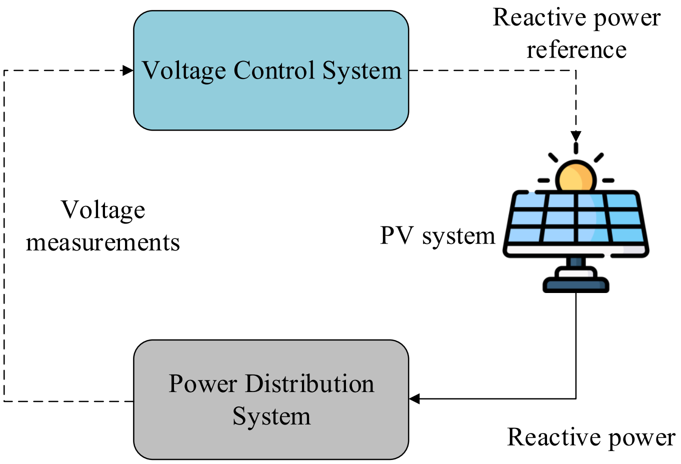


Figure 3.8: PV-integrated IEEE 33-bus power system

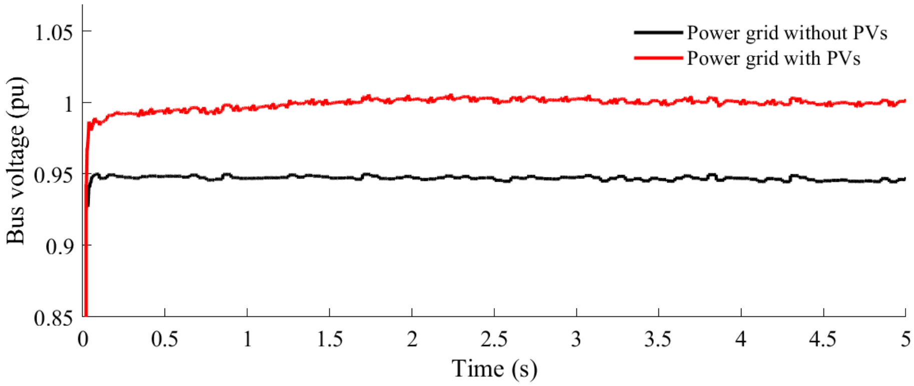


Figure 3.9: Impact of PV arrays on the voltage regulation

panels are assumed to have a negligible impact on the overall output power, and can be ignored for the purpose of analysis.

The voltage regulation method depicted in Fig. 3.8 involves a central proportional-integral (PI) controller that acquires voltage measurements from the IEEE 33-bus PDS. The controller then computes the necessary current to generate reactive power to adjust the voltage. The PDS being studied here involves the installation of two additional PVs on Buses 18 and 33, which have lower voltage levels than the others. Fig. 3.9 displays how the central PI controller and PVs in the PDS under analysis can regulate voltage to maintain it within the desired range as the system load fluctuates.

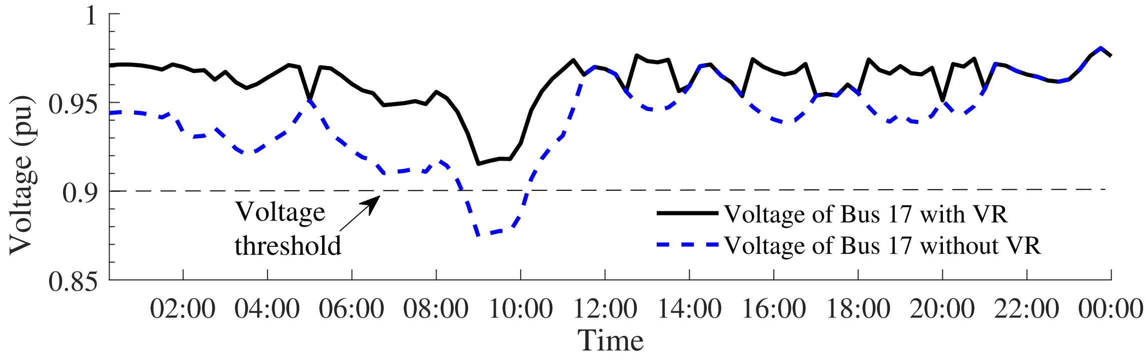


Figure 3.10: Voltage Regulation system impact

3.4.2 Steady-state conditions

In steady-state conditions, voltage regulation schemes aim to maintain voltage levels within specified limits over a more extended period. This requires a more nuanced approach to control, as the system must balance reactive power output and other operational factors to ensure stable and reliable power supply. Voltage regulation systems in steady-state conditions use a variety of control strategies, such as voltage control and droop control, to maintain voltage levels within acceptable limits. For example, voltage control systems use feedback control to regulate the reactive power output of the generators and adjust voltage levels to within specified tolerances. Droop control, on the other hand, uses a distributed control strategy to maintain voltage levels by adjusting the reactive power output of the generators based on their operating characteristics and load conditions. Overall, voltage regulation schemes are critical to ensuring the stability and reliability of PDS, particularly as the system operates under varying conditions.

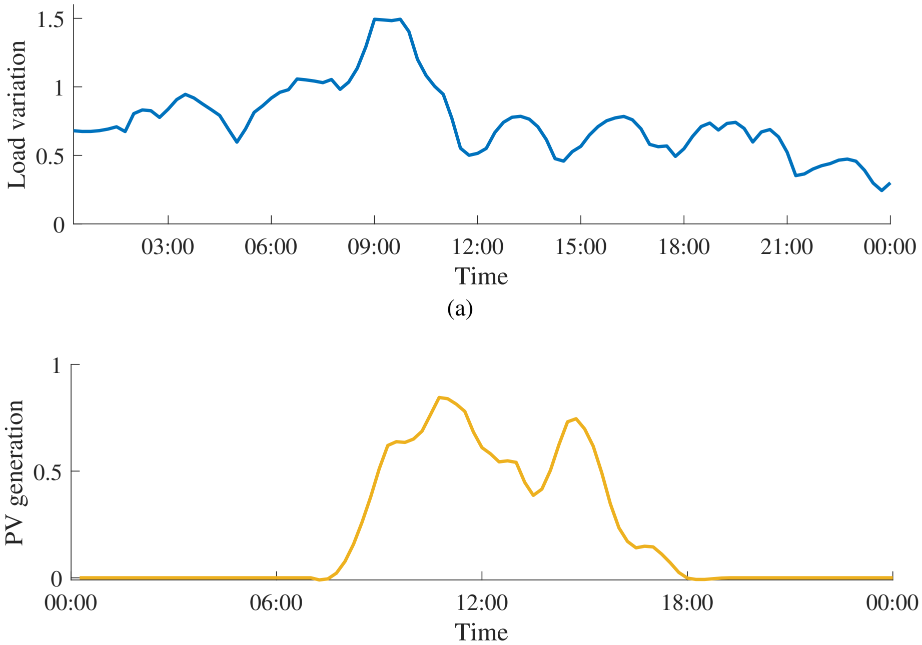
Fig. 3.10 displays the effect of utilizing PVs for voltage regulation on the voltage levels of the IEEE 33-bus PDSs. The method employed to regulate voltage in such a system is elaborated in [4] with a focus on its transient mode of operation.

The following Algorithm 2 has been used in this project for regulating voltage in steady-state conditions.

In steady-state conditions, the load variations and PV generations are determined by referring to the historical data. The data for historical load and PV profiles were collected from AESO datasets over a period of two years [1]. Figure. 3.11 illustrates an example of a daily load profile and PV

Algorithm 2: Voltage Regulation Algorithm

```
Input: Power flow data ($PFD$) of the system model, Acceptable voltage range ($VR$),
       Maximum reactive power limit ($RPL$), Time step size ($TS$), Simulation duration
       ($SD$)
Output: Regulated voltage value ($RV$)
Extract voltage measurements ($VM$) from $PFD$ for buses with PVs and voltage regulation
  system;
while Simulation duration $< SD$ do
    $VM' = VM$;
    while $VM'$ not in Acceptable voltage range $VR$ do
        Reactive power $RP = 0$;
        if $VM' < VR_{low}$ then
            $RP = RPL$;
        end
        if $VM' > VR_{high}$ then
            $RP = -RPL$;
        end
        Inject or absorb reactive power $RP$ to regulate voltage; Check constraints such as
          the apparent power of the PV's converter; Run power flow to get updated $VM'$;
    end
    Wait for $TS$ minutes; Simulation duration $= SD + TS$;
end
return Regulated voltage value $RV$;
```



(a)

(b)

Figure 3.11: (a) Load profile (b) PV generation profile

generation. Specifically, the load values are updated every 15 minutes based on the data provided in the dataset, which covers a period of two years.

Chapter 4

Threat model

The FDI attack is a type of deliberate manipulation of sensor measurements in a PDS and its objective is disturbing the voltage regulation scheme of the system. This type of attack has the potential to cause considerable harm to the power system, including power outages and financial losses. The attack involves introducing false data into the voltage measurements, resulting in the generation of inaccurate control signals by the voltage regulation controller. This type of attack is a serious threat to the stability and dependability of the power system and requires efficient measures to counteract it and avoid damage.

4.1 Threat Model

To carry out an attack on the power system, the attacker can modify the voltage measurements that are sent to the controller. This modification can be expressed as a manipulation of the measurement vector using constants a and b, and the current measurement vector $\boldsymbol{V}(t)$, denoted as:

$$\boldsymbol{z} = a \times \boldsymbol{V}(t) + b \tag{4}$$

Here, $\boldsymbol{V}$ represents the true voltage measurements, t denotes time, and $\boldsymbol{z}$ is the attack vector.

Particle Swarm Optimization (PSO) is a popular optimization algorithm that is useful for determining the optimal values of parameters in an FDI attack. In this study, PSO was used to optimize

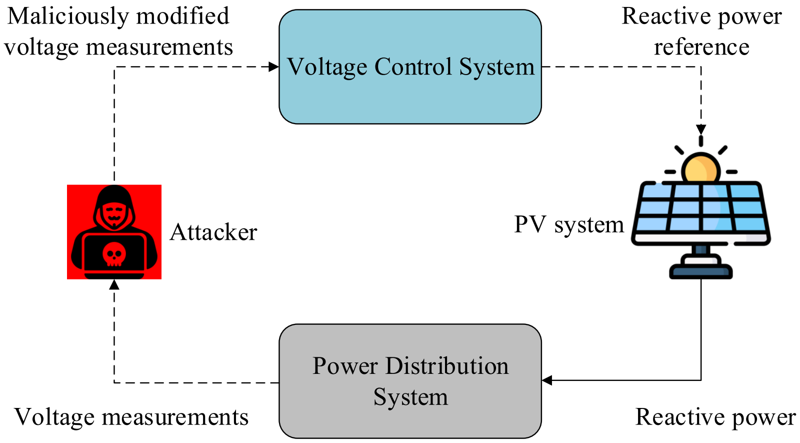


Figure 4.1: Implemented FDI attack in IEEE 33-bus power system.

the values of a and b in order to cause a voltage response that would take the voltage bus out of the acceptable range. The optimization was conducted for Bus 16 in the IEEE 33-bus and Bus 52 in the 141-bus system, and the objective was to minimize the difference between the measured voltage and the target value of 0.9 p.u. The search space consisted of all possible values of a and b, and the PSO algorithm was initialized with 50 particles, a maximum of 100 iterations, an inertia weight of 0.8, and learning factors. The fitness value was calculated for each particle in the swarm, and the particle positions and velocities were updated using the PSO update equations. This process continued until convergence was achieved or the maximum number of iterations was reached. The optimized values of a and b were determined to be a=1.05 and b=0.03. It is important to note that these values were selected to ensure that the FDI attack did not trigger the bad data detection (BDD) algorithms of PDS.

Figure. 4.1 illustrates the voltage regulation process and a potential FDI attack. Wireless communication, indicated by dashed lines, transmits measurement signals to the centralized controller in the IEEE 33-bus and 141-bus systems. In the FDI attack, the attacker obtains bus voltage measurements, maliciously modifies the voltage values, and then transmits them back to the centralized controller.

4.2 ICT vulnerability

This study employs IEC 60870-5-104 protocol to transfer voltage measurements from PDS to the central controller. It is presumed that the attacker has the ability to eavesdrop on this communication protocol and manipulate the voltage sensors without getting detected. IEC 60870-5-104 protocol is a widely accepted communication protocol used for real-time data transfer and control of electric power systems over wide-area networks (WANs) between remote terminal units (RTUs) and control centers. The attacker can carry out a man-in-the-middle (MITM) attack by altering the data sent between the PDS and the central controller. The attacker can use various techniques, such as ARP spoofing, DNS spoofing, or IP spoofing, to execute the MITM attack on IEC 60870-5-104 protocol, which reroutes the communication channel through the attacker's device. Once the communication is rerouted, the attacker can modify the voltage measurement values by changing the data based on (4).

4.3 FDI attack impact on the under study PDS' voltage

It is crucial to promptly detect and mitigate malicious activities that can have severe consequences on power system operation. During an FDI attack, the voltage levels of specific buses in a power system can undergo significant changes and deviate from the normal range. Fig. 4.2 demonstrates the adverse effect of an FDI attack on the voltage regulation of the IEEE 33-bus system in transient conditions. The reactive power injection is used to regulate the voltage in response to the load variation. Under normal conditions, the voltage is maintained within the acceptable range. However, when an attack is launched with $\alpha = 1.1$ and $\beta = 0.02$ between $t = 4s$ and $t = 7s$, the controller makes a wrong decision resulting in a significant drop in voltage below the acceptable threshold. Once the FDI attack is terminated at $t = 7s$, the controller takes corrective action to raise the voltage back to 1 p.u. and restore the operation. Machine learning algorithms are discussed in the next section for detecting such attacks.

The voltage levels of all buses in the IEEE 33-bus and 141-bus power systems during an FDI attack are illustrated in Figure 4.3 in the steady-state conditions. The negative effects of an FDI attack on the voltage regulation of the IEEE 33-bus and 141-bus systems in steady-state conditions

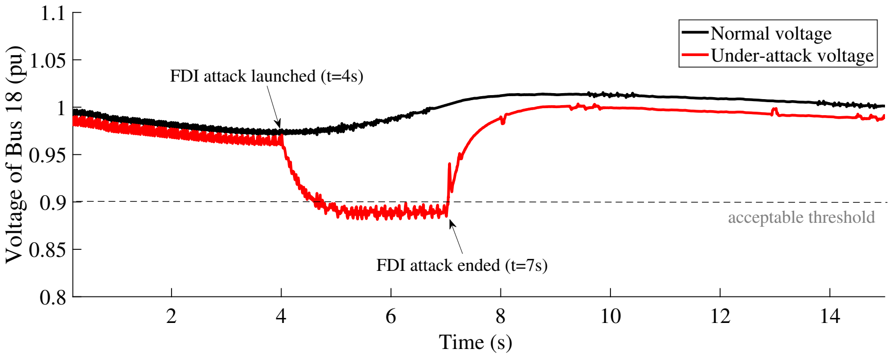
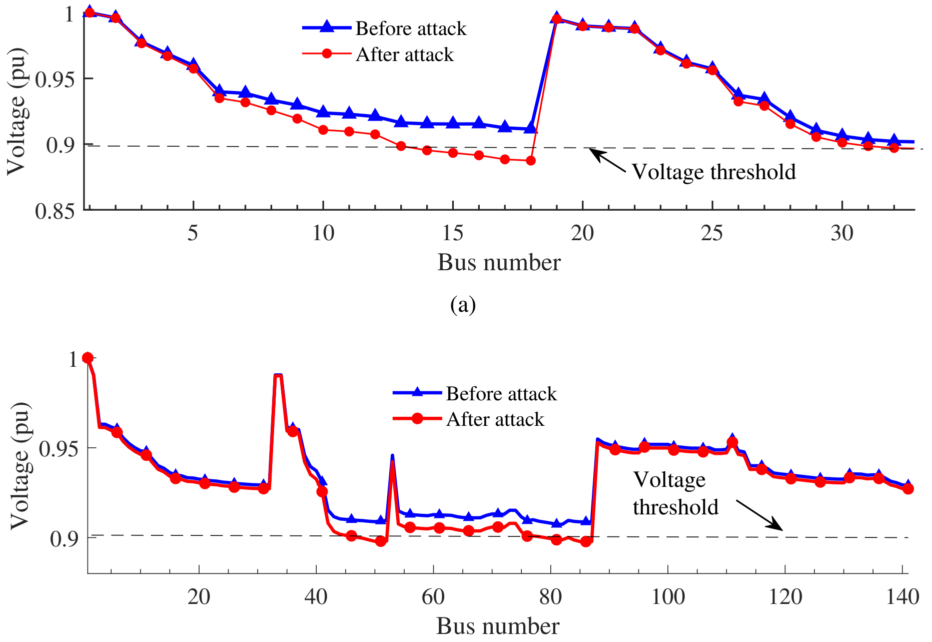


Figure 4.2: Implemented FDI's impact on the bus voltage in the transient conditions

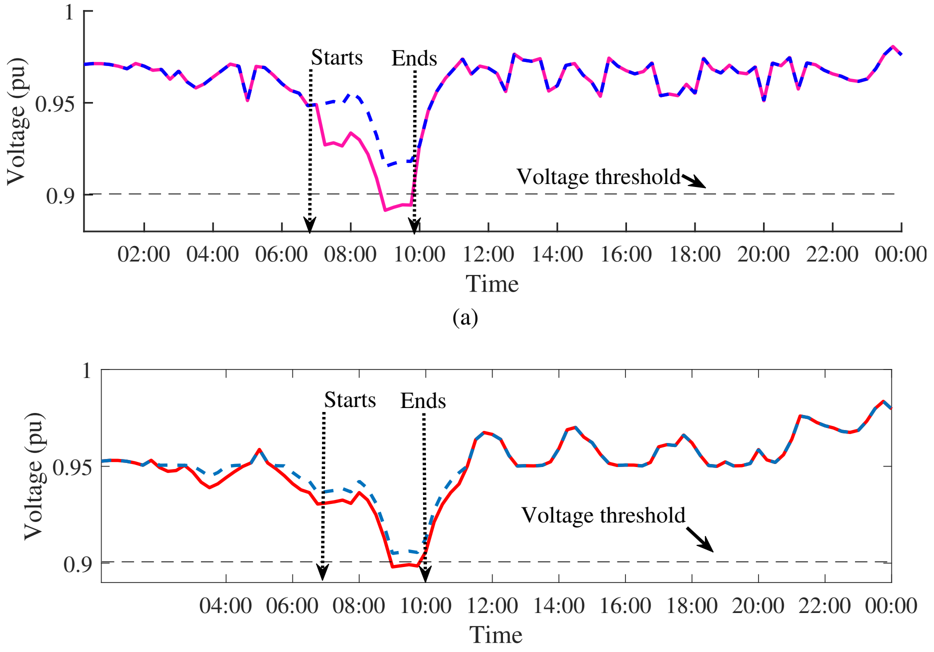
are demonstrated in Fig. 4.4 using time-domain analysis. The system experiences load variation as per a dataset of 15-minute power consumption data for two years, and the controller aims to maintain the voltage within the acceptable range by adjusting reactive power. The voltage levels are within the acceptable range during normal operation. However, an attack with the parameters of $\alpha = 1.05$ and $\beta = 0.03$, which starts at $t = 7AM$ and ends at $t = 10AM$, causes a significant decrease in the voltage levels below the acceptable range due to the controller's wrong decision. After the attack ends at $t = 10AM$, the controller tries to restore the voltage levels to $0.95pu$ to resume normal operation. The upcoming section elaborates on the deep learning algorithm utilized to detect this type of attack.



(a)

(b)

Figure 4.3: Bus voltages for (a) IEEE 33-bus and (b) 141-bus during FDI attack



(a)

(b)

Figure 4.4: Voltage of under attack bus in IEEE 33-bus and 141-bus in the steady-state conditions

Chapter 5

Machine learning and deep learning methods

ML has gained popularity in detecting cyber-attacks due to its ability to adapt to new and evolving attack techniques. Unlike traditional model-based systems, ML-based methods can learn complex patterns from large amounts of data, making them capable of detecting sophisticated attacks that may not be detected by human analysts. There are two main categories of ML algorithms: supervised and unsupervised methods. Supervised learning algorithms are trained using labeled data and feedback to predict the outcome of a process, while unsupervised learning systems use unlabeled data for training and do not rely on feedback. These methods can detect subtle anomalies and complex patterns that may go unnoticed by traditional rule-based methods. This project utilizes ML methods to address the problem of transient and steady-state conditions. In the following two sections, we will provide a detailed explanation of the details of the methods and how ML methods are applied in these conditions.

5.1 Poposed SVM framework

In this study, we have developed a framework based on a supervised ML algorithm, i.e., SVM. In this section, we discuss the details of the SVM algorithm and its performance. We also present several examples of other supervised and unsupervised methods in Appendix A, which are briefly used to detect FDI attacks in our test system, and compare the results with the proposed technique.

This particular method is commonly used for historical data classification and regression, although it is typically used as a classification algorithm. To use this algorithm, the data is first mapped onto an n-dimensional space, where n represents the number of features. Then, a hyperplane is used to distinguish between the various samples, as described in [78]. To achieve the best results, certain parameters such as the kernel type ('linear', 'rbf', 'poly', and 'sigmoid'), Gamma, C, and degree (for polynomial kernel) are used to fine-tune the SVM method. The kernel function is used to transform the samples mathematically, Gamma is used to adjust the decision boundary, and C is used to modify the penalty for error during data training. In this project, the values of these parameters are detailed in Section 6.1. In power grids, it is essential to take secure measures along with detecting measurements that have been targeted by attackers. If a secure measurement is misinterpreted as an under-attack data, it can trigger an alarm, causing financial damage to power grids. Thus, performance analysis is crucial to ensure that the detection system is both accurate and reliable.

SVM's main idea is to find a hyperplane in a high-dimensional space that best separates different classes of data points. The hyperplane is chosen to maximize the margin, which is the distance between the hyperplane and the nearest data points of each class. SVM can handle non-linearly separable data by mapping the data to a higher-dimensional space using a kernel function, which transforms the original data to a new feature space where the data can be linearly separated. The objective of SVM is to find the hyperplane that maximizes the margin while minimizing the classification error. Given a set of training data points x_i with corresponding labels $y_i \in -1, 1$, SVM solves the following optimization problem:

$$\begin{aligned} \underset{\boldsymbol{w}, b, \boldsymbol{\xi}}{\text{minimize}} \quad & \frac{1}{2}|\boldsymbol{w}|^2 + C\sum_{i=1}^{n} \xi_i \\ \text{subject to} \quad & y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0, i = 1, \ldots, n \end{aligned} \tag{5}$$

where $\boldsymbol{w}$ is the weight vector, b is the bias term, ξ_i is the slack variable, y_i is the class label for the ith example, $\boldsymbol{x}_i$ is the feature vector for the ith example, C is a hyperparameter that controls the trade-off between maximizing the margin and minimizing the classification error, and n is the number of training examples. The objective function aims to minimize the L2 norm of the weight vector while penalizing examples that violate the classification rule. The slack variables ξ_i allow for some flexibility in the classification rule by allowing some examples to be misclassified, but at a cost to the objective function. The constraints ensure that all examples are classified correctly (or within some margin) and that the slack variables are non-negative.

If the data is not linearly separable, a kernel function can be used to map the data to a higher-dimensional space. The most commonly used kernel functions are:

- Linear kernel: $K(x_i, x_j) = x_i^T x_j$
- Radial basis function (RBF) kernel: $K(x_i, x_j) = \exp(-\gamma ||x_i - x_j||^2)$
- Polynomial kernel: $K(x_i, x_j) = (x_i^T x_j + c)^d$
- Sigmoid kernel: $K(x_i, x_j) = \tanh(\gamma x_i^T x_j + c)$ where γ, c, and d are kernel parameters that can be tuned to achieve better performance.

A comprehensive explanation of the various stages involved in implementing the proposed approach, starting from data collection and concluding with evaluation is provided in Section 6.2. The chapter offers a detailed account of the methods and techniques employed at each stage, along with a description of the relevant parameters and assumptions made during the simulation. Additionally, it includes information about the software tools and technologies utilized to perform the simulations, along with their respective roles in the implementation process.

5.2 Proposed CNN algorithm

Deep learning, which is a type of supervised machine learning, has also been increasingly used for attack detection in power grids. Power grids are critical infrastructures that require reliable and robust security measures to protect against cyber-attacks that can cause blackouts and compromise the safety of the grid. In this section, information about the proposed deep learning-based CNN is provided.

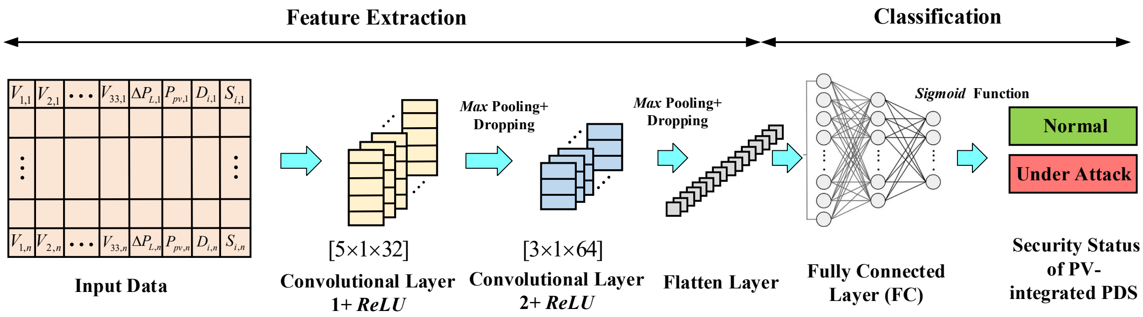

Feature Extraction
Classification
$V_{1,1}$ $V_{2,1}$ … $V_{33,1}$ $\Delta P_{L,1}$ $P_{pv,1}$ $D_{i,1}$ $S_{i,1}$
$V_{1,n}$ $V_{2,n}$ … $V_{33,n}$ $\Delta P_{L,n}$ $P_{pv,n}$ $D_{i,n}$ $S_{i,n}$
Input Data
[5×1×32]
Convolutional Layer 1+ *ReLU*
Max Pooling+ Dropping
[3×1×64]
Convolutional Layer 2+ *ReLU*
Max Pooling+ Dropping
Flatten Layer
Fully Connected Layer (FC)
Sigmoid Function
Normal
Under Attack
Security Status of PV-integrated PDS

Figure 5.1: Deep CNN structure for screening using image processing-like technique.

CNNs or convolutional neural networks have become popular for detecting attacks in power grids due to their robustness. They are machine learning algorithms commonly used for image recognition tasks but can be applied to detecting malicious activities in a PDS integrated with PV panels. Each row of data is transformed into a matrix and processed as an image by the CNN, enabling analysis of high-dimensional and complex data. These methods are suitable for detecting attacks that involve FDI into sensor measurements, using sensor readings of voltages and load demands as data. The CNN model extracts important features from the data through multiple layers of filters and activation functions that are trained using labeled data. The resulting model can detect the presence of an attack in real-time even in the presence of noise or uncertainty in the data.

The performance of a model is significantly influenced by the design of the CNN architecture. Numerous aspects must be considered during the process, such as the number and types of layers(e.g., convolutional, pooling, and fully connected layers), the choice of activation functions, and the selection of an optimization algorithm for training. To obtain the optimal model for the problem, it is essential to test various configurations of layers, activation functions, and optimization algorithms. This process may involve a trial-and-error approach where the model is trained and assessed multiple times using diverse configurations until the optimal model is identified.

The Sequential API of the TensorFlow library in Python is used to define a CNN model for binary classification. The model is constructed by adding layers one by one, beginning with the input layer and ending with the output layer. The process of designing the model is illustrated in Fig. 5.1, and is explained in detail step by step below.

- The layer described here plays a crucial role in the CNN model for FDI attack detection. It performs a 2D convolution operation on the input image, which enables the model to recognize and extract spatial features from the data. The layer is designed using 32 filters of size (5, 1) and applies the Rectified Linear Unit (ReLU) activation function to introduce non-linearity to the output. This function is commonly used in deep learning models as it helps to improve the performance of the network by increasing its ability to learn complex relationships between inputs and outputs. to maintain the spatial resolution of the input image and ensures that the edge pixels of the input image are used in the convolution operation, we used 'same' padding, which pads the input with zeros along the edges to ensure that the output size is the same as the input size. This technique is useful when processing images with edges or borders, as it prevents the loss of information at the edges of the image during the convolution process.

- Next layer is responsible for performing 2D max pooling on the output of the previous Conv2D layer. The max pooling technique helps to reduce the size of the feature maps and extract the most important features, while also decreasing the dimensionality of the data. In this specific implementation, a window of size 2x1 is used to perform the max pooling operation. This window size results in a reduction of the row dimension by a factor of 2, while keeping the column dimension unchanged. By down-sampling the feature maps in this way, the model is able to focus on the most important spatial information while reducing the computational complexity of the network. This can help prevent overfitting and improve the generalization ability of the model.

- A subsequent layer of Conv2D was included to enhance the model's ability to identify more intricate patterns in the input data. The new Conv2D layer was implemented with 64 filters that were each of size (3,1). By introducing this additional layer, the model gains the capacity to identify more complex relationships between the features and the target variable. The layer's output serves as an input to the subsequent layer of the model, and the features it identifies will be used to make better predictions during the model's evaluation stage. As a result, the performance of the model is likely to be improved, as the new layer allows it to better capture the complex relationships present in the data.

- In order to further reduce the dimensionality of the data and extract the most important features, we have incorporated an additional MaxPooling2D layer into the CNN architecture. This layer works by down-sampling the feature maps obtained from the previous Conv2D layer. By using a 2x1 window size, the layer reduces the row dimension by a factor of 2 and maintains the column dimension. The down-sampling process results in a smaller representation of the input image, which can be more efficiently processed in the subsequent layers. The addition of this layer to the CNN architecture can improve the model's ability to extract the most relevant features from the data, potentially leading to better performance in tasks such as classification.

- A Dropout layer was included in the model to enhance its performance during training. The layer works by randomly dropping 20% of the input units to zero at each training iteration. This method helps to improve the generalization of the model and prevents overfitting by reducing the complexity of the network. By doing so, the remaining neurons are forced to learn more robust and representative features of the data, which ultimately leads to better performance on unseen data.

- The Flatten layer was integrated into the model to transform the multi-dimensional output obtained from the previous layer into a one-dimensional vector that can be passed as input to the next Dense layer. The purpose of this step is to prepare the data in a suitable format for processing by the next layer. It's like flattening a 3D object into a 2D plane so that it can be printed on paper. The output of the Flatten layer is a long array of values that represents the flattened features of the input image. This layer is particularly useful when transitioning from a convolutional layer to a fully connected layer, as it simplifies the data by removing spatial information and reshaping it into a linear array.

- To enhance the model's ability to learn complex patterns in the data and introduce non-linearity to the output, we incorporated a Dense layer with 128 units and ReLU activation function. This layer creates a fully connected neural network, where each neuron is connected to all the neurons in the previous layer. The ReLU activation function sets negative values to 0 and keeps positive values unchanged, which helps to introduce non-linearity to the output. By using 128 units, we provide the model with sufficient capacity to learn more complex patterns in the data.

- To enhance the model's performance and reduce the risk of overfitting, we included an additional Dropout layer. During training, this layer randomly deactivates 20% of the input units, forcing the remaining neurons to learn more robust and diverse features. By doing so, the model becomes less likely to memorize the training data and better able to generalize to unseen data. This can lead to improved accuracy and reliability in real-world applications.

- To enhance the model's ability to learn complex patterns in the data, we included an additional Dense layer in the CNN architecture. This layer consists of 64 units and utilizes the ReLU activation function, which helps to introduce non-linearity to the output. The ReLU function applies the function $f(x) = max(0, x)$ to each neuron's output, ensuring that the output values are non-negative. This allows the network to learn more complex and non-linear relationships between the input and output.

- In order to enhance the model's capability of generalizing well to unseen data and avoid overfitting, we included an additional Dropout layer. During training, this layer randomly deactivates 20% of the input units, which forces the remaining neurons to learn more robust and meaningful features that are generalizable to new data. This ultimately helps the model to avoid fitting too closely to the training data and better capture the underlying patterns in the data.

- The last layer in the CNN model is a Dense layer with a single unit that uses the sigmoid activation function. This layer produces a binary probability indicating the predicted class, which means it predicts the likelihood of the input belonging to the positive class. The sigmoid activation function is used in this layer because it guarantees that the output probability falls within the range of 0 and 1. By setting the output to be a probability, we can easily interpret the model's prediction, and it becomes easier to compare the performance of the model with other classification algorithms.

In order to accurately detect attacks in the PDS, it is crucial to have a well-designed CNN architecture. The combination of Conv2D layers, MaxPooling2D layers, Dropout layers, and Dense layers with ReLU activation functions plays a vital role in the ability of the model to learn spatial features in the data, down-sample the feature maps, introduce non-linearity, and prevent overfitting. These layers work together to improve the model's performance and accuracy in detecting attacks in the PDS. The CNN model is trained using an extensive dataset of power flow data collected over a considerable period of time. The objective of the training is to enable the model to identify patterns and irregularities in the data that suggest a potential attack. Once the model is trained, it can quickly analyze new power flow data and determine if there are any deviations from expected patterns. If there are, the model can recognize the existence of an attack and send an alert, allowing for prompt action to prevent significant damage to the power grid. Therefore, the CNN model's primary purpose is to provide early warning of potential attacks and prevent severe harm to the power grid.

If the CNN architecture is not properly designed, the model may not be able to learn the necessary features from the data, leading to poor performance and incorrect attack detection. Therefore, it is important to experiment with different layer configurations, activation functions, and optimization algorithms to find the best model for the problem. This can involve a process of trial and error, with the model being trained and evaluated several times using different configurations until the best model is found. The success of the CNN architecture lies in its ability to effectively analyze the data and predict the presence of attacks in the PDS.

Chapter 6

Simulation and results

In this chapter, the proposed frameworks are demonstrated to be effective in detecting attacks in a modified IEEE 33-bus and 141-bus PDS. Additionally, an evaluation and comparison of various algorithms such as ML, Deep Learning classifiers, and clustering methods are conducted for this purpose. The upcoming two Subsections, 6.1 and 6.2, contain the simulations and outcomes for both the transient and steady-state conditions of the two proposed frameworks. Each task involves several steps that need to be followed. Figure 6.1 presents a step-by-step process for designing and implementing SVM and CNN methods to detect attacks in a PV-integrated PDS, respectively, in transient and steady state operation.

6.1 SVM framework for transient conditions

This section aims to analyze and compare the effectiveness of the proposed SVM with the classifier and clustering machine learning algorithms for detecting attacks in our modified IEEE 33-bus PDS under transient conditions. The simulations were conducted using MATLAB Simulink with a time step of 1 ms on a supercomputer with 32 cores and 500 GB of memory. To do so, the developed models in the previous sections are simulated in detail.

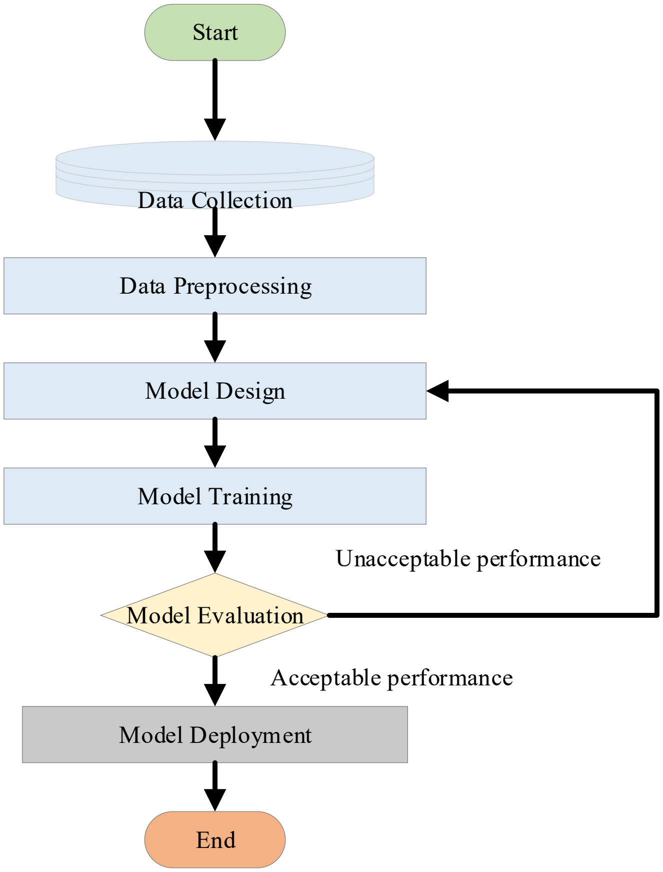


Figure 6.1: Design and implementation of a CNN method flowchart

6.1.1 Data Collection

To generate a comprehensive dataset, the simulations were executed for a total of 20 iterations where the load was progressively increased/decreased by 2.5% from zero to 50% of the nominal value. The duration of each simulation was set to 60 seconds, ensuring that the dataset covered all possible load variations. The resulting dataset was comprised of 1,200,000 rows, calculated by multiplying the number of iterations with the duration of each simulation and then dividing it by the simulation time step. Specifically, since the simulation time step was set to 0.001 seconds, multiplying it by 60 seconds and then by 20 iterations yielded 1,200,000 rows. The dataset had a total of 34 columns, with 33 columns representing bus voltages and one column indicating the percentage of load alteration.

Supervised learning methods require labeled data, which means that each data point is assigned a specific label indicating whether it belongs to a particular class or not. In this case, the period of attack occurrence is already known, and all the data during this period are considered as under-attack data with the labels set to 1. On the other hand, the remaining data during the simulation are normal and labeled as 0. The attack period is 3 seconds, which contains 3,000 samples with a time step of 1ms in the simulations. It is important to note that the simulation lasts for 60 seconds, which corresponds to 60,000 samples. Thus, only about 5% of the data can be considered as attack samples. Once the labels have been assigned to the data, the next step is to perform some preprocessing to prepare the dataset for learning.

6.1.2 Data Preprocessing

Preprocessing is important in this case because it involves transforming raw data into a format that can be easily understood and analyzed by ML algorithms. This can include tasks such as data cleaning, normalization, feature selection, and feature engineering. Proper preprocessing can help to improve the accuracy and efficiency of ML models, and can also help to avoid issues such as overfitting or underfitting. In the context of attack detection in power grids, preprocessing may involve tasks such as removing outliers, scaling data, and selecting relevant features to improve the performance of the models. Overall, prior to model training, some preprocessing steps need to be performed.

The first step in data preparation is to divide the dataset into two portions: training and testing. The training set is utilized for learning while the testing set is used to test the algorithm and calculate its performance. To achieve this, 20% of the dataset is randomly separated and reserved for testing, while the remaining 80% is utilized for training. In addition, scaling is a critical step in preparing numerical features in the dataset. To accomplish this, we use the MinMax scaler, which scales the data to a specified range between 0 and 1 while maintaining the distribution of values. Normalization of the data is also required. The process of scaling individual samples to a unit normal distribution is referred to as normalization.

The learning process can be complex and inefficient when dealing with a large number of features, as in our case with 34 features. Therefore, feature reduction algorithms are used to decrease the complexity and facilitate processing large amounts of data. One common feature reduction method is Principal Component Analysis (PCA), which transforms the source data from its original 34-dimensional form into 2-dimensional data, allowing for the visualization of the data distribution and simplifying the learning process. It is worth noting that each principal component does not represent a specific physical parameter after the reduction in dimensions. Once the feature reduction step is complete, the dataset is prepared for the learning phase.

6.1.3 Model Design

Section 5.1 provides detailed information on the design of the SVM model.

6.1.4 Model Training

The goal of the model training step is to adjust the SVM algorithm's parameters to best fit the training data and minimize errors in classifying the attack and normal data. Once the SVM model is trained, it can be used to classify new, unseen data and detect attacks in real-time operation.

To obtain the best performance for detecting attacks, certain values are taken into account to fine-tune the SVM method. The kernel used in this study is the non-linear ”Radial Basis Function (RBF)” kernel. In order to achieve both high performance and time efficiency, the values of Gamma and C are set to 0.1 and 0.01, respectively. These values have been chosen after careful consideration and experimentation to ensure that the SVM model works optimally for detecting attacks.

6.1.5 Model Evaluation

To ensure that an SVM model performs well in real-world scenarios, evaluating its performance on unseen data is a crucial step in the development process. Model evaluation techniques are used to assess how well the SVM model performs, and various common techniques are available. These techniques use the True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) values to evaluate the model's performance.

(1) Accuracy is a widely used evaluation metric for classification models that measures the proportion of correctly classified samples in the test set. It can be calculated by dividing the number of correctly classified samples by the total number of samples in the test set. For

example, if the test set contains 100 samples and the model correctly classifies 90 of them, the accuracy is 90%. Accuracy can be a useful metric when the classes in the dataset are balanced, i.e., the number of samples in each class is roughly equal. However, in imbalanced datasets, where one class has significantly more samples than the other(s), accuracy can be misleading. In such cases, other metrics such as precision, recall, and F_1-score are more informative.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

(2) Confusion matrix: A confusion matrix provides a detailed breakdown of the model's performance by showing the number of true positives, true negatives, false positives, and false negatives.

	Predicted Negative	**Predicted Positive**
Actual Negative	TN	FP
Actual Positive	FN	TP

(3) Precision and Recall: Precision measures the proportion of true positive predictions among all positive predictions, and it is calculated by dividing the number of true positives by the sum of true positives and false positives. Recall, on the other hand, measures the proportion of true positive predictions among all actual positive samples, and it is calculated by dividing the number of true positives by the sum of true positives and false negatives.

$$Precision = \frac{TP}{TP + FP} \tag{7a}$$

$$Recall = \frac{TP}{TP + FN} \tag{7b}$$

(4) F_1-score: The F_1-score is the harmonic mean of precision and recall, and it is a useful metric when both precision and recall are important.

$$F_1 score = \frac{2 \times (Precision \times Recall)}{Precision + Recall} \tag{8}$$

By evaluating the model using these metrics, we can determine how well the SVM model performs in detecting attacks in the PDS. results and comparisons are provided in 6.1.6.

6.1.6 Results

To start the evaluation of our proposed methods, the first one considered is the decision tree (DT). To calculate the performance of this algorithm, we use the testing dataset and predict labels for each sample. Then, we compare the predicted labels with the true labels to calculate the F_1 score, precision, and recall metrics. Table 6.1 shows the values for true positive, false positive, true negative, and false negative for all the algorithms, including the DT. The total number of testing samples is 240,000, which is equivalent to 20% of the total samples in our dataset as we previously discussed. Table 6.1a illustrates the performance of the supervised DT algorithm on the dataset. The DT algorithm could successfully detect 9,872 samples as under-attack data, but it missed 2,002 samples. It was able to accurately identify 218,056 samples as normal data, but it wrongly classified 10,070 samples as under-attack data. Although this technique can differentiate between most normal and under-attack data, there are still a large number of false positives, which may lead to an unacceptable number of false alarms. Additionally, there are many false negatives, which could result in undetected attacks that could potentially damage the power system. Table. 6.1b presents the results obtained from the RF method. The table indicates that the performance of the RF model is relatively better than the DT method. The RF model successfully identifies 10,510 samples as under-attack data and 225,593 samples as normal data. However, it also produces 2,537 false positive samples, which is still a high number of false alarms and not acceptable. Additionally, the RF method fails to detect 1,360 samples as being under-attack data, and wrongly considers them as normal. Table. 6.1c presents the results of the LR algorithm. Based on the results, LR exhibits the smallest number of false positive samples, which is 622. This indicates that the LR approach has fewer false alarms compared to the other methods. Additionally, the true negative (t_n) value achieved by the LR algorithm is greater than that of the other supervised techniques, indicating that the LR method performs better in identifying normal data.

The SVM algorithm, proposed in this study, is capable of identifying under-attack samples, as demonstrated in Table. 6.1d. It can be seen that this method can recognize 13,932 samples as under-attack data and 224,222 samples as normal data. This approach has a distinct feature

Table 6.1: Comparison between ML Algorithms

(a) DT

True Positive	9,872
False positive	10,070
True negative	218,056
False negative	2,002

(b) RF

True positive	10,510
False positive	2,537
True negative	225,593
False negative	1,360

(c) LR

True positive	10,607
False positive	622
True negative	227,504
False negative	1,267

(d) SVM

True positive	13,932
False positive	716
True negative	224,222
False negative	1,130

of producing a low percentage of false positive samples, which can help avoid unacceptable false alarms. Additionally, the number of false negative samples identified by this method, which is 1,130, is also lower than that produced by the other algorithms, such as DT, LR, and RF.

If both precision and recall are high, then the F_1 score will also be high, according to equation (8). Thus, the F_1 score is considered an appropriate evaluation metric to compare the algorithms discussed. As shown in Figure 6.2, the DT method has a poor performance compared to the others, with a low F_1 score of 62.04% compared to LR's 91.82%, due to the small number of evaluation metrics. On the other hand, the SVM approach performs better than the others because of its high F_1 score, precision, and recall values.

These findings indicate that the SVM method suggested in this study outperforms other approaches in accurately distinguishing between normal and under-attack data, achieving a high level of precision for detecting attacks while minimizing false alarms. The SVM algorithm demonstrated the highest F_1 score of 93.87%, indicating superior classification performance.

6.1.7 Model Deployment

Model deployment is the final step in the development of any machine learning model, where the trained model is put into production to make predictions on new, unseen data. In the case of the SVM-based attack detection method, the trained model can be deployed on a real-time system that monitors the power grid for any potential cyber attacks.

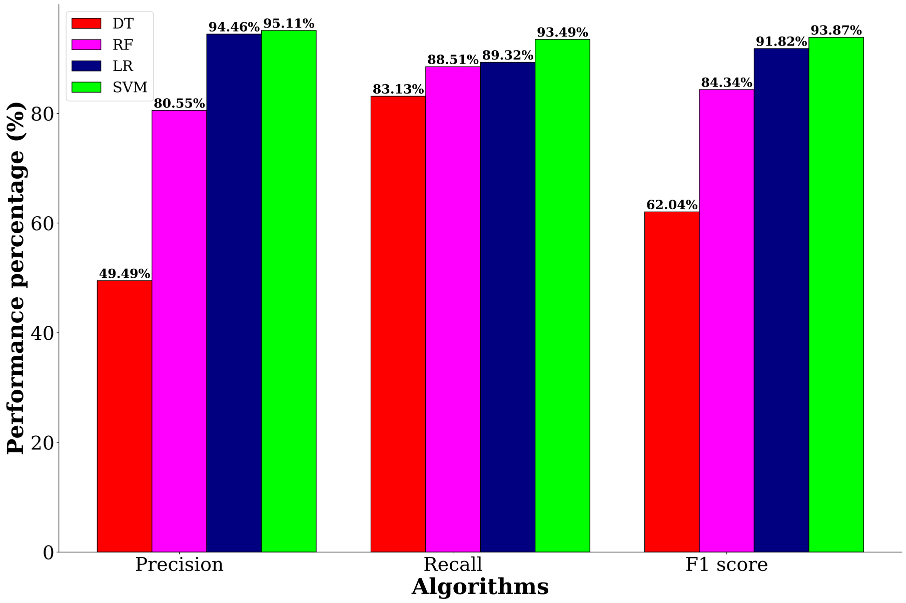


Figure 6.2: Supervised ML performance bar graph

During deployment, the SVM model is integrated with the monitoring system of the power grid and is used to classify the incoming data as normal or under attack. The real-time data is first preprocessed, then feature reduced using PCA, and finally fed into the SVM model for classification.

The output of the SVM model is then used to trigger appropriate action based on the nature of the detected attack. For example, if an attack is detected, the monitoring system may automatically isolate the affected part of the power grid to prevent further damage.

It is important to continuously monitor the performance of the deployed model to ensure that it is working as intended and providing accurate predictions. Any significant drop in performance or unexpected results should be investigated and the model re-evaluated and possibly retrained if necessary.

6.2 CNN framework for steady-state conditions

When it comes to identifying malicious activities in a PDS that is integrated with PV panels, CNNs can be applied to detect attacks by converting each row of data into a matrix that is processed as an image. These methods are effective in analyzing complex and high-dimensional data, making them well-suited for identifying attacks in power systems where false data is injected into sensor measurements. The data used for attack detection in a typical power grid setting includes sensor readings of voltages and load demands. CNNs can extract important features from this data and use them to make accurate predictions regarding the presence of an attack. This is accomplished using several layers of filters and activation functions that are trained on a large volume of labeled data. The result is a model that can detect attacks in real-time, even in situations where there is significant noise or uncertainty in the data.

A vast collection of power flow data from the PDS is used to train the CNN model, which learns to detect patterns and anomalies in the data that signify an attack. The model is then able to swiftly recognize any deviations from the anticipated patterns and determine if an attack is occurring when fresh data is entered into the model. As a result, the model aids in detecting possible attacks at an early stage and avoids significant damage to the power system.

For this part of the project, a personal computer equipped with an Intel i7 core processor with a clock speed of 2.9 GHz, 16 GB of RAM, and running the Windows operating system is used. To run deep learning tasks, I utilized Google Colab, which is a platform that uses Python 3.8. I also used MATLAB R2020 b to collect power flow data.

The procedure for creating and deploying CNN techniques for attack detection in a PDS that is integrated with PV panels can be summarized as follows.

6.2.1 Data Collection

In designing and implementing a CNN for attack detection in a PV-integrated PDS, one crucial step is data collection. This step entails gathering relevant data from the PDS that can be utilized to train and test the CNN model. It is vital to collect data that accurately represents the typical operating conditions of the system and any possible malicious activities that could occur. The importance of data collection lies in the fact that it provides input to the CNN model, which is employed to make predictions about the presence of attacks in the PDS. If the data collected is incorrect or incomplete, it can cause bias in the model and lead to poor performance. Furthermore, a large and diverse dataset can lead to a more robust and generalizable model that can detect attacks more accurately in various scenarios. Hence, the quality of the data collected is a critical factor in the success of the attack detection framework based on CNNs.

To create a comprehensive dataset for developing a deep learning model, a power flow analysis was conducted using the MATPOWER tool in MATLAB. The analysis was carried out at 15-minute intervals for two years to balance the completeness of the dataset with its size. During each interval, VR was used to maintain voltage within normal limits. The voltage regulation system is shown in Algorithm 2. However, PV and inverter component limitations may sometimes prevent complete voltage compensation and regulation. The dataset was created by extracting relevant information such as bus voltages, load demand levels, time-related variables like seasons, weekdays, and weekends, as well as data labels from the power flow analysis.

6.2.2 Data Preprocessing

The preprocessing stage is a crucial step that sets the groundwork for the entire process. It is crucial to clean, structure, and convert the data into a suitable format for the CNN model to use. The goal of this step is to convert the raw data into a structured format that can be used for training the model. The data preprocessing step typically involves cleaning the data, dealing with missing values, normalizing the data, scaling, and transforming the data into a compatible format. These tasks are vital as they help to reduce possible errors and biases in the data, and they make the data more suitable for the CNN model to use.

To prepare our dataset, the first step is to check for any missing data. Fortunately, this is not an issue as our data source is power flow data in MATLAB. Next, the data is processed and cleaned to remove any incorrect, corrupted, duplicated, incorrectly formatted, or incomplete data. To simplify the analysis, the data is then scaled, which standardizes the values across the dataset, making the model learning and problem understanding more efficient. Standardization of features involves transforming the data by removing the mean and scaling it to unit variance. This is accomplished by calculating standard scores, which are the difference between a sample (x) and the mean of the training samples (u), divided by the standard deviation of the training samples (s). Mathematically, this is represented as:

$$z = \frac{(x - u)}{s} \tag{9}$$

The aim of standardizing the data is to maintain a mean of zero and unit variance, which leads to a more reliable comparison of feature values across the dataset. Another technique used to enhance the data is normalization, which involves adjusting the numeric column values to a common scale while maintaining their differences in value ranges. In this particular study, the normalization method used is Max normalization, which is based on the infinity norm of the data. This mode was chosen because it is believed that data in 2-dimensional space can be more clearly distinguished through the application of Max normalization.

To make the data suitable for a CNN approach, it is important to transform each data row into a matrix format. This can be done by converting each row of data, which contains 36 characteristics such as 33 bus voltages, load demand levels, season, and weekdays/weekends, into a 12×3 matrix.

In the last step of preparing the dataset for the CNN method, it is necessary to divide it into two separate sets, i.e., the training set and the testing set. This division is critical to evaluating the model's performance accurately. The training set, which constitutes 70% of the data, is used to train the model, i.e., to teach it how to identify and classify different patterns and anomalies in the data that are indicative of an attack. The remaining 30% of the data make up the testing set, which is used to validate the performance of the model. During the testing phase, the model is presented with new, unseen data that it has not been trained on. The accuracy of the model's predictions on the testing data is a measure of its ability to generalize to new, previously unseen scenarios. The 70:30 ratio is a common split used in machine learning applications to ensure that enough data is available for training while still providing sufficient data for testing and evaluation. However, depending on the size and complexity of the dataset, other ratios may also be used to ensure optimal performance of the model.

6.2.3 Model Design

Section 5.2 of the document contains a detailed description of the design of the model. It includes information such as the architecture of the model, the number of layers, and the activation functions used to train the model. Additionally, it provides insights into the rationale behind the choice of these design elements and how they contribute to the performance of the model.

6.2.4 Model Training

Training the CNN model is a crucial stage in the design and implementation process. It involves using an optimization algorithm to train the model on the preprocessed data. The aim is to minimize the discrepancy between the predicted outputs and the actual outputs, resulting in the most precise model possible. This process enables the model to learn from the training data and generalize its predictions to unseen data. By iteratively adjusting the model's parameters, the optimization algorithm ensures that the model accurately captures the underlying patterns in the data, allowing it to make reliable predictions. The success of the model's performance relies heavily on the effectiveness of the optimization algorithm used during the training phase.

In the process of model training, the goal is to develop a CNN model that can establish correlations between the inputs, such as voltage levels, active and reactive power levels, and load demand, and the outputs, which is the identification of potential attacks, using the data obtained during training. This trained model can subsequently be used to detect attacks in new, unseen data, allowing for the identification of malicious activities in real-world situations.

It is crucial to recognize that the process of Model Training is an iterative one, and it may require multiple attempts with various hyperparameters or architectures to find the most suitable model for the task at hand. Model Training is a computationally intensive process that necessitates the use of powerful GPUs and can take a significant amount of time, ranging from several hours to multiple days, depending on the size of the data and the intricacy of the model.

Our CNN model uses the Adam optimization algorithm, which is a popular method for updating the network weights during iterative gradient descent with training data [37]. Typically, the Adam optimization algorithm is used in deep learning in conjunction with the binary cross-entropy loss function. It is an adaptive learning rate optimization algorithm that tracks the first and second moments of the parameters' gradients and adjusts the learning rate based on the estimated variance and mean of the gradients. By doing this, the algorithm can dynamically adjust the learning rate during training, helping to avoid getting stuck in local minima and speeding up convergence.

In order to utilize the Adam optimization algorithm along with the binary cross-entropy loss, we first need to compute the gradient of the loss function concerning the parameters of the neural network. Then, based on the estimated mean and variance of the gradients, the Adam algorithm updates these parameters. This technique effectively reduces the oscillations that may occur in weight updates and facilitates faster convergence.

The Binary cross-entropy loss function is suitable for problems that involve binary classification and it is a good fit with Adam optimization due to its differentiability and clear gradient that can be utilized to update the weights of the neural network. Algorithm 3 provides a detailed explanation of the Adam algorithm. In our optimization process, we utilized the binary cross-entropy loss function [26], which is defined as follows:

$$\begin{aligned} L &= -\sum_{i=1}^{2} t_i \log(p_i) \\ &= -[t_1 \log(p_1) + t_2 \log(p_2)] \\ &= -[t \log(p) + (1-t) \log(1-p)] \end{aligned} \tag{10}$$

This particular equation computes the loss L by adding up the values for the two classes ($i = 1$ and $i = 2$), and taking the negative of the product of the actual label t_i and the logarithm of the predicted probability p_i for each class. The expression for calculating loss can be made simpler by combining terms. We can express the loss in terms of a single class by defining t_1 as t and t_2 as $1 - t$, and p_1 as p and p_2 as $1 - p$.

$$L = \begin{cases} -\log(p) & \text{if} \quad t = 1 \\ -\log(1-p) & \text{if} \quad t = 0 \end{cases} \tag{11}$$

In order to optimize the model and improve its performance, a single expression for calculating the cross-entropy loss is obtained by simplifying the initial formula. This expression takes into account the true label and the predicted probability, and is used during the training process to adjust the model parameters and minimize the loss. The Model Training step is essential in developing a CNN model for PDS attack detection, as it allows the model to learn from the input data and make accurate predictions. A well-trained model can contribute to enhanced security and reliability of the PDS.

```
Algorithm 3: Adam Optimization Algorithm
Input: Input data $X$, True output labels $y$, Initial parameter values $\theta$, Learning rate $\alpha$,
       Exponential decay rates $\beta_1$, $\beta_2$, Small constant $\epsilon$, Maximum number of iterations
       $max\_iterations$
Output: Learned model parameters $\theta$
Initialize the first and second moment variables $m = 0$, $v = 0$, $t = 0$;
while $t < max\_iterations$ and not converged do
    $t = t + 1$;
    Compute the gradient of the binary cross-entropy loss function: $z = X \cdot \theta$,
      $y_{pred} = sigmoid(z)$, $g = \frac{X^T(y_{pred} - y)}{y.shape[0]}$;
    Update the first and second moment estimates: $m = \beta_1 \cdot m + (1 - \beta_1) \cdot g$,
      $v = \beta_2 \cdot v + (1 - \beta_2) \cdot g^2$;
    Compute bias-corrected estimates of the moments: $\hat{m} = \frac{m}{(1-\beta_1^t)}$, $\hat{v} = \frac{v}{(1-\beta_2^t)}$;
    Update the model parameters using the Adam update rule: $\theta = \theta - \alpha \cdot \frac{\hat{m}}{(\sqrt{\hat{v}}+\epsilon)}$;
end
return $\theta$;
```

6.2.5 Model Evaluation

The model evaluation step for a CNN based method involves assessing the performance of the trained model on a separate set of data that was not used for training, called the testing set. This step is important to verify that the model can accurately classify new data, rather than just memorizing the data it was trained on.

The evaluation of the CNN-based method involves the use of various metrics, as described in Subsection 6.1.5. These metrics include accuracy, precision, recall, and F_1-score. These metrics are essential in assessing the performance of the method in detecting attacks in the PDS.

Apart from the metrics mentioned above, the evaluation of the CNN-based method also involves feature analysis. This analysis is used to determine the most relevant features in the dataset that contribute to the detection of attacks in the PDS. By identifying these relevant features, the model can be optimized to improve its accuracy in detecting attacks. The feature analysis involves the use of various techniques such as principal component analysis (PCA), t-distributed stochastic neighbor embedding (t-SNE), and correlation analysis. These techniques help in identifying the most important features in the dataset and their correlation with the target variable. Overall, the use of these metrics and feature analysis is crucial in evaluating the performance of the CNN-based method and improving its accuracy in detecting attacks in the PDS.

6.2.6 Results

The CNN model takes in 33 and 141 voltage measurements from the historical data along with load demand, PV power generation, and temporal information like weekdays/weekends, and season as inputs. To fit into the CNN model, these inputs are reshaped into an image format. In addition, we assess the classification performance of the CNN model against several standard machine learning algorithms, including RF, KNN, LR, SVM, and Multilayer Perceptron (MLP). We used the default settings provided by the scikit-learn and keras libraries for all models since a comprehensive explanation of these models is beyond the scope of this paper.

Feature Importance Analysis

Figures 6.3 and 6.4 demonstrate the results of a feature importance analysis conducted on the datasets of IEEE 33-bus and 141-bus systems, respectively. The datasets for both systems include various features such as voltage measurements, load condition, PV generation, weekdays/weekend, season, and label. For instance, the dataset for the IEEE 33-bus system has a total of 36 features, while the dataset for the 141-bus system has 144 features. However, the first column, which has a constant value of 1 p.u., was excluded, and the last column was used as the target. The feature importance analysis was conducted to identify the most significant features that influence the performance of the CNN model.

The results of the feature importance analysis for the datasets of IEEE 33-bus and 141-bus systems are presented in Fig.6.3 and Fig.6.4, respectively. According to Fig. 6.3, the most significant feature for detecting FDI attacks in the IEEE 33-bus dataset is the load condition. The PV generation and time-related features, such as weekdays/weekend and season, are less important in identifying FDI attacks.

Fig. 6.4 reveals that bus number 52, which is connected to PV and voltage regulation systems and is under attack, is the most critical feature in detecting FDI attacks in the 141-bus system dataset. In contrast, the time-related features are found to be less significant.

The analysis of feature importance plays a crucial role in comprehending the characteristics

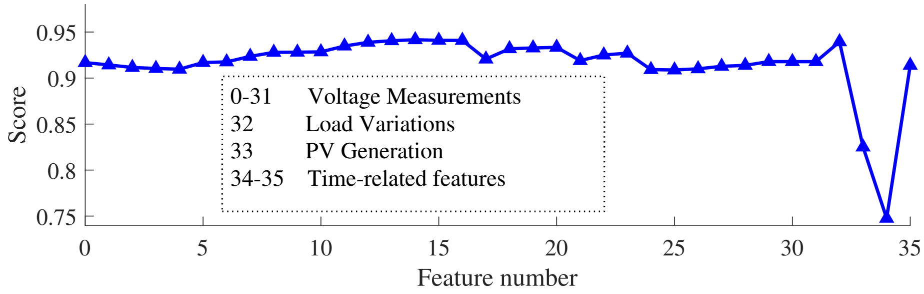


Figure 6.3: Importance of each feature in the dataset for IEEE 33-bus

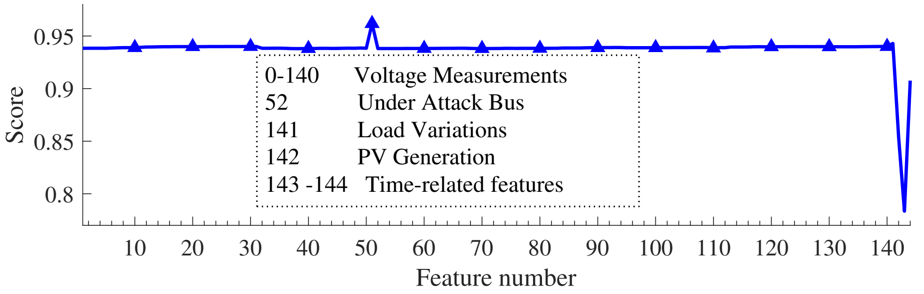


Figure 6.4: Importance of each feature in the dataset for 141-bus

of the dataset and recognizing the features that make the most significant contribution to the performance of the model. By identifying these crucial features, the model can concentrate on them, enhance its performance, and simultaneously reduce the complexity and computational time.

The relationship between features is a crucial factor in examining a dataset. Fig.6.5 and Fig.6.6 display a feature correlation map for the IEEE 33-bus and 141-bus systems, respectively. The colormap uses a scale from 1 to -1, where 1 represents complete correlation between two features, and -1 indicates a complete inverse correlation. In Fig.6.5, it is apparent that most voltage measurements have a strong correlation with each other. However, feature 32, the load condition, has a negative correlation because increased load leads to decreased voltage. Furthermore, the PV generation data and time-related features have a weak correlation with voltage measurements. The light red cells on the colormap represent buses that are under attack and nearby buses with lower voltage due to the attack. In Fig.6.6, a similar correlation pattern is visible, but feature 51, the voltage of bus 52, is lighter red, indicating that its voltage changes significantly more than other buses due to the attack.

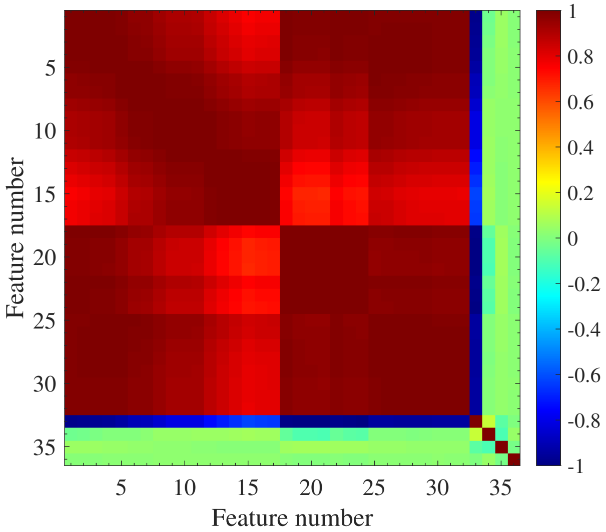


Figure 6.5: features correlation colormap for IEEE 33-bus

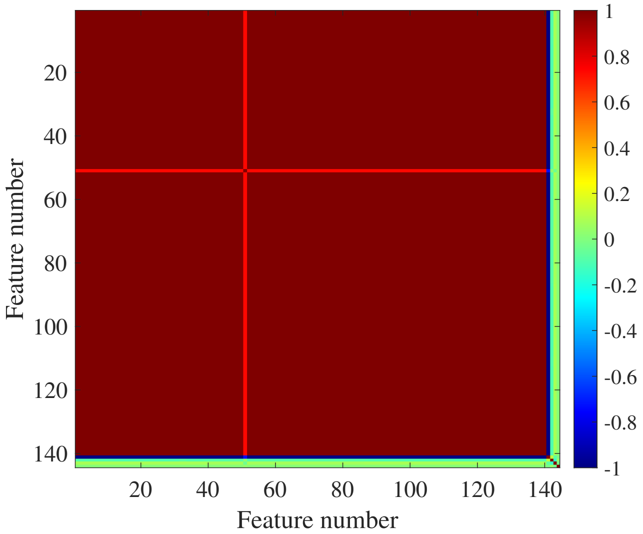
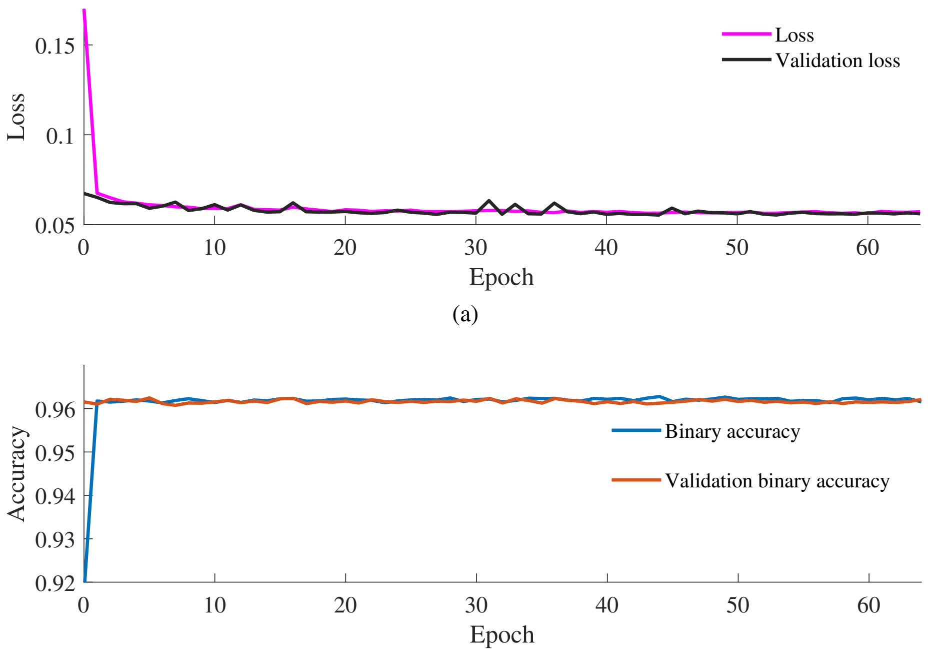


Figure 6.6: features correlation colormap for 141-bus

Performance of Customized CNN Method and its Scalability

The training loss and accuracy for the IEEE 33-bus and 141-bus systems are shown in Fig. 6.7 and Fig. 6.8, respectively. In Fig. 6.7, it is apparent that the cross-entropy loss reduces significantly in the initial stages of training, signifying that the CNN model is learning and becoming more adept at detecting FDI attacks in IEEE 33-bus PDS. As the number of epochs increases, the decrease in



(a)

(b)

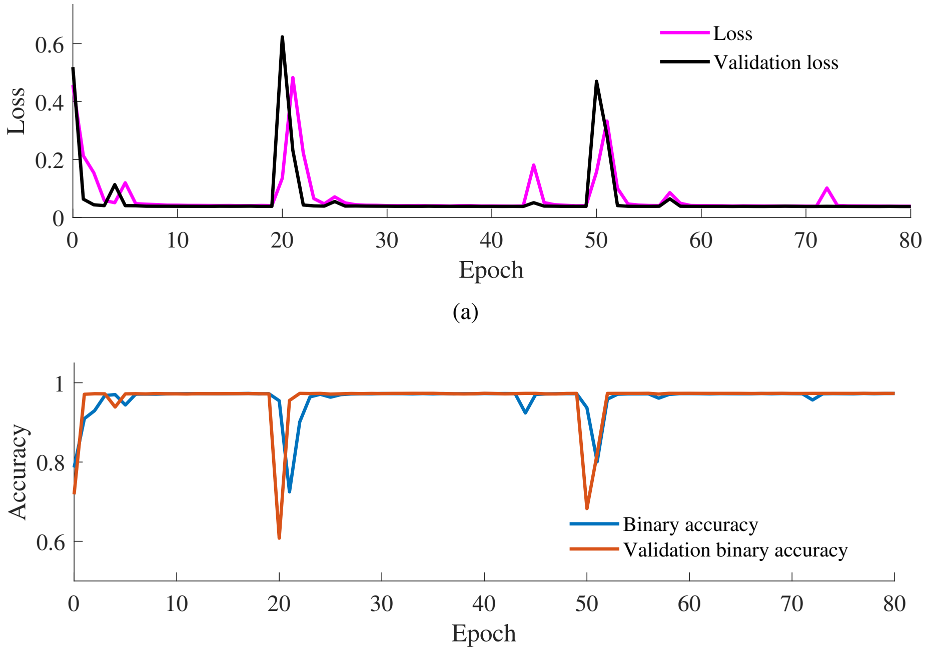
Figure 6.7: Training loss and accuracy data for IEEE 33-bus

loss becomes more gradual, which is expected as the model approaches convergence. Meanwhile, the binary accuracy, which is the proportion of correctly classified instances, improves steadily. This suggests that the model is becoming better at distinguishing between normal and attack instances.

Fig. 6.8 illustrates that during the training process for the 141-bus system, there are some fluctuations in both the loss and accuracy. These fluctuations are caused by the stochastic nature of the training algorithm and are to be expected during training. As the training continues, these fluctuations decrease in magnitude, and the model becomes more stable, ultimately reaching convergence.

The results of our proposed CNN method for IEEE 33-bus PDS are compared with five different ML models in Table 6.2. The evaluation of the models is based on the metrics discussed in Subsection 6.1.5, which include accuracy, precision, recall, and F_1-score.

According to the data presented in Table 6.2, our proposed CNN model performs better than the other five ML models in terms of accuracy, precision, recall, and F_1-score. The CNN model has an accuracy rate of 96.24%, which is the highest compared to the second-best model, LR, which has an accuracy rate of 94.50



(a)

(b)

Figure 6.8: Training loss and accuracy data for 141-bus

The evaluation results in Table 6.2 indicate that the proposed CNN model outperforms all other ML models regarding precision, recall, and F_1-score in addition to accuracy. The CNN model obtained 95.71% precision, 96.81% recall, and 96.26% F_1-score, which are the highest achieved by any model. These findings demonstrate CNN's superior effectiveness in detecting attacks in the IEEE 33-bus PDS. Despite the superior performance of the proposed CNN model, some of the other models also demonstrated high accuracy. For instance, LR attained an accuracy of 94.50%, while SVM recorded a comparable accuracy of 94.61%. MLP also performed well, with an accuracy of 94.75% and the highest recall among all models at 97.09%. Although KNN and RF performed slightly worse than the other models, they still recorded relatively high accuracies of 92.33% and 91.06%, respectively.

The results presented in Table 6.3 demonstrate the effectiveness and scalability of our proposed CNN approach for identifying attacks in the 141-bus PDS. Our CNN model outperformed all other models with an F_1-score of 97.36%, which was significantly higher than those achieved by LR, SVM, and MLP at 95.53%, 96.92%, and 96.82%, respectively. In contrast, KNN and RF achieved even lower F_1-scores of 93.90% and 95.30%, respectively, which were more than 1% lower than

Metric Model	Accuracy	Precision	Recall	F_1-Score
RF	91.06	91.08	91.02	91.05
KNN	92.33	92.35	92.30	92.33
LR	94.50	95.26	93.67	94.46
SVM	94.61	94.99	94.19	94.59
MLP	94.75	92.75	97.09	94.87
Proposed CNN	96.24	95.71	96.81	96.26

Table 6.2: Detection results for IEEE 33-bus PDS

Metric Model	Accuracy	Precision	Recall	F_1-Score
RF	95.30	95.28	95.31	95.30
KNN	93.90	93.94	93.86	93.90
LR	95.71	99.70	91.69	95.53
SVM	97.01	1.0	94.03	96.92
MLP	96.92	1.0	93.85	96.82
Proposed CNN	97.31	95.41	99.40	97.36

Table 6.3: Detection results for 141-bus PDS

the CNN's F_1-score. Furthermore, our CNN model showed high precision and recall, indicating that it could accurately identify attacks while minimizing false alarms.

In another scenario, the voltage regulation system of certain buses equipped with PV systems receives feedback from another bus, which is not the PCC, and adjusts the voltage of that bus instead of regulating its own bus. For example, in the IEEE 33-bus PDS, the PV system installed in Bus 7 and 17 regulates the voltage of Bus 18, which has the lowest voltage among all the buses, while Bus 30 regulates the voltage of Bus 33. The PV system must also ensure that its own voltage remains within the specified range. This regulation strategy can address voltage instability issues in PDS and maintain the voltage within an acceptable range.

In this scenario, an attacker intercepts the communication, receives the voltage measurement of Bus 18, and maliciously alters it according to Equation (4). Based on the results presented in Table 6.4, the CNN-based attack detection method performs better than the alternative scenario, where the PVs attempt to regulate the PCC voltage. This is because Bus 18 has a higher probability of exceeding the acceptable voltage range due to its comparatively lower voltage than the other buses in the system.

Table 6.4: Detection results in IEEE 33-bus PDS when the PV regulates the voltage of other buses

Metric / Model	Accuracy	Precision	Recall	F1 Score
RF	92.06	92.46	92.31	92.38
KNN	94.83	94.87	94.77	94.81
LR	96.33	97.15	95.90	96.52
SVM	96.94	97.35	96.82	97.08
MLP	97.84	97.46	98.15	97.80
Proposed CNN	99.91	99.92	99.90	99.91

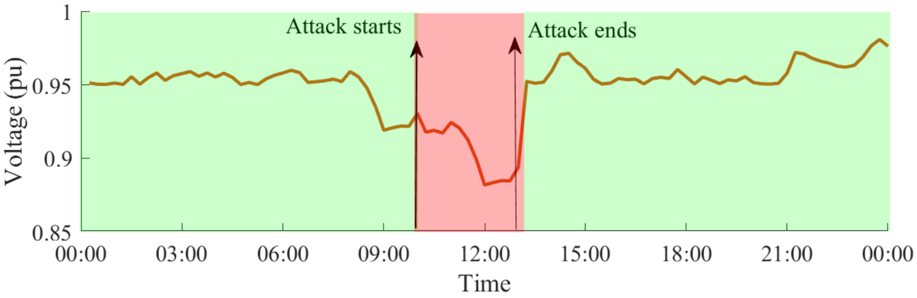


Figure 6.9: Results of the proposed method for attack detection during a day

The scalability of proposed CNN approach is demonstrated by the results of the 141-bus PDS test, where it achieved a high detection accuracy on a larger and more intricate power system than the IEEE 33-bus PDS. This is significant because real-world power systems are typically more extensive and complicated than the research test systems. The 141-bus system has more buses and branches than the IEEE 33-bus system, providing more data for training and testing ML models. The additional data enables ML algorithms to learn more intricate patterns and correlations between features, resulting in easier detection of anomalies and attacks.

Performance of the Proposed CNN Method during Online Detection

Figure 6.9 illustrates the results of the proposed method in the time domain. The red area represents an attack, while the green area indicates normal system operation. The model is capable of detecting attacks, but since the controller's performance is not taken into account during training, the CNN also identifies the controller's response to attacks as problematic. Nevertheless, the method is trained on both normal and attacked data, making it effective in identifying attacks.

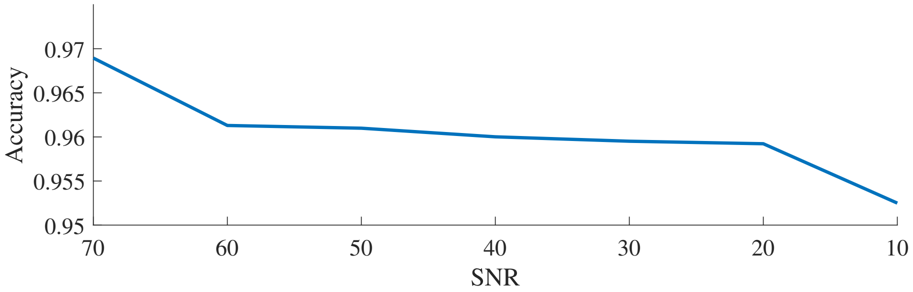


Figure 6.10: Noise robustness of the proposed method for IEEE 33-bus

Verifying the Robustness of the Proposed Method against Noise

In PDS, data transmission can be susceptible to noise, making it difficult to accurately detect attacks. To overcome this challenge, it is essential to employ a detection method that is resilient to noise. The newly developed detection method ensures reliable detection of attacks in real-world scenarios, even in the presence of noise. The accuracy of the detection method when the signal-to-noise ratio (SNR) is varied from 10 to 70 is depicted in Fig. 6.10. The results demonstrate that the accuracy remains relatively consistent despite the presence of noise. However, when the SNR is reduced, a slight decrease in accuracy may occur.

The CNN's excellent performance is due to its capacity to learn relevant features from raw input data, which in this case are the measurements of the power system. The CNN uses convolutional layers to recognize spatial correlations within the input data and extract significant features. The fully connected layers then utilize these features to make the final classification decision. The CNN also has the advantage of being able to handle input data with high dimensions, a feature that is often needed in real-world applications. Its ability to process large volumes of data makes it a suitable choice for complex tasks such as detecting attacks in power systems. Moreover, the CNN method is very adaptable and can be tailored to meet the specific demands of the application. Various designs, hyperparameters, and optimization algorithms can be utilized to fine-tune the CNN's performance for a specific task. The CNN has several practical advantages in addition to its high performance and flexibility. Unlike some other models, the CNN does not need feature engineering, which can be a complex and time-consuming task. Furthermore, the interpretability of the CNN can be enhanced by using visualization techniques such as feature visualization and activation maps, which can provide insights into the decision-making process of the CNN.

In summary, the outcomes of our research show that the CNN method is an effective and expandable technique to detect attacks in power systems. The CNN's capability to learn significant features from the raw data, along with its accuracy and scalability, demonstrates its potential for practical use in power system security.

6.2.7 Model Deployment

Model deployment is the process of integrating a trained model into a production environment for use in real-world scenarios. In the case of attack detection in power systems using a CNN, model deployment would involve implementing the trained CNN model in the power system infrastructure to monitor the system for attacks.

To deploy the CNN model, several steps need to be taken. First, the CNN model needs to be converted into a format that can be easily deployed in the production environment. This could involve converting the model into a framework such as TensorFlow, PyTorch, or Keras.

Next, the model needs to be integrated into the power system infrastructure. This involves creating an interface between the model and the data sources, such as sensors or monitoring equipment, to receive input data for analysis.

Once the interface is established, the model can be deployed to monitor the power system in real-time. The output of the model can then be used to trigger alerts or actions in response to an attack detected by the model.

It is important to note that model deployment is not a one-time process, as the model needs to be continually updated and retrained to adapt to changes in the power system and to improve its performance. Thus, a continuous monitoring and updating process needs to be established to ensure that the deployed model remains accurate and effective over time.

Chapter 7

Conclsuion

This thesis suggests that FDI attacks can affect the voltage regulation schemes in PV-integrated PDSs, leading to the protection system's suboptimal performance in serving the loads. To address this issue, the first solution proposed a ML-based framework based on SVM for detecting FDI attacks in transient conditions, with SVM delivering better performance than other methods. The second solution proposed a data-driven framework based on a CNN model, which outperformed other techniques in detecting FDI attacks in steady-state conditions and demonstrated scalability in a larger PDS and noise robustness in presence of noise. Both frameworks provide effective protection against cyber threats and ensure the secure and reliable operation of power systems.

The proposed ML-based and data-driven frameworks have demonstrated their effectiveness in detecting FDI attacks in PV-integrated PDSs, and they can be potentially deployed in real-world systems to enhance the security of power systems. However, there are still some challenges that need to be addressed. For instance, more complex attacks like replay attacks can be considered to evaluate the performance of the proposed frameworks. Moreover, future work can focus on designing a localization system to determine the location of the attack, which can facilitate the system's response to the attack. Additionally, other deep learning models like transfer learning can be applied to handle non-comprehensive datasets, which can further improve the accuracy of the proposed frameworks. Therefore, the proposed frameworks provide a solid foundation for future research to enhance the security of PV-integrated PDSs.

Appendix A

Supervised and Unsupervised ML methods

A.1 Isolation Forest

Isolation Forest is a machine learning algorithm used for anomaly detection, which means detecting data points that are significantly different from the majority of the dataset.

The algorithm works by randomly selecting a feature and a split point between the maximum and minimum values of that feature. It then creates a binary tree structure by recursively repeating this process on each of the two partitions generated by the split. The process stops when a predefined stop criterion is met, such as when the tree has reached a maximum depth or the number of data points in a partition falls below a threshold.

The intuition behind this algorithm is that anomalies are typically few and far between and therefore require fewer splits to be isolated from the majority of the data. On the other hand, normal points require more splits to be isolated. This means that anomalies will have a shorter average path length in the tree than normal points. The algorithm uses the average path length of each data point in the tree to measure its abnormality. Points with shorter path lengths are considered anomalies.

To determine the anomaly score of a data point, the algorithm constructs an ensemble of such trees and averages their path lengths. The anomaly score is then computed as the exponential function of the negative average path length. The algorithm can be tuned by adjusting the number of trees in the ensemble and the maximum depth of each tree.

Isolation Forest has several advantages over other anomaly detection algorithms. It is scalable, meaning it can handle large datasets efficiently. It is also robust to outliers and can detect anomalies in high-dimensional datasets.

A.2 Logistic Regression

Logistic Regression is a machine learning algorithm used for binary classification, which means separating data points into two classes based on a set of input features. The goal of logistic regression is to find a linear decision boundary that separates the two classes as best as possible.

The algorithm works by modeling the probability of a data point belonging to the positive class (class 1) as a function of its input features. This function is usually expressed using the logistic or sigmoid function, which maps any real-valued input to a value between 0 and 1:

$P(y = 1|x) = \frac{1}{1+e^{-z}}$

where z is a linear combination of the input features:

$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n$

where $x_1, x_2, ..., x_n$ are the input features, $\beta_0, \beta_1, ..., \beta_n$ are the model parameters (also known as coefficients or weights), and y is the target variable, which takes on a value of 1 for the positive class and 0 for the negative class.

The logistic function transforms the linear combination of input features into a probability value between 0 and 1. If the probability value is greater than or equal to a certain threshold (usually 0.5), the data point is classified as belonging to the positive class; otherwise, it is classified as belonging to the negative class.

The model parameters are learned by minimizing a cost function, such as the cross-entropy loss function, using an optimization algorithm, such as gradient descent. The cost function measures the difference between the predicted probabilities and the true class labels of the training data. The optimization algorithm updates the model parameters iteratively to minimize the cost function.

Logistic Regression has several advantages over other classification algorithms. It is simple and interpretable, meaning it can provide insights into which features are important for predicting the target variable. It also works well with both numerical and categorical input features and can handle noisy data.

A.3 Decision Tree

Decision Tree is another machine learning algorithm used for classification, which means separating data points into different classes based on a set of input features. The algorithm creates a tree-like model of decisions and their possible consequences to predict the target variable.

The algorithm works by recursively partitioning the input feature space into smaller regions or nodes, where each node corresponds to a subset of the data points. The partition is performed based on the value of one of the input features at a time, and the partitioning feature and value are chosen such that they minimize the impurity of the resulting subsets.

The impurity of a node can be measured by different metrics, such as entropy, Gini impurity, or misclassification rate. These metrics measure how much the classes are mixed within a node. A node is considered pure if all the data points in the node belong to the same class.

The algorithm continues to recursively partition the input space until a stopping criterion is met, such as reaching a maximum depth, having a minimum number of data points in a node, or having a pure node. At each node, the algorithm selects the partitioning feature and value that minimize the impurity of the resulting subsets.

Once the tree is constructed, new data points can be classified by following the decision path from the root node to a leaf node based on their input feature values. The class label of the leaf node is then assigned to the data point as the predicted class.

Decision Tree has several advantages over other classification algorithms. It is simple and interpretable, meaning it can provide insights into which features are important for predicting the target variable. It can also handle both numerical and categorical input features and can capture complex nonlinear relationships between the input features and the target variable. However, Decision Tree can suffer from overfitting, where the model captures noise in the training data and performs poorly on new unseen data. To address this, various regularization techniques, such as pruning or ensemble methods like Random Forest, can be used.

A.4 Random Forest

Random Forest is a machine learning algorithm used for classification and regression tasks, which is an ensemble method of decision trees. The algorithm creates multiple decision trees and combines their predictions to make a final prediction.

The algorithm works by building a set of decision trees using a random subset of the training data and a random subset of the input features at each split. This randomness helps to reduce the correlation between the trees and improve the generalization ability of the model.

The number of trees in the ensemble is a hyperparameter that can be tuned based on the performance on a validation set. The decision trees can be constructed using any of the decision tree algorithms, such as CART or ID3.

To make a prediction for a new data point, each decision tree in the ensemble independently predicts the class label, and the final prediction is obtained by combining the predictions of all the trees. The most common way to combine the predictions is to use the majority vote. That is, the class label with the most votes across all the trees is assigned as the final prediction.

Random Forest has several advantages over other classification algorithms. It is less prone to overfitting than a single decision tree, as the randomness reduces the correlation between the trees and improves the generalization ability of the model. It can handle both numerical and categorical input features and can capture complex nonlinear relationships between the input features and the target variable. It is also computationally efficient and can handle large datasets.

Bibliography

[1] Data requests. URL https://www.aeso.ca/market/market-and-system-reporting/data-requests/.

[2] Ansi c84.1 electric power systems and equipment - voltage ranges. URL http://www.powerqualityworld.com/2011/04/ansi-c84-1-voltage-ratings-60-hertz.html.

[3] A. Abdallah and X. S. Shen. Efficient prevention technique for false data injection attack in smart grid. In *2016 IEEE International Conference on Communications (ICC)*, pages 1–6, 2016. doi: 10.1109/ICC.2016.7510610.

[4] M. Ahmadzadeh, A. Abazari, and M. Ghafouri. Detection of fdi attacks on voltage regulation of pv-integrated distribution grids using machine learning methods. In *2022 IEEE Electrical Power and Energy Conference (EPEC)*, pages 73–78, 2022. doi: 10.1109/EPEC56903.2022.10000189.

[5] F. Aloul, A. Al-Ali, R. Al-Dalky, M. Al-Mardini, and W. El-Hajj. Smart grid security: Threats, vulnerabilities and solutions. *International Journal of Smart Grid and Clean Energy*, 1(1):1–6, 2012.

[6] M. E. Andreoni López, F. J. Galdeano Mantiñan, and M. G. Molina. Implementation of wireless remote monitoring and control of solar photovoltaic (pv) system. In *2012 Sixth IEEE/PES Transmission and Distribution: Latin America Conference and Exposition (TD-LA)*, pages 1–6, 2012. doi: 10.1109/TDC-LA.2012.6319050.

[7] A. Anwar, A. N. Mahmood, and Z. Tari. Ensuring data integrity of opf module and energy database by detecting changes in power flow patterns in smart grids. *IEEE Transactions on Industrial Informatics*, 13(6):3299–3311, 2017. doi: 10.1109/TII.2017.2740324.

[8] S. Aoufi, A. Derhab, and M. Guerroumi. Survey of false data injection in smart power grid: Attacks, countermeasures and challenges. *Journal of Information Security and Applications*, 54:102518, 2020.

[9] M. Ashrafuzzaman, S. Das, Y. Chakhchoukh, S. Shiva, and F. T. Sheldon. Detecting stealthy false data injection attacks in the smart grid using ensemble-based machine learning. *Computers & Security*, 97:101994, 2020.

[10] S. Bhattacharjee and S. K. Das. Detection and forensics against stealthy data falsification in smart metering infrastructure. *IEEE Transactions on Dependable and Secure Computing*, 18 (1):356–371, 2021. doi: 10.1109/TDSC.2018.2889729.

[11] N. Bhusal, M. Gautam, and M. Benidris. Detection of cyber attacks on voltage regulation in distribution systems using machine learning. *IEEE Access*, 9:40402–40416, 2021. doi: 10.1109/ACCESS.2021.3064689.

[12] P. Chaudhary and M. Rizwan. Voltage regulation mitigation techniques in distribution system with high pv penetration: A review. *Renewable and Sustainable Energy Reviews*, 82:3279–3287, 2018.

[13] Z. Chu, O. Kosut, and L. Sankar. Detecting load redistribution attacks via support vector models. *IET Smart Grid*, 3(5):551–560, 2020.

[14] M. A. G. de Brito, L. Galotto, L. P. Sampaio, G. d. A. e Melo, and C. A. Canesin. Evaluation of the main mppt techniques for photovoltaic applications. *IEEE Transactions on Industrial Electronics*, 60(3):1156–1167, 2013. doi: 10.1109/TIE.2012.2198036.

[15] R. Deng, P. Zhuang, and H. Liang. False data injection attacks against state estimation in power distribution systems. *IEEE Transactions on Smart Grid*, 10(3):2871–2881, 2019. doi: 10.1109/TSG.2018.2813280.

[16] S. H. Dolatabadi, M. Ghorbanian, P. Siano, and N. D. Hatziargyriou. An enhanced ieee 33 bus benchmark test system for distribution system studies. *IEEE Trans. Power Syst.*, 36(3): 2565–2572, 2021. doi: 10.1109/TPWRS.2020.3038030.

[17] M. Esmalifalak, L. Liu, N. Nguyen, R. Zheng, and Z. Han. Detecting stealthy false data injection using machine learning in smart grid. *IEEE Systems Journal*, 11(3):1644–1652, 2014.

[18] T. Esram and P. L. Chapman. Comparison of photovoltaic array maximum power point tracking techniques. *IEEE Transactions on Energy Conversion*, 22(2):439–449, 2007. doi: 10.1109/TEC.2006.874230.

[19] J. A. Fay and D. S. Golomb. Energy and the environment. 2 2002. URL https://www.osti.gov/biblio/20727641.

[20] N. Femia, G. Petrone, G. Spagnuolo, and M. Vitelli. Optimization of perturb and observe maximum power point tracking method. *IEEE Transactions on Power Electronics*, 20(4): 963–973, 2005. doi: 10.1109/TPEL.2005.850975.

[21] T. Gangwar, N. P. Padhy, and P. Jena. Storage allocation in active distribution networks considering life cycle and uncertainty. *IEEE Transactions on Industrial Informatics*, 19(1):339–350, 2023. doi: 10.1109/TII.2022.3167382.

[22] S. Gönen, H. H. Sayan, E. N. Yılmaz, F. Üstünsoy, and G. Karacayılmaz. False data injection attacks and the insider threat in smart systems. *Computers & Security*, 97:101955, 2020.

[23] H. Guo, J. Sun, and Z.-H. Pang. Stealthy fdi attacks against networked control systems using two filters with an arbitrary gain. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 69(7):3219–3223, 2022. doi: 10.1109/TCSII.2022.3153481.

[24] L. Hansen, V. Lacy, and D. Glick. A review of solar pv benefit & cost studies. *Rocky Mountain Institute*, 2013.

[25] Y. He, G. J. Mendis, and J. 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. doi: 10.1109/TSG.2017.2703842.

[26] Y. Ho and S. Wookey. The real-world-weight cross-entropy loss function: Modeling the costs of mislabeling. *IEEE Access*, 8:4806–4813, 2020. doi: 10.1109/ACCESS.2019.2962617.

[27] V. Hodge and J. Austin. A survey of outlier detection methodologies. *Artificial intelligence review*, 22:85–126, 2004.

[28] H. Ibrahim, J. Kim, P. Enjeti, P. R. Kumar, and L. Xie. Detection of cyber attacks in grid-tied pv systems using dynamic watermarking. In *2022 IEEE GreenTech*, pages 57–61, 2022. doi: 10.1109/GreenTech52845.2022.9772036.

[29] H. Ishii, S. Yoshizawa, Y. Fujimoto, I. Ono, T. Onoda, and Y. Hayashi. Cyber security for voltage control of distribution systems under data falsification attacks. In *Design and Analysis of Distributed Energy Management Systems: Integration of EMS, EV, and ICT*, pages 145–165. Springer, 2020.

[30] Y. Isozaki, S. Yoshizawa, Y. Fujimoto, H. Ishii, I. Ono, T. Onoda, and Y. Hayashi. Detection of cyber attacks against voltage control in distribution power grids with pvs. *IEEE Transactions on Smart Grid*, 7(4):1824–1835, 2016. doi: 10.1109/TSG.2015.2427380.

[31] D. Jafarigiv, K. Sheshyekani, M. Kassouf, Y. Seyedi, H. Karimi, and J. Mahseredjian. Countering fdi attacks on ders coordinated control system using fmi-compatible cosimulation. *IEEE Transactions on Smart Grid*, 12(2):1640–1650, 2021. doi: 10.1109/TSG.2020.3034745.

[32] R. Jiang, R. Lu, Y. Wang, J. Luo, C. Shen, and X. Shen. Energy-theft detection issues for advanced metering infrastructure in smart grid. *Tsinghua Science and Technology*, 19(2):105–120, 2014. doi: 10.1109/TST.2014.6787363.

[33] T. B. Johansson, H. Kelly, L. Burnham, A. K. Reddy, and R. Williams. *Renewable energy: sources for fuels and electricity*. Island press, 1993.

[34] A. Joseph, K. Smedley, and S. Mehraeen. Secure power distribution against reactive power

control malfunction in der units. *IEEE Transactions on Power Delivery*, 36(3):1552–1561, 2021. doi: 10.1109/TPWRD.2020.3011376.

[35] R. Kabiri, D. G. Holmes, and B. P. McGrath. The influence of pv inverter reactive power injection on grid voltage regulation. In *2014 IEEE 5th International Symposium on Power Electronics for Distributed Generation Systems (PEDG)*, pages 1–8, 2014. doi: 10.1109/PEDG.2014.6878640.

[36] R. Kaviani and K. W. Hedman. A detection mechanism against load-redistribution attacks in smart grids. *IEEE Transactions on Smart Grid*, 12(1):704–714, 2020.

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

[38] E. Lakervi and E. J. Holmes. *Electricity distribution network design*. Number 212. IET, 1995.

[39] F. Li, R. Xie, B. Yang, L. Guo, P. Ma, J. Shi, J. Ye, and W. Song. Detection and identification of cyber and physical attacks on distribution power grids with pvs: An online high-dimensional data-driven approach. *IEEE Journal of Emerging and Selected Topics in Power Electronics*, 10(1):1282–1291, 2022. doi: 10.1109/JESTPE.2019.2943449.

[40] Q. Li, J. Zhang, J. Zhao, J. Ye, W. Song, and F. Li. Adaptive hierarchical cyber attack detection and localization in active distribution systems. *IEEE Transactions on Smart Grid*, 13(3):2369–2380, 2022. doi: 10.1109/TSG.2022.3148233.

[41] G. Liang, J. Zhao, F. Luo, S. R. Weller, and Z. Y. Dong. A review of false data injection attacks against modern power systems. *IEEE Transactions on Smart Grid*, 8(4):1630–1638, 2017. doi: 10.1109/TSG.2015.2495133.

[42] X. Liu, Z. Li, X. Liu, and Z. Li. Masking transmission line outages via false data injection attacks. *IEEE Transactions on Information Forensics and Security*, 11(7):1592–1602, 2016. doi: 10.1109/TIFS.2016.2542061.

[43] Y. Liu, P. Ning, and M. K. Reiter. False data injection attacks against state estimation in

electric power grids. *ACM Transactions on Information and System Security (TISSEC)*, 14(1): 1–33, 2011.

[44] C.-H. Lo and N. Ansari. Consumer: A novel hybrid intrusion detection system for distribution networks in smart grid. *IEEE Transactions on Emerging Topics in Computing*, 1(1):33–44, 2013. doi: 10.1109/TETC.2013.2274043.

[45] H. Long, Z. Wu, C. Fang, W. Gu, X. Wei, and H. Zhan. Cyber-attack detection strategy based on distribution system state estimation. *J. Mod. Power Syst.*, 8(4):669–678, 2020. doi: 10.35833/MPCE.2019.000216.

[46] I. Lukicheva, D. Pozo, and A. Kulikov. Cyberattack detection in intelligent grids using nonlinear filtering. In *2018 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, pages 1–6, 2018. doi: 10.1109/ISGTEurope.2018.8571457.

[47] L. Luo, W. Gu, Y. Wang, and C. Chen. An affine arithmetic-based power flow algorithm considering the regional control of unscheduled power fluctuation. *Energies*, 10(11):1794, 2017.

[48] J. Machowski, J. Bialek, J. R. Bumby, and J. Bumby. *Power system dynamics and stability*. John Wiley & Sons, 1997.

[49] M. Markou and S. Singh. Novelty detection: a review—part 1: statistical approaches. *Signal processing*, 83(12):2481–2497, 2003.

[50] M. Markou and S. Singh. Novelty detection: a review—part 2:: neural network based approaches. *Signal processing*, 83(12):2499–2521, 2003.

[51] MATPOWER. MATPOWER/case33bw: Power flow data for 33 bus distribution system. https://github.com/MATPOWER/matpower/blob/master/data/case33bw.m, 2021. Accessed on February 16, 2023.

[52] MATPOWER Development Team. MATPOWER documentation: NEWTONPF Solves the power flow using a full Newton's method. https://matpower.org/docs/ref/matpower5.0/newtonpf.html, 2022. Accessed on February 16, 2023.

[53] S. McLaughlin, D. Podkuiko, and P. McDaniel. Energy theft in the advanced metering infrastructure. In *Critical Information Infrastructures Security: 4th International Workshop, CRITIS 2009, Bonn, Germany, September 30-October 2, 2009. Revised Papers 4*, pages 176–187. Springer, 2010.

[54] V. Mehta and R. Mehta. *Principles of Power System: Including Generation, Transmission, Distribution, Switchgear and Protection: for BE/B. Tech., AMIE and Other Engineering Examinations*. S. Chand Publishing, 2005.

[55] A. Meng, H. Wang, S. Aziz, J. Peng, and H. Jiang. Kalman filtering based interval state estimation for attack detection. *Energy Procedia*, 158:6589–6594, 2019.

[56] F. Mohammadi. Emerging challenges in smart grid cybersecurity enhancement: A review. *Energies*, 14(5):1380, 2021.

[57] D. K. Molzahn and J. Wang. Detection and characterization of intrusions to network parameter data in electric power systems. *IEEE Transactions on Smart Grid*, 10(4):3919–3928, 2019. doi: 10.1109/TSG.2018.2843721.

[58] R. Moslemi, A. Mesbahi, and J. M. Velni. A fast, decentralized covariance selection-based approach to detect cyber attacks in smart grids. *IEEE Transactions on Smart Grid*, 9(5):4930–4941, 2018. doi: 10.1109/TSG.2017.2675960.

[59] A. S. Musleh, G. Chen, and Z. Y. Dong. A survey on the detection algorithms for false data injection attacks in smart grids. *IEEE Transactions on Smart Grid*, 11(3):2218–2234, 2020. doi: 10.1109/TSG.2019.2949998.

[60] R. P. Narasipuram, C. Somu, R. T. Yadlapalli, and L. S. Simhadri. Efficiency analysis of maximum power point tracking techniques for photovoltaic systems under variable conditions. *International Journal of Innovative Computing and Applications*, 9(4):230–240, 2018.

[61] M. Ozay, I. Esnaola, F. T. Yarman Vural, S. R. Kulkarni, and H. V. Poor. Machine learning methods for attack detection in the smart grid. *IEEE Transactions on Neural Networks and Learning Systems*, 27(8):1773–1786, 2016. doi: 10.1109/TNNLS.2015.2404803.

[62] M. H. Rehmani, M. Reisslein, A. Rachedi, M. Erol-Kantarci, and M. Radenkovic. Integrating renewable energy resources into the smart grid: Recent developments in information and communication technologies. *IEEE Transactions on Industrial Informatics*, 14(7):2814–2825, 2018. doi: 10.1109/TII.2018.2819169.

[63] P. Rose. Underground power transmission. *Science*, 170(3955):267–273, 1970.

[64] A. Sayghe, Y. Hu, I. Zografopoulos, X. Liu, R. G. Dutta, Y. Jin, and C. Konstantinou. Survey of machine learning methods for detecting false data injection attacks in power systems. *IET Smart Grid*, 3(5):581–595, 2020.

[65] S. N. Singh. *Electric power generation: transmission and distribution*. PHI Learning Pvt. Ltd., 2008.

[66] Y.-H. Song and A. T. Johns. *Flexible ac transmission systems (FACTS)*. Number 30. IET, 1999.

[67] S. Sridhar and M. Govindarasu. Model-based attack detection and mitigation for automatic generation control. *IEEE Transactions on Smart Grid*, 5(2):580–591, 2014. doi: 10.1109/TSG.2014.2298195.

[68] A. Teixeira, G. Dán, H. Sandberg, R. Berthier, R. B. Bobba, and A. Valdes. Security of smart distribution grids: Data integrity attacks on integrated volt/var control and countermeasures. In *2014 American Control Conference*, pages 4372–4378, 2014. doi: 10.1109/ACC.2014.6859265.

[69] L. Vegh. Cyber-physical systems security through multi-factor authentication and data analytics. In *2018 IEEE International Conference on Industrial Technology (ICIT)*, pages 1369–1374, 2018. doi: 10.1109/ICIT.2018.8352379.

[70] Q. Wang, W. Tai, Y. Tang, and M. Ni. Review of the false data injection attack against the cyber-physical power system. *IET Cyber-Physical Systems: Theory & Applications*, 4(2): 101–107, 2019.

[71] S. Wang, S. Bi, and Y.-J. A. Zhang. Locational detection of the false data injection attack in a smart grid: A multilabel classification approach. *IEEE Internet of Things Journal*, 7(9): 8218–8227, 2020.

[72] Y.-B. Wang, C.-S. Wu, H. Liao, and H.-H. Xu. Steady-state model and power flow analysis of grid-connected photovoltaic power system. In *2008 IEEE International Conference on Industrial Technology*, pages 1–6, 2008. doi: 10.1109/ICIT.2008.4608553.

[73] H. L. Willis. *Power distribution planning reference book*. CRC press, 2004.

[74] J. Yan, B. Tang, and H. He. Detection of false data attacks in smart grid with supervised learning. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 1395–1402, 2016. doi: 10.1109/IJCNN.2016.7727361.

[75] J. Ye, A. Giani, A. Elasser, S. K. Mazumder, C. Farnell, H. A. Mantooth, T. Kim, J. Liu, B. Chen, G.-S. Seo, W. Song, M. D. R. Greidanus, S. Sahoo, F. Blaabjerg, J. Zhang, L. Guo, B. Ahn, M. B. Shadmand, N. R. Gajanur, and M. A. Abbaszada. A review of cyber–physical security for photovoltaic systems. *IEEE Journal of Emerging and Selected Topics in Power Electronics*, 10(4):4879–4901, 2022. doi: 10.1109/JESTPE.2021.3111728.

[76] Y. Yuan, Z. Li, and K. Ren. Modeling load redistribution attacks in power systems. *IEEE Transactions on Smart Grid*, 2(2):382–390, 2011.

[77] J. Zhao, G. Zhang, M. La Scala, Z. Y. Dong, C. Chen, and J. Wang. Short-term state forecasting-aided method for detection of smart grid general false data injection attacks. *IEEE Transactions on Smart Grid*, 8(4):1580–1590, 2017. doi: 10.1109/TSG.2015.2492827.

[78] W. Zhe, C. Wei, and L. Chunlin. Dos attack detection model of smart grid based on machine learning method. In *2020 IEEE International Conference on Power, Intelligent Computing and Systems (ICPICS)*, pages 735–738, 2020. doi: 10.1109/ICPICS50287.2020.9202401.

[79] Y. Zhou, H. Li, and L. Liu. Integrated autonomous voltage regulation and islanding detection for high penetration pv applications. *IEEE Transactions on Power Electronics*, 28(6):2826–2841, 2013. doi: 10.1109/TPEL.2012.2218288.

[80] P. Zhuang, R. Deng, and H. Liang. False data injection attacks against state estimation in multiphase and unbalanced smart distribution systems. *IEEE Trans. Smart Grid*, 10(6):6000–6013, 2019. doi: 10.1109/TSG.2019.2895306.