January 2022

Real-Time Machine Learning Models To Detect Cyber And Physical Anomalies In Power Systems

Zakaria El Mrabet

REAL-TIME MACHINE LEARNING MODELS TO DETECT CYBER AND PHYSICAL ANOMALIES IN POWER SYSTEMS

by

Zakaria El Mrabet
Master of Science, Ibn Tofail University, 2014

A Dissertation
Submitted to the Graduate Faculty
of the
University of North Dakota
In partial fulfillment of the requirements

for the degree of
Doctor of Philosophy

Grand Forks, North Dakota

May 2022

Name: Zakaria El Mrabet

Degree: Doctor of Philosophy

This document, submitted in partial fulfillment of the requirements for the degree from the University of North Dakota, has been read by the Faculty Advisory Committee under whom the work has been done and is hereby approved.

DocuSigned by:
Prakash Ranganathan

Prakash Ranganathan

DocuSigned by:
Hossein Salehfar

Hossein Salehfar

DocuSigned by:

Adams Ryan

DocuSigned by:
Michael Mann

Michael Mann

This document is being submitted by the appointed advisory committee as having met all the requirements of the School of Graduate Studies at the University of North Dakota and is hereby approved.

DocuSigned by:
Chris Nelson

Chris Nelson
Dean of the School of Graduate Studies

5/4/2022

Date

PERMISSION

Title	Real-time Machine Learning Models to Detect Cyber and Physical Anomalies in Power Systems
Department	School of Electrical Engineering and Computer Science
Degree	Doctor of Philosophy

In presenting this dissertation in partial fulfillment of the requirements for a graduate degree from the University of North Dakota, I agree that the library of this University can make it freely available for inspection. I further agree that permission for extensive copying for scholarly purposes may be granted by the professor who supervised my dissertation work or, in her absence, by the Chairperson of the department or the dean of the School of Graduate Studies. It is understood that any copying or publication or other use of this dissertation or part thereof for financial gain shall not be allowed without my written permission and that of my advisor. It is also understood that due recognition shall be given to me and to my advisor in any scholarly use which may be made of any material in my dissertation.

Zakaria El Mrabet May 4th, 2022

ACKNOWLEDGMENT

I extend my sincerest thanks to Dr. Prakash Ranganathan for his valuable mentoring, precious advice, and constant encouragement during my Ph.D. journey. Dr. Prakash has been instrumental in my successful preparation of this dissertation thanks to his rich scientific experience, proactive support, and remarkable guidance. His student-centric and empathetical management style helped me move smoothly through the rigorous academic processes of the Ph.D. program. I would like to express my gratitude to Dr. Shrirang Abhyankar, who guided me through my internship at Pacific North West National Laboratory (PNNL) where I interacted with research scholars from diverse disciplines and explored new areas of research. I would like to thank the National Science Foundation (NSF) (award #153756), the School of Electrical Engineering and Computer Science (SEECS), and the North Dakota Established Program to Stimulate Competitive Research (ND EPSCoR) for granting financial support during my Ph.D. I would like to thank Dr. Naima Kaabouch for welcoming me into her research lab and for all of her support.

Lastly, I would like to thank my parents, my aunt, Naima, my wife, Soumaya, my two brothers, and my friend, Youness, for their constant support and encouragement over the last 4 years.

DEDICATION

In the memory of my aunt, Malika

ABSTRACT

A Smart Grid is a cyber-physical system (CPS) that tightly integrates computation and networking with physical processes to provide reliable two-way communication between electricity companies and customers. However, the grid availability and integrity are constantly threatened by both physical faults and cyber-attacks which may have a detrimental socio-economic impact. The frequency of the faults and attacks is increasing every year due to the extreme weather events and strong reliance on the open internet architecture that is vulnerable to cyber-attacks. In May 2021, for instance, Colonial Pipeline, one of the largest pipeline operators in the U.S., transports refined gasoline and jet fuel from Texas up the East Coast to New York was forced to shut down after being attacked by ransomware, causing prices to rise at gasoline pumps across the country. Enhancing situational awareness within the grid can alleviate these risks and avoid their adverse consequences. As part of this process, the phasor measurement units (PMU) are among the suitable assets since they collect time-synchronized measurements of grid status (30-120 samples/s), enabling the operators to react rapidly to potential anomalies. However, it is still challenging to process and analyze the open-ended source of PMU data as there are more than 2500 PMU distributed across the U.S. and Canada, where each of which generates more than 1.5 TB/month of streamed data. Further, the offline machine learning algorithms cannot be used in this scenario, as they require loading and scanning the entire dataset before processing. The ultimate objective of this dissertation is to develop early detection of cyber and physical anomalies in a real-time streaming environment setting by mining multi-variate large-scale synchrophasor data. To accomplish this objective, we start by investigating the cyber and physical anomalies, analyzing their impact, and critically reviewing the current detection approaches. Then, multiple machine learning models were designed to identify physical and cyber anomalies; the first one is an artificial neural network-based approach for detecting the False Data Injection (FDI) attack. This attack was specifically selected as it poses a serious risk to the integrity and availability of the grid; Secondly, we extend this approach by developing a Random Forest Regressor-based model which not only detects anomalies, but also identifies their location and duration; Lastly, we develop a real-time hoeffding tree-based model for detecting anomalies in steaming networks, and explicitly handling concept drifts. These models have been tested and the experimental results confirmed their superiority over the state-of-the-art models in terms of detection accuracy, false-positive rate, and processing time, making them potential candidates for strengthening the grid's security.

Keywords: cyber-attacks, physical faults, Hoeffding Tree, Transfer learning, ADWIN, FDI, RFR, PMU

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ADWIN	Adaptive sliding windows
ANN	Artificial Neural Network
AMI	Advanced Metering Infrastructure
BNP	Back Propagation Based Neural Network
CFG	Configuration Frame
CNN	Convolutional Neural Network
CPS	Cyber-Physical System
CRC	Cyclic Redundancy Check
DBSCAN	Density-Based Spatial Clustering Of Applications With Noise
DDM	Drift Detection Method
DLG	Double Line To Ground
DNN	Deep Neural Network
DNP	Distributed Network Protocol
DoS	Denial of Service
DT	Decision Trees
DWT	Discrete Wavelet Transform

EDDM	Early Drift Detection Methods
ELE	Event Location Estimation
FDI	False Data Injection
FNN	Feed-Forward NN
GA	Genetic Algorithm
GPS	Global Positioning System
HAT	Hoeffding Adaptive Tree
HT	Hoeffding Tree
HMI	Human-Machine Interfaces
IED	Intelligent Electronic Devices
IT	Information Technology
IJ	Just in time
KNN	K-Nearest Neighbors
LFR	Linear Four Rate
LG	Line To Ground
LL	Line To Line
LLL	Three-Phase To The Ground
LSTM	Long-Short-Term-Memory
MDMS	Meter Data Management System
MITM	Man-In-The-Middle Attack
MLE	Maximum Likelihood Estimation
MTU	Master Terminal Unit
MOA	Massive Online Analysis
NB	Naive Bayes
NIST	National Institute of Standards and Technology
OT	Operational Technology
PDC	Phasor Data Concentrator
PDT	Physics-Based Decision Tree
PFA	Probability of False Alarm
PLC	Power Line Communication
PLO	Phase Locked Oscillator

PMU	Phasor Measurement Unit
PNN	Probabilistic Neural Network
RBF	Radial Basis Function
RF	Random Forest
RFR	Random Forest Regressor
RL	Reinforcement Learning
RNN	Recurrent Neural Networks
RTU	Remote Terminal Units
SCADA	Supervisory Control And Data Acquisition
SVM	Support Vector Machine
SLG	Single Line To Ground
TP	Three-Phase Short Circuit
WAMS	Wide Area Measurement System
WSGN	Wireless Smart Grid Network

Chapter I

Introduction

1. Motivation and Problem Statement

The power grid plays a critical role in the smooth functioning of modern society by supplying electricity to all its key pillars including transportation, communication, and health systems. Thus, ensuring the availability and the integrity of the power grid is indispensable for providing a good quality of life. Additionally, the power system is an important pillar for U.S national security since the U.S military installations and operations rely heavily on it. Due to its critical nature, the power grid can be subject to several malicious cyber-attacks which could cause fear in society and leave a serious socio-economic impact. On December 23, 2015, the Ukrainian power grid was subject to a large cyber-attack where three different distribution companies were attacked. This incident resulted in several outages that caused approximately 225, 000 customers to lose power across various areas. The attack was due to a third party's illegal penetration into the distribution companies' computer and Supervisory control and data acquisition (SCADA) system which disconnected seven 110 kV and twenty-three 35kV substations for more than three hours [1]. A few years later, on Marth 5th 2019, the power grid control systems in Utah, Wyoming, and California were subject to a Denial of Service (DoS) attack causing disturbances and loss of visibility in certain sections of the utility's SCADA system [2]. Recently, in May 2021, Colonial Pipeline, one of the largest pipeline operators in the U.S., transports refined gasoline and jet fuel from Texas up the East Coast to New York was forced to shut down after being attacked by ransomware, causing prices to rise at gasoline pumps across the country [3]. These cyber incidents, which could cost the U.S. economy $1 trillion [4], illustrate the detrimental impacts of cyber-attacks and the economic burden that they can bring to any nation's critical infrastructure.

Although the wide variety of industrial cyber-attacks, including the time synchronization attack, replay attack, DoS attack, man-in-the-middle attack (MITM), and false data injection attack (FDI) [5]–[8], they typically fall into one of the three major categories of attacks targeting availability, integrity, or confidentiality. The first category includes the attacks which aim at delaying or blocking the communication in the power system, such as the time synchronization and DoS attacks [5], [7]. The IEC standard 618501 [9] mainly used for power substation automation, defines several message types with specific timing constraints. The most time-critical message types are the Type 1A/P1 and Type 1A/P2 which are used for Generic Object-Oriented Substation Event trip protection purposes. These messages have two end-to-end delay constraints: 3 ms and 10 ms [9], respectively. In other words, compromising the availability or even causing a delay for more than 10 ms in substations can block the exchange of critical protection messages. The second category includes attacks that alter and modify the exchanged instructions in the power system, such as the relay attack [6]. In the SCADA system where the Modbus protocol is used for exchanging instructions between a Master (e.g. Master Terminal Unit (MTU)) and slave (e.g. Programmable Logic Controller (PLC)), the replay attack can be used to intercept and inject falsified instructions. The last category comprises attacks that aim at getting unauthorized access to the data in the power grid network, such as the MITM attack [10], [11]. The order of precedence of security criteria is different depending on the type of network operations. For instance, in the conventional communication network, confidentiality is the most important security criterion followed by integrity and then availability. However, in the Smart Grid network, especially in the Advanced Metering Infrastructure (AMI) and the Home Area Network, the availability and integrity precede confidentiality. In recent years, the electrical distribution system is greatly supported by AMI which integrates smart meters with communication networks to provide advanced functionalities to the customers. Unfortunately, the cyber-attacks on AMI present a clear danger to both utilities and customers. FDI attack was the major reason behind the most devastating scenario of the Ukraine blackout [12].

Improving situational awareness within the grid is an effective preventive measure to detect potential anomalies, and avoid their adverse consequences. PMUs are reliable sensors that can collect relevant measurements and assist in increasing visibility within the grid [13]. PMUs collect magnitude, phase angle, frequency, voltage, and current, with a precise GPS-based timestamp [14]. So far, there are more than 2500 PMUs deployed in the North American power grid, where each of which generates between 30 and 60 samples/s [15], roughly 1.5 TB/month of streamed data.

However, processing such large PMU streams requires expensive computational resources and faster algorithms. Further, the conventional ML algorithms, cannot be used in this scenario, as they require loading and scanning the entire dataset before processing. Thus, two important criteria have to be met by any streaming ML technique for handling PMU data: 1) training the model with recent history holding shorter signatures; and 2) adapting to concept drifts (e.g., fluctuations) in real-time. It is important to note that the model trained on a historical record will no longer have relevance to the incoming data stream, and thus may fail to capture any critical or new events. Hence, meeting the above two criteria is key for successful and future real-time streaming algorithms.

2. Dissertation goal and objectives

Considering the cyber and physical security issues mentioned above, the goal of this dissertation is **"to develop efficient and accurate models for detecting potential anomalies and cyberattacks in the smart grid in real-time"**. To achieve this goal, the following objectives were set and met:

- Investigate the existing anomaly detection approaches for PMU data and design approaches to model both physical and cyber anomalies in the Smart Grid environment.
- Investigate existing algorithms to detect fault location and duration, and examine their performances and their application in a real-time environment.
- Develop real-time machine learning models to classify concept drifts and anomaly signatures using transfer learning.

3. Contributions

The contributions of this dissertation are as follows:

1. Conducting a state-of-the-art review on anomaly detection approaches

The first contribution of this dissertation provides a comprehensive investigation of the existing anomalies in the smart grid. The anomalies are categorized into physical and cyber events; their impact on confidentiality, integrity, and availability are also examined. Then we provide a critical review and classification of existing anomaly detection approaches into conventional, machine learning, and hybrid approaches, as well as their main advantages and limitations.

2. Modeling False Data Injection (FDI) attacks and developing an Artificial Neural Network (ANN) detection approach.

The second contribution provides an ANN-based approach to detecting FDI attacks. These attacks have been particularly selected as they target the *integrity* of the system and threaten its availability. First, the FDI attacks will be modeled and used to simulate two attack scenarios to generate the appropriate dataset. Next, an Artificial Neural Network based model is developed, trained, and compared against state-of-the-art models using several performance metrics including accuracy and probability of false alarm.

3. Developing Random Forest Regressor (RFR) model to detect fault locations and predict their duration.

In addition to detecting anomalies, as given by the first contribution, it is necessary to predict their location, especially in such heterogeneous and highly interconnected systems as the power system. Thus, this contribution is about developing a regression model to detect both fault location and duration. The RFR model has been trained on a synthetic dataset generated based on various fault attack scenarios simulated in the GridPACK framework, which is an open-source framework designed to support the development and implementation of Smart Grid applications developed by the Pacific Northwest National Laboratory (PNNL). Then, the RFR model has been extensively tested throughout four case studies: detection of fault location, predicting fault duration, handling missing, and streaming data.

4. Developing a Real-time anomaly detection approach for streaming PMU data.

The streaming nature of the PMU data requires quick and accurate scanning when it comes to detecting anomaly events. Conventional machine learning approaches are not reliable to detect anomalies from the PMU data streaming environment. In this contribution, a transfer learning-based hoeffding tree with ADWIN (THAT) is proposed to detect anomalies in a real-time environment setting. The THAT model is trained on four event signatures with varying durations and extensively tested and compared to existing online machine learning models.

The aforementioned contributions were published and presented in the following peer-reviewed journals and conferences:

- El Mrabet, Z.; Sugunaraj, N.; Ranganathan, P.; Abhyankar, S. Random Forest Regressor-Based Approach for Detecting Fault Location and Duration in Power Systems. Sensors 2022, 22, 458. https://doi.org/10.3390/s22020458. (Journal)
- Z. E. Mrabet, D. F. Selvaraj and P. Ranganathan, "Adaptive Hoeffding Tree with Transfer Learning for Streaming Synchrophasor Data Sets," 2019 IEEE International Conference on Big Data (Big Data), 2019, pp. 5697-5704, DOI: 10.1109/BigData47090.2019.9005720. (Conference)
- Z. E. Mrabet, D. F. Selvaraj, A. S. Nair, and P. Ranganathan, "Detection of the False Data Injection Attack in Home Area Networks using ANN," 2019 IEEE International Conference on Electro Information Technology (EIT), 2019, pp. 176-181, DOI: 10.1109/EIT.2019.8834036. (Conference)
- Z. Mrabet and P. Ranganathan, “Cyber-Physical Attack Detection Approaches in the Wide Area Measurement System: A Survey”, submitted to International Journal of Electrical Power and Energy Systems, Elsevier. (Journal)

4. Dissertation organization

This dissertation is organized as follows: Chapter 2 provides an overview of the existing approaches to identifying anomalies in the power system. In particular, it describes conventional, machine learning-based, and hybrid approaches to detecting cyber and physical anomalies. Chapter 3 describes the proposed ANN-based approach to detecting FDI attacks. The attack has been modeled and then simulated to generate the appropriate dataset. The proposed machine learning model has been trained, assessed, and compared with state-of-the-art models using several metrics. Chapter 4 focuses on detecting anomaly locations and predicting their duration. Several fault scenarios were considered and simulated in GridPACK to generate the dataset used for training and testing a Random Forest Regressor model. In addition, a comprehensive parametric study will be presented in which seven state-of-the-art models are hyper-tuned and then compared to the proposed model based on several metrics. Chapter 5 introduces the proposed model for detecting anomalies in streaming networks. In addition, it explains the dataset used to train and test the models, and it discusses the model performance results. Chapter 6 concludes the dissertation and sheds light on some future research directions.

Chapter II

Cyber and Physical Anomalies Detections Approaches in Smart Grid

In this chapter, we set the stage for this research. We begin by defining the research context, which is the Smart Grid and its cybersecurity requirements. Then, it describes the various Smart Grid domains, their key systems, and the various cyber and physical anomalies. Next, it reviews thoroughly the existing detection approaches and discusses their advantages and limitations. The remainder of the chapter is organized as follows: [1]Section I explains the Smart Grid, its domains, and key systems. Section II discusses the fundamental parameters of cybersecurity in Smart Grid. Section III categorized anomalies into cyber and physical and discussed their security impact. Section IV is dedicated to reviewing the detection approaches, which are classified into conventional, machine learning-based, and hybrid approaches, and highlighting their advantages and limitations.

1. Introduction

The National Institute of Standards and Technology (NIST) describes a Smart Grid as a collection of seven logical domains: generation, transmission, distribution, markets, customers, service providers, and operations, as shown in Figure 1. Each domain includes both actors and applications; actors are programs, devices, and systems, while applications

[1] This chapter is a slightly modified version of our paper: Z.E. Mrabet, P. Ranganathan, "Cyber-Physical Attack Detection Approaches in the Wide Area Measurement System: A Survey" Submitted to International Journal of Electrical Power and Energy Systems, Elsevier.

are tasks performed by one or more actors in each domain [16]. The following is a detailed description of each domain, along with its principal actors and applications.

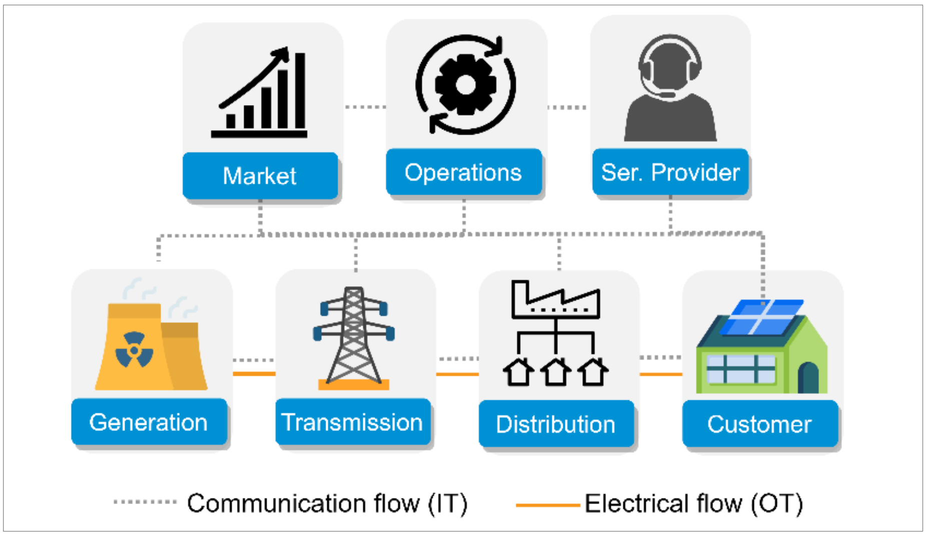


Figure 1. NIST's seven domains of the Smart Grid

The main actor in the customer domain is the end-user. There are generally three types of customers: residential, commercial/building, and industrial. Besides consuming electricity, these actors may also produce, store, and manage it. This domain is electrically connected to the distribution domain and interacts with the distribution, operation, service provider, and market domains [16].

The bulk generation domain actors include electricity producers that generate electricity in significant quantities using resources such as oil, flowing water, coal, nuclear fission, and solar radiation. This domain is electrically connected to the transmission domain and communicates via an interface with the market domain, transmission domain, and operations domain [16].

In the transmission domain, electrical power is transmitted over long distances from the generation domain to the distribution domain via multiple substations. It may also be used to store and generate electricity. Monitoring and controlling the transmission network is achieved through a SCADA system, which is made up of a communication network, control devices, and monitoring devices [16].

The distribution domain includes all entities involved in the distribution of electricity to and from end-users. There are various designs for electrical distribution systems including radial, looped, and meshed. This domain is interconnected to the transmission domain, customer domain, and consumption metering points; in addition to supplying energy to the final consumer, this domain may also be involved in producing and storing energy [16].

Actors in the market domain are the operators and participants in the electricity markets. This domain maintains the balance between electricity supply and demand. To match production with demand, the market domain communicates with energy supply domains, including the bulk generation domain and distributed energy resources [16].

Actors in the operations domain are those responsible for managing the movement of electricity. This domain ensures efficient and optimal operations throughout the transmission and distribution networks. During transmission, it utilizes energy management systems whereas, during distribution, it uses distribution management systems [16].

The service provider domain includes organizations providing services to both electrical customers and utilities, as well as managing services such as billing, customer account, and use of energy. The service provider interacts with the operation domain to provide situational awareness, system control, as well as communicating with the customer and marketplace domains to develop smart services such as enabling customer interaction with the market and energy generation at home [16].

2. Smart grid’s system

The seven domains discussed above are interconnected through devices and applications. The Customer domain includes applications and devices such as smart meters, appliances, thermostats, energy storage, electric vehicles, and distributed generation. In the Operations domain, applications and devices include SCADA as well as computers or display systems in the operation center. In the Transmission and Distribution domains, applications and devices include PMUs in transmission line substations, substation controllers, distributed generation, and energy storage [17]. We will discuss in detail some of these Smart Grid's key components in the following section.

2.1. Substation automation

A substation is an instrumental component of the Smart Grid network; it carries out a variety of functions, including receiving power from generating facilities, regulating the distribution, and controlling power surges. The majority of these operations are automated within the substation to ensure greater grid reliability [18]. They are performed through a variety of systems and network protocols, including remote terminal units (RTUs), the global positioning system (GPS), human-machine interfaces (HMI), and intelligent electronic devices (IEDs) [21], and IEC 61850 [19].

2.2. PMU-based Wide Area Measurement System (WAMS)

The Phasor Measurement Unit provides situational awareness, operation, and reliability of the power system network [20]. It plays a major role in the Smart Grid network by

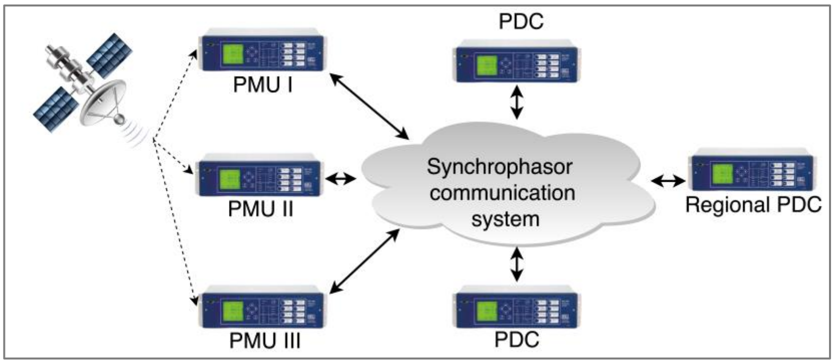


Figure 2 A typical PMU-based WAMS architecture

providing a near time-synchronized measurement that enables the grid operator to analyze the operating status of the power grid. Such a feature is achieved through the utilization of the synchrophasor which synchronizes the critical measurements received from different grid components widely distributed in the Smart Grid with a precise clock through the GPS. The PMU is regarded as the nervous system of the Smart Grids it is capable of providing time-synchronized measurements of the grid status, which enable the grid operators to react quickly to accidental and unexpected events.

Since the time-synchronized measurements produced by the PMUs are sent to Phasor Data Concentrator (PDC) or a control center, it is important to protect all the components in the synchrophasor communication system. this system is called the PMU-based WAMS. A typical WAMS is composed of PMUs, and local and regional PDC, as illustrated in Figure 2.

The PMU receives the three-phase voltages and currents coming from the power system network in the form of an analog signal. This signal is filtered through an anti-aliasing module to limit the bandwidth of the incoming signal and then converted into a digital signal via the (A/D) converter module. The digital signal is fed to the Central Processing Unit to compute the phase and magnitude of the signal using the Discrete Fourier

Transformer in conjunction with the timed GPS signal. The output signal is time-synchronized using a sampling clock that is phase-locked to the one-pulse-per-second provided by the GPS receiver. The pulse signals from the satellite are phase-locked with the sampling clock through the Phase Locked Oscillator (PLO) module. PLO divides the one pulse per second signal from GPS into the required number of pulses per second for sampling [21][22]. The time-stamped signal is transmitted to the local PDC. Then, the measurements from several local PDC are reported to the regional PDC. As shown in Figures 2 and 3.

The communication between the PMUs and PDC is defined through various communications technology including the Power Line Communication (PLC), Optical Fiber communication, Microwave communication, Cellular communication, and Satellite communication. The end-to-end communication between the PDC and the PMU is also based on the IEEE C37.118.2-2011 standard which defines the format of the messages for exchanging data between the PMU and the PDC or between PDCs. There are four types of messages types: data, configuration, header, and command frames. The header, configuration, and data frames are sent from the PMU to the PDC while the command frames are sent from the PDC to the PMU. The header frame provides information about the PMU and helps the PDC identify the PMU. In the configuration frame, there are three

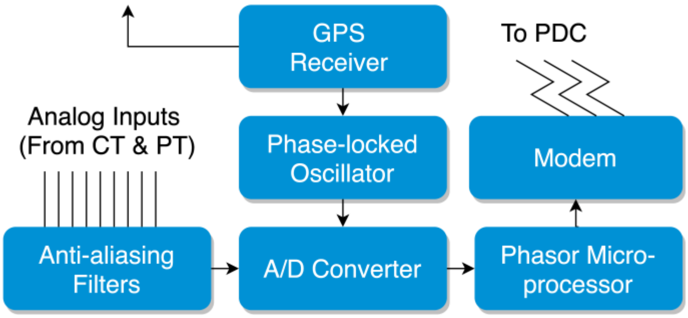


Figure 3. Phasor measurement unit components

types: CFG-1 which provides information about the capacity of the PMU, such as reporting rates and noise suppression, and CFG-2 and CFG-3 which provides information about the currently reported measurements. The data frame includes the real-time synchrophasor data measured by the PMU including the current, frequency and voltage, and amplitudes.

Figure 4 shows a typical real-time communication between a PMU and a PDC through the IEEE C37.118.2-2011 standard. The PDC initiates the communication by sending a command frame to turn on the transmission and sends a request for a configuration frame (CFG). The PMU responds to this request, and then PDC makes another request for synchrophasor data. The PMU sends back the requested data which is included in a data frame format. In the data frame parquets, various fields serve in initiating and terminating the communications. For instance, the SYNC field which marks the beginning of the frame, a Frame Size field provides the length of the entire frame, an IDCODE field which is used as an ID to identify the frame, a SOC field which provides information about the time stamp, A FRACSEC field which presents the time quality or time of measurements for data frames, and then a Cyclic Redundancy Check (CRC) which marks the end of the data frame [23] [24].

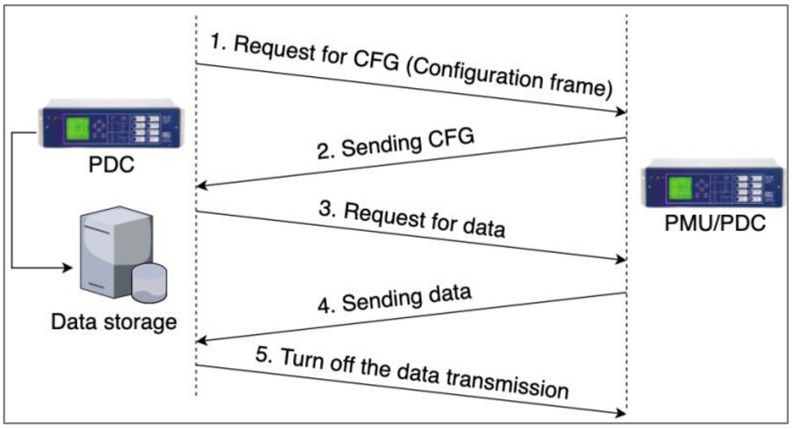

1. Request for CFG (Configuration frame)
PDC
2. Sending CFG
PMU/PDC
3. Request for data
Data storage
4. Sending data
5. Turn off the data transmission

Figure 4. PMU-PDC communication via the IEEE C37.118.2 protocol

2.3. Advanced Metering Infrastructure (AMI)

The AMI falls under the domains of the customer and the distribution and is responsible for collecting, measuring, and analyzing energy, water, and gas consumption. It consists of three main components: smart meters, an AMI headend, and network communications [25]. A smart meter is a digital meter that automatically collects in real-time the measurement data and transmits them to the AMI backend. A headend consists of two components: the AMI server, which collects meter data, and the meter data management system (MDMS), which manages collected data and shares it with other systems, such as demand response, historians, and billing systems. Communication between the smart meters and the AMI headend is defined through several network protocols, including power line communications (PLC), Zigbee, Z-Wave, and Wireless M-Bus [25].

2.4. Supervisory control and data acquisition (SCADA)

A SCADA system is used to measure, monitor, and control electrical power grids and is composed of three main components, including the RTU, MTU, and HMI [26]. Essentially, an RTU is composed of three components: the first is intended for data acquisition, the second is responsible for executing the instructions coming from the MTU, and the third is designed for communication; The MTU is responsible for controlling the RTU; HMI is a graphical user interface for SCADA system operators. SCADA systems communicate using a variety of industrial protocols, including distributed network protocol 3.0 (DNP3) and IEC 61850 [26].

3. Cybersecurity fundamental for Smart Grid

The NIST has identified three criteria for maintaining security and protecting information within the Smart Grid, including confidentiality, integrity, and availability [16]. The following paragraphs describe each criterion.

3.1. Confidentiality

Confidentiality refers to protecting both personal privacy and proprietary information from being accessed or disclosed by unauthorized entities, individuals, or processes [10]. In AMI systems, customers' energy consumption data such as patterns of energy consumption, metering usage, and billing information exchanged between a customer and various entities must be treated confidentially and privately; otherwise, the customer's information may be misused by unauthorized people or marketing firms [27]. Thus, the metrology and energy information contained within the smart meters should be treated with the utmost confidentiality, including the prevention of physical theft of meters and subsequently the stored data.

3.2. Availability

Availability is defined as ensuring timely and reliable access to information and systems. Reliable and interrupted communication is vital for continuously monitoring the state of the grid [10]. The unavailability of the system may cause a delay or discontinuity in the communication, which could adversely affect the system's stability, potentially resulting in a loss of power. An interruption in the network availability can, for example, disrupt the operation of a control system by blocking the flow of information across the network and thereby preventing operators from controlling the system [10], [27].

3.3. Integrity

Integrity refers to the preservation of data and systems against unauthorized modification or destruction [10]. The integrity of the grid can be compromised if an adversary succeeds in altering the sensors' measurements and relaying that biased information to the state estimator, resulting in an inaccurate estimation of the current state of the grid. The integrity requirement for AMI systems relates to both the integrity of the data being retrieved from the meter as well as the integrity of the control commands, such as preventing unauthorized control commands from being sent from the AMI headend to the smart meter [28]. Integrity requires both nonrepudiation and authenticity of the information. The principle of nonrepudiation states that individuals, entities, or organizations cannot perform a specific action and then deny it later; authenticity implies the fact that data originates from a reliable source [10], [27].

Accountability is another complimentary security criterion that can strengthen the security posture of the grid [29]. It refers to ensuring the traceability of the system and that the actions performed by a person, device, or public authority can be verified so that no one can deny their actions [29]. The recording of this information may be used as evidence in a court of law to identify the perpetrator [29].

4. Cyber-physical anomalies in Smart Grid

A Smart grid is a Cyber-Physical System that tightly integrates computation and networking with physical processes and relies on actuators and sensors to monitor and control complex physical processes, creating complex feedback loops between the physical and cyber worlds [30]. The cyber and physical systems are interconnected via information technology (IT) and operational technology (OT). IT refers to the application of networks for storing and transmitting data, such as AMI; while OT refers to monitoring and controlling specific devices, such as the SCADA system [31].

Broadly speaking, anomalies correspond to patterns and abnormal behavior in data. These nonconforming patterns are also referred to as outliers, discordant observation, and exceptions [32]. In Smart Grid contexts, anomalies refer to all malfunctions either intentionally or unintentionally caused by physical or cyber events. Throughout this dissertation, we classified anomalies into two classes: cyber and physical; all anomalous events that are directed at compromising the IT network are considered cyber-anomalous events, while those that target the OT network are termed physical malicious events. Example of physical anomalies includes aurora attacks, power system faults, and FDI attacks; while cyber anomalies include GPS spoofing, jamming, scanning, MITM, Viruses, and DoS attacks. Figure bellow illustrates how these various cyber and physical anomalies are directed to the Smart Grid's domains. In the following section, we will investigate several cyber and physical anomalies along with their impact on the Smart Grid's systems.

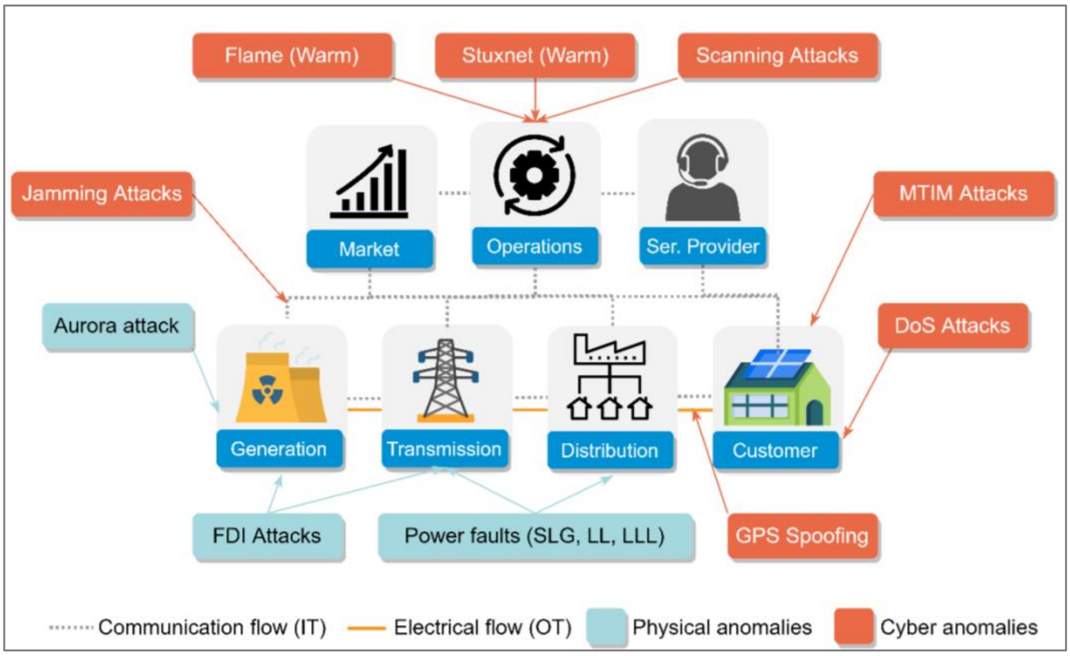


Figure 5. Cyber and Physical anomalies in the Smart Grid

4.1. Physical anomalous events

4.1.1. Aurora attack

A successful connection between a generation source and an electric grid requires coordination and synchronization of several parameters, such as frequency, voltage, and phase rotation. The protective relays are responsible for checking these parameters and allowing the connection only when they are within a pre-set tolerance (synchronism). Such tolerance contributes to a more reliable and robust power supply from the generator, as they permit a small degree of variation over a short period without permanently separating the generation sources [33]. The Aurora attack took advantage of this tolerance by intentionally tripping a breaker out of synchronism, causing mechanical and electrical stress, resulting in damage to equipment, as shown in Figure 6. Aurora attack targets mainly the availability of the generators, but it can also impact other equipment including motors, transformers, and adjustable frequency drives [33].

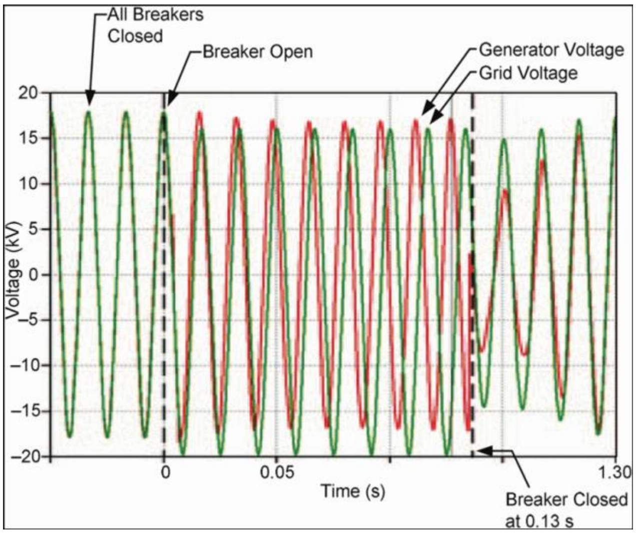


Figure 6. An Aurora attack scenario [33]

4.1.2. Power system faults

Power system physical faults are mostly caused by natural weather events, such as storms, high winds, or fallen trees. Sometimes, the cause can be due to an equipment failure, or a failing tower. These faults are generally classified into balanced and unbalanced faults [34][35]. An unbalanced fault, also known as an asymmetrical fault, is a commonly occurring fault in the power system and can be of a series or shunt type. In the series fault type, the voltage and frequency values increase, while the current level decreases at the faulty phases. In shunt fault, the current level rises, while the frequency and voltage levels decrease at the faulty phases. There are several shunt fault types: single line to ground (SLG), line to line (LL), double line to ground (DLG), and Three-phase to the ground (LLL). An SLG occurs when a transmission line phase touches a neutral wire or the ground; the DLG or LL fault occurs when two or more phases make a connection with the ground. According to [35], [36] the likelihood of occurrence of each fault type is 70% for SLG, 15% for LL, 10% for DL, and 5% for LLL. Although there is a low likelihood of LL fault occurrence, it is considered a severe fault that can cause a rise in fault current magnitude and thereby result in outages or large damage to grid assets. Hence, necessitates the need for fault detection and location identification model.

There are additional anomalous events that disturb the power system, such as voltage spike, sage and swell, and fluctuation. Voltage spikes are instantaneous or drastic rises in voltage within a short period. Voltage fluctuation may result from turbine governor problems or rapid changes in loads. Sag results mostly from an abrupt increment of impedance due to losing contact; while swell may come from rapid reactive power injection in the network [37]

4.1.3. False Data Injection (FDI) attacks

FDI attacks are designed to alter data stored in the control center or transmitted via communications infrastructure, which ultimately compromises the integrity of the power systems or even their availability. An attack of this type was first introduced by [38] in connection with DC system models, in which the authors assumed that the adversary had knowledge of the power system topology and parameters and could alter meter measurements while remaining consistent with physical laws such as Kirchhoff's circuit law to bypass the bad data detection process [31]. Such an attack could result in inaccurate estimates of the power system state, which is essential for a variety of power system applications, including economic dispatch and contingency analysis, which in turn could result in the state estimator sending incorrect values to the operator and leading to unstable system states, voltage collapses, and eventually physical and economic losses [39]. The authors of [40] examined the implications of a hidden FDI attack at the RTU level on AC state estimation and concluded that it had two negative consequences.

If the data is altered in a way that the state estimation schemes cannot detect as false, there will be an incorrect perception of the observable state of the system, which may lead the grid operator to take actions that endanger the system's security. Despite the detection of an attack, parts of the system may become unobservable, meaning the state estimator is unable to estimate changes in the state values (e.g., voltage magnitudes and voltage angles), placing the transmission grid at risk from a local physical attack. It may already be too late to prevent an outage of a greater part of the system by the time the effects of the physical attack have propagated into the rest of the system in which the state is observable. It has also been concluded that the FDI attacker using a DC model has a greater chance of introducing errors in the measurements and ultimately triggering bad data detection. Thus, the nonlinearity of the power flow equations provides advantages to the system operator to detect this type of FDI attack. However, if the attacker knows the estimated state, they can launch an AC FDI attack without being detected by the AC state estimation. But in reality, it is difficult for an attacker to obtain the same estimated state as the operators. Thus, authors in [41] proposed a sufficient condition for an imperfect and undetectable FDI attack. Authors in [42] proposed the design of blind FDI attacks based on little to no knowledge of the Smart Grid topology which would significantly reduce the attack cost.

Authors in [43] have suggested another FDI attack on the electricity market, using small changes to the price signals to increase the difference between the generated and consumed power. This attack is extended here [44], in which the attacker can inject false pricing data at any time and repeatedly over some time, resulting in over-generation, economic losses, and poor power quality. Further detail about this attack will be discussed in the next chapter.

4.2. Cyber anomalous events

4.2.1. GPS spoofing attacks

GSP spoofing against PMU device. A study conducted by Daniel P. et al. [45] showed how a GPS spoofer was able to manipulate the PMU readings and cause a plant to trip. Through their experiments, they demonstrate that an attacker can manipulate the PMU time stamp by injecting a falsified set of the GPS signal into the antenna of the PMU's time reference receiver. Injecting timing error for a few microseconds was enough for the PMU to violate the maximum phase error allowed by the applicable standards, Such errors can provide a false perception of the status of the grid, leading to unnecessary control actions [45]. Therefore, it is important to protect the PMU from such manipulations.

4.2.2. Jamming attacks

In the jamming attack, an adversary exploits the shared nature of the wireless network and sends a random or continuous flow of packets to keep the channel busy and then prevents legitimate devices from communicating and exchanging data [46]. Due to its time-critical nature, Smart Grid requires a highly available network to meet the quality of service requirements and such an attack can severely degrade its performance [46]. Keke G. et al. [47] proposed a jamming attack named maximum attacking strategy using spoofing and jamming (MAS-SJ) that targets mainly the wireless Smart Grid network (WSGN). Because WSGN is important for monitoring the Smart Grid along with the PMUs, which play a key component by providing time-synchronized data of power system operating states [47], attacks like MAS-SJ can disturb the operation of the system or even make it unavailable [10].

4.2.3. Scanning attacks

Scanning attacks aim at discovering all the systems and network protocols alive in the Smart Grid network. Obtaining such information will provide the attacker with valuable insight into the network topology and the deployed system, enabling them to launch a customized and efficient attack. Modbus and DNP3 are two industrial protocols susceptible to scanning attacks. As Modbus/TCP was designed for communication and not for security purposes, it can be compromised by an attack known as Modbus network scanning [48]. In this attack, a benign message is sent to all devices connected to the network, and information is gathered about them [48]. Modscan is a SCADA Modbus network scanner designed to detect open Modbus/TCP and identify device slave IDs and their IP addresses [49]. Nicolas R. has proposed an algorithm to scan the DNP3 protocol and discover hosts, specifically slaves, their addresses, and their corresponding master [50].

4.2.4. Man-in-the-middle (MITM) attacks

MITM attacks occur when an attacker inserts themselves between two legitimate devices and listens for, performs an injection, or intercepts the traffic between them. Upon connecting to these devices, the attacker retransmits the traffic between them; the two legitimate devices appear to communicate directly, however they are communicating through a third device [10], [19]. For example, an attacker could conduct a MITM, by placing himself on an Ethernet network to intercept the exchanged I/O values to the HMI and PLC. The MITM could also be used to intercept TCP/IP communications between a SCADA server and a substation gateway [10], [19]. In [51], the authors who the impact of MITM attacks on SCADA communication integrity. Additionally, authors in [52] emphasized the vulnerability of DNP3, a protocol used in SCADA, and experimented with

MITM attacks with two scenarios that demonstrated the possibility of intercepting messages exchanged between the master station and the outstations, modifying their content, and injecting them into the network.

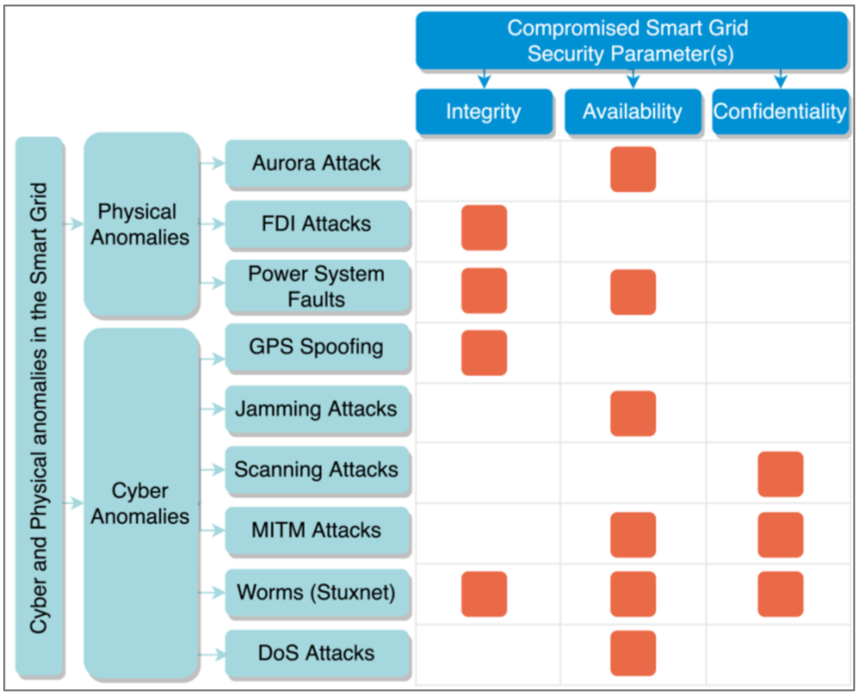
4.2.5. Denial of Service (DoS) attack

DoS attacks aim to adversely affect the availability of a system. In the context of a Smart Grid, DoS attacks are typically used to prevent the control center from receiving sensor measurements or actuators from receiving control commands using a variety of methods, including jamming the communication channel by transmitting a signal with high transmission power to flood the targeted channel and eventually block it or violating network protocols to increase packet collisions. While the system operator can detect the attack due to the loss of measurement data, they are unable to stop it due to the inability to send control signals to the actuators [53], [54]. Authors in [7] proposed a DoS-based attack known as the puppet attack that targets the AMI network by exploiting a vulnerability in the dynamic source routing protocol, then exhausting the communication network bandwidth, causing a drop of packet delivery of 10% to 20%. The time -delay-switch attack [55] is another DoS attack that introduces a delay in the control system leading to instability in the system.

4.2.6. Virus and warms

Viruses are programs that are used to infect a specific device or system. Worms are self-replicating programs that take advantage of a network to spread, replicate, and infect other devices and systems [10], [19]. Trojan horses are programs that appear to carry out legitimate functions on the target system, but run malicious code in the background [10], [19]. In June 2010, Roel Schouwenberg, an analyst at Kaspersky Lab, detected Stuxnet, the first worm that attacks SCADA systems [56]. Stuxnet, a worm of 500 KB, exploited many zero-day vulnerabilities, which were not yet disclosed by the software owner. It infected at least 14 industrial sites based in Iran, including a uranium-enrichment plant. More than one year later, two more worms that targeted industrial control systems were discovered, Duqu and Flame. Unlike Stuxnet, Duqu was designed to gather and steal information about industrial control systems. Flame, on the other hand, was created to be used in cyber espionage in industrial networks. It has been found in Iran and other Middle East countries [19], [28].

The cyber and physical anomalies described above target different parts of the Smart Grid system and attempt to compromise their confidentiality, integrity, and availability to varying degrees. DoS attacks, for instance, target primarily the availability of the system, while FDI attacks compromise the integrity, as well as possibly the availability. Figure 7 illustrates the various anomalies in Smart Grid and the corresponding compromised security parameters.



			Compromised Smart Grid Security Parameter(s)		
			Integrity	Availability	Confidentiality
Cyber and Physical anomalies in the Smart Grid	Physical Anomalies	Aurora Attack		🟥	
		FDI Attacks	🟥		
		Power System Faults	🟥	🟥	
	Cyber Anomalies	GPS Spoofing	🟥		
		Jamming Attacks		🟥	
		Scanning Attacks			🟥
		MITM Attacks		🟥	🟥
		Worms (Stuxnet)	🟥	🟥	🟥
		DoS Attacks		🟥	

Figure 7. Cyber and Physical anomalies impact on the Smart Grid Security Parameters

5. Anomaly detection approaches

Several approaches for detecting cyber and physical anomalies have been reported in the literature [57]- [58]. Some authors have proposed conventional approaches, such as impedance-based and traveling wave techniques, while others have proposed machine learning-based approaches, which can be categorized into supervised, unsupervised, semi-supervised, and reinforcement learning. Other authors have proposed hybrid approaches comprising conventional and machine learning approaches. In this section, we will examine all these various approaches and analyze their advantages and limitations.

5.1. Conventional approaches

Some existing conventional approaches rely on the traveling wave [57] and impedance-based methods [35],[59] for detecting anomalies in the grid. The traveling wave approach needs high-speed data acquisition equipment, a GPS, sensors, and a transient fault recorder to detect the transient waveform for fault location. The location of the fault is computed by tracking" time-tagging the arrival of the traveling wave at each end of the line and comparing against the time difference to the total propagation time of the line with the help of GPS" [57]. This approach has several advantages, as the approach is not impacted by excessive resistance, load variance, reflection, grounding resistance, refraction of the traveling waves, or series capacitor bank [57]. However, the accuracy of the approach relies on capacitance and line inductance. Unlike the time wave method, the impedance-based approaches [35],[59] are simple and easy to implement, as they require only measurement data that include fault voltages and fault currents collected from the digital fault recorder or relays to compute the impedance. The accuracy of this approach can be affected in the case of a grounded fault, where the fault resistances can reach higher values.

5.2. Machine learning-based approaches

The machine learning techniques used to detect anomalies in the Smart Grid are categorized as supervised, unsupervised, semi-supervised, and reinforcement learning (RL) [60]-[61]. Supervised learning algorithms build regression or classification models from a set of input features and their corresponding outputs. Although there is a wide range of supervised models, including logistic regression and neural networks as well, they usually fall into one of the following categories: Decision Tree models, Function models, Bayesian models, and Ensemble models [62]. Unsupervised learning is the process of discovering hidden patterns and associations within unlabeled data. The most common unsupervised learning algorithm is clustering [63]. Semi-supervised learning refers to a combination of supervised learning and unsupervised learning, and it is typically used in cases where training data is insufficient [64]. RL refers to the process of learning directed towards a specific objective in which an agent learns by interacting with an unknown environment, typically in a try-and-error way. In this process, the agent receives feedback from the environment in the form of a reward (or punishment); it then uses this feedback to train itself, as well as to acquire experience and knowledge about the environment [65].

5.3. Supervised learning

5.3.1. Function-based models

Several authors have proposed supervised function-based models for detecting anomalies in the Smart Grid [60]-[66]. Most of these approaches were trained on data including input features such as voltage, current, phase angle, and fault location as output. For example, authors in [60] proposed a back propagation-based neural network (BPN) to estimate fault location in distribution networks. Here, fault current was selected as a key feature to train the NN model. A Levenberg-Marquardt algorithm (also known as damped lease square) is applied to BPN for faster convergence. Then, the BNN model was deployed to run on the DIgSILENT Power Factory 13.2. Similarly, a feed-forward NN (FNN) based approach is proposed in [67]. Here, fault voltages and fault currents are selected as two features to train the model. A sigmoid activation function was used to normalize the data. Their results showed a detection error of less than 3%. Another NN-based approach was proposed in [68] to estimate fault distances from substation(s). The selected input features include: three-phase voltage, current, fault conditions, and active power gathered from substation(s). This approach was trained on different fault locations, resistances, and loads.

The approach was tested on an IEEE 34-bus system and yielded promising results, even under dynamic changes in network topology. Additionally, this approach showed more tolerance to noise. Although high accuracy was reported for NN based approaches in the above-mentioned studies, the training time required for NN is longer does not suit for dynamic or real- time environments. On the contrary, the SVM based approach is faster and relatively accurate, even for larger size data. However, it requires careful selection of appropriate kernel type and hyper-parameters.

In [69], the authors proposed a convolutional neural network (CNN) based approach using bus voltages. This method has been trained and tested on IEEE 39-bus and IEEE 68-bus systems under uncertain conditions for system observability and measurement quality. Their results show that CNN can localize the faulted line even in low visibility (7% of buses) conditions. Another Recurrent Neural Networks (RNN) has been proposed in [70] to deal with the electricity stealth issue. Specifically, a deep autoencoder has been coupled with a long-short-term-memory (LSTM)-based sequence-to-sequence (seq2seq) structure to capture false data injected in the AMI network.

A KNN based approach for detecting faults in a photovoltaic (PV) system is proposed in [71]. This approach has been trained and tested on data generated from a developed PV model. The reported results show a classification accuracy of 98.70% with an error value ranging between 0.61% and 6.5%. Another KNN based approach algorithm was applied to classify three-phase faults (3LG), voltage oscillation, and voltage sag scenarios in [37]. However, the model accuracy was not provided.

5.3.2. Decision tree-based models

Several decision tree-based approaches have been proposed to deal with streaming data and detect the eventual anomalies. For instance, authors [72] proposed a Hoeffding Tree (HT) combined with two concept drift detectors: drift detection method (DDM) and Adaptive sliding windows (ADWIN) for building a fast decision tree that is adaptable for changes. This model has been trained on a synthetic dataset where different attack scenarios were simulated. The dataset contains normal and anomaly events, physical and cyber events. The physical event includes power system faults while the cyber event includes trip command injection, 1LG Fault replay, and Replay Disabled attacks. The model HT+ADWIN+DDM has been trained and tested on this heterogeneous dataset and reported a classification accuracy greater than 98% for binary classification.

Similarly, authors in [73] have proposed a Hoeffding Adaptive Tree (HAT) based approach for detecting events from continuous streams of PMU data within computational boundaries of memory and processing time. The authors have conducted three experiments to generate a training dataset of binary classes. In the first one, synchrophasor data with three-phase faults has been generated and load fluctuations have been injected by changing the True Power (P) and Reactive Power (Q) at a regular interval to evaluate the ability of the HAT model in adapting to concept drift. In the second one, two classes of events have been generated: fault and normal classes. Fault class includes the SLG faults while the normal class includes normal power system operations without changing the load conditions. In the third experiment, non-adaptive HT with fixed size has been evaluated on the previously generated dataset to illustrate the impact of the tree size on the classification accuracy. Based on the reported results, HT showed the concept drift compared to traditional Decision Trees (DTs) J48 and REPTree. However, in these two studies, the duration of the physical fault events was not considered while generating the training synchrophasor dataset. Some faults have to be monitored for a certain time to identify their natures. For instance, a frequency deviation of 0.15 to 1.0 Hz is considered as an inter-area oscillation only if the fluctuation lasts for 60 seconds [74]. Otherwise, it is considered a normal power system operation. In addition, some physical events are time-sensitive and if they are not early detected then they may lead to cascaded outages. Thus, it is necessary to include the event duration as a feature in the training dataset.

5.3.3. Bayesian-based model

Authors in [75] proposed a Bayesian-based approach for detecting false injected data. Here, the authors proposed a dynamic Bayesian game-theoretic approach for detecting FDI attacks with incomplete information with a 98% detection rate. Other authors in [76] proposed a Naïve Bayed model along with RF, DT, and Logistic regression for detecting various cyberattacks in SCADA networks. The reported results showed satisfactory results in terms of detecting attacks in an offline and online network.

5.3.4. Ensemble model

Authors in [77] proposed an ensemble approach based on the Active learning-based extreme gradient Boosting (AL-XGBoost). This approach was developed to detect FDI attacks in a cyber-physical energy system. The average detection accuracy obtained with the AL-XGBoost method is 0.9784 and 0.9845 for IEEE 57- and 118-bus test systems, which is higher than SVM and KNN. Authors in [78] proposed an AdaBoost combined with a Genetic Algorithm (GA) and Deep Neural Network (DNN) to detect electricity theft attacks. The model has shown superiority over SVM, ANN, and RF-based on accuracy, true positive and false-positive rates. In [79] authors discuss a fault line identification and localization approach using RF and decision tree classifiers. Here, the models were trained on an IEEE 68 Bus system. The generated data consists of several fault types including a Three-phase short circuit (TP), Line to ground (LG). The reported experiment results show that classification accuracy of 91%. Another RF-based approach was proposed in [80]. Here, the model was trained on three-phase current and voltage data, validated using an IEEE-34 system, and achieved an accuracy above 90%.

5.4. Unsupervised learning

Authors in [81] proposed an unsupervised learning model for real-time event classification and fault localization in synchrophasor data. Their methodology relies on three processes. The first process focuses on removing bad data from collected PMU measurements using the Maximum Likelihood Estimation (MLE) approach. In the second process, the events are classified using a combination of Density-based spatial clustering of applications with noise (DBSCAN), and logic rules were generated using a physics-based decision tree (PDT) method. This PDT method uses parameters such as active power, reactive power, and fault event types. The third process reports localizing events in real-time using a graph theory. Finally, a score metric is computed using Shannon entropy, and descriptive statistical parameters (e.g., standard deviation, range, mean difference, and crest factor). Three case studies have been considered using metrics such as precision and recall and their reported result show that their proposed data cleansing approach outperforms Chebyshev and K-means methods, with a 95% precision. Additionally, the average run-time taken for their classification algorithm is around 0.09s for a typical window size of 30 samples involving five PMU sensors.

Authors in [61] proposed another unsupervised K-mean-based model for detecting DoS attacks, including UDP and ICMP flooding attacks, against the AMI network. The proposed model was able to cluster the normal and abnormal behaviors using unlabeled data, but the model's performances were not provided.

5.5. Semi-supervised models

Authors in [82] proposed a semi-supervised and deep representation learning for detecting anomalies in WAMS including Short-circuit fault, Remote tripping command injection attack, and Data injection attack.

5.6. Reinforcement learning (RL)

Authors in [83] proposed a multiagent RL-based approach for detecting the simple and coordinated FDI attacks on the distributed control layer in a DC microgrid. Similarly, authors in [84] modeled the anomaly detection problem as a partially observable Markov decision process (POMDP) problem and developed a universal online detection algorithm based on RL for detecting FDI, DoS, and jamming attacks. The model was evaluated in an IEEE 14-bus system and showed a precision above 99%.

5.7. Hybrid approaches

Several hybrid methods combine conventional and ML approaches. For instance, authors in [85] proposed a wavelet transform and Support Vector Machine (SVM) to locate faults in transmission lines and can be described in three stages. In the first stage, voltage and current values emitted by a transmitter were used to locate the fault; the second phase feeds a multi- class SVM model to the training based on selected influential features, and classification of fault location is done using a regression approach. Here, the fault classification error is below 1% for all fault types and specifically 0.26% for SLG, 0.74% for LLG, 0.20% for LL, and 0.39% for LLLG. Authors in [86] have proposed an event location estimation (ELE) algorithm for the wide-area monitoring system for PMU data. Their approach relies on clustering and wavelet analysis to detect and localize events in real-time. In this work, the network is initially divided into several clusters, where each cluster is defined as an electrical zone using K-means. Next, a wavelet-based event detection approach is used to detect and localize event occurrences by tracking any large (e.g. event magnitudes) disturbance levels. Once the event is detected, its magnitude is defined using a Modified Wavelet Energy value, and its location is estimated at each electrical zone's. The authors implemented the ELE approach in real-world PMU-setting containing 32 dynamic events with an excellent localizing accuracy value. It is important to note that the authors did not consider data quality issues in the PMU measurements. Some probable causes for data quality issues could be irregular sampling or data rate, bandwidth challenges, and time synchronization errors.

In [87], the authors discuss a wavelet decomposition technique combined with fuzzy logic to identify both faulty line(s) and their locations in a multi-terminal high voltage direct current network. In their paper, wavelet coefficients of both positive and negative currents were initially computed and then fed to a fuzzy logic-based voting system to identify the faulty line(s). Once the line is identified, a traveling wave-based algorithm is used to determine the exact fault location using the Daubechies wavelets. A discrete wavelet transform (DWT) combined with SVM for fault detection in distribution networks has been proposed in [88]. Here, features are extracted using SVM and DT and then optimized using a GA. Their model performance was evaluated on two active distribution networks (e.g., IEEE 13-Bus and IEEE 34-Bus systems), and the authors claim that their model outperforms the probabilistic neural network (PNN).

Figure 8 and Table 1 provide a summary of these approaches along with their potential advantages and limitations.

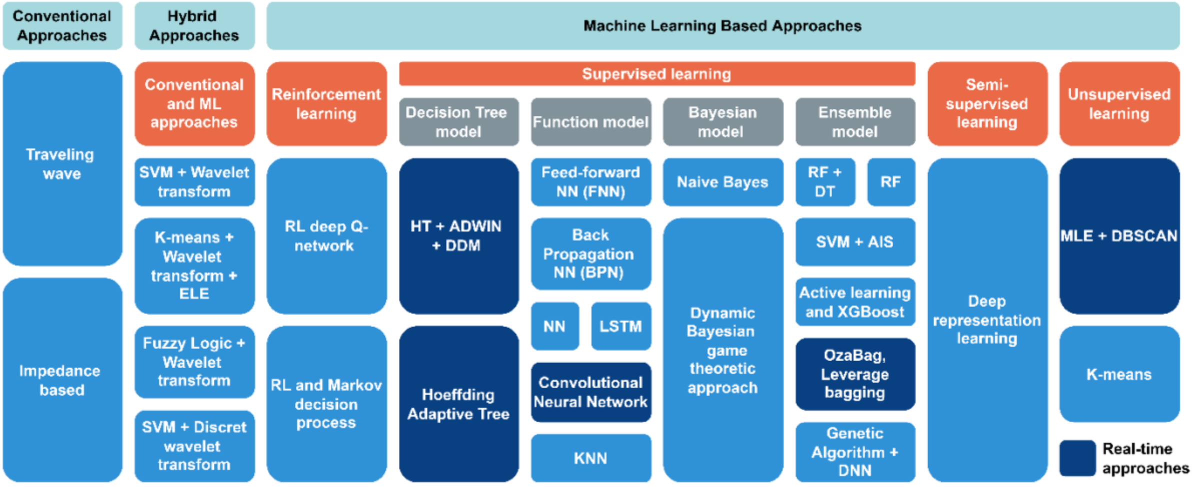


Figure 8. Conventional, ML, and hybrid approaches for detecting anomalies in the Smart Grid system

Table 1. Cyber and physical anomaly detection approaches

Category	Approach	Anomalous event category	Anomalous event type	Targeted Smart Grid domain	Advantages	Limitations
Conventional	Traveling wave [57]	Physical	Single-phase grounding fault	Transmission domain	High tolerance to the excessive resistance, load variance, reflection, grounding resistance, refraction of the traveling wave or series capacitor bank	The accuracy of the model relies heavily on capacitance and line inductance
	Impedance-based [35]		Unbalanced faults	Transmission domain	Simple and easy to implement	The accuracy can be

	Impedance-based [59]		Shunt fault	Distribution domain		affected in the case of a grounded fault, where the fault resistances can reach higher values.
Machine learning	Back propagation-based neural network (BPN)[60]	Physical	1 phase, 2 phase, and 3 phases faults	Distribution domain	The model is fast to converge and requires few features	The selected features are insufficient to capture various anomaly signatures.
	KNN [71]	Physical	Open circuit faults, line-line (LL) faults, partial shading with and with-out bypass diode faults and partial shading with inverted bypass diode faults in real-time	Generation domain	Fault location accuracy reaches 98.70% with an error between 0.61% and 6.5%.	The study considered the PV data only
	Feed-forward NN (FNN) [67]	Physical	Shunt faults	Distribution domain	The detection error is less than 3%. High tolerance to the fault resistance, fault type, and fault location.	The fault location was not considered and they are unsuitable for real-time applications
	NN [68]	Physical	Ground fault, short circuit	Distribution domain	High tolerance to noise.	
	CNN [69]	Physical	TP, LG, double line to ground (DLG) and	Transmission domain	Optimal localization estimation even under low	

Machine learning			line to line (LL) faults		visibility (7% of buses)	
	MLE + DBSCAN [81]	Physical	Three-phase fault, P load increased, Q load increased	Distribution domain	The proposed data cleansing approach outperforms Chevyshev and K-means and achieve a precision of 95%. Less than 0.9 s to classify events for a typical window size of 30 sample data.	
	RF+ DT [79]	Physical	Three-phase fault, line to ground (LG), line-to-line to ground, line-to-line (LL), loss of line	Distribution domain	Fault location detection accuracy is 91%	
	RF [80]	Physical	Single-phase to ground faults	Distribution domain	Fault location detection accuracy is 90.96% in distribution systems	
	HT+ADWIN+DDM [72]	Cyber and physical	Symmetric, unsymmetrical faults, and trip command injection attack	Transmission domain	Classification accuracy is greater than 94% for multiclass and greater than 98% for binary class.	
	HAT [73]	Cyber and physical	Three-phase faults, SLG	Transmission domain	Optimal adaptability to the concept drift events	
	OzaBag and Active classifier	Cyber	DoS, unauthorized	Distribution domain	ActiveClassifier and	

Machine learning	[25]		access from a remote machine, to local super user (root) privileges, surveillance, and probing.		SingleClassDrift algorithms reported satisfactory results in terms of time processing and memory consumption	Fault location and duration were not considered. This model is not suitable for real-time network
	RL deep Q-network [83]	Cyber and physical	FDI attack	Generation and transmission domains	Effective in detecting simple and coordinated FDI attacks	
	SVM+AIS [58]	Cyber	DoS attack and unauthorized access from a remote machine	Distribution and Consumer domains	Tthe detection accuracy is high	Processing time is high and it is not suitable for streaming data. The fault duration was not considered
	POMPD [84]	Cyber and physical	FDI, DoS, and jamming attacks	Transmission domain	Model efficient in detecting various attacks with a 99% precision	Model in unsuitable for real-time deployment.
	LSTM [70]	Cyber and physical	FDI attack and electricity theft	Distribution system	The model outperforms the Naïve Bayes, SVM, and ARIMA in terms of detection rate and false alarm with an improvement of 4−21% and 4−13%, respectively	

Machine learning	NB, RF, DT, LR [76]	Cyber	Scanning attack	Operations domain	Results showed that these models are efficient in detecting SCADA cyberattacks in real-time.	Neither attack location nor duration were considered.
	Dynamic Bayesian game-theoretic approach [75]	Physical	Load measurement	Operation domain	The model is effective in detecting anomalies with incomplete information and with a detection rate of 98%	
	AL-XGBoost [77]	Physical	FDI attack and cascading failure	Transmission domain	The proposed model reaches a detection accuracy of 99% and outperforms SVM and KNN	
Machine learning	Semi-supervised Deep representation learning [82]	Cyber and Physical	Short-circuit fault, remote tripping command injection attack, and FDI attack	Transmission domain	The model outperforms the supervised ML model in terms of the true positive rate	Model suffers from a high false-positive rate compared to supervised ML models
	AdaBoost with GA and DNN [78]	Physical	FDI attack	Distribution domain	Model is superior to SVM, RF, and NN in terms of detecting electrical theft with 94.8% of true positive rate	The attack model was not provided.
	K-means [61]	Cyber	DoS attack	Distribution domain	Model is efficient in handling unlabeled data	Model's performances were not provided.

Hybrid	SVM+ Wavelet transform [85]	Physical	SLG, LLG, LL	Transmission domain	The fault classification error is below 1% for all fault types.	The fault duration was not considered and the model is not suitable for streaming power system data.
	K-mean + Wavelet transform + ELE [86]	Physical	32 dynamic events including generator and line trips	Transmission domain	Fault location accuracy attain 100%	The fault duration was not considered and the processing time is high, which make them inappropriate for streaming data.
	Fuzzy logic + Wavelet transform [87]	Physical	Negative to ground and the positive to ground faults	Transmission domain	The error between the actual fault location and the predicted one is low then 0.002%	
	SVM+ Discrete wavelet transform [88]	Physical	High impedance fault	Distribution domain	Fault location accuracy is 98.27% for IEEE 13-Bus and 98.29% for the IEEE 34-Bus test systems	

6. Conclusions

This chapter laid the groundwork for the following chapters by providing a comprehensive review of the existing cyber and physical anomalies in the Smart Grid and their impact on the different subsystems' security. Additionally, it reviewed in depth the existing detection approaches and analyzed their strengths and limitations. The next chapter will analyze the FDI attack in detail and introduce a machine learning-based detection method.

Chapter III

Detection of the False Data Injection Attack in Home Area Networks using ANN

This chapter focus on developing an Artificial Neural Network model for detecting False Injection Data (FDI) attack in the home area networks. The proposed approach is trained on a dataset containing the electricity demand profiles for 200 households for the Midwest region of the United States. However, this data set does not include any false measurement data. Thus, the first step is falsifying the data by injecting the false data intentionally using several membership functions. These membership functions are carefully selected to mimic the malicious manipulation caused by the FDI attack. Then, the proposed ANN model is trained on the prepared dataset and compared against two other machine learning approaches namely, SVM and RF. This Chapter is organized as follows: Section I describes an FDI attack model. Section III describes the methodology adopted to model the FDI attack using two membership functions. In addition, it explains the ANN-based detection approach with the relevant features that are used for training the model. Section IV presents the various performance metrics used to evaluate the proposed approach and discusses the obtained results. The conclusions and future directions are provided in the last section [2].

[2] This chapter is a slightly modified version of our paper "Detection of the False Data Injection Attack in Home Area Networks using ANN" published in the IEEE International Conference on Electro Information Technology (EIT), 2019, pp. 176-181, doi: 10.1109/EIT.2019.8834036.

1. Introduction

In the previous chapter, we discussed several FDI attack scenarios and their impact on the Smart Grid domains, such as the customer, operations, and market domains. These FDI attacks break into the system by compromising the weakest devices and network protocols, as shown in Figure 9. At the device level, for instance, a smart meter can be used as an attack vector to inject false data or even create breaches in the metering data set. Since the smart meters can disconnect-reconnect remotely and control the user appliance and devices to manage load and demands, a compromised smart meter can forge the demand request, such as requesting a large amount of energy or even misleading the electrical utility about electricity consumption and cost. The home appliances that interact with the service provider or with the AMI can also be maliciously manipulated causing undesirable consequences in the residential areas.

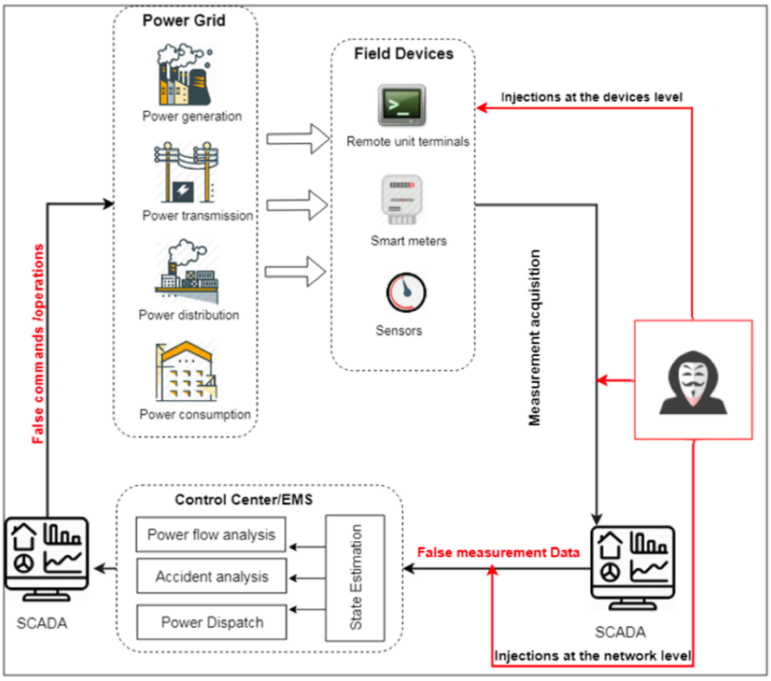


Figure 9. A false data injection attack scenario

At the network level, an attacker can inject malicious traffic into the wireless network through attacks such as the MITM attack. Figure 10 illustrates a MITM attack scenario where an adversary falsifies the actual energy consumption by intercepting the data exchanged between the customer and the utility via the ZigBee protocol. This protocol is widely used for short-rand wireless data transfer, particularly for wireless low-power devices, but it suffers several vulnerabilities [89]. Thus, the attacker can easily compromise this weakest point of the system and then tap on it to launch more advanced attacks from within the network. The attacker can also inject false data in the sensor network in the Grid by tampering, misrepresenting, or forging the sensor's data [90], [91]. These types of FDI, either at the device or at the network level, can cause undesirable consequences such as disturbing the power system state estimation and sending wrong information to the system operators [92]. Moreover, the FDI attack can negatively impact the electricity market by manipulating the real-time electricity price at any given compromised bus [93]. In [94]–[96], the authors demonstrate that the FDI attack can interfere with the electricity market operations as the manipulated state estimation output affects the economic dispatch. Authors in [97], [98] investigate how the FDI attack can be exploited for continuous financial arbitrage, such as virtual bidding at a given set of buses.

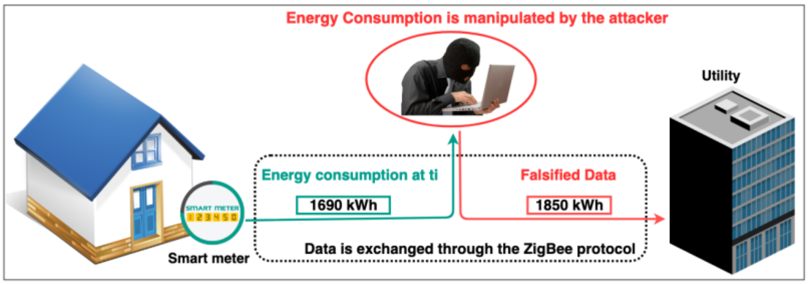

Energy Consumption is manipulated by the attacker
Utility
Energy consumption at ti
1690 kWh
Falsified Data
1850 kWh
SMART METER
123456
Smart meter
Data is exchanged through the ZigBee protocol

Figure 10. An FDI attack scenario at the Smart Grid's network-level

2. Related work

In addition to the anomaly detection approaches discussed in the previous chapter, considerable efforts have been made specifically on the identification and mitigation of FDI attacks in the electric Smart Grid [38], [99]. Some of the proposed approaches are intended to protect the physical part of the grid while others are cyber-oriented approaches. From the physical security perspective, authors in [100], [101] suggested the physical protection of basic measurement devices by locking them in boxes or even replacing them with PMUs. However, the implementation of physical protection schemes of every measurement device in the grid can pose serious challenges in terms of cost and feasibility. On the other hand, an alternative and optimal solution have been proposed in [102], where a graphical method is used to locally protect only the vulnerable components. However, the question was how to identify those critical and vulnerable components in the grid? Authors in [103] have answered this question by proposing a contraction factor particle swarm optimization-based hybrid cluster technique to rank and classify each component in the grid based on its vulnerability, from most to least vulnerable, then to protect the weakest one. Authors in [11] and [99] propose cyber-security approaches as supplementary to the physical security countermeasures. Authors in [11] proposed SVM and a statistical-based anomaly detection method to detect the FDI attack. Authors in [99] suggest a deep learning algorithm to recognize the features of the FDI attack using the historical measurement data, and these features are then used in detecting the FDI attacks in real-time.

3. Methodology

3.1. FDI attack model

Training and testing a supervised machine learning model require an appropriate dataset with the relevant features. To the best of the authors' knowledge, no publicly available dataset includes the FDI attack that can be used to train the proposed classifier. Thus, the false measurements are deliberately introduced in a legitimate dataset to replicate an FDI attack scenario. This dataset contains the electricity demand profiles for 200 households available in the 2009 Residential Energy Consumption Survey dataset for the Midwest region of the United States for one year. The profiles have been generated using the approach proposed by Muratori et al. [104], which produces realistic patterns of residential power consumption, then validated using metered data with a resolution of 10 minutes. Households vary in size and number of occupants and the profiles represent total energy consumption. The dataset consists of four main features: date, time, energy demand, and the cost per kWh.

Generally, the energy consumption of a household is the simple aggregation of all the individual appliances and plugin loads, thus each household has a varying load demand throughout the day. An FDI attack aims at introducing bias in some measurements, such as frequency, voltage, or energy consumption, resulting in a change of load pattern of a household. Figures 11 and 12 illustrate a normal energy consumption behavior and a falsified one using an FDI attack. By taking a reverse approach, one can simulate the FDI attack by simply changing the distribution of some measurements over time through the injection of false values. These falsifications are not injected randomly, but they follow specific patterns that reflect the FDI attacker behavior and they can be modeled using the membership functions. In this study, two FDI attack scenarios are considered.

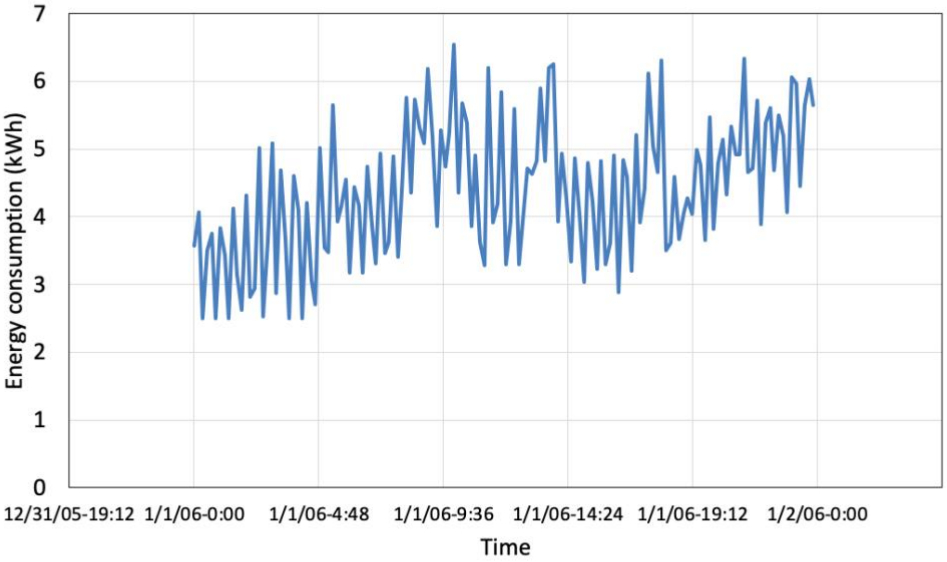


Figure 11. A normal energy consumption behavior for a given household

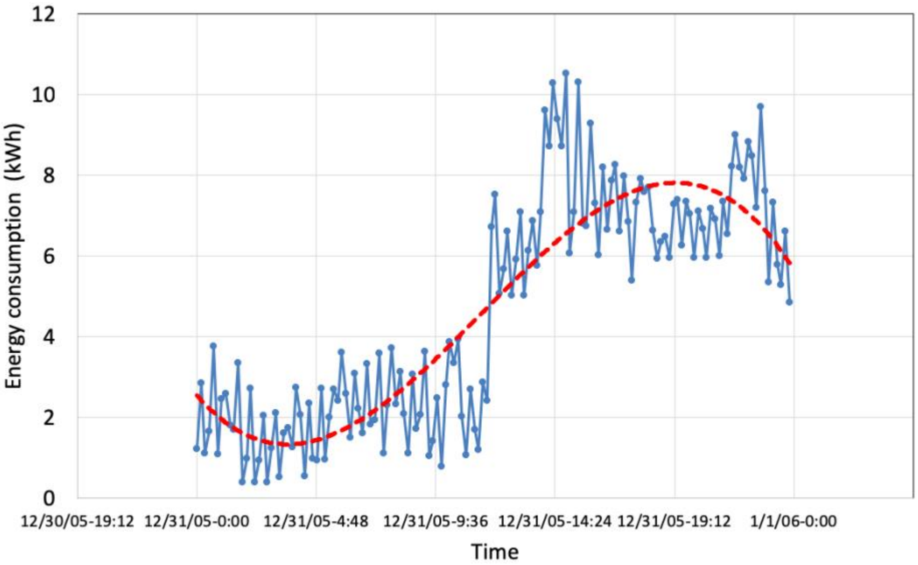


Figure 12. A falsified energy consumption via an FDI attack

- **Scenario 1: Increasing energy consumption during peak hours**

During the peak hours, the use of electricity is typically the highest from Monday through Friday between 8 a.m. and 10 p.m.; usually, the average price of the electricity usage during the peak hours is twice as that of the off-peak hours. For instance, the average price of electricity in California is about 10 cents/kWh during the off-peak hours and 20 cents/kWh during peak hours [34]. In this scenario, it is assumed that an adversary seeks to increase the energy consumption of a compromised smart meter during peak hours to increase drastically the electricity bills of the targeted user. Usually, communication within the AMI network is defined by the Z-wave and Zigbee protocols. These protocols can be exploited by several attacks including the MITM attack and replay attack [11], [99]. It should be noted that proposing an attack model to compromise the AMI network is out of the scope of this paper and it is planned as future work. To model this behavior, the Trapezoidal membership function is used to falsify the legitimate data. This function is expressed as [105]:

$$u(x,\alpha,\beta,\gamma,\delta) = \begin{cases} 0, & x < \alpha \\ \dfrac{x-\alpha}{\beta-\alpha}, & \alpha \leq x \leq \beta \\ 1, & \beta < \alpha \leq \gamma \\ \dfrac{\gamma - x}{\gamma - y}, & \gamma < x \leq \delta \\ 0, & x > \delta \end{cases} \tag{1}$$

Here x is the time variable and [β, γ] is the peak electricity usage time interval. Figure 13 depicts the redistribution of the energy demand following the Trapezoidal function.

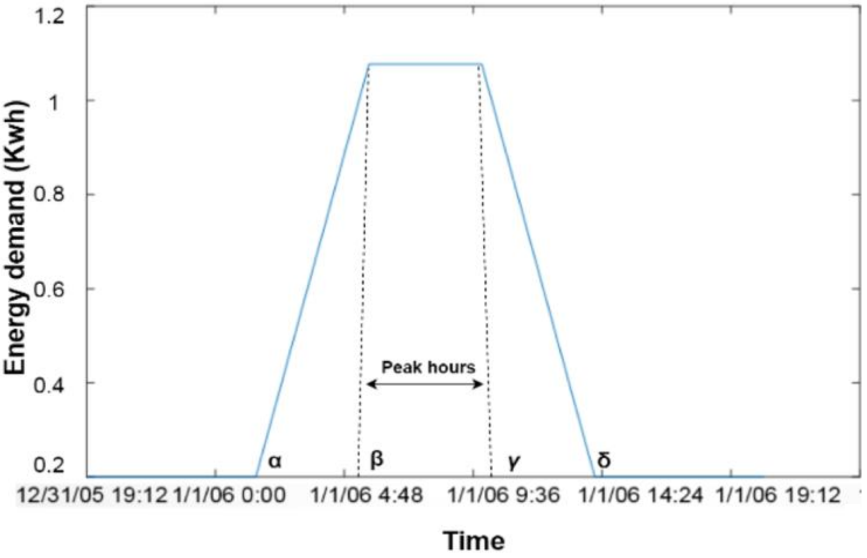


Figure 13. Scenario 1: Increasing the energy demand consumption during peak hours.

- **Scenario 2: Increasing the energy demand during the off-peak hour for a long period of time**

In this case, it is assumed that the attacker performs an abnormal load increase in energy consumption for several hours during the off-peak hour to affect the electricity bill of a targeted user. Such a behavior can be modeled using the Sigmoid function which is expressed as [106]:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2}$$

Here x represents the energy demand value. Figure 14 illustrates how the energy consumption during the off-peak hour is manipulated using the sigmoid function.

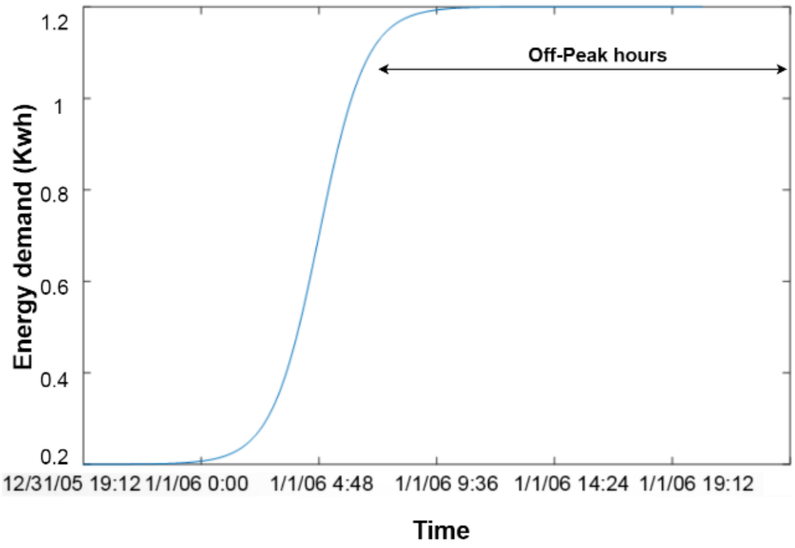


Figure 14. Scenario 2: Increasing the energy demand consumption during the off-peak hours.

3.2. Artificial Neural Network model

Once the falsified data are injected into the data set, the next step consists of preparing the data set for training the classifier. As illustrated by Figure 15, this step includes labeling the data, normalizing the values of the relevant features (date, time, energy demand, and the cost per KWh) in addition to the removal of the redundant records. Before feeding the dataset to the supervised classifier, it is essential to label the data. The original dataset has one class corresponding to the normal energy consumption (normal event). After injecting the false measurements, the dataset includes an additional class corresponding to the falsified data (FDI event). In data normalization, the values of the features are standardized and aligned to decrease the convergence time of the classifier. In addition, the redundant data are eliminated to reduce the bias toward one class and at the same time increase the detection accuracy of the classifier. The relevant features that are selected to detect the FDI attacks are the energy demand, corresponding cost, and time. The labeled, normalized, and standardized data set is split into training (70%) and testing (30%) data.

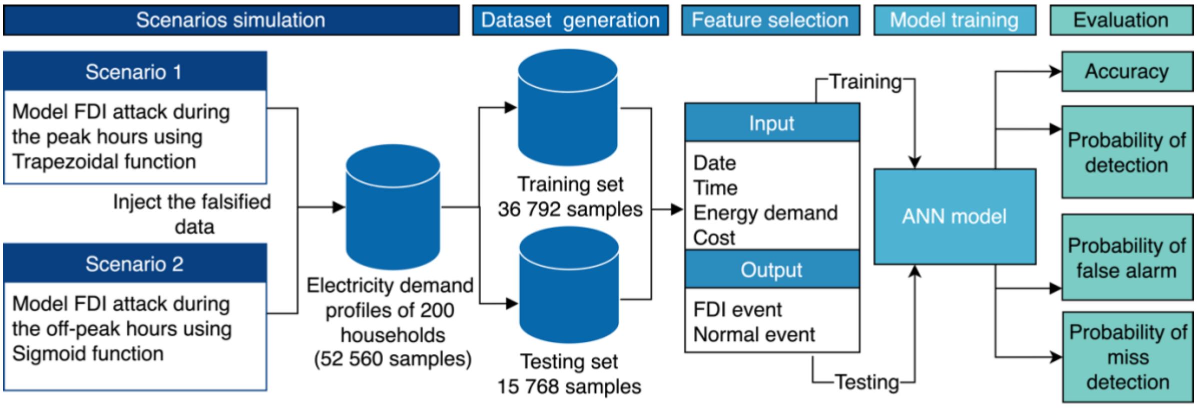

Scenarios simulation
Dataset generation
Feature selection
Model training
Evaluation
Scenario 1
Model FDI attack during the peak hours using Trapezoidal function
Inject the falsified data
Scenario 2
Model FDI attack during the off-peak hours using Sigmoid function
Electricity demand profiles of 200 households (52 560 samples)
Training set 36 792 samples
Testing set 15 768 samples
Training
Input
Date
Time
Energy demand
Cost
Output
FDI event
Normal event
Testing
ANN model
Accuracy
Probability of detection
Probability of false alarm
Probability of miss detection

Figure 15. Conceptual diagram of the proposed approach

In this study, ANN is used as a classifier and it is a supervised machine learning algorithm used for classification and regression prediction. The two terms ANN and NN are used interchangeably throughout the dissertation. ANN is composed of an input layer, two hidden layers, and the output layer where each layer is composed of several neurons. A neuron is a computation unit that takes a set of inputs $x_1, x_2, \ldots., x_n$ where each input is associated with weight, $w_1, w_2, \ldots., w_n$ and predicts the output using a non-linear activation function.

A typical neuron takes several inputs $x_1, x_2, \ldots., x_n$ each of which is multiplied by a given weight, $w_1, w_2, \ldots., w_n$. These inputs are multiplied by their corresponding weights and summed together to pass them through a non-linear activation function. The equation of a given node is expressed as:

$$z = f(b + x * w)f(b + \sum_{i=0}^{n} w_i x_i) \quad (3)$$

Where b is the bias term that allows to shift the results of the activation function to the left or right and to train the model when the input features are 0. In this study, the inputs $x_1, x_2, \ldots., x_n$ are the features which represent each event in the dataset. In particular, these features are date, time, energy consumption, and cost. These inputs are fed into an activation function. In this study, three activation functions are investigated with the ANN: Sigmoid function, hyperbolic tangent (Tanh) function, and rectified linear unit (Relu) function [107]. For the Sigmoid function, the output is bounded by 0 and 1. When the input is very large, the output is approximately 1, and when the input is very small, the output is approximately 0. The values between these two extremes are shaped like an S-curve, and it is expressed as [107]:

$$f(z) = \frac{1}{1 + e^{-z}} \tag{4}$$

Where z is the input value of a given feature.

In contrast to the sigmoid function, the Tanh function ranges between -1 and 1, and as such is preferred over the sigmoid function since it is zero-centered; it is expressed as [107]:

$$f(z) = \tanh\,(0, z) \tag{5}$$

Where z is the input value of a given feature.

The Relu function is a non-linear function expressed as follows [107]:

$$f(z) = \max\,(0, z) \tag{6}$$

Where z is the input value of a given feature.

In this research, the three types of activation functions are considered to train and test the proposed model. Training the model involves two main steps: forward propagation and backpropagation. The forward propagation process entails weighing the training samples and passing them through the activation function to compute a predicted output for each node; this output is compared with the actual one to measure the error using a loss function. Various loss functions can be adopted, including the mean absolute error, mean square error, mean bias error, and Cross-Entropy loss function. Given that we are dealing with a classification issue, we choose the Cross-Entropy loss function which can be expressed as follows [108]:

$$CrossEntropyLoss = -(y_i \log(y'_i) + (1 - y_i)\log(1 - y'_i)) \qquad (7)$$

Where the y_i is the actual output and y'_i is the computed one.

Following the computation of the loss function, backpropagation is used to propagate the error to all nodes in the network and reduce the error by updating the weights using gradient descent optimization algorithms. There are many types of optimizers, including adaptive moment estimation (Adam), Adadelta, nesterov accelerated gradient, Adagrad, and RMSprop. According to [109], Adam is the most optimal algorithm compared to other gradient descent optimization algorithms. As a result, Adam is selected as an optimizer in this study.

The last layer includes the softmax function to compute the probability of distribution over a set of mutually exclusive labels, 0 for a normal event and 1 for an FDI event, with a certain level of confidence. It is given by:

$$y_i = \frac{e^{Z_i}}{\sum_{j=0}^{k} e^{Z_j}} \qquad (8)$$

Where Z_i is the activation function of a neuron i and k is the total number of hidden neurons. After training the ANN, a portion of the training data set is used to validate the model. The trained model is tested against 30% of the dataset and each record is classified either as a normal event or an FDI attack event.

4. Results and discussion

The results obtained from the trained and tested ANN model are shown in Figures 16 and 17. The optimal ANN model is then compared against two other classifiers: SVM and

RF. To increase the detection accuracy while avoiding the overfitting and under-fitting issues, a parametric study is conducted individually on these algorithms. For instance, three types of kernels are selected for SVM: Radial Basis Function (RBF), Sigmoid, and polynomial functions. In RF, the number of trees used is 10 and 100. A comparison between these algorithms is carried out based on several performance metrics including the accuracy, probability of detection (Pd), the probability of miss detection (Pmd), and the probability of false alarm (Pfa) [37], [38], and the results are summarized in Table 2.

The accuracy corresponds to the number of times a normal or an FDI event is correctly classified among the total number of events in the dataset and it is expressed as:

$$Accuracy = \frac{Number\ of classified\ events}{Total\ number\ of\ events} \tag{9}$$

The probability of detection, Pd, is defined as the number of times an FDI event is correctly classified against the total number of events and it is given by:

$$Pd = \frac{Number\ of\ classified\ FDI\ events}{Total\ number\ of\ FDI\ events} \tag{10}$$

The probability of miss detection, Pmd, is expressed as the number of times an FDI event is incorrectly classified as a normal event against the total number of FDI events and it is given by:

$$Pmd = \frac{Number\ of\ miss\ detected\ FDI\ events}{Total\ number\ of\ FDI\ events} \tag{11}$$

The probability of false alarm, Pfa, corresponds to the number of times a normal event is wrongly classified as an FDI event among the total number of normal event signals and it is expressed as:

$$Pfa = \frac{Number\ of\ false\ detected\ FDI\ events}{Total\ number\ of\ normal\ \ events} \tag{12}$$

Figure 16 shows the accuracy of the ANN with Sigmoid, Tanh, and Relu functions versus the number of instances events. As it can be seen, the ANN with the Relu function reports higher accuracy with more than 25,000 instances compared to Tanh and Sigmoid function. The accuracy of NN with Relu function changes slightly between 98.4% and 98.5% between 5,000 and 18,000 instances and then it increases and stabilizes at 99% with more than 23,000 instances. Tanh function's accuracy ranges between 98.3% and 98.6% for more than 5,000 instances and less than 18,000, then it increases and stabilizes at 99% between 22,000 and 43,000 instances before it drops to 98.85% with more than 45,000

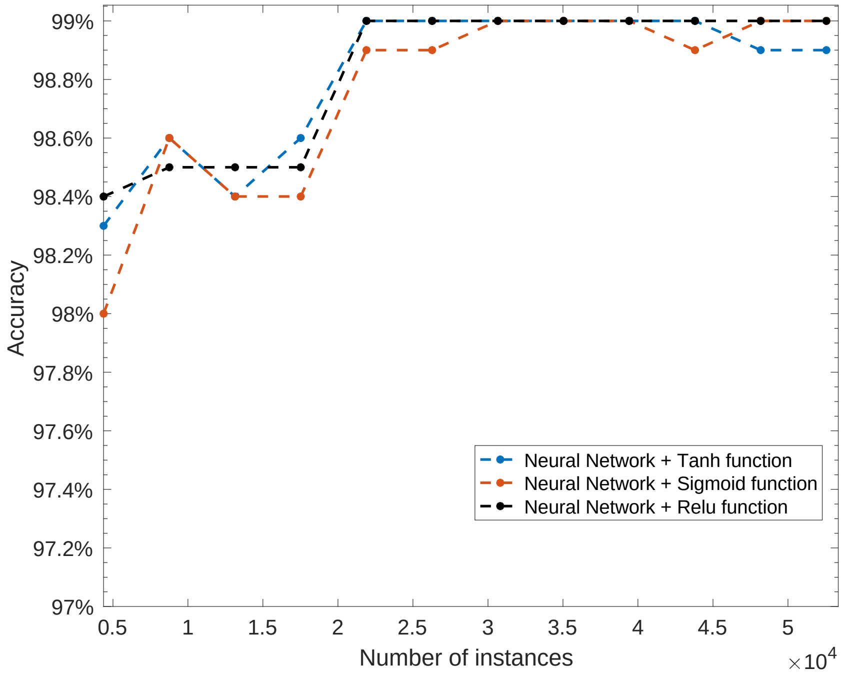


Figure 16. The accuracy of NN with three activation functions: Relu, Sigmoid, and Tanh function, as a function of the number of instances.

instances. ANN with Sigmoid function has the lowest accuracy value, which equals 98% with 5,000 instances, and its higher accuracy, which is 99%, for a number of instances between 31,000 and 39,000. On average, Relu and Tanh functions report almost the same accuracy, which is 98.8%, followed by Sigmoid function with 98.7% accuracy. Although the 0.1% difference seems a very small number, it constitutes more than 52 instances in the dataset.

Figure 17 shows the Pfa of NN with Relu, Tanh, and Sigmoid as a function of the number of instances. As one can see, the Pfa decreases as the number of instances decreases. Relu and Tanh function reports the lowest Pfa value, which is equal to 0.9%, with 21, 000 instances, while Sigmoid function reaches the same Pfa value with 31 000 instances. The highest Pfa value is reported by the Sigmoid function with 4,830 instances. On average, the Relu and Tanh functions have the lowest Pfa value, which is equal to 1.16%, followed by the Sigmoid function with 1.24%.

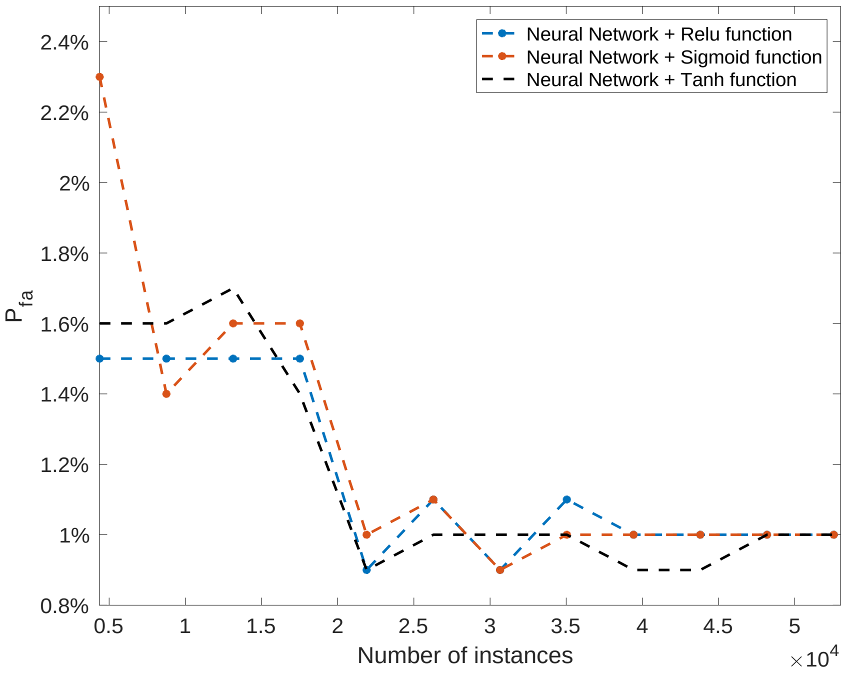


Figure 17. Probability of false alarm of NN with three activation functions: Relu, Sigmoid, and Tanh function, as a function of the number of instances.

The results obtained from these simulations suggest that the NN with the Relu activation function exhibits better performance as compared to NN with Sigmoid and Tanh functions. Although the Relu function is outperformed by the other activation function, it shows more stability. Thus, it can be considered as the optimal activation function for NN compared to Sigmoid and Tanh in terms of accuracy and Pfa. Table 2 illustrates a comparison between NN with the Relu activation function, RF and SVM. As it can be seen, the NN reports the higher accuracy, which is 99%, followed by RF (100 trees) with 94.3%, RF (10 trees) with 92.8%, and then SVM-RBF with 86%. The lowest accuracy rate is reported by SVM with the polynomial kernel. In terms of probability of detection, the NN reports the highest accuracy, which is equal to 99.4% followed by RF (100 trees) with 88.2%, RF (10 trees) with 85.9%, and then SVM-Sigmoid with 80.5%. In terms of probability of miss detection, the NN reports the lowest value, which is 0.6%, followed by RF (100 trees) with 11.8%, RF (10 trees) with 14.1%, and then SVM-Sigmoid with 19.5%. In terms of the Pfa, RF (100 trees) reports the lowest value, which is 0.2%, followed by NN with 0.9%, RF (10 trees) with 1.1%, and then SVM-RBF with 1.8%. These results suggest that the NN with the Relu activation function is an optimal classifier to detect FDI attack as compared to that of SVM and RF in terms of accuracy, probability of detection, and probability of miss detection. However, RF with 100 trees outperforms the NN in terms of Pfa with less than 0.7% which is about 367 instances in the data set. These false alarms can impact network performance by consuming additional bandwidth and thus can overloading the network.

Table 2. Comparison between the ANN, SVM, and RF in terms of accuracy, Pd, Pfa, and Pmd.

Algorithm	Accuracy	Pd	Pmd	Pfa
SVM- RBF	86%	72.7%	27.3%	1.8%
SVM- Sigmoid	84.3%	80.5%	19.5%	12.3%
SVM- Polynomial	82.9%	66.9%	33.1%	2.7%
ANN- Relu (100 neuron nodes)	99%	99.4%	0.6%	0,9%
RF (10 trees)	92.8%	85.9%	14.1%	1.1%
RF (100 trees)	94.3%	88.2%	11.8%	0.2%

5. Conclusions

In this chapter, an ANN-based approach is implemented to detect the FDI attack in Home Area Networks. Two attack scenarios are considered to model the FDI attack using Sigmoid and Trapezoidal membership functions. The falsified dataset is fed to ANN for training and testing the model. During the training phase, three activation functions are investigated: Relu, Sigmoid, and Tanh functions. The results are evaluated based on several performance metrics and compared against SVM and RF methods. From the simulation results, it is observed that ANN with the Relu activation function and 100 neuron nodes detects the falsified injected data with an accuracy of 99%. In addition, it outperforms the RF and SVM in terms of probability of detection, and probability of miss detection. However, the RF with 100 trees exhibits a low Pfa, which is 0.2% followed by ANN with 0.9%. In the next chapter, we will improve the proposed detection approach for detecting the anomalies' location and predict their duration.

Chapter IV

Random Forest Regressor-Based Approach for Detecting Fault Location and Duration in Power Systems

This chapter is an extension of Chapter III. Specifically, the focus here is on a random forest regressor (RFR) based model to detect anomaly locations and predict their duration simultaneously. From a machine learning perspective, fault location detection is usually approached as a multiclass classification problem where the output would be a class label, fault position; while fault duration prediction is regarded as a regression problem as the output would be a continuous value, fault duration. GridPACK framework [110] was used to train the model by simulating several three-phase fault scenarios on a nine-bus system to generate appropriate datasets. A collection of four experiments are formulated to evaluate the performance of the RFR model. This chapter is organized into the following sections: Section 2 focuses on RFR model description, with details on simulated fault scenarios, feature selections, and training/testing process; Section 3 discusses the analysis of four experiment scenarios for classifying and predicting fault location and duration with off-line/streaming conditions, and Section 4 draws conclusions and recommendations for future work[3].

[3] This chapter is a slightly modified version of our paper "Random Forest Regressor-Based Approach for Detecting Fault Location and Duration in Power Systems" published in Sensors 2022, 22, 458. https://doi.org/10.3390/s22020458

1. Introduction & Related work

As we discussed in Chapter II, anomaly identification is critical for seamless Smart Grid operations and utilities are working around the clock to reduce outage rates from interruptions such as contact with natural vegetation, animals, or weather events [111]–[113]. The unplanned outages can lead to long service interruptions and a significant economic impact on the customers. The cost to various consumers for a one-hour outage during a summer afternoon was estimated to be approximately USD 3 for a typical customer, USD 1200 for small and medium organizations, and USD 82,000 for large organizations [114]. These outage costs increased substantially depending on the time of year and outage duration, especially when they occur during winter. Thus, predicting faults in the system along with their duration is the first step towards reducing the number of unplanned outages and providing a prediction-based plan to the utility for deploying the appropriate maintenance crews and the sequence of operations [115], [116].

Although Many power system fault detection approaches have been reported in peer-reviewed [57]- [58], relatively few works have been carried out to predict fault duration. However, this is arguably pertinent information from the customer's perspective. When a fault or an outage occurs and consumers ask when the power will be restored, utilities have to provide an accurate estimation of the recovery time. Seattle City Light provides a real-time outage map with an estimated restoration time; however, the difference between the actual outage duration and the estimation time is large, possibly because of the conventional techniques used [115], [117]. There are few approaches in the literature that have attempted to address this issue. Authors in [115] proposed a real-time approach for detecting outages in distributed systems. This approach was based on recurrent neural network (RNN) and was trained on three sources of historical data: outage report provided by Seattle City Light and 15 years of data, repair logs and weather information. Another approach for predicting faults duration in transmission systems was proposed in [116]. This approach was based on Naive Bayes (NB) classifier and SVM and it was trained on non-temporary fault-type data including features such as substation, asset type, fault category, and outage start time. The reported results indicated an accuracy above 97%.

To bridge this gap, we propose an approach that maps first the location and duration of the fault into one single value and then applies an RFR-based model for detecting fault locations and predicts their duration simultaneously. Additionally, the proposed model is capable of predicting various faults' duration including short, medium, and large. It is adaptable to other case scenarios and power system datasets as it includes an ensemble of multiple uncorrelated trees that achieves strong generalizations; It predicts various fault duration including short, medium, and large duration; and It is convenient for real-time applications as it requires less processing time compared to the existing approaches. GridPACK framework [110] was used to train the model by simulating several three-phase fault scenarios on a nine-bus system to generate appropriate datasets. A collection of four experiments are formulated to evaluate the performance of the RFR model. The model was evaluated in experiment 1 for fault detection accuracy, then compared to seven classifiers: ANN, DNN, SVM, KNN, NB, DT, and HT. The RFR model was evaluated in experiment 2 for predicting fault duration, then compared to the regression version of models such as support vector regressor and decision tree regressor. Mean squared error (MSE) and mean absolute error (MAE) were used as evaluation metrics. In experiment 3, the RFR was examined in terms of handling missing data possibly caused by equipment failure, data storage issues, or unreliable communication. The RFR was tested in a streaming data environment in experiment 4, where multiple window sizes were considered. The MSE and processing time for the RFR were then compared to HT and DNN. The HT and DNN models are commonly suggested for power system streaming data [99], [118].

2. Methodology

2.1. Random Forest Regressor (RFR) Model

Random forest F is an ensemble approach with several independent and uncorrelated DT $F = t_1, t_2, \ldots., t_t$. These uncorrelated trees assist model F in achieving an accurate generalization by injecting randomness into the DT. F is a bagging approach that combines several high variance and low bias trees to create low bias and low variance models using bootstrapping and aggregation techniques [119], [120]. Consider a training set $\mathrm{S} = \{X^m, Y^m\}_{(M=1)}^{m}$, where $\mathrm{X} \subset R^D$and consists of input feature space with parameters such as voltage (v), phase angle (φ), current (i), and frequency (f). Y is a multidimensional continuous space $Y \subset R^{D'}$, and includes both the fault location and corresponding fault duration. M is the number of samples, and bootstrap is a subset S_{t} of the entire training set S, where each instance has been randomly sampled using a uniform distribution with or without replacement. The resulting bootstrap data includes the same number of instances as the original data set S; however, approximately 1/3 of these samples are duplicated and approximately 1/3 of the instances are removed from the bootstrap sample. Multiple passes are performed on the input data to create bootstraps for each tree. Once the training and testing are completed on the bootstrap data, the prediction of all the independent trees is averaged as one aggregated value. By distributing the training set across multiple trees and training each one on a subset of the training data, the model generated does not overfit; and since the model is built using low bias trees, the underfitting problem is also avoided.

Assuming that output variables follow a multivariate Gaussian distribution with mean μ and covariance Σ, the regression posterior can be modeled as

$$P(y|x, P_t) = N_t(y|\mu_t, \Sigma_t) \tag{13}$$

where P_t is a partition built by a random tree t_t, N_t is a multivariate Gaussian with mean μ_t, and covariance Σ_t is predicted in the output space Y from the subsets of the training dataset. The purpose of training the trees is to reduce the uncertainty related to the multivariate Gaussian model, especially when an appropriate splitting function f must be selected to split the subset S_l of the training set. These calculations are performed at each arriving node N_l in the tree t_t to reduce any prediction uncertainty caused due to "splitting".

An example of function f includes information gain and the Gini index. The unweighted differential entropy function, which is a continuous version of Shannon's entropy (SE), is considered an optimal function for computing information gain in a regression task [119], [120]. The SE function was selected, as it reported satisfactory results in terms of prediction error, defined as

$$f(S_l) = \int_{(y\in Y)} \sum_{i=1}^{n} P(y|S_l) \log(P(y|S_l)) dY \tag{14}$$

Where i is a given input instance and y is output including both fault duration and location. As we model the posterior using multivariate Gaussian, f can be rewritten as [34]

$$f(S_l) = \frac{1}{2}\log((\pi \exp)^{D'} |\Sigma^{(S_l)}) \tag{15}$$

where $\Sigma^{(S_l)}$ is the covariance matrix estimated from the subset S_l . After splitting the subsetS_l at node N_t into two subsets nodes, S_l^{right} and S_l^{left}, using function f, the information gain Δ is calculated using

$$\Delta = f(S_l) - w_l f\left(S_l^{left}\right) - w_r f\left(S_l^{right}\right) \tag{16}$$

where $w_l = \frac{|S_l|}{|S_l^{left}|}$ and $w_r = \frac{|S_l|}{|S_l^{right}|}$. Once the training phase is completed, the prediction phase consists of sending the new received instances through the trees of the forest and the posteriors of all the trees are estimated using the following equation:

$$P(y|x) = \frac{1}{T}\sum_{t=1}^{T} P(y|x, P_t) \tag{17}$$

where T is the number of trees in the forest and P_t is the partition introduced by P_t. Given any new instance, the model can predict its corresponding fault duration and location by maximizing a posterior:

$$\hat{Y} = argmax_{y\in Y} P(y|x) \tag{18}$$

2.2. Dataset

The simulated fault scenarios were completed using GridPACK software, an open-source framework designed to support the development and implementation of Smart Grid applications. Examples of these applications include power flow simulations for the electric grid, contingency analysis of the power grid, state estimation based on electric grid measurements, and the dynamic simulation of the power grid. These applications are capable of running on high-performance computing architecture (HPC) [110]. The dynamic simulation application package in GridPACK was selected to simulate a three-phase fault at various bus locations with different fault duration(s) using a nine-bus system. The faults duration was varied from 0.05 to 0.5 s along with fault strength levels, such as magnitude. An example of scenario one is depicted in Figures 18 and 19. Three features were selected to capture both the fault location and duration: the voltage magnitude (V_m) at each bus, the phase angle (φ) at each bus, and the frequency (f) of the generators. The timing of the fault applied to each bus was ten seconds. The total number of samples for all simulated scenarios equaled 53,512 samples. A summary of the training and test data is listed in Table 3. Additionally, Figure 20 illustrates a conceptual diagram of the proposed approach starting from simulation fault scenarios to evaluating the RFR model's performances.

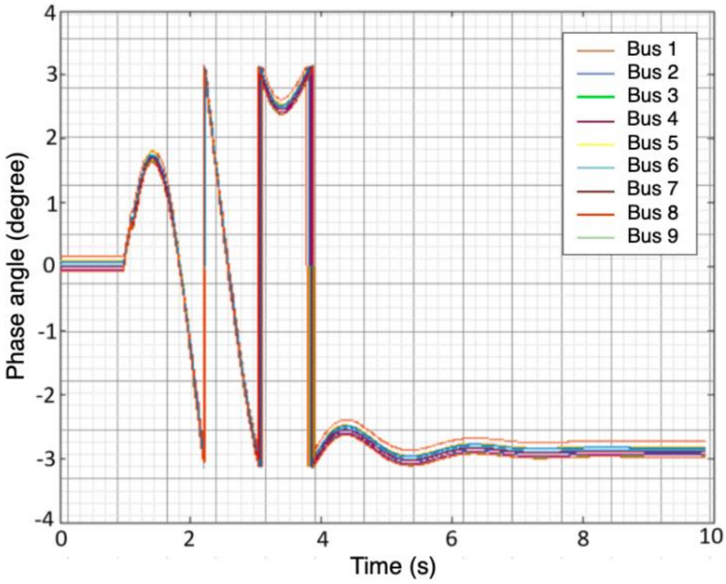


Figure 18. The phase angle of the 9 buses after injecting a three-phase fault.

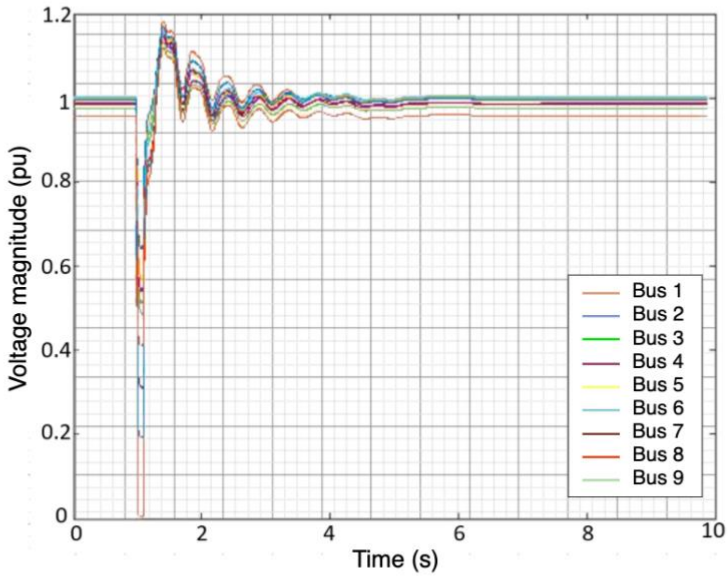


Figure 19. Voltage magnitude of the 9 buses after injecting three-phase fault.

Table 3. Common three-phase fault modeling for nine scenarios with different duration

Scenario	Fault location	Fault duration	Simulation time	Number of generated sample for each fault duration	Number of generated samples for each scenario
Scenario 1-9	Apply fault at bus 1-9	0.05s to 0.5s with a step of 0.05s	10s	594 samples	5945 samples/scenario. Total number of samples is 53512

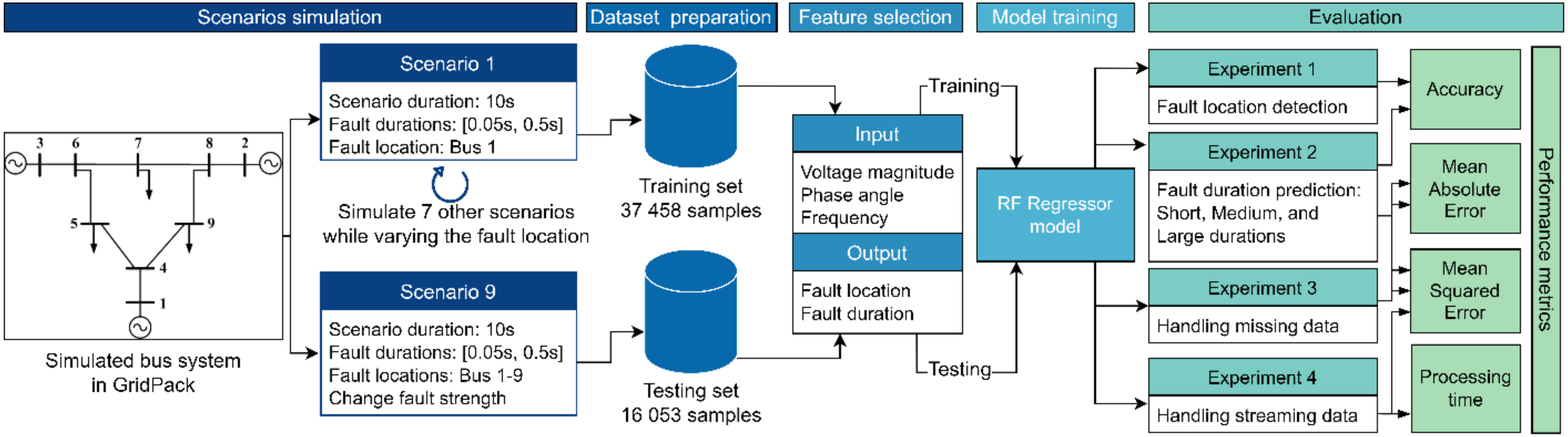


Figure 20. Conceptual diagram of the proposed RFR-based model

3. Experiments and Metrics

Four experimental scenarios were considered for the evaluation of the RFR model performance. The proposed model was assessed based on the accuracy metric in experiment 1, which is the ratio of the correctly classified fault location cases over the total number of cases. The $accuracy$ metric can be expressed as

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{19}$$

where TP is the true positive, TN is the true negative, FP is the false positive, and FN is the false negative. These values were obtained from the confusion matrix. The second set of experiments evaluated the model's performance when predicting the fault duration. As this feature is a continuous value, the accuracy metrics cannot be used; therefore, other performance metrics, such as MAE and MSE, were selected. The MAE is the average of the absolute differences between the actual and predicted fault duration, and it is given by

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |(\hat{y} - y)| \tag{20}$$

where $\hat{y}$ is the predicted fault duration, y is the actual fault duration, and n is the number of instances or cases. Unlike MAE, MSE has the benefit of penalizing for significant errors because it averages the squared differences between the actual fault duration and the predicted one, expressed as

$$MSE = \frac{1}{n}\sum_{i=1}^{n} (\hat{y} - y)^2 \tag{21}$$

The MSE and MAE of the proposed model were compared to the regression version of the other seven models listed above. The fault duration and location were evaluated in a streaming window environment during experiment 3.

3.1. Models Hyper-parameters Tuning

To conduct a fair comparison between RFR and the other models, a hyper-parameter study was conducted to determine the optimal parameters. Two approaches can be investigated to select the best hyper parameters: GridSearch and RandomizedSearch [121]. The former is convenient for an exhaustive search for the best-performing hyper parameters given advanced computing resources, whereas the latter defines a grid of hyper parameters and randomly selects the optimal one [121]. GridSearch was employed to examine, in-depth, the relevant parameters for each model and their optimal values using a subset of the data. For KNN, two weighting functions were chosen with varying numbers of neighbors: uniform and distance. In uniform weighting, all points within the neighborhood are weighted equally, while in distance weighting, closer neighbors are given more weight [122]. In the RFR method, two maximum features methods were selected, sqrt and log2, to determine the number of features to consider when looking for the best split. For SVM, two kernel types were chosen, polynomial and RBF. A range of regularization parameters (C) was also considered. For NN, two activation functions were selected in conjunction with a variety of hidden nodes. In DT, the minimum number of samples needed to split a node internally was determined; additionally, various values were investigated to control randomness within the tree. Alpha and lambda were selected as the shape parameters for NB; alpha is the shape parameter for the Gamma distribution before alpha, and lambda is the shape parameter for the Gamma distribution before lambda [122]. For DNN, two numbers of hidden layers were chosen, each of which has multiple hidden neuron nodes. Finally, two split functions were investigated for HT with varying split confidence values.

Table 4 provides the GridSearch methods results for the various models. The optimal parameters for each model are highlighted. KNN reported the lowest MSE values, 6.7 and 0.16 standard deviation, with the distance weight function and 100 neighbors. According to these results, KNN fits data more smoothly with an increasing number of neighbors; this is because more neighbors reduce the edginess by taking into account more data, thus lowering the overall error of the model. SVM reaches low error, 5.9, when using RBF kernel and a regularization parameter (C) set to 10. These results reflect that increasing the C value can contribute to low error rates, possibly because there are more potential data points within the margin or that were incorrectly classified, which can be corrected by using a high C value. DT performs better with a leaf size of six and a random state of one. Based upon these results, it appears that increasing the minimum leaf size will increase the model's ability to determine the appropriate pruning strategy, and, as a result, improve its performance.

Table 4. Hyper tuning parameters for KNN, RF, DNN, DT, NB, HT, NN, and SVM.

Model	Hyperparameters			Mean Squared Error	Standard Deviation
KNN	Weight function	Uniform	1	11.21	2.6
			10	7.25	0.45
			100	6.71	0.17
		Distance	1	11.21	2.6
			10	7.24	0.43
			100	6.7	0.16
SVM	Polynomial kernel	C=1		6.013	0.11
		C=5		6.13	0.14
		C=10		6.16	0.08
	Radial basis function (RBF) kernel	C=1		6.09	0.14
		C=5		6.17	0.08
		C=10		5.9	0.1
DT		Random state	0	10.51	3.56

Model	Hyperparameters			Mean Squared Error	Standard Deviation
	Minimum leaf size = 1		1	10.39	3.65
			2	10.58	3.57
	Minimum leaf size = 6	Random state	0	9.32	3.15
			1	9.29	3.12
			2	9.31	3.15
DNN	Relu function	5 hidden layers	50 hidden nodes	1.20 × 10−2	2.40 × 10−3
			100 hidden nodes	1.12 × 10−2	1.39 × 10−3
			150 hidden nodes	1.14 × 10−2	1.39 × 10−3
		10 hidden layers	50 hidden nodes	1.12 × 10−2	3.51 × 10−3
			100 hidden nodes	1.14 × 10−2	1.39 × 10−3
			100 hidden nodes	1.20 × 10−2	2.40 × 10−3
RFR	Max feature: sqrt	Number of trees	1	10.31	2.68
			10	6.45	0.53
			100	6.2	0.67
	Max feature: log2	Number of trees	1	10.52	2.68
			10	6.75	1.26
			100	6.15	0.63
NN	Relu function	Number of hidden nodes	150	4.37	0.18
			300	4.64	0.23
			450	4.62	0.12
	Identity function	Number of hidden nodes	150	6.15	0.08
			300	6.15	0.06
			450	6.16	0.08
NB	Alpha = 1 × 10−6	Lambda	1 × 10−6	1.26 × 10−3	1.42 × 10−4
			1 × 10−4	1.17 × 10−3	1.56 × 10−4
			1 × 10−2	1.07 × 10−3	1.66 × 10−4
	Alpha = 1 × 10−4	Lambda	1 × 10−6	1.14 × 10−3	1.98 × 10−4
			1 × 10−4	1.19 × 10−3	1.51 × 10−4
			1 × 10−2	1.15 × 10−3	2.37 × 10−4
HT	Split function: Gini Index	Split confidence	1 × 10−5	12.41	4.88
			1 × 10−4	14.53	6.13
			1 × 10−3	14.91	6.24

Model	Hyperparameters			Mean Squared Error	Standard Deviation
	Split function: Information gain	Split confidence	1 × 10−5	10.88	2.89
			1 × 10−4	11.24	8.13
			1 × 10−3	17.64	7.22

The optimal DNN configuration entails five hidden layers, each of which contains 100 hidden nodes with the Relu function. These results suggest that the number of hidden layers and hidden neuron nodes did not dominate the model's performance; that is, the model obtained the best results without overfitting by using five hidden layers, each containing 100 neuron nodes. The RFR showed optimal results using log2 as a maximum feature and 100 trees. When splitting a node with log2 as a maximum feature, RFR is better able to find the optimal size of the random subset of features. An optimal configuration of NN includes a Relu function and 150 hidden nodes. NN results indicate that an increased number of trees does not improve the model's performance, but rather the choice of an activation function; Relu has demonstrated a lower error rate than identity. The optimal alpha and lambda settings for NB were set to 1 × 10−6 and 1 × 10−2, respectively. NB's results indicated that changes to alpha or lambda values do not have a significant impact on model performance. HT's optimal parameters are information gain, as a split function, and split confidence set to 1 × 10−5. HT's results indicated that selecting a lower confidence level while using information gain reduced error rates and their standard deviations significantly.

3.2. Experiment Result #1: Fault Location Detection

The results of experiment 1 are depicted in Figure 21. This figure explores the comparison between the proposed model (RFR) and the other seven models in terms of fault location detection accuracy at nine different fault locations. At fault location 1, the

RFR approach detects approximately 92% of the faults, followed by DNN with 80% accuracy. NB reports the poorest performance with an accuracy rate below 2%. At the second location, DNN and RF report similar results, 71%, followed by KNN, NN, and SVM. At the third fault location, RFR reports the highest accuracy rate, 94%, followed by DNN with 78%, then KNN with 46%. RFR detects 76% of the faults at fault location 4, compared to DNN at 60%. Table 5 provides the processing time along with accuracy for the training and testing of these models. Although the testing time for the NN, NB, and DT models is relatively low, the accuracy was under 20%. Alternatively, RFR and DNN reported respective accuracy of 84% and 72% with a short test time below 0.046 s.

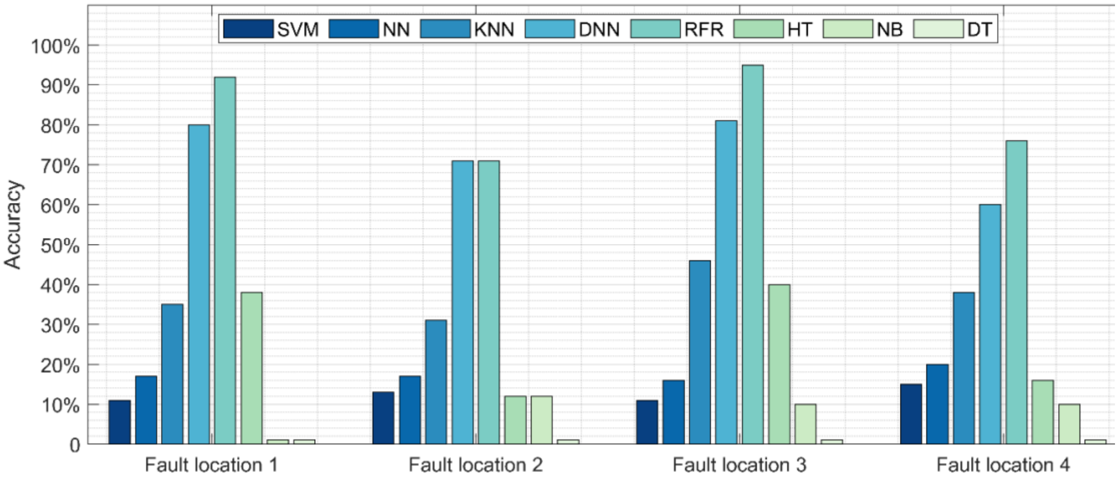


Figure 21. Comparison between the proposed model (RFR) and NN, DNN, SVM, NB, DT, and HT in terms of fault location detection accuracy at various locations.

3.3. Experiment Results #2: Fault Duration Prediction

The fault duration prediction results are illustrated in Figure 22. The lowest reported MAE values were from the RFR, HT, and DNN models. The highest MAE value was from

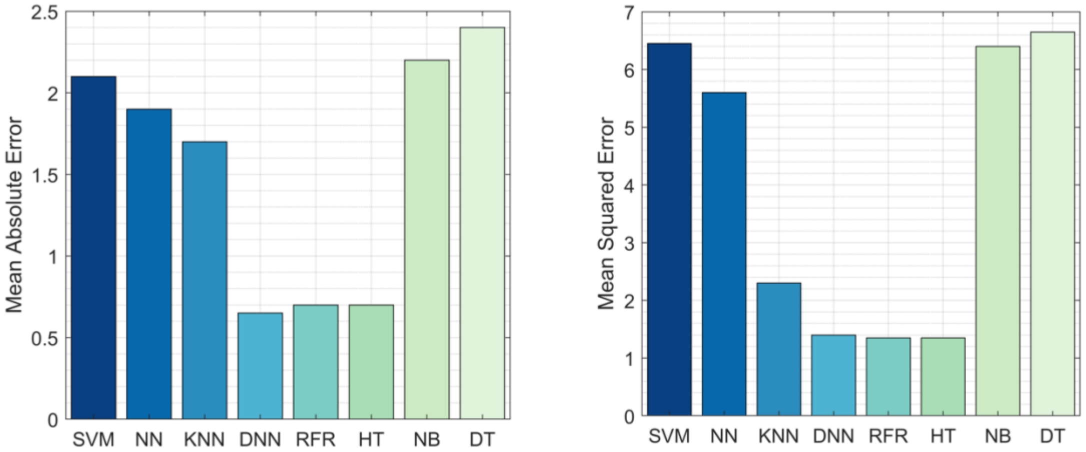


Figure 22. The MAE and MSE of RFR, NN, DNN, SVM, NB, DT, and HT in terms of fault duration prediction.

DT, at 2.4 s. These results suggest that DNN, HT, and RFR are the optimal models for predicting fault duration as the difference between the actual and predicted duration for the entire testing dataset was less than 0.6 s. Figure 22 also depicts the MSE of RFR compared to the other models. The RFR and HT models reported the lowest MSE value, close to 1 s; however, the prediction error for DNN was more than 1.5 s. The RFR, HT, and DNN models yield optimal results for MAE and MSE; therefore, these models were selected for the next experiment.

Figure 23 illustrates fault location detection by comparing the three optimal performing models, DNN, RF, and HT, tested with three different fault durations: a short fault duration ranging between 0.05 and 0.15 s, a medium fault duration ranging between 0.2 and 0.35 s, and a long fault duration ranging between 0.4 and 0.5 s. The RFR model outperforms DNN and HT when detecting faults with short, medium, and long durations. The RFR model reports a 65% accuracy when detecting the short fault duration, followed by DNN with 12%, then HT with 10%. The accuracy of the RFR model increases to 91% detection for the medium duration, followed by HT with 22%, then DNN with 16%. The RFR model reports a 91% fault detection accuracy for the long duration, followed by HT with 24%, then DNN 16%. These results suggest that the RFR model is an appropriate model for detecting short, medium, and long fault durations. DNN requires a larger dataset to achieve optimal performance, which may explain its poor performance. We split the dataset into three parts with specific fault durations: 1. A short fault duration with 16,211 instances; 2. Medium fault duration with 21,500 instances; and 3. Long fault duration with 15,800 instances. Training and testing DNN on each sub-dataset was not sufficient for it to achieve optimal detection accuracy, suggesting that RFR can achieve its highest accuracy with a relatively small number of instances compared to the DNN model.

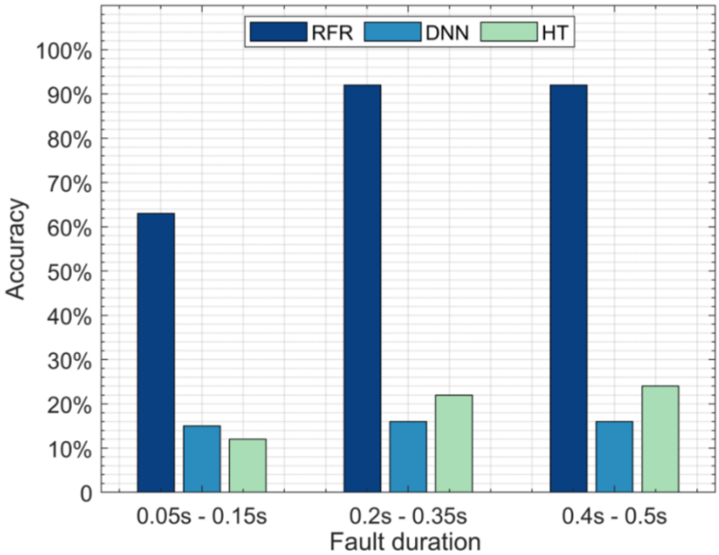


Figure 23. Accuracy of RF, DNN, and HT in terms of detecting fault location with various duration.

3.4. Experiment Results #3: Handling Missing Data

Figure 24 illustrates MSE and MAE as a function of the percentage of missing data for the three selected models: DNN, RFR, and HT. The purpose of this experiment was to evaluate the model’s robustness when handling missing data. The collected measurements within a real power system network, including voltage, magnitude, and frequency, can be incomplete due to equipment failure, data storage issues, or unreliable communication [31];

therefore, it is crucial to evaluate the model's capacity for accurately predicting the fault duration. The MSE of the three models increases as the percentage of missing data increases (Figure 24). RFR's MSE values of 1.8 and 7.8 correlate to missing data percentages of 10% and 90%, respectively. DNN reports an MSE value of 2.5 with 10% of the data missing, while HT reports an MSE value of 6 for the same percentage of data missing. The MSE values of DNN and HT increase as the percentage of missing data increases, reaching the highest value of 10. Figure 24 also illustrates the MAE as a function of the percentage of missing data. The evolution of the MAE value for the three models indicates similar behavior to the previous one. RFR has an MAE value of 0.85, followed by DNN with 1.29, then HT with 2.1, with 10% of the data missing. The MAE values of the three models increase as the percentage of missing data increases to reach their highest values, which are 2.25 for RFR, 2.58 for HT, and DNN, with 90% of the data missing. These results suggest that the RFR model is more resilient and tolerant to missing data; therefore, it is the optimal model for fault duration prediction even with incomplete data.

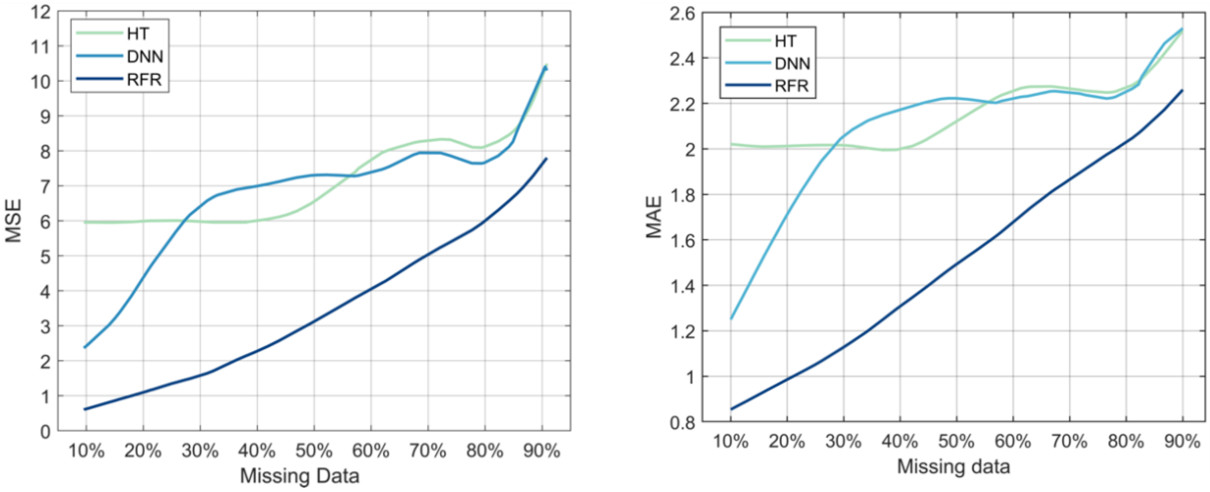


Figure 24. MSE and MAE as a function of the percentage of missing data for the three models: DNN, HT, and RFR.

3.5. Experiment Results #4: Handling Streaming Data

The RFR, DNN, and HT models selected from the previous experiments were evaluated with streaming data. The models were trained incrementally: they were not trained and tested on the entire dataset, they were incrementally trained with one sample at a time. The MSE and the processing time of each model were then evaluated (Figure 25). The MSE of the RFR values were consistently below 0.1 s as the number of samples increased. For HT, the MSE dropped sharply from 28 s to 0.5 s; for samples between 0 and 10,702, it stagnated at 0.5 s, and then dropped to 0.1 s. The DNN's MSE values decreased from 30 s to 2 s as the number of samples increased to 32,107, then decreased slowly to reach the lowest value of 0.1 s before stabilizing. RFR reported the lowest value for processing time per sample: 0.0028 ms, followed by DNN with 0.0032 ms, then HT with 0.7 ms. The results obtained in this experiment set suggest that RFR is a potential model for detecting fault locations within a near-real-time streaming environment. A summary of the obtained results for the four experiments is provided in Table 5. The overall accuracy, MAE, MSE, processing

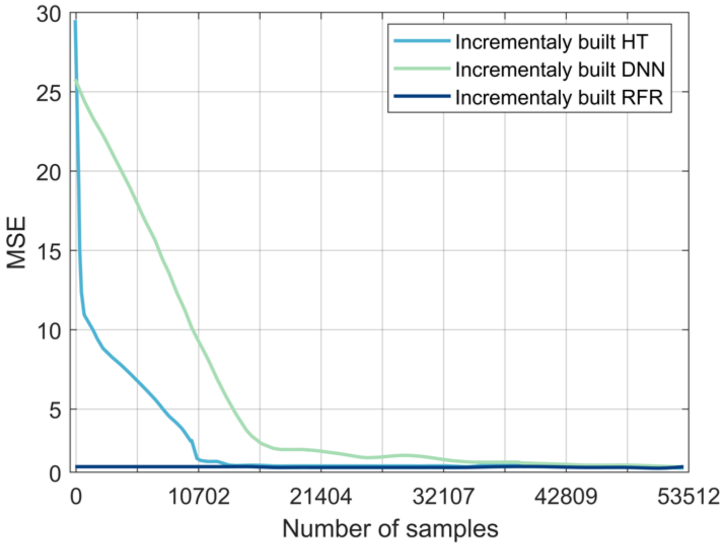


Figure 25. Comparison between DNN, HT, and RFR in terms of MSE.

time, and overall rankings are high for RFR, medium for DNN, and low for the other models.

Table 5. Summary of the RFR's performances compared to those of DNN, HT, NN, SVM, DT, NB, and KNN, obtained in the four experiments.

Experiment	Performance Metrics	RFR	DNN	HT	NN	SVM	DT	NB	KNN
1. Detecting fault location	Overall accuracy for four fault locations	84%	72.5%	27%	18.75%	14%	2%	8.25%	41%
2. Predicting fault duration	MSE	1.1 s	1.2 s	1.1 s	5.6 s	6.5 s	6.6 s	6.2 s	5.1 s
	MAE	0.6 s	0.6 s	0.6 s	1.9 s	2.2 s	2.5 s	2.2 s	1.8 s
3. Handling missing data	MSE	4.6 s	8.4 s	8.7 s	-	-	-	-	-
	MAE	1.5 s	2.09 s	2.14 s	-	-	-	-	-
4. Detecting fault in streaming data	Processing time per sample	0.0028 ms	0.0032 ms	0.7 ms	-	-	-	-	-
Overall ranking		High	Medium	Low	Low	Low	Low	Low	Low

3.6. Discussion

Experimental results show that the performance of the ensemble method used in this paper, i.e., RFR, consistently outperforms the other models in simultaneously detecting the location and duration of faults on a multi-bus system. With an overall accuracy of over 84%, the RFR model performed optimally and consistently at various fault locations as well as with various fault duration (short, medium, and long), suggesting that RFR is the appropriate model for this dual-purpose task. For the same task, DNN also demonstrated consistent overall performance, albeit at the expense of a long processing time that makes it unsuitable for real-time applications.

While machine learning models can perform fairly well in an ideal and deterministic environment that is free from anomalies, it was pragmatic and necessary to evaluate the performances of these models in scenarios where data might be missing due to equipment failure, data storage issues, or unreliable communication. Based on the results in this paper, all three models, i.e., HT, DNN, and RFR, show MSE and MAE values of under 11 and 2.6, respectively, when the percentage of missing data is progressively increased from 10% to 90%. Depending on the severity of the said factors, RFR was proven to be the optimal candidate, followed by HT and DNN, in extrapolating system status during non-steady-state operations.

Another critical component of a model's capability, when deployed in a real-world environment, is its ability to evolve and adapt to unexpected data distribution changes and concept drifts. From the experimental results, RFR achieved an MSE 0.0028 ms when trained and tested incrementally on streaming data, which makes it suitable for detecting faults in near-real-time. Overall, the RFR model performed optimally and consistently under four different scenarios, indicating that the model can generalize and adapt to new, previously unseen, data without the risk of overfitting or underfitting

4. Conclusions

An RFR-based model was successfully implemented to identify the location and duration of faults. Various fault scenarios were modeled using PNNL's GridPACK software to generate the training dataset. A total of nine fault scenarios was simulated by injecting faults on specific buses over a specified period of time. The RFR models were trained and evaluated within the context of four study cases: detecting fault location, predicting fault duration, handling missing data, and streaming data. A comparison was also conducted between the RFR model and several state-of-the-art models using multiple performance metrics, including accuracy, MSE, and processing time. Results indicate that both RFR and DNN models are suitable of detecting the location and duration of a fault with an accuracy of 84% and 72%, respectively. The RFR, DNN, and HT models yielded better results when predicting faults in streaming networks. Overall, the RFR model consistently outperformed the other models, making it appropriate for real-time situational awareness deployments to determine both the location and duration of the faults while handling missing data. The next chapter will be focus on developing a machine learning model to detect anomalies in real-time and to handle the potential concept drifts due to load change, blip, or noise.

Chapter V

Adaptive Hoeffding Tree with Transfer Learning for Streaming Synchrophasor Data

This chapter extends the previous one by developing anomaly detection approaches for real-time networks with adaptability to concept drifts. The proposed approach is titled Transfer Adaptive Hoeffding Tree (THAT) based on Hoeffding Tree and transfer learning. This Chapter is divided into the following sections: Section II covers an overview of THAT model, it details the synch phasor training dataset, and the relevant features used for detecting the oscillation events; and section III evaluates the proposed model with some performance metrics. Section VI draws some conclusions[4].

1. Introduction & Literature review

Phasor measurement units (PMUs) are key assets in the Smart Grid for improving situational awareness within the grid and detecting potential anomalies. Each PMU generates between 30 and 60 samples/s [15], roughly 1.5 TB/month of streamed data. Thus, Loading and processing an open-ended source of fast and huge volumes of PMU data is challenging in terms of affording the required computational resources. The conventional

[4] This chapter is a slightly modified version of our paper "Adaptive Hoeffding Tree with Transfer Learning for Streaming Synchrophasor Data Sets” published in the IEEE International Conference on Big Data (Big Data), 2019, pp. 5697-5704, doi: 10.1109/BigData47090.2019.9005720.

ML approaches, used for extracting insights are no longer relevant since they require loading and scanning the entire dataset [60]. Thus, two important criteria have to be satisfied by any ML approach when it comes to dealing with streaming PMU data: building the model with a limited dataset as it is challenging to store the entire streaming PMU measurements into memory, and the model must adapt itself to concept drift when the data distribution changes gradually or abruptly over time, which could be due to load changes. Otherwise, the model built in the past will no longer be consistent with the data received in the present and its performance will decrease. Developing a classifier that meets these requirements is challenging, so approximate solutions can be considered with an associated error to be minimized.

Relatively few studies have considered the streaming nature of the PMU data and developed a machine learning accordantly. For instance, In [123], the authors proposed a Hoeffding Tree (HT) combined with two concept drift detectors (e.g., drift detection method: (DDM) and Adaptive sliding windows: (ADWIN)) for building dynamic DT that are adaptable to quick changes. This method has been trained on a synthetic PMU dataset, where multiple attack scenarios were modeled. The dataset contains normal, anomalous, physical, and cyber events. The physical event includes relay-based faults, and the cyber events include injections of various trip commands, SLG fault replay commands, and disable command attacks. The model (e.g., HT+ADWIN+DDM) trained and tested reported a classification accuracy greater than 98%. Similarly, authors in [124] proposed a HAT-based approach for detecting events on PMU data. The authors modeled two scenarios in their PMU data sets; 1) a three-phase fault has been generated with some load fluctuations altering true power (P) and reactive power (Q) at a regular interval, and 2) cataloged two classes such as fault and normal. Fault class includes a SLG fault, while the normal class includes normal power system variables such as voltage (v), phase angle (φ), current (i), and frequency (f). Based on the reported results, HT showed a good ability in detecting the power system faults and adaptation to the concept drifts in comparison with traditional DT such as J48 and REPTree.

However, in the above two studies, the duration of the physical fault events was not considered in their data set. There is a certain types of power system faults that are time-sensitive and thus require early detection. The cause of inter-area oscillations is primarily due to system events coupled with a poorly damped power system. Generally, these low-frequency oscillations (0.1-0.8 Hz) are noticed in a larger grid with multiple generators or renewable plants (with high wind or solar penetration) that are connected to weak tie-line connections. This can lead to a high degree of uncertainties in the system and it is often difficult to detect in real-time. Specifically, smaller frequency deviations that range from 0.15-1.0 Hz lasting 60 seconds or longer may cause inter-area oscillations and quickly destabilize the grid [74]. Similarly, momentary voltage or current instabilities (e.g., surges or spikes of 2-5 seconds) may lead to asset failures (e.g., transformer, relays, or circuit breakers). Thus, including signatures with various event duration is a key feature for training real-time machine learning algorithms. In this chapter, a transfer learning-based HT with ADWIN is proposed to detect anomalies in the streaming PMU data using four-event signatures with varying durations. Multiple HT classifier is trained on normal and anomalous signatures, while ADWIN is included at the leaf for adaptation to concept drift (e.g., fluctuations).

The proposed THAT model is trained on four event signatures with varying durations. As HT and ADWIN require fixed features during the training phase and they cannot be trained on events with different durations, these models will be improved by incorporating a transfer learning (TL) technique. Here, TL refers to the transfer (or retention) of knowledge that can be learned across multiple similar tasks, but not identical [125]. The process of TL is as follows: the model is trained on the first signature (e.g., first task) and any acquired knowledge will be transferred to perform the training on the next task, that focuses on the second signature. Thus, the process of transferring the learned training can be repeated to subsequent signatures. To the best of the author's knowledge, this is the first attempt to integrate transfer learning with any streaming classifier for anomaly detection containing signatures with varying durations. In [125], the authors proposed transfer learning in a decision tree, but it was not for streaming data applications.

2. Methodology

2.1. Transfer Adaptive Hoeffding tree (THAT)

An efficient real-time anomalies detection model in PMU data streams must satisfy two fundamental requirements: 1) scanning only a small sample of the data in order to build the model rapidly and efficiently; and 2) adapting to concept drifts (e.g., changes in data distribution) in real-time. Conventional batch models are not suitable to operate in such an environment as they do not evolve over time and may fail to capture new anomalous events.

Alternatively, the Hoeffding Tree (HT) is a prominent model candidate as it requires a minimum number of arriving samples to build trees with a certain confidence level [126]. Instead of loading the entire data into memory, HT loads a small sample of PMU data to generate the tree, which will be stored in memory along with some statistics relating to incoming data streams in the leaves to allow the model to evolve over time. This was made possible due to the fact that a small sample can often be enough to choose an optimal splitting attribute and start building the tree, assuming that the data distribution does not change over time [126]. Additionally, the HT model is mathematically supported by the Hoeffding bound, which quantifies the number of samples necessary to estimate a statistic within a prescribed precision (in this case, the significance of an attribute). Unlike other incremental decision tree models, using the Hoeffding bound one can demonstrate that the HT output is asymptotically nearly identical to that of a batch decision tree model using infinite samples [127], [128]. Furthermore, the Hoeffding bound provides a strong bound when compared with Markov's and Chebyshev's bounds [129], [130]. However, while HT has shown exceptional performance in detecting anomalies in a streaming data setting [72], [124], it is unable to adapt to potential concept drift. In this dissertation, we develop a model named Transfer Adaptive Hoeffding Tree which leverages the HT as a base model and includes a concept drift detector, Adaptive Windowing (ADWIN), and supervised transfer learning to build a dynamic model suitable for detecting anomalies in a high-speed PMU data stream, adaptable to the eventual concept drifts and with better generalizability. Algorithm 1 shows the pseudocode of the THAT algorithm for detecting anomalies in the PMU stream data. The algorithm has two stages, stage 1: create a new HT, and stage 2: carry transfer learning operations between two HT models by training the target model on the subsequent signature in the queue using the learned HT tree.

The details for these two stages are as follows: In stage 1, (lines 1-20), a new HT target tree is created if the HT source tree is not provided. During this early stage, transfer learning is not required. In line 2, a target tree is initialized by creating the first node (root). In lines 3-19, a for loop is performed for all the training instances, where each sample is filtered down the tree to the appropriate leaf l based on the test sequences present in the HT built to that point (line 4). During this process, sufficient statistics on the samples and Hoeffding bound are computed. Each leaf l contains enough statistics to make decisions about the further growth of the tree. These statistics need to be sufficient to enable the calculation of the Information Gain afforded by each possible split. However, storing unnecessary information would increase the total memory requirement for the tree. In line 5, the statistics hold by l are updated to estimate the Information Gain of splitting each attribute. In line 6, the n_l is the number of samples seen at leaf l (computed from the sufficient statistics), is updated. Lines 7-18 are executed only when a mix of different classes enables further splitting. In line 8, the splitting criterion G is used to estimate the $\overline{G_l}$ value for each attribute. The function G measures the average amount of purity that is gained in each subset of a split and indicates how well a given attribute separates the training examples according to their target classification [128]. In this study, two different approaches will be investigated to compute the function G: Information Gain (entropy) and the Gini index.

Algorithm 1. Transfer learning Hoeffding Adaptive Tree (THAT)

```
Input: training set S, S’, $HT_{source}$
Output: $HT_{target}$
If $HT_{source}$ is NULL
  Let $HT_{target}$ be a tree with a single leaf (the root)
  For all instances in $S$ do
    Sort instances into leaf $l$ using $HT_{target}$
    Update sufficient statistics in $l$
    Increment $n_l$, the number of instances seen at $l$
    If $n_l$ mod $n_{min} = 0$ and instances seen at $l$ not all of same class
      Compute $\overline{G_l}(A_i)$ for each attribute
      Let $A_a$ be attribute with highest $\overline{G_l}$
      Let $A_b$ be attribute with second highest $\overline{G_l}$
      Compute Hoeffding bound $\epsilon = \sqrt{\frac{R^2 \ln(\frac{1}{\delta})}{2n_l}}$
      If $A_a \neq A_0$ and $\overline{(G_l}(A_a) - \overline{G_l}(A_b) > \epsilon$ or $\epsilon < \tau)$
        Replace $l$ with an internal node that splits on $A_a$
        For all branches of the split do
          Add a new leaf with sufficient statistics
        End for
      End if
    End if
  End for
Return HT$_{target}$
Else
HT$_{target}$ = HT$_{source}$
$Q \leftarrow$ all attributes of $S'$ not in $HT_{target}$
For each attribute $A'$ dequeued from $Q$
  For each training instance $I$ in $S$’
     Classify $I'$ using the $HT_{target}$
     If $I$‘ is predicted correctly then
       Do nothing
     Else
      Replace $HT_{target}$ ‘s class node with a new node for attribute $A'$
      Add a new branch to node $A'$, labeled with $A$'s value in $I'$
      Add a new leaf node labeled with $I'$s target class label
     End if
  End for
End for
Return HT$_{target}$
End if
```

If the distribution of the two classes (e.g., oscillation event class, and normal event class) in the PMU stream contains the probabilities p1, and p2 of the classes, then the entropy of a given attribute A in a training data set S is calculated by:

$$Entropy\ (A) = \sum_{i=1}^{n} -p_i log_2 p_i \tag{22}$$

Here n is the number of classes and it is equal to 2. The attribute, A, is one of the selected features which could be voltage (v), phase angle (φ), current (i), and frequency (f). Then, the Information Gain is computed by:

$$Information\ Gain(S,\ X) = Entropy(X) - \sum_{k\in(X)} \frac{|S_k|}{|S|} Entropy(S_k) \tag{23}$$

Where k is the value of the attribute A, and S_k is a subset of S where $A = k$.

The other metric considered for evaluation is known as Gini Index, and this index is computed as:

$$Gini(X) = 1 - \sum_{i=1}^{n} {p_j}^2 \tag{24}$$

Here n is the number of classes (e.g., n=2 in our case). In lines 9 and 10, the attributes with the largest gain information are used for the next steps. Line 11 computes the Hoeffding bound such that probability $1 - \delta$ corresponds to a confidence value of $\delta \in \{1,0\}$, where the true mean of a random variable of range R will not differ from the estimated mean after n independent observations by more than ϵ and it is given by:

$$\epsilon = \sqrt{\frac{\mathrm{R}^2 \ln(\frac{1}{\delta})}{2\mathrm{n}}} \tag{25}$$

In line 12, a split criterion test is performed between the largest gain value attribute and A_0 and compared with the Hoeffding bound. The value of τ is used to analyze trade-off conditions. If the attribute obtained is the best choice, then the node is split and the tree grows (lines 13-15) [124], [131], [132].

Lines 21-37 focus on the concept of "transfer learning", where the knowledge gained between two HT models trained on two different signatures (i.e., tasks) are shared or transferred. The model is scalable to transfer knowledge beyond two HT models if needed. Before we define transfer learning and how it is used with HT, we must understand the term 'standard learning' (SL). The authors in [28] define SL using the following description process: "Let D be a domain consisting of r-dimensional feature space X, a label space Y, a probability distribution $P = (x, y)$ where $x \in$ X is the feature vector, and y $\in$ Y. With a finite set of labeled examples $S = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n))$ drawn from P and a loss function J, standard learning consists of defining a function $f{:}\, X \to Y$ with a minimum J value". In the case of transfer learning, a source domain $D_S = (X_S, Y_S, P_S)$ and a target domain $D_T = (X_T, Y_T, P_T)$ are both used to learn a function $f'{:}\, X_T \to Y_T$ given a set of labeled examples S drawn from P_T and some information about D_S, such that the value of the loss function J' is as small as possible [133]. There are two possible types of transfer learning: inductive and supervised. In the inductive type, the source and target tasks are different, but they share some common features. The supervised type is used when $|\, S_T| >> |\, S_S|$ and aims to improve the task learning of domain D_T given S_T [133]. As shown in Figure 26, the proposed THAT model is trained on four signatures (oscillation and normal events) with different magnitude and durations: 400s, 120s, 60s, and 50s, respectively. Then supervised transfer learning can be applied to transfer knowledge between different THAT models since $|\, S_4| >> |\, S_3| >> |\, S_2| >> |\, S_1|$.

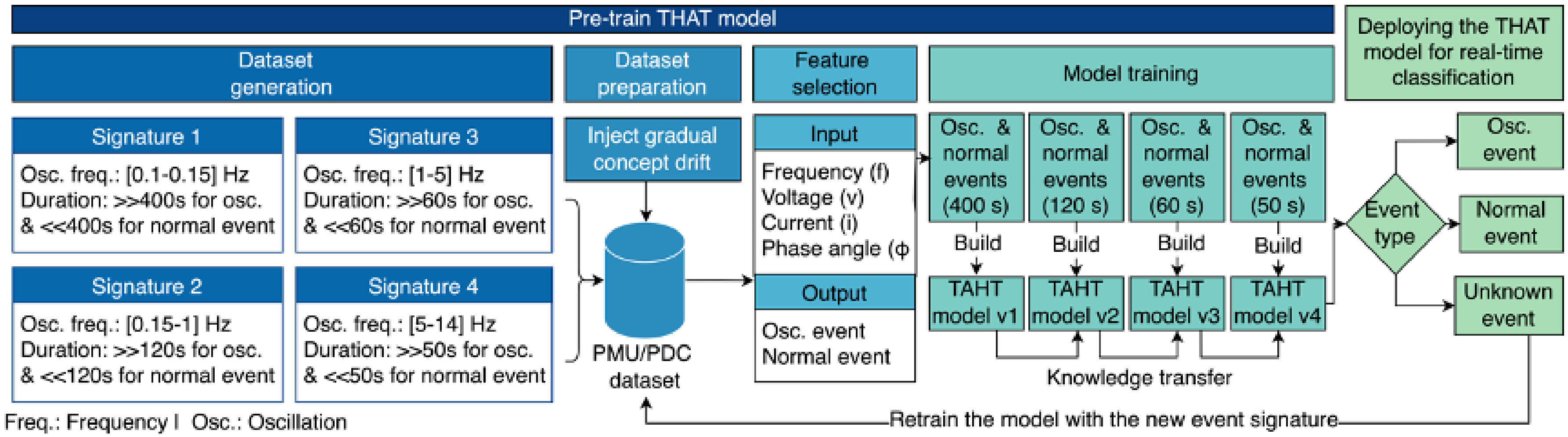


Figure 26. Conceptual diagram of the THAT model.

2.2. Concept drift detector

Due to the high potential of unexpected events that could occur in the distribution of the streaming PMU data, it is important to deal with these unexpected events and help the ML model in adapting and classifying these unknown events, this event is known as concept drift. A concept drifts when the law underlying the data changes gradually or abruptly. Additional concept drift types are given in Figure 27. Thus, the model built in the past is no longer consistent with the data received in the present. Concept drifts could be due to different reasons including noise, the influence of the environment, variation in characteristics not considered in the model, seasonal alterations. A convenient treatment of concept drift means that recent data that conflict with past assumptions should prioritize the construction of the model. But, at the same time, assumptions based on old data that are still consistent should be preserved.

Several approaches have been proposed for tackling such events. These approaches include DDM, Early Drift Detection Methods (EDDM), Linear Four Rate (LFR), Just in time (JIT), and ensemble methods [134], [135], [136]. DDM and EDDM don't require to store the data but they are susceptible to false alarms. LFR [134] is constant in space complexity and can deal with imbalance classes as it uses different error types separately, but it still suffers from the labeling cost. JIT [136] does not require labeled data and can detect the abrupt change but it is inefficient with the gradual drift. Ensemble methods [25] are effective in detecting the recurring concepts, but they use a large batch which makes them ineffective in identifying precisely the change location. ADWIN turns out to be an optimal concept drift detector since it is better for change localization and uses a dynamic window which adapts its size according to the change rate observed in the data within the window. When the data is stationary, the window size dynamically increases, and it decreases when a change is detected.

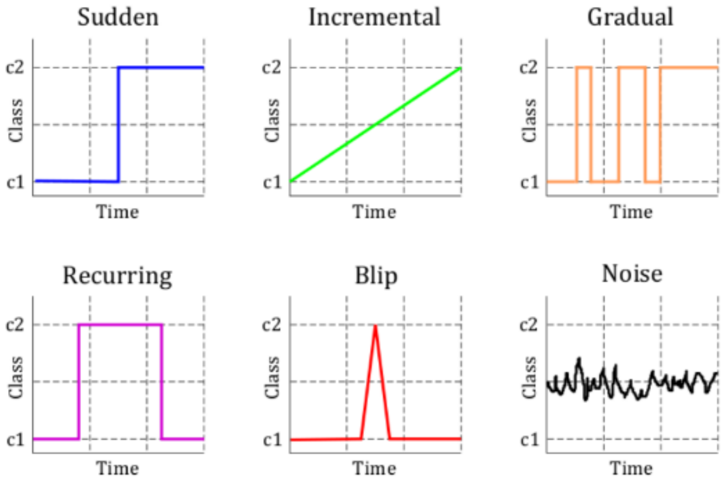


Figure 27. Concept drift types

In order to improve the HT, we include the ADWIN as a concept drift detector and make the model adaptable to the eventual concept drifts. ADWIN is also used to store temporarily the recent data for rebuilding or revising the classifier [138].

Let $x'_1, x'_2, x'_3, \dots x'_t$ be a sequence of real PMU values where the value of x'_t is available only at time t. Each x'_t was generated based on some distribution P_t and independently for every t. Let u_t be the expected value when it is drawn according to P_t. ADWIN uses a sliding window W with the recently received data. Let n denote the length of W, $\hat{u}_W$ the observed (known) average of the data in W, and u_W the expected (unknown)

Algorithm 2. ADWIN: Adaptive Windowing Algorithm

```
Initialize Window $W$
for each $t > 0$
  do $W \leftarrow W \cup \{x'_t\}$ (i.e., add $x'_t$ to the head of $W$)
    repeat drop elements from the tail of $W$
      until $|\hat{u}_{W0} - \hat{u}_{W1}| \geq \epsilon$ holds for every split of $W$ into
    $W = W_0 . W_1$
  output $\hat{u}_W$
```

average of the u_t for $t \in W$. If there are two sub-windows W_0 and W_1 with sufficiently different averages, then it can be concluded that the expected values will be different and the older portion of W, i.e. W_0, is dropped (Algorithm 2) [139]. The difference between the observed average ($\hat{u}_W$) and the expected average (u_W) is compared to ϵ which is computed based on the Hoeffding bound and is defined as:

$$\epsilon = \sqrt{\frac{1}{2m} \ln \frac{4}{\delta'}} \tag{26}$$

Where m is defined as:

$$m = \frac{1}{1/n_0 + 1/n_1} \tag{27}$$

Where n_0, $n_1$are the lengths of W_0 and W_1, respectively. The δ' is defined as:

$$\delta' = \frac{\delta}{n} \tag{28}$$

Where δ is confidence value and $n = n_0 + n_1$.

2.3. Dataset

The dataset used in this study includes a collection of oscillatory events (e.g., four signatures) recorded by PMUs across multiple substations at various locations of a power system [140]. Each signature represents a fault in the Smart Grid with associated parameters, e.g., voltage, current, phase angle and frequency information of varying durations, ranging from 3 to 6 minutes. The dataset is modified to introduce concept drift events in the four signatures. Each signature is identified by its oscillation frequency, duration, and potential cause. For example, a signature containing frequency ranges from 0.1 Hz to 0.15 Hz if held for up to 400 seconds then it is classified as an oscillation event, otherwise, it is considered as a normal event. The specific range and duration of oscillations have been selected according to the analysis conducted in [74], [141]. Table 6 provides the specifications of the four signatures. To evaluate the adaptability of the uncertain events, the data stream is injected with additional fluctuations or concept drifts using a Massive Online Analysis (MOA) generator [131]. A gradual pace of concept drift is modeled to mimic most fault progression in the power system [123]. The final dataset includes four signatures each of which includes 2000 normal events and 2000 oscillation events with gradual concept drifts.

Table 6. PMU Dataset description

Signatures	Oscillation frequency	Duration	Potential event cause	Classes	Concept drift
Signature 1	0.1 Hz – 0.15 Hz	>> 400s	Generators	Oscillation event	Gradual
	0.1 Hz – 0.15 Hz	<< 400s	-	Normal event	
Signature 2	0.15 Hz – 1 Hz	>> 120s	Local plant control	Oscillation event	
	0.15 Hz – 1 Hz	<< 120s	-	Normal event	
Signature 3	1.0 Hz – 5.0 Hz	>> 60s	Inter-area oscillation	Oscillation event	
	1.0 Hz – 5.0 Hz	<< 60s	-	Normal event	
Signature 4	5.0 Hz – 14.0 Hz	>> 50s	Local plant control	Oscillation event	
	5.0 Hz – 14.0 Hz	<< 50s	-	Normal event	

3. Simulation, Results, and Discussion

With streaming PMU data, it is challenging to store the entire dataset and sectioning it into training, validation, and testing data in order to evaluate the THAT model. Thus, other evaluation techniques are considered including Holdout, interleaved test-then-train, and prequential. In Holdout, a section of the incoming data is used to train the model and small test cases were used to compute the performance. In interleaved test-and-train, each instance of the data stream is used for testing and training the model. Prequential is similar to interleaved test-and-train and uses a sliding window or a decaying factor. The proposed model (THAT) was evaluated using these three techniques and it was observed that the prequential technique exhibited higher detection accuracy with moderate processing time against other approaches.

In order to evaluate the performance of the THAT model, a comparative analysis with OzaBag has been carried out using several performance metrics. An MOA platform [26] was used to run the simulations. The ensemble approach (i.e., OzaBag) is selected, as some literature [17], [32] reported satisfactory results with power system data. The performance metrics used are accuracy, Kappa and evaluation time. The first metric evaluates the algorithm in terms of detecting accurately the instances of oscillation and normal events (Equation (29)). However, the accuracy may produce overly optimistic predictions with imbalanced classes. Kappa metrics can be used to resolve this issue and reduce instances that were incorrectly classified by chance (Equation (30)). Thus, Kappa is used to help in better tuning the THAT model. The evaluation time computes the required time to process each instance.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{29}$$

Where TP is the true positive, TN is the true negative, FP is the false positive, and FN is the false negative.

$$Kappa = \frac{(Accuracy - random\ accuracy\ )}{(1 - random\ accuracy)} \tag{30}$$

Where $Accuracy$ is given by Equation (29), and $random\ accuracy$ is defined as:

$$\begin{aligned} &random\ accuracy \\ &= \frac{(TN + FP) * (TN + FN) + (FN + TP) * (FP + TP)}{Total^2} \end{aligned} \tag{31}$$

Here $Total$ is $TP + TN + FN + FP$.

3.1. Experiment (I): THAT without supervised transfer learning

For this case, the THAT model was separately trained on the four signatures without applying supervised transfer learning. A parametric study of THAT and OzaBag models has been conducted in order to define the appropriate value of each hyper-parameter of these models.

For the THAT, two Information Gain functions have been explored: Gini index and Entropy, and different values of δ have been selected ranging from 0 to 1. The results of this experiment are given in Figures 28 to 31. These figures illustrate the experimental results of one signature. The results of the other three signatures are not included due to space constraints. Figure 28 illustrates the accuracy of the THAT model as a function of the number of instances with the Gini index and different δ values. As it can be seen, the accuracy increases sharply between 0 and 250 instances to reach the highest accuracy value, which is 99%, and stabilizes at that value. The accuracy decreases after the insertion of concept drifts and finally increases again to stabilize at 97%. In addition, one can notice that the model recovers better after gradual concept drift for values of δ greater than 0.2.

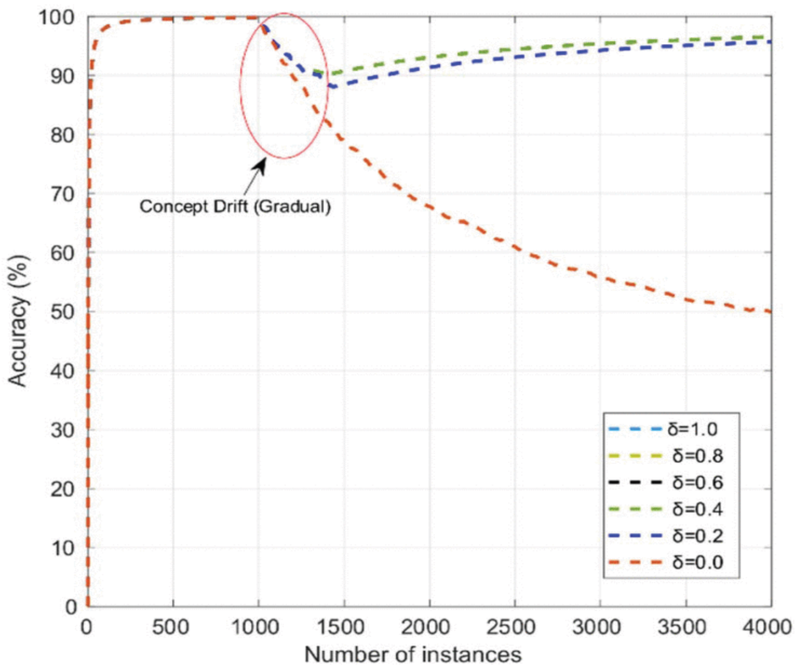


Figure 28. Accuracy vs Number of Instances for THAT
with Gini Index function and different δ values.

Figure 29 shows the accuracy of the THAT model as a function of the number of instances with the Information Gain and different δ values. It can be seen that these functions follow the trends similar to those of Figure 28, the model accuracy increases and reach the higher accuracy value, which is 99%, and it stabilizes at this value. Then it decreases after the concept drifts and reaches the lowest accuracy value of 72% with $\delta = 0.0$, and then it recovers and stabilizes at 97% after 3500 instances. Additionally, it is worth mentioning that all the highest accuracy values are reported when δ is set to 0.2 or higher value. As seen in Figures 28 and 29, THAT + Gini Index and THAT + Information Gain reported the same performance in terms of accuracy, especially with a δ value equal or higher than 0.2. However, the Information Gain is computationally demanding since it uses the logarithmic scale in computing the entropy of each feature. Thus, the Gini Index is selected for the next experiments.

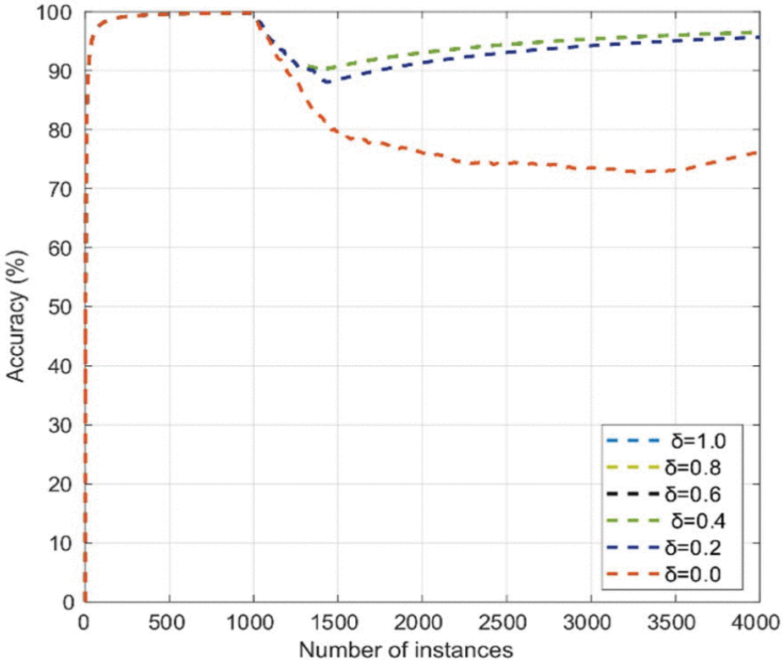


Figure 29. Accuracy Vs Number of Instances for THAT

with Information Gain and different δ values.

Figure 30 illustrates the Kappa value of the THAT model as a function of the number of instances with the Gini Index and different δ values. As can be seen, the Kappa value increases exponentially between 0 and 250 instances and stabilizes at 99%, and drops at

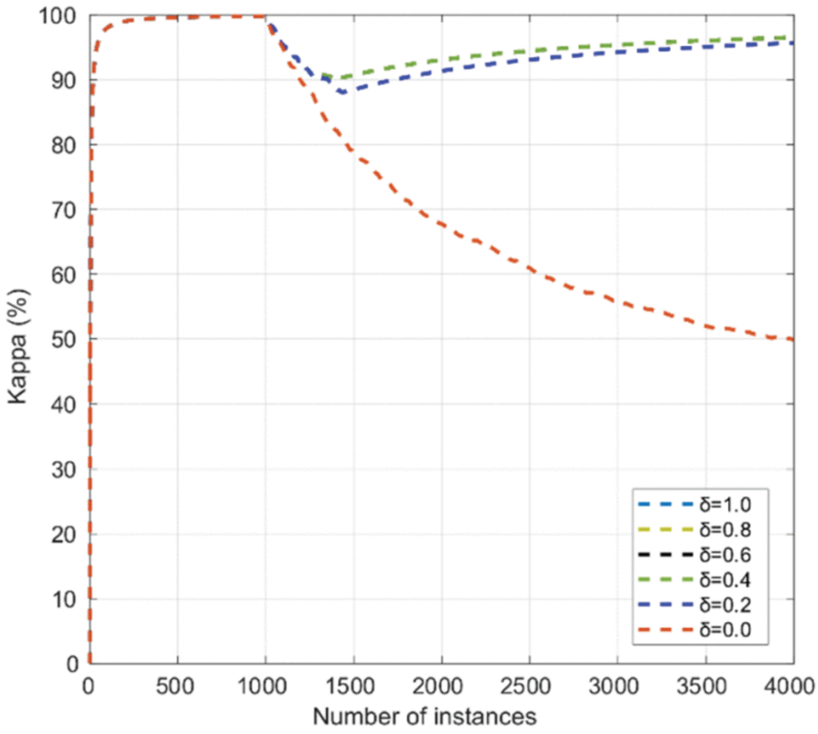


Figure 30. Kappa Vs Number of Instances for THAT with

Gini Index function and different δ values.

1000 instances, on the occurrence of concept drifts. After 1000 instances, the model recovers quickly for all δ values greater than 0.2. Additionally, one can notice that the Kappa function follows a similar trend as that of Figure 29 since the classes are equally distributed in the dataset (50% oscillation event and 50% normal event).

Figure 31 illustrates the evaluation time as a function of the number of instances for THAT with the Gini Index and different δ values. It can be seen that the evaluation time increases slightly between 0ms and 25ms with instances less than 1000 instances. After the concept drifts, the evaluation time increases sharply with all δ values which are equal to or greater than 0.2. The highest evaluation time is reported by δ=1 and the lowest value is reported with δ= 0. The reported results suggest that the optimal THAT performance is

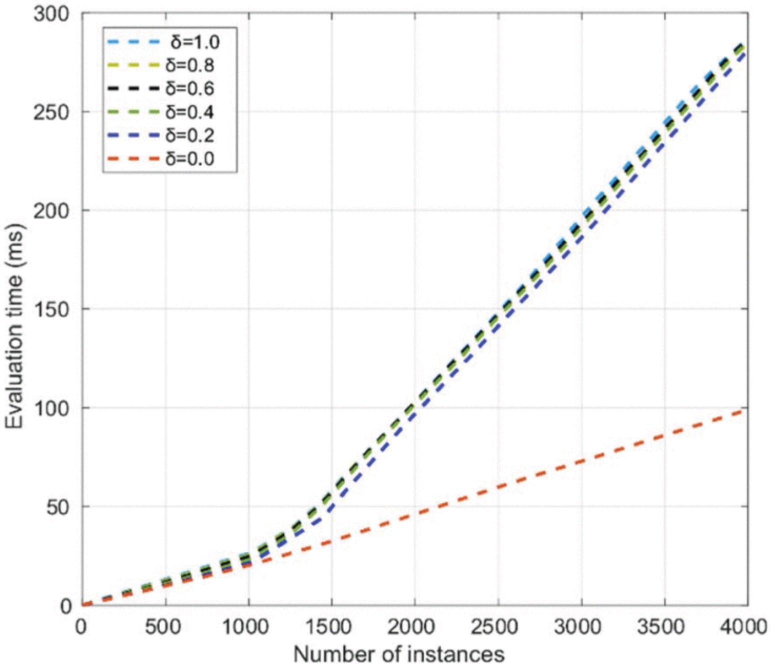


Figure 31. Evaluation time Vs Number of Instances for THAT with Gini Index function and different δ values.

achieved by the Gini Index function and a δ value which is equal to or greater than 0.2. In addition, the THAT model with a δ greater than 0.4 is slightly demanding in terms of evaluation time. By setting the δ to 0.2, a tradeoff can be made between high accuracy and optimal evaluation time. Regarding OzaBag, a parametric study has been conducted in terms of the number of HTs: 5, 10, 15, and 20 HTs. The obtained results showed that better performance is achieved by choosing 5 HTs. Increasing the number of HTs does not increase significantly the OzaBag model accuracy but it increases the evaluation time. Thus, OzaBag with 5 HTs is the optimal number in terms of accuracy, Kappa, and evaluation time.

After defining the appropriate parameters for THAT and OzaBag, the models are trained on the four signatures and the corresponding results are given in Figures 32 through 34. Figure 32 shows a comparison between THAT and OzaBag in terms of accuracy based on one signature. It is seen that the accuracy of the two models is similar between 0 and 1000 instances. Both instances then reach a peak value of 99% then stabilizes at that value. models' accuracy increases sharply between 0 and 250. After the concept drifts, the accuracy of the models drops to 96% and 88% for THAT and OzaBag, respectively. In addition, one can notice that the THAT model recovers quickly after the concept drift events and increases slightly to reach 98% with 1999 instances. At 2000 instances, the accuracy of the THAT model decreases slightly due to another concept drift, but it recovers and regains its previous accuracy value. On the other hand, the accuracy of the OzaBag model is affected by concept drifts. Later, it increases slightly and decreases again after the second concept drift before reaching a final accuracy value of 90% at 4000 instances.

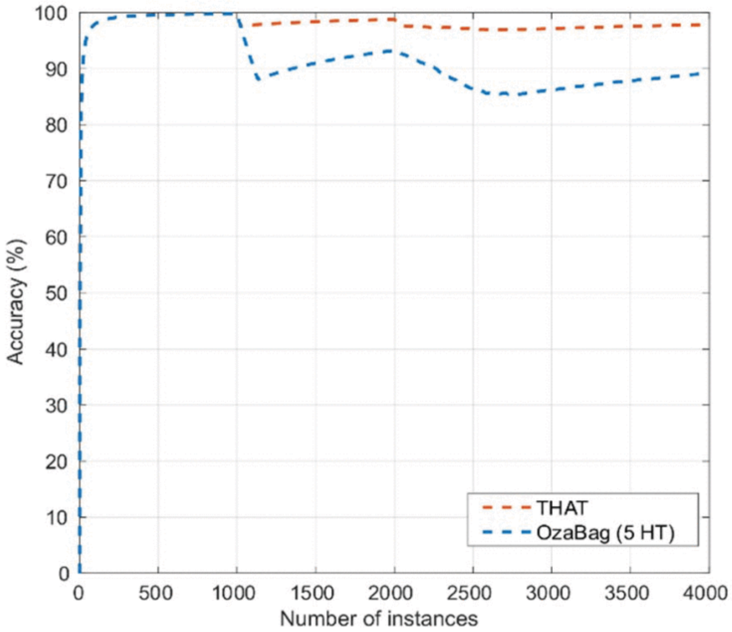


Figure 32. THAT Vs OzaBag in terms of accuracy as function of the number of instances.

Figure 33 illustrates a comparison between the THAT and OzaBag in terms of the evaluation time. It is observed that the evaluation time of OzaBag increases exponentially as the number of instances increases, while the growth is linear for the THAT model. For instance, OzaBag needs 500 ms to process 2000 instances and 2150 ms to process 4000 instances. On the other hand, THAT requires 80 ms for processing 2000 instances and 230 ms for processing 4000 instances. This can be explained by the fact that OzaBag is an ensemble approach that uses several models (e.g., 5 HTs), and training all these models requires a longer processing time. However, the THAT model only includes one HT model with Gini Index and hence does not demand larger run-time.

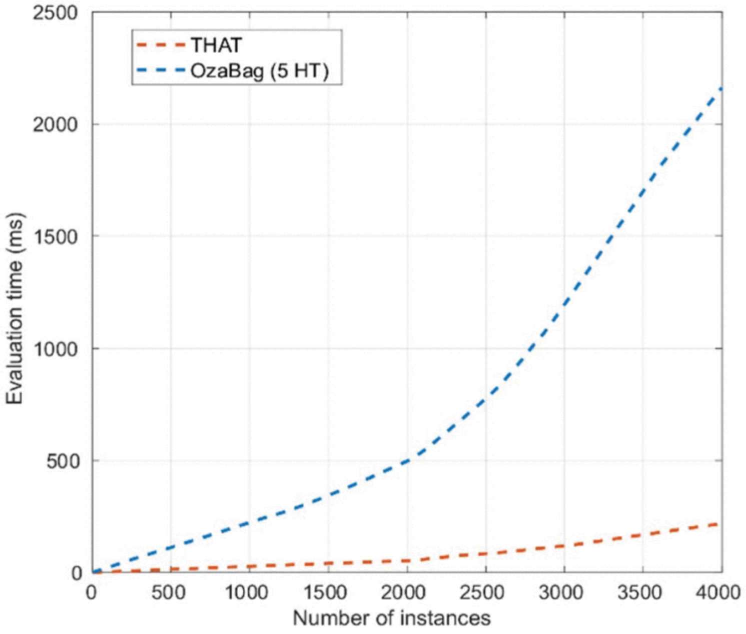


Figure 33. THAT Vs OzaBag in terms of evaluation time as a function of the number of instances.

By using the prequential technique, accuracy is computed for each instance. In order to provide an overall accuracy of different models, an average accuracy can be considered which is computed by summing up the accuracy of each instance and dividing the results by the total number of instances. The average accuracy is computed after training the models on the four signatures discussed in Section II and the results are given in Figure 34. As can be seen, the two models reported similar accuracy, which is 94%, for the first signature. For the second signature, the THAT model outperforms OzaBag with a difference of 6%; an accuracy of 97% is reported by THAT while OzaBag's is at 91%. For the third signature, the THAT model is performing slightly better (99%) than OzaBag (98.5%) For the fourth signature, both models reported an accuracy value of 99%.

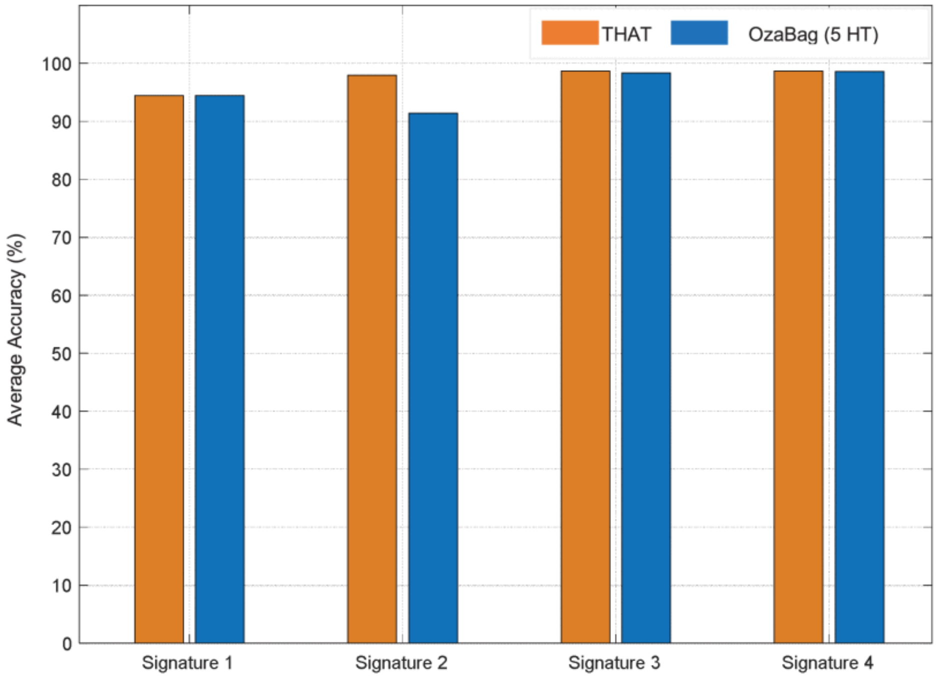


Figure 34. THAT Vs OzaBag in terms of average accuracy.

3.2. Experiment(II): THAT with supervised transfer learning

Here, the THAT model has been trained on four signatures using the supervised transfer learning concept. Examples of the obtained results are given in Table 7. It is noted that both the THAT and the OzaBag models report the same average accuracy of 94%. One can notice that the accuracy of both models has decreased when compared to the first experiment. This is due to the fact that in the first set of experiments, the models have been

Table 7. Comparison between THAT model and OzaBag.

Data stream models	Average accuracy	Evaluation time/instance
THAT model with supervised transfer learning	94%	0.34s
OzaBag (5 HT)	94%	1.04s

trained on four signatures separately, while in the second case, these models were trained on four signatures at the same time. Usually, when the models become more generalized, their accuracy decreases. In terms of the evaluation time, the THAT model with transfer learning requires a lesser evaluation time than the OzaBag. THAT takes 0.34ms to classify a given instance, while OzaBag takes 1.04ms. A typical PMU with a data rate of 120 samples/s requires 8.33ms to process one sample. Thus, the obtained results suggest that THAT is more suitable for the PMU streaming data in terms of detecting accurately the faults events in near-real-time.

4. Conclusions

A transfer learning technique using the Hoeffding tree and ADWIN is proposed for synchrophasor data. The proposed model, called THAT, can be easily trained for any PMU signatures of varying and shorter durations. It does not require loading the entire data into memory to build the decision tree model and is thus suitable for real-time processing. Additionally, ADWIN is included, so the THAT model is easily adaptable to gradual concept drifts. A prequential technique was used to evaluate the performance of the model and the results have been compared to the OzaBag method. After performing a parametric study and tuning the models, two sets of experiments have been conducted. In the first case, the THAT model has been trained separately on each signature. In the second case, supervised transfer learning has been applied to the THAT model. The obtained results showed that the THAT and OzaBag models report higher average accuracy ranging between 91% and 99% (case 1). For case 2, the average accuracy was decreased to 94% in both models, but the THAT model required smaller computational run-time than OzaBag.

Further, the results suggested that increasing the number of HT in the OzaBag model did not significantly improve the overage accuracy, but it did increase processing time considerably. Thus, the THAT model is more suitable than the OzaBag model for the PMU data stream, since it provides a near-real-time response to the dynamic fault event conditions.

Chapter VI

Conclusions and Future Work

In this dissertation, we aimed at consolidating the grid's security posture by mining multi-variate large-scale synchrophasor data to reveal potential physical and cyber anomalies. Particularly, chapter II provides a holistic review of the existing cyber and physical anomalies in the Smart Grid and their impact on the different subsystems' security. Additionally, it critically reviewed in depth the existing detection approaches and analyzed their strengths and limitations. An ANN-based approach is presented in chapter III to detect FDI attacks; two attack scenarios are used to model the FDI attack using sigmoid and trapezoidal membership functions. A falsified dataset is used in ANN for training and testing; during the training phase, three types of activation functions are explored: Relu, Sigmoid, and Tanh functions. Based on simulation results, it is concluded that ANN with the Relu activation function and 100 neuron nodes detects the falsified injected data with an accuracy of 99%, and it outperforms SVM in terms of probability of detection, and probability of miss detection. However, the RF with 100 trees exhibits a low Pfa, which is 0.2% followed by ANN with 0.9%. In Chapter IV, we developed an RFR-based approach for determining, not only anomalies but also their respective location and duration. We trained the model on several fault scenarios simulated on GridPACK; a total of nine fault scenarios were simulated by injecting faults on specific buses over a specified time period. Four study cases were used to train and evaluate the RFR models: detecting fault location, predicting fault duration, handling missing data, and streaming data. The obtained results showed that RFR outperformed several state-of-the-art models in terms of accuracy, MSE, and processing time. Results indicate that RFR is capable of detecting the location and duration of a fault with an accuracy of 84%. The RFR model consistently outperformed all other models, and it can therefore be utilized in real-time situational awareness systems to determine both the location and duration of faults while coping with missing data.

In chapter V, we extended the RFR, by developing an online machine learning model suitable for the streaming synchrophasor data. THAT, the proposed model, can be easily trained for PMU signatures of different durations and duration variations. The model does not require loading the entire data into memory in order to build the decision tree model and is thus suitable for real-time processing. Additionally, ADWIN is included, so the THAT model can be easily adapted to gradual concept drifts. The THAT model's performance was evaluated and compared to the OzaBag method. Two sets of experiments have been conducted after performing a parametric study and tuning the models. The model in the first case has been trained separately on each signature; in the second case, supervised transfer learning has been applied to the THAT model. Based on the obtained results, the THAT and OzaBag models reported higher average accuracy levels ranging from 91% to 99% (case 1). For case 2, the average accuracy declined to 94% in both models, but the THAT model required a smaller run-time than OzaBag. That model is therefore more appropriate for the PMU data stream since it provides a near-real-time response to dynamic fault event conditions.

In summary, this dissertation aims to provide early restoration, enhanced resilience, and improved observability of the Smart Grid network through modeling and detecting cyber and physical anomalies; these are the main contributions:

- Conducting a state-of-the-art review on anomalies in Smart Grid and their respective detection approaches.
- Modeling False Data Injection (FDI) attacks and developing an Artificial Neural Network (ANN) detection approach.
- Developing Random Forest Regressor (RFR) model to detect fault locations and predict their duration.
- Developing a Transfer Adaptive Hoeffding Tree (THAT) for detecting anomalies in streaming PMU data in real-time.

One of the challenges encountered while conducting this research was the lack of reliable and comprehensive PMU data containing physical and cyber events. In light of this, one of the future directions of this research will be to establish a testbed using a Real-Time Digital Simulator (RTDS) to simulate more realistic cyber and physical events and assess our model's efficiency and scalability. A further extension of this work would be to realize the deployment of these models in hardware devices such as FPGAs at the PMU level to overcome the computational burden at the PDC level. This would assist in inspecting traffic, scanning it rapidly, and detecting potential anomalies in real-time.

References

[1] D. U. Case, “Analysis of the cyber attack on the Ukrainian power grid,” *Electr Inf Shar Anal Cent E-ISAC*, 2016.

[2] “SECURITY: Experts assess damage after first cyberattack on U.S. grid.” https://www.eenews.net/stories/1060281821 (accessed Oct. 29, 2019).

[3] “Hackers Breached Colonial Pipeline Using Compromised Password,” *Bloomberg.com*, Jun. 04, 2021. Accessed: Feb. 15, 2022. [Online]. Available: https://www.bloomberg.com/news/articles/2021-06-04/hackers-breached-colonial-pipeline-using-compromised-password

[4] “Cyber attack on U.S. power grid could cost economy $1 trillion: report,” *Reuters*, Jul. 08, 2015. Accessed: Feb. 20, 2020. [Online]. Available: https://www.reuters.com/article/us-cyberattack-power-survey-idUSKCN0PI0XS20150708

[5] Z. Zhang, S. Gong, A. D. Dimitrovski, and H. Li, “Time Synchronization Attack in Smart Grid: Impact and Analysis,” *IEEE Trans. Smart Grid*, vol. 4, no. 1, pp. 87–98, Mar. 2013, doi: 10.1109/TSG.2012.2227342.

[6] J. Zhao, J. Wang, and L. Yin, “Detection and Control against Replay Attacks in Smart Grid,” in *12th International Conference on Computational Intelligence and Security*, China, 2016, pp. 624–627.

[7] P. Yi, T. Zhu, Q. Zhang, Y. Wu, and J. Li, “A denial of service attack in advanced metering infrastructure network,” in *2014 IEEE International Conference on Communications (ICC)*, 2014, pp. 1029–1034.

[8] Y. Yang *et al.*, “Man-in-the-middle attack test-bed investigating cyber-security vulnerabilities in Smart Grid SCADA systems,” in *Sustainable Power Generation and Supply (SUPERGEN 2012), International Conference on*, 2012, pp. 1–8.

[9] Z. Lu, W. Wang, and C. Wang, *Modeling and Evaluating Denial of Service Attacks for Wireless and Mobile Applications*. Springer, 2015.

[10] Z. El Mrabet, N. Kaabouch, H. El Ghazi, and H. El Ghazi, “Cyber-security in smart grid: Survey and challenges,” *Comput. Electr. Eng.*, vol. 67, pp. 469–482, 2018.

[11] M. Esmalifalak, L. Liu, N. Nguyen, R. Zheng, and Z. Han, “Detecting Stealthy False Data Injection Using Machine Learning in Smart Grid,” *IEEE Syst J*, vol. 11, no. 3, pp. 1644-1652, Sep. 2017.

[12] G. Liang, S. R. Weller, J. Zhao, F. Luo, and Z. Y. Dong, “The 2015 Ukraine Blackout: Implications for False Data Injection Attacks,” *IEEE Trans Power Syst*, vol. 32, no. 4, pp. 3317-3318, Jul. 2017.

[13] D. Lee and D. Kundur, “Cyber attack detection in PMU measurements via the expectation-maximization algorithm,” in *2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Dec. 2014, pp. 223–227. doi: 10.1109/GlobalSIP.2014.7032111.

[14] X. Fan, L. Du, and D. Duan, “Synchrophasor Data Correction Under GPS Spoofing Attack: A State Estimation Based Approach,” in *2018 IEEE/PES Transmission and Distribution Conference and Exposition (T D)*, Apr. 2018, pp. 1–9. doi: 10.1109/TDC.2018.8440488.

[15] S. Alison, “SYNCHROPHASORS & THE GRID.” North American Synchrophasor Initiative, 2017. [Online]. Available:

https://www.naspi.org/sites/default/files/reference_documents/naspi_naruc_silverstein_20170714.pdf

[16] N. Framework, "Roadmap for Smart Grid Interoperability Standards, Release 2.0 (2012)," *NIST Spec. Publ.*, vol. 1108.

[17] G. W. Arnold *et al.*, "NIST Framework and Roadmap for Smart Grid Interoperability Standards, Release 1.0," 2010.

[18] V. C. Gungor *et al.*, "A survey on smart grid potential applications and communication requirements," *IEEE Trans. Ind. Inform.*, vol. 9, no. 1, pp. 28–42, 2013.

[19] E. D. Knapp and R. Samani, *Applied cyber security and the smart grid: implementing security controls into the modern power infrastructure*. Amsterdam: Elsevier, Syngress, 2013.

[20] C. Chapman, P. Guan, T. LeRoy, E. Miller, and D. Park, "Wide Area Measurement System Utilizing Open Source Tools," *Proc. Natl. Conf. Undergrad. Res. NCUR*, p. 9, 2013.

[21] E. Jamil, M. Rihan, and M. A. Anees, "Towards optimal placement of phasor measurement units for smart distribution systems," in *2014 6th IEEE Power India International Conference (PIICON)*, Dec. 2014, pp. 1–6. doi: 10.1109/POWERI.2014.7117773.

[22] B. Appasani and D. K. Mohanta, "A review on synchrophasor communication system: communication technologies, standards and applications," *Prot. Control Mod. Power Syst.*, vol. 3, no. 1, Dec. 2018, doi: 10.1186/s41601-018-0110-4.

[23] H. Gharavi and B. Hu, "Synchrophasor Sensor Networks for Grid Communication and Protection," *Proc. IEEE Inst. Electr. Electron. Eng.*, vol. 105, no. 7, pp. 1408–1428, Jul. 2017, doi: 10.1109/JPROC.2017.2696881.

[24] “IEEE Standard for Synchrophasors for Power Systems,” *IEEE Std C37118-2005 Revis. IEEE Std 1344-1995*, pp. 1–65, Mar. 2006, doi: 10.1109/IEEESTD.2006.99376.

[25] M. A. Faisal, Z. Aung, J. R. Williams, and A. Sanchez, “Data-Stream-Based Intrusion Detection System for Advanced Metering Infrastructure in Smart Grid: A Feasibility Study,” *IEEE Syst. J.*, vol. 9, no. 1, pp. 31–44, Mar. 2015, doi: 10.1109/JSYST.2013.2294120.

[26] D. Choi, S. Lee, D. Won, and S. Kim, “Efficient secure group communications for SCADA,” *IEEE Trans. Power Deliv.*, vol. 25, no. 2, pp. 714–722, 2010.

[27] M. Z. Gunduz and R. Das, “Cyber-security on smart grid: Threats and potential solutions,” *Comput. Netw.*, vol. 169, p. 107094, 2020.

[28] F. M. Cleveland, “Cyber security issues for Advanced Metering Infrasttructure (AMI),” Jul. 2008, pp. 1–5. doi: 10.1109/PES.2008.4596535.

[29] J. Liu, Y. Xiao, and J. Gao, “Achieving accountability in smart grid,” *IEEE Syst. J.*, vol. 8, no. 2, pp. 493–508, 2014.

[30] B. Tekinerdogan, D. Blouin, H. Vangheluwe, M. Goulão, P. Carreira, and V. Amaral, *Multi-Paradigm Modelling Approaches for Cyber-Physical Systems*. Academic Press, 2020.

[31] H. Zhang, B. Liu, and H. Wu, “Smart Grid Cyber-Physical Attack and Defense: A Review,” *IEEE Access*, vol. 9, pp. 29641–29659, 2021, doi: 10.1109/ACCESS.2021.3058628.

[32] H.-H. Hsu, C.-Y. Chang, and C.-H. Hsu, *Big data analytics for sensor-network collected intelligence*. Morgan Kaufmann, 2017.

[33] M. Zeller, “Myth or reality — Does the Aurora vulnerability pose a risk to my generator?,” in *2011 64th Annual Conference for Protective Relay Engineers*, Apr. 2011, pp. 130–136. doi: 10.1109/CPRE.2011.6035612.

[34] S. S. Gururajapathy, H. Mokhlis, and H. A. Illias, “Fault location and detection techniques in power distribution systems with distributed generation: A review,” *Renew. Sustain. Energy Rev.*, vol. 74, pp. 949–958, 2017.

[35] G. A. Ajenikoko and S. O. Sangotola, “An overview of impedance-based fault location techniques in electrical power-transmission network,” *Int. J. Adv. Eng. Res. Appl. IJA-ERA*, vol. 2, no. 3, pp. 2454–2377, 2016.

[36] P. K. Lim and D. S. Dorr, “Understanding and resolving voltage sag related problems for sensitive industrial customers,” in *2000 IEEE Power Engineering Society Winter Meeting. Conference Proceedings (Cat. No. 00CH37077)*, 2000, vol. 4, pp. 2886–2890.

[37] M. Al Karim, M. Chenine, K. Zhu, L. Nordstrom, and L. Nordström, “Synchrophasor-based data mining for power system fault analysis,” in *2012 3rd IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe)*, Oct. 2012, pp. 1–8. doi: 10.1109/ISGTEurope.2012.6465843.

[38] Y. Liu, P. Ning, and M. K. Reiter, “False data injection attacks against state estimation in electric power grids,” *ACM Trans. Inf. Syst. Secur. TISSEC*, vol. 14, no. 1, p. 13, 2011.

[39] 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 Trans. Smart Grid*, vol. 8, no. 4, pp. 1630–1638, Jul. 2017, doi: 10.1109/TSG.2015.2495133.

[40] G. Hug and J. A. Giampapa, "Vulnerability Assessment of AC State Estimation With Respect to False Data Injection Cyber-Attacks," *IEEE Trans. Smart Grid*, vol. 3, no. 3, pp. 1362–1370, Sep. 2012, doi: 10.1109/TSG.2012.2195338.

[41] J. Zhao, L. Mili, and M. Wang, "A Generalized False Data Injection Attacks Against Power System Nonlinear State Estimator and Countermeasures," *IEEE Trans. Power Syst.*, vol. 33, no. 5, pp. 4868–4877, Sep. 2018, doi: 10.1109/TPWRS.2018.2794468.

[42] J. Kim, L. Tong, and R. J. Thomas, "Subspace Methods for Data Attack on State Estimation: A Data Driven Approach," *IEEE Trans. Signal Process.*, vol. 63, no. 5, pp. 1102–1114, Mar. 2015, doi: 10.1109/TSP.2014.2385670.

[43] R. Tan, V. Badrinath Krishna, D. K. Yau, and Z. Kalbarczyk, "Impact of integrity attacks on real-time pricing in smart grids," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, 2013, pp. 439–450.

[44] S. Maharjan, Q. Zhu, Y. Zhang, S. Gjessing, and T. Basar, "Dependable demand response management in the smart grid: A Stackelberg game approach," *IEEE Trans. Smart Grid*, vol. 4, no. 1, pp. 120–132, 2013.

[45] "Evaluation of the vulnerability of phasor measurement units to GPS spoofing attacks," *Int. J. Crit. Infrastruct. Prot.*, vol. 5, no. 3–4, pp. 146–153, Dec. 2012, doi: 10.1016/j.ijcip.2012.09.003.

[46] Z. Lu, W. Wang, and C. Wang, "From jammer to gambler: Modeling and detection of jamming attacks against time-critical traffic," in *INFOCOM, 2011 Proceedings IEEE*, 2011, pp. 1871–1879.

[47] K. Gai, M. Qiu, Z. Ming, H. Zhao, and L. Qiu, "Spoofing-Jamming Attack Strategy Using Optimal Power Distributions in Wireless Smart Grid Networks," *IEEE Trans. Smart Grid*, pp. 1–1, 2017, doi: 10.1109/TSG.2017.2664043.

[48] F. Aloul, A. R. Al-Ali, R. Al-Dalky, M. Al-Mardini, and W. El-Hajj, “Smart grid security: Threats, vulnerabilities and solutions,” *Int. J. Smart Grid Clean Energy*, vol. 1, no. 1, pp. 1–6, 2012.

[49] M. Bristow, “ModScan: a SCADA Modbus network scanner,” 2008.

[50] N. R. Rodofile, K. Radke, and E. Foo, “DNP3 Network Scanning and Reconnaissance for Critical Infrastructure,” in *Proceedings of the Australasian Computer Science Week Multiconference*, New York, NY, USA, 2016, p. 39:1-39:10. doi: 10.1145/2843043.2843350.

[51] P. Maynard, K. McLaughlin, and B. Haberler, “Towards Understanding Man-In-The-Middle Attacks on IEC 60870-5-104 SCADA Networks,” Sep. 2014. doi: 10.14236/ewic/ics-csr2014.5.

[52] I. Darwish, O. Igbe, O. Celebi, T. Saadawi, and J. Soryal, “Smart Grid DNP3 Vulnerability Analysis and Experimentation,” Nov. 2015, pp. 141–147. doi: 10.1109/CSCloud.2015.86.

[53] A. A. Cárdenas, S. Amin, and S. Sastry, “Research challenges for the security of control systems,” in *Proceedings of the 3rd conference on Hot topics in security*, USA, Jul. 2008, pp. 1–6.

[54] J. Qin, M. Li, L. Shi, and X. Yu, “Optimal Denial-of-Service Attack Scheduling With Energy Constraint Over Packet-Dropping Networks,” *IEEE Trans. Autom. Control*, vol. 63, no. 6, pp. 1648–1663, Jun. 2018, doi: 10.1109/TAC.2017.2756259.

[55] A. Sargolzaei, K. Yen, and M. Abdelghani, “Delayed inputs attack on load frequency control in smart grid,” in *ISGT 2014*, Feb. 2014, pp. 1–5. doi: 10.1109/ISGT.2014.6816508.

[56] D. Kushner, “The real story of stuxnet,” *IEEE Spectr.*, vol. 50, no. 3, pp. 48–53, Mar. 2013, doi: 10.1109/MSPEC.2013.6471059.

[57] G. Ma, L. Jiang, K. Zhou, and G. Xu, “A Method of line fault location based on traveling wave theory,” *Int. J. Control Autom.*, vol. 9, 2016.

[58] Y. Zhang, L. Wang, W. Sun, R. C. G. II, and M. Alam, “Distributed Intrusion Detection System in a Multi-Layer Network Architecture of Smart Grids,” *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 796–808, Dec. 2011, doi: 10.1109/TSG.2011.2159818.

[59] S. F. Alwash, V. K. Ramachandaramurthy, and N. Mithulananthan, “Fault-location scheme for power distribution system with distributed generation,” *IEEE Trans. Power Deliv.*, vol. 30, no. 3, pp. 1187–1195, 2014.

[60] S. A. M. Javadian, A. M. Nasrabadi, M.-R. Haghifam, and J. Rezvantalab, “Determining fault’s type and accurate location in distribution systems with DG using MLP Neural networks,” in *2009 International Conference on Clean Electrical Power*, 2009, pp. 284–289.

[61] D. M. Menon and N. Radhika, “Anomaly detection in smart grid traffic data for home area network,” in *2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT)*, Mar. 2016, pp. 1–4. doi: 10.1109/ICCPCT.2016.7530186.

[62] C. R. Rao and V. N. Gudivada, *Computational analysis and understanding of natural languages: principles, methods and applications*. Elsevier, 2018.

[63] P. Skoda and F. Adam, *Knowledge Discovery in Big Data from Astronomy and Earth Observation: Astrogeoinformatics*. Elsevier, 2020.

[64] S. Armoogum and X. Li, “Big data analytics and deep learning in bioinformatics with hadoop,” in *Deep learning and parallel computing environment for bioengineering systems*, Elsevier, 2019, pp. 17–36.

[65] M. Naeem, S. T. H. Rizvi, and A. Coronato, “A Gentle Introduction to Reinforcement Learning and its Application in Different Fields,” *IEEE Access*, vol. 8, pp. 209320–209344, 2020, doi: 10.1109/ACCESS.2020.3038605.

[66] Z. Ni, S. Paul, X. Zhong, and Q. Wei, “A reinforcement learning approach for sequential decision-making process of attacks in smart grid,” in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2017, pp. 1–8.

[67] Y. Aslan, “An alternative approach to fault location on power distribution feeders with embedded remote-end power generation using artificial neural networks,” *Electr. Eng.*, vol. 94, no. 3, pp. 125–134, 2012.

[68] F. Dehghani and H. Nezami, “A new fault location technique on radial distribution systems using artificial neural network,” 2013.

[69] W. Li, D. Deka, M. Chertkov, and M. Wang, “Real-time faulted line localization and pmu placement in power systems through convolutional neural networks,” *IEEE Trans. Power Syst.*, vol. 34, no. 6, pp. 4640–4651, 2019.

[70] A. Takiddin, M. Ismail, U. Zafar, and E. Serpedin, “Deep Autoencoder-Based Anomaly Detection of Electricity Theft Cyberattacks in Smart Grids,” *IEEE Syst. J.*, pp. 1–12, 2022, doi: 10.1109/JSYST.2021.3136683.

[71] S. R. Madeti and S. N. Singh, “Modeling of PV system based on experimental data for fault detection using kNN method,” *Sol. Energy*, vol. 173, pp. 139–151, 2018.

[72] U. Adhikari, T. H. Morris, and S. Pan, “Applying Hoeffding Adaptive Trees for Real-Time Cyber-Power Event and Intrusion Classification,” *IEEE Trans. Smart Grid*, vol. 9, no. 5, pp. 4049–4060, Sep. 2018, doi: 10.1109/TSG.2017.2647778.

[73] N. Dahal, O. Abuomar, R. King, and V. Madani, “Event stream processing for improved situational awareness in the smart grid,” *Expert Syst. Appl.*, vol. 42, no. 20, pp. 6853–6863, 2015.

[74] NASPI Control Room Solutions Task Team Paper, “Using Synchrophasor Data forOscillation Detection,” *Oct. 2017.*

[75] M. Tian, Z. Dong, and X. Wang, “Analysis of false data injection attacks in power systems: A dynamic Bayesian game-theoretic approach,” *ISA Trans.*, vol. 115, pp. 108–123, Sep. 2021, doi: 10.1016/j.isatra.2021.01.011.

[76] M. A. Teixeira, T. Salman, M. Zolanvari, R. Jain, N. Meskin, and M. Samaka, “SCADA System Testbed for Cybersecurity Research Using Machine Learning Approach,” *Future Internet*, vol. 10, no. 8, Art. no. 8, Aug. 2018, doi: 10.3390/fi10080076.

[77] W. Xue and T. Wu, “Active Learning-Based XGBoost for Cyber Physical System Against Generic AC False Data Injection Attacks,” *IEEE Access*, vol. 8, pp. 144575–144584, 2020, doi: 10.1109/ACCESS.2020.3014644.

[78] Z. Qu, H. Liu, Z. Wang, J. Xu, P. Zhang, and H. Zeng, “A combined genetic optimization with AdaBoost ensemble model for anomaly detection in buildings electricity consumption,” *Energy Build.*, vol. 248, p. 111193, Oct. 2021, doi: 10.1016/j.enbuild.2021.111193.

[79] A. Zainab, S. S. Refaat, D. Syed, A. Ghrayeb, and H. Abu-Rub, “Faulted Line Identification and Localization in Power System using Machine Learning

Techniques," in *2019 IEEE International Conference on Big Data (Big Data)*, 2019, pp. 2975–2981.

[80] H. Okumus and F. M. Nuroglu, "A random forest-based approach for fault location detection in distribution systems," *Electr. Eng.*, vol. 103, no. 1, pp. 257–264, Feb. 2021, doi: 10.1007/s00202-020-01074-8.

[81] S. Pandey, A. Srivastava, and B. Amidan, "A Real Time Event Detection, Classification and Localization using Synchrophasor Data," *IEEE Trans. Power Syst.*, pp. 1–1, 2020, doi: 10.1109/TPWRS.2020.2986019.

[82] R. Qi, C. Rasband, J. Zheng, and R. Longoria, "Detecting Cyber Attacks in Smart Grids Using Semi-Supervised Anomaly Detection and Deep Representation Learning," *Information*, vol. 12, no. 8, Art. no. 8, Aug. 2021, doi: 10.3390/info12080328.

[83] A. J. Abianeh, Y. Wan, F. Ferdowsi, N. Mijatovic, and T. Dragičević, "Vulnerability Identification and Remediation of FDI Attacks in Islanded DC Microgrids Using Multiagent Reinforcement Learning," *IEEE Trans. Power Electron.*, vol. 37, no. 6, pp. 6359–6370, Jun. 2022, doi: 10.1109/TPEL.2021.3132028.

[84] M. N. Kurt, O. Ogundijo, C. Li, and X. Wang, "Online Cyber-Attack Detection in Smart Grid: A Reinforcement Learning Approach," *IEEE Trans. Smart Grid*, vol. 10, no. 5, pp. 5174–5185, Sep. 2019, doi: 10.1109/TSG.2018.2878570.

[85] S. Ekici, "Support Vector Machines for classification and locating faults on transmission lines," *Appl. Soft Comput.*, vol. 12, no. 6, pp. 1650–1658, 2012.

[86] D.-I. Kim, A. White, and Y.-J. Shin, "PMU-Based Event Localization Technique for Wide-Area Power System," *IEEE Trans. Power Syst.*, vol. 33, no. 6, pp. 5875–5883, Nov. 2018, doi: 10.1109/TPWRS.2018.2824851.

[87] A. Hossam-Eldin, A. Lotfy, M. Elgamal, and M. Ebeed, “Combined traveling wave and fuzzy logic based fault location in multi-terminal HVDC systems,” in *2016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC)*, 2016, pp. 1–6.

[88] Y. Mohammadnian, T. Amraee, and A. Soroudi, “Fault detection in distribution networks in presence of distributed generations using a data mining–driven wavelet transform,” *IET Smart Grid*, vol. 2, no. 2, pp. 163–171, 2019.

[89] O. Olawumi, K. Haataja, M. Asikainen, N. Vidgren, and P. Toivanen, “Three practical attacks against ZigBee security: Attack scenario definitions, practical experiments, countermeasures, and lessons learned,” in *2014 14th International Conference on Hybrid Intelligent Systems*, Dec. 2014, pp. 199–206. doi: 10.1109/HIS.2014.7086198.

[90] D. Sadhukhan and S. V. Rao, “Minimum cost event driven WSN with spatial differentiated QoS requirements,” *Wirel Netw*, Jan. 2019.

[91] D. Sadhukhan and S. V. Rao, “Effect of Clock Skew in Event Driven, Delay Constrained Heterogeneous WSN with Anycast,” *Wirel Commun*, vol. 97, no. 4, pp. 4967-4980, Dec. 2017.

[92] R. Deng, G. Xiao, R. Lu, H. Liang, and A. V. Vasilakos, “False Data Injection on State Estimation in Power Systems—Attacks, Impacts, and Defense: A Survey,” *IEEE Trans Ind Inf.*, vol. 13, no. 2, pp. 411-423, Apr. 2017.

[93] S. Bi and Y. J. Zhang, “False-data injection attack to control real-time price in electricity market,” in *IEEE Global Communications Conference (GLOBECOM*, 2013, pp. 772–777.

[94] O. Kosut, L. Jia, R. J. Thomas, and L. Tong, “Malicious Data Attacks on the Smart Grid,” *IEEE Trans Smart Grid*, vol. 2, no. 4, pp. 645-658, Dec. 2011.

[95] O. Kosut, L. Jia, R. J. Thomas, and L. Tong, “Limiting false data attacks on power system state estimation,” in *44th Annual Conference on Information Sciences and Systems (CISS*, 2010, pp. 1–6.

[96] O. Kosut, L. Jia, R. J. Thomas, and L. Tong, “On malicious data attacks on power system state estimation,” in *45th International Universities Power Engineering Conference UPEC2010*, 2010, pp. 1–6.

[97] L. Xie, Y. Mo, and B. Sinopoli, “False Data Injection Attacks in Electricity Markets,” in *2010 First IEEE International Conference on Smart Grid Communications*, 2010, pp. 226–231.

[98] L. Xie, Y. Mo, and B. Sinopoli, “Integrity Data Attacks in Power Market Operations,” *IEEE Trans Smart Grid*, vol. 2, no. 4, pp. 659-666, Dec. 2011.

[99] 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 Trans. Smart Grid*, vol. 8, no. 5, pp. 2505–2516, Sep. 2017, doi: 10.1109/TSG.2017.2703842.

[100] G. Chaojun, P. Jirutitijaroen, and M. Motani, “Detecting False Data Injection Attacks in AC State Estimation,” *IEEE Trans Smart Grid*, vol. 6, no. 5, pp. 2476-2483, Sep. 2015.

[101] T. T. Kim and H. V. Poor, “Strategic Protection Against Data Injection Attacks on Power Grids,” *IEEE Trans Smart Grid*, vol. 2, no. 2, pp. 326-333, Jun. 2011.

[102] S. Bi and Y. J. Zhang, “Graphical Methods for Defense Against False-Data Injection Attacks on Power System State Estimation,” *IEEE Trans Smart Grid*, vol. 5, no. 3, pp. 1216-1227, May 2014.

[103] A. Anwar, A. N. Mahmood, and Z. Tari, “Identification of vulnerable node clusters against false data injection attack in an AMI based Smart Grid,” *Inf. Syst.*, vol. 53, pp. 201–212, Oct. 2015, doi: 10.1016/j.is.2014.12.001.

[104] M. Muratori, M. C. Roberts, R. Sioshansi, V. Marano, and G. Rizzoni, “A highly resolved modeling technique to simulate residential power demand,” *Appl. Energy*, vol. 107, pp. 465–473, Jul. 2013, doi: 10.1016/j.apenergy.2013.02.057.

[105] G. Wei, “Approaches to Interval Intuitionistic Trapezoidal Fuzzy Multiple Attribute Decision Making with Incomplete Weight Information,” *Int J Fuzzy Syst*, vol. 17, no. 3, pp. 484-489, Sep. 2015.

[106] Y. Wang, “A Survey and Formal Analyses on Sequence Learning Methodologies and Deep Neural Networks,” in *IEEE 17th International Conference on Cognitive Informatics Cognitive Computing (ICCI*CC*, 2018, pp. 6–15.

[107] N. Buduma and N. Locascio, *Fundamentals of Deep Learning: Designing Next-Generation Machine Intelligence Algorithms*. O’Reilly Media, Inc., 2017.

[108] Y. Lin, H. Lin, J. Wu, and K. Xu, “Learning to Rank with Cross Entropy,” in *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, New York, NY, USA, 2011, pp. 2057–2060. doi: 10.1145/2063576.2063889.

[109] S. Ruder, “An overview of gradient descent optimization algorithms,” *ArXiv160904747 Cs*, Sep. 2016, Accessed: Jan. 28, 2019. [Online]. Available: http://arxiv.org/abs/1609.04747

[110] B. Palmer *et al.*, “GridPACKTM: A framework for developing power grid simulations on high-performance computing platforms,” *Int. J. High Perform. Comput. Appl.*, vol. 30, no. 2, pp. 223–240, 2016.

[111] H. Haes Alhelou, M. E. Hamedani-Golshan, T. C. Njenda, and P. Siano, "A survey on power system blackout and cascading events: Research motivations and challenges," *Energies*, vol. 12, no. 4, p. 682, 2019.

[112] M. R. Salimian and M. R. Aghamohammadi, "A three stages decision tree-based intelligent blackout predictor for power systems using brittleness indices," *IEEE Trans. Smart Grid*, vol. 9, no. 5, pp. 5123–5131, 2017.

[113] Y. Zhang, Y. Xu, and Z. Y. Dong, "Robust ensemble data analytics for incomplete PMU measurements-based power system stability assessment," *IEEE Trans. Power Syst.*, vol. 33, no. 1, pp. 1124–1126, 2017.

[114] L. Lawton, M. Sullivan, K. Van Liere, A. Katz, and J. Eto, "A framework and review of customer outage costs: Integration and analysis of electric utility outage cost surveys," Lawrence Berkeley National Lab.(LBNL), Berkeley, CA (United States), 2003.

[115] A. Jaech, B. Zhang, M. Ostendorf, and D. S. Kirschen, "Real-time prediction of the duration of distribution system outages," *IEEE Trans. Power Syst.*, vol. 34, no. 1, pp. 773–781, 2018.

[116] I. Joyokusumo, H. Putra, and R. Fatchurrahman, "A Machine Learning-Based Strategy For Predicting The Fault Recovery Duration Class In Electric Power Transmission System," in *2020 International Conference on Technology and Policy in Energy and Electric Power (ICT-PEP)*, 2020, pp. 252–257.

[117] M.-Y. Chow, L. S. Taylor, and M.-S. Chow, "Time of outage restoration analysis in distribution systems," *IEEE Trans. Power Deliv.*, vol. 11, no. 3, pp. 1652–1658, 1996.

[118] A. Muallem, S. Shetty, J. W. Pan, J. Zhao, and B. Biswal, "Hoeffding Tree Algorithms for Anomaly Detection in Streaming Datasets: A Survey," *J. Inf. Secur.*, vol. 8, no. 4, pp. 720–726, Oct. 2017, doi: 10.4236/jis.2017.84022.

[119] B. Glocker, O. Pauly, E. Konukoglu, and A. Criminisi, "Joint classification-regression forests for spatially structured multi-object segmentation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2012, vol. 7575 LNCS, no. PART 4, pp. 870–881. doi: 10.1007/978-3-642-33765-9_62.

[120] H. Linusson, *Multi-output random forests*. University of Bor\aas/School of Business and IT, 2013.

[121] D. Paper and D. Paper, "Scikit-Learn Regression Tuning," *Hands- Scikit-Learn Mach. Learn. Appl. Data Sci. Fundam. Python*, pp. 189–213, 2020.

[122] R. Cheng, Y. Fang, and M. Renz, "Data Classification: Algorithms and Applications." Chapman & Hall CRC Data Mining and Knowledge Discovery Series, New York, USA …, 2014.

[123] U. Adhikari, T. H. Morris, and S. Pan, "Applying Hoeffding adaptive trees for real-time cyber-power event and intrusion classification," *IEEE Trans. Smart Grid*, vol. 9, no. 5, pp. 4049–4060, 2017.

[124] N. Dahal, O. Abuomar, R. King, and V. Madani, "Event stream processing for improved situational awareness in the smart grid," *Expert Syst. Appl.*, vol. 42, no. 20, pp. 6853–6863, Nov. 2015, doi: 10.1016/j.eswa.2015.05.003.

[125] J. won Lee and C. Giraud-Carrier, "Transfer learning in decision trees," in *2007 International Joint Conference on Neural Networks*, 2007, pp. 726–731.

[126] P. Domingos and G. Hulten, “Mining high-speed data streams,” in *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD ’00*, Boston, Massachusetts, United States, 2000, pp. 71–80. doi: 10.1145/347090.347107.

[127] A. Bifet, R. Gavaldà, G. Holmes, and B. Pfahringer, “Machine learning for data streams: with practical examples in MOA,” *MIT Press*, 2018.

[128] A. Bifet and R. Kirkby, “DATA STREAM MINING A Practical Approach,” 2009.

[129] H. Pishro-Nik, “Introduction to Probability, Statistics, and Random Processes”, doi: 10.25334/Q40H8J.

[130] W. Hoeffding, “Probability Inequalities for Sums of Bounded Random Variables,” *J. Am. Stat. Assoc.*, vol. 58, no. 301, pp. 13–30, Mar. 1963, doi: 10.1080/01621459.1963.10500830.

[131] A. Bifet, *Machine learning for data streams: with practical examples in MOA*. Cambridge, Massachusetts: MIT Press, 2017.

[132] A. Bifet *et al.*, “Extremely Fast Decision Tree Mining for Evolving Data Streams,” in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, 2017, pp. 1733–1742. doi: 10.1145/3097983.3098139.

[133] N. Segev, “Transfer Learning Using Decision Forests,” Institute of Technology Elul, Haifa, 2016.

[134] H. Wang and Z. Abraham, “Concept Drift Detection for Streaming Data,” *ArXiv150401044 Cs Stat*, Apr. 2015, Accessed: May 24, 2019. [Online]. Available: http://arxiv.org/abs/1504.01044

[135] M. Baena-Garcıa, R. Gavalda, and R. Morales-Bueno, “Early Drift Detection Method,” p. 10.

[136] G. Ditzler and R. Polikar, “Incremental Learning of Concept Drift from Streaming Imbalanced Data,” *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 10, pp. 2283–2301, Oct. 2013, doi: 10.1109/TKDE.2012.136.

[137] B. Mirza, Z. Lin, and N. Liu, “Ensemble of subset online sequential extreme learning machine for class imbalance and concept drift,” *Neurocomputing*, vol. 149, pp. 316–329, Feb. 2015, doi: 10.1016/j.neucom.2014.03.075.

[138] D. Jankowski, K. Jackowski, and B. Cyganek, “Learning Decision Trees from Data Streams with Concept Drift,” *Procedia Comput. Sci.*, vol. 80, pp. 1682–1691, Jan. 2016, doi: 10.1016/j.procs.2016.05.508.

[139] A. Bifet and R. Gavalda, “Learning from time-changing data with adaptive windowing,” in *Proceedings of the 2007 SIAM international conference on data mining*, 2007, pp. 443–448.

[140] “Test Cases Library.” http://web.eecs.utk.edu/~kaisun/Oscillation/actualcases.html (accessed Sep. 20, 2019).

[141] M. Donnelley, “Implementation and Operating Experience with Oscillation Detection at Bonneville Power Administration,” p. 36.

Appendices

Appendix A – Additional Resources

More resources regarding this research, including video presentations, code sources, and datasets, can be found at: https://github.com/zakaria-grid/Real-Time-ML-models-for-anomaly-detection