Ensemble Feature Learning-Based Event Classification for Cyber-Physical Security of the Smart Grid

Chengming Hu

A Thesis

in

Concordia Institute for Information Systems Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Master of Applied Science (Quality System Engineering) at

Concordia University

Montréal, Québec, Canada

September 2019

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Chengming Hu**

Entitled: **Ensemble Feature Learning-Based Event Classification for Cyber-Physical Security of the Smart Grid**

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

Master of Applied Science (Quality System Engineering)

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

Signed by the Final Examining Committee:

_____ Chair
Dr. Fereshteh Mafakheri

_____ External Examiner
Dr. Chunyan Lai

_____ Examiner
Dr. Amr Youssef

_____ Supervisor
Dr. Chun Wang

_____ Co-supervisor
Dr. Jun Yan

Approved by _____
Abdessamad Ben Hamza, Director
Department of Concordia Institute for Information Systems Engineering

August 2019

_____
Amir Asif, Dean
Gina Cody School of Engineering and Computer Science

Abstract

Ensemble Feature Learning-Based Event Classification for Cyber-Physical Security of the Smart Grid

Chengming Hu

The power grids are transforming into the cyber-physical smart grid with increasing two-way communications and abundant data flows. Despite the efficiency and reliability promised by this transformation, the growing threats and incidences of cyber attacks targeting the physical power systems have exposed severe vulnerabilities. To tackle such vulnerabilities, intrusion detection systems (IDS) are proposed to monitor threats for the cyber-physical security of electrical power and energy systems in the smart grid with increasing machine-to-machine communication. However, the multi-sourced, correlated, and often noise-contained data, which record various concurring cyber and physical events, are posing significant challenges to the accurate distinction by IDS among events of inadvertent and malignant natures. Hence, in this research, an ensemble learning-based feature learning and classification for cyber-physical smart grid are designed and implemented. The contribution of this research are *(i)* the design, implementation and evaluation of an ensemble learning-based attack classifier using extreme gradient boosting (XGBoost) to effectively detect and identify attack threats from the heterogeneous cyber-physical information in the smart grid; *(ii)* the design, implementation and evaluation of stacked denoising autoencoder (SDAE) to extract highly-representative feature space that allow reconstruction of a noise-free input from noise-corrupted perturbations; *(iii)* the design, implementation and evaluation of a novel ensemble learning-based feature extractors that combine multiple autoencoder (AE) feature extractors and random forest base classifiers, so as to enable accurate reconstruction of each feature and reliable classification against malicious events. The simulation results validate the usefulness of ensemble learning approach in detecting malicious events in the cyber-physical smart grid.

Acknowledgments

I express my gratitude to all those that have contributed towards my thesis and to those who have supported my progression towards its completion.

I am grateful to my supervisors, Dr. Jun Yan and Dr. Chun Wang from Concordia Institute for Information Systems Engineering, for the time and commitment they have invested in my research. Their vast experience and insight within the domain of my research has truly expanded my understanding and appreciation of the field, both from a theoretical and practical perspective. I also extend my gratitude to Dr. Fereshteh Mafakheri, Dr. Chunyan Lai, and Dr. Amr Youssef for being part of the examination committee for my thesis.

In addition, I would like to thank Concordia Institute for Information Systems Engineering for the wonderful hardware facilities as well as the courses that prepared me for this research. This work is also supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) under grants RGPIN-2018-06724 and RGPIN-2016-06691, and by the Fonds de Recherche du Québec - Nature et Technologies (FRQNT) under grant 2019-NC-254971.

I would like to acknowledge all my teammates: Moshfeka Rahman, Luyang Hou and Yongyuan Zhang for all the hours of fun and hard working that we have shared. Finally, I am deeply grateful to my family for their immense support and company throughout my academic studies. My parents have shown me vast amounts of encouragements as I progressed through my studies. For this I am truly thankful, and I love you all dearly.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

1.1.1 Smart Grid

Since the creation of electricity over several centuries ago, the power grid has been evolved into one of the largest networks in the human history. The power grid, a massive interconnected physical network, is the infrastructure backbone for electric power generation, transmission and distribution to the customers. As the demand and variety of consumption increases, more new technologies have been integrated into the power grid, such as renewable energy generation, the electric vehicle (EV) charging system, smart meters, etc., which all may contribute to the communication complexity in the power grid. The ever increasing reliance on electricity and request of power quality have been constantly calling for better power delivery, more flexible pricing, faster power restoration, among others.

In response to the above challenges, smart grid is a new paradigm which aims to intelligently conduct and control the devices in the system for energy supply, delivery and use [1]. The essential idea of smart grid is to integrate intelligent devices with the large-scale power network, for the purpose of enabling energy generation, transmission and distribution to be more effective, efficient, secure and economical [1]; however, the term smart grid has various definitions and meanings in

different countries. U.S. Department of Energy in 2014 [7] explained smart grid that is the transformation of the electric industry from a centralized, producer-controlled network to one that is more consumer interactive. In Europe, Bamberger *et al.* [8] described that smart grid refers to broad society participation and integration of all European countries. In China 2009, Xiao [9] proposed that smart grid is a highly coordinated intelligent power grid with extreme high voltage, so as to achieve the more reliable, responsive and economical energy supply in an environmentally-sustainable manner.

Fig. 1.1 shows the connection of various components in the large-scale smart grid [1]. The power could be generated by conventional power plants and renewable energy, such as hydro power and wind power, among others. From the generation domain, the power is transmitted along the several substations and long-distance transmission lines, and is finally distributed to the industry and resident for different applications. In order to maintain the flexibility, portability, and security of such large-scale smart grid, there is great necessary to achieve the effective cooperation and interaction between power infrastructures and intelligent devices. To meet this call, cyber-physical systems (CPS) can implement the effective integration and communication between each components in smart grid. In the following sections, we will first introduce cyber-physical systems, and then describe the application in smart grid.

1.1.2 Cyber-Physical Systems (CPS)

CPS are the integration of communication and computation (cyber domain) with physical devices and processes (physical domain) [10]. Embedded computers and systems monitor and control the physical processes, usually with feedback loops where physical processes affect computations and vice versa [10]. Similar with the various definitions of smart grid, there are also some different interpretations of CPS in different countries. The U.S. vision of CPS is more concentrated on connection between the embedded system and the physical world, while the European version highlights interaction with the cloud/cyberspace and human factors [11]. In China, CPS refer to a large-scale, embedded, hybrid complex system focusing on integration of sensing, processing, intelligence, and control as a whole [1]. As a multi-discipline research area, CPS has attracted extensive interests and efforts from different research communities. For example, system science and

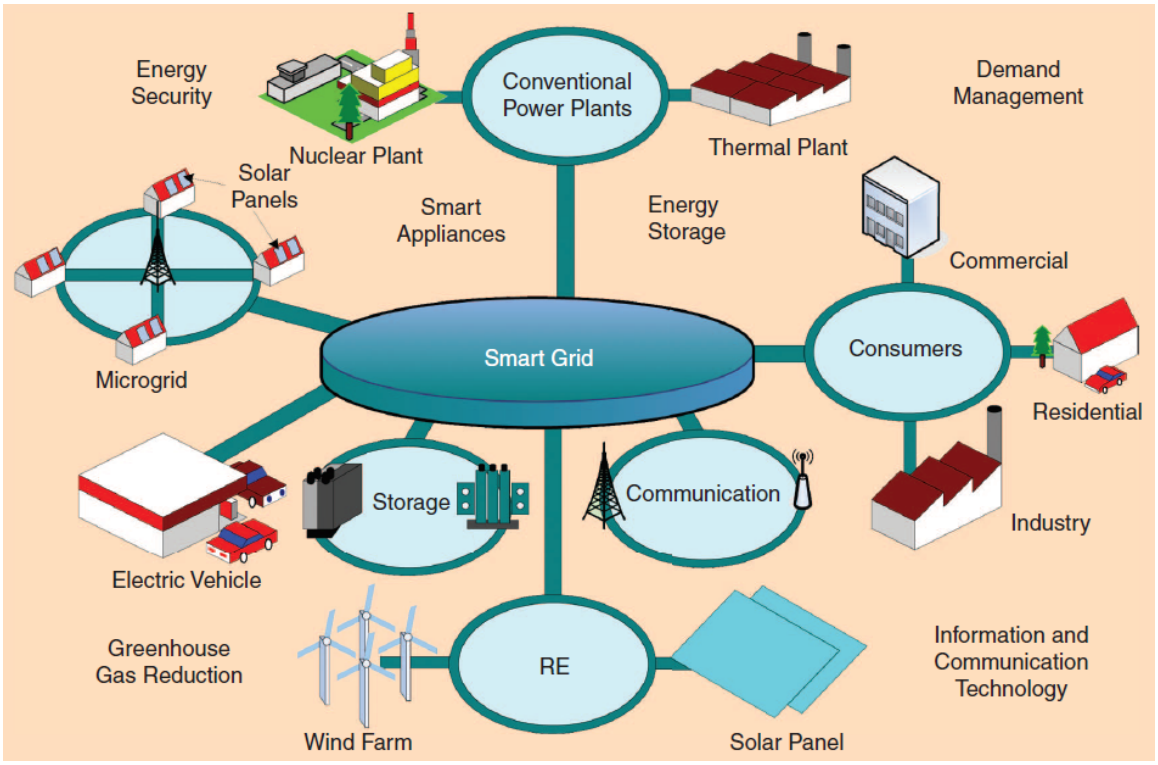


Figure 1.1: The connection network of smart grid components [1].

engineering focuses on temporal information to analysis system performance and synthesis [12], such as sensing and controlling. Computer science and engineering aims to implement real-time computation, visualization, embedded design, and model formulation by dealing with large-scale spatial information [12]. CPS bring different fields together to effectively tackle modern industrial problems which are related to high-dimensional and complex data flow and time-critical responses.

As mentioned in the last subsection, smart grid consists of many components from generation to transmission and distribution, such as power plants, smart meters and control centers, etc. The increasing communication of such components poses the requirement to design smart grid from CPS perspective. Such cyber-physical smart grid involves several devices that can interact through complex, highly interconnected physical environments. In the following, we will discuss the development of cyber-physical smart grid.

1.1.3 CPS in the Smart Grid and Security Challenges

The recent efforts in grid modernization is leading towards a cyber-physical smart grid to improve the system reliability, flexibility and efficiency. Cyber-physical smart grid regards the power network infrastructure as physical system and integrate all sensing, processing and controlling as cyber system, and some significant characteristics can be found as follows [13].

- The information from the physical system, such as voltage measurement, are dynamically changing and reported in real-time to control centers in the cyber system. The control centers can analyze such information and perform the corresponding actions to control the performance of physical system in the future times;
- The real-time interactions and computation can be achieved between components in both physical and cyber system through a two-way communication network, which helps to deliver timely decisions for smart grid operations across generation, transmission, and distribution layers through the CPS;
- Cyber-physical smart grid can respond to faults, attacks, and emergencies in the self-adaption, self-organization and self-learning ways, to maintain the system resilience, safety and great efficiency.

With the incorporation of communication and computer networks to empower the traditional power delivery, the cyber-physical smart grid is already an emerging gigantic intelligent network system in which power flows are regulated by highly automatic systems like the Supervisory Control and Data Acquisition (SCADA) system. SCADA is a control system that gather and analyze real-time data to monitor and control equipment and processes [14]. The core of SCADA system is the supervisory computers in the control centers, which are responsible for analyzing collected information from sensors, and sending control commands to the actuators. Compared with a Wide Area Monitoring, Protection and Control (WAMPC) system [15, 16], which mainly utilizes phasor data from phasor measurement units (PMU) to protect the transmission networks, SCADA system could collect data from different sources. For the transmission grid as shown in Fig. 1.2, the modernizing grid infrastructure encompasses a complex cyber-physical environment of intelligent sensors

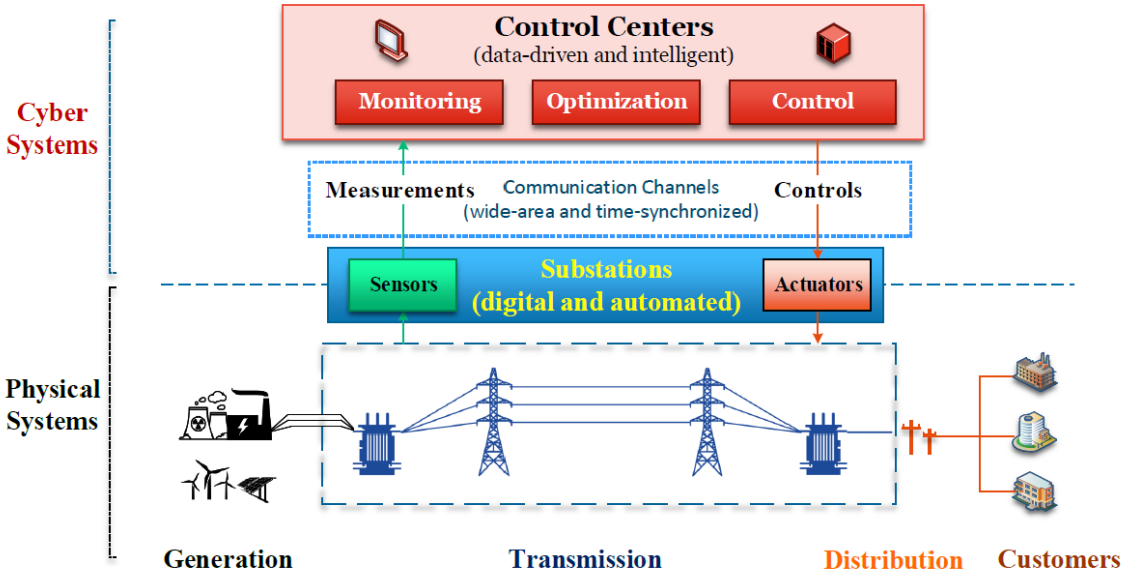


Figure 1.2: The cyber-physical architecture of a smart transmission grid [2].

and controllers, wide-area networks, secured gateways, and high-performance servers to improve the flexibility, efficiency, and reliability of power transmission [2].

However, the cyber-physical smart grid yields not only a boost of economic benefits but also a growing number of potential vulnerabilities. For example, The Aurora attack is conducted to destroy the primary controller of automatic generation control (AGC), which can desynchronise the power generator and lead to a short-term power outage even a long-term generation deficiency [17, 18]. Meanwhile, the transmission network is becoming more vulnerable to interception or unauthorized modification, which can be utilized to either cheat the meters for an unfair price or disrupt power system operations. Besides, the knowledge of the cyber-physical smart grid can be learned by the attackers, and attackers have the ability to conduct some illegal operations in the system, such as false data injection attack, etc. Hence, it becomes crucial to realize and react to the potential attacks in the cyber-physical smart grids.

In the cyber-physical smart grid, the intrusion detection systems (IDS) are main security tools to monitor and report various types of attacks. There are several basic types of detection, including the signature-based IDS, the anomaly-based IDS and the hybrid-based IDS [19], which mainly

leverage knowledge of known attacks and rule-based technique to detect any suspicious and unknown attacks. However, considering the multi-sourced, correlated and heterogeneous data in the fast-expanding, interwoven cyber-physical smart grids, such data stream are posing challenges to an effective attack distinction in the traditional IDS, since the analysis process is complex and time-consuming. Such inefficient attack detection can lead to false negatives or positives in the IDS, resulting in possibly severe impacts from undetected false commands and data, misled tripping or dispatch, equipment damage or attrition, among others. Besides, the practical system topology and behaviors can be dynamically changing in real time while the existing IDS approaches have several limitations in fully protecting the cyber-physical smart grid from increasingly sophisticated attacks like Stuxnet [20]. For instance, in 2013 Iran, Stuxnet worm impaired the nuclear-enrich capacity by compromising Siemens programmable logic controllers for nuclear centrifuges and exploiting zero-day vulnerabilities [20]. Specifically, Stuxnet could obtain control of the centrifuges and provide false feedback to the controllers, which ensured that Stuxnet continuously led the centrifuges to spin themselves to failure until the controllers observed such Stuxnet worm. Hence, the severity of such challenges is calling for machine learning algorithms to improve detection rates and efficiency in the cyber-physical smart grid. Some advanced event detectors and classifiers, such as decision tree, neural network, and ensemble learning, among others, can be introduced to better distinguish among normal operations, inadvertent faults and adversarial attacks in the complicated real-time system. Meanwhile, considering the multi-sourced nature of data that contain phasor measurement unit (PMU) measurements and system logs, some robust feature learning approaches, such as feature selection and feature extraction, are also essential to be introduced which can generate more discriminative features for malicious event classification.

1.2 Problem Statement

The objective of this research is to design, implement and evaluate the advanced machine learning for intrusion detection to identify various types of events. In the smart grids, there are often the cyber-physical, multi-sourced, correlated and noise-contained data. For example, the dataset [6] contains various concurring cyber and physical events, which could record the relay logs of the four

relays, the binary control panel logs and the digital PMU measurement data. There would be some multi-sourced features in the cyber-physical dataset [6]; for instance, the value of voltage magnitude could be over 100,000, while the maximal value of frequency is around 66 [6]. Some features could be correlated as well, such as the correlations between the current and voltage measurement data, and the correlations between the frequency and frequency delta, among others. In addition, the measurement data might contain noises that could be generated by electronic devices in the power system. Thermal noise could be generated by Brownian Motion of the electrons inside electrical devices [21], and communication delays could be also regarded as one type of noise in the power system [22]. Such cyber-physical, multi-sourced, correlated and noise-contained data could cover the useful knowledge and hidden patterns in the intrusion detection process. Hence, we propose stack denoising autoencoder (SDAE) [23] as the feature extractor to capture significant features from the dataset [6]. The unsupervised learning-based SDAE can automatically learn highly-representative feature sets by reconstruction of noise-free inputs from noise-contained data, and the algorithm will be discussed in the following Chapter 4. Such SDAE-extracted features will be adopted as the input for the event classification. Besides, to obtain the accurate intrusion detection for diverse events, the ensemble learning-based extreme gradient boosting (XGBoost) [24] that combines multiple Classification and Regression Tree (CART) [25] is determined as our event classifier. Combining the two preliminary studies, we additionally introduce a novel ensemble learning-based feature extractors, which will combine multiple autoencoder (AE) feature extractors to eliminate the potential biased results in the feature learning, so that more robust extracted features can lead to reliable intrusion detection process.

The major contributions of this research are as follows:

- Based on SDAE feature extraction that will be implemented and evaluated in Chapter 4, new relatively low-dimensional features are learnt from the heterogeneous, correlated and noise-contained data in the cyber-physical smart grids. Such low-dimensional discriminative features can represent the original information on various events and lead to effective intrusion detection process;

- The accurate and efficient event classification can be achieved by implementing the proposed

ensemble learning-based XGBoost classifiers. Compared with the other existing works that will be described in Section 3.1, XGBoost classifier has an effective improvement over the existing works achieving less than 95% [26, 27, 28, 29].

- In the novel feature ensemble learning framework that will be introduced in Chapter 5, multiple specific AE feature extractors are developed and boosted to learn discriminative features. Each feature extractor has a specific input which is determined based on feature resampling with replacement. After implementing multiple feature extractors, their learnt discriminative features will be used as the input of random forest classifiers, and can lead to an effective and reliable intrusion detection.

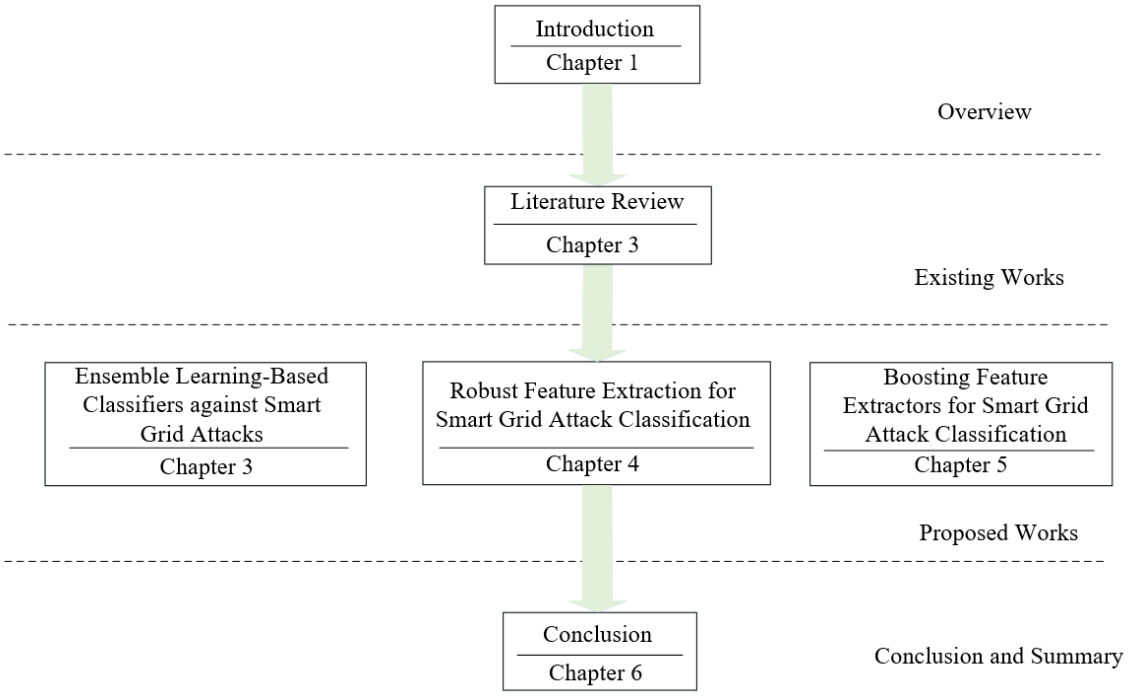


Figure 1.3: Overall structure of the thesis

1.3 Structure of the Thesis

The thesis is organized into 6 chapters with an overall structure illustrated in Figure 1.3. Following Chapter 1 of introduction, Chapter 2 will first provide the literature review of cyber-physical

security in the smart grid, from the two perspectives of physical and cyber security, respectively. IDS against attacks is also discussed in Chapter 2. Chapter 3 describes an overview of ensemble learning, and implements and evaluates an ensemble-learning classifier based on the extreme gradient boosting (XGBoost) algorithm. In Chapter 4, the overview of feature learning techniques, such as feature selection and extraction, is first given, and then the description of autoencoder (AE) and stacked denoising autoencoder (SDAE) is given as well. SDAE feature extractor and XGBoost classifier will be implemented and evaluated on our benchmark dataset [6]. Chapter 5 will introduce a new ensemble learning-based feature extractor. This novel model can boost multiple AE feature extractors to obtain critical feature information, after randomly feature resampling and selection, which will be followed by multiple Random Forest classifiers. The conclusion and future work are with Chapter 6.

Chapter 2

Literature Review

2.1 Cyber-Physical Security in the Smart Grid

With the increasing communication and data streams in the smart grid, some security challenges have been risen in both physical and cyber systems [30] [31]. On one hand, the inherent physical vulnerabilities in the physical space could lead to severe blackouts [32], such as the 2003 Northeast blackout in Canada and U.S [33]. On the other hand, attackers could remotely launch intrigue, simultaneous, and coordinated attacks as tremendous threats from the cyber space [2], such as the cyber attacks on the Ukraine power grid in 2015 [34]. In this section, we will briefly discuss the physical and cyber security in the smart grid.

2.1.1 Physical Security in the Smart Grid

Due to several physical attacks, such as equipment failures, extreme weathers and human destruction, among others [35], the physical system has been exposing certain vulnerabilities in the smart grid. For instance, the 2012 transmission line overload in India lasted 15-hour blackout and affected 620 million people [36, 37, 38]. The 2005 extreme Hurricane Katrina affected over one million customers and 9,000 MW lost load in Southeastern U.S. [39]. In 2015, the transmission

lines damaged in Pakistan caused the lost load of 11,160 MW and lasted over 5 hours [40]. Besides, cyber attacks could compromise the physical security in the smart grid as well. The synchronized and coordinated cyber-attack compromised three Ukrainian regional electric power distribution companies, resulting in power outages affecting over 225,000 customers for several hours in 2015 [34, 41, 42]. The interdiction attack refers to the tripping of generations, buses, transmission lines and substations, which could cause disconnection with the smart grid [2]. The system would suffer a severe blackout when substation attackers have just launched attacks on a few transmission substations [2], since the attackers could completely obtain the control of substations after penetrating firewalls and password protections [43].

Considering the above security problems, a security defence mechanism has been established to secure the smart grid, which mainly contains three procedures: prevention, detection and migration. The prevention against attacks mainly aims to enable the secure communication, the alleviation of exposed vulnerabilities and the protection of information. Thanks to the efforts in the studies [44, 45, 46, 47], innovative systems, protocols and technologies have been developed for improving the protection of smart grid security [48], and the secure communication can meanwhile be achieved in the smart grid. Meanwhile, the prevention can be implemented to eliminate certain attacks. For the effective reduction of false data injection (FDIA), since the stealthiness is dependent on the number of measurements being compromised, the prevention could be achieved by installation of secured or encrypted devices on critical locations [2]. Hence, the critical measurement information would be protected in such way, which are immune to the potential injections. Besides, the prevention against FDIA can also be achieved through the preservation or rearrangement of crucial information in the system. Covert power network topological information can improve the security of state estimation, when a subset of the line attackers reactance has been preserved from the knowledge [49]. Reconfiguration of the topological information in the cyber space can also eliminate the risk of FDIA in large-scale distribution systems [50]. For advanced meter infrastructures, secure management and distribution mechanisms have been developed as the most effective prevention against unauthorised accesses to smart meters [51, 52, 53]. In addition, game-theoretic approach for the optimal deployment of encrypted smart meters with limited budget [54] and Markov decision process (MDP)-based preventative maintenance strategy [55] have both been

developed for the protection of smart meters.

In case of protection failures, IDSs are employed as the major defence mechanisms at the second stage. Signature and anomaly-based IDSs have been developed against known and unknown attacks, respectively, which are deployed at various layers and locations to detect the traces of imminent attacks. These early warnings allow system operators to react with proper countermeasures and/or emergency plans so the attack impacts can be minimised. A model-based IDS has been developed against the input attacks on the AGC system [56]. The IDS utilises RT load forecast to predict the ACEs over time, and their performances are compared with that of the actual ACEs obtained. With statistical and temporal characterisations of these performance, anomaly detection in the IDS is able to detect scaled and ramped inputs before they are sent into the AGC system. Interdiction in the transmission system maybe observed by effective online contingency screening [57]. The nature of these tripping can be further examined by IDSs deployed at substations. A standard-specific IDS for automated substations has been proposed in [58]. A dedicated IDS has been developed to identify temporal anomalies induced by multi-substation attacks [59]. Host-based and network-based IDSs have been integrated in an innovative strategy against simultaneous multi-substation intrusions [60]. Detection mechanisms have also been developed against both manipulation and spoofing attacks on phasor measurement unit (PMU). Innovative IDSs have been proposed against generic manipulations of PMU data based on whitelist/behaviour [61], network topology [62, 63], and data mining [64]. Detection mechanisms against the FDIA schemes have been developed along multiple directions. An integrative Kalman filter-based detector against both bad data and false data has been developed in [65]. High-performance FDIA detectors have been proposed based on adaptive cumulative sum detection [66] and quickest detection [67]. An online anomaly detection considering load forecasts, generation schedules, and synchrophasors has also been developed in [68]. Furthermore, machine learning approaches have been proposed to identify the false data based on statistic information. Both supervised distributed SVM based on alternating direction method of multipliers and semi-supervised anomaly detection based on PCA have been developed recently to classify the false data from the normal data even with incomplete measurements [69]. A variety of supervised and semi-supervised classifiers have also been evaluated in [70], which have displayed robust performances in both online and offline scenarios. Using historic data,

generalised likelihood ratio detector can provide the optimal detection against weak FDIA schemes when the attacker could not compromise the minimal required number of measurement to construct undetectable FDIA schemes [71].

Mitigation efforts are made by the system operator to minimize the potential disruptions and damages. If attacks have been cleared from the system, existing mitigation and restoration mechanisms can effectively resume the secure and reliable power system operations. However, if attacks have not been resolved, the operator needs to consider persisting malicious attempts in the system. In such interactive scenarios, mitigation strategies are commonly modelled and solved by bi-level optimisation or game-theoretic approaches. For mitigation against interdiction, an optimisation scheme has been developed, which introduces countermeasures of the defender in a third-level minimisation into the problem [72]. Alternatively, line switching has been proposed as an effective strategy to mitigate the interdiction, which is directly introduced in the lower-level of the optimisation. The solutions with the minimal cost of the operator can be obtained by genetic algorithm [73] and Benders decomposition [74]. In addition, MDP has also been integrated to model the attack-defence interaction in a substation intrusion [75]. By modelling a successful intrusion as a probabilistic event, the investigation has formulated a competition to gain access of multiple substations between attackers and operators. The optimal solution is obtained with consideration of system parameters, the attackers resources, and the operators budgets. Acoordinated mitigation framework for the mitigation against FDIA has been developed in [76]. Security metrics have been proposed within the framework to evaluate the importance of substations and measurements. Strategies in both the network and the application layers have been developed to mitigate the attacks.

2.1.2 Cyber Security in the Smart Grid

The cyber security has been identified as a major element in the smart grid security [77]. In the following, we will describe three high-level requirements in the cyber security, availability, integrity and confidentiality.

- **Availability**: Ensuring timely and reliable access to and use of information is the most critical objectives in the smart grid. This is because a loss of availability is the disruption of access

to or use of information, which may further undermine the real-time decision making, the criticality based on time latency and the normal power delivery.

- **Integrity**: Guarding against improper information modification or destruction is to ensure the information non-repudiation and authenticity. Data cannot be modified without authorization, and the timestamp, source and quality of data are known and authenticated. A loss of integrity is the unauthorized modification or destruction of information and can further induce incorrect decision regarding power management.

- **Confidentiality**: Preserving authorized restrictions on information access and disclosure is mainly to protect personal privacy and proprietary information. Such objective aims to prevent unauthorized disclosure of information that is not open to the public and individuals. Confidentiality is the least critical requirement for system security; however, it has become more important, particularly in systems involving interactions with customers, such as demand response and AMI networks.

Considering that availability is the primary security requirement of smart grid, we first discuss attacks targeting availability. Denial-of-Service (DoS) attacks attempt to delay, block or corrupt the communication in the smart grid, which can severely degrade the communication performance and further impair the operation of electronic devices. There would be catastrophic for power infrastructures, when DoS attackers intentionally delay the transmission of some time-critical messages to violate the timing requirement. For instance, an attacker can cause severe damages to power equipments if it successfully delays the transmission of a protection message in the case of trip protection in substations [78].

Attacks targeting integrity attempt to modify data without authorization, in order to corrupt critical information communication in the smart grid. The false data injection attack (FDIA) that was discovered and designed in [79], have drawn increasing attention in the research community. FDIA attackers can compromise certain sensors to inject falsified data and manipulate the measurements, such as voltage, current, and frequency, etc. The state estimation of the system could be maliciously changed, and meanwhile pass the data integrity check used in current state estimation process. For instance, FDIA has been extended to the electric market to deliberately manipulate the market price

information [80], which could result in significant financial losses to the social welfare. The load redistribution attack [81] is one type of realistic FDIA to limit the access to specific meters, in which the load bus injection measurements and line power flow measurements are vulnerable.

Compared with attackers targeting integrity, attackers targeting confidentiality have no intent to modify information transmitted over power networks. They eavesdrop on communication channels in power networks to acquire desired information, such as a customers personal information and electricity usage. From the information theory perspective, the work in [82] theoretically studied the communication capacity in a dynamic power system under an eavesdropper targeting confidentiality. The concept of competitive privacy and information-theoretic approach [83] was proposed for the intriguing privacy issues in the smart grid information and communication infrastructures.

Most components are not designed with sufficient security scheme against malignant events, particularly from the cyber system. The lack of such sufficient protection would be catastrophic, as illustrated in the cyber attacks on a Ukraine regional grid [84]. Thanks to the efforts in the cyber-physical smart grid security [2, 85, 86], intrusion detection systems (IDSs) and firewalls have been applied to prevent the smart grid from attack intrusions. Secure protocols aim to protect the two-way communications between control centers, substations, actuators and sensors, among others. Secured wired and wireless networks have also provided trustworthy communications for the emerging PMU and AMI systems [2].

It is notable that the anomaly and signature-based IDSs need to accommodate diverse patterns in the smart grid to effectively identify the malicious events. Besides, the cyber security in the smart grid also needs to consider physical properties, requirements and dependencies of the system [2]. For example, while the access could mostly be rejected after many failed log-in attempts, it is seldom allowed in the control systems. Attackers may utilise the mechanism to lock operators out of the system that will result in disastrous consequences.

2.2 Intrusion Detection against Cyber-Physical Attacks in the Smart Grid

From the perspective of cybersecurity, the smart transmission grid exposes various access points, physical vulnerabilities, and potential benefits to exploit. Based on such challenges in the smart grid, intrusion detection systems (IDS) [87] are proposed as the advanced security systems. IDS can monitor computer systems and network traffic, and analyzes that traffic for possible hostile attacks originating from outside the organization and also for system misuse or attacks originating from inside the organization [87].

2.2.1 Intrusion Detection Techniques

Thanks to the efforts in power system and cyber-physical attack studies [2, 85, 86, 88], IDS can be categorized based on the detection techniques: knowledge-based intrusion detection, anomaly-based intrusion detection and hybrid intrusion detection.

Knowledge-Based Intrusion Detection

Knowledge-based intrusion-detection techniques apply the knowledge accumulated about specific attacks and system vulnerabilities. The knowledge-based IDS contains information about these vulnerabilities, and compares the known vulnerabilities with the patterns of actions. If an action matches one of such vulnerabilities, an alarm would be raised and this action will be recognized as attack, whereas the actions is considered acceptable. In terms of techniques, knowledge-based intrusion detection approaches were first implemented using first order logic and expert systems [89]. Commercial products mostly used a signature approach [90] and other additional techniques, such as Petri nets [91] and state-transition analysis [92], have been proposed as well.

One advantage of the knowledge-based approaches is low false-alarm rates and only react to known bad actions The contextual analysis proposed by the intrusion-detection system is detailed, making it easier for the security officer using this intrusion-detection system to understand the problem and to take preventive or corrective action [19]. The major drawback is that the knowledge-based technique relies on known incidences and cannot respond well to new variants or zero-day

threats. The dictionary of IDS needs to gather the required information on the known attacks and update with new vulnerabilities and environments in real time. Besides, maintenance of the dictionary of IDS requires careful analysis of each vulnerability and is therefore a time-consuming task. Knowledge-based approaches also have to face the generalization issue. Knowledge about attacks is very focused, dependent on the operating system, version, platform, and application. The resulting intrusion-detection system is therefore closely tied to a given environment. Also, detection of insider attacks involving an abuse of privileges is deemed more difficult because no vulnerability is actually exploited by the attacker.

Anomaly-Based Intrusion Detection

Anomaly-based intrusion detection assumes that an intrusion can be detected by observing a deviation from the normal or expected behavior of the system or the users. Anomaly-based intrusion detection is based on the observations of system behaviors, and would compare current user activities with predefined behavior profiles that record normal or valid system behaviors. An alarm would be generated, when the deviation is observed. In the other words, any current activities that could not correspond to predefined behaviors, would be considered as intrusions.

One advantage of anomaly-based intrusion detection is to effectively detect against unknown attacks or zero-day attacks. Anomaly-based intrusion detection could contribute to the automatic discovery of new unknown attacks, and are also less dependent on operating-system-specific mechanisms [19]. However, anomaly-based intrusion detection might cause the drawback of the high false alarm rate [93, 94]. The behavior profiles are needed to be updated periodically, since the system behaviors are changing over time; otherwise, the unavailability of intrusion detection system and additional false alarms might be found [19].

Hybrid Intrusion Detection

In order to overcome the disadvantages and integrate the advantages of the above knowledge-based and anomaly-based intrusion detection, hybrid intrusion detection [88] has been proposed in the system. Hybrid intrusion detection could apply both the knowledge about system attacks, and the predefined behavior profiles to detect potential hostile attacks in the system.

2.2.2 Learning-Based Intrusion Detection

Learning-based intrusion detection is the other perspective of the observation of intrusion detection. Many researchers proposed machine learning algorithm for intrusion detection to reduce the false alarm rates and produce the accurate IDS. In this section, we will show some existing studies that apply machine learning algorithms for intrusion detection against attacks.

Ferhat *et al.* [95] applied cluster machine learning technique on KDD Cup 99 dataset, which used k-Means method to determine whether the network traffic is an attack or a normal one. Peng *et al.* [96] proposed a clustering method for IDS based on Mini Batch K-means combined with principal component analysis (PCA) [97], using full KDD Cup 99 dataset as the training and testing set. The pre-processing procedure is considered to figure the strings and normalization the data to ensure the data quality in [98]. After data pre-processing, the authors introduced decision tree classification for IDS and compared this method with KNN and Naïve Bayes methods [98]. The experimental results showed that the decision tree classifier is effective and precise on KDD CUP 99 dataset. Also, Manzoor and Morgan [99] proposed real-time intrusion detection system based on libSVM and C-SVM classification. The proposed approach was trained and evaluated on KDD 99 dataset. In the proposed work [100], Canonical Correlation Analysis (CCA) and Linear Discriminant Analysis (LDA) algorithms were used as the feature learning techniques, and seven classification algorithms (Naïve Bayes, REP TREE, Random Tree, Random Forest, Random Committee, Bagging and Randomizable Filtered) were trained and evaluated on the two sets of UNSW-NB15 dataset [101]. The experimental result of the proposed work found the combination of LDA and Random Tree is more accurate, whose AUROC is 99.1 and 97.4 for two datasets, respectively.

Chapter 3

Ensemble Learning-Based Classifiers against Smart Grid Attacks

3.1 Problem Statement

As mentioned in Section 2.2, the traditional knowledge-based detection can not effectively tackle the heterogeneous data in the fast-expanding cyber-physical smart grid, so that many researchers applied machine learning techniques to produce accurate intrusion detection. In recent decades, intrusion detection and other classification problems can benefit from the combination of multiple classifiers, known as ensemble classifier, which often outperforms single-classifier based approaches. The advantages are particularly evident in the field of intrusion detection, since there are various types of attacks in the system, and multiple classifiers should be considered to identify attacks [102]. Specifically, the other classifiers could adjust the misclassified results, if one classifier fails to identify an attack [103]. With the dataset [6] that will be discussed in Section 3.4, the Oak Ridge National Laboratory randomly sampled the dataset at 1% to reduce the size and conducted a comparative study among different classifiers [29], including OneR, Nearest Neighbor, Random Forest, Support Vector Machine, Naïve Bayes, JRip, Adaboost. The best accuracy was around 95% achieved by Adaboost classifier, an ensemble learning algorithm [104]. Belouch *et al.* [105] conducted four classifiers; SVM, Naïve Bayes, C4.5 Decision Tree and Random Forest on UNSW-NB15 dataset [101]. Their results showed that Random Forest classifier performed better than all

the remaining single classifiers in terms of accuracy and sensitivity, with 97.49% and 97.75%, respectively. An ensemble method Greedy-Boost was introduced in [106], and the authors performed the comparative experiment among C4.5 Decision Tree and Greedy-Boost, for the classification of the KDD99 dataset. Reported results indicated that Greedy-Boost had the better precision of the prediction of all classes, varying between 88% and 100%; however, the algorithm were not tested on the unseen (new) attacks. In another work for the classification of NSL-KDD dataset, Syarif *et al.* [107] implemented Bagging, Boosting and Stacking ensemble methods which all considered Naïve Bayes, J48 Decision Tree, JRip and IBK Nearest Neighbor as base classifiers. Compared to single classifiers, the Stacking ensemble approach could effectively improve the accuracy and reduce the false positive by 11.34% and 97.17%, respectively, at the detection of known attacks; unfortunately, both Bagging and Boosting methods were unable to significantly improve the accuracy, and three ensemble approaches could not tackle the detection of novel attacks.

In this chapter, we propose an extreme gradient boosting (XGBoost) [24] based solution that ensembles multiple Classification and Regression Tree (CART) classifiers [25] for attack classification. As an effective and efficient tree boosting algorithm, XGBoost has been adopted in problems such as load forecasting [108] and attack detection [109]. The algorithm will be detailedly explained in the later Section 3.3, and some following benefits can be achieved in our research [110]:

- The optimization is performed at building each new tree, which can allow to reduce false alarms and generate the accurate intrusion detection;
- Regularization is an essential factor to measure model complexity in the XGBoost algorithm, since it aims to avoid data overfitting problems in the system;
- The ability of parallel computation can help to effectively tackle bulk smart grid with heterogeneous and multi-sourced data stream.

3.2 Ensemble Learning

Ensemble learning is a machine learning algorithm where multiple learners are trained and combined to solve the same classification, regression, detection, or other learning tasks. Fig. 3.1

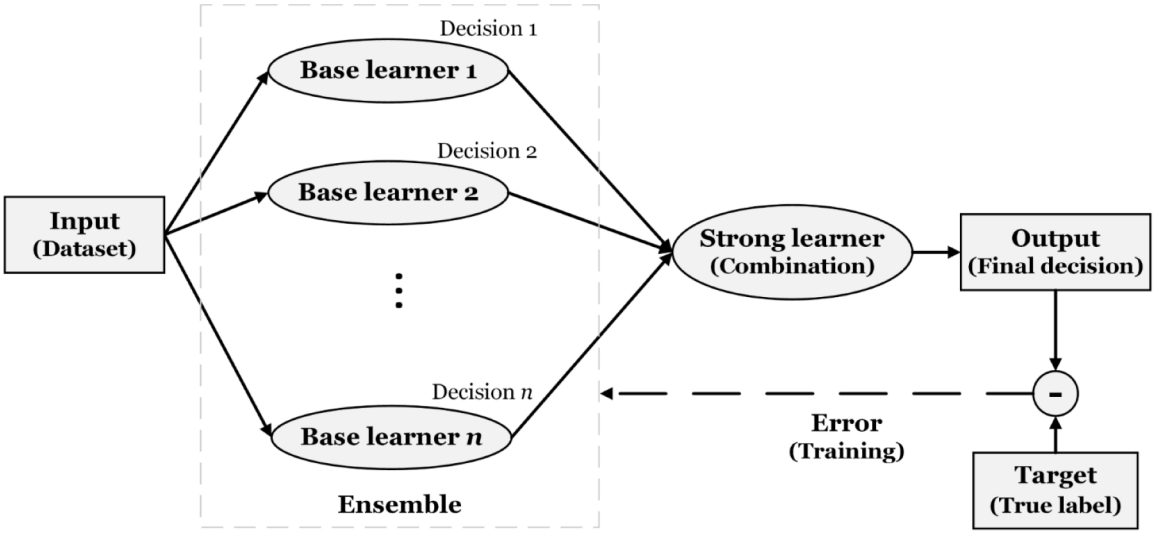


Figure 3.1: Architecture of ensemble learning algorithms.

shows the common ensemble learning architecture. Compared with traditional machine learning which aims to train one single "learner" to make the best learned decision for the task, ensemble learning methods train multiple learners and combine their decisions into a single one at the end.

An ensemble contains a number of learners called the "base learner," and the combination of base learners can form a strong learner or classifier for the ultimate task. The base learner could be implemented with only one (the homogeneous ensemble) or multiple (the heterogeneous ensemble) types of base learners. The popular choice for base learners includes Decision Trees, Neural Networks, and Support Vector Machines, among others, making ensemble learning one of the most versatile and adaptive machine learning algorithms. By combining the base learners systematically such as majority voting, an ensemble classifier can integrate the strengths and diversities among base learners while minimizing the disadvantages and misclassifications, so that the performance of individual learners can be boosted in the final decision. In the following sections, we discuss three different ensemble learning techniques, called bagging, boosting and stacking. Generally speaking, there is no one ensemble method which absolutely outperforms the other ensemble methods consistently.

3.2.1 Bagging

Bagging [3] trains several base learners based on different bootstrap samples. Fig. 3.2 shows the pseudo-code of bagging algorithm, where T is the number of learning iterations that is equal to the number base learners. At each learning iteration, the bootstrap sample is generated by sampling the original training set with replacement, where the size of such bootstrap sample is same as that of the training set. Thus, in a bootstrap sample, there might be some duplicated training samples while some may not repeatedly appear. After training all base learners from the bootstrap samples, Bagging will combine all base learners to make the final decision. Bagging can be applied with many classification algorithms such as SVM, Naïve Bayes, Decision Tree and Nearest Neighbour, among others. In the real-world applications. such as intrusion detection, Bagging could tackle the classification of the high-dimensional data, where it is hard to find a perfect single learner to work, due to the the complex and large-scale system [107].

Input:

Dataset $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)\}$;

Base learning algorithm $\mathcal{L}$;

Number of learning iterations T;

Process:

for $t = 1, ..., T$:

$\mathcal{D}_t$ =*Bootstrap*($\mathcal{D}$); % Generate a bootstrap sample from $\mathcal{D}$

h_t =$\mathcal{L}(\mathcal{D}_t)$ % Train a base learner h_t from the bootstrap sample

end.

Output:

$H(x) = argmax_y \sum_{t=1}^{T} l(y = h_t(x))$ % the value of $l(a)$ is 1 if a is $true$ and 0 otherwise

---

Figure 3.2: The bagging algorithm. [3]

3.2.2 Boosting

Boosting, which was introduced by Schapire *et al.* [111], is a common ensemble learning method for boosting the performance of several base learners into a strong leaner, typically for classification and regression tasks. Boosting is capable of sequential learning of the base learners, and the subsequent base learners are dependent on the the performance of previous learners. Fig. 3.3 shows the pseudo-code of boosting algorithm, where T is the number of learning iterations that is equal to the number base learners. The first learner is trained based on the whole dataset with original weight distribution. After obtaining the performance of each base learner, the weight distribution would be accordingly updated. Specifically, the weights of incorrectly predicted samples would be increased, which aims to focus on such incorrectly predicted samples in the subsequent learning. Hence, the subsequent base learners could be adjusted in favor of those samples incorrectly learnt by the previous learners. The strong learner would be generated through a weighted combination of these base learners. It results in different machines being specialized in predicting different areas of the dataset [112]. Among the recent efforts in the intrusion detection studies [106, 107, 113], some variants of boosting, such as Adaboost [104], Greedy-Boost [107] and XGBoost [24], have been effectively applied to classify various events in the smart grid.

In this research, we determine to design and implement the XGBoost algorithm [24], which is one of the most widely used boosting techniques for constructing a strong classifier. The impact of XGBoost algorithm has been widely recognized in a number of machine learning and data mining challenges. For example, there are around the 29 challenge wining solutions during 2015, and 17 solutions used XGBoost [24]. The success of XGBoost algorithm was also witnessed in KDDCup 2015, where XGBoost was used by every winning team in the top10. Details of the XGBoost algorithm will be discussed in Section 3.3.

3.2.3 Stacking

Stacking [5] is the other approach in the ensemble learning, which generally combines multiple types of base learners, such as Decision Tree, Neural Network and SVM, among others, to form the heterogeneous ensemble. As shown in the pseudo-code of Fig. 3.4, stacking consists of two levels

Input:

Dataset $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)\}$;

Base learning algorithm $\mathcal{L}$;

Number of learning iterations T;

Process:

$D_1 = \frac{1}{m}$; % Initialize the weight distribution of samples

for $t = 1, ..., T$:

$h_t = \mathcal{L}(\mathcal{D}, D_t)$ % Train a base learner h_t from $\mathcal{D}$ using distribution D_t

$\epsilon_t = Pr[h_t(x_i) \neq y_i]$ % Measure the error of base learner h_t

Determine the base learner weight α_t using the error ϵ_t

Construct the new weight distribution D_{t+1} using the error ϵ_t

end.

Output:

$H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t\right)$

Figure 3.3: The boosting algorithm. [4]

which are several base learners as the first-level learning algorithms and stacking model learner as the second-level learning algorithm. In the first level, all base learners are trained from the original dataset based on different machine learning algorithms. The predicted results of all first-level base learners are combined with the target value of each sample, and such combination will be used as the input of second-level learner. The output of second-level learner would be our final result [112]. For instance, we design and implement three base classifiers in the first level, Decision Tree, Naïve Bayes and Neural Network, and then the predicted labels of three first-level base classifiers will be integrated with the true labels of all samples. The second-level classifier Nearest Neighbour will be built to make the final decision based on such new combination.

Input:

Dataset $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)\}$;

First-level learning algorithms $\mathcal{L}_1, ..., \mathcal{L}_T$

Second-level learning algorithm $\mathcal{L}$

Process:

for $t = 1, ..., T$:

$h_t = \mathcal{L}_t(\mathcal{D})$ % Train a first-level individual learner h_t on the original dataset $\mathcal{D}$

end;

$\mathcal{D}' = \emptyset$; % Generate a new empty dataset

for $i = 1, ..., m$:

for $t = 1, ..., T$:

$z_{it} = h_t(x_i)$ % Use h_t to classify the training sample x_i

end;

$\mathcal{D}' = \mathcal{D}' \bigcup \{((z_{i1}, z_{i2}, ..., z_{iT}), y_i)\}$

end;

$h' = \mathcal{L}(\mathcal{D}')$ % Train the second-level learner h' on the new dataset D'

Output:

$H(x) = h'(h_1(x), ..., h_T(x))$

Figure 3.4: The stacking algorithm. [5]

3.3 Extreme Gradient Boosting (XGBoost)

XGBoost is one of the latest ensemble learning algorithms that witness an increasing adoption in real-world applications [24]. Compared to the stochastic gradient descent commonly used in machine learning, XGBoost leverages gradient boosting to find the optimal objective values towards the best decision. A brief overview of XGBoost as follows.

3.3.1 Base Learner

In this work, we choose the Classification and Regression Tree (CART) [25] as the base learner for boosting based on a few considerations: first, CART can simultaneously learn from both discrete and continuous feature values; this allows the discovery of patterns across both continuous analog measurements and discrete status signals. The decision tree-based approach also provides more interpretability and transparency of decisions made by CART, which is essential in security-related analysis. The binary tree structure of CART and its ability to handle sparse data and instance weights efficiently reduces the computational costs significantly compared with traditional decision trees and neural networks [114]. Finally, CART is also relatively friendly to parallel and distributed implementation [115, 116], which allows it to be trained faster in bulk power grids, paving the way for real-time monitoring and analysis against prominent threats.

3.3.2 Objective Function

For a dataset $D = \{(x_i, y_i)\}$ with n samples and m features, XGBoost integrates t CART base learners to obtain a strong learner that makes the ultimate classification decision. The objective function of XGBoost is:

$$obj(\theta) = \sum_{i=1}^{n} l(\hat{y}_i, y_i) + \sum_{k=1}^{t} \Omega(f_k), \quad f_k \in \mathcal{F} \tag{1}$$

where $\Omega(f_t) = \gamma \mathrm{T} + \frac{1}{2}\lambda ||w||^2$ is the complexity of each CART. T is the number of leafs in a CART so that the k-th base learner f_k represents a tree structure q with the leaf value w. γ and λ are the coefficients of the regularization terms that are determined automatically. $\mathcal{F} = \{f(x) = w_{q(x)}\}$ is called the CART space, where $w \in \mathbb{R}^T$ and $q : \mathbb{R}^m \rightarrow \mathrm{T}$ is the tree structure that maps a sample to a leaf node in that tree.

The objective function includes the loss function $l(\hat{y}_i, y_i)$, which represents the difference between the predicted label $\hat{y}_i$ and the target y_i. It is notable that the XGBoost objective function differs from traditional gradient tree boosting by the second regularization term Ω. Compared to

traditional decision trees, this term removes the need to prune a tree after it is created, which reduces model complexity to avoiding over-fitting and the computational cost in implementation.

3.3.3 Addictive Training

XGBoost model is trained in an addictive way which means that the optimization is performed at each iteration rather than after whole training. $f_t(x_i)$ means the predicted result of i-th sample at the t-th iteration and $\hat{y}_i^{(t)}$ is the combination of $f_t(x_i)$ until the t-th iteration. Let $\hat{y}_i^{(0)} = 0$, the addictive training process is to update:

$$\hat{y}_i^{(t)} = \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i) \tag{2}$$

In other words, at the t-th iteration, we need to find the optimized tree structure f_t that minimizes the objective. Here, if we choose square loss for $l(\hat{y}_i, y_i)$ and take Taylor expansion of $obj(\theta)$, then:

$$\begin{aligned} obj^{(t)} &= \sum_{i=1}^{n} l(\hat{y}_i^{(t-1)} + f_t(x_i), y_i) + \Omega(f_t) + C \\ &\simeq \sum_{i=1}^{n} [l(\hat{y}_i^{(t-1)}, y_i) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] \\ &+ \Omega(f_t) + C \end{aligned} \tag{3}$$

where $g_i = 2(\hat{y}^{(t-1)} - y_i)$, $h_i = 2$, and C is a constant that can be removed to obtain the simplified objective function at the t-th iteration below:

$$obj^{(t)} = \sum_{i=1}^{n} [g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t) \tag{4}$$

Let the sample set of leaf j be $I_j = \{i | q(x_i) = j\}$ and regroup the objective function of each leaf, the objective function can be further reduced to:

$$
\begin{aligned}
obj^{(t)} &= \sum_{i=1}^{n}[g_i w_{q(x_i)} + \frac{1}{2}h_i w^2_{q(x_i)}] + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \\
&= \sum_{j=1}^{T}[G_j w_j + \frac{1}{2}(H_j + \lambda)w_j^2] + \gamma T
\end{aligned}
\tag{5}
$$

where $G_j = \sum_{i \epsilon I_j} g_i$ and $H_j = \sum_{i \epsilon I_j} h_i$. For a fixed tree structure $q(x)$, the optimal weight in each leaf and the resulting ultimate objective value could be calculated by substituting $w_j^* = -\frac{G_j}{H_j + \lambda}$:

$$
obj = -\frac{1}{2}\sum_{j=1}^{T}\frac{G_j^2}{H_j + \lambda} + \gamma T \tag{6}
$$

3.3.4 Optimal Tree Structure Selection

A decision tree may have an infinite number of structures, so to find the best one in decision making, the scoring function below is used.

$$
Gain = \frac{1}{2}[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda}] - \gamma \tag{7}
$$

where $\frac{G_L^2}{H_L + \lambda}$ and $\frac{G_R^2}{H_R + \lambda}$ are scores of the left child and the right child after splitting a node on the tree, respectively; $\frac{(G_L + G_R)^2}{H_L + H_R + \lambda}$ is the score without splitting. The tree structure with the maximal gain is chosen to determine the optimal tree structure for the decision, i.e., if a sample belongs to the class of normal, fault, or attack events.

3.4 Cyber-Physical Attack Dataset in the Power System Benchmark

The power system benchmark and configuration are shown in Fig. 3.5, which was built by the Center for Cybersecurity Research and Engineering (CCRE) at the University of Alabama in Huntsville [6]. This is a three-bus two-machine system with four circuit breakers; each breaker is controlled by an Intelligent Electronic Device (IED) that applies a distance protection scheme to trip each breaker whenever faults are detected. Operators can also issue commands to the IEDs R1 through R4 to manually trip the breakers BR1 through BR4. These IEDs information back through a substation switch through a router back to the SCADA. The dataset used to validate this

work includes 2 non-attack contingency scenarios, 3 types of cyber-attacks, and a normal operation scenario listed below.

(1) **Data injection**: In data injection attacks [79], we suppose that malicious attackers have the full knowledge of system, and also are capable of manipulating some measurements in the system. An attacker aims to manipulate the sensor measurements to induce an arbitrary change in the estimated value of state variables without being detected by the bad measurement detection algorithm of the state estimator. Hence, this attack could blind the operator and causes a blackout by changing values to parameters such as current, voltage, and frequency, among others. In our benchmark power system, the SLG fault replay attacks, from scenario 7 to 12, attempt to emulate a valid fault by altering system measurements followed by sending an illicit trip command to relays at the certain locations of the transmission line [27]. This attack may lead to confusion and potentially cause an operator to take invalid control actions.

(2) **Remote tripping command injection**: In our benchmark power system, the attack scenarios 15 to 20 represent the remote tripping command injection against each single relay, or two relays on the same line (R1 and R2, or R3 and R4 respectively). Attackers remotely send unexpected relay trip commands to closely mimic the line maintenance scenarios. The malicious trip command originates from another node on the communications network with a spoofed legitimate IP address. Such malicious attacks would cause breakers to unpredictable open, and attackers could appear as penetrating outsides defenses.

(3) **Relay setting change**: The disabled relay attacks (including 16 scenarios) mimic the effects of insiders taking illicit control actions or malware taking control of software systems to manipulate control devices [27]. Such disabled relay attacks overlap fault and maintenance events. Relays are configured with a distance protection scheme, and attackers could conduct a script that accesses a relays internal registers via Modbus/TCP commands sent from the attackers computer; as result the relays will not trip for a valid fault or a valid command.

A dataset of 37 scenarios has been generated on this system and made available online. These data belong to three classes labeled as No Events, Natural Events, and Attack Events, as shown in Table 3.1.

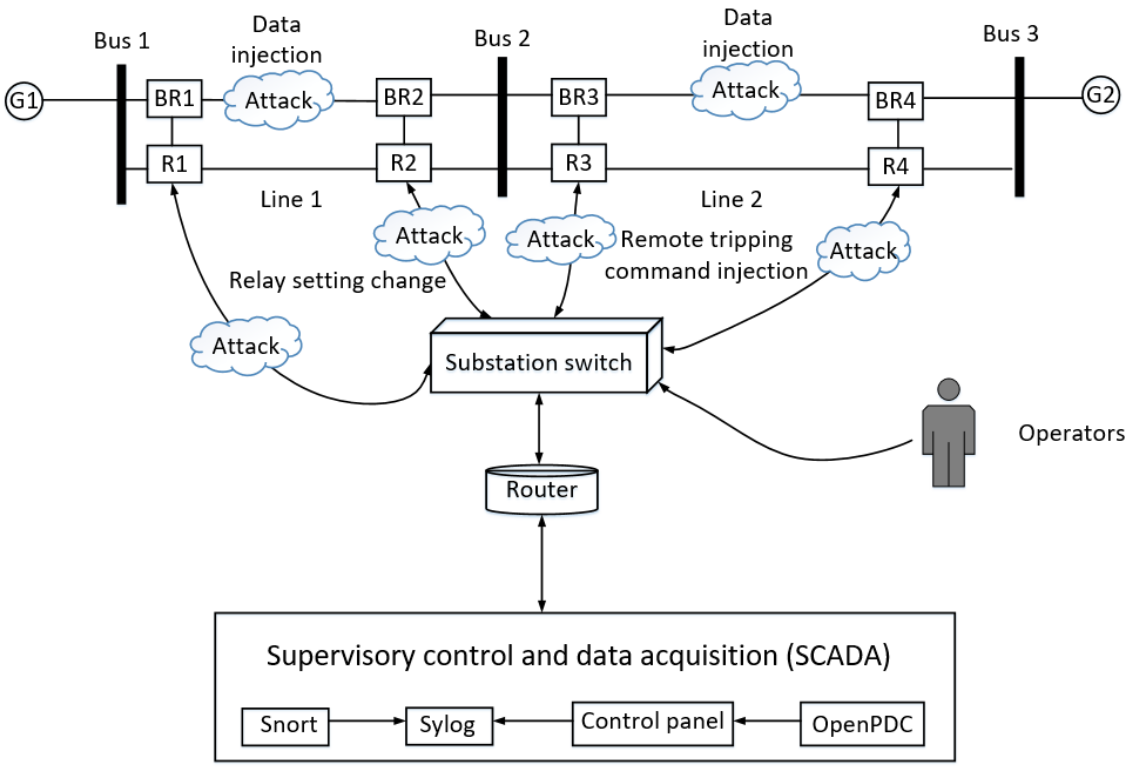


Figure 3.5: The power system benchmark [6].

Each sample in this dataset contains 128 features: 116 of them are from four phasor measurement units (PMUs); another 12 features are collected from the relay logs of the four PMUs, the control panel logs, and the Snort alert [6]. Table 3.2 gives the feature description of such cyber-physical attack dataset.

Table 3.1: The Cyber-Physical Attack Dataset

Class	Description	Scenarios	Scenario IDs
0	No Event	Normal operation with load demand variation	41
1	Natural Event	Single line-to-ground faults (SLG) and line maintenance	1–6,13,14
2	Attack Event	Data injection, remote tripping, command injection, relay settings change	7–12,15–30, 35–40

Table 3.2: Features of the Cyber-Physical Attack Dataset

Feature	Description	Range
PA1:VH - PA3:VH	Phase A - C Voltage Phase Angle	$(-180°, 180°)$
PM1:V PM3:V	Phase A - C Voltage Magnitude	[0, 152,000 V)
PA4:IH PA6:IH	Phase A - C Current Phase Angle	$(-180°, 180°)$
PM4:I PM6:I	Phase A - C Current Magnitude	[0, 1,780 A)
PA7:VH PA9:VH	Pos. Neg. Zero Voltage Phase Angle	$(-180°, 180°)$
PM7:V PM9:V	Pos. Neg. Zero Voltage Magnitude	[0, 152,000 V)
PA10:IH - PA12:IH	Pos. Neg. Zero Current Phase Angle	$(-180°, 180°)$
PM10:I - PM12:I	Pos. Neg. Zero Current Magnitude	[0, 1,265 A)
F	Frequency for relays	[0, 66 Hz]
DF	Rate of Change of Frequency (ROCOF)	[-4 dF/dt, 3.8 dF/dt]
PA:Z	Apparent Impedance	$(0.1\Omega, 1,300\Omega)$
PA:ZH	Apparent Impedance Angle	$(-3.15°, 3.15°)$
S	Status Flag for relays	[0, 272,400)
Relay_Log	Relay Log	binary values with 0 (close breakers) and 1 (open breakers)
Control_Log	Control Panel Log	binary values with 0 (illegitimate trip commands) and 1 (legitimate trip commands)
Snort_Log	Snort Alert Log	binary values with 0 (no Snort alerts) and 1 (Snort alerts)

3.5 Simulations and Results

3.5.1 Experiment Setup and Data Preprocessing

The original dataset from [6] is imbalanced, where the number of samples of in Classes 0, 1, and 2 are 174, 927, and 3,865, respectively. In this case, a classifier that flags every sample as an attack can achieve approximately 76% accuracy, which would be unacceptable. To re-balance the data and make a fair comparison with the literature, we first oversample the original dataset to re-balance the data without changing the distribution in each class. The original dataset contains 4,946 samples in total, out of which 80% of each class are chosen for training samples and the remaining 20% for testing. The 139 No Event samples and the 741 Natural Event samples in the training set are over-sampled to 3,095, i.e., the number of Attack Event samples in the training set. We refer to this

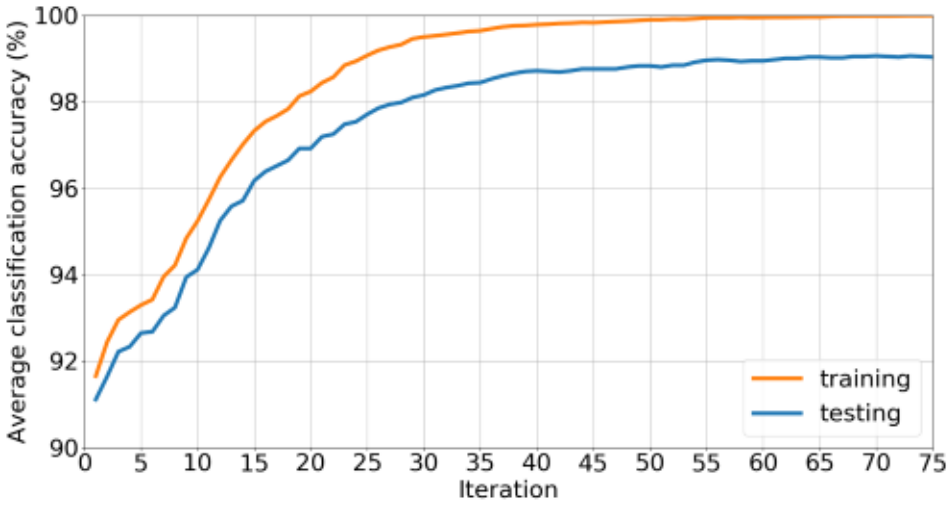
new dataset as an oversampled dataset of the original data, and it is notable that the testing data are not oversampled.

Additionally, we noted that the original dataset constitutes very limited normal operating points but extensive attack samples. Such distribution is unlikely to be observed in practice as we will observe an extensive number of scenarios in normal operations but much fewer attack samples due to the rarity of such events. Based on these considerations, we also consider a different pre-processing by introducing additional No Event and Natural Events samples from other datasets in [117]. We also select 80% for training and 20% for testing. This new dataset is referred to as the extended dataset. After data pre-processing, a five-fold cross-validation is used to select the best classifier from training and the average of 100 independent experiments will be reported as the classification performance.

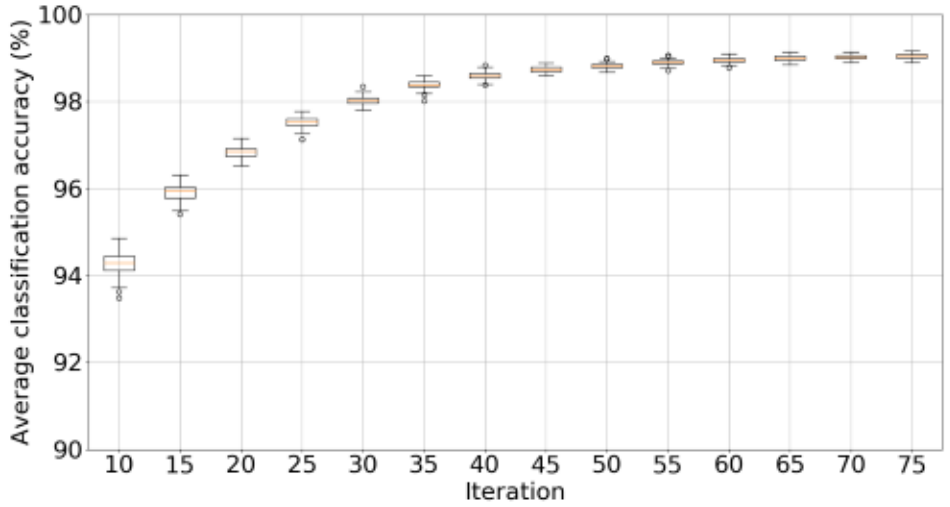
3.5.2 Simulation Results

The average classification performance of the oversampled dataset is shown in Fig. 3.6. The initial training and testing accuracy of the training and testing are both around 91% in Fig. 3.6a, and both errors are dropping with the increase of iterations. The maximal testing accuracy was 99.07% after 70 iterations, after which over-fitting was observed with the performance decreased to 99.04%. The boxplot of classification accuracy at every 5 iterations is shown in Fig. 3.6b. The boxplot reveals that as the XGBoost classifier learns from the oversampled data, not only the accuracy increases over time, but the robustness against outliers has also improved. Compare with the state-of-the-art, the proposed XGBoost classifier has reached an effective improvement over 4%.

The classification performance of the more challenging extended dataset is shown in Fig. 3.7. In Fig. 3.7a, as the training accuracy improves from 89.15% to 100% after 120 iterations, the testing accuracy increases from 84.51% to 95.18%. The robustness has also improved over time, with the variation and number of outliers decreasing as shown in Fig. 3.7b. Considering that the extended dataset adds a large variety of No Event and Natural Event samples without the counterpart of Attack Event samples, it is considerably more difficult then the ones tested with original dataset, e.g.,[117][118], but the performance of XGBoost still remains competitive to their 95% accuracy.



(a)

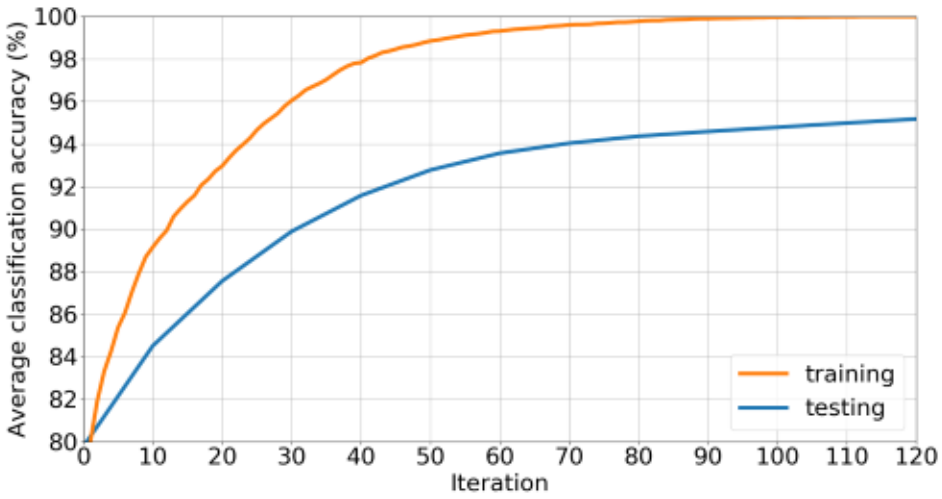


(b)

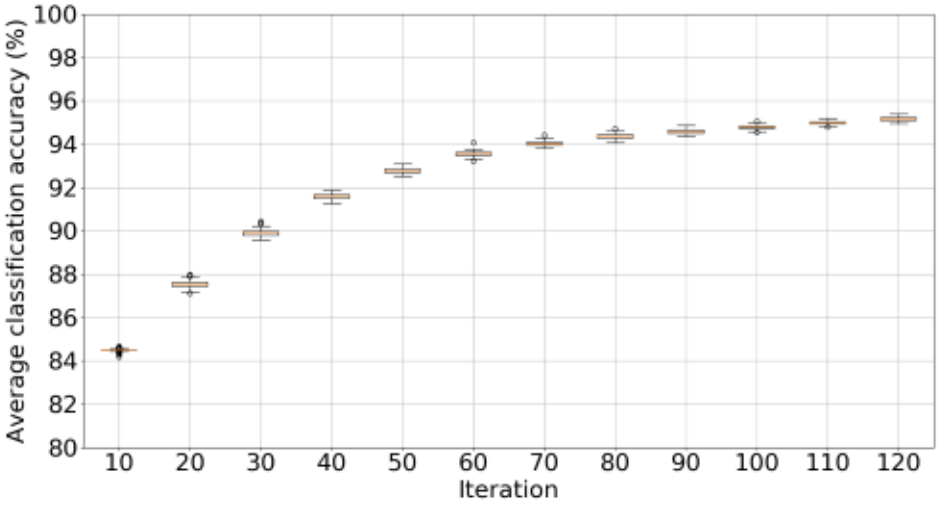
Figure 3.6: Classification performance of the oversampled dataset.

3.6 Discussions

While the testing accuracy above seem satisfactory, it is important to note that smart grid is a critical infrastructure where a single misclassification can have a potentially disastrous impact. The errors, despite being a small percentage, shall be carefully investigated and eliminated. To analyze the extend and severity of the errors in the reported performance, the confusion matrix of the three classes is shown in Table 3.3. It can be found that Class 0 (No Event) could be fully classified correctly in both the oversampled and the extended datasets. There is also no Class 1 (Natural Event) samples misclassified as Class 0 in the less-challenging oversampled dataset. The testing



(a)



(b)

Figure 3.7: Classification performance of the extended dataset.

accuracy over the two datasets is 99.89% and 93.69% for Class 1 , respectively, and 96.69% and 92.08%, for Class 2 (Attack Event), respectively.

A closer look at the misclassified samples and attack locations revealed interesting observations. Among attacks sending false remote tripping commands to the relays, the XGBoost classifier trained on oversampled data reported over 98% average testing accuracy on Relays R2 and R3; the performance dropped to 93.67% on R1 and 89.53% on R4. Similar cases are observed on the extended data, where the accuracy is 90.55%, 98.48%, 96.25%, and 91.74% for attacks on R1 to R4, respectively. By taking a second look at the benchmark topology, attacks on R2 and R3 of the load bus are both easier to classify than those on R1 and R4 protecting the generation buses. Studies on the

Table 3.3: Confusion Matrix of Testing Performances

	predict / actual	Class 0	Class 1	Class 2
Oversampled dataset	Class 0	3,500	0	0
	Class 1	0	18,581	19
	Class 2	200	2,351	74,449
Extended dataset	Class 0	3,500	0	0
	Class 1	452	17,427	721
	Class 2	1,018	5,078	70,904

relations of these performance with the grid topology and generator/load bus locations may further reveal the reasons behind so the performance can be improved toward real-world deployment.

Chapter 4

Robust Feature Extraction for Smart Grid Attack Classification

4.1 Problem Statement

In the last chapter, we introduced XGBoost ensemble classifier for effective cyber-physical attack classification. Although the accuracy seem satisfactory, it is important to note that the multi-sourced, correlated, and often noise-contained data, which record various concurring cyber and physical events, are still posing significant challenges to the accurate distinction by IDS among events of inadvertent and malignant natures. These redundant information covers the useful knowledge and hidden patterns in the distinction by IDS. To tackle such challenges, not only advanced classification model, but also robust feature learning needs to be considered, such as feature extraction. Feature extraction can learn new relatively low-dimensional feature space that represents original feature information, and such new feature set is often used for some supervised learning-based tasks, such as classification or regression. Heba *et al.* [119] implemented principal component analysis (PCA) for feature extraction approach, and then applied 23 extracted features as the input of SVM classifier for NSL-KDD dataset classification. While the accuracy was improved in some types of attacks (DoS and Probe), the overall classification accuracy decreased, compared to the classification using dataset with full 41 features. In the other combination of deep brief network (DBN) and SVM working on the NSL-KDD dataset [120], DBN was adopted to learn new features

from the original data, which was followed by one SVM classifier. Compared with the individual SVM classifier, the result indicated that such combination could improve overall accuracy and reduce the computational cost. However, they did not consider the potential imbalance of training samples in practical scenarios, which could degrade the classification performance.

In our research, we introduce a robust feature extraction approach called Stacked denoising autoencoder (SDAE) [23]. The unsupervised learning-based SDAE can automatically learn highly-representative feature sets by reconstruction of noise-free inputs from noise-contained data, and the algorithm will be discussed in the later section. We further apply XGBoost event classifier that combine multiple CART to classify samples based on the SDAE-extracted features. The objectives of applying SDAE are as follows:

- To extract the new highly-representative feature sets from the multi-sourced, correlated, and often noise-contained data, which record various concurring cyber and physical events. Such lower-dimensional features can preserve and present information on normal, fault and attack events against different synthetic but realistic noises for better classification;
- To implement a robust feature extraction approach and prevent the overfitting of the model by reconstruction of a noise-free input from noise-corrupted perturbations;

4.2 Robust Feature Learning

In the majority of classification problems, the accurate classifiers can be achieved based on the better representation of relevant features. However, in many real-world applications such as intrusion detection systems, there are many neither irrelevant nor redundant features which could cover the critical information and further pose challenges to the classification. Feature learning is an application based on a number of machine learning algorithms, which aims to capture robust features for better supervised learning tasks, such as classification and regression. In this chapter, we will discuss two common approaches: feature selection and feature extraction.

4.2.1 Feature Selection

As the raw data is large volume, it is desired to select a data subset by creating critical feature vectors. Feature selection is the process of choosing a subset of original features, and the selected feature space is optimally generated according to some evaluation criterion. The selection process can help to identify and remove some potential redundant and irrelevant information. In such way, the data dimensionality can be reduced, and meanwhile the critical feature information is captures, so that it will result in a faster and more accurate supervised learning task.

The feature selection mainly involves three approaches: filter, wrapper and embedded. Filter methods mainly use feature ranking techniques as the principle criteria for feature selection [121]. Ranking methods are chose due to their simplicity and good success is reported for practical applications. A suitable ranking criterion, such as Pearson correlation coefficient [122] and Mutual information [123], is used to score each feature and a threshold is used to remove features below the threshold. Ranking methods are applied before classification to filter out the less relevant features. The advantages of such filter feature selection is the light computation and can avoid overfitting problems [122, 124, 125]. However, a drawback of filter methods is that the selected feature set might not be optimal [121, 125]. Also, the determination of ranking threshold is mainly dependent on the subjective human expertise.

Wrapper methods consider the predictor performance as the objective function to evaluate the possible feature subsets [121]. Since the predictor performance is introduced as the evaluation criteria, wrapper methods could generally select more robust features than filter methods. However, there would be more computational cost than filter methods, since a new predictor would be built to obtain the classification accuracy for each feature subset evaluation. The overfitting problems might exist in the classification, as the classification accuracy is determined as the objective function [126].

Embedded methods want to reduce the computational cost for reclassifying different subsets which is conducted in wrapper methods [127, 128]. The main approach is to incorporate the feature selection as part of the training process of classifier. Embedded methods can reach a solution faster by avoiding retraining a predictor from scratch for every variable subset investigated [122].

4.2.2 Feature Extraction

Feature extraction is a widely-used technique for dimensionality reduction in the feature learning. Compared with the selection of original features, feature extraction projects original data, high-dimensional space to a relatively low-dimensional space and this transformation can be liner or nonlinear. The objective function of feature extraction method is to minimize the difference between the original space and the new highly-representative space, so that the useful information can be mapped into a low-dimensional feature space. Classification tasks can be effectively performed with new extracted feature space.

In the linear feature extraction, the most popular and widely-used feature extraction approach is Principle Component Analysis (PCA) that transforms original correlated features into linear uncorrelated features [97]. PCA could produce a set of new uncorrelated features according to decreasing magnitude of eigenvalues of a second order-statistics covariance matrix. Such new features are called principal components, and the last several principal components with less variability would be eliminated. In addition, many non-linear feature extraction approaches have been commonly applied in the real-world application [129, 130, 131], such as Autoencoder (deep networks) [23], Kernel PCA [132], Manifold Learning [133], among others. In the following section, we will briefly discuss the autoencoder-based techniques for feature extraction.

4.3 Deep Networks for Feature Extraction

4.3.1 Autoencoder (AE)

AE is one of the more popular unsupervised feature learning algorithm in several IDS detection studies [134, 135, 136], which applies artificial neural network to learn deep and abstract features from original inputs, typically for feature extraction. The AE architecture is shown in Fig. 4.1, where original features as input h_0 in the first layer are extracted with multiple hidden layers and eventually are reconstructed at the output layer h_{2l-2}.

The training process consists of two parts: an encoder and a decoder. The encoder compresses original features into a small representation h_{l-1} by weights $W = [w_1, w_2, \ldots, w_{l-1}]$ and bias b

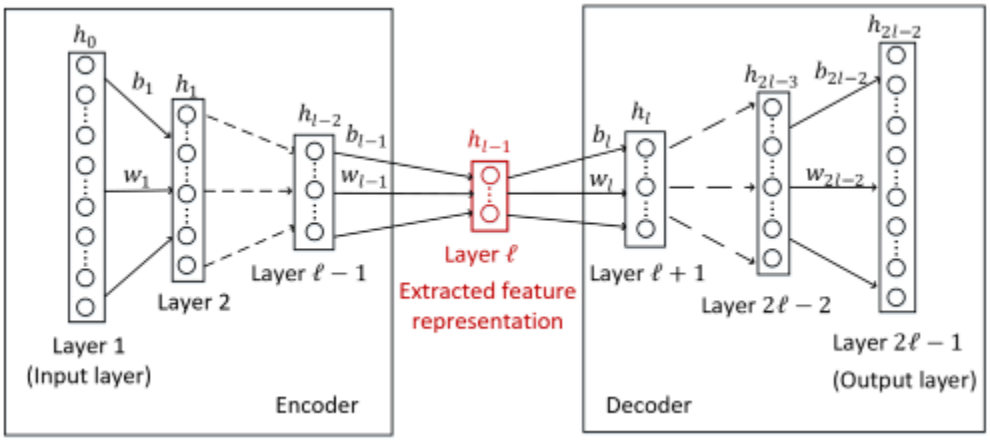


Figure 4.1: The architecture of AE for feature extraction.

$= [b_1, b_2, \ldots, b_{l-1}]$ till the middle l-th layer; the decoder reconstructs output layer from the hidden representation h_{l-1} by the other weights $W^{'} = [w_l, w_{l+1}, \ldots, w_{2l-2}]$ and bias $b^{'} = [b_l, b_{l+1}, \ldots, b_{2l-2}]$. Given a dataset x_n with n features, the feature vector h_i at the layer $i + 1$ can be calculated which is as follows:

$$h_i = f(w_i h_{i-1} + b_i), \quad i = 1, 2, ..., 2l - 2 \tag{8}$$

where $f(.)$ is the activation function of the layer h_i and the network output h_{2l-2} can also be represented as $\hat{x}_i$.

To optimize AE, the reconstruction error between the input and output should be minimized as follows:

$$\min \frac{1}{n} \sum_{i=1}^{n} l(x_i, \hat{x}_i) \tag{9}$$

where $l(x_i, \hat{x}_i) = |x_i - \hat{x}_i|^2$ is the reconstruction error of the i-th feature.

The objective function is optimized by tuning all weights and bias in the back propagation way. Once the network parameters are determined, the middle hidden layer h_{l-1} can represent the original feature information and be employed as new extracted feature sets.

4.3.2 Stacked Denoising Autoencoder (SDAE)

Considering the increasing communication and complexity in the real environment, samples are often disturbed with stochastic noises while basic AE is hard to reconstruct the noise-free inputs from noise-corrupted perturbations. Besides, AE could cause the obvious solution by identity mapping or similarly uninteresting ones that trivially maximizes mutual information as well [23]. To find out a solution, Vincent *et al.* [23] proposed denoising autoencoder (DAE) as the variant of basic AE to discover more robust features by additionally introducing the denoising criterion in the reconstruction.

Instead of noise-free inputs x_n in the basic AE, DAE disturbs the original inputs by the means of a stochastic mapping $\tilde{x}_n \sim q_D(\tilde{x}_n|x_n)$. The new extracted feature vector is determined by applying activation function to the corrupted inputs rather than the original inputs. Hence the objective function is to minimize the reconstruction error between the original noise-free inputs and the reconstruction outputs from noise-contained inputs.

As a neural network-based feature extraction method, DAE can be handily stacked to generate different levels of new feature representations of the original data, by iteratively adding new hidden layers after the previous one(s) has been trained and fixed, with the aim of discovering highly-nonlinear and complex patterns in the data [137, 138, 139]. In this paper, we introduce DAE as the basic architecture and stack multiple DAEs to form our deep model SDAE.

The overall architecture and training process of SDAE is shown in the Fig. 4.2, where several DAEs are treated as individual blocks stacked in the deep architecture. The first DAE is trained to reconstruct the raw data x_0 from the disturbed input $\tilde{x}_0$ including random Gaussian noise n_0, where $\tilde{x}_0 = x_0 + n_0, n_0 \sim N(0,1)$. We measure the difference between reconstruction result $\hat{x}_0$ and the raw data x_0 as the reconstruction error e_0 at the first training iteration, and the parameters $w_0, w_0^{'}, b_0, b_0^{'}$ are continuously tuned with the optimization of reconstruction error in the back propagation way [140]. It is notable that there are no noises in the testing process and DAE directly extracts features from the original feature x_0.

Once the first DAE is completely built, the hidden layer x_1 is its new extracted feature vector which will be then combined with random Gaussian noise n_1 for the training of the second DAE.

The overall training and testing process is similar with that of the first DAE, and eventually the extracted features x_2 can be determined with optimization of the network parameters. Repeat the above process, each successor DAE reconstructs the extracted features of predecessor DAE from its disturbed data with Gaussian noise in the training, and the low-dimensional feature space can be generated as the training and testing process is completed. After building all individual DAEs, all highly-representative hidden layers are stacked to form the SDAE model whose extracted features can be conducted in several supervised learning algorithms.

4.4 Simulation Results

4.4.1 Experiment Setup

The original data contain the raw features of different ranges, such as voltage magnitude, current angle and frequency, among others, so data normalization is essential to be first performed to improve the uniformity of the data by adjusting all raw data values into [0, 1]. Considering that the original dataset is imbalanced, where the number of samples in Class 0, 1, and 2 are 4,405, 18,309, and 55,663, respectively. In this case, a classifier that labels a sample as Class 2 (Attack Event) can achieve around 71% accuracy, which is unacceptable. To address this imbalanced problem, after determining 80% of each class for the training set and the remaining 20% for testing, we oversample the training samples as the second procedure of data pre-processing. The 3,524 No Event samples and the 14,647 Natural Event samples are oversampled to the same number (44,530) of the Attack Event samples in the training set, while the testing set is not oversampled. After data pre-processing, SDAE feature extractor is trained with the corrupted inputs that are generated by adding Gaussian noise following $N(0, 0.01)$ into the oversampled training samples, while the SDAE feature extractor is tested without Gaussian noise and obtains the extracted features from original testing samples. XGBoost ensemble learning-based classifier is performed based on the SDAE extracted features, which can distinguish the normal operation, fault and attack events.

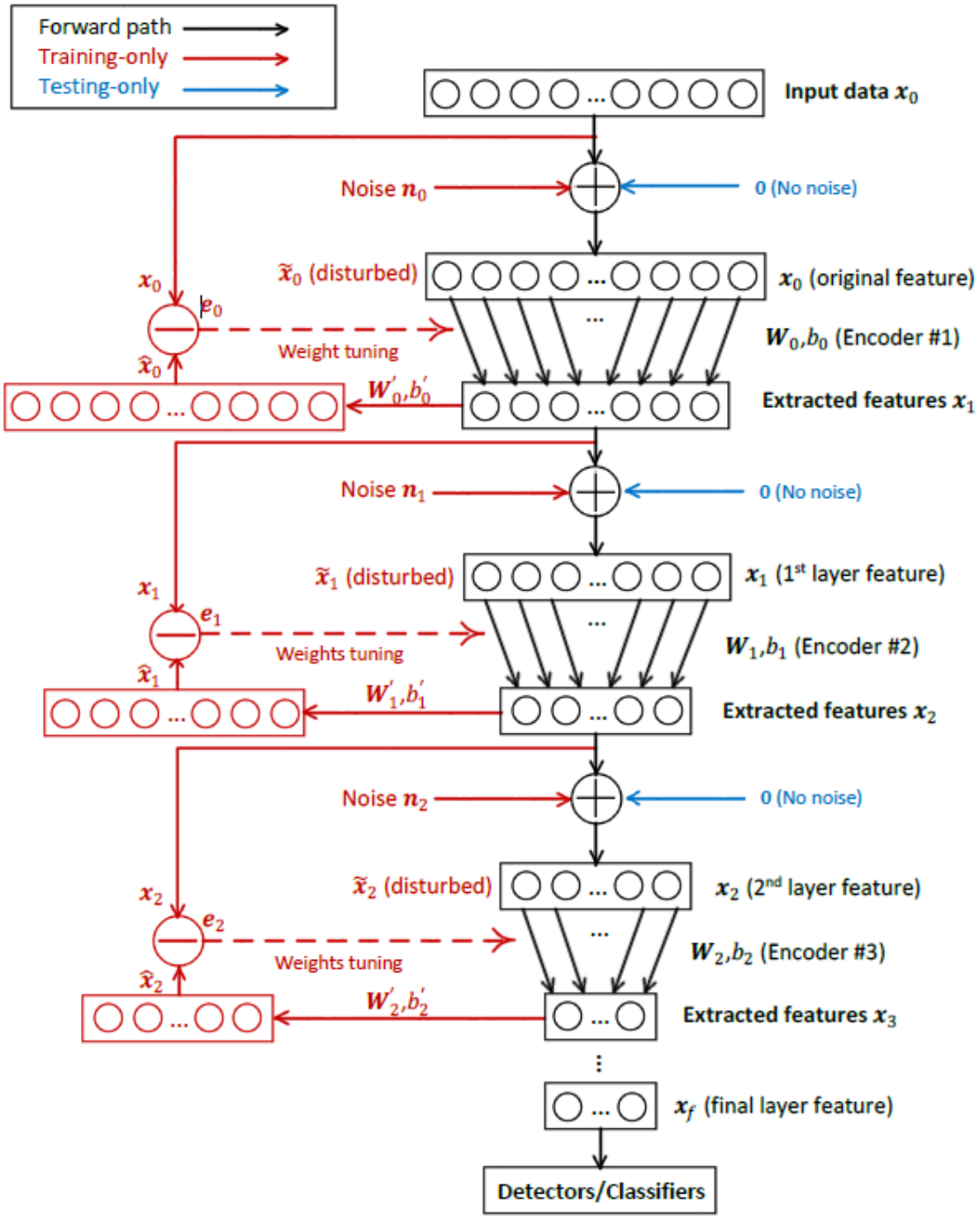


Figure 4.2: The overall architecture and training process of SDAE.

4.4.2 Simulation Results

Table 4.1 shows the classification performance comparison (per class and overall) between individual XGBoost, SAE with XGBoost, SDAE with XGBoost, individual Random Forest, SAE with Random Forest and SDAE with Random Forest models, respectively. As a balanced dataset, we use both per-class and overall accuracies to evaluate the performance of each classifier. Also, we decompose the performance into two sets of false positive rate (FPR) and false negative rate (FNR) between events vs. non-events and faults vs. attacks as follows:

- FPR_1: the fraction of normal samples (non-events) misclassified as non-normal (fault or attack);
- FNR_1: the fraction of non-normal samples (fault or attack) misclassified as normal (non-events);
- FPR_2: the fraction of fault samples misclassified as attack;
- FNR_2: the fraction of attack samples misclassified as fault;

All models based feature extraction can outperform individual XGBoost and Random Forest classifiers, whose overall accuracy are 82.24% and 80.03%, respectively. Specifically, SDAE with XGBoost model has the best overall accuracy with 90.48%, and meanwhile SDAE with XGBoost classifier is the best one in Class 0 (No Events) and Class 2 (Attack Events) with the accuracy of 86.95% and 95.75%, respectively. Besides, SDAE with XGBoost classifier can also outperform the other classifiers in the term of FPR_1, FNR_1 and FNR_2, whose values are 0.009, 0.031, and 0.124, respectively. In addition, it can be found that SAE with Random Forest model has the best accuracy in Class 1 (Natural Events) with 77.44% and the lowest FPR_2 with the value of 0.073. Table 4.1 shows that the misclassification mainly arises from the values of FPR_2 and FNR_2, which represents that there is more challenging in the classification between Natural Events and Attack Events.

Fig. 4.3 shows the XGBoost classification accuracy with and without SDAE feature extraction, respectively. The training accuracy of individual XGBoost classifier is improved from 73.98% to 100% after 1,200 iterations, and meanwhile its testing accuracy increases from 63.24% to 82.24%.

Table 4.1: Testing Performance Comparison (per Class and Overall)

Model type	Predict / Actual	Class 0	Class 1	Class 2	Per-Class Accuracy (%)	FPR_1	FNR_1	FPR_2	FNR_2	Overall Accuracy (%)
XGBoost	Class 0	612	50	219	69.47	0.022	0.144	0.129	0.266	82.24
	Class 1	31	2,208	1,423	60.29					
	Class 2	96	966	10,071	90.46					
SAE + XGBoost	Class 0	750	25	106	85.13	0.010	0.037	0.088	0.151	89.16
	Class 1	9	2,673	980	72.99					
	Class 2	24	553	10,556	94.82					
SDAE + XGBoost	Class 0	766	21	94	**86.95**	**0.009**	**0.031**	0.081	**0.124**	**90.48**
	Class 1	7	2,756	899	75.26					
	Class 2	20	453	10,660	**95.75**					
Random Forest	Class 0	556	55	270	63.11	0.027	0.226	0.141	0.291	80.03
	Class 1	43	2,066	1,553	56.42					
	Class 2	156	1,054	9,923	89.13					
SAE + Random Forest	Class 0	725	25	131	82.29	0.012	0.057	**0.073**	0.153	89.95
	Class 1	13	2,836	813	**77.44**					
	Class 2	37	557	10,539	94.66					
SDAE + Random Forest	Class 0	738	17	126	83.77	0.011	**0.031**	0.076	0.138	90.30
	Class 1	4	2,810	848	76.73					
	Class 2	23	503	10,607	95.28					

Compared with individual XGBoost classifier, the testing accuracy with SDAE feature extraction increases from 41.31% to 90.48%, as the training accuracy improves from 57.97% to 100% after 1,340 iterations. Besides, the above SDAE is built with two DAE layers whose reconstruction errors are shown in Fig. 4.4, respectively. The first DAE layer is built to extract 90 features from the original 128 features and its reconstruction error is reduced from 0.18806 to 0.00226 within 6,000 iterations, after the difference of reconstruction error between any two consecutive iterations is lower than 10^{-5} in 100 iterations. After that, the second DAE layer is generated to extract 60 features from the 90 features that is produced by the first DAE layer, and the reconstruction error decreases from -0.78157 to -3.00877 on a log-scale.

4.5 Discussions

Both these existing works [29, 141, 134] achieved over 95% accuracy with the partial data, which may not reflect the accurate overall performance on more comprehensive scenarios, as the accuracy of individual XGBoost classifier drops from 99.01% to 82.24% with the increase of scenario instances [142]. Table 4.2 shows the accuracy comparison of different SDAE architectures

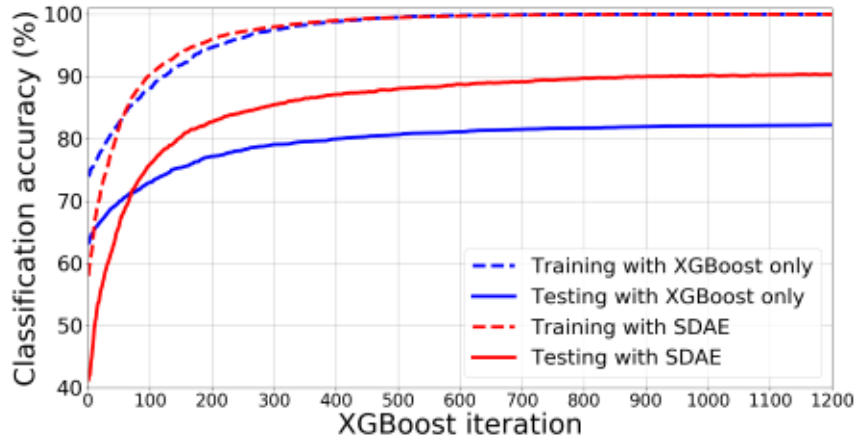


Figure 4.3: Classification accuracy without and with SDAE feature extraction.

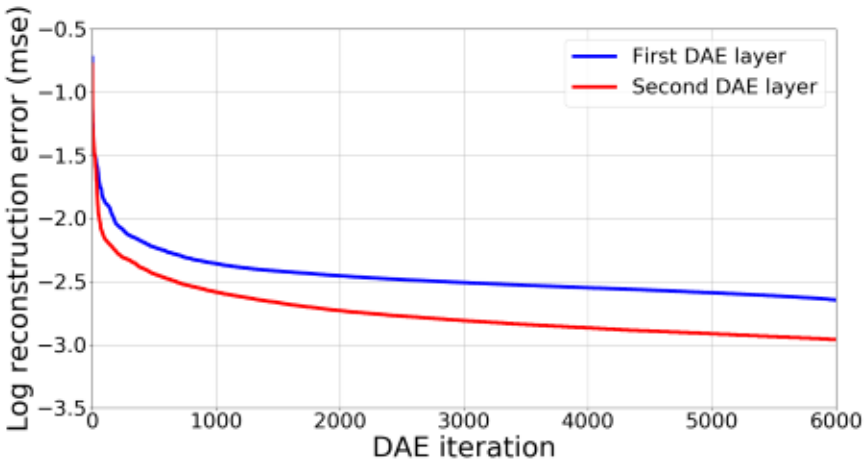


Figure 4.4: Reconstruction errors of the stacked DAE layers in SDAE.

which consider the numbers of extracted features from 60 to 30, and the numbers of DAE layers from single layer to three layers. The design of two DAE layers with 60 extracted features is recommended due to the best accuracy of 90.48%. To reduce the model complexity as well, we can adopt the simple SDAE architecture with single layer that consists of 40 or 30 neurons, due to the better accuracy of 89.58%.

A further look at the accuracy comparison on different SDAE architectures suggests that the choice and design of feature extractors may have a strong impact on the classification performance.

Table 4.2: Comparison of Accuracy for Different SDAE Architectures

Numbers of extracted features / Numbers of DAE layers	60	50	40	30
1	89.48%	89.34%	**89.58%**	**89.58%**
2	**90.48%**	87.69%	87.78%	87.44%
3	**86.21%**	86.03%	86.08%	86.09%
Best accuracy	**90.48%**	89.34%	89.58%	89.58%

Future studies can be developed on introducing more advanced feature extraction methods and finding best architectures of the proposed SDAE model, so that new noise/attack-informed features can be extracted from cyber-physical system data and the classification performance will further be improved in the real smart grid. In addition, temporal information may also be considered to improve the feature extraction and attack classification performance using recurrent neural networks, such as Long Short-Term Memory [143].

Chapter 5

Boosting Feature Extractors for Smart Grid Attack Classification

5.1 Problem Statement

Considering the the multi-sourced, correlated, and often noise-contained data in the smart grid, we introduced the framework based on SDAE and XGBoost to tackle such challenge in the last chapter; as a result, the classification accuracy can achieve over 90% with the SDAE features and XGBoost ensemble classifiers. However, there are still remaining some potential limitations in such framework. From Section 3.6, we know that the grid topology and different generator/load bus locations may impact the performance, so features of different relay locations might achieve different reconstruction performances in the feature extraction. Besides, the ranges of voltage and current features are obviously different from the ranges of frequency and status logs; therefore, the acceptance criteria of error in the voltage could not be similarly applied in the frequency and status logs. Due to the such diversity of the features, such as PMU measurements, system logs and IDS alerts, among others, the single feature extractor is hard to enable the accurate reconstruction for each feature, while it could minimize the overall difference between the original space and the new low-dimensional space. Such inaccurate extracted feature set will cause inefficient event classification in the smart grid. Therefore, according to different natures of features, there is a great necessary to design several feature extractors or selectors separately, and then combine them through

some integration techniques. In this way, the ensemble learning-based approach can eliminate the biased results from a single feature extractor or selector, and capture reliable features to lead a more accurate intrusion detection process.

Among the recent efforts, the ensemble learning-based feature learning has been applied as a pre-processing technique to capture reliable features to enable accurate classification in the smart grid. A classifier based on stacked sparse autoencoder (SSAE) and XGBoost was proposed in [113], which achieves overall 88.14% overall precision on the entire NSL-KDD dataset [144]. For each class, a specific SSAE was applied to learn latent representation of data, and then a XGBoost binary classifier was adopted to identify samples of this class. Finally, the paper considered the majority voting method to combine all SSAE-XGBoost classifiers. The authors also applied one-hot encoding for converting categorical features, and the fusion of Synthetic Minority Oversampling Technique (SMOTE) [145] and Ensemble of Undersampling (EUS) [146] to tackle potential data imbalance problem. The paper [147] designed an ensemble-based filter feature selection (EMFFS) method that combined information gain [148], gain ratio [149], chi-squared test [150] and ReliefF [151] using the majority voting scheme. 13 selected most important features of NSL-KDD dataset were used as the input to train the J48 decision tree classifier [152]. With the selected features, the J48 classifier could achieve better accuracy than the classifier with individual filter feature selection and with all features. However, the result was obtained with the partial data, which may not reflect the accurate overall performance on more comprehensive scenarios, since the accuracy of individual XGBoost classifier drops from 99.01% to 82.24% with the increase of scenario instances [142]. Wang and Gombault [153] proposed a combination of information gain [148] and chi-squared [150] to select 9 most important features from all 41 features of KDD99 dataset. Bayesian network and C4.5 classifier were used to identify DoS attack based on 9 selected features, respectively. Results obtained showed that there was limited improvement on both detection accuracy and false positive rate while the overall computational cost was significantly improved.

5.2 Boosting Feature Extractors with Ensemble Learning

Considering the limitation of above works that mainly combines multiple feature selection methods using the majority voting, we propose one novel boosting feature extractors with ensemble learning in this section. In the conventional Boosting procedure [154], the instance weights are continually updated with the training of multiple base classifiers, and each base classifier has a specific weight based on the training error. The algorithm would combine multiple base classifiers and their weights to form a strong classifier and make the final decision. In order to obtain adaptive combination of feature extractors, the proposed novel model aims to integrate multiple base classifiers whose inputs are the extracted feature sets obtained from multiple feature extractors. Besides, the input of each feature extractor is determined by the specific feature sampling and selection, so that the feature extractors could capture new specific relatively low-dimensional feature spaces.

5.2.1 Adaptive Feature Boosting

In contrast to apply a single feature extractor on all heterogeneous features, our proposed algorithm can generate different feature sets based on adaptive feature boosting. Each feature set will be applied to train one corresponding feature extractor, as shown in Algorithm. 1, where M is the number of features. Initially, the same probability $P_1(m) = \frac{1}{M}$ is given to each feature and all non-constant features should be normalized in the following way:

$$x_m^* = \frac{x_m - min(x_m)}{max(x_m) - min(x_m)} \tag{10}$$

where all normalized feature values x_m^* will be in the range [0,1].

All original features are selected to train the first feature extractor E_1, and the probabilities of all features are to update after building the first feature extractor:

$$P_2(m) = \frac{\frac{1}{M}\sqrt{\frac{e_1(m)}{1-e_1(m)}}}{Z_1} \tag{11}$$

where $e_1(m)$ is the error of each feature in the first feature extractor, and Z_1 is a normalization

factor to enable the sum of probabilities $P_2(m)$ to be 1. In the Eqn 11, the probabilities will be increased where the feature error exceeds 0.5 (MSE) and less than 1 (MSE), whereas the probabilities could be decreased.

In the adaptive feature boosting, the adaptive random feature sampling and feature selection are conducted to generate new feature set after building each feature extractor and base classifier. If the feature sampling without replacement approach is applied in our adaptive random feature sampling procedure, some features might only be once selected and trained at the first feature extractor, which could cause biased reconstruction results in the all features. Such challenge could also be faced when we determine to remove the features with better reconstruction performance in the adaptive random feature sampling. Besides, it is hard to define one threshold to measure whether the reconstruction performance of one feature can be accepted or not. Hence, in the feature sampling procedure, we determine to apply the adaptive random feature sampling with replacement based on the probability P_2. In our research, we determine that the times of feature sampling is consistent with the the number of original features. The features with large probabilities could be easily selected in this process, which represents that the successive feature extractor can be adjusted in favor of those features weakly learnt by the previous feature extractor. More reconstructable features are continuously learnt with the training of new feature extractors.

When the second feature extractor is completely built, the feature probabilities $P_3(m)$ will be calculated based on:

$$P_3(m) = \begin{cases} \frac{P_2(m)}{Z_2}\sqrt{\frac{e_2(m)}{1-e_2(m)}}, & \text{if feature is selected} \\ \frac{P_2(m)}{Z_2}, & \text{if feature is not selected} \end{cases} \tag{12}$$

where $e_2(m)$ is the error of each feature in the second feature extractor, and Z_2 is a normalization factor to enable the sum of probabilities $P_3(m)$ to be 1.

Repeating such process, each feature extractor E_t is trained on a new feature set F_t that is generated based on the adaptive feature boosting with the feature probability $P_t(m)$. The probability will be updated after building each feature extractor:

$$P_{t+1}(m) = \begin{cases} \frac{P_t(m)}{Z_t}\sqrt{\frac{e_t(m)}{1-e_t(m)}}, & \text{if feature is selected} \\ \frac{P_t(m)}{Z_t}, & \text{if feature is not selected} \end{cases} \tag{13}$$

where $e_t(m)$ is the error of each feature in the t-th feature extractor, and $P_{t+1}(m)$ is the new feature probability.

5.2.2 AE-Based Feature Extraction

AE is applied as the feature extractor in our proposed model. AE is one common unsupervised feature learning algorithm in several IDS detection studies [134, 135, 136], which applies artificial neural network to learn deep and abstract features from inputs. The training process consists of two parts: an encoder and decoder. The encoder could compress inputs into a small representation by weights and biases till the middle layer; the decoder reconstructs output layer from hidden representations by the pair of weights and biases. The reconstruction error is determined as the difference between the input and output layer. The objective function is the optimization of reconstruction error by adjusting all weights and bias in the back propagation way. Once the network parameters are determined, the hidden representation at middle layer can represent the original feature information and be employed as new extracted feature sets.

Algorithm 1 shows that all original features are selected as the inputs of the first feature extractor E_1. After training the first AE, the feature probability can be updated according to the reconstruction error of each feature. The new feature set F_2 would be generated based on the adaptive feature sampling in Section 5.2.1. Due to the feature sampling with replacement way, some features might not be selected and some features could be repeatedly selected. In the new feature set, in order to remain same feature number as the number of original features, we determine to assign the unselected features as 0 value and meanwhile retain the original feature values in the selected features.

After the adaptive feature sampling, the second AE E_2 can be trained based on the feature set F_2, whose feature number is same as the number of original features. Repeating such process, each new feature set is generated based on the feature probability $P_t(m)$ in the adaptive feature sampling procedure, and one AE feature extractor E_t would be trained using such feature set.

Algorithm 1 Boosting Feature Extractors with Ensemble Learning.

Given: $\{x_1,...,x_m,y\}$ where x_m represents the m-th feature and y is the target label.

Initialize feature probability $P_1(m) = \frac{1}{M}$ and normalize all feature values into [0, 1]

For $t = 1, ..., T$:

(1) If $t == 1$:

Generate feature set F_1 using all features.

else:

Sample features with replacement using probability $P_t(m)$;

Generate feature set F_t by retaining selected features as original values and setting unselected features as 0;

(2) Train a feature extractor on F_t as the same way in Section 4.3.1.

(3) Obtain a feature extractor E_t.

(4) Update: $P_{t+1}(m) = \begin{cases} \frac{P_t(m)}{Z_t}\sqrt{\frac{e_t(m)}{1-e_t(m)}}, & \text{if feature is selected} \\ \frac{P_t(m)}{Z_t}, & \text{if feature is not selected} \end{cases}$

where $e_t(m)$ is the error of each selected feature, and Z_t is a normalization factor.

(5) Train a base classifier on the extracted feature set from E_t.

(6) Obtain a base classifier h_t.

(7) Choose: $\alpha_t = \frac{1}{2}(\frac{1-\varepsilon_t}{\varepsilon_t})$

where ε_t is the training error of base classifier h_t

Output the final classifier: H(x) = sign($\sum_{t=1}^{T} \alpha_t h_t(x)$)

5.2.3 Ensemble Learning Classification

As shown in Fig. 5.1, the ensemble learning-based structure is considered in the classification, where each base classifier h_t follows one corresponding feature extractor E_t and n is the number of feature extractors. Each base classifier has one specific weight that is calculated based on the training error of corresponding base classifier, and we would combine all base classifiers and weights to make the final decision. The base classifier could be implemented with single classifier, such as decision trees, neural networks, and support vector machine, among others, or even ensemble classifier such as random forest. The input of base classifier is the extracted features of the corresponding feature extractor. Once the base classifier has been completed, a parameter α_t will be calculated as

follows:

$$\alpha_t = \frac{1}{2}(\frac{1-\varepsilon_t}{\varepsilon_t}) \tag{14}$$

where ε_t is the training error of base classifier h_t. α_t is used as the weight of base classifier, which aims to measure the importance of base classifier h_t. Note that the base classifier h_t with smaller training error ε_t could get the larger weight α_t, which represents that the base classifier with better accuracy will be more significantly in the final combination. As all base classifiers are completely generated, a final classifier $H(x)$ will be determined to make the final decision:

$$H(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x)) \tag{15}$$

where the final classifier is $H(x)$ a weighted majority vote of the T base classifiers.

5.3 Simulations and Results

5.3.1 Experiment Setup

With the dataset [6] that was discussed in Section 3.4, same data pre-processing techniques, including data normalization and oversampling, are performed to improve the uniformity and imbalance of data before the classification, respectively. Data normalization is first performed to adjust all raw data values into [0,1]. After determining 80% of each class for the training set and the remaining 20% for testing, we oversample the training samples as the second procedure of data pre-processing. Note that the testing set is still not oversampled. The sample number of each class in the non-oversampled and oversampled dataset are shown in Table 5.1, respectively, which are same as sample distribution of Chapter 4.

Fig. 5.1 shows the algorithm architecture, we consider three basic AEs as the feature extractors that can extract 90, 50, 30 features, respectively. The learnt features are applied as the inputs of three Random Forest (RM) base classifiers, respectively, and each RM base classifier consists of 10 CART. At the end, we combine three base classifiers and their weights to form the strong classifier to achieve an efficient classification among the normal operation, natural fault and attack events.

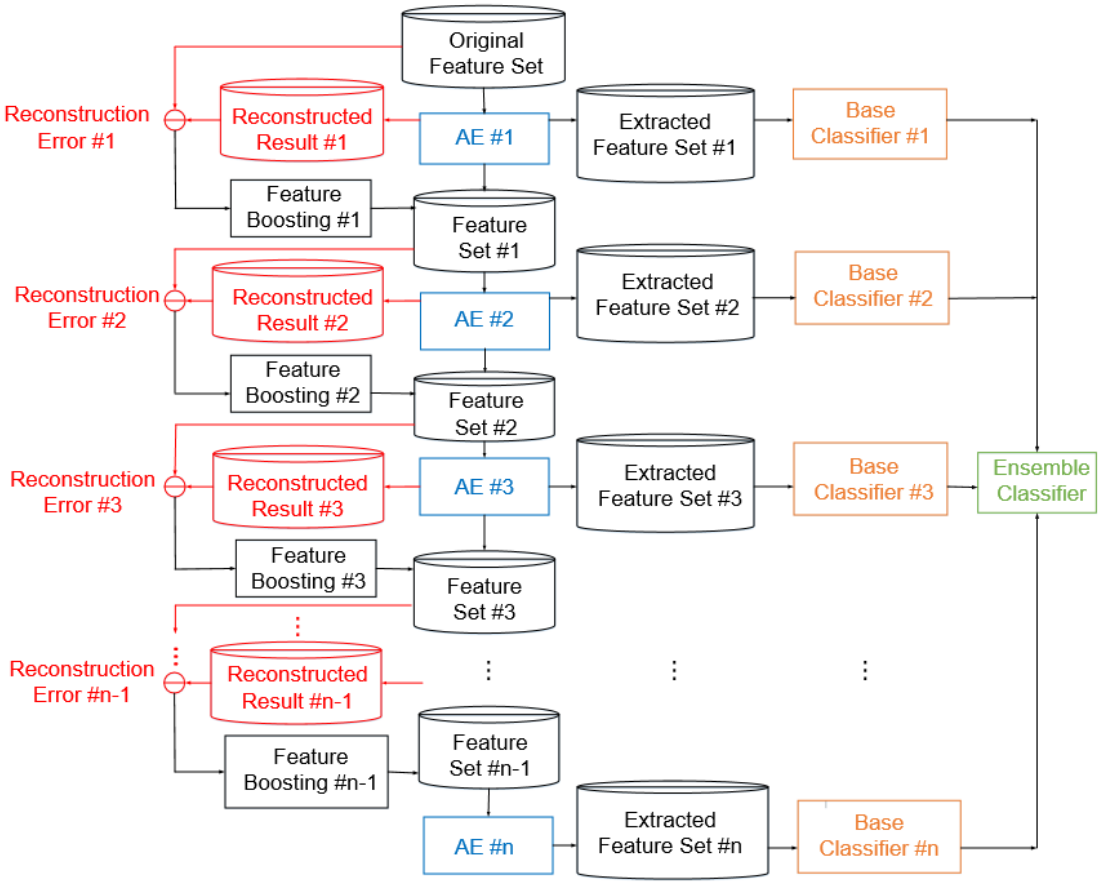


Figure 5.1: The overall boosting architecture of feature extractors and base classifiers.

5.3.2 Simulation Results

Fig. 5.3.2 shows the information of selected features at the three feature samplings. The value will be 2 when features are uniquely selected, whereas the value of repeated features are given as 1. The first AE was built with single hidden layer to extract 90 features from the whole original 128 features, and all feature probabilities were updated based on their reconstruction errors after training first AE. For generating the second feature set, we first repeatedly performed random feature sampling with replacement in 128 times, and then assigned the original feature values to the 82 unique selected features and replaced other feature values as 0. After that, the second AE was trained to extract 50 features on such feature set. The last AE aimed to learn 30 extracted features from the third feature set, consisting of the 41 original feature values and 87 zero-value features.

Besides, the above three AEs whose reconstruction errors are shown in Fig. 5.3, respectively.

Table 5.1: Sample Number of Non-oversampled and Oversampled dataset

	Class	Number of training samples	Number of testing samples
Non-Oversampled dataset	0	3,524	881
	1	14,647	3,662
	2	44,530	11,133
Oversampled dataset	0	44,530	881
	1	44,530	3,662
	2	44,530	11,133

The reconstruction error of the first AE can be reduced from 0.11605 to 0.00049 after around maximal 20,000 iterations, and the error of second AE decreased from 0.07259 to 0.00018, after the difference of reconstruction error between any two consecutive iterations is lower than 10^{-8} in 100 iterations. The last AE can achieve a minimal error of 0.000029 after around 5,900 iterations. With the extracted feature sets learnt from three AEs, all three Random Forest base classifiers were trained to classify No Events, Natural Events and Attack Events, whose testing accuracy could achieve 88.69%, 88.86% and 88.91% with around 100% training accuracy, respectively. The overall final testing accuracy could be improved to 91.78% after integrating above three base classifiers, and the confusion matrix of three classes is shown in Table 5.2. Class 2 (Attack Events) has 95.54% accuracy which can outperform over around 10% and 20% accuracy than Class 0 (No Events) and Class 1 (Natural Events), respectively.

5.4 Discussions

From Fig. 5.3, the final reconstruction error of three AEs could be minimized to -3.3098, -3.7447 and -4.5229 on a log-scale, respectively. With the continuous decrease of reconstruction error, the

Table 5.2: Confusion Matrix of Testing Performances

predict / actual	Class 0	Class 1	Class 2	Accuracy
Class 0	767	15	99	87.06%
Class 1	5	2,990	667	81.65%
Class 2	43	459	10,631	95.49%

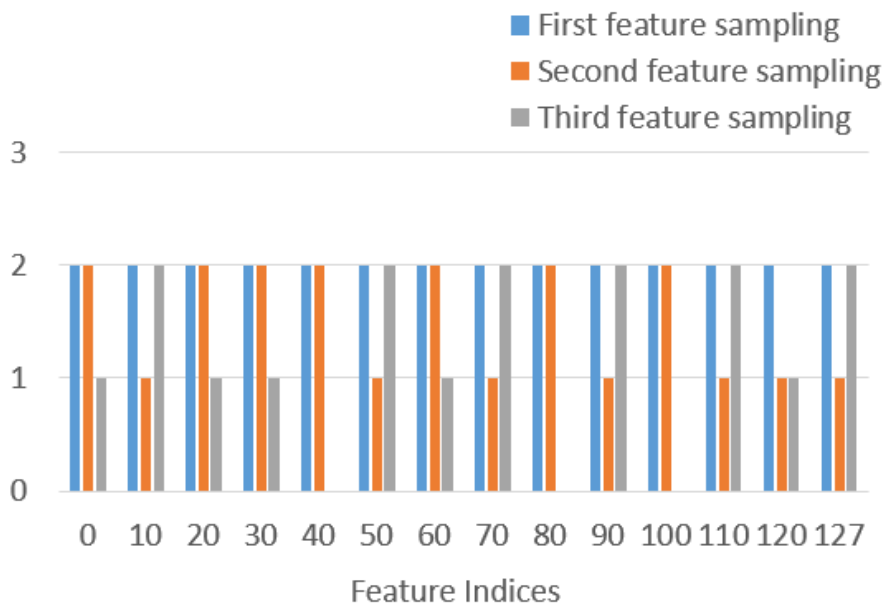


Figure 5.2: Selected feature information at each feature sampling

classification accuracy was improved from 88.69% to 88.91% in the three RM base classifiers, which represents that the reduction of reconstruction error can lead to an effective improvement on the cyber-physical event classification.

Compared to the current works [141, 134, 29, 27, 155] that achieved over 95% accuracy with partial data, our model can achieve around 100% accuracy using the same sample number of each class as these existing works. In Table 5.3, we performed t-Test based on 100 repeated experiments and 5% level of significance, and compared our results with the other two works [28, 26]. In both binary and multi-classifications of these two works, the corresponding t_0 of all classes are absolutely larger than $t_{0.05,99}$, which represents that the rejection region is satisfied and we have to reject null hypothesis H_0. Therefore, we accept the hypothesis H_1 and there is strong evidence that our classification accuracy is better than the existing results.

Considering that the diversity of base learners is an essential nature in the ensemble learning, further studies can be developed on improving diversities among the multiple feature extractors and base classifiers, such as implementing heterogeneous ensemble types or increasing the architecture complexity of base learners, etc. In addition, DAE could be developed as feature extractor in the

Table 5.3: T-Test for Benchmark Performance Comparison

Existing work	Binary classification	Five-class classification
Applying non-nested generalized exemplars classification for cyber-power event and intrusion detection [28]	$H_0 : \mu = 98$ $H_1 : \mu > 98$ $t_0 = \frac{\bar{x}-\mu_0}{S/\sqrt{n}} = 47,255$ $> t_{0.05,99} = 1.6$ **accept** H_1 **and reject** H_0	Normal operation: $t_0 = 238,557$ Line maintenance: $t_0 = 3,874$ Command injection: $t_0 = 31,424$ SLG fault replay: $t_0 = 64,624$ Relay disabled: $t_0 = 138,785$ **all classes accept** H_1 **and reject** H_0
Applying hoeffding adaptive trees for real-time cyber-power event and intrusion classification [26]	$H_0 : \mu = 98$ $H_1 : \mu > 98$ $t_0 = \frac{\bar{x}-\mu_0}{S/\sqrt{n}} = 131,250$ $> t_{0.05,99} = 1.6$ **accept** H_1 **and reject** H_0	Normal operation: $t_0 = 147,120$ Line maintenance: $t_0 = 12,575$ Command injection: $t_0 = 15,223$ SLG fault replay: $t_0 = 18,019$ Relay disabled: $t_0 = 63,888$ **all classes accept** H_1 **and reject** H_0

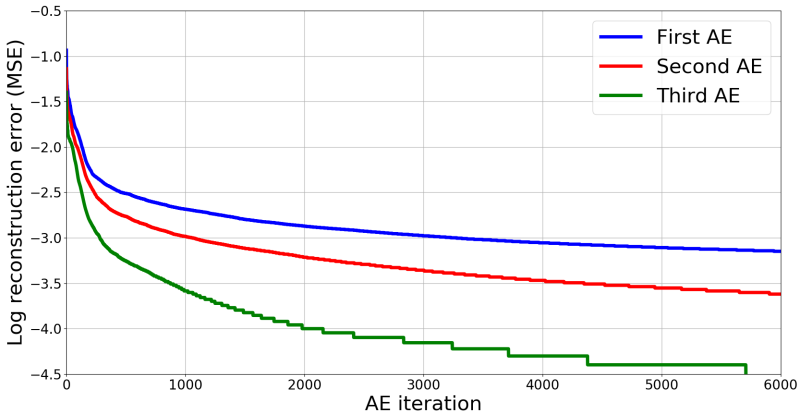


Figure 5.3: Reconstruction errors of three AEs in the feature extractor boosting

future studies, which aims to reconstruct the noise-free samples from the noise-corrupted perturbations. And our model can also be evaluated on the broader set of power system data, such as NSL-KDD dataset and data collected from IEEE 9-BUS system, among others.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

While the cyber-physical smart grid can conduct and control the devices in the system for reliable energy supply, delivery and use, is also facing some potential vulnerabilities, due to the multi-sourced, correlated, noise-contained and heterogeneous data in the fast-expanding cyber-physical smart grid. Such redundant data stream are posing challenges to smart grid cyber-physical security and an effective IDS in the system. Hence, this thesis aims to design, implement and evaluate the advanced machine learning for improving smart grid security and IDS.

The traditional intrusion detection approaches, such as knowledge-based and single-classifier techniques, can not perfectly tackle the multi-source data in the fast-expanding cyber-physical smart grid. Therefore, we determine an ensemble learning-based XGBoost classifier [24] that combines multiple CART base classifiers [25] to distinguish normal, fault and attack events. One advantage of XGBoost algorithm is that the model is trained in an addictive way, which means that the optimization is performed at each iteration rather than after whole training. Such addictive training way can allow to reduce false alarms and generate the accurate intrusion detection. Besides, XGBoost model also considers the regularization item that aims to measure the complexity of the model, so that the model could avoid overfitting problems. Considering the limitation of imbalance in the original dataset [6], we apply pre-processing to obtain an oversampled dataset and an extended dataset to evaluate the performance comprehensively. The XGBoost classier with CART as a base learner

reported 99.07% testing accuracy over the oversampled dataset, which is an effective improvement over the existing works achieving less than 95% [26, 27, 28, 29]. The proposed learning-based approach requires less expert knowledge and manual configuration, while the best performance was achieved in less than 150 iterations. Considering the relatively inexpensive computational and implementation cost of XGBoost, the proposed approach can be a promising solution to improve smart grid security against cyber-physical attacks.

Considering that data are heterogeneous, correlated and noise-contained in the cyber-physical smart grid, such redundant and irrelevant information would cover the critical knowledge and hidden patterns, and further pose challenges for intrusion detection. Hence, robust feature learning needs to be considered to capture critical feature information with low-dimensional feature space. In this thesis, we determine SDAE [23] as our feature extraction approach. SDAE is an unsupervised learning algorithm, which can automatically learn highly-representative feature sets by reconstruction of noise-free inputs from noise-contained data. The SDAE-extracted features are used to train the ensemble learning-based XGBoost classifier that combines multiple CART base classifiers. Normalization and oversampling are also applied to improve the uniformity and balance of the original data. Our proposed SDAE with XGBoost classifier reported 90.48% testing accuracy which is an effective improvement over the 82.24% accuracy achieved by individual XGBoost classifier with the original features.

Although the above proposed framework, based on SDAE feature extractor and XGBoost classifier, could achieve around 90% classification accuracy; however, there are still remaining some potential limitations in such framework. Due to the multi-sourced features in the cyber-physical smart grid, such as PMU measurements, system logs and IDS alerts, among others, the single SDAE feature extractor is hard to enable the accurate reconstruction for each feature, while could optimize the overall difference between the original feature space and the new low-dimensional space. To tackle such limitations, we propose the ensemble learning-based feature extractors that combine multiple feature extractors to learn discriminative features for the smart grid. The specific input of each feature extractor is determined based on random feature resampling with replacement. Such feature resampling approach is performed based on the feature probability that is calculated using the reconstruction error of each feature. One specific AE feature extractor will be followed by

a according Random Forest base classifier for the malicious event classification. After boosting all feature extractors and base classifiers, our proposed framework could achieve around 90.08% testing accuracy over the whole dataset, and meanwhile achieved 100% accuracy using the same sample number of each class as the existing works [141, 134, 29, 27, 155], which has a significant improvement over such works.

6.2 Future Work

Regarding the future work, on the one hand, there is room to evaluate our proposed frameworks on broader dataset which considers more comprehensive scenarios, such as datasets of IEEE 9-BUS system and NSL-KDD dataset, among others. The objective is to enhance the reliability of our proposed approaches. On the other hand, from the observation of Chapter 3, there are different performance in the several relays of the load bus. Attacks on R2 and R3 are both easier to classify that those on R1 and R4 protecting the generation buses. Hence, the further study could be investigated on the relations of the performance with the generator/load bus location. Considering that one learner may not accurately classify scenarios from all different locations, we could implement corresponding classifiers for each different bus locations. Besides, due to the heterogeneous features in the cyber-physical smart grid, such as PMU measurement, system logs and Snort alert, among others, we could also design specific feature extractor for each type of feature, so that each feature extractor may specifically learn the critical information of corresponding feature. Also, since the cyber-physical smart grid is a time-critical system, our proposed approaches will consider the malicious event classification from time series data in the future.

Bibliography

[1] Xinghuo Yu, Carlo Cecati, Tharam Dillon, and M Godoy Simoes. The new frontier of smart grids. *IEEE Industrial Electronics Magazine*, 5(3):49–63, 2011.

[2] Haibo He and Jun Yan. Cyber-physical attacks and defences in the smart grid: a survey. *IET Cyber-Physical Systems: Theory & Applications*, 1(1):13–27, 2016.

[3] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[4] Zhi-Hua Zhou. *Ensemble methods: foundations and algorithms*. CRC press, 2012.

[5] David H Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.

[6] U. Adhikari, T. Morris, and S. Pan. Wams cyber-physical test bed for power system, cyber-security study, and data mining. *IEEE Transactions on Smart Grid*, 8(6):2744–2753, Nov 2017.

[7] GridWise Alliance and US DOE. The future of the grid: Evolving to meet americas needs. *US Department of Energy*, 2014.

[8] ETP SmartGrids. Vision and strategy for europe's electricity networks of the future. *European Commission*, 2006.

[9] XIAO Shijie. Consideration of technology for constructing chinese smart grid. *Automation of Electric Power Systems*, 9(33):1–4, 2009.

[10] Edward A Lee. Cyber physical systems: Design challenges. In *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*, pages 363–369. IEEE, 2008.

[11] Volkan Gunes, Steffen Peter, Tony Givargis, and Frank Vahid. A survey on concepts, applications, and challenges in cyber-physical systems. *KSII Transactions on Internet & Information Systems*, 8(12), 2014.

[12] Radhakisan Baheti and Helen Gill. Cyber-physical systems. *The impact of control technology*, 12(1):161–166, 2011.

[13] Junhua Zhao, Fushuan Wen, Yusheng Xue, Xue Li, and Zhaoyang Dong. Cyber physical power systems: architecture, implementation techniques and challenges. *Dianli Xitong Zidonghua(Automation of Electric Power Systems)*, 34(16):1–7, 2010.

[14] Stuart A Boyer. *SCADA: supervisory control and data acquisition*. International Society of Automation, 2009.

[15] Mladen Perkov, Neven Baranović, Igor Ivanković, and Ivan Višić. Implementation strategies for migration towards smart grid. In *Powergrid Europe2010, Conference&Exhibition*, 2010.

[16] Neven Baranovic, Per Andersson, I Ivankovic, K Zubrinic-Kostovic, Domagoj Peharda, and Jan Eric Larsson. Experiences from intelligent alarm processing and decision support tools in smart grid transmission control centers. *Proceedings of the CIGRÉ Session*, 46, 2016.

[17] M Zeller. Common questions and answers addressing the aurora vulnerability. *Schweitzer Engineering Laboratories Technical Report*, pages 20150812–081908, 2011.

[18] Anurag Srivastava, Thomas Morris, Timothy Ernster, Ceeman Vellaithurai, Shengyi Pan, and Uttam Adhikari. Modeling cyber-physical vulnerability of the smart grid with incomplete information. *IEEE Transactions on Smart Grid*, 4(1):235–244, 2013.

[19] Hervé Debar, Marc Dacier, and Andreas Wespi. A revised taxonomy for intrusion-detection systems. In *Annales des télécommunications*, volume 55, pages 361–378. Springer, 2000.

[20] David Kushner. The real story of stuxnet. *ieee Spectrum*, 3(50):48–53, 2013.

[21] John Bertrand Johnson. Thermal agitation of electricity in conductors. *Physical review*, 32(1):97, 1928.

[22] Hasan Ali and Dipankar Dasgupta. Effects of time delays in the electric power grid. In *International Conference on Critical Infrastructure Protection*, pages 139–154. Springer, 2012.

[23] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(Dec):3371–3408, 2010.

[24] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 785–794. ACM, 2016.

[25] Leo Breiman. *Classification and regression trees*. Routledge, 2017.

[26] Uttam Adhikari, Thomas H Morris, and Shengyi Pan. Applying hoeffding adaptive trees for real-time cyber-power event and intrusion classification. *IEEE Transactions on Smart Grid*, 9(5):4049–4060, 2017.

[27] Shengyi Pan, Thomas Morris, and Uttam Adhikari. Developing a hybrid intrusion detection system using data mining for power systems. *IEEE Transactions on Smart Grid*, 6(6):3104–3113, 2015.

[28] Uttam Adhikari, Thomas H Morris, and Shengyi Pan. Applying non-nested generalized exemplars classification for cyber-power event and intrusion detection. *IEEE Transactions on Smart Grid*, 9(5):3928–3941, 2016.

[29] Raymond Hink, Justin Beaver, Mark Buckner, et al. Machine learning for power system disturbance and cyber-attack discrimination. In *2014 7th International symposium on resilient control systems (ISRCS)*, pages 1–8, 2014.

[30] Manimaran Govindarasu, Adam Hann, and Peter Sauer. Cyber-physical systems security for smart grid. *Future Grid Initiative White Paper, PSERC, Feb*, 2012.

[31] Yilin Mo, Tiffany Hyun-Jin Kim, Kenneth Brancik, Dona Dickinson, Heejo Lee, Adrian Perrig, and Bruno Sinopoli. Cyber–physical security of a smart grid infrastructure. *Proceedings of the IEEE*, 100(1):195–209, 2011.

[32] Kip Morison, Lei Wang, and Prabha Kundur. Power system security assessment. *IEEE power and energy magazine*, 2(5):30–39, 2004.

[33] A Muir and J Lopatto. Final report on the august 14, 2003 blackout in the united states and canada: causes and recommendations, 2004.

[34] Gaoqi Liang, Steven R Weller, Junhua Zhao, Fengji Luo, and Zhao Yang Dong. The 2015 ukraine blackout: Implications for false data injection attacks. *IEEE Transactions on Power Systems*, 32(4):3317–3318, 2016.

[35] Prabha Kundur, Neal J Balu, and Mark G Lauby. *Power system stability and control*, volume 7. McGraw-hill New York, 1994.

[36] Loi Lei Lai, Hao Tian Zhang, S Mishra, Deepak Ramasubramanian, Chun Sing Lai, and Fang Yuan Xu. Lessons learned from july 2012 indian blackout. 2012.

[37] Yong Tang, Guangquan Bu, and Jun Yi. Analysis and lessons of the blackout in indian power grid on july 30 and 31, 2012. In *Zhongguo Dianji Gongcheng Xuebao(Proceedings of the Chinese Society of Electrical Engineering)*, volume 32, pages 167–174. Chinese Society for Electrical Engineering, 2012.

[38] Joshua J Romero. Blackouts illuminate india's power problems. *IEEE spectrum*, 49(10):11–12, 2012.

[39] Paul Hines, Jay Apt, and Sarosh Talukdar. Large blackouts in north america: Historical trends and policy implications. *Energy Policy*, 37(12):5249–5259, 2009.

[40] Olga P Veloza and Francisco Santamaria. Analysis of major blackouts from 2003 to 2015: Classification of incidents and review of main causes. *The Electricity Journal*, 29(7):42–49, 2016.

[41] Dennis Palij Bråten. Taser or shock collar? energy as a weapon and interdependence in the ukraine crisis 2014-2015. Master's thesis, 2017.

[42] Hassan Haes Alhelou, Mohamad Esmail Hamedani-Golshan, Takawira Cuthbert Njenda, and Pierluigi Siano. A survey on power system blackout and cascading events: Research motivations and challenges. *Energies*, 12(4):682, 2019.

[43] Chee-Wooi Ten, Chen-Ching Liu, and Govindarasu Manimaran. Vulnerability assessment of cybersecurity for scada systems. *IEEE Transactions on Power Systems*, 23(4):1836–1846, 2008.

[44] Anthony R Metke and Randy L Ekl. Security technology for smart grid networks. *IEEE Transactions on Smart Grid*, 1(1):99–107, 2010.

[45] Ye Yan, Yi Qian, Hamid Sharif, and David Tipper. A survey on cyber security for smart grid communications. *IEEE Communications Surveys & Tutorials*, 14(4):998–1010, 2012.

[46] Ruofei Ma, Hsiao-Hwa Chen, Yu-Ren Huang, and Weixiao Meng. Smart grid communication: Its challenges and opportunities. *IEEE transactions on Smart Grid*, 4(1):36–46, 2013.

[47] Wenye Wang and Zhuo Lu. Cyber security in the smart grid: Survey and challenges. *Computer networks*, 57(5):1344–1371, 2013.

[48] Dong Wei, Yan Lu, Mohsen Jafari, Paul M Skare, and Kenneth Rohde. Protecting smart grid automation systems against cyberattacks. *IEEE Transactions on Smart Grid*, 2(4):782–795, 2011.

[49] Suzhi Bi and Ying Jun Zhang. Using covert topological information for defense against malicious attacks on dc state estimation. *IEEE Journal on Selected Areas in Communications*, 32(7):1471–1485, 2014.

[50] Morteza Talebi, Jianan Wang, and Zhihua Qu. Secure power systems against malicious cyber-physical data attacks: Protection and identification. In *International Conference on Power Systems Engineering*, pages 11–12, 2012.

[51] Nian Liu, Jinshan Chen, Lin Zhu, Jianhua Zhang, and Yanling He. A key management scheme for secure communications of advanced metering infrastructure in smart grid. *IEEE Transactions on Industrial electronics*, 60(10):4746–4756, 2012.

[52] Jinyue Xia and Yongge Wang. Secure key distribution for the smart grid. *IEEE Transactions on Smart Grid*, 3(3):1437–1443, 2012.

[53] Jia-Lun Tsai and Nai-Wei Lo. Secure anonymous key distribution scheme for smart grid. *IEEE transactions on smart grid*, 7(2):906–914, 2015.

[54] Ziad Ismail, Jean Leneutre, David Bateman, and Lin Chen. A game theoretical analysis of data confidentiality attacks on smart-grid ami. *IEEE Journal on Selected Areas in Communications*, 32(7):1486–1499, 2014.

[55] Yonghe Guo, Chee-Wooi Ten, Shiyan Hu, and Wayne W Weaver. Preventive maintenance for advanced metering infrastructure against malware propagation. *IEEE Transactions on Smart Grid*, 7(3):1314–1328, 2015.

[56] Siddharth Sridhar and Manimaran Govindarasu. Model-based attack detection and mitigation for automatic generation control. *IEEE Transactions on Smart Grid*, 5(2):580–591, 2014.

[57] Vaibhav Donde, Vanessa López, Bernard Lesieutre, Ali Pinar, Chao Yang, and Juan Meza. Severe multiple contingency screening in electric power systems. *IEEE Transactions on Power Systems*, 23(2):406–417, 2008.

[58] Upeka Kanchana Premaratne, Jagath Samarabandu, Tarlochan S Sidhu, Robert Beresh, and Jian-Cheng Tan. An intrusion detection system for iec61850 automated substations. *IEEE Transactions on Power Delivery*, 25(4):2376–2383, 2010.

[59] Chee-Wooi Ten, Junho Hong, and Chen-Ching Liu. Anomaly detection for cybersecurity of the substations. *IEEE Transactions on Smart Grid*, 2(4):865–873, 2011.

[60] Junho Hong, Chen-Ching Liu, and Manimaran Govindarasu. Integrated anomaly detection for cyber security of the substations. *IEEE Transactions on Smart Grid*, 5(4):1643–1653, 2014.

[61] Yu Yang, K McLaughlin, S Sezer, T Littler, B Pranggono, P Brogan, and HF Wang. Intrusion detection system for network security in synchrophasor systems. 2013.

[62] Seemita Pal and Biplab Sikdar. A mechanism for detecting data manipulation attacks on pmu data. In *2014 IEEE International Conference on Communication Systems*, pages 253–257. IEEE, 2014.

[63] Seemita Pal, Biplab Sikdar, and Joe H Chow. Detecting malicious manipulation of synchrophasor data. In *2015 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 145–150. IEEE, 2015.

[64] Thomas Morris, Shengyi Pan, Uttam Adhikari, Nicolas Younan, Roger King, and Vahid Madani. Cyber security testing and intrusion detection for synchrophasor systems. *International Journal of Network Science*, 1(1):28–52, 2016.

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

[66] Yi Huang, Jin Tang, Yu Cheng, Husheng Li, Kristy A Campbell, and Zhu Han. Real-time detection of false data injection in smart grid networks: An adaptive cusum method and analysis. *IEEE Systems Journal*, 10(2):532–543, 2014.

[67] Shang Li, Yasin Yılmaz, and Xiaodong Wang. Quickest detection of false data injection attack in wide-area smart grids. *IEEE Transactions on Smart Grid*, 6(6):2725–2735, 2014.

[68] Aditya Ashok, Manimaran Govindarasu, and Venkataramana Ajjarapu. Online detection of stealthy false data injection attacks in power system state estimation. *IEEE Transactions on Smart Grid*, 9(3):1636–1646, 2016.

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

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

[71] Bo Tang, Jun Yan, Steven Kay, and Haibo He. Detection of false data injection attacks in smart grid under colored gaussian noise. In *2016 IEEE Conference on Communications and Network Security (CNS)*, pages 172–179. IEEE, 2016.

[72] Yiming Yao, Thomas Edmunds, Dimitri Papageorgiou, and Rogelio Alvarez. Trilevel optimization in power network defense. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(4):712–718, 2007.

[73] Joes M Arroyo and Federico J Fernández. A genetic algorithm approach for the analysis of electric grid interdiction with line switching. In *2009 15th International Conference on Intelligent System Applications to Power Systems*, pages 1–6. IEEE, 2009.

[74] Andrés Delgadillo, José Manuel Arroyo, and Natalia Alguacil. Analysis of electric grid interdiction with line switching. *IEEE Transactions on Power Systems*, 25(2):633–641, 2009.

[75] Ying Chen, Junho Hong, and Chen-Ching Liu. Modeling of intrusion and defense for assessment of cyber security at power substations. *IEEE Transactions on Smart Grid*, 9(4):2541–2552, 2016.

[76] Ognjen Vukovic, Kin Cheong Sou, Gyorgy Dan, and Henrik Sandberg. Network-aware mitigation of data integrity attacks on power system state estimation. *IEEE Journal on Selected Areas in Communications*, 30(6):1108–1118, 2012.

[77] Victoria Y Pillitteri and Tanya L Brewer. Guidelines for smart grid cybersecurity. Technical report, 2014.

[78] Michael LeMay, Rajesh Nelli, George Gross, and Carl A Gunter. An integrated architecture for demand response communications and control. In *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008)*, pages 174–174. IEEE, 2008.

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

[80] Le Xie, Yilin Mo, and Bruno Sinopoli. False data injection attacks in electricity markets. In *2010 First IEEE International Conference on Smart Grid Communications*, pages 226–231. IEEE, 2010.

[81] Yanling Yuan, Zuyi Li, and Kui Ren. Modeling load redistribution attacks in power systems. *IEEE Transactions on Smart Grid*, 2(2):382–390, 2011.

[82] Husheng Li, Lifeng Lai, and Weiyi Zhang. Communication requirement for reliable and secure state estimation and control in smart grid. *IEEE Transactions on Smart Grid*, 2(3):476–486, 2011.

[83] Lalitha Sankar, Soummya Kar, Ravi Tandon, and H Vincent Poor. Competitive privacy in the smart grid: An information-theoretic approach. In *2011 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 220–225. IEEE, 2011.

[84] D Alert. Cyber-attack against ukrainian critical infrastructure, 2016.

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

[86] Chih-Che Sun, Adam Hahn, and Chen-Ching Liu. Cyber security of a power grid: State-of-the-art. *International Journal of Electrical Power & Energy Systems*, 99:45–56, 2018.

[87] Abhijit Sarmah. Intrusion detection systems; definition, need and challenges. *White Paper from SANS Institute*, 2001.

[88] Suad Mohammed Othman, Fadl Mutaher Ba-Alwi, Nabeel T Alsohybe, and Amal Y Al-Hashida. Intrusion detection model using machine learning algorithm on big data environment. *Journal of Big Data*, 5(1):34, 2018.

[89] Teresa F Lunt and R Jagannathan. A prototype real-time intrusion-detection expert system. In *Proceedings. 1988 IEEE Symposium on Security and Privacy*, pages 59–66. IEEE, 1988.

[90] Ming-Yang Su, Gwo-Jong Yu, and Chun-Yuen Lin. A real-time network intrusion detection system for large-scale attacks based on an incremental mining approach. *Computers & security*, 28(5):301–309, 2009.

[91] Sandeep Kumar and Eugene H Spafford. A pattern matching model for misuse intrusion detection. 1994.

[92] Phillip A Porras and Richard A Kemmerer. Penetration state transition analysis: A rule-based intrusion detection approach. In *[1992] Proceedings Eighth Annual Computer Security Application Conference*, pages 220–229. IEEE, 1992.

[93] Karen Scarfone and Peter Mell. Guide to intrusion detection and prevention systems (idps). Technical report, National Institute of Standards and Technology, 2012.

[94] Herve Debar. An introduction to intrusion-detection systems. *Proceedings of Connect*, 2000, 2000.

[95] Ferhat Karataş and Sevcan Aytaç Korkmaz. Big data: controlling fraud by using machine learning libraries on spark. *Int J Appl Math Electron Comput*, 6(1):1–5, 2018.

[96] Kai Peng, Victor CM Leung, and Qingjia Huang. Clustering approach based on mini batch kmeans for intrusion detection system over big data. *IEEE Access*, 6:11897–11906, 2018.

[97] Ian Jolliffe. *Principal component analysis*. Springer, 2011.

[98] Kai Peng, Victor Leung, Lixin Zheng, Shangguang Wang, Chao Huang, and Tao Lin. Intrusion detection system based on decision tree over big data in fog environment. *Wireless Communications and Mobile Computing*, 2018, 2018.

[99] Muhammad Asif Manzoor and Yasser Morgan. Real-time support vector machine based network intrusion detection system using apache storm. In *2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pages 1–5. IEEE, 2016.

[100] Priyanka Dahiya and Devesh Kumar Srivastava. Network intrusion detection in big dataset using spark. *Procedia computer science*, 132:253–262, 2018.

[101] Nour Moustafa and Jill Slay. Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In *2015 military communications and information systems conference (MilCIS)*, pages 1–6. IEEE, 2015.

[102] Stefan Axelsson. Intrusion detection systems: A survey and taxonomy. Technical report, Technical report, 2000.

[103] Wenke Lee, Salvatore J Stolfo, and Kui W Mok. Adaptive intrusion detection: A data mining approach. *Artificial Intelligence Review*, 14(6):533–567, 2000.

[104] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

[105] Mustapha Belouch, Salah El Hadaj, and Mohamed Idhammad. Performance evaluation of intrusion detection based on machine learning using apache spark. *Procedia Computer Science*, 127:1–6, 2018.

[106] Emna Bahri, Nouria Harbi, and Hoa Nguyen Huu. Approach based ensemble methods for better and faster intrusion detection. In *Computational Intelligence in Security for Information Systems*, pages 17–24. Springer, 2011.

[107] Iwan Syarif, Ed Zaluska, Adam Prugel-Bennett, and Gary Wills. Application of bagging, boosting and stacking to intrusion detection. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pages 593–602. Springer, 2012.

[108] Huiting Zheng, Jiabin Yuan, and Long Chen. Short-term load forecasting using emd-lstm neural networks with a xgboost algorithm for feature importance evaluation. *Energies*, 10(8):1168, 2017.

[109] A. Kuehlkamp, A. Pinto, A. Rocha, et al. Ensemble of multi-view learning classifiers for

cross-domain iris presentation attack detection. *IEEE Transactions on Information Forensics and Security*, in press.

[110] Sukhpreet Dhaliwal, Abdullah-Al Nahid, and Robert Abbas. Effective intrusion detection system using xgboost. *Information*, 9(7):149, 2018.

[111] Robert E Schapire, Yoav Freund, Peter Bartlett, Wee Sun Lee, et al. Boosting the margin: A new explanation for the effectiveness of voting methods. *The annals of statistics*, 26(5):1651–1686, 1998.

[112] Magdalena Graczyk, Tadeusz Lasota, Bogdan Trawiński, and Krzysztof Trawiński. Comparison of bagging, boosting and stacking ensembles applied to real estate appraisal. In *Asian conference on intelligent information and database systems*, pages 340–350. Springer, 2010.

[113] Baoan Zhang, Yanhua Yu, and Jie Li. Network intrusion detection based on stacked sparse autoencoder and binary tree ensemble method. In *2018 IEEE International Conference on Communications Workshops*, pages 1–6, 2018.

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

[115] Huan Zhang, Si Si, and Cho-Jui Hsieh. Gpu-acceleration for large-scale tree boosting. *arXiv preprint arXiv:1706.08359*, 2017.

[116] Guolin Ke, Qi Meng, Thomas Finley, et al. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, pages 3146–3154, 2017.

[117] Uttam Adhikari, Thomas H Morris, and Shengyi Pan. Applying hoeffding adaptive trees for real-time cyber-power event and intrusion classification. *IEEE Transactions on Smart Grid*, 9(5):4049–4060, 2018.

[118] David Wilson, Jun Yan, Lu Zhuo, and Yufei Tang. Deep learning-aided cyber-attack detection

in power transmission systems. In *2018 IEEE Power and Energy Society General Meetings*, in press.

[119] F Eid Heba, Ashraf Darwish, Aboul Ella Hassanien, and Ajith Abraham. Principle components analysis and support vector machine based intrusion detection system. In *2010 10th international conference on intelligent systems design and applications*, pages 363–367. IEEE, 2010.

[120] Mostafa A Salama, Heba F Eid, Rabie A Ramadan, Ashraf Darwish, and Aboul Ella Hassanien. Hybrid intelligent intrusion detection scheme. In *Soft computing in industrial applications*, pages 293–303. Springer, 2011.

[121] George H John, Ron Kohavi, and Karl Pfleger. Irrelevant features and the subset selection problem. In *Machine Learning Proceedings 1994*, pages 121–129. Elsevier, 1994.

[122] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

[123] Roberto Battiti. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on neural networks*, 5(4):537–550, 1994.

[124] Cosmin Lazar, Jonatan Taminau, Stijn Meganck, David Steenhoff, Alain Coletta, Colin Molter, Virginie de Schaetzen, Robin Duque, Hugues Bersini, and Ann Nowe. A survey on filter techniques for feature selection in gene expression microarray analysis. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 9(4):1106–1119, 2012.

[125] Huan Liu, Rudy Setiono, et al. A probabilistic approach to feature selection-a filter solution. In *ICML*, volume 96, pages 319–327. Citeseer, 1996.

[126] Ron Kohavi and George H John. Wrappers for feature subset selection. *Artificial intelligence*, 97(1-2):273–324, 1997.

[127] Pat Langley. Selection of relevant features in machine learning. In *Proceedings of the AAAI Fall symposium on relevance*, pages 1–5, 1994.

[128] Avrim L Blum and Pat Langley. Selection of relevant features and examples in machine learning. *Artificial intelligence*, 97(1-2):245–271, 1997.

[129] Mahmood Yousefi-Azar, Vijay Varadharajan, Len Hamey, and Uday Tupakula. Autoencoder-based feature learning for cyber security applications. In *2017 International joint conference on neural networks (IJCNN)*, pages 3854–3861. IEEE, 2017.

[130] Fangjun Kuang, Siyang Zhang, Zhong Jin, and Weihong Xu. A novel svm by combining kernel principal component analysis and improved chaotic particle swarm optimization for intrusion detection. *Soft Computing*, 19(5):1187–1199, 2015.

[131] Kai-mei Zheng, Xu Qian, and Pei-chong Wang. Dimension reduction in intrusion detection using manifold learning. In *2009 International Conference on Computational Intelligence and Security*, volume 2, pages 464–468. IEEE, 2009.

[132] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Kernel principal component analysis. In *International conference on artificial neural networks*, pages 583–588. Springer, 1997.

[133] Lawrence Cayton. Algorithms for manifold learning. *Univ. of California at San Diego Tech. Rep*, 12(1-17):1, 2005.

[134] David Wilson, Yufei Tang, Jun Yan, and Zhuo Lu. Deep learning-aided cyber-attack detection in power transmission systems. In *2018 IEEE Power & Energy Society General Meeting (PESGM)*, pages 1–5. IEEE, 2018.

[135] Miguel Nicolau, James McDermott, et al. A hybrid autoencoder and density estimation model for anomaly detection. In *International Conference on Parallel Problem Solving from Nature*, pages 717–726. Springer, 2016.

[136] Mayu Sakurada and Takehisa Yairi. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, page 4. ACM, 2014.

[137] Geoffrey Hinton and Ruslan Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[138] Guoqian Jiang, Haibo He, Ping Xie, and Yufei Tang. Stacked multilevel-denoising autoencoders: A new representation learning approach for wind turbine gearbox fault diagnosis. *IEEE Transactions on Instrumentation and Measurement*, 66(9):2391–2402, 2017.

[139] G. Jiang, P. Xie, H. He, and J. Yan. Wind turbine fault detection using a denoising autoencoder with temporal information. *IEEE/ASME Transactions on Mechatronics*, 23(1):89–100, Feb 2018.

[140] Robert Hecht-Nielsen. Theory of the backpropagation neural network. In *Neural networks for perception*, pages 65–93. Elsevier, 1992.

[141] Dhanalakshmi Sadhasivan and Kannapiran Balasubramanian. A novel LWCSO-PKM-based feature optimization and classification of attack types in SCADA network. *Arabian Journal for Science and Engineering*, 42(8):3435–3449, 2017.

[142] C. Hu, J. Yan, and C. Wang. Advanced cyber-physical attack classification with extreme gradient boosting for smart transmission grids. In *2019 IEEE Power Energy Society General Meeting (PESGM)*, accepted.

[143] Y. Lin, J. Wang, and M. Cui. Reconstruction of power system measurements based on enhanced denoising autoencoder. In *2019 IEEE Power Energy Society General Meeting (PESGM)*, accepted.

[144] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A Ghorbani. A detailed analysis of the kdd cup 99 data set. In *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, pages 1–6. IEEE, 2009.

[145] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[146] Salvador García and Francisco Herrera. Evolutionary undersampling for classification with imbalanced datasets: Proposals and taxonomy. *Evolutionary computation*, 17(3):275–306, 2009.

[147] Opeyemi Osanaiye, Haibin Cai, Kim-Kwang Raymond Choo, Ali Dehghantanha, Zheng Xu, and Mqhele Dlodlo. Ensemble-based multi-filter feature selection method for ddos detection in cloud computing. *EURASIP Journal on Wireless Communications and Networking*, 2016(1):130, 2016.

[148] Basant Agarwal and Namita Mittal. Optimal feature selection for sentiment analysis. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 13–24. Springer, 2013.

[149] Zubair A Baig, Sadiq M Sait, and AbdulRahman Shaheen. Gmdh-based networks for intelligent intrusion detection. *Engineering Applications of Artificial Intelligence*, 26(7):1731–1740, 2013.

[150] Nir Nissim, Robert Moskovitch, Lior Rokach, and Yuval Elovici. Detecting unknown computer worm activity via support vector machines and active learning. *Pattern Analysis and Applications*, 15(4):459–475, 2012.

[151] Noelia Sánchez-Maroño, Amparo Alonso-Betanzos, and María Tombilla-Sanromán. Filter methods for feature selection–a comparative study. In *International Conference on Intelligent Data Engineering and Automated Learning*, pages 178–187. Springer, 2007.

[152] Tina R Patil, SS Sherekar, et al. Performance analysis of naive bayes and j48 classification algorithm for data classification. *International journal of computer science and applications*, 6(2):256–261, 2013.

[153] Wei Wang and Sylvain Gombault. Efficient detection of ddos attacks with important attributes. In *2008 Third International Conference on Risks and Security of Internet and Systems*, pages 61–67. IEEE, 2008.

[154] Yoav Freund, Robert E Schapire, et al. Experiments with a new boosting algorithm. In *icml*, volume 96, pages 148–156. Citeseer, 1996.

[155] Shengyi Pan, Thomas Morris, and Uttam Adhikari. Classification of disturbances and cyber-attacks in power systems using heterogeneous time-synchronized data. *IEEE Transactions on Industrial Informatics*, 11(3):650–662, 2015.