A Transfer Learning Framework for Self-Adaptive Intrusion Detection in the Smart Grid based on Transferability Analysis and Domain-Adversarial Training

Pengyi Liao

A Thesis
in
The Department
of
Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of
Master of Applied Science in Electrical and Computer Engineering at
Concordia University
Montréal, Québec, Canada
December 2022

CONCORDIA UNIVERSITY
School of Graduate Studies

This is to certify that the thesis prepared

By: **Pengyi Liao**

Entitled: **A Transfer Learning Framework for Self-Adaptive Intrusion Detection in the Smart Grid based on Transferability Analysis and Domain-Adversarial Training**

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

Master of Applied Science in Electrical and Computer 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. Chunyan Lai

_____ Examiner
Dr. Suryadipta Majumdar

_____ Thesis Supervisor
Dr. Jun Yan

_____ Co-supervisor
Dr. Mohsen Ghafouri

Approved by _____
Dr. Zahangir Kabir, Graduate Program Director

December 6, 2022 _____
Dr. Mourad Debbabi, Dean
Gina Cody School of Engineering and Computer Science

Abstract

A Transfer Learning Framework for Self-Adaptive Intrusion Detection in the Smart Grid based on Transferability Analysis and Domain-Adversarial Training

Pengyi Liao

Machine learning is a popular approach to security monitoring and intrusion detection in cyber-physical systems (CPS) like the smart grid. General ML approaches presume that the training and testing data are generated by identical or similar independent distribution. This assumption may not hold in many real-world systems and applications like the CPS, since the system and attack dynamics may change the data distribution and thus fail the trained models. Transfer learning (TL) is a promising solution to tackle data distribution divergence problem and maintain performance when facing system and attack variations. However, there are still two challenges in introducing TL into intrusion detection: when to apply TL and how to extract effective features during TL.

To address these two challenges, this research proposes a transferability analysis and domain-adversarial training (TADA) framework. This work first proposes a divergence-based transferability analysis to decide whether to apply TL, then develops a spatial-temporal domain-adversarial (DA) training model to reduce distribution divergence between two domains and improve attack detection performance. The main contributions include: *(i)* A divergence-based transferability analysis to help evaluate the necessity of

TL in security monitoring for CPS, such as intrusion detection in the smart grid; *(ii)* A spatial-temporal DA training approach to extract the spatial-temporal domain-invariant features to mitigate the impact of distribution divergence and enhance detection performance. The extensive experiments demonstrate that the transferability analysis is capable of predicting accuracy drop and determining whether to apply TL. Compared to the state-of-the-art models, TADA can achieve high and more robust detection performance under system and attack variations.

Acknowledgments

I would love to express my heartiest gratitude to my incredible supervisors, Dr. Jun Yan, Dr. Mohsen Ghafouri, and Dr. Jean Michel Sellier for their constructive guidance and insightful comments throughout the journey. My research work won't be a possibility without their supervision. I also want to show gratitude to Dr. Chunyan Lai, and Dr. Suryadipta Majumdar for being the committee members to examine my thesis.

In addition, I am grateful to Concordia for its facilities, services, and courses. This research is funded in part by the Ericsson GAIA Montréal AI hub Canada and Mitacs Accelerate grant IT-15923, in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) under grants RGPIN-2018-06724, and in part by the Fonds de Recherche du Québec–Nature et Technologies (FRQNT) under grant 2019-NC-254971.

Next, I would like to thank all my team fellows: Tianyu Chen, Yuanliang Li, Yongxuan Zhang, Juanwei Chen, Hangdu Du, Moshfeka Rahman, Jeremy Frandon, Chengming Hu, William Lardier, Luyang Hou, and Quentin Varo for helping me in various ways.

Finally, this journey won't be possible without my loving father, mother, and sister. I also want to show my sincerest gratitude to everyone else for all the help and accompany over the past three years.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

1.1.1 Smart Grid Security Challenges

In recent years, cyber-physical systems (CPS), which integrate smart sensors, networking, computing, and control technology, are causing great changes in modern industry [9, 10]. As a trans-continental CPS infrastructure, the smart grid connects utilities and customers with two-way power and information flows to provide more efficiency, reliability, and safety of power delivery. As illustrated in Figure 1, a smart grid is mainly composed of physical systems and cyber systems [11]. The physical system, i.e. power grids, consists of power generation, transmission, distribution, and customers. As a next-generation power system, the physical system also includes various distributed generation and storage, such as solar and wind energy. The cyber systems leverage various communication networks, like local area networks (LANs), field area networks (FANs), wide area

networks (WANs), etc., to connect multiple utilities and customers [12]. With the in-depth communication and integration of physical systems and cyber systems, smart grids can realize intelligent and reliable management of power generation, transmission, distribution, and power consumption.

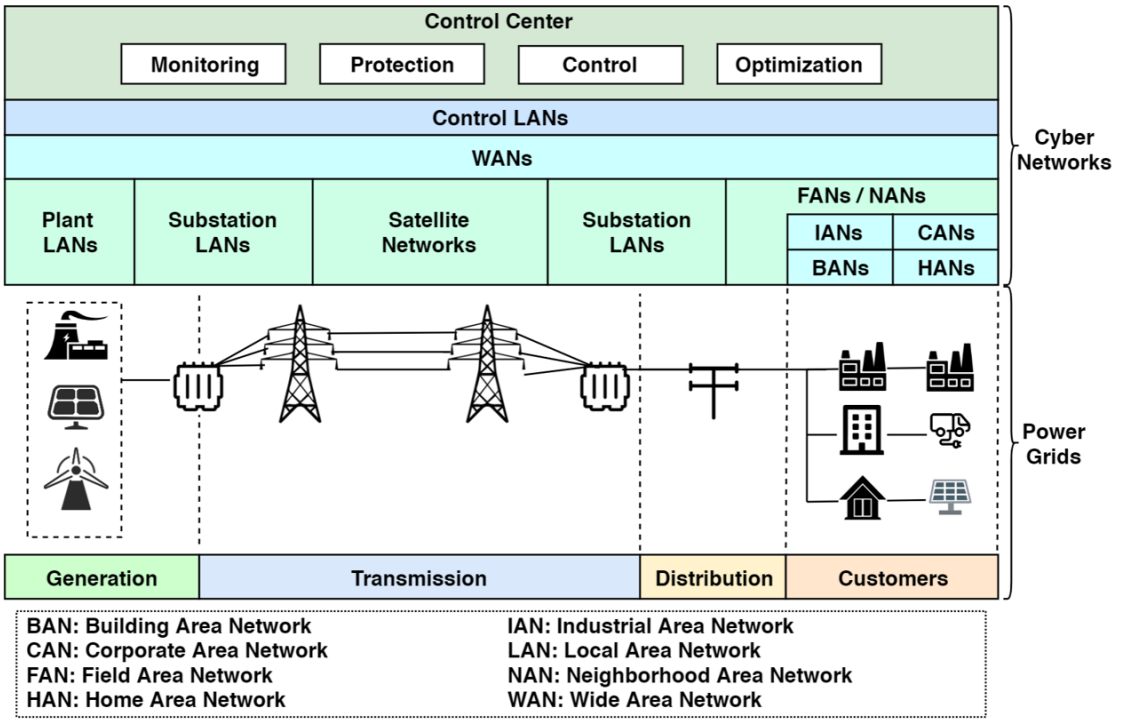


Figure 1: The cyber-physical architecture of the smart grid [1].

However, the growing number of interconnections among billions of cyber-physical devices creates complex interdependence and vulnerabilities that will inevitably raise the occurrence of cyber-attacks in power systems. In recent years, power grid security accidents have brought great threats to life and the economy, as shown in Figure 2. For example, in 2010, the Stuxnet infected the supervisory control and data acquisition (SCADA) systems in an Iranian nuclear program and damaged the centrifuges [13]. In 2015, hackers infected the SCADA system with the BlackEnergy Trojan virus, and carried out targeted

network attacks on a number of energy companies and power distribution companies, causing power outages in the Ivano-Frankivsk region of Ukraine [14]. 225,000 customers suffered from power outages that lasted for hours. In 2016, the Israel electricity authority was infected by a computer virus, and many computers in the authority were paralyzed for two days [15]. In the same year, the power supply of Kyiv's electricity grid was disrupted by Crashoverride malware, causing a partial blackout [16]. In 2017, the electricity transmission lines in Turkish were attacked by cyber-incidents, resulting in electricity cuts in Istanbul [17]. In 2018, cyber-attacks accessed control systems at U.S. power plants to shut down power plants [18]. In 2019, the electricity grid in Venezuela was attacked by hackers, affecting 23 states of the country [19]. In 2020, the Maharashtra power system in India was targeted by cyber-attackers during the Galwan crisis [20].

Based on the aforementioned accidents, we can find that the impact of cyber-physical attacks on the smart grid could be grievous and disastrous, as reported in recent studies [14, 21, 22, 23, 24]. Hence, it is of great importance to achieve high cyber security in the smart grid.

1.1.2 Intrusion Detection System

In general, in the smart grid, the unauthorized activities that could compromise the confidentiality, integrity, and availability (CIA) of the power systems could be treated as an intrusion [25]. Intrusion detection systems (IDS) are important CPS security techniques that protect the state of networks or systems from internal and external malicious activities [5]. The objective of IDS is to identify the malicious actions that compromise the CIA of the smart grid. IDS monitors and analyzes the behaviors of networks or systems,

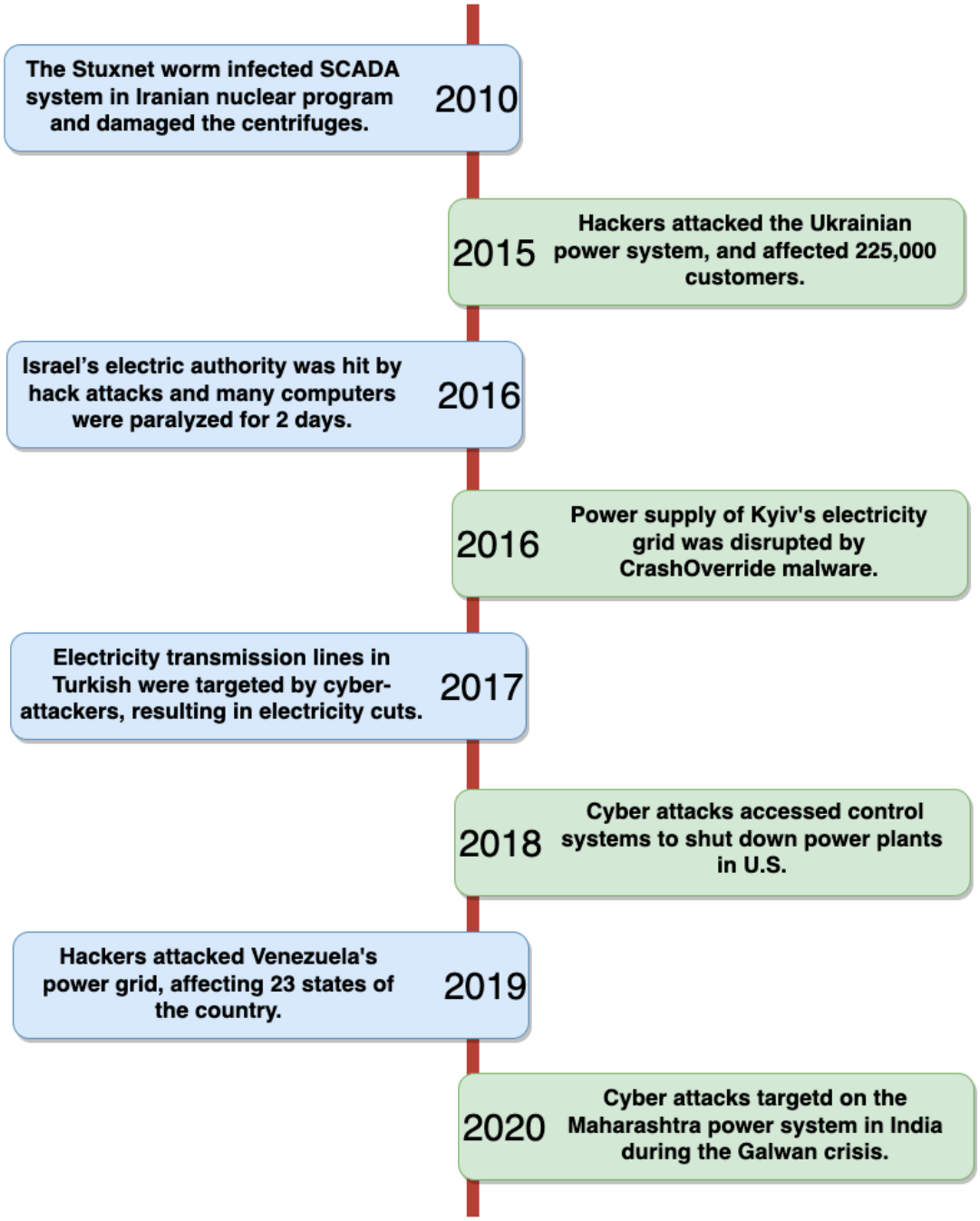


Figure 2: Power grid security incidents caused by cyber-physical attacks [2].

generates alerts, and responds to malicious activities [5]. Based on the differences in the database maintained, detection strategies, and challenges, the IDS can be categorized into 2 groups: misuse-based methods and anomaly-based methods [5, 4, 26].

- **Misuse-based IDS:** Misuse-based IDS are also called signature-based IDS. Misuse-based IDS basically take each attack as a signature and maintain a signature database. The detection process is to match an attack with the signatures of the previous intrusion in the signature database. When an attack is matched, the alarm will be triggered. The advantage of misuse-based IDS is its low false alarm rate. The disadvantage is that it has a high missed alarm rate because it can not identify unknown attacks since there are no prior signatures of such attacks in its profiles. Moreover, the misuse-based IDS requires maintaining and updating a large signatures database, which is expensive, time-consuming, and sometimes impossible due to emerging and fast-evolving attacks [27].

- **Anomaly-based IDS:** Anomaly-based IDS can identify the zero-day attacks to overcome the limitation of the misuse-based IDS. The basic idea of anomaly-based IDS is to set a database of normal behaviors and identify the abnormal behaviors that deviate from the normal behavior database. The advantage of anomaly-based IDS is that it can detect unknown attacks. The disadvantage is that it may have a high false alarm rate (FAR) since the new normal activities may be categorized as anomalies.

1.1.3 Smart Grid Data Integrity Attacks

Traditional IDS need to regularly update databases, which are expensive, time-consuming, and sometimes impossible due to emerging and fast-evolving attacks. While many statistical and knowledge-based IDS approaches have been proposed to identify attacks, there is a particular coordinated cyber attack in the power grid, called data integrity attack, that can access the measurement data and bypass the traditional surveillance and detection techniques [28].

In power systems, the state estimation is a key procedure to control power in the energy management system (EMS), which is responsible for estimating the current state of power systems, and can be further leveraged to filter the measurement noise and detect attacks, to maintain the security and stability of the systems [29]. As shown in Figure 3, the raw measurements of the power system are collected through the remote terminal units (RTUs) and sent via the SCADA networks to the control center. Then the state estimator leverages the collected measurements to estimate the system operation state and detect fault data, and the system controller will use the estimated state in the optimal power flow, economic dispatch, and contingency analysis [30].

However, with the incorporation of the cyber networks and the physical power grid, the state estimation is no longer immune to some data integrity cyber attacks [8]. In this research, we consider data integrity cyber attacks that can compromise power meter measurements and pose profound disruptions in state estimation, energy distribution, and real-time pricing. For instance, false data injection (FDI) attack is a particular class of intrusion that targets data integrity [3]. FDI can compromise measurements, inject malicious data to manipulate the estimated states stealthily, and evade bad data detection (BDD). As shown

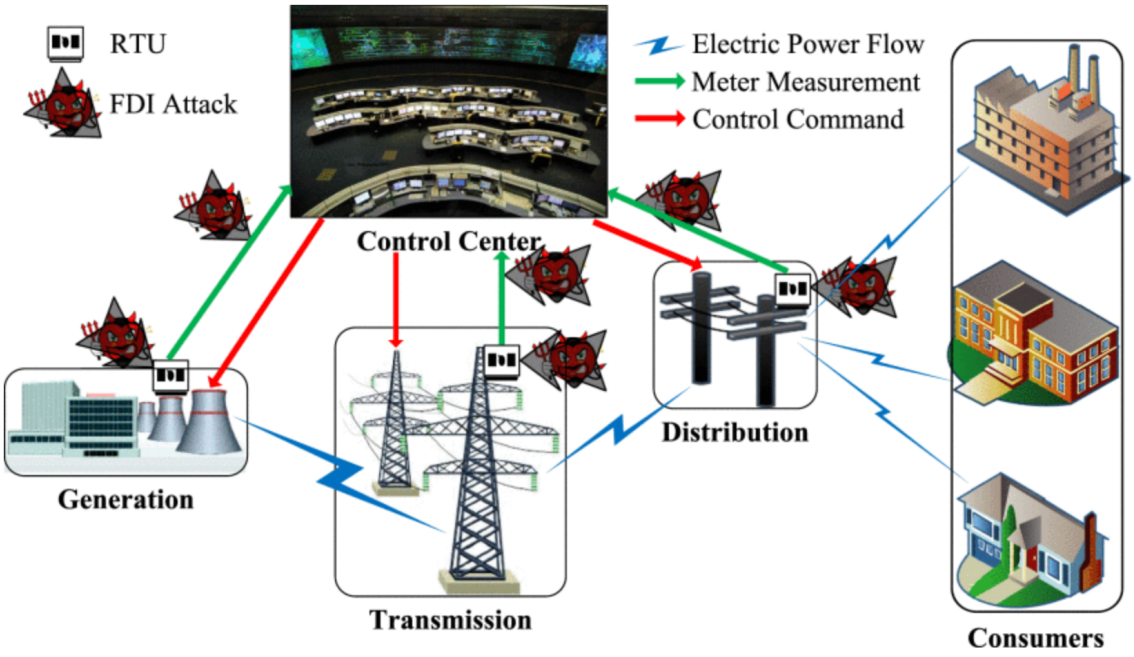


Figure 3: FDI attacks on state estimation in a power system [3].

in Figure 3, FDI attacks could have several ways to manipulate the measurement data with access to the end devices or the networks [3]. FDI can compromise the RTUs directly to manipulate the measurements such as voltage magnitudes and angles. Alternatively, FDI can also intercept the RTUs communication to manipulate the data from the meters to the control center. A successful FDI attack will evade detection and pose a severe threat to power system state estimation, possibly inflicting severe impacts like power outages, physical damages, and monetary losses.

1.2 Problem Statement

With increasing coupling among CPS devices, the traditional IDS approaches may not be adequate to detect data integrity cyber attacks. In the last several years, machine learning (ML) has claimed significant attention in detecting FDI in the smart grid security research community [31, 32]. Compared to traditional IDS, ML approaches can learn and leverage complex non-linear relationships to discover the difference between normal and attack data for attack detection [5]. Moreover, ML methods can also generalize well when exposed to novel attacks [32]. Hence, various ML detection mechanisms have been exploited extensively and demonstrated high accuracy and efficient computation in attack detection [31, 33, 34], such as k-nearest neighbor (kNN) and support vector machine (SVM).

Many ML-based attack detection models assume that the training and testing data are in the same space and have the same or similar independent distributions [1]. However, this assumption is unlikely to hold in most real-world CPS scenarios because of system dynamics and attack changes. For instance, in power grid, the system load demand is continuously changing, and the system topology may also be altered by normal operations. Meanwhile, the same scheme attacks may happen at different times and target different buses, and new scheme attacks are emerging as well. These variations will alter the data distribution and render a well-trained ML detection model to perform poorly on a new dataset. Moreover, labelled attack data is extremely rare compared to labelled normal data in real-world power systems. Models trained on insufficient data are fragile, and a small change in the attack data distribution may cause a significant drop in detection accuracy.

Transfer learning (TL) is hence proposed to help solve these problems. This technique

enables models to transfer the knowledge learned from a labelled domain to another unseen domain with distribution divergence [35]. Over the last few years, TL has shown remarkable achievements in image identification and semantic parsing tasks [36]. Recently, TL is also introduced into highly dynamic CPS scenarios to enhance cyber-security situation awareness [37]. However, there are still two challenges that need to be considered when applying TL for attack detection in the smart grid. The first challenge might be referred to as *transferability*: when a model will suffer a severe performance drop and thus TL should be applied? The second challenge is how to extract effective features of the power system data during the TL.

To tackle these two challenges, we propose a transferability analysis and domain-adversarial training (TADA) framework. The proposed framework has two steps. The first step is to leverage selected data divergence metrics and regression models to predict detection accuracy drop and identify the tasks calling for TL. Then the framework develops a spatial-temporal DA training approach to extract spatial-temporal domain-invariant features to enhance attack detection performance. The approach leverages parallel long short-term memory (LSTM) networks and convolutional neural networks (CNN) to extract spatial-temporal features, and employ DA training to reduce distribution divergence between two domains and thus improve detection performance.

1.3 Contributions

The main contributions of this research are summarized as follows:

- We formulate the problems of when to apply TL and how to extract effective features during TL for attack detection in power systems and tackle the problems by proposing a two-step TL framework.
- We propose a divergence-based transferability analysis to help evaluate the necessity of TL in security monitoring for CPS, such as intrusion detection in the smart grid.
- We develop a spatial-temporal DA training approach, which is able to extract spatial-temporal domain-invariant features to enhance attack detection performance under system variations and attack variations.

1.4 Thesis Structure

The thesis is organized into 6 chapters, and the overall structure is shown in Figure 4.

Chapter 1 presents the background, problem, and motivation of the thesis.

Chapter 2 discusses the related work about intrusion detection techniques in the smart grid, and various TL approaches that are applied in the IDS, especially the domain adaptation methods. This chapter also reviews the data distribution divergence metrics used to evaluate transferability.

Chapter 3 presents the overview of the proposed two-step attack detection framework. The first step is to determine when to apply TL, and the second step is to effectively extract critical features during TL. The details of each step are illustrated in the following two chapters, respectively.

Chapter 4 proposes a divergence-based transferability analysis to justify the necessity

of TL. Specifically, we first leverage three commonly-used metrics to evaluate the distribution divergence, and train two regression models for each metric to approximate the relation between accuracy drop and divergence. Then the regression models are used to predict an accuracy drop in determining whether to apply TL. Moreover, considering combining different metrics to extract complementary distribution divergence information, we also propose an ensemble method that combines all metrics to further improve prediction performance.

Chapter 5 proposes a spatial-temporal DA training approach to learn domain-invariant representations to detect returning threats at different times and locations. The framework leverages LSTM and CNN as the feature extractor to concurrently extract spatial-temporal domain-invariant features from the multivariate data on time and space dimensions, mitigating the impact of distribution divergence and thus improving detection performance.

Chapter 6 draws the conclusions and presents the potential future works of the research.

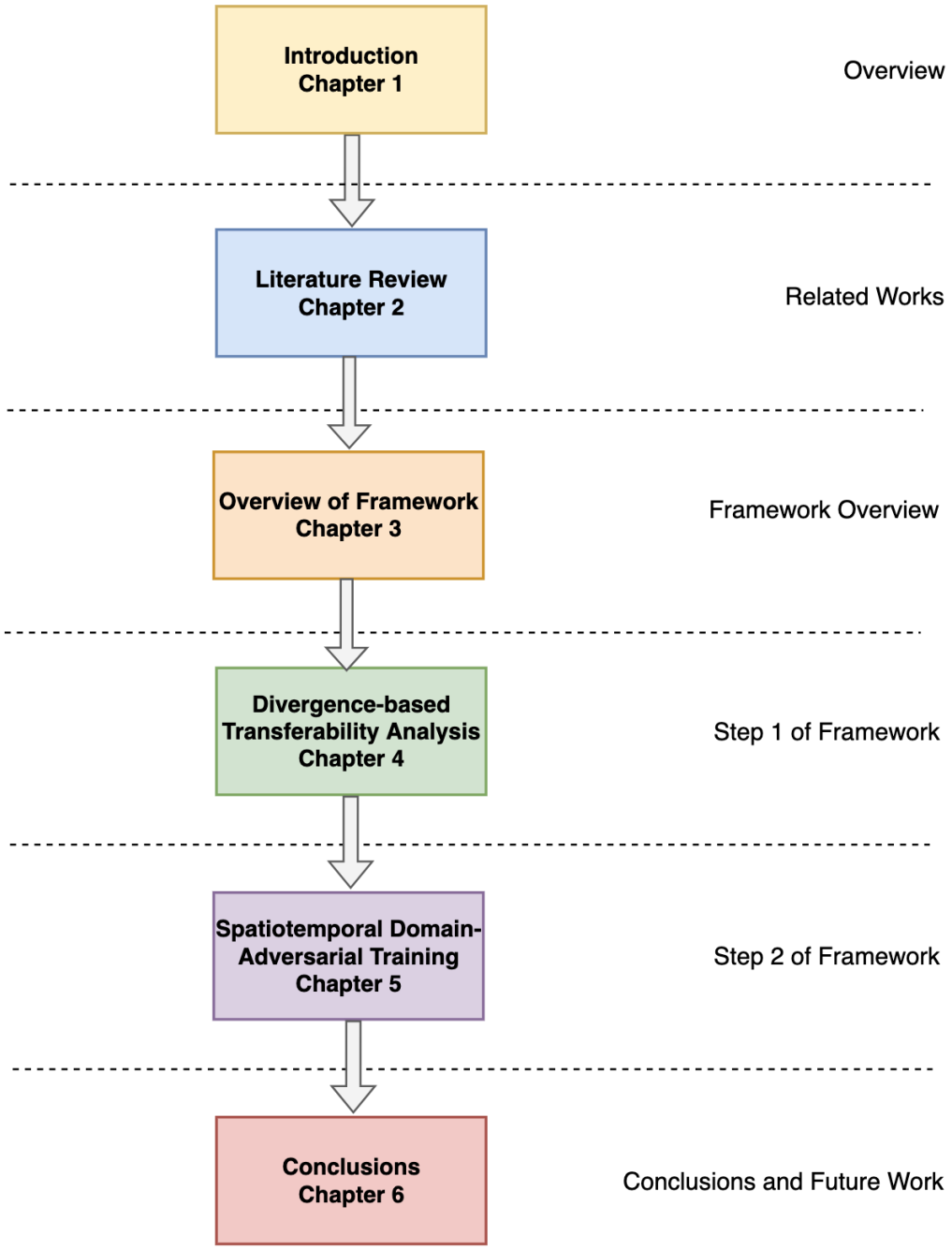


Figure 4: Structure of the Thesis.

Chapter 2

Literature Review

2.1 Intrusion Detection in the Smart Grid

A wide range of IDS approaches has been proposed to detect attacks in the smart grid [38, 39, 32, 40]. There are two common methods to classify IDS: detection-based and source-based. According to the difference in detection methods and database maintained, IDS can be categorized into misuse-based methods and anomaly-based methods, as illustrated in the Subsection 1.1.2. According to the difference in input data sources used to detect malicious activities, IDS can also be categorized into host-based methods and network-based methods. In this research, since we are interested in applying ML techniques to detect attacks in the smart grid, we analyze the intrusion detection literature and categorize them into non-learning-based and learning-based techniques, as shown in Figure 5.

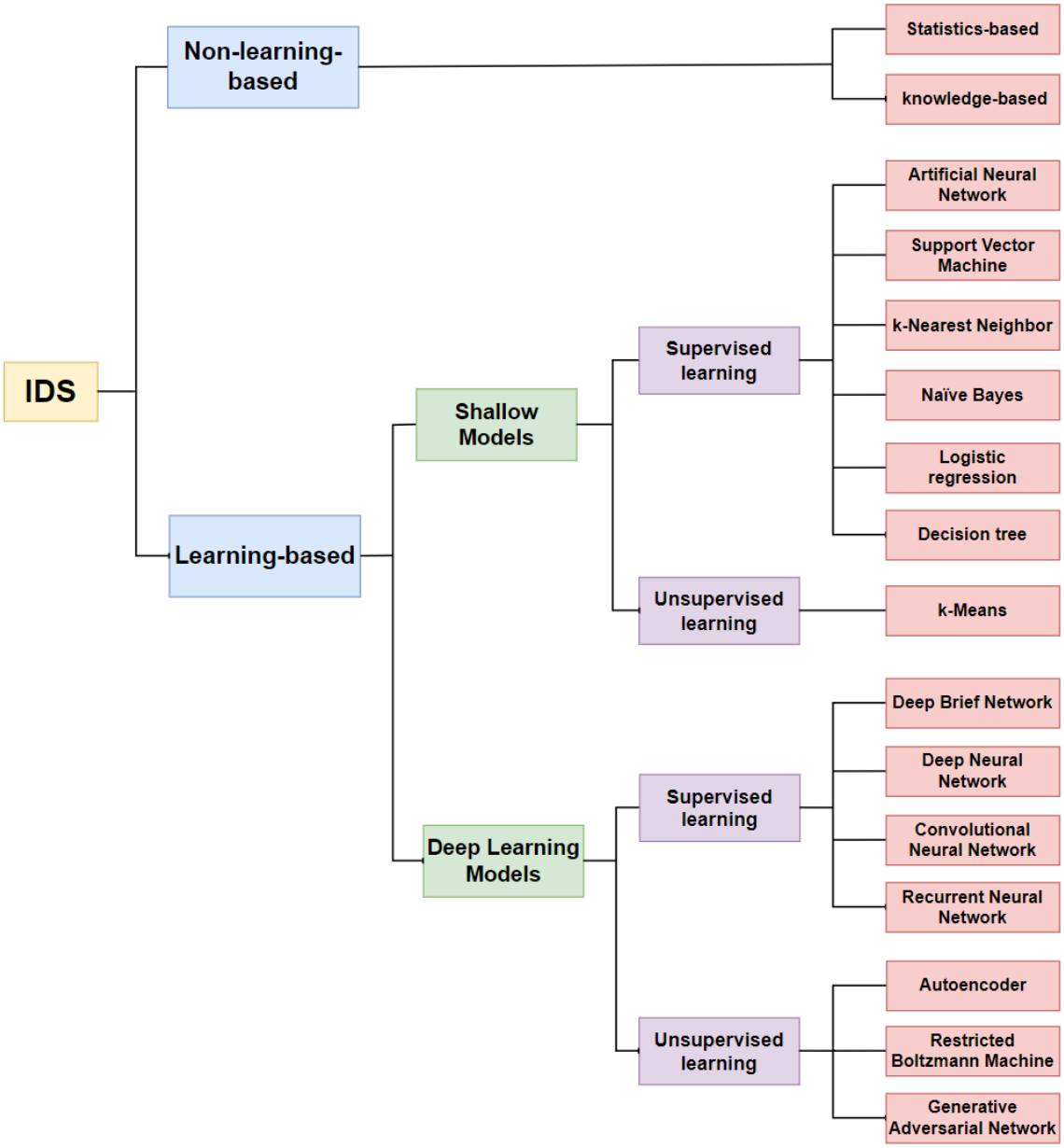


Figure 5: Taxonomy of IDS [4, 5].

2.1.1 Non-Learning-based Techniques

Khraisat *et al.* [4] conducted a comprehensive survey and classified the non-learning-based IDS techniques into two categories: statistics-based and knowledge-based. Statistics-based IDSs [41, 42, 43, 44] first measure the statistical metrics of packets like mean, median, and deviation, then leverage the statistical metrics to build a statistical model for the normal data. Finally, a statistical reference test is used to calculate the probability that a downstream instance belongs to the normal model. An instance with a low probability will be considered an attack behaviour [45, 46, 47].

Boero, Dusi, and Aiello *et al.* [48, 49, 50] propose a statistical fingerprint-based IDS. Instead of inspecting the packets, they examine a group of parameters related to the traffic flow and get the statistical measurements as the fingerprint. Then, the fingerprint is used to determine whether the traffic is normal. Zheng *et al.* [38] propose a hierarchical IDS, which identifies the intrusions based on statistical processing. Their method collects and abstracts traffic information and converts it into status-related statistical variables. The statistical model then compares the statistical variables with the typical network activities and determines whether the traffic is affected by the malware. Wattenberg *et al.* [51] introduce a nonrestricted α-stable first-order model to detect intrusions in the network traffic. They use α-stable functions to model the marginal distribution of real traffic, and apply the means of a generalized likelihood ratio test (GLRT) to classify traffic patterns. The results indicate their method has high accuracy in detecting floods and flash crowds.

Knowledge-based IDS [52, 53, 54, 55, 56] first uses normal traffic data to establish a knowledge base. The instances that vary from the standard knowledge base will be considered intrusions. This knowledge base is basically created based on human knowledge, i.e.,

a group of human-defined rules. Thus, knowledge-based IDS is also called an expert system. The advantage of knowledge-based IDS is that it can decrease the false positive rate because the system has a collective knowledge base of various normal behaviours. However, in an open and dynamic environment, like CPS, it is expensive and time-consuming to update the knowledge regularly. Alcaraz *et al.* [39] propose a rule-based expert system where only the authenticated and authorized entities can access distributed elements. Practical studies show that their method can not only accept or deny access but also provide approaches to dealing with extreme situations. Le *et al.* [57] introduce a finite state machine (FSM) based IDS. If the FSM frequently changes its topology and the number of changes exceeds the predefined threshold, the behaviour will be considered an attack. The experiment results indicate their method can effectively monitor topology attacks with reasonable overhead.

2.1.2 Learning-based Techniques

The purpose of intrusion detection is to distinguish intrusion behaviours from normal behaviours and to categorize specific intrusion behaviours according to their characteristics. From the perspective of ML, intrusion detection can be regarded as a standard classification problem. ML has been proven to be efficient for classification tasks in many application domains. In IDS, ML can not only learn significant differences between intrusions and normal behaviours but also have better adaptability to new intrusions than conventional IDS because of its generalization. In recent years, a wide variety of ML techniques have been introduced into intrusion detection in the smart grid [26, 32, 58, 59].

In this subsection, according to the depth of the model structure, we follow [5] and categorize common ML algorithms into two types and discuss their applications in the IDS.

Shallow Models

Some shallow ML models have been studied for several decades, and their methodologies are mature. These models focus on detection accuracy and application efficiencies, such as computation time and deployment complexity [5]. Shallow ML models can be generally classified into two types, namely, supervised and unsupervised.

Supervised learning uses labelled data to train a classifier, then applies the classifier to classify the testing data into intrusions or normal behaviours [31]. A rich line of supervised learning methods, such as multi-layered perceptron (MLP), kNN, SVM, decision tree (DT), naïve Bayes, etc., has been explored in the IDS in the literature [33, 34]. Ozay *et al.* [32] test and compare three classic supervised ML algorithms for false data injection (FDI) attacks, including MLP, kNN, and SVM. They find that kNN is sensitive to the system size, and its performance may degrade in large-size systems. On the contrary, SVM and MLP perform well in large-size systems. Jindal *et al.* [60] use a two-phase approach to detect energy fraud in the smart grid. The electricity consumption data is firstly processed by the DT and then fed into SVM. The experiments show that this top-down approach can identify energy fraud behaviours with an accuracy of over 92% and a false positive rate (FPR) of 5%.

In the CPS, sometimes there is not sufficient labelled data, manually labelling data is expensive, time-consuming, and sometimes impossible due to emerging and fast-evolving

attacks. Unsupervised learning can address this problem by extracting interesting information from unlabelled data. Menon *et al.* [61] propose a k-Means approach to cluster traffic between utility centers and smart homes, and identify anomalies. The experimental results show their approach can achieve higher accuracy in detecting attacks than other clustering algorithms. Alseiari *et al.* [62] introduce a mini-batch K-means IDS to monitor the data traffic in the advanced metering infrastructure (AMI) and detect anomalies in real time. Compared with other online clustering techniques, their approach has a high detection rate and a low FPR.

Deep Learning Models

Deep learning models can take advantage of their complex structures and vast numbers of parameters to outperform shallow models in many application scenarios, thus attracting more and more interest in the community of CPS security [5]. The number of deep learning-based IDS studies has increased incredibly since 2015 [63]. Deep learning models can also be categorized into supervised learning and unsupervised learning.

Supervised deep learning models include deep belief network (DBN), deep neural network (DNN), CNN, recurrent neural network (RNN), etc. He *et al.* [40] propose a deep learning approach to detect FDI with historical measurements. They use a conditional deep belief network (CDBN) to extract high-dimensional temporal features from different sensor measurements and identify FDI attacks. The experiments on IEEE 118-bus and 300-bus power systems indicate the proposed method has a high detection accuracy. Min *et al.* [64] introduce a CNN-based IDS, which makes use of both statistical features

and payload features. They utilize CNN to extract effective information and train a sophisticated random forest for classification. Wang *et al.* [65] propose a novel IDS which combines the CNN and LSTM. The approach first uses CNN to learn low-level spatial features, then learn high-level temporal features with LSTM. The experimental results show that their method can effectively decrease the false alarm rate.

Unsupervised deep learning models contain autoencoder, restricted Boltzmann machine (RBM), and generative adversarial network (GAN), etc. Rigaki *et al.* [66] use GAN to modify the malware's traffic behaviours to mimic legitimate traffic to avoid detection. The GAN guides the malware to generate network traffic similar to real Facebook chat traffic. Results demonstrate that the GAN can successfully improve the chances of the intrusions not being blocked. Zhang *et al.* [27] introduce a GAN-based IDS to address the problem of a limited number of samples. The proposed method first uses Monte Carlo methods to generate synthesized intrusion data, augments the synthesized data by GAN, and finally utilizes the augmented data to train a classifier to identify intrusions. Results show their approach outperforms other methods in accuracy, precision, recall, and F1-score.

Based on the aforementioned literature review, we can find that learning-based methods have been widely employed in the IDS. However, the CPS has some peculiarities that may make ML methods harder to use. General ML-based attack detection models assume that the training and testing data are in the same space and have the same or similar independent distributions [1]. In these cases, an ML model is trained and can be used for a long time without changes. However, this assumption is unlikely to hold in most real-world CPS scenarios. The data distribution of a CPS may change dynamically because

of system and attack variations. If the data distribution changes, the trained model will degrade, and we need to train a new model from scratch with the newly collected data. In many real-world applications, recollecting data and retraining the models are expensive and time-consuming. Such a gap requires techniques that can help trained models adapt to new datasets with different data distributions.

2.2 Transfer Learning

2.2.1 Overview of Transfer Learning

Transfer Learning is hence proposed to address the data distribution divergence. TL is an ML technique that transfers the knowledge learned from one domain to a different but similar domain [67]. In the real world, there are lots of cases of applying TL. For instance, learning to play badminton may help one learn to play tennis. Similarly, learning to ride a bicycle may help one learn to ride a motorcycle.

To elaborate on how TL addresses the aforementioned gap, we first define several notations of TL. In TL, a domain $\mathcal{D}$ consists of a feature space $\mathcal{X}$ and a marginal probability distribution *P(X)*. A task consists of a label space $\mathcal{Y}$ and an objective predictive function $f(\cdot)$ from $\mathcal{X}$ to $\mathcal{Y}$ [37]. The objective function $f(\cdot)$ also refers as conditional probability $P(Y|X)$ from a probabilistic view, which is learned from the training data [35]. Given a source domain $\mathcal{D}_S$ and a target domain $\mathcal{D}_T$, the objective of TL is to learn and transfer the knowledge from the source domain to achieve a high performance on the target domain. Depending on label availability in the source and target domains, TL problems can be categorized into three types: transductive TL, inductive TL, and unsupervised TL [68], as

shown in Figure 6.

In inductive TL, labelled data is available in the target domain no matter whether the labelled data is available in the source domain or not [69, 70, 71, 72, 73]. The tasks of the source and target are different regardless of whether their domains are identical. If some labelled data is available in the source domain, inductive TL is similar to multitask learning. If the labelled data is unavailable in the source domain, inductive TL is close to self-taught learning.

In transductive TL, the labelled data is only available in the source domain [74, 75, 76, 77, 78]. The source task and target task are the same, but their domains are different. According to the differences between the two domains, transductive TL can be categorized into two types: either the feature spaces of the source and target domains are different, or the marginal distributions are different.

In unsupervised TL, there is no labelled data in either the source or the target domains [68]. The source task and target task are different but related. Unsupervised TL can benefit special tasks, where sufficient labelled data in both the source domain and target domain is not available.

TL has been extensively adopted and witnessed remarkable advances in image and video applications [79]. Lately, TL methods are applied to anomaly detection in Internet [67, 68, 80, 81, 82] and cloud [83] applications, which shed new light on introducing TL to empower intrusion detection in highly dynamic cyber-physical power systems [1, 35].

2.2.2 Domain Adaptation in TL

Domain adaptation is a subcategory of transductive TL techniques that reduces domain divergence and improves models' generalization [67]. In domain adaptation, given a labelled source domain $\mathcal{D}_S = \{(x_{S_1}, y_{S_1}), ..., (x_{S_{n_1}}, y_{S_{n_1}})\}$, and an unlabelled target domain $\mathcal{D}_T = \{(x_{T_1}), ..., (x_{T_{n_2}})\}$, we assume that the feature space of the source and target domains is identical, i.e., $\mathcal{X}_S = \mathcal{X}_D$, the label space is identical, i.e., $\mathcal{Y}_S = \mathcal{Y}_D$, and their conditional distributions are same, i.e., $P_{D_S}(Y|X) = P_{D_T}(Y|X)$. But the marginal probability distributions of the source and target domains are different, i.e., $P_{D_S}(X) \neq P_{D_T}(X)$. The aim of domain adaptation is to leverage the labelled data in the source domain to learn a function $f(\cdot)$ to predict the label in the target domain. Csurka and Wang *et al.* [84, 85] reviewed the domain adaptation methods and classified them into three types: discrepancy-based methods, adversarial-based methods, and reconstruction-based methods.

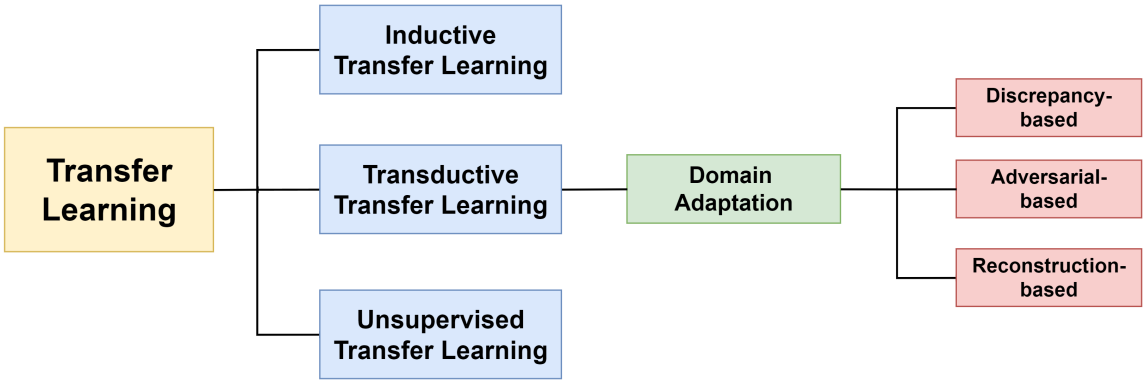


Figure 6: Classification of TL.

Discrepancy-based methods [86, 87, 88, 89, 90] reduce the generalization error of the target domain by reducing the difference between the two domains. Maximum mean discrepancy (MMD) is one of the most commonly used measures of the difference between the distributions of the source and target domains. Domain adaptive neural network [91]

and deep domain confusion (DDC) [92] are the earliest MMD-based deep domain adaptation methods. DDC only uses linear kernel MMD in a layer to measure the discrepancy. Long *et al.* improve the DDC by using multiple kernels MMDs and applying them in several layers [93]. They further propose the joint adaptation network (JAN) [94]. JAN considers the joint probability distribution of features and labels, resulting in further performance improvement.

Adversarial-based methods [77, 95, 96, 97, 98] introduce the idea of GAN [99] into the domain adaptation problem. Ganin *et al.* propose a domain-adversarial (DA) training strategy to extract feature representations that are both label-discriminative and domain-invariant [72]. The training process of adversarial domain adaptation is a game process between the feature extractor and the domain discriminator: the domain discriminator learns to distinguish the source domain samples from the target domain samples, while the feature extractor tries to confuse the domain discriminator by learning domain-invariant features [100]. Volpi *et al.* introduce an adversarial discriminative domain adaptation (ADDA), where the weights are not shared, and the source domain features and the target domain features are extracted independently [101]. Long *et al.* propose a conditional domain-adversarial network (CDAN) [102] to align multimodal distributions.

Reconstruction-based methods [103, 104, 105, 106, 107] use autoencoders to reconstruct data to keep the inter-class representation distinguishable and inter-domain representation indistinguishable. Autoencoder [108], including two processes of encoding and decoding, is an unsupervised learning method that can be used to suppress information loss. Glorot and Chen *et al.* [109, 110] train an autoencoder using all samples from the

source and target domains, then train a classification model on the source domain representations, and apply the classification model directly to the target domain. Bousmalis *et al.* [107] introduce a domain separation network (DSN). In DSN, the source and target domains use the domain-shared encoder to encode the domain-shared information and use the domain-specific encoder to encode the domain-specific information. Hence, both common properties and domain-specific properties are extracted.

2.3 Transferability Analysis

2.3.1 Overview of Transferability Analysis

In this research, transferability analysis refers to the approach to analyzing the data and determining whether there is a need to apply TL. In most TL papers, researchers assume there is a need for TL and propose novel TL approaches to achieve state-of-the-art performance. However, in real-world applications, frequently applying TL is costly and time-consuming. If the ML model trained on the source domain can generally retain a good performance when applied to the target domain, there is no need for TL because the performance improvement would be trivial. Meanwhile, Weiss *et al.* state in their survey that if the source domain and target domain are not well-related, the knowledge learned from the source domain will have a negative impact on the target domain, which is referred to as negative transfer [111]. In this case, one would better train a new model from scratch instead of applying TL. Thus, it is vital to identify the cases suitable for TL and apply TL promptly.

Ben *et al.* [112] conducted a theoretical study and proved that a classifier's error in

the target domain is bound by its error in the source domain and the divergence between the source and target domains. Other studies also indicate that the model generalization error in the target domain is affected by the difference between the source domain and the target domain [67, 100]. Since when to transfer depends on whether the performance of a trained model trained in the source domain has degraded significantly in the target domain, and the accuracy drop from the source domain to the target domain is related to the data distribution divergence of them, the transferability analysis could lead to a data distribution divergence measurement problem.

There are some studies that have used data distribution divergence to detect attacks directly in the power system without TL. Gu *et al.* [113] use Kullback–Leibler (KL) divergence to calculate the distance between normal and false data to identify the latter directly. Pal *et al.* [114] measure the Euclidean distance between real and tampered data to detect the data manipulation attacks directly. Gupta *et al.* [115] use the relative entropy between normal and the perturbed power flow data, to predict the blackout risk. However, these studies mainly focus on measuring the dissimilarity between two data distributions and generating alerts for anomalies directly. Little attention has been paid to relating the distribution divergence with accuracy. Compared to these works, our transferability analysis focuses on predicting the performance degradation based on the divergence to decide when TL should be triggered, instead of measuring the dissimilarity for a direct alert.

Recently, several studies outside the field of TL have started to establish a connection between distribution divergence and accuracy. Among them, the most related work is from Elsahar and Deng *et al.* [116, 117], who used various methods such as H-divergence,

Fréchet distance, and confidence-based metrics to predict the accuracy drop of modern natural language processing (NLP) and computer version (CV) models under domain shifts. Both studies used predicted accuracy drops to evaluate the robustness of trained models. However, these studies did not further explore the use of the predicted accuracy to determine whether/how the model shall be updated to retain the previous performance. Instead, in our work, we use divergence metrics as an indicator to determine whether one should apply TL, benchmarking divergence metrics in predicting the accuracy drop to create a reliable predictor that will help operators decide whether to apply TL based on the predicted performance degradation of trained machine learning models.

2.3.2 Domain Divergence Metrics

Based on the aforementioned analysis, we can find that domain divergence plays a crucial role in predicting the accuracy drop when a trained model is applied to the target domain. Given its importance, researchers have invested much effort in leveraging metrics to evaluate distribution divergence. Kashyap and Ruder *et al.* [118, 119] conduct a comprehensive survey on the domain divergence metrics, and classify them into four groups: geometry-based, domain discrimination-based, mutual information-based, and higher-order moment-based, as shown in Figure 7.

Geometry-based metrics use the statistical descriptions of data distribution, like mean and standard deviation, to capture geometry-related information. Wang *et al.* [120] use Euclidean distance to measure the distance between different sentences and select the in-domain sentences. The adaptation results indicate their method can improve the neural

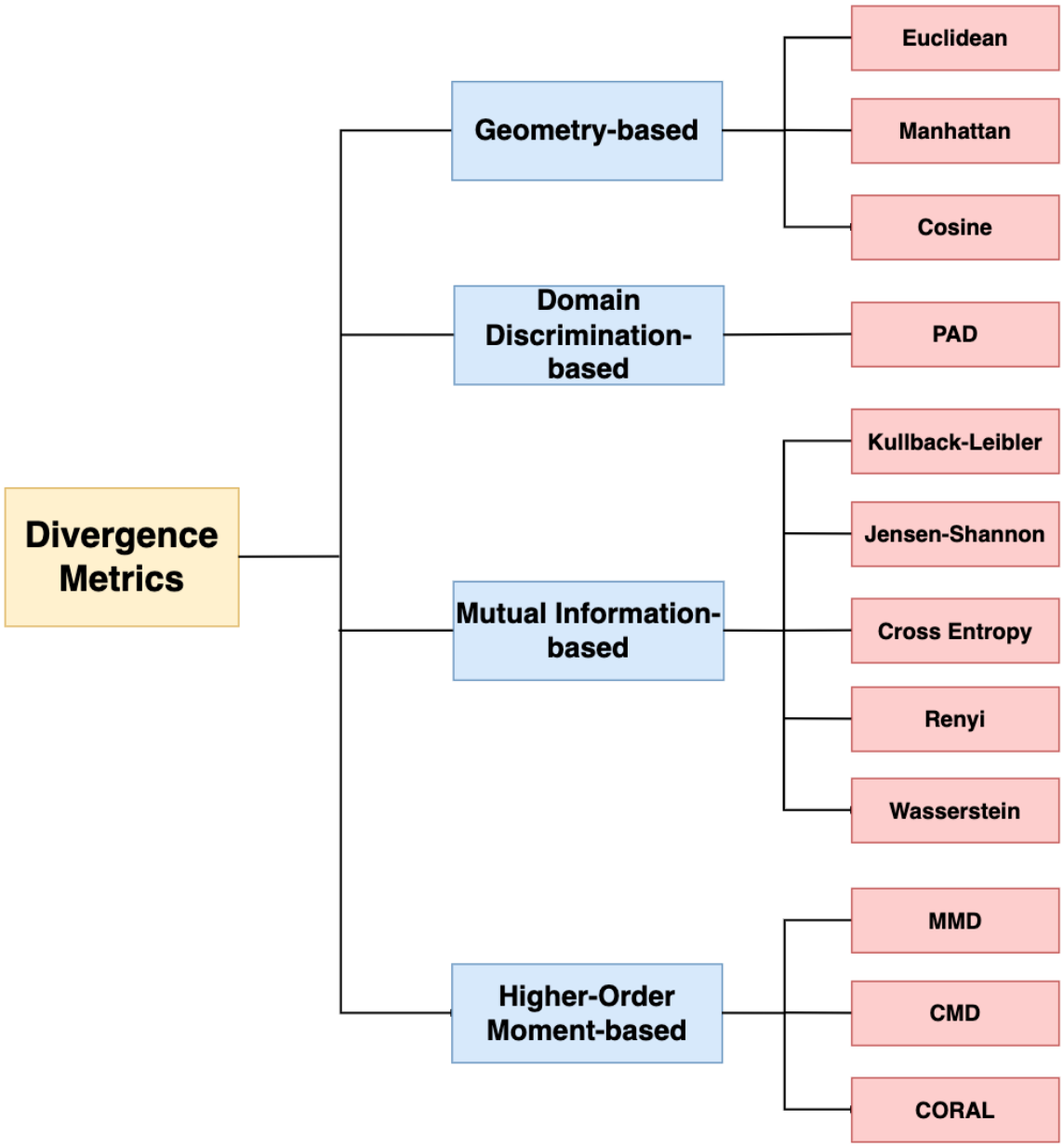


Figure 7: Taxonomy of Domain Divergence Metrics.

machine translation (NMT) performance. Ruder *et al.* [119] take the probability distributions as the vectors and leverages cosine similarity to calculate the distance. The experiments show that the proposed approach can be used to select data. Geometry-based metrics have the advantage of fast computing. However, geometry-based metrics may lose their effectiveness in measuring the divergence in high dimensional space [121]. As dimensionality grows, the volume of the space increases, and the data becomes sparse. In high dimensional space, the difference between the maximum and the minimum distance from a sample to the centroid or other reference points tends to be zero [122, 123].

Domain discrimination-based metrics look at datasets from the perspective of classifiers and train classifiers to extract the high-dimensional feature space information of the two distributions in a representation layer. Domain discrimination-based metrics train a classifier to discriminate the samples generated from the source domain and the target domain and use the classification error-related value to characterize the divergence. Bousmalis *et al.* [107] introduce proxy $\mathcal{A}$-distance (PAD) into DSN to extract both domain-shared and domain-private features simultaneously. The experiments on the unsupervised domain adaptation scenarios demonstrate the effectiveness of their method. Kim *et al.* [124] use PAD to check if the proposed model successfully extracts the domain-invariant features and generalizes well on the target domain.

Mutual information-based metrics look at datasets from an information theory perspective and capture the probability information between two distributions by measuring the amount of information required to convert one distribution to the other. Asch *et al.* [125] leverage KL divergence and Rényi divergence to compare different domains and predict in NLP applications. Remus *et al.* [126] use Jensen-Shannon (JS) divergence to measure

the divergence between source and target domains, and their results are compared to the state-of-the-art domain adaptation approaches. Duh *et al.* [127] use cross entropy (CE) to measure the similarity between two probability distributions and select similar sentences in machine translation.

Higher-order moment-based metrics capture the moment information between two distributions. Wang *et al.* [128] use MMD to reduce the domain discrepancy of feature representations in the named entity recognition (NER) and show that their approach can outperform the best baseline in most tasks. Zellinger *et al.* [129] introduce central moment discrepancy (CMD) into domain-invariant representations for domain adaptation. They test their scheme on Office and Amazon reviews datasets and prove CMD can achieve high accuracy on most domain adaptation tasks.

Inspired by the existing work, we investigate and choose different divergence measurement metrics to measure the data distribution divergence between different domains. With these metrics, we want to identify the potential relation between attack detection accuracy drop in the smart grid and distribution divergence and approximate the relation through regression models. Specifically, the research first selects three commonly used metrics [112, 130, 128] and uses each metric in isolation to explore the relation between accuracy drop and divergence, predict accuracy drop, and determine when to apply TL. Moreover, considering combining different metrics to extract complementary distribution divergence information, we systematically analyze the divergence metrics published in the literature, compare the information we can obtain from different metrics, and propose an ensemble method to combine them to further improve prediction performance.

Chapter 3

Overview of Framework

3.1 Problem Overview

General ML-based attack detection techniques assume that the training and testing data have the same feature space and are from similar independent distributions [1]. However, this assumption may not hold in CPS scenarios because many CPSs operate in open environments, and the data constantly changes. TL is proposed to address this problem by transferring learned knowledge from a labelled source domain to a related target domain. It has been extensively adopted and witnessed remarkable advances in image and video applications [79]. Lately, TL methods are applied to anomaly detection in Internet [131] and cloud [83] applications, which shed new light on introducing TL to empower intrusion detection in highly dynamic cyber-physical power systems [1, 35].

While various TL models have been proposed to transfer the knowledge learned from one domain to another unseen domain to enhance the attack detection performance, they

have not considered when TL should be applied to retain the attack detection performance. Meanwhile, there is another question to consider during TL: how to effectively extract the internal spatial and temporal features of CPS data to improve detection performance? This is because the spatial-temporal features have been proven to help discriminate attacks from normal data [132]. For instance, on the spatial side, the smart grid can be regarded as an image. To launch FDI attacks on a particular bus, measurements of several specific buses need to be manipulated simultaneously according to the physical topology [133]. Thus, exploiting these spatial correlations of measurement data is crucial for intrusion detection systems (IDS). Moreover, the temporal feature can be extracted from the measurement flow over a continuous period to enhance the detection of well-constructed attacks, such as FDI [28].

To tackle these two challenges, we propose a two-step attack detection framework based on transferability analysis and TL. The overview of the proposed framework is shown in Figure 8. The first step is leveraging transferability analysis to identify the tasks that require TL. Then the framework will apply effective TL techniques to reduce distribution divergence between two domains and improve attack detection performance.

3.2 Divergence-based Transferability Analysis

In transferability analysis, we want to analyze whether a trained model will degrade significantly and require TL. Studies have indicated that TL performance is related to the similarity between the source and target domains [134], and the effectiveness of TL may remain high in a certain range of distribution divergence, depending on how critical the

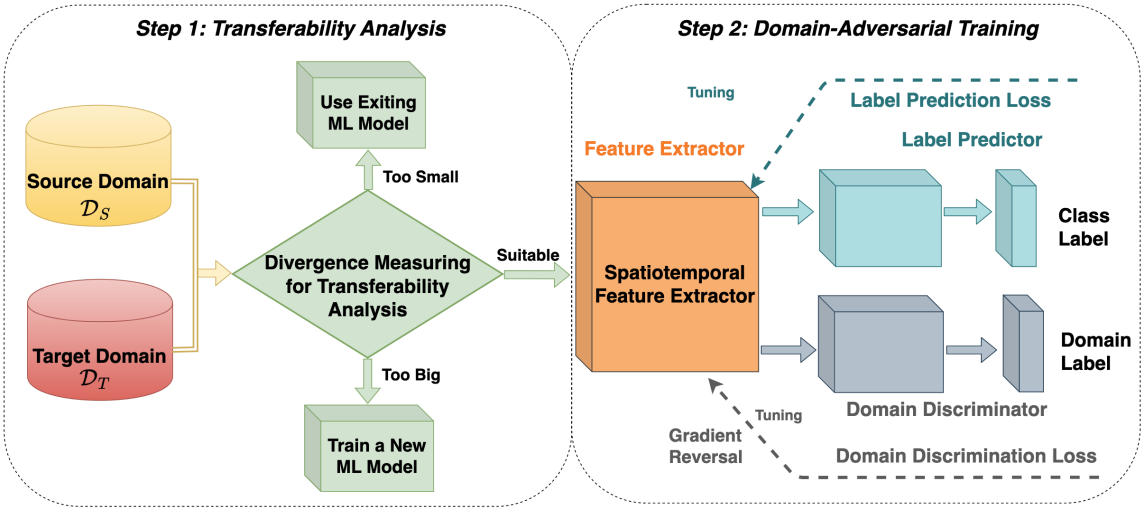


Figure 8: Overview of the proposed transferability analysis and domain-adversarial training (TADA) framework.

application scenarios are.

We can use Figure 9 as an illustrative example of the possible relation between the effectiveness of TL and distribution divergence:

1. If the divergence between the source domain and target domain is within a small range, the ML model trained on the source domain can generally retain a good performance when applied to the target domain, so TL is unnecessary as the performance boost would be trivial, while the adaptation can be costly.

2. If the divergence is too large, even a TL model could suffer a severe accuracy drop on the target domain as the case is beyond transferable [134]. In this case, one would better train a new model from scratch instead of applying TL.

3. If the divergence is somewhere in between, it may be significant enough to degrade the performance of a trained ML model but not beyond what a TL model can handle.

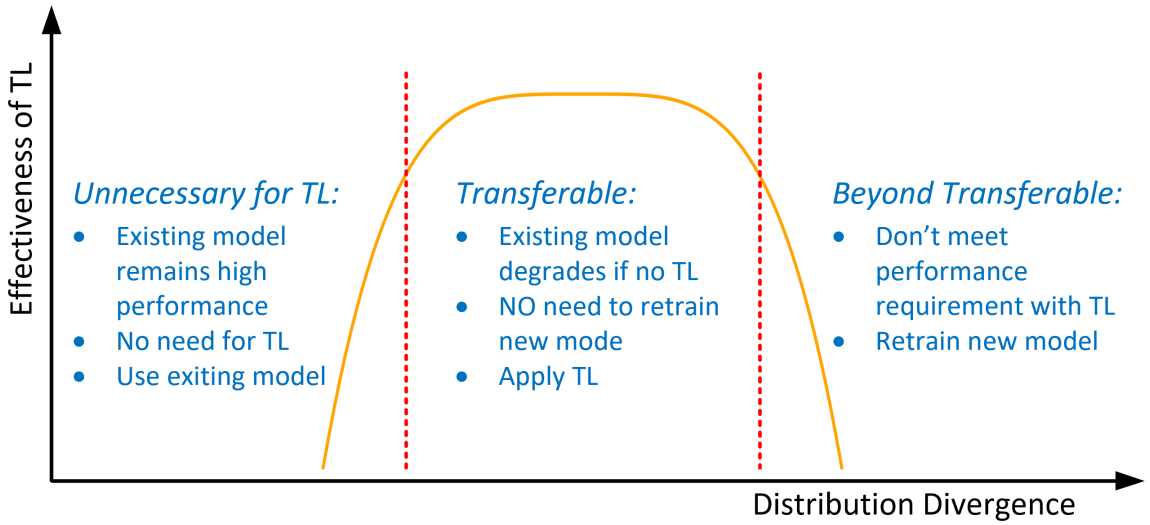


Figure 9: The relation between divergence and effectiveness of TL.

This will be the sweet spot where we can leverage TL to retain a good performance against the divergence by adapting - instead of re-applying or re-creating an ML model.

In this research, we will mainly focus on the transferability between the first and the third situations, where one needs to decide if the divergence makes it necessary to transfer an existing ML model with effective TL methods. Since the model's performance drop is related to the distribution divergence between two domains [67, 100, 112], we want to measure the distribution divergence, and then use the divergence to predict the performance drop and determine whether to apply TL.

3.3 Divergence-based TL

To achieve high detection performance during TL in CPS, effective feature extraction techniques need to be adopted to extract the internal spatial and temporal features of CPS data. Various TL approaches, such as fine-tuning [135], knowledge distillation [136], and one-short learning [137], have been proposed to transfer spatial-temporal features to enhance the robustness of IDS. Xu *et al.* [135] introduce fine-tuning into a CNN-based network detector to enhance the detection ability against new intrusions. Tariq *et al.* [137] propose a CANTransfer to detect intrusions in multivariate time series data. While the aforementioned TL schemes can transfer spatial-temporal features learned from one domain to a different but similar domain, they do not consider reducing the data distribution divergence that leads to performance drop and thus can not solve the problem in this research.

Domain-adversarial (DA) training is a subcategory of TL techniques that reduces domain divergence and improves model generalization by extracting domain-invariant features [67]. The domain-adversarial neural networks (DANN) [72] proposed by Ganin *et al.* is one of the most studied and promising methods. Zhang *et al.* [1, 35] further extend DANN with customized classifiers and propose a semi-supervised DA training model. The extensive experiments show that DANN is capable of decreasing distribution divergence in dynamic time-varying power systems. In this research, we follow their method and adopt DANN as our TL approach.

As shown in Step 2 of Figure 8, there are three essential networks in DANN: feature extractor, label predictor, and domain discriminator. In DANN, the goal of the domain

discriminator is to minimize the domain discrimination loss, while the goal of the feature extractor is to maximize the domain discrimination loss to extract domain-invariant features. To satisfy these two opposed objectives simultaneously during the training process, a gradient reversal layer (GRL) is added between the feature extractor and the domain discriminator. With the GRL, after the gradient of the domain discrimination loss is back-propagated away from the domain discriminator, the gradient is negated and then continues to be back-propagated to the feature extractor. In this way, the DANN enables the feature extractor to extract the domain-invariant features by maximizing the domain classification loss. The model trained on the source domain can also generalize well to the target domain. Meanwhile, during TL, we also want to adopt advanced and effective feature extraction techniques to extract CPS data's internal spatial and temporal features to enhance detection performance.

Based on the remaining gaps and aforementioned analysis, the research proposes a two-step attack detection framework based on transferability analysis and spatial-temporal DA training, as shown in Figure 8. The framework first determines whether a model will suffer a significant accuracy drop on the target domain and thus require TL, then uses DA training to extract the domain-invariant spatial-temporal feature to improve the attack detection performance against distribution divergence. The details of each step are illustrated in Chapter 4 and Chapter 5, respectively.

Chapter 4

Divergence-based Transferability Analysis

The goal of transferability analysis is to analyze whether a trained model will degrade significantly and thus require TL. It has been proven that the performance degradation of a trained mode is related to data distribution divergence between two domains [112]. So the data distribution divergence measurement plays a crucial role in predicting performance drop and triggering TL. Therefore, the thesis first selects three commonly used metrics [112, 130, 128] and uses each of them in isolation to explore the relation between accuracy drop and divergence. The approximated relation is then leveraged to predict the accuracy drop on the unlabelled target domain based on measured distribution divergence, and determine when to apply TL. Moreover, considering that different metrics can extract complementary distribution divergence information, we systematically analyze the divergence metrics published in the literature, compare the information we can obtain from

different metrics, and propose an ensemble method to combine them to further improve prediction performance.

4.1 Single Metric Transferability Analysis

4.1.1 Problem Formulation

CPS operating in open environments may have significant data distribution divergence, which may lead to accuracy degradation for a model trained on the source domain and tested on the target domain. Given a source domain $\mathcal{D}_S$ and a target domain $\mathcal{D}_T$, the distribution divergence may be caused by a variety of reasons. It can be due to covariate divergence, where only the feature distribution changes, i.e., $P_{D_S}(X) \neq P_{D_T}(X)$, but the conditional distribution remains the same, i.e., $P_{D_S}(Y|X) = P_{D_T}(Y|X)$. It may be caused by concept divergence, where $P_{D_S}(X) = P_{D_T}(X)$ and $P_{D_S}(Y|X) \neq P_{D_T}(Y|X)$, or label divergence, where $P_{D_S}(Y) \neq P_{D_T}(Y)$ and $P_{D_S}(X|Y) = P_{D_T}(X|Y)$, or a combination of the above divergence.

This research focuses on the covariate divergence, which often occurs in CPS intrusion detection scenarios because the system variations and attack variations will influence the normal and attack data distribution. In the power system, the system variations may be caused by the different load demands, normal operations, or topology changes. The attack variations could arise when the same scheme is launched again at different periods or locations in the grid. This research considers the binary intrusion detection problem in the smart grid, which intends to classify the multivariate time series measurement data as attack events or normal operations. We focus on the scenarios where two consecutive

attacks target at different times and/or different locations.

We are interested in the attack detection accuracy drop that requires the TL. We assume that the source domain consists of labelled normal data and attack data, where $\mathcal{D}_S = \{(x_{S_1}, y_{S_1}), ..., (x_{S_{n_1}}, y_{S_{n_1}})\}$. If we had a fully labelled target domain $\mathcal{D}_\mathcal{T}$, the accuracy drop could be measured by empirical data:

$$\Delta Pr = Pr_{D_S} - Pr_{D_T}, \tag{1}$$

where Pr_{D_S} is the accuracy of a model trained on a source domain $\mathcal{D}_\mathcal{S}$, and Pr_{D_T} is the accuracy when this model is applied to the target domain $\mathcal{D}_\mathcal{T}$. However, the detection system deployed in the smart grid detects attacks online, and the new generated target domain is unlabelled, i.e., $\mathcal{D}_T = \{(x_{T_1}), ..., (x_{T_{n_2}})\}$. So, the accuracy drop can not be calculated via Eq. (1) with the unlabelled target domain.

To solve the problem, we propose to employ dataset pairs from the labelled source domain to explore the relationship between divergence and accuracy drop, then use the relation to predict the accuracy drop of the unlabelled target domain. If we have the divergence-accuracy drop relation, the accuracy drop of the target domain can be predicted by:

$$\Delta Pr' = A(d), \tag{2}$$

where $\Delta Pr'$ is the predicted accuracy drop, A is the relation between accuracy drop and divergence, and d is the distribution divergence of the source domain and the target domain.

The proposed transferability analysis aims to predict accuracy drop and identify the

unlabelled target datasets where a trained model will degrade significantly and call for TL. The challenges are how to measure the data distribution divergence in intrusion detection and how to approximate the relation between accuracy drop and divergence, which are tackled by the proposed analysis in Figure 10. We measure the accuracy drop and divergence between each pair of datasets from source domains, then train regression models to approximate the divergence-accuracy drop relation. With the approximated relation model, we can calculate the divergence between the target domain and source domain, and leverage the relation model to predict the accuracy drop on the unlabelled target domain. If the predicted accuracy drop is in a suitable range, trigger the TL.

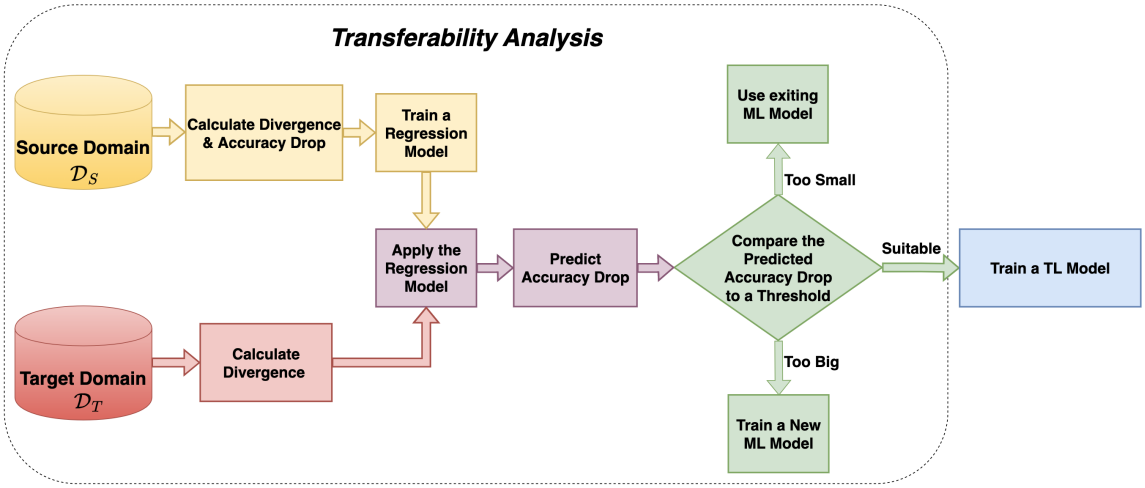


Figure 10: The proposed divergence-based transferability analysis for the smart grid intrusion detection.

4.1.2 Methodology

Data Distribution Divergence Metrics

The distribution divergence information can be characterized with different metrics. First, we first select three widely used divergence metrics to evaluate distribution divergence:

Proxy $\mathcal{A}$-Distance (PAD)

PAD is a domain discrimination-based metric proposed by Ben-David *et al.* [112]. Ben-David *et al.* have proven that the error of a trained model on a target domain is bounded by its error on the source domain and the $\mathcal{H}$-divergence between the source domain and the target domain. $\mathcal{H}$-divergence depends on the capability of a trained classifier to discriminate between samples generated from source and target domains. To calculate the $\mathcal{H}$-divergence with finite data sampled from the source and target domains, Ben-Davidr *et al.* propose the PAD to approximate $\mathcal{H}$-divergence.

To compute PAD, source domain data and target domain data are mixed, and samples from source and target domains are labelled as 0 and 1, respectively. Then a classifier G_d is trained on the mixed dataset to distinguish between samples from source and target domains. Finally, the classifier is tested on the held-out test dataset. The PAD is defined as:

$$\epsilon\left(G_d\right) = \frac{1}{|D|} \sum_{x_i \in D'_s, D'_t} \left| G\left(x_i\right) - I\left(x_i\right) \right|, \tag{3}$$

$$PAD = 2(1 - 2\epsilon\left(G_d\right)), \tag{4}$$

where G_d is the trained classifier. $\epsilon\left(G_d\right)$ is the classifier's error on the held-out dataset

D'_s and D'_t. I is the true domain label. In the experiments of this research, following the approach of Ben-David *et al.*[112], we train a linear SVM as our classifier.

Kullback–Leibler (KL) Divergence

KL divergence [138] is a mutual information-based metric and has shown effectiveness in predicting performance in sentiment analysis [139]. KL divergence measures the relative entropy between two probability density functions $p(x)$ and $q(x)$:

$$D_{KL}(P||Q) = \int p(x) log \frac{p(x)}{q(x)} dx. \tag{5}$$

We adopt the work of John *et al.* [140] and consider the two datasets following Gaussian mixture models (GMM). The marginal densities of $x \in R^d$ under p and q are:

$$\begin{aligned} p(x) &= \sum_a \pi_a \mathcal{N}(x; \mu_a; \Sigma_a), \\ q(x) &= \sum_b \pi_b \mathcal{N}(x; \mu_b; \Sigma_b). \end{aligned} \tag{6}$$

To estimate $D(P||Q)$, we could conduct a Monte Carlo simulation. Using n i.i.d. samples $\{x_i\}_n^{i=1}$, we have:

$$D_{MC}(P||Q) = \frac{1}{n} \sum_{i=1}^{n} log \frac{p(x_i)}{q(x_i)} \rightarrow D(P||Q). \tag{7}$$

The variance of the estimation error could be decreased when $n \rightarrow \infty$.

Maximum Mean Discrepancy (MMD)

MMD is a higher-order moment-based metric and has been widely used in TL. MMD estimates divergence between two distributions based on the Reproducing Kernel Hilbert

Space (RKHS)[141]. Given two datasets $X = \{x_1, x_2, ..., x_{n_1}\}$ and $Y = \{y_1, y_2, ..., y_{n_2}\}$ that come from two distribution P and Q, the empirical estimation of the distance is defined by:

$$D_{MMD}(X||Y) = \left\| \frac{1}{n_1} \sum_{i=1}^{n_1} \varphi(x_i) - \frac{1}{n_2} \sum_{j=1}^{n_2} \varphi(y_j) \right\|_H . \tag{8}$$

where $\varphi(x)$: $\mathcal{X} \to \mathcal{H}$, is a kernel-based function mapping samples to a feature representation space in RKHS.

The feature representation varies with the different choices of kernels. In this research, the radial basis function (RBF) kernel is adopted since the RBF kernel can take advantage of the Taylor expansion of the Gaussian function to match all the moments of two distributions [107].

Divergence-based Performance Drop Prediction

After measuring the accuracy drop and data distribution divergence by the selected metrics, the potential relation between the distribution divergence and the detector accuracy drop are approximated through regression models, including linear regression and neural network regression.

Linear Regression

A strong positive relation between detection accuracy drop and divergence can be observed in Figure 14: Pearson correlation coefficient ρ [142] is above 0.83 in all cases. According to Haldun and Dancey *et al.* [143, 144], $0.8 < \rho <= 1$ shows a strong relation between two variables. Based on this observation, we first introduce a linear regression model.

$$\Delta Pr = w_1 d + w_0, \tag{9}$$

where w_0 and w_1 are parameters of the linear regression model, d is the distance between $\mathcal{D}_S$ to $\mathcal{D}_T$, and ΔPr is the accuracy drop of a model that is trained on $\mathcal{D}_S$ and applied to $\mathcal{D}_T$.

Neural Network Regression

Considering that the divergence-accuracy relation may not be linear, we also leverage a neural network (NN) regression model. The neural network regression model can learn a non-linear and complicated relation between accuracy drop and divergence.

$$\Delta Pr = f_{neural}(d), \tag{10}$$

where f_{neural} is a fully connected neural network, we adopt the same configuration as [116]. The input d is the distribution divergence of two domains. The output ΔPr is the accuracy drop.

With the regression models, we can measure the divergence between the source domain and the unlabelled target domain, and predict the accuracy drop according to the divergence. If the divergence exceeds the accuracy drop threshold Π, we will trigger TL for the target domain. The entire proposed transferability analysis process is summarized in Algorithm 1.

4.1.3 Experiments Setup

This subsection will introduce our experimental setup to validate the single metric transferability analysis for intrusion detection.

Algorithm 1 Transferability Analysis

Input: The set $\mathcal{S}$ of labelled source dataset $\mathcal{D}_S$; The set $\mathcal{T}$ of unlabelled target dataset $\mathcal{D}_T$; Accuracy drop upper bound Π and lower bound π

Output: TL decision

for source dataset pair $\mathcal{D}_{S_m}$ and $\mathcal{D}_{S_n}$ in $\mathcal{S}$ **do**
  # Measure divergence
  $d \leftarrow D(D_{S_m}||D_{S_n})$
  # Measure accuracy drop
  Train a classifier on $\mathcal{D}_{S_m}$, calculate accuracy $Pr_{D_{S_m}}$
  Apply classifier on $\mathcal{D}_{S_n}$, calculate accuracy $Pr_{D_{S_n}}$
  $\Delta Pr \leftarrow Pr_{D_{S_m}} - Pr_{D_{S_n}}$
end for
# Train a regression model
$\Delta Pr = A(d)$
# Predict accuracy drop for target domain
for dataset pair $\mathcal{D}_S$ and $\mathcal{D}_T$ in $\mathcal{S}$ and $\mathcal{T}$ **do**
  $d' \leftarrow D(D_S||D_T)$
  $\Delta Pr' = A(d')$
  # Make TL decision
  if $\Delta Pr' <= \pi$ **then**
    Use exiting ML model
  else if $\pi < \Delta Pr' < \Pi$ **then**
    Train a TL model
  else
    Train a new ML model
  end if
end for

Data

Normal Data

To establish experiments based on realistic scenarios, we obtain public load demand from ISO New England [6] from August 24th to 30th, 2019, as shown in Figure 11. In ISO New England, the demand was reported every 5 minutes. To increase the sampling rate and maintain the trend of the demand curve, the demand data is interpolated with a 1-second interval by the Spline method in MATLAB.

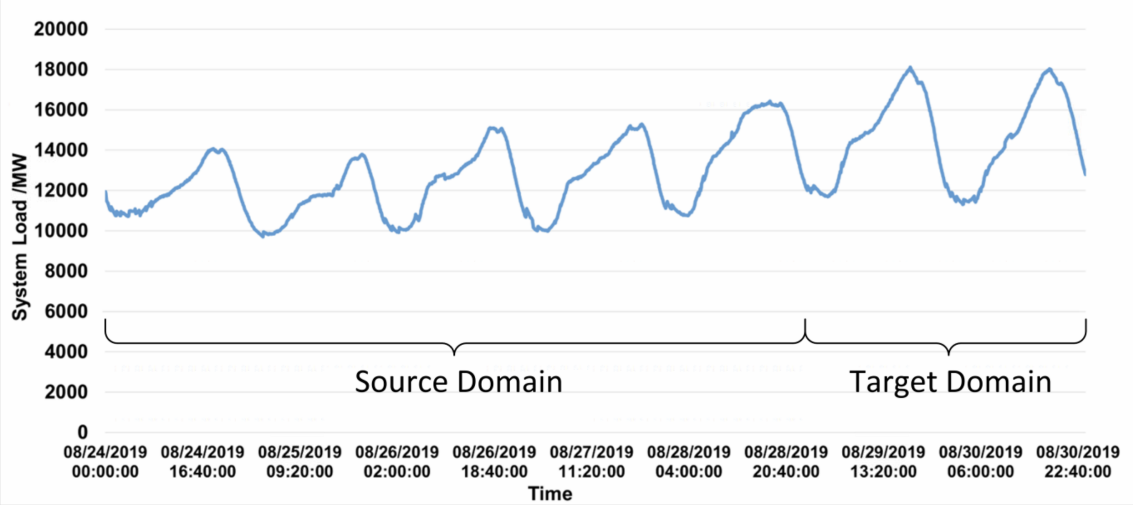


Figure 11: One-week load demand of ISO New England [6].

The IEEE 30-bus system [7] is selected as the simulation scenario, and MATLAB toolbox MATPOWER is leveraged to generate and synthesize the above load demand. As illustrated in Figure 12, the system consists of 30 buses and 41 branches with a total load demand of 189.2 MW. We first assume that the default operating point in the 30-bus system is at its peak (100%) and match the total load demand to the peak load of the data we obtained from ISO New England. Then we assume that the total demand of the IEEE

30-bus system follows the same changes as that of the ISO New England grid (in terms of percentage w.r.t. the peak load). For example, if the total demand of the ISO New England drops from 100% at the peak to 80% after 2 hours, the total demand for the IEEE 30-bus system will also decrease to 80% of its own peak after 2 hours. This matching will allow us to apply the same aggregated load profile of the ISO New England to that of the IEEE 30-bus system.

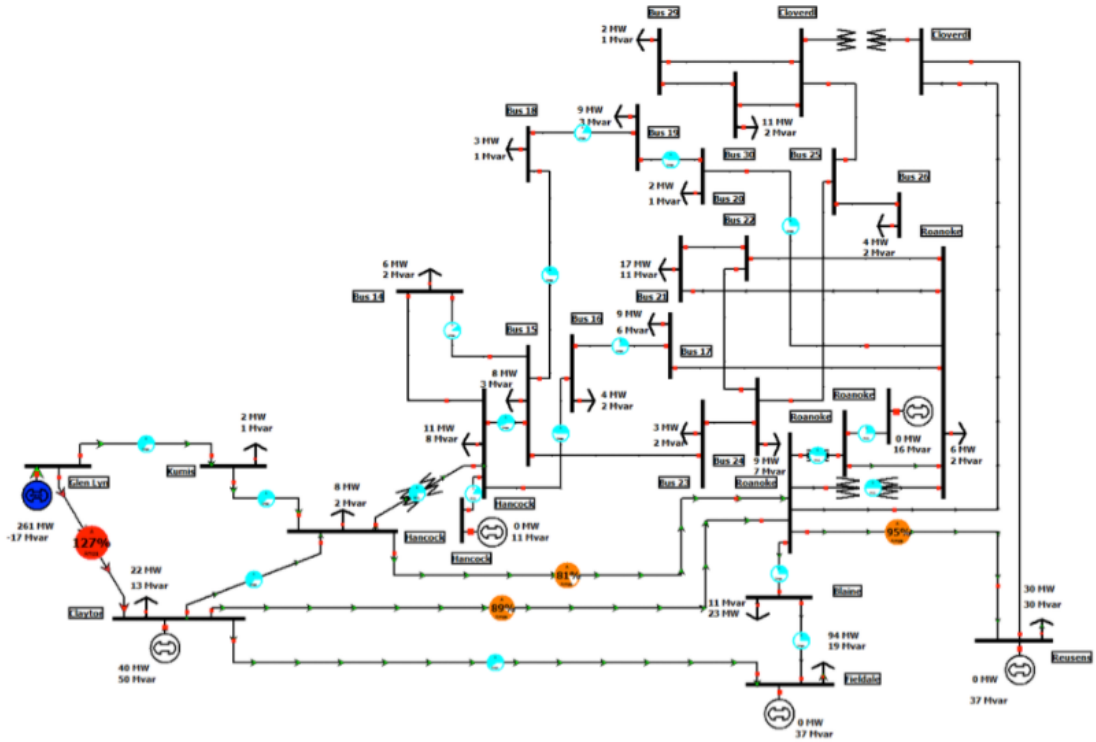


Figure 12: The IEEE 30-bus system by the Illinois Center for a Smarter Electric Grid (ICSEG) [7].

Meanwhile, we follow [145] and introduce variations into a load for each node over time. Based on [145], we assume that at a given period, if the load of the entire grid

is changed by $x\%$, the corresponding individual load change across 30 buses follows a normal distribution with a mean of $x\%$ and a variance of $y\%$. For example, from t_k to t_{k+1}, if the total load of the grid is increased by 3.0%, the load of each node may increase similarly but with potential random variations, such as 3.2% or 2.6%, and the average will be 3.0%. In our experiments, x is the change obtained from the ISO New England load profile, $y = x/100$. 142 measurements over a 1-second interval are calculated and collected through the DC optimal power flow (DC-OPF) solver in MATPOWER as normal data.

Attack Data

Distinct attack models have been proposed and developed to analyze and enhance the security of the smart grid in the past two decades [23]. The FDI, first proposed by Liu *et al.* [133], is one of the most widely studied threat models and is therefore chosen as the attack model in this research. The attack scheme of FDI is shown in Figure 13. To elaborate on how FDI introduces treats on power systems, we first introduce conventional bad data detector (BDD) in the power system state estimators (PSSE). In the DC state estimation, the relation between state variables and observed measurements can be formulated as follows [133]:

$$\mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{e}, \tag{11}$$

where $\mathbf{z} = z_1, z_2, ..., z_n$ represent the physical measurements, $\mathbf{x} = x_1, x_2, ..., x_m$ are the state variables, $\mathbf{H}$ is an $n \times m$ Jacobian matrix of power grid topology, and $\mathbf{e} = e_1, e_2, ..., e_n$ are measurement errors often models by the white Gaussian noise. Based on the observed measurements $\mathbf{z}$ and $\mathbf{H}$, the estimated states $\hat{\mathbf{x}}$ can be obtained with the following weighted

least square (WLS) solution,

$$\hat{\mathbf{x}} = (\mathbf{H}^{\mathsf{T}}\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^{\mathsf{T}}\mathbf{W}\mathbf{z}, \tag{12}$$

where $\mathbf{W}$ is a diagonal matrix.

$$\mathbf{W} = \begin{bmatrix} \sigma_1^{-1} & & & & \\ & \sigma_2^{-1} & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & \sigma_n^{-1} \end{bmatrix} \tag{13}$$

where each element in $\mathbf{W}$ is the reciprocal of the variance of meter error.

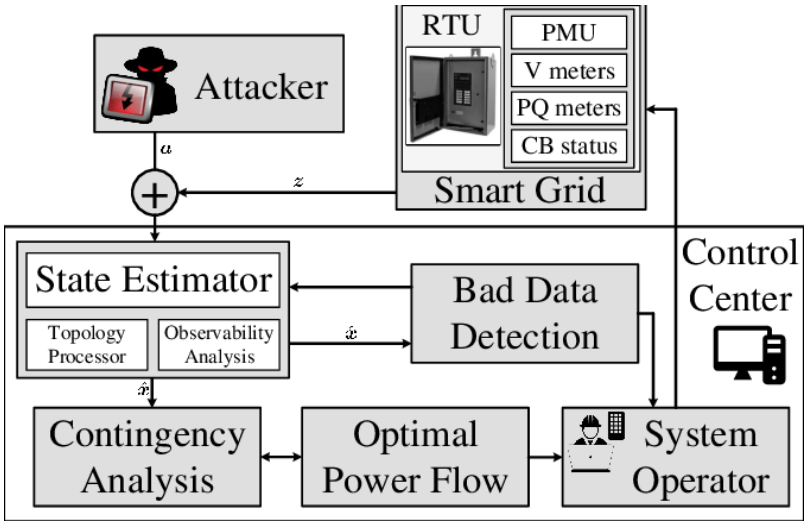


Figure 13: Attack scheme of FDI [8].

With the estimated states $\hat{\mathbf{x}}$, the traditional BDD first measures residual between observed measurements $\mathbf{z}$ and estimated measurements $\mathbf{H}\hat{\mathbf{x}}$,

$$\mathbf{r} = \mathbf{z} - \mathbf{H}\hat{\mathbf{x}}. \tag{14}$$

where $\hat{\mathbf{x}} = \hat{x}_1, \hat{x}_2, ..., \hat{x}_m$ is the estimated states, and $\mathbf{r}$ is the residual. Then, BDD calculates the $L_2 - norm$ of residual and adopts the statistical residual tests to detect the presence of bad data via comparing the residual with a threshold τ. If $\|r\| > \tau$ indicates that there is an bad data.

To bypass the residual-based BDD and stealthily compromises measurements from electricity grid sensors in a coordinated fashion, the FDI attack is designed to exploit a mathematical vulnerability in the residual-based BDD. The FDI attack is assumed to have the knowledge of topology matrix $\mathbf{H}$, so the attacker can choose to generate attack vector $\mathbf{a}$ as follows,

$$\mathbf{a} = \mathbf{H}\mathbf{c}, \tag{15}$$

where where $\mathbf{c} \sim N(0, \sigma_c^2)$ is the false state error injected into the system. Then attacker injects attack vector $\mathbf{a}$ into the normal measurements $\mathbf{z}$ and generate the manipulated measurements $\mathbf{z_a}$ by,

$$\mathbf{z_a} = \mathbf{z} + \mathbf{a}. \tag{16}$$

In this case, the states $\hat{\mathbf{x}}_\mathbf{a}$ estimated from the manipulated measurements $\mathbf{z_a}$ can be

computed by,

$$\begin{aligned}\hat{\mathbf{x}}_\mathbf{a} &= (\mathbf{H}^\mathsf{T}\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^\mathsf{T}\mathbf{W}\mathbf{z}_\mathbf{a} \\ &= (\mathbf{H}^\mathsf{T}\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^\mathsf{T}\mathbf{W}(\mathbf{z}+\mathbf{a}) \\ &= \hat{\mathbf{x}} + (\mathbf{H}^\mathsf{T}\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^\mathsf{T}\mathbf{W}\mathbf{a}\end{aligned} \tag{17}$$

where $\mathbf{W}$ is a diagonal matrix.

Since $\mathbf{z}_\mathbf{a} = \mathbf{z} + \mathbf{a}$, the new residual will be [133]:

$$\begin{aligned}\mathbf{r}_\mathbf{a} &= \mathbf{z}_\mathbf{a} - \mathbf{H}\hat{\mathbf{x}}_\mathbf{a} \\ &= \mathbf{z} + \mathbf{a} - \mathbf{H}(\hat{\mathbf{x}} + (\mathbf{H}^\mathsf{T}\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^\mathsf{T}\mathbf{W}\mathbf{a}) \\ &= \mathbf{z} - \mathbf{H}\hat{\mathbf{x}} + (\mathbf{a} - \mathbf{H}(\mathbf{H}^\mathsf{T}\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^\mathsf{T}\mathbf{W}\mathbf{a}) \\ &= \mathbf{z} - \mathbf{H}\hat{\mathbf{x}} + (\mathbf{H}\mathbf{c} - \mathbf{H}(\mathbf{H}^\mathsf{T}\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^\mathsf{T}\mathbf{W}\mathbf{H}\mathbf{c}) \\ &= \mathbf{z} - \mathbf{H}\hat{\mathbf{x}} + (\mathbf{H}\mathbf{c} - \mathbf{H}\mathbf{c}) \\ &= \mathbf{z} - \mathbf{H}\hat{\mathbf{x}}.\end{aligned} \tag{18}$$

Compare the residual without FDI attack in Eq. (14) and the residual with FDI attack in Eq. (18), we can find that the residual remains the same, allowing the FDI attack to bypass the residual-based BDD.

A successful FDI attack will evade detection and pose a severe threat to PSSE in the SCADA systems, with the possibility of inflicting severe impacts like power outages, physical damages, and monetary losses [14]. The FDI attack has successfully attracted the attention of many researchers with more than 2,000 citations. Therefore, in this research, we choose the FDI attack as the attack model. This research will use the manipulated measurements $\mathbf{z}_\mathbf{a}$ as the attack data. The false state $\mathbf{c}$ is set with a mean of zero and a variance

of $\sigma_c^2 = 0.1$.

Case Setup

Three scenarios are considered to validate the effectiveness of the proposed approach, including temporal, spatial, and spatial-temporal cases. Considering the labelled attack data could be extremely rare in the smart grid compared with the labelled normal data, if there is no attack data available, the power grid operators can review the attack models [23] in the literature and synthesize the most prominent attacks. This can still be helpful in defence planning and operations against the most prominent subset of attacks.

Temporal Scenario

First, we consider a known attack returning at different times. We assume that attackers launch the attack vector at the same locations but across different periods in temporal cases. Since the load demand and its patterns vary significantly throughout the day, we select the 4-hour time window data as the source domain and target domain to best capture the characteristics of data distributions.

As illustrated in Table 1, source domain datasets are generated from the labelled historical normal and attack data from Day 1 to Day 5. Considering the load patterns distinct in different periods of a day, we define 4 cases based on our previous work [35] according to the variation of load demand: the valley, the ascending slope, the peak, and the descending slope. We also use the 4-hour time window but divide each day of Day 6 and Day 7 into six intervals as the target domain datasets for testing.

Spatial Scenario

For spatial scenarios, we consider attacks returning at different locations. We assume

Table 1: Setup of cases in the temporal scenario.

Cases	Source Domain from Day 1 to 5		Target Domain on Day 6 and 7
	Load Patterns	Hours	
1	Valley	2–5	
2	Ascending	11–14	2 hours of normal data followed
3	Peak	17–20	by 2 hour of attack data
4	Descending	21–24	

the load demand will be similar in source and target domains. We select the same 4-hour time window of the different days for source and target data while choosing different attack locations for the target data.

Since we use the IEEE 30-bus system, there are a total of 30 potential buses to be attacked. However, some buses carry zero loads and are non-attackable. We follow the reference [146] and notice that attackers will not choose to attack 15 specific buses. Hence for the IEEE 30-bus system, we have 15 attackable source domain datasets and target domain datasets. We also assume that the attackers only inject one bus when launching the attack. By conducting training and testing on 15×15 pairs experiment, the non-transfer methods perform worst when target buses are 14, 16, 19, thus we select 15 buses as the source domain datasets separately and Buses 14, 16, 19 as target domain datasets.

spatial-temporal Scenario

For spatial-temporal scenarios, we consider attacks that happen at different times and locations. We assume that the time and locations of attack in the target domain both vary from the source domain. To this end, we select 4 hours ("valley") as the source load demand pattern and another 4 hours ("Peak") as the target load demand pattern. And we inject different buses for source domain datasets and target domain datasets.

Classifier Architecture

We use the DANN [72] as our benchmark TL model, which aims to learn domain-invariant features by maximizing the domain discriminator loss and minimizing the label predictor loss. Yongxuan *et al.* [1] propose a DANN-based TL approach in the smart grid and show their approach is sufficiently powerful to perform well on intrusion detection in the smart grid. For the basic classification model used to calculate the classification accuracy drop in the transferability analysis, we extract the feature extractor and the label predictor from DANN and combine them into an MLP [147], which contains 5 layers and 592 neurons in total.

We train the MLP on the source domain and test it on the target domain to acquire the classification accuracy drop. A threshold Π is set to indicate whether the data distribution divergence could have a significantly negative effect on the trained ML model. Considering FDI is a severe threat, we use 10% of accuracy drop as the threshold in triggering TL in experiments.

To evaluate TL performance after identifying the tasks, we compare the detection accuracy of DANN and a non-transfer ML method. Since MLP has demonstrated superior accuracy and computation efficiency in intrusion detection [33], we choose MLP as the non-transfer ML method and follow the same configuration as that in transferability analysis. All classifiers are implemented in Scikit-learn [148] and Keras with manually optimized parameters releasable upon request.

4.1.4 Results and Discussion

Figure 14 shows the relation between actual detection accuracy drop (y-axis) and the divergence (x-axis) measured by selected metrics in temporal, spatial, and spatial-temporal scenarios. In Figure 14, each dot corresponds to a pair of datasets. We also plot the linear regression line and neural network regression line in green and red.

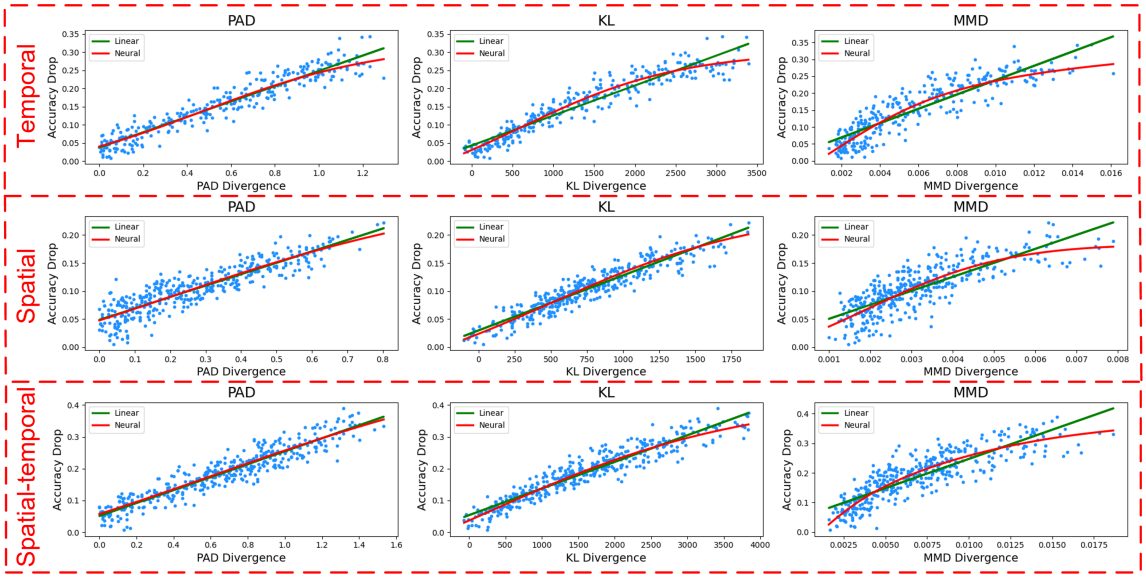


Figure 14: Relation between actual detection accuracy drop and divergence measured by selected metrics in temporal, spatial, and spatial-temporal experiments.

Comparison between Three Scenarios

We can find a strong positive relation between accuracy drop and distribution divergence with a high Pearson correlation coefficient: ρ is above 0.83 in all experiments. This observation also indicates that it is feasible to predict the accuracy drop by distribution

divergence. Among the three scenarios, spatial cases have the lowest divergence and accuracy drop. This is because the normal data of source and target domains are from the same load demand pattern and share similar distributions. Meanwhile, spatial-temporal cases have the biggest divergence and accuracy drop, since the source and target domains vary in both temporal and spatial variables.

Table 2: Error (%) of accuracy drop prediction in the target domain.

Scenarios	**Metrics**	**Linear Regression**		**Neural Network Regression**	
		RMSE	**MaxAE**	**RMSE**	**MaxAE**
Temporal	**PAD**	2.38	8.62	2.34	7.72
	KL	2.52	7.88	2.22	7.43
	MMD	3.65	10.94	3.26	9.14
Spatial	**PAD**	1.88	6.35	1.87	6.32
	KL	1.43	4.48	1.41	4.25
	MMD	2.45	7.51	2.36	7.79
spatial-temporal	**PAD**	2.77	8.28	2.75	8.74
	KL	2.60	7.46	2.46	7.53
	MMD	4.20	12.74	3.84	12.72

Comparison of Divergence Metrics and Regression Models

Table 2 shows the root mean squared error (RMSE) and maximum absolute error (MaxAE) of different metrics with the two regression models. To show the accuracy drop prediction ability of the proposed transferability analysis, following Elsahar *et al.* [116], we first make a comparison in predicting classification accuracy between the selected metrics and baseline. The baseline divides the divergence of the historical source dataset pairs

into small intervals, and calculates the mean accuracy drop in each small divergence interval as the expected accuracy drop. For the target dataset, if the measured divergence is located in a specific interval, the baseline takes the expected accuracy drop of that interval as the predicted accuracy drop of the target domain. The baseline RMSE, are 7.09%, 4.18%, and 8.19% in temporal, spatial, and spatial-temporal scenarios, respectively. The baseline MaxAE are 18.91%, 13.87%, 20.28% in each scenario.

Overall, all our selected metrics improve significantly over the baseline in both RMSE and MaxAE. For instance, in temporal cases, PAD and KL with either linear regression or neural network regression both decrease RMSE to below 2.38% and MaxAE to under 8.62%. MMD performs slightly worse than the first two metrics but still achieves high performance compared to the baseline. MMD has an RMSE of 3.65% and MaxAE of 10.94% with linear regression, and an RMSE of 3.26% and MaxAE of 9.14% with neural network regression. Among the three metrics, PAD and KL have comparable performance and show robust prediction power in all three scenarios. In addition, PAD and KL are more accurate than MMD in RMSE and MaxAE.

Neural network regression has a slightly smaller RMSE than linear regression in all scenarios, but their general performance is close. Overall, the RMSE of two regression models with all three selected metrics is lower than 4.20% in all cases, implying the predicted accuracy drop is close to the ground accuracy drop. This indicates a strong relation between accuracy drop and divergence, and we can use this relation to predict accuracy drop.

TL Performance

Based on the above observation, we can measure the divergence and leverage the regression relation to predict the accuracy drop of an unlabelled target domain dataset, and determine whether to trigger TL accordingly. We set 10% as the accuracy drop threshold and classify all 6,240 experiments in temporal, spatial, and spatial-temporal scenarios into TL-unnecessary and TL-necessary cases. TL-unnecessary cases refer to experiments whose predicted accuracy drop is lower than 10%. TL-necessary cases refer to experiments whose predicted accuracy drop is greater than 10%. Then, we apply non-TL and TL methods to both cases to evaluate the detection performance. The average detection accuracy of non-TL and TL methods on both TL-unnecessary and TL-necessary cases is illustrated in Table 3. Specifically, the accuracy shown in the row of "TL-unnecessary" is the average of 1704 experiments in temporal, spatial, and spatial-temporal scenarios. The accuracy shown in the row of "TL-necessary" is the average of 4536 experiments in three scenarios.

Table 3: Average Detection Accuracy (%) of non-TL and TL on Different Cases.

Cases	**Non-TL Accuracy**	**TL Accuracy**	**Accuracy Improvement**
TL-unnecessary	89.53	94.65	+5.12
TL-necessary	76.87	91.79	+14.92

In the experiments, the transferability analysis can successfully identify all TL-necessary cases and trigger TL to improve the attack detection performance. In all TL-necessary cases, TL has an average accuracy improvement of 14.92%. Compared with the TL-necessary cases, TL gives less improvement in the TL-unnecessary cases. In these cases,

there are less distribution divergence and accuracy drop, and thus TL gives less accuracy improvement margin.

4.2 Ensemble Metrics Transferability Analysis

4.2.1 Problem Formulation

Based on the literature review in Subsection 2.3.2, we can find that different types of metrics can look at the data distribution from different angles and extract different distribution divergence information [118]. Ruder *et al.* [119] have proved the importance of combining different metrics to capture complementary information in divergence measuring. In the last section, we use every single metric in isolation to explore the relation between accuracy drop and divergence. However, it might be difficult for a single divergence metric to capture multiple data distribution divergence information since each metric may only cover limited aspects of data distribution divergence information.

To tackle this problem, this section further proposes an ensemble method that combines different types of metrics to improve the accuracy drop prediction and justify the need for TL in cyber-security monitoring. Specifically, the research first selects one metric from each divergence metric category, then trains the neural network regression model using all selected metrics to approximate the relationship between divergence and accuracy drop. To compare the ensemble metrics method with the single metric method in predicting accuracy drop, we also train linear regression models for each metric.

4.2.2 Methodology

Distribution Divergence Metrics

We systematically analyze different divergence measurement metrics and classify them into four categories by comparing the information each metric provides [118, 119]. Considering different categories of metrics can capture various and complementary distribution divergence information, we choose one metric from each category that has been shown to have good divergence measurement and performance prediction ability:

Geometry-based Metrics: Geometry-based metrics, such as Euclidean distance and Manhattan distance [118], use the statistical descriptions of data distribution, like mean and standard deviation, to capture the geometry-related information. We choose cosine similarity since it has demonstrated effectiveness in measuring similarity between two domains [149]. The cosine distance (Cos) is defined as $1 - cosine\ similarity$:

$$D_{Cos} = 1 - cos(\vec{m}, \vec{n}) = 1 - \frac{\vec{m} \cdot \vec{n}}{\|m\| \cdot \|n\|}, \tag{19}$$

where $\vec{m}$ and $\vec{n}$ are two statistical vectors used to describe two distributions.

Domain Discrimination-based Metrics: Domain discrimination-based metrics look at datasets from the perspective of classifiers and train classifiers to extract the high-dimensional feature space information of the two distributions in a representation layer. The classifier is trained to discriminate the data domains between the source and target, and the divergence is characterized by the classification error. Among the available metrics, the proxy $\mathcal{A}$-distance (PAD) [116] performs best in measuring divergence and predicting the performance drop in tasks like part of speech tagging [118], and thus is picked

in this paper. The approach to calculate PAD is present in Eq. (4).

Mutual Information-Based Metrics: Mutual information-based metrics look at datasets from an information theory perspective and capture the probability information between two distributions by measuring the amount of the information required to convert one distribution to the other, like Kullback–Leibler (KL) divergence [118] and cross entropy [119]. We choose Jensen-Shannon (JS) divergence as it is a symmetric variance of KL divergence and has been proven to be a reliable indicator for measuring domain similarity in tasks such as sentiment analysis [126].

$$D_{JS}(P||Q) = \frac{1}{2}D_{KL}(P||M) + \frac{1}{2}D_{KL}(Q||M), \tag{20}$$

where $D_{KL}(P||Q) = \int p(x)log\frac{p(x)}{q(x)}dx$, $M = \frac{1}{2}(P+Q)$. $p(x)$ and $q(x)$ are probability density functions of two distributions.

Higher-Order Moment-Based Metrics: Higher-order moment-based metrics, like correlation alignment (CORAL) [118] and CMD [129], capture the moment information between two distributions. MMD is chosen in this work since it has been extensively adopted to measure the domain discrepancy in domain adaptation works [93]. The approach to calculate MMD is present in Eq. (8).

Regression Models

Using the aforementioned metrics, we can measure distribution divergence with labelled historical data in the source domain, then train a neural network regression model

with ensemble metrics:

$$\Delta Acc = f_{ensemble}\left(d_{Cos}, d_{PAD}, d_{JS}, d_{MMD}\right), \tag{21}$$

where $f_{ensemble}$ is a fully connected neural network with all selected metrics as the input.

We also train a linear regression model for every single metric and compare the ensemble metrics method to the single metric method in predicting accuracy drop. The single metric regression model is present in Eq. (9).

The regression models are leveraged to predict accuracy drop based on the measured divergence for the unlabelled target domains. As shown in Figure 10, if the predicted drop is neither too small that TL is unnecessary nor too large that it is beyond transferable, TL will be leveraged to improve the detection accuracy.

4.2.3 Experiments Setup

In this experiment, we extend to longer periods to validate the effectiveness of the proposed transferability analysis and focus on the more challenging spatial-temporal cases. We obtain seven years of real-world load demand of ISO New England [6] from 2015 to 2021, as shown in Figure 15, for normal data simulation. Following the previous experiments, the FDI is chosen as the attack model to generate the attack data. We use the standard IEEE 30-bus system as the simulation system and assume it may be attacked at different times and locations. In terms of attack times, considering the load demand of different seasons distinct, we set up 4 cases from each year's data: winter, spring, summer, and fall, to capture the temporal divergence, as shown in Table 4. Source domains are

generated from the labelled historical normal and attack data from 2015 to 2018. Target domains contain unlabelled normal data and attack data from 2019 to 2021. Source and target domains can choose different locations from the 15 attackable buses to inject attack data. In this way, we can generate the source and target domains where attacks occur at different times and buses.

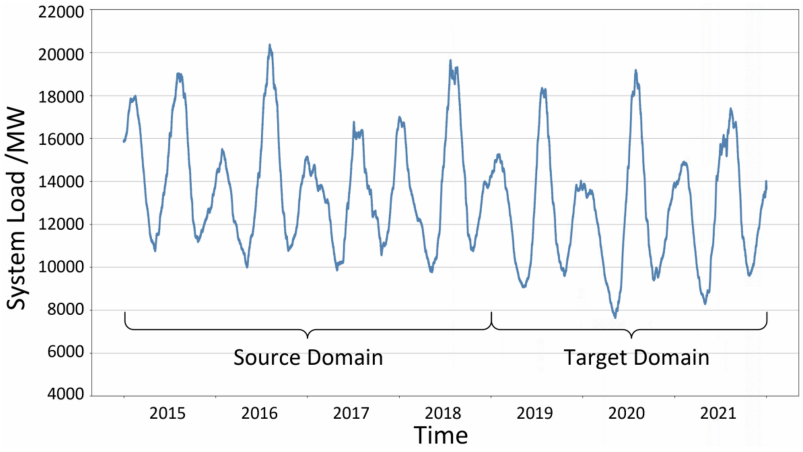


Figure 15: ISO New England seven-year load demand [6].

We randomly chose two domains from the labelled historical data between 2015 and 2018 as a pair of training and validation domains. Then we measure the distribution divergence between the two domains and calculate the model's accuracy drop from the training domain to the validation domain. Regression models with single metric and ensemble metrics are trained to learn the relationship between the measure divergence and accuracy drop. Then, the trained regression models are leveraged to predict the accuracy degradation on the unlabelled target domain between 2019 and 2021. If the predicted accuracy degradation exceeds the predefined threshold, the second step of the proposed framework will be applied to maintain the performance.

Table 4: Cases setup of temporal variation.

Cases	Seasons	Months	Source Domain from Year 2015 to 2018		Target Domain from Year 2019 to 2021	
			Mean of Load (MW)	Standard Deviation of Load (MW)	Mean of Load (MW)	Standard Deviation of Load (MW)
1	Winter	Mid-December to Mid-March	14,482.95	750.09	13,851.43	500.32
2	Spring	Mid-March to Mid-June	12,744.30	560.54	11,838.29	627.72
3	Summer	Mid-June to Mid-September	15,390.25	953.51	14,890.62	961.39
4	Fall	Mid-September to Mid-December	13,107.20	533.23	12,501.28	613.43

4.2.4 Results and Discussion

The RMSE and MaxAE of each regression model in predicting the accuracy drop are shown in Figure 16. The left four are the performance of each metric with linear regression, and the right one is that of the ensemble method with all metrics. We also compare selected metrics with the baseline. Following [116], the baseline does not learn regression models but takes the mean of the actual accuracy drop on the validation domains as its prediction. The RMSE and MaxAE of baseline are 7.24% and 19.76%, respectively. All selected metrics outperform the baseline in both RMSE and MaxAE. Among the four selected metrics with linear regression, PAD and JS have the highest prediction performance, reducing RMSE and MaxAE significantly to under 2.88% and 8.27%, respectively. MMD performs slightly worse than PAD and JS, but still improves the baseline by 3.03% in RMSE and 7.31% in MaxAE. Cos performs worst among the selected metrics but still achieves smaller RMSE and MaxAE than the baseline.

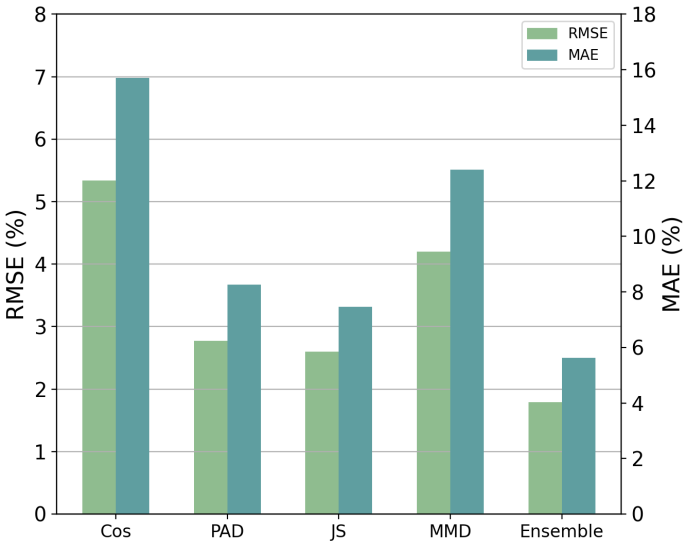


Figure 16: RMSE and MAE of accuracy drop prediction.

Moreover, compared to using a single metric to predict accuracy drop, the ensemble method provides better performance. The ensemble method achieves an RMSE as low as 1.79% and a MaxAE of 5.62%. This is because the ensemble method takes advantage of different metrics that can capture complementary distribution divergence information to further improve prediction performance [119]. Overall, the ensemble method decreases the RMSE to below 1.80%, indicating that the predicted accuracy drop with the ensemble metrics is close to the ground truth. This also implies that it is feasible to predict models' performance drop by distribution divergence.

4.3 Summary

This chapter studies the problem of when one should apply TL for intrusion detection in the smart grid. We propose a divergence-based transferability analysis to justify the

necessity of TL. We first leverage three metrics of different properties to evaluate the distribution divergence, and train linear regression models and neural network regression models for each metric to approximate the relation between accuracy drop and divergence. Moreover, we also train an ensemble model which takes advantage of all metrics selected from four divergence categories. Afterward, the regression models are applied to predict the accuracy drop on the unlabelled target domains and determine whether to apply TL.

Datasets from real normal operation profiles and simulated attacks are used to validate the effectiveness of the proposed transferability analysis against variations in attack timing, locations, and both. The result shows that the selected metrics and regression models are capable of predicting the accuracy drop of a trained model on an unseen dataset. Specifically, in all three scenarios, the proposed analysis with individual metrics demonstrates high accuracy in predicting accuracy drop with an RMSE lower than 4.20%, and DANN can be timely triggered to achieve an average accuracy improvement of 14.92%. Moreover, compared to using individual metrics with linear regression, the ensemble method provides better prediction performance, with an RMSE as low as 1.79%. The work of single metric transferability analysis has been published in the journal *IEEE Access* [150]. The work of ensemble metrics transferability analysis has been published in the journal *Energies*.

Chapter 5

Spatial-Temporal DA Training

5.1 Problem Formulation

With the divergence metrics and regression models, we can measure the divergence between the source domain and the unlabelled target domain, and predict the accuracy drop according to the divergence. If the divergence exceeds the accuracy drop threshold Π, we will trigger TL to maintain the performance. After identifying the tasks that require TL, there is another question to consider: how to extract effective features in the dynamic CPS scenarios during TL?

In this work, we focus on unsupervised TL, where we have the labelled source domain $\mathcal{D}_S = \{(x_{S_1}, y_{S_1}), ..., (x_{S_{n_S}}, y_{S_{n_S}})\}$, and unlabelled target domain $\mathcal{D}_T = \{(x_{T_1}), ..., (x_{T_{n_T}})\}$, where $x \in R^{T \times C}$, T is the length of the time series, and C is the dimension of feature space. Unsupervised TL is a common situation in real power systems intrusion detection, as the IDS deployed in the smart grid needs to detect intrusions in real time, and the newly

generated dataset is usually unlabelled.

We assume that source and target domains contain both normal and attack data, but the data distributions of the two domains are different. This research considers one case of data distribution divergence in the TL, covariate divergence, where two domains have the same conditional distribution, i.e., $P_{D_S}(Y|X) = P_{D_T}(Y|X)$, but their feature distributions are different, i.e., $P_{D_S}(X) \neq P_{D_T}(X)$. Specifically, this work considers a spatial-temporal TL problem, where attackers target the power systems during different periods when load demand has changed, and inject intrusions on different buses in the power grid.

To tackle this problem, this work aims to build a deep TL model that can learn informative features to mitigate the impact of data distribution divergence. The challenge is how to effectively extract spatial-temporal domain-invariant features for CPS data during TL, which is tackled by the proposed spatial-temporal DA training in Figure 17. If the predicted accuracy drop in the transferability analysis falls in the pre-defined range, the TL will be triggered. Specifically, a DA training model with CNN and LSTM is applied to extract spatial-temporal domain-invariant features, reduce distribution divergence, and improve detection performance against attacks at different times and locations.

5.2 Methodology

5.2.1 Domain-Adversarial Training

The domain-adversarial (DA) training of the proposed approach aims to extract the domain-invariant representations to reduce divergence between source and target domains. In this way, the model trained on the labelled source domain could also generalize well to

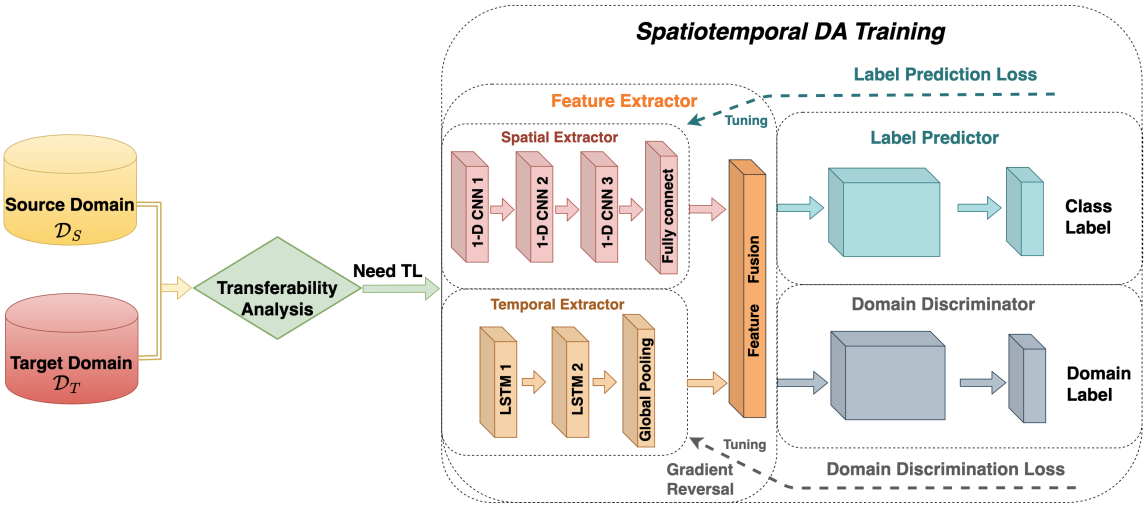


Figure 17: The proposed spatial-temporal DA training approach.

the unlabelled target domain. Our model builds on DANN [72], which introduces adversarial training into TL to extract domain-invariant features in an embedded representation layer. Moreover, we customize the design of the feature extractor to extract the spatial-temporal features from CPS data.

The DANN consists of three networks, namely, feature extractor, label predictor, and domain discriminator, as shown in Figure 17. The feature extractor is trained to extract the critical features from the source and target domains, and then feed the extracted features into the label predictor and the domain classifier simultaneously. The label predictor is trained to classify the training samples as normal operations or attack events. During the training process, only the features extracted from the source domain will be fed into the label predictor. The data from the target domain are not labelled, so they can not be used to train the label predictor. The features extracted from both source and target domains will be fed into the domain discriminator. The data from the source and target domains will

be labelled as 0 and 1, respectively. The domain discriminator is utilized to distinguish whether the data comes from the source domain or the target domain.

Moreover, a gradient reversal layer (GRL) is added between the feature extractor and the domain discriminator to make the domain discriminator perform poorly. In this way, the gradient back-propagated from the domain discriminator to the feature extractor is reversed, so the feature extractor will update its parameters in the direction to fail the domain discriminator.

By training three networks simultaneously, the feature extractor tries to minimize the label predictor loss and maximize the domain discriminator loss, thereby extracting domain-invariant and label-discriminative features. The total loss function is constructed as:

$$L(\theta_f, \theta_y, \theta_d) = \sum_{i=1}^{m} L_y^i(\theta_f, \theta_y) - \lambda \sum_{j=1}^{n} L_d^j(\theta_f, \theta_d), \tag{22}$$

where L_y is the label predictor loss, L_d is the domain discriminator loss, λ is the adaptation factor used to tune the trade-off between two network losses [72], and the minus sign indicates the adversarial training. θ_f, θ_y, and θ_d denote the sets of parameters in the feature extractor, label predictor, and domain discriminator, respectively. Through DA training, the domain divergence is minimized, and inter-class distance is maximized. Consequently, the attack detector trained on the source domain could generalize well to the target domain.

5.2.2 Spatial-Temporal Feature Extraction

We further customize the design of the feature extractor to extract the deep spatial-temporal features from CPS data. Motivated by the success of deep learning on CV

and NLP tasks, the feature extractor in this work consists of CNN and LSTM to extract domain-invariant spatial and temporal features, as depicted in Figure 17. The CNN is used to extract cross-measurement correlation of CPS data since they can effectively extract spatial features [151]. The LSTM, which is capable of learning long-term dependencies [28], works on mining the context information of the sequential measurement flow. A parallel combination of three layers of CNN and two layers of LSTM is adopted in this work because extensive experiments conducted by Zhang *et al.* [152] have proved that this combination could effectively extract spatial-temporal features. A feature fusion layer is leveraged to contact the extracted spatial and temporal features as the spatial-temporal features and feed them to the label predictor and domain discriminator.

Typically, the raw measurements from different smart meters at time index t is a one-dimensional (1-D) vector:

$$v_t = [m_t^1,\ m_t^2,\ ...,\ m_t^C], \tag{23}$$

where m_t^i is the reading of the ith measurement. For an observation period $[t,\ t+N]$, there will be a measurement flow with $N+1$ vectors, each containing C measurements. To extract temporal features with LSTM, we employ the sliding window to divide measurement flow into individual segments. Each segment has a fixed length of time series vectors and is defined as:

$$s_j = [v_t,\ v_{t+1},\ ...,\ v_{t+T-1}]^{\mathsf{T}}, \tag{24}$$

where T is the fixed sliding window size, and s_j denotes the jth segment fed into the feature extractor. Each segment is fed into CNN and LSTM concurrently.

The CNN is responsible for learning spatial features. Following [153], we employ

three layers of 1-D CNN to extract the spatial features of each measurement vector in the segment s_j:

$$r_k = Conv1D(v_k), \tag{25}$$

where r_k is the extracted spatial features corresponding to the measurement vector v_k. To be comparable to the temporal features in terms of size, the extracted spatial features r_k in the same segment are fed into a global average pooling (GAP) layer. The GAP can reduce the computational burden and avoid overfitting, thus enhancing the generality of spatial features. The final spatial features can be expressed as:

$$f_{spatial} = G_{GAP}(r_t,\ r_{t+1},\ ...,\ r_{t+T-1}), \tag{26}$$

where G_{GAP} is the pooling layer. $f_{spatial}$ denotes a single vector representing the spatial features.

The LSTM is leveraged to extract the temporal features of multivariate time series measurements. Specifically, the LSTM networks have two LSTM layers, and each LSTM layer has T units since each segment contains T measurement vectors. Since we are interested in segment-level intrusion detection, the output of the last unit in the second layer is selected to generate temporal features:

$$h^2_{t+T-1} = LSTM(s_j), \tag{27}$$

where h^2_{t+T-1} is the output of the last unit in the second layer.

Then, a fully connected layer is added to improve temporal feature representation [152]:

$$f_{temporal} = G_{FC}(h^2_{t+T-1}), \tag{28}$$

where G_{FC} is the fully connected layer. $f_{temporal}$ denotes a single vector representing the temporal features.

Finally, the extracted spatial and temporal features are contacted as the spatial-temporal features in the feature fusion layer:

$$f_{spatial-temporal} = [f_{spatial},\ f_{temporal}]. \tag{29}$$

Then the spatial-temporal features are fed into the label predictor and domain discriminator for DA training. By training three networks simultaneously, the feature extractor can learn the domain-invariant and label-discriminative spatial-temporal features, and improve the attack detection performance.

5.3 Experiments Setup

5.3.1 Data & Case Setup

We use the same dataset and experiment setup as Section 4.2. We select 60 minutes as the sliding window, i.e., $T = 60$, to transform measurement data into time series data. Moreover, this paper considers that the attack data percentage is not always consistent in real-world power systems. Based on the fact that the attack data is generally rare compared to the normal data in the smart grid [154], this paper sets the attack data percentage range

of [5%, 40%]. Meanwhile, considering many ML algorithms are tested and developed on balanced datasets, this work also sets up relatively balanced datasets with an attack data percentage range of [45%, 55%]. Combining this two, this work has datasets with an attack data percentage range of [5%, 55%]. The attack data percentages are chosen in every 5% among [5%, 55%] to validate the robustness of the proposed framework, that is, 5%, 10%, 15%, 20%, 25%, 30%, 35%, 40%, 45%, 50%, and 55%.

5.3.2 Comparison Models

The performance of TADA is validated by comparing three non-TL models and two state-of-art TL models. The three non-TL models are MLP, linear SVM, for their high performance and low computational complexity in intrusion detection [33, 34], and fully convolutional network (FCN) [153], for its ability to learn deep spatial features. For the state-of-art TL models, we choose DANN and the convolutional deep domain adaptation model for time series data (CoDATS) [155] for their capacity in domain adaptation.

We adopt accuracy and F1-score for evaluation and comparison, which can be calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{30}$$

where TP, TN, FP, and FN represent true positive, true negative, false positive, and false negative, respectively. Since we use imbalanced domains in this work, examining the accuracy alone could sometimes be misleading. Hence we also introduce the F1-score, which is the harmonic mean of precision and recall:

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall}, \tag{31}$$

where $Precision = \frac{TP}{TP+FP}$, $Recall = \frac{TP}{TP+FN}$.

5.3.3 Model Implementation

The 3-layer CNN is set with kernel sizes of {8, 5, 3} and kernel numbers of {128, 256, 128}. Following [151], we use grid research and find that combining the LSTM time step size of 60 (in minutes) and the hidden state size of 100 achieves robust detection performance. We refer [72] and gradually change the adaptation factor λ in Eq. (22) from 0 to 1 to tune the trade-off between the label predictor loss and the domain discriminator loss. In this way, the domain discriminator loss will be suppressed at the early training stage. Furthermore, an annealing learning rate that decreases from 0.01 to 0.001 is applied in this work.

We use MATLAB R2020b and MATPOWER v7.1 to generate datasets. All models are implemented in Python v3.6, TensorFlow v2.4.0, Keras v2.4.3, and Scikit-learn v0.24.2. The hardware environment for training and testing is an AMD Ryzen 9 3900X 12-Core Processor 3.80 GHz with 32GB RAM, and an NVIDIA GeForce RTX 2070 Super GPU.

5.4 Results and Discussion

5.4.1 FDI Detection Performance

Table 5 illustrates the accuracy of TADA and five compared models in detecting FDI attacks at different seasons. The accuracy shown in the table is the average of every 1080 experiments, where we inject attacks on individual buses of different locations. We use the ensemble method to predict the accuracy drop of each case. Since FDI attacks may

severely impact the power systems, this work sets an accuracy drop of 10% as the threshold for activating TL. The target seasons where the actual accuracy drop is smaller than 10% are underlined. In these seasons, the accuracy drop is not significant enough to call for TL, because this small accuracy drop may be the normal accuracy variation. In this case, TL is unnecessary, because frequently applying TL can be costly but the performance boost would be trivial. Table 5 shows that the predicted accuracy drop of all the underlined seasons are less than 10%, indicating the ensemble method successfully identifies all TL-unnecessary cases. Overall, the predicted accuracy drop is close to the actual accuracy drop. We can also find that except for winter in Case 4, the source and target domain pairs between the same season, winter and summer, spring and fall, demonstrate less accuracy drop. This is because the load demand of source and target domains from the aforementioned pairs is similar, as shown in Table 4. Similar load demand indicates less data distribution divergence and accuracy drop.

Among all methods, SVM and MLP have the lowest detection accuracy, with an average accuracy of 72.55% and 74.10%, respectively. This is because they can neither learn deep spatial-temporal features nor use domain adaptation to mitigate the impact of distribution divergence. FCN performs slightly better than SVM and MLP with an average accuracy of 78.06%, because FCN can leverage CNN to extract spatial features within the smart grid measurements. But FCN is also a non-TL model, so it will suffer performance degradation when facing significant distribution divergence. Moreover, compared to three non-TL models (SVM, MLP, and FCN), TL models (DANN, CoDATS, and TADA) achieve higher detection accuracy. This suggests that the three TL models can extract domain-invariant features to improve classification accuracy, while the non-TL

Table 5: Comparison of TADA and five ML classifiers in detecting FDI attacks at different seasons

Cases	Source Seasons	Target Seasons	Predicted Drop	Actual Drop	TADA	CoDATS	DANN	FCN	SVM	MLP	Best-Case Margin	Worst-Case Margin
1	Winter	Winter	8.97	8.89	**97.31**	93.86	91.17	86.68	79.56	81.03	+17.74	+3.44
		Spring	26.01	25.20	**94.87**	87.69	85.69	72.44	67.65	66.93	+27.94	+7.18
		Summer	11.47	11.23	**95.74**	93.31	89.13	79.82	74.13	75.57	+21.61	+2.43
		Fall	20.97	20.65	**94.84**	90.93	85.51	71.41	66.02	69.27	+28.82	+3.91
2	Spring	Winter	18.55	18.74	**96.68**	89.14	88.16	77.30	71.74	70.42	+26.25	+7.54
		Spring	12.81	13.02	**96.05**	93.00	89.74	81.50	75.66	78.91	+20.39	+3.04
		Summer	19.62	19.14	**95.42**	90.49	88.03	70.23	67.72	71.04	+27.70	+4.93
		Fall	6.26	6.36	**97.89**	93.21	90.23	86.22	78.85	82.69	+19.04	+4.68
3	Summer	Winter	17.28	17.62	**95.08**	90.55	86.03	78.55	72.15	73.56	+22.93	+4.53
		Spring	28.21	27.19	**92.90**	89.47	84.39	70.83	64.86	66.98	+28.04	+3.43
		Summer	7.19	7.29	**96.87**	89.79	90.12	85.07	79.25	81.89	+17.61	+6.75
		Fall	23.17	23.80	**94.99**	86.57	82.80	71.50	68.19	64.67	**+30.32**	+8.42
4	Fall	Winter	11.76	11.49	**96.52**	90.78	91.06	84.12	78.53	78.10	+18.42	+5.46
		Spring	14.48	14.75	**94.98**	93.30	90.04	78.42	74.35	76.44	+20.63	+1.68
		Summer	24.77	23.92	**93.08**	89.68	81.19	72.99	67.32	67.40	+25.75	+3.39
		Fall	9.20	9.09	**96.08**	94.51	92.56	81.91	74.76	80.70	+21.32	**+1.57**

The target seasons are underlined where the actual accuracy drop is smaller than a predefined threshold (10%) and thus DA training is unnecessary.

models fail to mitigate the impact of distribution divergence.

Among the three TL models, although DANN can learn domain-invariant features, it has the lowest detection accuracy since it can not extract temporal or spatial features. TADA outperforms CoDATS by an average improvement of 4.56%. This is because CoDATS can only learn temporal features, but TADA can learn both temporal and spatial features concurrently to further improve FDI detection performance. Overall, TADA demonstrates the highest accuracy in all cases. The best-case and the worst-case improvements reach +30.32% compared to MLP during fall in Case 3, and +1.57% compared to CoDATS during fall in Case 4. The results suggest that TADA can not only take advantage of DA training to extract domain-invariant features but also leverage LSTM and CNN to learn spatial-temporal features, to achieve superior FDI detection performance against distribution divergence.

Considering we are using imbalanced domains in this work, we further present the F1-score of TADA and other compared models under different attack data percentages, as shown in Figure 18. The results show that the detection performance of all methods is generally increasing as the percentage of attack data increases and the dataset becomes more balanced. When the attack data percentage is less than 25%, TADA demonstrates a significant improvement compared to other models. The F1-score of TADA does not further improve when the attack data percentage is higher than 25%, but it still outperforms other models. Overall, TADA shows the highest F1-score when the attack data percentage varies, which indicates that TADA can achieve robust detection performance against variations in attack data percentage.

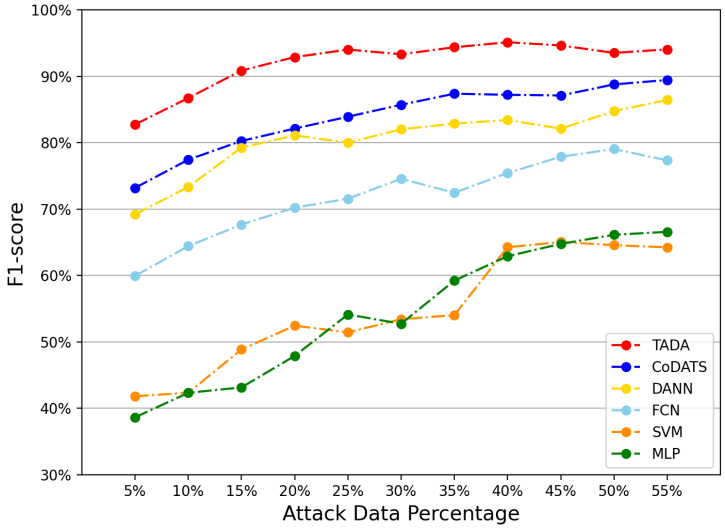
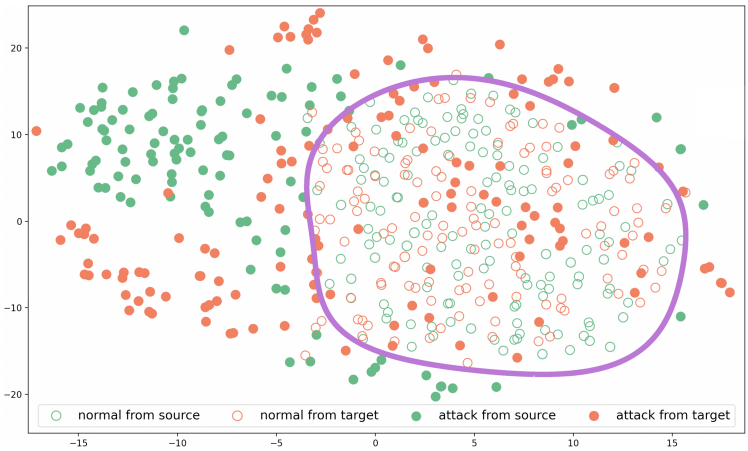


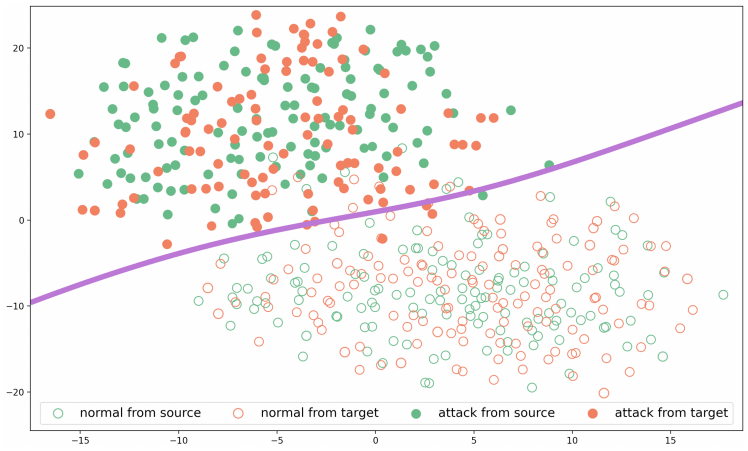
Figure 18: Comparison of F1-score of TADA and other ML classifiers under different attack data percentages.

5.4.2 Visualization of Data Distribution

To vividly visualize the results of the domain-invariant feature extraction, Figure 19 employs t-SNE and presents normal and attack data distribution without and with applying DA training. Specifically, for Figure 19 (a) where DA training is not performed, we deactivate the domain discriminator and train a serial connection of the feature extractor and the label predictor, then leverage t-SNE to present the output of the feature extractor, i.e., the feature fusion layer in Figure 17. For Figure 19 (b) where DA training is performed, we train the whole TADA model and present the output of the feature extractor. We also plot the decision boundary on the attack detection problem, which is given by the label predictor in Figure 17. Specifically, a sample will be classified as attack data if the output of the label predictor is greater than 0.5. Otherwise, it will be classified as normal data.



(a)



(b)

Figure 19: Distribution of normal and attack data in the feature fusion layer when (a) DA training is not applied; and (b) DA training is applied. The circles represent normal data, while the dots represent the attack data. The green dots and circles correspond to data from the source domain, and the orange dots and circles correspond to the data from the target domain. We also plot the decision boundary in purple.

Figure 19 (a) shows that without DA training, the distributions of source and target domains are different, especially for the attack data injected at different times and locations. Moreover, the decision boundary can distinguish between normal and attack data from the source domain, but can not perfectly classify normal and attack data from the target domain. This is because the classifier is trained based on the labelled source domain, and the target domain has a different data distribution. After applying DA training, however, the distribution divergence between the two domains is decreased. The source and target domains share a similar distribution in the feature fusion layer, as shown in Figure 19 (b). Specifically, the attack data are clustered on the upper left, while the normal data are clustered on the lower right. Therefore, the label predictor trained on features extracted from the source domain could also generalize well to the target domain, and achieve a high and robust detection performance. The results demonstrate that TADA can effectively reduce distribution divergence and thus improve detection performance.

5.5 Summary

This chapter studies the problem of how to extract features effectively during TL for attack detection in dynamic CPS scenarios. Considering the internal spatial and temporal features of CPS data, this chapter proposes a spatial-temporal DA training approach. The approach develops a DA training architecture with CNN and LSTM to extract the spatial-temporal domain-invariant features to reduce distribution divergence and thus improve detection performance. The TADA is evaluated in extensive experiments where FDI attacks are injected at different times and locations.

The attack detection results show that the TADA can extract effective spatial-temporal domain-invariant features to improve attack detection performance under system and attack variations. Compared to the state-of-the-art models, TADA demonstrates the highest detection accuracy, achieving an average accuracy of 95.58%. Moreover, the robustness of the framework is validated under different attack data percentages, with an average F1-score of 92.02%. The work of spatial-temporal DA adversarial training has been published in the journal *Energies*.

Chapter 6

Conclusions

As one of the national CPS infrastructures, the smart grid provides efficient, secure, and sustainable electricity in a growing power-demanding society. The application of sensing, communications, and distributed computing empowers the smart grid in monitoring and controlling, however, it renders the smart grid exposed to various cyber-attacks and increases its vulnerability. As reported in recent studies [156, 157, 146], cyber-attacks on critical infrastructures could have severe social, economic, and physical impacts. Aware of the importance of cyber-security situation awareness to the power systems, various ML detection mechanisms have been exploited extensively and demonstrated high accuracy and efficient computation in attack detection [31, 158, 159], such as kNN and SVM. While ML has been extensively studied to detect attacks in the smart grid, traditional ML models may suffer performance degradation when facing the system and attack variations. TL is a promising approach to mitigate the impact of data distribution divergence and maintain attack detection performance.

While various TL approaches have been proposed to achieve state-of-the-art performance, there is still limited work on a more fundamental question that can be called *transferability*: when should one consider the performance of a trained model has degraded significantly enough to justify the need for TL, without having to retrain a new model from scratch? To address this problem, we propose a divergence-based transferability analysis to justify the necessity of TL. First, three metrics of different properties are selected to evaluate the distribution divergence. Then, two regression models are trained to approximate the relation between accuracy drop and divergence and applied to predict accuracy drop on the unlabelled target domain. We also train an ensemble regression model that takes all selected metrics as the input to predict the accuracy drop. The experiment results show that the proposed analysis demonstrated high accuracy in predicting accuracy drop from the divergence. The single metric method has an RMSE lower than 4.20% in all experiments, and the ensemble method achieves an RMSE as low as 1.79%.

Meanwhile, we also study the problem of how to extract effective features during TL for attack detection in power systems. There are rich spatial and temporal features in the CPS data that can be used to help discriminate attacks from normal data [132]. To tackle this challenge, we propose a spatial-temporal DA training approach based on DA training and deep feature extraction techniques. In detail, two deep learning models, CNN and LSTM, are leveraged to extract spatial and temporal features, respectively. Then, the DA training model is applied to extract spatial-temporal domain-invariant features, and reduce distribution divergence between source and target domains. We consider attacks that may happen at different times and locations in a dynamic power system and evaluate TADA on realistic datasets. The attack detection results show that the TADA can extract effective

spatial-temporal domain-invariant features to improve attack detection performance and achieve an average accuracy of 95.58%. The results also demonstrate that the TADA can achieve robust detection performance against variations of attack data percentages, with an average F1-score of 92.02%.

For future work, on the one hand, the accuracy drop and divergence are highly correlated in the transferability analysis experiments. It would be interesting to conduct additional experiments to validate whether this is the case for other potential real-world scenarios, dig into the reason behind it, and improve the understanding of transferability analysis. On the other hand, in this work, we assume that the attackers only inject one bus when launching the attack. However, there are more sophisticated attacks in the studies that can inject several buses simultaneously, e.g., coordinated cyber-physical attacks (CCPAs) [160] and coordinated topology attacks [161]. We could study more advanced coordinated attack scenarios to get a more profound understanding of TL in CPS.

While the concept of FDI was originally introduced in smart grid applications, it can occur in other scenarios where state estimation is applied. For instance, various state estimation techniques have been developed and used for aircraft engine health management and fault diagnosis [162]. For example, in measurement validation and diagnostics procedure, the Kalman filter estimates the degradations of the components' performance parameters by comparing the error between predicted measurements and raw measurements [163]. With the development of sensor measurement technology, more sensors are utilized in aircraft engines for health management, which also exposes this complex multi-sensor system to cyber attacks [164]. Some data integrity attacks, such as FDI, can

compromise the aircraft sensor measurements and inject malicious data into state estimation stealthily. Since the reliability of aircraft engine state estimation is crucial to the aircraft's performance and flight safety, it would be interesting to test our proposed detection framework on aircraft engine health management and see if the model generalizes well. Furthermore, FDI specifically means the cases when attackers compromise sensor readings in a stealthy way to bypass the detector. With the increasing interconnections among CPS devices, attackers are also interested in exploiting similar attacks in other scenarios. We would like to extend the proposed method to other application domains, like smart healthcare [165], finance [166], and governance [167].

Bibliography

[1] Y. Zhang and J. Yan, "Domain-adversarial transfer learning for robust intrusion detection in the smart grid," in *2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. IEEE, 2019, pp. 1–6.

[2] Canadian Centre for Cyber Security. "Cyber threat bulletin: Cyber threat to operational technology". [Online]. Available: https://cyber.gc.ca/en/guidance/cyber-threat-bulletin-cyber-threat-operational-technology#fn34

[3] R. Deng, G. Xiao, R. Lu, H. Liang, and A. V. Vasilakos, "False data injection on state estimation in power systems—attacks, impacts, and defense: A survey," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 2, pp. 411–423, 2016.

[4] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, pp. 1–22, 2019.

[5] H. Liu and B. Lang, "Machine learning and deep learning methods for intrusion detection systems: A survey," *applied sciences*, vol. 9, no. 20, p. 4396, 2019.

[6] "ISO New England - energy, load, and demand reports," https://www.iso-ne.com/isoexpress/web/reports/load-and-demand/-/tree/dmnd-five-minute-sys, 2019.

[7] Illinois Center for a Smarter Electric Grid (ICSEG). "IEEE 30-bus system". [Online]. Available: https://icseg.iti.illinois.edu/ieee-30-bus-system/

[8] C. Konstantinou and M. Maniatakos, "A case study on implementing false data injection attacks against nonlinear state estimation," in *Proceedings of the 2nd ACM workshop on cyber-physical systems security and privacy*, 2016, pp. 81–92.

[9] A. Humayed, J. Lin, F. Li, and B. Luo, "Cyber-physical systems security—a survey," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1802–1831, 2017.

[10] R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, "Cyber-physical systems: The next computing revolution," in *Design Automation Conference*, 2010, pp. 731–736.

[11] X. Yu and Y. Xue, "Smart grids: A cyber–physical systems perspective," *Proceedings of the IEEE*, vol. 104, no. 5, pp. 1058–1070, 2016.

[12] Y. Yan, Y. Qian, H. Sharif, and D. Tipper, "A survey on cyber security for smart grid communications," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 4, pp. 998–1010, 2012.

[13] R. Langner, "Stuxnet: Dissecting a cyberwarfare weapon," *IEEE Security Privacy*, vol. 9, no. 3, pp. 49–51, 2011.

[14] G. Liang, S. R. Weller, J. Zhao, F. Luo, and Z. Y. Dong, "The 2015 ukraine blackout: Implications for false data injection attacks," *IEEE Transactions on Power Systems*, vol. 32, no. 4, pp. 3317–3318, 2016.

[15] R. Deng, P. Zhuang, and H. Liang, "Ccpa: Coordinated cyber-physical attacks and countermeasures in smart grid," *IEEE Transactions on Smart Grid*, vol. 8, no. 5, pp. 2420–2430, 2017.

[16] A. Bindra, "Securing the power grid: Protecting smart grids and connected power systems from cyberattacks," *IEEE Power Electronics Magazine*, vol. 4, no. 3, pp. 20–27, 2017.

[17] A. Dabrowski, J. Ullrich, and E. R. Weippl, "Grid shock: Coordinated load-changing attacks on power grids: The non-smart power grid is vulnerable to cyber attacks as well," in *Proceedings of the 33rd Annual Computer Security Applications Conference*, 2017, pp. 303–314.

[18] N. Perlroth and D. E. Sanger, "Cyberattacks put russian fingers on the switch at power plants, us says," *New York Times*, vol. 15, 2018.

[19] F. Li, X. Yan, Y. Xie, Z. Sang, and X. Yuan, "A review of cyber-attack methods in cyber-physical power system," in *2019 IEEE 8th International Conference on Advanced Power System Automation and Protection (APAP)*, 2019, pp. 1335–1339.

[20] S. Tzu, "The concept of deception and its applicability in india-china framework," *Centre for Land Warfare Studies*, no. 298, pp. 1–13, 2021.

[21] M. Z. Alom and T. M. Taha, "Network intrusion detection for cyber security on neuromorphic computing system," in *2017 International Joint Conference on Neural Networks (IJCNN)*, May 2017, pp. 3830–3837.

[22] M. Yousefi-Azar, V. Varadharajan, L. Hamey, and U. Tupakula, "Autoencoder-based feature learning for cyber security applications," in *2017 International Joint Conference on Neural Networks (IJCNN)*, May 2017, pp. 3854–3861.

[23] H. He and J. Yan, "Cyber-physical attacks and defences in the smart grid: a survey," *IET Cyber-Physical Systems: Theories & Applications*, vol. 1, no. 1, pp. 13–27, 2016.

[24] M. Trevor and B. Nick, "Business blackout: The insurance implications of a cyber attack on the us power grid," Lloyd's and the University of Cambridge Centre for Risk Studies, Tech. Rep., 2015.

[25] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 16–24, 2013.

[26] A. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016.

[27] C. Zhang, D. Song, Y. Chen, X. Feng, C. Lumezanu, W. Cheng, J. Ni, B. Zong, H. Chen, and N. V. Chawla, "A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 1409–1416.

[28] Q. Deng and J. Sun, "False data injection attack detection in a power grid using rnn," in *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, 2018, pp. 5983–5988.

[29] A. Giani, E. Bitar, M. Garcia, M. McQueen, P. Khargonekar, and K. Poolla, "Smart grid data integrity attacks," *IEEE Transactions on Smart Grid*, vol. 4, no. 3, pp. 1244–1253, 2013.

[30] Q. Yang, D. An, R. Min, W. Yu, X. Yang, and W. Zhao, "On optimal pmu placement-based defense against data integrity attacks in smart grid," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 7, pp. 1735–1750, 2017.

[31] G. Cheng, Y. Lin, J. Zhao, and J. Yan, "A highly discriminative detector against false data injection attacks in ac state estimation," *IEEE Transactions on Smart Grid*, vol. 13, no. 3, pp. 2318–2330, 2022.

[32] M. Ozay, I. Esnaola, F. Yarman Vural, S. Kulkarni, and H. Poor, "Machine learning methods for attack detection in the smart grid," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 8, pp. 1773–1786, Aug. 2016.

[33] K. Shaukat, S. Luo, V. Varadharajan, I. A. Hameed, S. Chen, D. Liu, and J. Li, "Performance comparison and current challenges of using machine learning techniques in cybersecurity," *Energies*, vol. 13, no. 10, 2020. [Online]. Available: https://www.mdpi.com/1996-1073/13/10/2509

[34] A. Kumar, N. Saxena, S. Jung, and B. J. Choi, "Improving detection of false data injection attacks using machine learning with feature selection and oversampling," *Energies*, vol. 15, no. 1, 2022. [Online]. Available: https://www.mdpi.com/1996-1073/15/1/212

[35] Y. Zhang and J. Yan, "Semi-supervised domain-adversarial training for intrusion detection against false data injection in the smart grid," in *2020 International Joint Conference on Neural Networks (IJCNN)*, 2020, pp. 1–7.

[36] S. Houidi, D. Fourer, F. Auger, H. B. A. Sethom, and L. Miègeville, "Comparative evaluation of non-intrusive load monitoring methods using relevant features and transfer learning," *Energies*, vol. 14, no. 9, 2021. [Online]. Available: https://www.mdpi.com/1996-1073/14/9/2726

[37] Y. Zhang, "Domain adversarial transfer learning for robust cyber-physical attack detection in the smart grid," Ph.D. dissertation, Concordia University, 2020.

[38] Z. Zhang, J. Li, C. Manikopoulos, J. Jorgenson, and J. Ucles, "Hide: a hierarchical network intrusion detection system using statistical preprocessing and neural network classification," in *Proc. IEEE Workshop on Information Assurance and Security*, vol. 85, 2001, p. 90.

[39] C. Alcaraz, J. Lopez, and S. Wolthusen, "Policy enforcement system for secure interoperable control in distributed smart grid systems," *Journal of Network and Computer Applications*, vol. 59, pp. 301–314, 2016.

[40] Y. He, G. J. Mendis, and J. Wei, "Real-time detection of false data injection attacks in smart grid: A deep learning-based intelligent mechanism," *IEEE Transactions on Smart Grid*, vol. 8, no. 5, pp. 2505–2516, 2017.

[41] W. Lu and A. A. Ghorbani, "Network anomaly detection based on wavelet analysis," *EURASIP Journal on Advances in Signal Processing*, vol. 2009, pp. 1–16, 2008.

[42] J. Li, Z. Zhao, and R. Li, "Machine learning-based ids for software-defined 5g network," *Iet Networks*, vol. 7, no. 2, pp. 53–60, 2018.

[43] M. Yu, "A nonparametric adaptive cusum method and its application in network anomaly detection," *International J. Advancements in Computing Technology*, vol. 4, no. 1, pp. 280–288, 2012.

[44] P. Chhabra, C. Scott, E. D. Kolaczyk, and M. Crovella, "Distributed spatial anomaly detection," in *IEEE INFOCOM 2008-The 27th Conference on Computer Communications*. IEEE, 2008, pp. 1705–1713.

[45] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: methods, systems and tools," *Ieee communications surveys & tutorials*, vol. 16, no. 1, pp. 303–336, 2013.

[46] K. Wang and S. J. Stolfo, "Anomalous payload-based network intrusion detection," in *International workshop on recent advances in intrusion detection*. Springer, 2004, pp. 203–222.

[47] B. G. Atli, Y. Miche, A. Kalliola, I. Oliver, S. Holtmanns, and A. Lendasse, "Anomaly-based intrusion detection using extreme learning machine and aggregation of network traffic statistics in probability space," *Cognitive Computation*, vol. 10, no. 5, pp. 848–863, 2018.

[48] L. Boero, M. Cello, M. Marchese, E. Mariconti, T. Naqash, and S. Zappatore, "Statistical fingerprint-based intrusion detection system (sf-ids)," *International Journal of Communication Systems*, vol. 30, no. 10, p. e3225, 2017.

[49] M. Dusi, M. Crotti, F. Gringoli, and L. Salgarelli, "Tunnel hunter: Detecting application-layer tunnels with statistical fingerprinting," *Computer Networks*, vol. 53, no. 1, pp. 81–97, 2009.

[50] M. Aiello, M. Mongelli, and G. Papaleo, "Dns tunneling detection through statistical fingerprints of protocol messages and machine learning," *International Journal of Communication Systems*, vol. 28, no. 14, pp. 1987–2002, 2015.

[51] F. Simmross-Wattenberg, J. I. Asensio-Perez, P. Casaseca-de-la Higuera, M. Martin-Fernandez, I. A. Dimitriadis, and C. Alberola-Lopez, "Anomaly detection in network traffic based on statistical inference and stable modeling," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 4, pp. 494–509, 2011.

[52] Z. Zhu and T. Dumitraş, "Featuresmith: Automatically engineering features for malware detection by mining the security literature," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 767–778.

[53] Q. Rajput, N. S. Khan, A. Larik, and S. Haider, "Ontology based expert-system for suspicious transactions detection," *Computer and Information Science*, vol. 7, no. 1, p. 103, 2014.

[54] M. Gajewski, J. M. Batalla, G. Mastorakis, and C. X. Mavromoustakis, "A distributed ids architecture model for smart home systems," *Cluster Computing*, vol. 22, no. 1, pp. 1739–1749, 2019.

[55] A. S. Ashoor and S. Gore, "Importance of intrusion detection system (ids)," *International Journal of Scientific and Engineering Research*, vol. 2, no. 1, pp. 1–4, 2011.

[56] J. Jabez and B. Muthukumar, "Intrusion detection system (ids): Anomaly detection using outlier detection approach," *Procedia Computer Science*, vol. 48, pp. 338–346, 2015.

[57] A. Le, J. Loo, Y. Luo, and A. Lasebae, "Specification-based ids for securing rpl from topology attacks," in *2011 IFIP Wireless Days (WD)*, 2011, pp. 1–3.

[58] J. Yan, B. Tang, and H. He, "Detection of false data attacks in smart grid with supervised learning," in *2016 International Joint Conference on Neural Networks (IJCNN)*, July 2016, pp. 1395–1402.

[59] N. I. Haque, M. H. Shahriar, M. G. Dastgir, A. Debnath, I. Parvez, A. Sarwat, and M. A. Rahman, "Machine learning in generation, detection, and mitigation of cyberattacks in smart grid: A survey," *arXiv preprint arXiv:2010.00661*, 2020.

[60] A. Jindal, A. Dua, K. Kaur, M. Singh, N. Kumar, and S. Mishra, "Decision tree and svm-based data analytics for theft detection in smart grid," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 3, pp. 1005–1016, 2016.

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

[62] F. A. A. Alseiari and Z. Aung, "Real-time anomaly-based distributed intrusion detection systems for advanced metering infrastructure utilizing stream data mining," in *2015 International Conference on Smart Grid and Clean Energy Technologies (ICSGCE)*, 2015, pp. 148–153.

[63] Z. Liu, M.-U.-D. Ghulam, Y. Zhu, X. Yan, L. Wang, Z. Jiang, and J. Luo, "Deep learning approach for ids," in *Fourth International Congress on Information and Communication Technology*. Springer, 2020, pp. 471–479.

[64] E. Min, J. Long, Q. Liu, J. Cui, and W. Chen, "Tr-ids: Anomaly-based intrusion detection through text-convolutional neural network and random forest," *Security and Communication Networks*, vol. 2018, 2018.

[65] W. Wang, Y. Sheng, J. Wang, X. Zeng, X. Ye, Y. Huang, and M. Zhu, "Hast-ids: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection," *IEEE Access*, vol. 6, pp. 1792–1806, 2018.

[66] M. Rigaki and S. Garcia, "Bringing a gan to a knife-fight: Adapting malware communication to avoid detection," in *2018 IEEE Security and Privacy Workshops (SPW)*, 2018, pp. 70–75.

[67] S. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[68] S. Niu, Y. Liu, J. Wang, and H. Song, "A decade survey of transfer learning (2010–2020)," *IEEE Transactions on Artificial Intelligence*, vol. 1, no. 2, pp. 151–166, 2020.

[69] F. Xu, J. Yu, and R. Xia, "Instance-based domain adaptation via multiclustering logistic approximation," *IEEE Intelligent Systems*, vol. 33, no. 1, pp. 78–88, 2018.

[70] S. Niu, J. Wang, Y. Liu, and H. Song, "Transfer learning based data-efficient machine learning enabled classification," in *2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*. IEEE, 2020, pp. 620–626.

[71] S. Niu, J. Wang, Y. Liu, and H. Song, "Transfer learning based data-efficient machine learning enabled classification," in *2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing,*

Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), 2020, pp. 620–626.

[72] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, Jan. 2016.

[73] M. J. Afridi, A. Ross, and E. M. Shapiro, "On automated source selection for transfer learning in convolutional neural networks," *Pattern recognition*, vol. 73, pp. 65–75, 2018.

[74] Q. Wu, H. Wu, X. Zhou, M. Tan, Y. Xu, Y. Yan, and T. Hao, "Online transfer learning with multiple homogeneous or heterogeneous sources," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 7, pp. 1494–1507, 2017.

[75] M. Long, Y. Cao, Z. Cao, J. Wang, and M. I. Jordan, "Transferable representation learning with deep adaptation networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 12, pp. 3071–3085, 2019.

[76] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2962–2971.

[77] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *International conference on machine learning*. PMLR, 2015, pp. 1180–1189.

[78] H. Yan, Y. Ding, P. Li, Q. Wang, Y. Xu, and W. Zuo, "Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 945–954.

[79] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big Data*, vol. 3, no. 1, p. 9, May 2016.

[80] Z. Wang, Y. Song, and C. Zhang, "Transferred dimensionality reduction," in *Joint European conference on machine learning and knowledge discovery in databases*. Springer, 2008, pp. 550–565.

[81] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, "Self-taught clustering," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 200–207.

[82] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, "Self-taught learning: transfer learning from unlabeled data," in *Proceedings of the 24th international conference on Machine learning*, 2007, pp. 759–766.

[83] R. Ahmadi, R. D. Macredie, and A. Tucker, "Intrusion detection using transfer learning in machine learning classifiers between non-cloud and cloud datasets," in *Intelligent Data Engineering and Automated Learning (IDEAL)*, 2018, pp. 556–566.

[84] G. Csurka, "Domain adaptation for visual applications: A comprehensive survey," *arXiv preprint arXiv:1702.05374*, 2017.

[85] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, pp. 135–153, 2018.

[86] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, "Simultaneous deep transfer across domains and tasks," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 4068–4076.

[87] X. Peng, J. Hoffman, X. Y. Stella, and K. Saenko, "Fine-to-coarse knowledge transfer for low-res image classification," in *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 3683–3687.

[88] S. Motiian, M. Piccirilli, D. A. Adjeroh, and G. Doretto, "Unified deep supervised domain adaptation and generalization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5715–5725.

[89] G. Hinton, O. Vinyals, J. Dean *et al.*, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, vol. 2, no. 7, 2015.

[90] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Unsupervised domain adaptation with residual transfer networks," *Advances in neural information processing systems*, vol. 29, 2016.

[91] M. Ghifary, W. B. Kleijn, and M. Zhang, "Domain adaptive neural networks for object recognition," in *Pacific Rim international conference on artificial intelligence*. Springer, 2014, pp. 898–904.

[92] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," *arXiv preprint arXiv:1412.3474*, 2014.

[93] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *International conference on machine learning*. PMLR, 2015, pp. 97–105.

[94] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *International conference on machine learning*. PMLR, 2017, pp. 2208–2217.

[95] E. Tzeng, C. Devin, J. Hoffman, C. Finn, P. Abbeel, S. Levine, K. Saenko, and T. Darrell, "Adapting deep visuomotor representations with weak pairwise constraints," in *Algorithmic Foundations of Robotics XII*. Springer, 2020, pp. 688–703.

[96] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[97] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan, "Unsupervised pixel-level domain adaptation with generative adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3722–3731.

[98] M.-Y. Liu and O. Tuzel, "Coupled generative adversarial networks," *Advances in neural information processing systems*, vol. 29, 2016.

[99] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[100] C. Fan, P. Li, T. Xiao, W. Zhao, and X. Tang, "A review of deep domain adaptation: general situation and complex situation," *Acta Automatica Sinica*, vol. 46, no. 3, pp. 515–548, 2020.

[101] R. Volpi, P. Morerio, S. Savarese, and V. Murino, "Adversarial feature augmentation for unsupervised domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5495–5504.

[102] M. Long, Z. CAO, J. Wang, and M. I. Jordan, "Conditional adversarial domain adaptation," in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31. Curran Associates, Inc., 2018. [Online]. Available: https://proceedings.neurips.cc/paper/2018/file/ab88b15733f543179858600245108dd8-Paper.pdf

[103] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[104] Z. Yi, H. Zhang, P. Tan, and M. Gong, "Dualgan: Unsupervised dual learning for image-to-image translation," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2849–2857.

[105] M. Ghifary, W. B. Kleijn, M. Zhang, and D. Balduzzi, "Domain generalization for object recognition with multi-task autoencoders," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2551–2559.

[106] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li, "Deep reconstruction-classification networks for unsupervised domain adaptation," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 597–613.

[107] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan, "Domain separation networks," *Advances in neural information processing systems*, vol. 29, 2016.

[108] Y. Bengio, *Learning deep architectures for AI*. Now Publishers Inc, 2009.

[109] X. Glorot, A. Bordes, and Y. Bengio, "Domain adaptation for large-scale sentiment classification: A deep learning approach," in *ICML*, 2011.

[110] M. Chen, Z. Xu, K. Weinberger, and F. Sha, "Marginalized denoising autoencoders for domain adaptation," *arXiv preprint arXiv:1206.4683*, 2012.

[111] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big data*, vol. 3, no. 1, pp. 1–40, 2016.

[112] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Machine learning*, vol. 79, no. 1, pp. 151–175, 2010.

[113] G. Chaojun, P. Jirutitijaroen, and M. Motani, "Detecting false data injection attacks in ac state estimation," *IEEE Transactions on Smart Grid*, vol. 6, no. 5, pp. 2476–2483, 2015.

[114] S. Pal, B. Sikdar, and J. Chow, "Detecting data integrity attacks on scada systems using limited pmus," in *2016 IEEE International Conference on Smart Grid Communications (SmartGridComm)*. IEEE, 2016, pp. 545–550.

[115] S. Gupta, S. Waghmare, F. Kazi, S. Wagh, and N. Singh, "Blackout risk analysis in smart grid wampac system using kl divergence approach," in *2016 IEEE 6th International Conference on Power Systems (ICPS)*. IEEE, 2016, pp. 1–6.

[116] H. Elsahar and M. Gallé, "To annotate or not? predicting performance drop under domain shift," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 2163–2173.

[117] W. Deng and L. Zheng, "Are labels always necessary for classifier accuracy evaluation?" in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15 069–15 078.

[118] A. Ramesh Kashyap, D. Hazarika, M.-Y. Kan, and R. Zimmermann, "Domain divergences: A survey and empirical analysis," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, jun 2021, pp. 1830–1849. [Online]. Available: https://aclanthology.org/2021.naacl-main.147

[119] S. Ruder and B. Plank, "Learning to select data for transfer learning with Bayesian optimization," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 372–382. [Online]. Available: https://aclanthology.org/D17-1038

[120] R. Wang, A. Finch, M. Utiyama, and E. Sumita, "Sentence embedding for neural machine translation domain adaptation," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2017, pp. 560–566.

[121] C. C. Aggarwal, A. Hinneburg, and D. A. Keim, "On the surprising behavior of distance metrics in high dimensional space," in *International conference on database theory*. Springer, 2001, pp. 420–434.

[122] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When is "nearest neighbor" meaningful?" in *International conference on database theory*. Springer, 1999, pp. 217–235.

[123] V. Spruyt, "The curse of dimensionality in classification," *Computer vision for dummies*, vol. 21, no. 3, pp. 35–40, 2014.

[124] Y.-B. Kim, K. Stratos, and D. Kim, "Adversarial adaptation of synthetic or stale data," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, Jul. 2017, pp. 1297–1307. [Online]. Available: https://aclanthology.org/P17-1119

[125] V. Van Asch and W. Daelemans, "Using domain similarity for performance estimation," in *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*. Uppsala, Sweden: Association for Computational Linguistics, Jul. 2010, pp. 31–36. [Online]. Available: https://aclanthology.org/W10-2605

[126] R. Remus, "Domain adaptation using domain similarity- and domain complexity-based instance selection for cross-domain sentiment analysis," in *2012 IEEE 12th International Conference on Data Mining Workshops*, 2012, pp. 717–723.

[127] K. Duh, G. Neubig, K. Sudoh, and H. Tsukada, "Adaptation data selection using neural language models: Experiments in machine translation," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Sofia, Bulgaria: Association for Computational Linguistics, aug 2013, pp. 678–683. [Online]. Available: https://aclanthology.org/P13-2119

[128] Z. Wang, Y. Qu, L. Chen, J. Shen, W. Zhang, S. Zhang, Y. Gao, G. Gu, K. Chen, and Y. Yu, "Label-aware double transfer learning for cross-specialty medical named entity recognition," *arXiv preprint arXiv:1804.09021*, 2018.

[129] W. Zellinger, T. Grubinger, E. Lughofer, T. Natschläger, and S. Saminger-Platz, "Central moment discrepancy (CMD) for domain-invariant representation learning," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. [Online]. Available: https://openreview.net/forum?id=SkB-_mcel

[130] M. Tajdinian and H. Samet, "Divergence distance based index for discriminating inrush and internal fault currents in power transformers," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 5, pp. 5287–5294, 2021.

[131] Z. Taghiyarrenani, A. Fanian, E. Mahdavi, A. Mirzaei, and H. Farsi, "Transfer learning based intrusion detection," in *2018 8th International Conference on Computer and Knowledge Engineering (ICCKE)*, Oct. 2018, pp. 92–97.

[132] M. Cui, J. Wang, and B. Chen, "Flexible machine learning-based cyberattack detection using spatiotemporal patterns for distribution systems," *IEEE Transactions on Smart Grid*, vol. 11, no. 2, pp. 1805–1808, 2020.

[133] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," *ACM Trans. Inf. Syst. Secur.*, vol. 14, no. 1, pp. 13:1–13:33, jun 2011.

[134] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" *Advances in neural information processing systems*, vol. 27, 2014.

[135] Y. Xu, Z. Liu, Y. Li, Y. Zheng, H. Hou, M. Gao, Y. Song, and Y. Xin, "Intrusion detection based on fusing deep neural networks and transfer learning," in *International Forum on Digital TV and Wireless Multimedia Communications*. Springer, 2019, pp. 212–223.

[136] Z. Yao, Y. Wang, M. Long, and J. Wang, "Unsupervised transfer learning for spatiotemporal predictive networks," in *International Conference on Machine Learning*. PMLR, 2020, pp. 10 778–10 788.

[137] S. Tariq, S. Lee, and S. S. Woo, "Cantransfer: Transfer learning based intrusion detection on a controller area network using convolutional lstm network," in *Proceedings of the 35th annual ACM symposium on applied computing*, 2020, pp. 1048–1055.

[138] S. Kullback, *Information theory and statistics*. Courier Corporation, 1997.

[139] A. R. Kashyap, D. Hazarika, M.-Y. Kan, and R. Zimmermann, "Domain divergences: a survey and empirical analysis," *arXiv preprint arXiv:2010.12198*, 2020.

[140] J. R. Hershey and P. A. Olsen, "Approximating the kullback leibler divergence between gaussian mixture models," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, vol. 4, 2007, pp. IV–317–IV–320.

[141] K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Schölkopf, and A. J. Smola, "Integrating structured biological data by kernel maximum mean discrepancy," *Bioinformatics*, vol. 22, no. 14, pp. e49–e57, 2006.

[142] K. Pearson, "Note on regression and inheritance in the case of two parents," *Proceedings of the Royal Society of London*, vol. 58, pp. 240–242, 1895. [Online]. Available: http://www.jstor.org/stable/115794

[143] H. Akoglu, "User's guide to correlation coefficients," *Turkish journal of emergency medicine*, vol. 18, no. 3, pp. 91–93, 2018.

[144] C. P. Dancey and J. Reidy, *Statistics without maths for psychology*. Pearson education, 2007.

[145] M. H. Hassan, S. Kamel, M. A. El-Dabah, T. Khurshaid, and J. L. Domínguez-García, "Optimal reactive power dispatch with time-varying demand and renewable energy uncertainty using rao-3 algorithm," *IEEE Access*, vol. 9, pp. 23 264–23 283, 2021.

[146] M. Rahman, Y. Li, and J. Yan, "Multi-objective evolutionary optimization for worst-case analysis of false data injection attacks in the smart grid," in *2020 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2020, pp. 1–8.

[147] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[148] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[149] B. Plank and G. van Noord, "Effective measures of domain similarity for parsing," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, June 2011, pp. 1566–1576. [Online]. Available: https://aclanthology.org/P11-1157

[150] P. Liao, J. Yan, J. M. Sellier, and Y. Zhang, "Divergence-based transferability analysis for self-adaptive smart grid intrusion detection with transfer learning," *IEEE Access*, vol. 10, pp. 68 807–68 818, 2022.

[151] W.-C. Hong, D.-R. Huang, C.-L. Chen, and J.-S. Lee, "Towards accurate and efficient classification of power system contingencies and cyber-attacks using recurrent neural networks," *IEEE Access*, vol. 8, pp. 123 297–123 309, 2020.

[152] D. Zhang, L. Yao, X. Zhang, S. Wang, W. Chen, and R. Boots, "Eeg-based intention recognition from spatio-temporal representations via cascade and parallel convolutional recurrent neural networks," *arXiv preprint arXiv:1708.06578*, pp. 1–8, 2017.

[153] Z. Wang, W. Yan, and T. Oates, "Time series classification from scratch with deep neural networks: A strong baseline," in *2017 International joint conference on neural networks (IJCNN)*. IEEE, 2017, pp. 1578–1585.

[154] L. Wei, D. Gao, and C. Luo, "False data injection attacks detection with deep belief networks in smart grid," in *2018 Chinese Automation Congress (CAC)*. IEEE, 2018, pp. 2621–2625.

[155] G. Wilson, J. R. Doppa, and D. J. Cook, "Multi-source deep domain adaptation with weak supervision for time-series sensor data," *arXiv preprint arXiv:2005.10996*, 2020.

[156] L. Ge, Y. Li, Y. Li, J. Yan, and Y. Sun, "Smart distribution network situation awareness for high-quality operation and maintenance: A brief review," *Energies*, vol. 15, no. 3, p. 828, 2022.

[157] Y. Li and J. Yan, "Cybersecurity of smart inverters in the smart grid: A survey," *IEEE Transactions on Power Electronics*, pp. 1–20, 2022.

[158] K. Shaukat, S. Luo, V. Varadharajan, I. A. Hameed, S. Chen, D. Liu, and J. Li, "Performance comparison and current challenges of using machine learning techniques in cybersecurity," *Energies*, vol. 13, no. 10, p. 2509, 2020.

[159] A. Kumar, N. Saxena, S. Jung, and B. J. Choi, "Improving detection of false data injection attacks using machine learning with feature selection and oversampling," *Energies*, vol. 15, no. 1, p. 212, 2021.

[160] R. Deng, P. Zhuang, and H. Liang, "Ccpa: Coordinated cyber-physical attacks and countermeasures in smart grid," *IEEE Transactions on Smart Grid*, vol. 8, no. 5, pp. 2420–2430, 2017.

[161] S. Liu, B. Chen, T. Zourntos, D. Kundur, and K. Butler-Purry, "A coordinated multi-switch attack for cascading failures in smart grid," *IEEE Transactions on Smart Grid*, vol. 5, no. 3, pp. 1183–1195, 2014.

[162] Q. Wang, J. Huang, and F. Lu, "An improved particle filtering algorithm for aircraft engine gas-path fault diagnosis," *Advances in Mechanical Engineering*, vol. 8, no. 7, p. 1687814016659602, 2016.

[163] P. Dewallef and O. Le´ onard, "On-line performance monitoring and engine diagnostic using robust kalman filtering techniques," in *Turbo Expo: Power for Land, Sea, and Air*, vol. 36843, 2003, pp. 395–403.

[164] F. Lu, T. Gao, J. Huang, and X. Qiu, "Nonlinear kalman filters for aircraft engine gas path health estimation with measurement uncertainty," *Aerospace Science and Technology*, vol. 76, pp. 126–140, 2018.

[165] M. Ahmed and A. S. Barkat Ullah, "False data injection attacks in healthcare," in *australasian conference on data mining*. Springer, 2017, pp. 192–202.

[166] E. Johns, "Cyber security breaches survey 2020," *London: Department for Digital, Culture, Media & Sport*, 2020.

[167] M. Ahmed and A.-S. K. Pathan, "False data injection attack (fdia): an overview and new metrics for fair evaluation of its countermeasure," *Complex Adaptive Systems Modeling*, vol. 8, no. 1, pp. 1–14, 2020.