Semi-Supervised Domain-Adversarial Training for Intrusion Detection against False Data Injection in the Smart Grid

Yongxuan Zhang* and Jun Yan†
*Department of Computer Science and Software Engineering (CSSE), Concordia University
†Concordia Institute for Information Systems Engineering (CIISE), Concordia University
Montréal, Québec, H3G 1M8, Canada
z_yongxu@encs.concordia.ca; jun.yan@concordia.ca

***Abstract*—The smart grid faces with increasingly sophisticated cyber-physical threats, against which machine learning (ML)-based intrusion detection systems have become a powerful and promising solution to smart grid security monitoring. However, many ML algorithms presume that training and testing data follow the same or similar data distributions, which may not hold in the dynamic time-varying systems like the smart grid. As operating points may change dramatically over time, the resulting data distribution shifts could lead to degraded detection performance and delayed incidence responses. To address this challenge, this paper proposes a semi-supervised framework based on domain-adversarial training to transfer the knowledge of known attack incidences to detect returning threats at different hours and load patterns. Using normal operation data of the ISO New England grids, the proposed framework leverages adversarial training to adapt learned models against new attacks launched at different times of the day. Effectiveness of the proposed detection framework is evaluated against the well-studied false data injection attacks synthesized on the IEEE 30-bus system, and the results demonstrated the superiority of the framework against persistent threats recurring in the highly dynamic smart grid.**

***Index Terms*—Adversarial training, false data injection, intrusion detection, smart grid security, transfer learning, domain adaptation.**

I. Introduction

The smart grid is connecting utilities and customers over two-way, high-speed, and frequently machine-to-machine communications. Thanks to the cyber-physical integration, next-generation power and energy systems will empower modern society with more efficient, resilient, and sustainable electricity. However, the growing interconnection among billions of interacting systems, devices, and processes creates complex interdependence and vulnerabilities that will inevitably expose to cyber-attacker in the wild. The threat of a cyber-attack could be both sophisticated and disastrous, as demonstrated by recent research efforts [1]–[3], business studies [4], and real-world incidences [5].

The rapid progress in machine learning (ML) has revealed the latter's strength in handling voluminous data streams, extracting informative features and tackling system complexities. A rich line of ML approaches has significantly energized research in the field of smart grid, in particular to enhance its cyber-physical security monitoring and situational awareness capacity [6]–[8].

Many machine learning approaches presume that training and testing data will share the same feature sets and follow the same or similar distributions [7]. However, in the power systems, labeled attack data is often limited and data distribution shift [9] may occur when loads, topology, or other system dynamics change. They can lead to bias in the training data and render the machine learning model intractable. This creates a strong incentive for effective algorithms that can close up the gap for robust and adaptive intrusion detection against advanced persistent threats.

A potential solution to the challenge is transfer learning, which has been proposed to consolidate knowledge learned from previous domains and tasks (the source) for a new related domain and/or task (the target). It has been widely adopted in various image/video applications [10] as a promising solution to transfer a learned model for new domains or tasks. Recently, researchers have also started to introduce transfer learning for anomaly detection in Internet [11] and cloud [12] applications, which demonstrated strong potential for cyber-security monitoring in dynamic power systems and environments [13].

Motivated by the remaining gaps, this paper proposes a Semi-Supervised Domain-Adversarial Training (SSDAT) framework based on the latest domain-adversarial learning technique [14], which extracts novel features to unify data distributions across two domains and improves classifier robustness against the shift. We tackle the challenge where the data of the attack incidence is rare compared to normal operations and address it by semi-supervised transfer learning from a single-class target domain where the attack incidence is absent.

For the threat model, we consider the stealthy false data injection (FDI) attack [15], which exploits the mathematical model and topological information to inject false measurements and bypass the traditional residual-based bad data detection. Existing work has proved the power of machine learning approaches on FDI detection. Ozay *et al.* proposed to utilize kNN and SVM to detect FDI attack built in measurement space [7].

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

To evaluate shifts in realistic cases, we take real-time load demand from ISO New England to synthesize datasets with different changing distributions and evaluate the performance of the proposed framework over other baseline machine learning algorithms. The results have demonstrated that the effectiveness of the proposed semi-supervised domain-adversarial training framework against the data distribution shift, particularly when the attack sample is rare due to its limited presence.

The rest of this paper is organized as follows: Section II discusses the related work about domain adaptation and transfer learning in cyber-secrity. Section III presents the proposed framework for smart grid intrusion detection. Section IV reviews the FDI attack model. Section V presents the experiment setup and the simulation results. Section VI draws the conclusions and future works.

II. Related Works

A. Domain Adaptation

Domain adaptation is a transductive transfer learning technique, where the marginal probability distributions of input data from source and target domains are different while the feature spaces are the same [16]. Existing domain adaptation approaches can be categorized into unsupervised [14], [17], [18], semi-supervised [19]–[21], and supervised domain adaptation [22]–[24].

Unsupervised domain adaptation leverages labeled source domain data and unlabeled target domain data to decrease domain discrepancy or find domain invariant representations. Thus models trained on the source domain can generalize well on the target domain. Fang *et al.* [17] proposed to reduce distribution discrepancy and increases inter-class margins via Sphere Retracting Transformation. Ganin *et al.* [14] introduced adversarial training into domain adaptation to find domain invariant features. A domain classifier is utilized to distinguish data from two domains. A feature extractor is trained to confuse the domain classifier by reversing the gradient from the domain classifier. Li *et al.* [18] proposed category transfer to improve the adversarial domain adaptation. It iteratively estimates and minimizes Wasserstein distance between categories in multi-category structures to avoid negative transfer between different category data from source and target.

Semi-supervised domain adaptation assumes that limited labeled target data is available and can be used to support domain adaptation during the training stage. Wang *et al.* [21] proposed a transfer fredholm multiple kernel learning approach. Fredholm integrals from two domains are calculated by labeled data to learn a kernel predictive model across two domains. Pereira *et al.* [20] made use of labeled data from two domains to minimize squared induced distance between instances from different classes and different domains and maximize squared induced distance between instances from different classes and different domains. Similarly Li *et al.* [19] considered to highlight the discrimination information of the labelled samples, which takes into account the class connections between source samples and target samples.

Our SSDAT framework belongs to semi-supervised domain adaptation but differs from traditional works in the sense that we assume only normal data in target domain is available for the attack detection. Then normal data from two domains are leveraged to reduce domain discrepancy. Detailed implementations will be discussed in section III.

B. Transfer Learning in Cyber-Security

Researchers have recently started to introduce transfer learning for anomaly detection in cyber-security. Bartos *et al.* [25] computed a self-similarity measure of the network traffic logs for the domain adaptation problems with conditional shift in network security. Juan *et al.* [26] proposed a feature-based transfer learning framework that was able to boost classifier performance on the well-known NSL-KDD dataset of TCP traffic. D. Nahmias *et al.* [27] applied feature transfer learning from pre-trained VGG19 neural network mode on malware detection. Inspired by the recent research, we leveraged domain-adversarial training to detect unknown threats in previous work [13]. In this paper, we make further contributions in mainly three aspects:

- We consider the situation where attack may happen at different time during the power system operation, which may fail the trained model when data distribution changes accordingly.
- We formulate a transfer learning problem and define the labeled training data as source domain, labeled normal data with similar data distribution as target domain. We address this problem by introducing a semi-supervised domain-adversarial training framework.
- We set up different cases regarding the trends of power demand in source domain as well as different time windows in target domain and evaluate the performance of the proposed framework with baseline models on balanced and imbalanced cases.

III. Domain-Adversarial Transfer Learning Against Data Distribution Shift

A. Problem Formulation

In this paper, we focus on a scenario where two consecutive attacks targeted the same grid during different periods when load demands have changed. In transfer learning, this suggests a data distribution shift, where the distribution $P(X)$ of inputs (samples) has changed but the conditional distribution $P(Y|X)$ of outputs (labels) remains the same.

To elaborate how transfer learning tackles the data distribution shifts, we need to first define several concepts. First, a domain $\mathcal{D}$ consists of two components: a feature space $\mathcal{X}$ and a marginal probability distribution *P(X)*. Given a specific domain $\mathcal{D}$= $\{\mathcal{X}$,*P(X)*$\}$, a task consists of two components: a label space $\mathcal{Y}$ and an objective predictive function $f(\cdot)$ from $\mathcal{X}$ to $\mathcal{Y}$ that will be learned from the training data.

In this work we consider the binary intrusion detection, which classifies the data as attack events or normal operations. With the shift, the source domain $\mathcal{D}_S$ and target domain $\mathcal{D}_T$ will have the same feature space $\mathcal{X}$ but different data distributions *P(X)*. We denote the labeled source domain as $\mathcal{D}_S =$

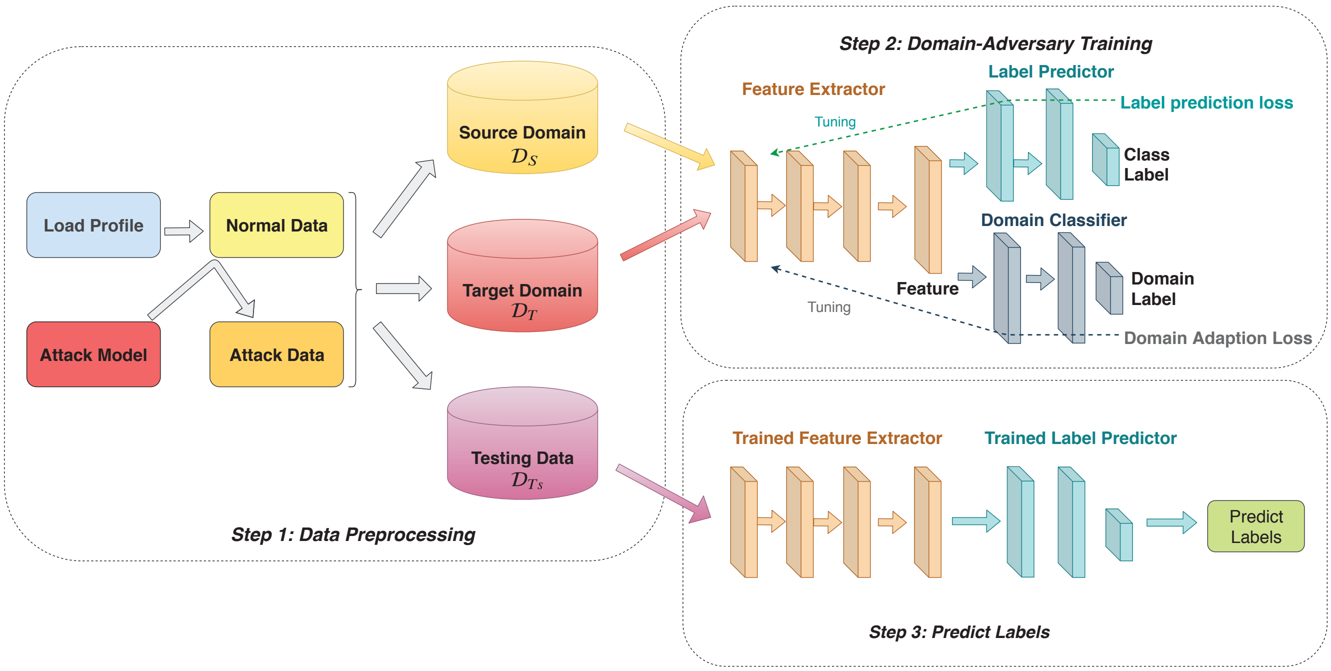


Fig. 1. The proposed semi-supervised domain-adversarial transfer learning framework based on [14] for self-adaptive smart grid intrusion detection.

$\{(x_{S_1}, y_{S_1}), ..., (x_{S_{n_S}}, y_{S_{n_S}})\}$, the labeled target domain as $\mathcal{D}_T = \{(x_{T_1}, y_{T_1}), ..., (x_{T_{n_T}}, y_{T_{n_T}})\}$, and the unlabeled testing dataset as $\mathcal{D}_{T_s} = \{(x_{T_{s_1}}, y_{T_{s_1}}), ..., (x_{T_{s_n}}, y_{T_{s_n}})\}$.

The goal of transfer learning is to learn the predictive function $f(\cdot)$ from the training data to predict labels in the target and testing data using the mapping to a new feature space, where the inter-class distance and inner-class similarity are both retained. The new feature space maps the shifting data distributions into a single one, where machine learning algorithms can be trained for better classification. The main challenge is how to find the best-performing mapping, which is tackled by the domain-adversarial training as follows.

B. Domain-Adversarial Transfer Learning for Neural Networks

The original domain-adversarial transfer learning [14] consists of three neural networks: a feature extractor, a domain classifier, and a label predictor. Data from source and target domains will be fed into the feature extractor and mapped into a same feature space.

The domain classifier is the "domain adversary", which is being trained to tell the difference of data from the source and the target domains and so as to "fail" the feature extractor. The label predictor is trained to determine the class label, which can use same soft-max function [28]

Given a sample-label pair $(\mathbf{x}, y)$, the loss function of the binary classification for label predictor is given as $L_y(\mathbf{x}, y)$. The domain adaption loss L_d is given as $L_d(\mathbf{x}, d_{\mathbf{x}})$, where $d_{\mathbf{x}} = 0$ if $x \in \mathcal{D}_S$ and $d_{\mathbf{x}} = 1$ if $x \in \mathcal{D}_T$.

Finally, the objective of domain-adversarial training can be then formulated as:

$$\begin{aligned}\underset{\mathbf{W},\mathbf{V},\mathbf{b},\mathbf{c},\mathbf{u},z}{min} \Bigg[\frac{1}{n_S} \sum_{\mathbf{x}\in\mathcal{D}_S} L_y(\mathbf{x}, y) - \frac{\lambda}{n_S} \sum_{\mathbf{x}\in\mathcal{D}_S} L_d(\mathbf{x}, d_{\mathbf{x}}) \\ - \frac{\lambda}{n_T} \sum_{\mathbf{x}\in\mathcal{D}_T} L_d(\mathbf{x}, d_{\mathbf{x}}) \Bigg]\end{aligned} \tag{1}$$

Based on (1), the overall loss from both label predictor and domain classifier will be back-propagated [29] to adjust the weights of the three neural networks.

C. Semi-Supervised Domain-Adversarial Training

Based on the architecture, the overall framework proposed in this paper is illustrated in Fig. 1. It consists of three steps: (1) given the rarity of labeled attack, we first collect labeled normal and attack data in the *source domain*, the labeled normal target data in the *target domain*, and the unlabeled target data in the *testing dataset*; (2) train the three networks with the source and target domains to obtain a new feature space via domain-adversarial training, which adapts the normal data over two domains; (3) map the testing dataset to the new feature space and predict their class labels.

The design is based on the consideration that in the context of cyber-physical security monitoring, labeled attack data can be extremely rare compared to the labeled normal data that are constantly sampled. We will have to face a rarity of positive class distribution problem in the target domain.

To address this challenge, after the feature extraction, the normal data from source domain and target domain will be fed into the domain classifier. The gradient from domain loss will be reversed when in the feature extractor to increase the domain similarity so that the data distribution shift can be mitigated. Both normal and attack data will be used for training the label predictor.

Then we follow the same mechanism in [13], we select the random forest classifier [30] as the new label predictor in SSDAT framework, which will be trained on the embeddings obtained from the labeled source data through the feature extractor after the domain-adversarial training.

IV. Attack Model

Over the last two decades, various attack models have been developed to analyze and enhance the cyber-physical security of smart grid [3]. Among them, the false data injection (FDI) attack [15] stands out as one of the most studied threat models. As FDI attacks exploit a mathematical vulnerability in the residual-based bad data detector (BDD) to inject false data onto measurements without raising alarms, it posed a severe threat to power system state estimators (PSSE) and the energy management systems. Successful FDI attacks can introduce arbitrary errors into certain state variables and cause the system operator to perform misinformed control actions, which may result in physical damage and monetary loss [31]. Specifically, the FDI targets the DC state estimation is defined as [32]:

$$\mathbf{z} = \mathbf{Hx} + \mathbf{n} \tag{2}$$

where $\mathbf{z}$ is the known measurement, $\mathbf{H}$ is the Jacobian matrix of power grid topology and $\mathbf{x}$ is the unknown state variable. To identify corrupted measurements, the BDD utilizes statistical tests based on the residual between observed and estimated measurements: $\mathbf{r} = \mathbf{z} - \mathbf{H\hat{x}}$, where $\hat{x}$ is the estimated state variable solved by the weighted least square method. The normalized L_2-norm of $\mathbf{r}$ is then compared with a preset threshold τ to detect the bad data.

The FDI attack model can be written in the following form:

$$\mathbf{z_a} = \mathbf{z} + \mathbf{a} = \mathbf{Hx} + \mathbf{n} + \mathbf{a} \tag{3}$$

where $\mathbf{a}$ is the injected attack vector and $\mathbf{z_a}$ is the manipulated measurements. In the stealthy FDI attack, we assume the attacker has the knowledge of $\mathbf{H}$. In order to bypass the bad data detection and manipulate the states of buses, a targeted false state $\mathbf{x_a}$ is generated by $\mathbf{x_a} = \mathbf{x} + \mathbf{c}$, where $\mathbf{c} \sim N(0, \sigma_c^2)$ is the false state error injected into the system. The attack vector $\mathbf{a}$ is computed by $\mathbf{a} = \mathbf{Hc}$ and injected into the measurements $\mathbf{z}$ by $\mathbf{z_a} = \mathbf{z} + \mathbf{a}$. Let $\mathbf{r} = \mathbf{z} - \mathbf{Hx}$ be the remain residual for bad data detection. Then the new residual $\mathbf{r_a}$ will remain the same and bypass the residual-based bad data detection:

$$\begin{aligned} \mathbf{r_a} = \mathbf{z_a} - \mathbf{Hx_a} &= \mathbf{z} + \mathbf{a} - \mathbf{H}(\mathbf{x} + \mathbf{c}) \\ &= (\mathbf{z} - \mathbf{Hx}) + (\mathbf{a} - \mathbf{HC}) \\ &= \mathbf{z} - \mathbf{Hx} \end{aligned} \tag{4}$$

The pre-attack measurements are obtained from realistic load demands over multiple consecutive days, which represents the challenging data distribution shifts to classic machine learning and calls for transfer learning against new attacks occurring at different hours of the day with different load demands.

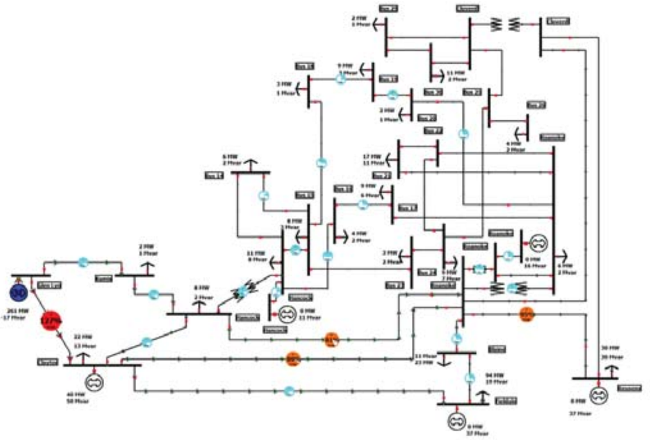


Fig. 2. The IEEE 30-bus system by the Illinois Center for a Smarter Electric Grid (ICSEG) [33].

V. Experiments and Results

A. Normal Data Simulation

We chose the IEEE 30-bus system [33] for simulation and evaluation of the performance, whose topology is illustrated in Fig. 2. There are 30 buses and 41 branches with a total load demand of 189.2 MW. A total 142 measurements are used to estimate 30 state variables under DC model.

To set up realistic load variation on this static benchmark, we obtained public data from ISO New England [34] and synthesized the normal operating points (OPs) over a week. We selected one week demand from August 24 to 30, 2019, as shown in Fig. 3, to synthesize a typical weekly load curve for the IEEE 30-bus system. The demand was reported every 5 minutes, or 288 samples per day. By assuming the default load of the 30-bus system the peak load of the week (100%), we calculated all OPs over the 5-minute intervals using the DC optimal power flow (DC-OPF) solver in MATPOWER to collect the normal measurement data of the system.

As illustrated in Table I, we followed the load variations at different hours of the day to create the source and target domains based on 4-hour time windows to best capture different patterns of data distribution in the 30-bus system. We assumed that the attack was launched on Day 0, and normal operations have been resumed on Day 1. Without loss of generality, we assumed that by Day 5 the data recording the attack period on Day 0 have been collected; the data recording the same period of normal operation on Day 1 have also been collected to form the labelled training set for the source domain. Then we assumed that day-to-day domain adaptation is performed until Day 6 when the attack was launched again but possibly at different hours. The target domain thus contains data recorded at corresponding periods on Day 5 when the last domain adaptation was performed.

B. Attack Data Generation

For the attack, We assumed that the attacker aims to manipulate the states with the least efforts. Since the number of compromised meters to manipulate the states varies between buses and depends on the topology $\mathbf{H}$. We searched the number of compromised measurements when attacking a single bus and identified three buses (Buses 11, 13, and 26) that require the minimal number of compromised measurements. The attack vectors were then generated by the FDI attack model $\mathbf{a} = \mathbf{Hc}$ and injected into the measurements $\mathbf{z}$. The false state $\mathbf{c}$ was set with a zero mean and a variance of $\sigma_c^2 = 0.1$.

C. Data set setup

1) Balanced Case: Considering that attacks may happen at a different time of the day when the load patterns can be distinctive, we defined 4 cases according to the variation of load demand: the valley, the ascending slope, the peak, and the descending slope. In each case, we assumed that the attack lasted for 4 hours on Day 0 before the system is restored. Once the attack period is located, we also extracted 4 hours of normal operation data on Day 1, recorded during the same 4-hour periods as the attack on Day 0, to create a balanced binary classification dataset in the source domain for SSDAT.

For the target domain and testing dataset, we also used the 4-hour time window but divided Days 5 and 6 into six intervals. On Day 5, we have only recorded normal operations, which contain natural data distribution shifts from load variation. The normal data from the target domain will be used for domain adaptation in SSDAT. For the fair comparison, target domain data will also be treated as labeled training data for baseline classifiers. On Day 6, we assumed that the attack last for 2 hours, which starts at the beginning of one of the six intervals, and the testing dataset is thus also a balanced set composed of 2 hours of attack data followed by 2 hours of normal data.

2) Imbalanced Case: Based on case 1 we further investigate imbalanced cases to explore the performance when the attack data has different proportions in testing data. We choose the same source domain and 4-hour time windows as the target domain. For the testing dataset, we create cases by adjusting the percentage of 4-hour time window attack data as 25% and 75% separately. We shift the one attack hour for 25% cases and one normal hour for 75% cases to generate 4 sub-cases for each time window. In this paper, We use the F_1 score to measure the accuracy of imbalanced cases [35].

To compare classic machine learning classifiers with the SSDAT framework, we have chosen four baseline classifiers: Artificial Neural Network (ANN) [36], Support Vector Machine (SVM) [37], Classification and Regression Tree (CART) [38], and Random Forest (RF) [30]. All classifiers are implemented in Scikit-learn [39] with manually optimized parameters releasable upon request.

D. Simulation Results

The detection accuracy for balanced cases of all classifiers over the 4 cases is shown in Table III. Overall, the framework shows a robust performance in most of the cases and better than other baseline classifiers in 19 of the 24 sub-cases. The best-case improvement reaches +36.0% compared to ANN during Hours 17–20 in Case 2. The results suggested that SSDAT can retain the accuracy when the same attack occurs at different hours while the baseline classifiers fail to adapt.

TABLE I
CASE 1 SETUP

#	Source Domain on Day 0 (attack) and Day 1 (normal)		Target Domain on Day 5	Testing Dataset on Day 6
	Cases	Hours		
1	Valley	2–5	4-hour windows of normal operations at different time of the day.	2 hours of attack followed by 2 hours of normal operations.
2	Ascending	11–14		
3	Peak	17–20		
4	Descending	21–24		

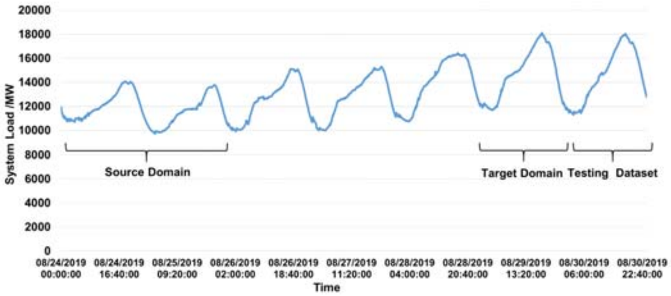


Fig. 3. Load demand of ISO New England between Aug. 24 to 30, 2019 [34], which is scaled down to map to the IEEE 30-bus system as the load curve.

It is notable that during Hours 1–4 and 5–8 on Day 5, the performance of some baseline classifiers are better than the SSDAT. The reason is that most of the load demand of Days 0 and 1 in the source domain has a significant overlap with the Hours 1–4 and 5–8 on Days 5 and 6. The overlapping demand suggests limited distribution shifts, which contributes to the performance of baseline classifiers. Outside of these hours, however, SSDAT achieves better accuracy. The observation poses an interesting question on when the adaptation is indeed needed and how to identify such moments.

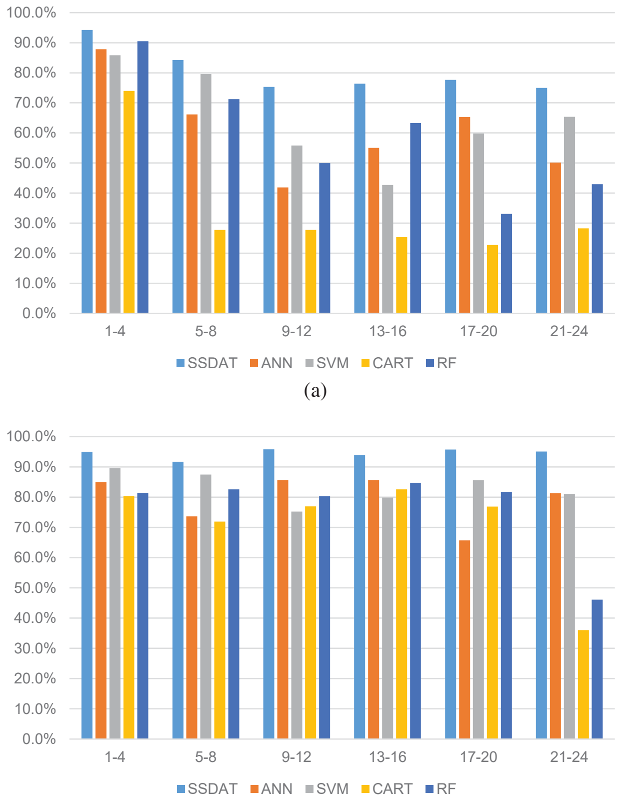
The averaged F_1 scores over 4 sub-cases of imbalanced cases with “valley” as source domain are illustrated in Fig. 4. The results suggest that when the source domain is ”Valley” data, SSDAT can outperform other baseline classifiers among 6 time windows. Especially when there is limited attack data, the SSDAT demonstrates significant improvements. The reason is that the source valley data has no overlap with the target domain. For other source domains not presented, there are still some cases in Hours 1–4 and 5–8 when some baseline classifiers achieve better performance. The reason is consistent with balanced cases.

TABLE II
CASE 2 SETUP

#	Source Domain on Day 0 (attack) and Day 1 (normal)		Target Domain on Day 5	Testing Dataset on Day 6
	Cases	Hours		
1	Valley	2–5	4-hour windows of normal operations at different time of the day.	1 hour attack as 25% cases and 3 hours attack as 75% cases.
2	Ascending	11–14		
3	Peak	17–20		
4	Descending	21–24		

TABLE III
COMPARISON OF DOMAIN-ADVERSARIAL AND MACHINE LEARNING CLASSIFIERS AGAINST RETURNING ATTACKS AT DIFFERENT HOURS

Cases	Source Hours	Target Hours	SSDAT	ANN	SVM	CART	RF	Best-Case Margin	Worst-Case Margin
1	2-5 (Valley)	1-4	**82.6%**	81.2%	81.3%	77.2%	81.9%	+5.4%	+0.7%
		5–8	**88.9%**	84.6%	87.5%	76.3%	81.0%	+12.6%	+1.4%
		9–12	**86.6%**	84.7%	81.3%	71.8%	76.2%	+14.8%	+1.9%
		13–16	**86.5%**	85.1%	72.9%	69.3%	72.0%	+17.2%	+1.4%
		17–20	**82.9%**	70.5%	64.6%	67.9%	66.0%	+18.3%	+12.4%
		21–24	**80.3%**	70.4%	68.8%	68.3%	69.0%	+12.0%	+9.9%
2	11-14 (Ascending)	1-4	95.3%	91.7%	79.2%	98.1%	**99.7%**	+16.1%	-4.4%
		5–8	**95.9%**	95.8%	87.5%	72.5%	87.3%	+23.4%	-0.1%
		9–12	**85.5%**	58.3%	81.3%	71.0%	79.0%	+27.2%	+4.2%
		13–16	**81.6%**	54.2%	70.8%	71.7%	78.1%	+27.4%	+3.5%
		17–20	**87.7%**	51.7%	70.8%	70.1%	74.7%	+36.0%	+13.0%
		21–24	**80.2%**	50.5%	68.8%	77.1%	79.8%	+29.6%	+0.3%
3	17-20 (Peak)	1-4	94.4%	93.7%	79.2%	91.2%	**95.4%**	+15.2%	-1.0%
		5–8	96.4%	91.7%	87.5%	97.2%	**97.7%**	+8.9%	-1.3%
		9–12	**85.3%**	75.6%	75.0%	66.8%	76.6%	+18.5%	+8.7%
		13–16	**91.1%**	70.7%	75.0%	66.9%	78.0%	+24.2%	+13.1%
		17–20	**85.1%**	68.9%	62.5%	62.9%	74.0%	+22.6%	+10.1%
		21–24	**94.3%**	74.4%	79.2%	83.0%	84.8%	+19.9%	+9.5%
4	21-24 (Descending)	1-4	94.9%	97.9%	85.4%	98.1%	**99.4%**	+9.5%	-4.5%
		5–8	96.9%	**100.0%**	91.7%	94.0%	96.5%	+5.2%	-3.1%
		9–12	**85.6%**	79.0%	60.4%	76.3%	83.4%	+25.2%	+2.2%
		13–16	**85.4%**	82.2%	60.4%	72.2%	83.4%	+25.0%	+2.0%
		17–20	**82.2%**	63.2%	58.3%	68.0%	78.6%	+23.9%	+3.6%
		21–24	**83.6%**	69.2%	56.3%	73.2%	80.8%	+27.3%	+2.8%



(a)

(b)

Fig. 4. F_1 scores ”Valley” with as the source domain (a) 25% attack data in testing and (b) 75% attack data in testing.

VI. CONCLUSIONS

This paper proposed a self-adaptive intrusion detection framework based on semi-supervised domain-adversarial training. The proposed framework is capable of mapping data shifting distributions into a unified feature space to improve attack detection performance under dynamic change load demands. The results have shown that the proposed framework can effectively tackle the rarity of attack samples and achieve robust performance against data distribution shifts than classic machine learning classifiers. In the future, we will further investigate when to transfer and how to retain the performance when no transfer is needed as our next research directions.

ACKNOWLEDGEMENT

This research is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) under grants RGPIN-2018-06724 and DGECR-2018-00022 and the Fonds de Recherche du Québec–Nature et Technologies (FRQNT) under grant 2019-NC-254971.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] “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.

[5] "Cyber-attack against ukrainian critical infrastructure," https://www.iso-ne.com/isoexpress/web/reports/load-and-demand/-/tree/dmnd-five-minute-sys, The Industrial Control Systems Cyber Emergency Response Team (ICS-CERT), Tech. Rep., 2019.

[6] 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.

[7] 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.

[8] 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.

[9] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. Lawrence, *Dataset Shift in Machine Learning*, 2009.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] Z. Fang, J. Lu, F. Liu, and G. Zhang, "Unsupervised domain adaptation with sphere retracting transformation," in *2019 International Joint Conference on Neural Networks (IJCNN)*, July 2019, pp. 1–8.

[18] L. Li, H. He, J. Li, and G. Yang, "Adversarial domain adaptation via category transfer," in *2019 International Joint Conference on Neural Networks (IJCNN)*, July 2019, pp. 1–8.

[19] L. Li and Z. Zhang, "Semi-supervised domain adaptation by covariance matching," *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[20] L. A. Pereira and R. da Silva Torres, "Semi-supervised transfer subspace for domain adaptation," *Pattern Recognition*, vol. 75, pp. 235–249, 2018.

[21] W. Wang, H. Wang, C. Zhang, and Y. Gao, "Fredholm multiple kernel learning for semi-supervised domain adaptation," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[22] A. Bergamo and L. Torresani, "Exploiting weakly-labeled web images to improve object classification: a domain adaptation approach," in *Advances in neural information processing systems*, 2010, pp. 181–189.

[23] L. Duan, D. Xu, and I. Tsang, "Learning with augmented features for heterogeneous domain adaptation," *arXiv preprint arXiv:1206.4660*, 2012.

[24] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *European conference on computer vision*. Springer, 2010, pp. 213–226.

[25] K. Bartos and M. Sofka, "Robust representation for domain adaptation in network security," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2015, pp. 116–132.

[26] J. Zhao, S. Shetty, and J. Pan, "Feature-based transfer learning for network security," in *2017 IEEE Military Communications Conference (MILCOM)*, Oct. 2017, pp. 17–22.

[27] D. Nahmias, A. Cohen, N. Nissim, and Y. Elovici, "Trustsign: Trusted malware signature generation in private clouds using deep feature transfer learning," in *2019 International Joint Conference on Neural Networks (IJCNN)*, July 2019, pp. 1–8.

[28] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[29] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1," D. Rumelhart, J. McClelland, and C. PDP Research Group, Eds. Cambridge, MA, USA: MIT Press, 1986, ch. Learning Internal Representations by Error Propagation, pp. 318–362.

[30] L. Brieman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[31] 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.

[32] A. Abur and A. Gomez-Exposito, *Power System State Estimation: Theory and Implementation*, 01 2004, vol. 24.

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

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

[35] Y. Sasaki, "The truth of the f-measure," *Teach Tutor Mater*, 01 2007.

[36] 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.

[37] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep 1995. [Online]. Available: https://doi.org/10.1023/A:1022627411411

[38] L. Breiman, *Classification and regression trees*. Routledge, 2017.

[39] 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.