ANOMALY DETECTION IN DIGITAL SUBSTATIONS USING SEMI-SUPERVISED LEARNING

Ho Trong Tai[1], Yong-Hwa Kim[1], Le Tien Dung[2]*

[1]Korea National University of Transportation Uiwang-si, Gyeonggi-do, South Korea
[2]The University of Danang - University of Science and Technology, Vietnam

*Corresponding author: ltdung@dut.udn.vn

(Received: April 15, 2025; Revised: June 08, 2025; Accepted: June 12, 2025)

DOI: 10.31130/ud-jst.2025.23(9C).522E

Abstract - The integration of Information and Communication Technology (ICT) into Operational Technology (OT) environments in modern substations has heightened cybersecurity risks. Effective Intrusion Detection Systems (IDS) are essential, yet current supervised learning methods struggle due to the scarcity and unreliability of labeled attack data. To address this, we propose a Semi-Supervised Anomaly Detection (SSAD) approach that leverages partially labeled datasets, where normal samples are labeled but anomalies are sparse or unlabeled. SSAD provides flexibility and efficiency in real-time detection, making it particularly suitable for digital substations and IEC104 network traffic. Our method enables reliable intrusion detection without requiring extensive labeled anomaly data, offering a practical solution to enhance both security and resilience.

Key words - Intrusion detection systems (IDS); semi supervised learning (SSL); anomaly detection; cyber-physical systems (CPS)

1. Introduction

The growing integration of Information and Communication Technology (ICT) and Cyber-Physical Systems (CPS) into the Operational Technology (OT) environments of modern digital substations has greatly enhanced monitoring and control capabilities. However, this integration also exposes substations to heightened cybersecurity risks. The close coupling between cyber and physical entities, alongside the use of Ethernet-based networks such as IEEE 802.3, renders substations vulnerable to cyber threats. Key communication protocols in Substation Automation Systems (SAS), including IEC 61850 (such as GOOSE, SV, and MMS) and IEC 60870-5-104 (IEC 104), are particularly vulnerable [1]. These protocols often handle real-time and sensitive information, and any security breach can compromise the Confidentiality, Integrity, and Availability (CIA) of data, potentially leading to operational failures or large-scale power outages. Therefore, deploying effective Intrusion Detection Systems (IDS) is critical for detecting and mitigating abnormal or malicious activities, ensuring the security of these vital power grid systems [2]. Traditional IDS methods often rely on statistical analysis, rule-based systems, or signature-based techniques to detect known attack patterns [3]. While these approaches are useful for general attack detection, they struggle with more sophisticated, unknown, or protocol-compliant attacks. Additionally, they tend to produce high false positive and false negative rates. For time-sensitive protocols such as GOOSE, many traditional IDS systems lack real-time detection capabilities, exacerbating the problem [4]. Machine Learning (ML) techniques-particularly Support Vector Machines (SVM), Random Forests, and Decision Trees-have been widely explored for anomaly detection in IDS [5]. These methods are typically supervised and require large, labeled datasets for training. While these ML models can be effective for identifying known attack patterns, they struggle to generalize to unseen threats and are highly dependent on the availability of comprehensive and reliable labeled data. Furthermore, these models can be vulnerable to adversarial attacks, reducing their robustness in dynamic environments. In contrast, Deep Learning (DL) models such as Deep Neural Networks (DNN) [4], [6] have shown considerable promise in IDS due to their ability to learn complex patterns from large datasets. Recent studies have demonstrated that DL-based IDS outperforms traditional ML methods, particularly when handling large volumes of data. However, similar to ML models, DNNs also require substantial labeled datasets for effective training and can still be susceptible to adversarial manipulations. Despite the potential of both ML and DL, a significant challenge in real-world applications is the scarcity of labeled anomaly (attack) data. Attacks are infrequent, and labeling a wide range of attack behaviors is very difficult, costly, or even unavailable. This creates a barrier for both traditional and ML/DL-based anomaly detection methods that rely on comprehensive labeled datasets.

Unlike traditional one-class methods like Deep One-Class Classification (Deep SVDD) [11], which consider only normal data during training, this paper proposes a Semi-Supervised Anomaly Detection (SSAD) approach to incorporates both labeled normal samples and a limited number of labeled anomaly samples, allowing the model to form a more informative decision boundary and improving its ability to detect previously unseen attacks. SSAD is specifically designed to leverage partially labeled datasets, in which normal data are abundantly labeled, while anomaly (attack) samples are sparse or entirely unlabeled. This makes it particularly suitable for network intrusion detection in substation environments, where labeled attack data is often unavailable. By combining deep neural networks with a specialized loss function, SSAD learns a latent representation space where normal samples cluster around a central point, while anomalies are pushed farther away, resulting in a clear separation. This enhances detection accuracy in data-scarce conditions and outperforms conventional supervised methods, offering a flexible and efficient solution for real-time cybersecurity in digital

substations. In practical deployments, unlabeled data streams are common due to the high cost of manual labeling. SSAD effectively leverages such data by learning decision boundaries from known normal patterns and isolating anomalous behavior without requiring prior labeling.

In summary, the main contributions of this study are as follows: We propose a semi-supervised anomaly detection (SSAD) approach that combines the strengths of semi-supervised learning with deep neural networks to detect both known and unknown cyber threats in digital substations. The proposed approach provides a scalable, efficient, and real-time solution for detecting anomalies in the critical communication protocol IEC 104, ensuring the security of substation systems even when labeled anomaly data are limited. We highlight the potential of SSAD for future research in cybersecurity for smart grids, emphasizing its ability to handle evolving and sophisticated cyber threats without the need for extensive retraining or large labeled datasets.

2. Substation architecture

Substations are crucial components within the electrical grid infrastructure. Their primary functions include stepping down electrical voltage to appropriate levels for transmission and distribution, ensuring system protection, and facilitating network stability through interconnection management. Additionally, substations enable fault isolation and support maintenance activities via advanced switching mechanisms. The adoption of digital technologies in substations, guided by communication protocols such as IEC 61850 and IEC 60870-5-104 (commonly referred to as IEC 104), is vital for enabling automation and reliable communication. However, this digital transformation also introduces new cybersecurity challenges [7]. Structurally, substations are generally divided into three hierarchical levels-Station, Bay, and Process-which are interconnected through Station and Process buses, as illustrated in Figure 1.

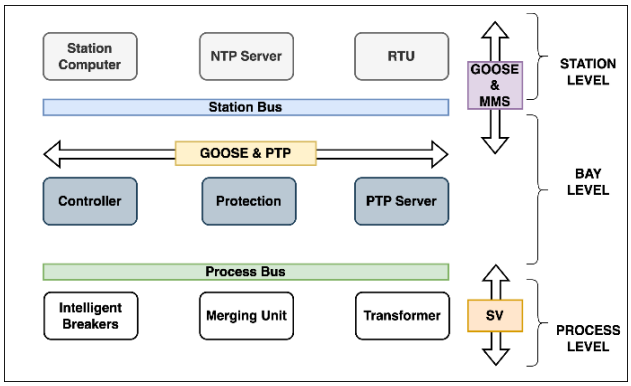

Figure 1. *Overview of a typical digital substation architecture, including Station, Bay, and Process levels, highlighting data flow and key components like SCADA, IEDs, and merging units*

Each level is described in further detail below. Monitoring, control, and communication with external systems such as control centers and other substations are the responsibilities of the Station Level. Protocols commonly employed at this level include IEC104, Network Time Protocol (NTP), and Precision Time Protocol (PTP). Typically encompassed at this level are a Supervisory Control and Data Acquisition (SCADA) system, which enables real-time monitoring and control of the entire substation via a Remote Terminal Unit (RTU); a Human-Machine Interface (HMI), which provides graphical displays to operators for interacting with substation control systems; additional servers and workstations that host software for data processing, visualization, and control; time synchronization servers; and a router that facilitates connectivity with the control center. The Bay Level is tasked with controlling and protecting individual substation sections, often referred to as "bays", which include transformers, feeders, and busbars. Control commands and protection algorithms are executed at this level. Components found here consist of Intelligent Electronic Devices (IEDs) responsible for bay-specific control, protection relays that detect faults and initiate protective actions such as circuit breaker tripping, and control panels accompanied by a local HMI to manage bay equipment operations. Direct interaction with physical electrical equipment is carried out at the Process Level. Real-time data acquisition from sensors and actuators is performed, alongside the transmission of control commands to primary equipment like transformers and circuit breakers. This level may incorporate multiple merging units, which digitize electrical signals and share measurements via the Sampled Values protocol, as defined by IEC61850. IEC61850 serves as a comprehensive standard aimed at modernizing substation automation by emphasizing interoperability and open system architectures. Seamless integration among devices from different manufacturers is enabled, alongside support for real-time communication and data modeling within substations. Each device is represented in an object-oriented manner as a collection of logical nodes, which facilitates efficient operation even within complex and large-scale environments. Moreover, several network protocols are defined within this standard. For instance, Manufacturing Message Specification (MMS) supports client-server communication between IEDs and control systems, allowing real-time exchange of data, control commands, and status information over TCP/IP. The Generic Object-Oriented Substation Event (GOOSE) protocol, designed for real-time protection and automation functions with strict delay requirements (with latencies as low as 3 milliseconds in certain cases), is transmitted directly over Ethernet. Similarly, Sampled Values (SV), used to convey digitized analog data such as current and voltage measurements from merging units to protective relays and other IEDs, is also sent over Ethernet. Finally, IEC104 extends the IEC60870-5 standard by incorporating network access through Ethernet, focusing on remote monitoring and control of substations. This protocol is especially suitable for telecontrol applications, leveraging existing network infrastructures through the standard TCP/IP stack.

3. Proposed scheme

In this study, the core idea behind the proposed SSAD is to learn a deep neural network function that maps input

samples to a latent space in which normal samples are clustered around a predefined center, while anomalous samples are pushed farther away. Assume that, in addition to the n unlabeled samples $x_1, x_2, \dots x_n \in \chi \subset \mathbb{R}^D$, we also have access to m labeled samples $(\widetilde{x_1}, \widetilde{y_1})$, $(\widetilde{x_2}, \widetilde{y_2})$,..., $(\widetilde{x_n}, \widetilde{y_n}) \in \chi \times Y$, where $Y = \{-1, +1\}$, with $\widetilde{y_j} = +1$ denoting known normal samples and $\widetilde{y_j} = -1$ denoting known anomalies. The goal of SSAD is to learn a function $f(\cdot;W)$ parameterized by weights W, that maps the input samples to a latent space where normal samples are clustered around a center c, and anomalous samples are pushed further away from this center. The loss function for SSAD consists of several terms that are designed to balance the contributions from normal and anomalous samples, as well as prevent overfitting. The total loss is formulated as [8]:

$$\mathcal{L}_{total} = \frac{1}{n+m}\sum_{i=1}^{n} \| f(x_i;W) - c \|^2 + \frac{\eta}{n+m}\sum_{j=1}^{m} (\| f(\widetilde{x_j};W) - c \|^2)^{\widetilde{y_j}} + \frac{\lambda}{2}\sum_{l=1}^{L} \| W^l \|_F^2 \quad (1)$$

Where n is the number of unlabeled training samples and m is the number of labeled training samples. The term $f(x_i;W)$ represents the embedding of the sample x_1 into latent space through the neural network. The center c is the center point in the latent space around which normal samples should cluster. The parameter η controls the relative importance of the labeled anomalies compared to the normal samples, while λ is the regularization parameter that prevents the model from overfitting. L is the number of layers in the neural network, W^l represents the weight matrix for the l-th layer of the network, and $\| W^l \|_F^2$ is the Frobenius norm regularization term applied to the weight matrices. The loss function is designed to guide the model in mapping normal samples close to the center c, while pushing anomalous samples further away. This is achieved through the first term that ensures that normal unlabeled samples are embedded close to the center c in the latent space, $\frac{1}{n+m}\sum_{i=1}^{n} \| f(x_i;W) - c \|^2$. This term encourages the neural network to learn a compact representation of the normal class. The second term $\frac{\eta}{n+m}\sum_{j=1}^{m} (\| f(\widetilde{x_j};W) - c \|^2)^{\widetilde{y_j}}$, incorporates labeled data. For normal samples ($\widetilde{y_j} = +1$), the term minimizes the distance from the center c. For anomalous samples, ($\widetilde{y_j} = -1$), the term maximizes the distance from c. The hyperparameter η controls the relative weight of the labeled anomaly samples compared to the unlabeled normal samples. The third term, $\frac{\lambda}{2}\sum_{l=1}^{L} \| W^l \|_F^2$, is a weight decay regularization term. It penalizes large weights to prevent overfitting and ensures that the model generalizes well to new, unseen data. During training, we optimize the loss function $\mathcal{L}_{total}$ using stochastic gradient descent (SGD) optimization techniques [9]. The network parameters W are updated iteratively to minimize the loss, ensuring that normal samples are embedded close to the center c, while anomalous samples are pushed away from the center. The regularization term helps prevent overfitting by keeping the network weights small. Once the model is trained, during inference, a new sample x_{new} is passed through the trained network to obtain its latent representation $f(x_{new};W)$. The anomaly score $s(x_{new})$ is computed as the Euclidean distance from the center c:

$$s(x_{new}) = \| f(x_{new};W) - c \| \quad (2)$$

To classify the sample x_{new} anomalous or normal, a threshold $\mathcal{T}$ must be set. A sample is then classified as anomalous if:

$$s(x_{new}) \geq \mathcal{T}$$

The choice of $\mathcal{T}$ plays a crucial role in balancing the false positive and false negative rates. Since SSAD is designed for semi-supervised settings with limited or no anomalous samples in training D_{train}^{nomal}, $\mathcal{T}$is typically selected based on the distribution of scores obtained from the normal training data. One common approach is to set $\mathcal{T}$ such that a fixed proportion $\alpha$of the training samples exceed it:

$$\mathcal{T} = \min \{t \in \mathbb{R} \mid \mathbb{P}_{x \sim D_{train}^{nomal}}(s(x) \leq t) \geq 1 - \epsilon\} \quad (3)$$

where $\epsilon \in (0, 1)$ reflects the expected false positive rate on the training data. Alternatively, the threshold can be optimized to maximize the performance metrics. Our proposed SSAD algorithm as presented in Algorithm 1.

Algorithm 1 Optimization of proposed SSAD

Input: Labeled samples: $(\widetilde{x_1}, \widetilde{y_1}),..., (\widetilde{x_m}, \widetilde{y_m}) \in \chi \times Y$
Unlabeled samples: $x_1, \dots x_n \in \chi \subset \mathbb{R}^D$
Hyperparameters: η, λ
Encoder: $f(\cdot;W)$
SGD learning rate: α
Output: Trained model weights: W^*
Initialize:
Neural network weights: W
Hypersphere center: c
for *each epoch* **do**
for *each mini batch* $\mathcal{B}$ **do**
Draw *mini batch* $\mathcal{B}$ from labeled and unlabeled data
$W \leftarrow W - \alpha . \nabla_W \mathcal{J}(W;\mathcal{B})$
end for
end for

4. Experimental setting

4.1. Dataset and parameter setting

The dataset used in this study exclusively comprises network traffic related to the IEC104 protocol. Operational data were collected from a substation environment, where frames adhering to the IEC104 protocol were captured. Since IEC104 operates over the transport layer using the TCP/IP stack, relevant TCP/IP flows were isolated and extracted using the CICFlowMeter tool [10]. The resulting flow-level data were exported in CSV format. The final dataset consists of 319,971 rows and 79 columns. Of these, 78 columns contain features extracted from the network flows, while the last column represents the label indicating whether the flow is normal (label 0) or anomalous (label

1). Table 1 provides a subset of representative features included in the dataset, such as destination port, protocol type, flow duration, forward and backward packet statistics, and various packet length metrics.

Table 1. *Selected network flow features used in model training and evaluation, including protocol types, flow durations, packet lengths, and statistical measures. These features form the input vector for the SSAD model*

Feature Name	Description	Type
Dst Port	Destination port number	Integer
Protocol	Network protocol used (e.g., TCP, UDP)	Categorical
Flow Duration	Duration of the flow in microseconds	Integer
Total Fwd Packet	Total number of packets sent in the forward direction	Integer
Total Bwd Packet	Total number of packets sent in the backward direction	Integer
Total Length of Fwd Packet	Total length of all forward packets in bytes	Integer
Total Length of Bwd Packet	Total length of all backward packets in bytes	Integer
Fwd Packet Length Max	Maximum length among forward packets	Integer
Fwd Packet Length Min	Minimum length among forward packets	Integer
Fwd Packet Length Mean	Mean length of forward packets	Float
Fwd Packet Length Std	Standard deviation of forward packet lengths	Float
Bwd Packet Length Max	Maximum length among backward packets	Integer
Bwd Packet Length Min	Minimum length among backward packets	Integer
Bwd Packet Length Mean	Mean length of backward packets	Float
Bwd Packet Length Std	Standard deviation of backward packet lengths	Float

Table 2. *Dataset distribution showing the number of normal and anomalous samples in both training and testing sets.*

Type	Training	Test	Total
Anomalous samples	204	51	255
Normal samples	255772	63944	319716
Total	255976	63995	319971

Table 2 provides a complete overview of the dataset distribution across both the training and test sets. It lists the number of normal and anomalous samples in each subset, offering a holistic view of the data composition. The dataset is notably highly imbalanced, with anomalous samples accounting for the vast majority in both splits.

Figure 2 illustrate exclusively on the training set, highlighting the stark imbalance between normal and anomalous samples during model learning. The chart adopts a logarithmic scale to better visualize the presence of normal samples, which would otherwise be barely visible due to their rarity.

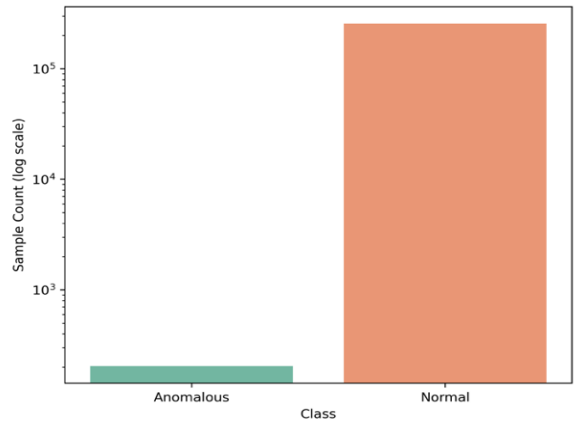

Figure 2. *Logarithmic-scale distribution of training data showing a significant imbalance between normal and anomalous samples, underscoring the need for robust detection methods*

4.2. Models and Metrics

The model structure and the model sizes of the implemented SSAD algorithm is depicted in Table 3. The optimizer is stochastic gradient descent (SGD) with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$. The batch size $\mathcal{B}$ is set to 64. For all datasets, we use cosine learning rate decay schedule [11], with $\eta = \eta_0 \cos(\frac{7\pi t}{16T})$, where $\eta_0 = 0.001$ is the initial learning rate, t is the current training step, and T is the total training step. All experiments in the paper are implemented on one NVIDIA GeForce RTX 4090 GPU with 125GB RAM.

Table 3. *Architecture of the SSAD neural network, including layer sizes and activation functions. The design aims to project input data into a latent space for anomaly separation.*

Layer types	Input size	Activation	Output size
Input layer	78	-	78
Fully Connected	100	ReLU	100
Fully Connected	100	ReLU	100
Fully Connected	100	ReLU	64
Fully Connected	64	-	32

To compare models, the Area Under the ROC Curve (AUC) is often recommended, particularly with imbalanced datasets, as it provides a balanced view of performance across all thresholds. F1-Score (F1) is particularly valuable in such scenarios, as it balances the importance of Precision (Prec.) and Recall.

$$\text{Prec} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{4}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{5}$$

and $$\text{F1} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}} \tag{6}$$

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

4.3. Experimental results

In this section, we compare the performance of three methods for anomaly detection: Deep SVDD [11], Deep AE [12], and the proposed SSAD model. The evaluation

is based on four metrics: AUC, F1, Prec, and Recall. As shown in Table 4 and Figure 3, the proposed SSAD consistently outperforms the other two methods across all metrics. Notably, the Proposed SSAD achieves an AUC of 0.98, indicating near-perfect discriminative capability between the two classes. The F1-score reaches 0.98, reflecting a well-balanced trade-off between Precision (1.00) and Recall (0.96). This is especially critical in imbalanced classification problems, where achieving high recall without sacrificing precision is often non-trivial. In comparison, Deep SSVD achieves an AUC of 0.92 and F1-score of 0.91-reasonable performance, yet it falls short in terms of recall (0.84), implying that it misses more positive instances. Deep AE performs the worst among the three, with an AUC of 0.89 and a recall of only 0.78, suggesting a high number of false negatives. The outstanding performance of the proposed SSAD lies not only in its high scores but also in the consistency across all metrics. These results validate the effectiveness of the proposed SSAD approach in improving detection quality compared to traditional deep learning techniques previously employed.

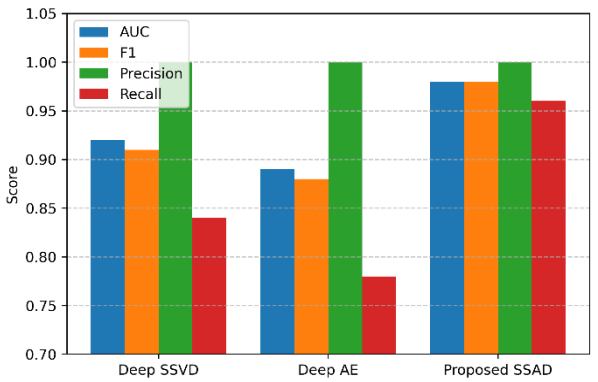

Figure 3. *Comparative performance of SSAD, Deep SVDD, and Deep AE in terms of AUC, F1-score, Precision, and Recall. SSAD demonstrates superior and consistent performance across all metrics*

Table 4. *Performance comparison of Deep SVDD, Deep AE, and the proposed SSAD method using standard metrics*

Metrics	AUC	F1	Prec	Recall
Deep SSVD	0.92	0.91	1.00	0.84
Deep AE	0.89	0.88	1.00	0.78
Proposed SSAD	0.98	0.98	1.00	0.96

5. Conclusion

This study presents a comprehensive approach to enhancing cybersecurity in digital substations by proposing SSAD framework tailored for the IEC 104 communication protocol. By leveraging semi-supervised learning combined with deep neural networks, the SSAD method effectively identifies both known and unknown cyber threats, addressing a critical challenge posed by limited labeled anomaly data in real-world environments. The experimental results demonstrate the superiority of the SSAD method in detecting anomalies with high precision, recall, and AUC performance, confirming its practical applicability in securing substation communication systems. This work not only contributes a novel detection framework but also highlights the potential of semi-supervised learning for future cybersecurity research in smart grids, where evolving threats demand adaptive and resource-efficient defense mechanisms. In conclusion, the integration of SSAD into substation security architectures can significantly strengthen the resilience and stability of modern power grids, paving the way toward safer and smarter energy infrastructures.

REFERENCES

[1] P. Blazek, A. Bohacik, R. Fujdiak, V. Jurak, and M. Ptacek, "Smart Grids Transmission Network Testbed: Design, Deployment, and Beyond". *IEEE Open Journal of the Communications Society*, vol. 6, pp. 51-76, 2024. DOI: 10.1109/OJCOMS.2024.3517340

[2] R. Holdbrook, O. Odeyomi, S. Yi, and K. Roy, "Network-Based Intrusion Detection for Industrial and Robotics Systems: A Comprehensive Survey", *Electronics,* vol. 13, no. 22, Art. no. 4440, 2024.

[3] G. Kumar, K. Kumar, and M. Sachdeva, "The use of artificial intelligence based techniques for intrusion detection: a review", *Artificial Intelligence Review*, vol. 34, pp. 369–387, 2010.

[4] H. Nhung-Nguyen, M. Girdhar, Y.-H. Kim, and J. Hong, "Machine-Learning-Based Anomaly Detection for GOOSE in Digital Substations", *Energies*, vol. 17, no. 15, 3745, 2024.

[5] T. S. Ustun, S. M. S. Hussain, A. Ulutas, A. Onen, M. M. Roomi, and D. Mashima, "Machine Learning-Based Intrusion Detection for Achieving Cybersecurity in Smart Grids Using IEC 61850 GOOSE Messages", *Symmetry*, vol. 13, no. 5, 826, 2021.

[6] A. Aljohani, M. AlMuhaini, H. V. Poor, and H. M. Binqadhi, "A Deep Learning-Based Cyber Intrusion Detection and Mitigation System for Smart Grids", *IEEE Trans. Artificial Intelligence*, vol. 5, no. 8, pp. 3902–3914, Aug. 2024.

[7] E. D. Gutiérrez Mlot, J. Saldana, R. J. Rodríguez, I. Kotsiuba, and C. Gañán, "A dataset to train intrusion detection systems based on machine learning models for electrical substations", *Data in Brief*, vol. 57, 111153, 2024.

[8] L. Ruff *et al.*, "Deep Semi-Supervised Anomaly Detection", *arXiv preprint arXiv:1906.02694*, 2019.

[9] L. Bottou, "Large-scale machine learning with stochastic gradient descent", in *Proc. 19th Int. Conf. Computational Statistics (COMPSTAT 2010)*, Paris, France, Aug. 22–27, 2010, pp. 177–186.

[10] Canadian Institute for Cybersecurity, CICFlowMeter – Applications, *www.unb.ca,* 2018. [Online]. Available: https://www.unb.ca/cic/research/applications.html. [Accessed: April 5, 2025].

[11] L. Ruff *et al.*, "Deep one-class classification", in *Proc. 35th Int. Conf. Machine Learning (ICML)*, PMLR, vol. 80, pp. 4393–4402, 2018.

[12] B. Zong *et al.*, "Deep autoencoding Gaussian mixture model for unsupervised anomaly detection", in *Proc. Int. Conf. Learning Representations (ICLR)*, Vancouver, BC, Canada, Apr.–May 2018.