IoT-Based Secure Framework for Smart Grids Using Machine Learning and Blockchain Technologies

Zeyad Al-Odat

Department of Computer and Communications Engineering, Faculty of Engineering,
Tafila Technical University, Tafila, Jordan
Email: Zeyad.alodat@ttu.edu.jo (Z.A.O.)

***Abstract*—The growing exposure of Industrial Control Systems (ICS), especially smart grids, to the internet and the proliferation of wireless networks exacerbate the cybersecurity risks to these systems. Meanwhile, developments in using cutting-edge technology for managing and controlling industrial systems enhance remote troubleshooting and operational flexibility; nevertheless, they also increase vulnerability to cyberattacks. A testing environment is required to simulate and assess the impact of cyberattacks on smart grids. This work introduces a security framework that offers intrusion detection and prevention functionalities at both the host and network levels, increasing the trust in threat detection approaches. Utilizing machine learning and artificial intelligence to enhance the detection of threats and assaults efficiently. Machine learning and blockchain are used to fulfill the security objectives of the suggested architecture. Extreme Gradient Boosting (XGBoost) and SHapley Additive exPlanations (SHAP) are used to analyze the architecture and identify the most significant elements of industrial data, while blockchain technology, coupled with smart contracts, is utilized to document and secure transactions between Internet of Things (IoT) devices. Moreover, the proposed design provides an explanatory environment to interpret the output of the machine learning algorithms, showing the significance of each feature element of the system. The proposed methodology has outstanding effectiveness in detecting cyberattacks, as shown by precision, recall, F1-Score, accuracy, and other performance metrics.**

***Keywords*—cybersecurity, Internet of Things (IoT), smart grid, blockchain, machine learning**

I. Introduction

The global IoT market, valued at USD 405.59 billion in 2023, is expected to grow exponentially to USD 3152.17 billion by 2033, as depicted in Fig. 1. This grow is driven by its transformative potential across industries such as smart grids and industrial automation [1].

Prior research adopted the integration of Machine Learning (ML) and blockchain which offers transformative potential for IoT security. While ML analyzes vast datasets to detect anomalies, blockchain ensures data integrity and decentralized trust via smart contracts. Together, they address critical challenges such as secure data sharing, adversarial resilience, and energy-efficient automation in industrial systems. Nevertheless, smart contracts employing digital signatures can markedly enhance the entire procedure [2].

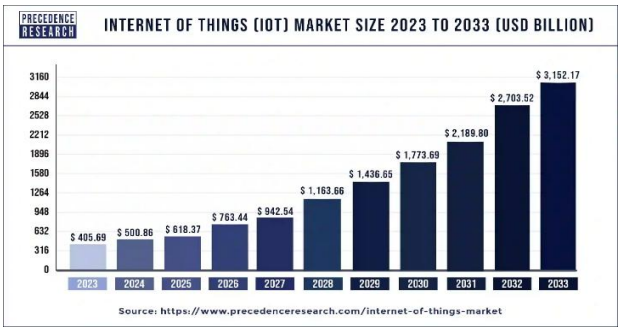

Fig. 1. The growth of the IoT market in industry. (Reprinted with permission from [1]).

The advantages of machine learning and blockchain in the industry comprise the following:

1. Enhance safety: The blockchain data is securely safeguarded using implicit encryption. Blockchain is optimal for the storage of highly sensitive personal information, including personal recommendations and medical records. There exists an additional facet of enhancing security. Although blockchain possesses intrinsic security, apps and supplementary layers may exhibit vulnerabilities [3].
2. Machine learning will assist in forecasting potential system faults and implementing blockchain applications [4].
3. Data Market Administration: Major corporations such as Google, Facebook, and Amazon possess extensive datasets that are advantageous for AI applications, yet this data remains inaccessible to others [5].
4. Blockchain enables startups and small enterprises to compete with large corporations by providing

Manuscript received January 7, 2025; revised February 12, 2025; accepted April 11, 2025; published August 8, 2025.

access to the same data resources and artificial intelligence [6].

5. Enhance energy efficiency: Data mining, an energy-intensive operation, poses a significant challenge in contemporary society. Through the training of DeepMind AI, they achieved a 40% reduction in energy use for cooling its data centers [7].
6. Applications of Machine Learning and Blockchain: Industrial Automation Companies (IAC) currently utilize smart contracts and Bitcoin blockchain-based processes in their manufacturing procedures to guarantee transparency, efficiency, security, and compliance verification [8].

In smart grid environments, many approaches have emerged to enhance security inside these environments. Blockchain technology has emerged as an evolving approach to ensure the enhancement of security procedures toward smart grids [9]. Blockchain provides a decentralized ledger that ensures the transactions and data exchange are securely recorded, which in turn reduces the risks accompanied by centralized data communication and control [10, 11]. Yet critical gaps persist:

1. Machine learning driven frameworks rely on black box models, limiting auditability and trust in threat detection. In addition, adversarial vulnerabilities leave the machine learning models exposed to evasion [12].
2. The centralized blockchain infrastructure introduces unintentional delays to make it incompatible with real-time industrial environments [13].
3. Baseline line machine learning algorithms, e.g., decision tree, naïve base, random forest, have less precision in detecting complicated threats [14].

This work presents a secure framework that employs machine learning, blockchain, and smart contracts. The proposed design addresses the gaps through an optimized framework that combines interpretable machine learning, blockchain and adversarial resilience. By exploiting the advantages of these techniques, the proposed design provides faster incident response and data analysis approaches. The machine learning approach is augmented by the eXtreme Gradient Boosting (XGBoost) that provides the required optimization and efficiency for the design. Moreover, SHapley Additive exPlanations (SHAP) is employed to signify the importance of features for the input data. This work helps in protecting smart grid architectures by providing a rigid security infrastructure that detects and responds to cybersecurity threats and vulnerabilities. Moreover, ensure reliable power management and secure data transmission alongside user privacy that allows users to interact and approve their data securely.

The rest of the paper is organized as follows: Section II provides a literature review for the state of the art in the field of smart grid security. The proposed architecture and design implementation will be presented in Section III. Results and discussion in Section IV. Section V concludes the paper.

II. Literature Review

The accelerated growth of the Internet of Things (IoT) technology came up with ecosystems that are compelling and convoluted. These ecosystems are characterized by the integrated IoT devices inside the interconnected mesh of different layers. Many challenges face the security and the performance of the IoT devices, which is considered a crucial challenge that significantly degrades the performance of IoT devices and makes the system vulnerable. Recent research considered innovative use of machine learning and blockchain technologies to help in addressing vulnerabilities and enhance the security of the ecosystems.

Further security frameworks employed Machine Learning (ML) to augment these frameworks by enabling anomaly detection and analysis algorithms. These algorithms analyze large data generated by the IoT devices and identify patterns to detect unusual behaviors. The ML algorithms continuously learn from new data accumulated to previously generated data, which helps the ML to adapt to the changing conditions and emerging threats. This adaptation increases algorithm effectiveness and responsiveness. Merging the blockchain integrity features and machine learning adaptive capability provides a comprehensive and enhanced resilience approach for securing the smart grid systems against cybersecurity attacks and threats [15, 16].

A convergence between digital technologies and physical systems is represented in the Industrial Control Systems (ICS), which is designed to manage processes such as production, operation, and distribution. These systems are usually termed Cyber-Physical Systems (CPSs), emphasizing the complex interaction between industrial activities and computing and communication infrastructures [17]. The ICS systems depend on their functionality, on the understanding of physical processes, and on communication with the Supervisory Control and Data Acquisition (SCADA) systems. CPSs are deployed across many industrial sectors. This deployment includes critical industries such as: 1) energy grids, 2) petroleum and gas pipelines, 3) water treatment, and 4) other manufacturing facilities. According to their importance, malicious cyberattacks trigger fallout of the system, coming up with unusual activities, operational malfunction, and safety risks [18].

Nazir *et al*. [16] proposed and enhanced IoT security through collaborative threat intelligence. The work presented the integration between blockchain and Machine Learning (ML), which offers an active collaboration against cyber threats. The proposed design leverages the blockchain's decentralized architecture and ML analysis capabilities to secure data sharing and detect threats, respectively. This approach signifies the usefulness of shared intelligence by exchanging verified threat data and responding to complicated risks and attacks, e.g., Distributed Denial of Service (DDoS) and unauthorized access. In contrast, the work indicates the use of ML to detect potential threats and blockchain to provide integrity requirements, which in turn provide a robust and secure infrastructure. The collaboration of machine learning and

blockchain undertake a storing combination to face cyber threats in the IoT ecosystems.

The accelerated development of the IoT interconnected networks signifies the need for security frameworks that can address the confrontation postured by enormous devices interconnected inside ecosystems. Alsharif *et al*. [18] proposed an Intrusion Detection (IDS) architecture that employs Convolutional Neural Network (CNN) machine learning and blockchain technologies. The proposed design takes advantage of the machine learning algorithm to detect anomalies inside the IoT network. While the blockchain provides a decentralized layer that makes the system immune. Combining both technologies provides a robust and secure architecture for secure data management and an effective approach against vulnerabilities.

With the increasing dependency on IoT deployments in smart cities, protection against a potential threat is essential. Havlena *et al.* [17] examines a framework that employs blockchain and machine learning to focus on the security breaches that are directed to smart cities. The proposed design records the interaction between IoT devices and ensures transparent data handling across locations. Machine learning is employed to analyze and detect possible threats and respond to cyber-attacks. The combined approach consolidates IoT security, enhances the smart city's resilience, and improves the reliability of the ecosystem.

When considering the security and privacy of smart cities, there must be a balance between them. Moawad *et al.* [19] proposed an intrusion detection model that incorporates machine learning and blockchain technology. The proposed design ensures data integrity through blockchain and decentralized storage while analyzing and detecting intrusions using a machine learning algorithm. This work protects sensitive information and adapts to building up a secure cyber threats' environment.

TABLE I. COMPARISON BETWEEN DIFFERENT WORKS IN THE LITERATURE

Work	Objectives	Technologies Used	Key Contribution	Limitations (Impact on IoT Security)
[16]	Incorporates blockchain for data integrity and machine learning for instantaneous threat detection.	Random Forest, Decision Tree classifier, CNN and Blockchain	Introduces a decentralized model for threat intelligence sharing that enhances the resilience of IoT networks against cyber-attacks.	Black-box ML models lack interpretability, high blockchain latency (reduces trust/auditability in threat detection, delays real-time responses to attacks)
[18]	To develop an Intrusion Detection System (IDS) that secures IoT networks by combining ML and blockchain.	Linear Regression (LR), Random Forest and Support Vector Machine (SVM) and Blockchain	Proposes a scalable and robust IDS that improves anomaly detection while ensuring secure data transmission through blockchain technology.	Supervised ML requires labeled data; centralized blockchain architecture creates single points of failure (impractical for dynamic IoT environments with evolving threats, creates single points of failure)
[19]	Developing a comprehensive security framework tailored for smart city applications	Blockchain, Machine Learning	Demonstrates the effectiveness of integrating blockchain and ML to secure smart city IoT networks, promoting data integrity and real-time threat responses.	No adversarial attack resilience (vulnerable to model evasion/data poisoning)
[20]	To enhance the Quality of Service (QoS) in IoT networks	Naive Bayes (NB), Decision Tree (DT) and Blockchain	Provides a solution to improve QoS in IoT systems, focusing on data security and real-time network performance optimization.	Using baseline ML algorithms (limited accuracy in detecting sophisticated IoT attacks).
[21]	To create a hybrid IDS that protects smart networks while ensuring user privacy.	Hybrid Decision Tree Method (HIDT) and Blockchain	Provides a solution to improve QoS in IoT systems, focusing on data security and real-time network performance optimization.	Narrow focus on QoS over security (prioritizes performance, leaving gaps in threat coverage)

Table I shows the comparison between different works from the literature regarding their objectives, technology used, and key contributions. As depicted in the table, several works explored the usage of machine learning and blockchain for anomaly detection and data integrity. However, these works are limited to the comprehensive integration of these technologies and do not fully address the intrusion detection and security management. This work bridges this gap by providing a framework that integrates machine learning, IoT devices, and Explainable Artificial Intelligence (XAI) augmented with blockchain technology. Moreover, the proposed design provides an explanatory environment to interpret the output of the machine learning algorithms, showing the significance of each feature element of the system. Furthermore, the suggested design addresses the shortcomings of previous efforts and establishes a comprehensive framework for recognizing and addressing problems. The incorporation of XAI resolves the explainability challenge associated with black-box machine learning techniques. The decentralized blockchain design resolves the scalability issue. The real-time training and detection provide the required adaptability to the machine learning method.

III. PROPOSED METHODOLOGY

Industrial datasets are frequently employed to assess security threats to machine learning models, especially with adversarial machine learning and model resilience. The attacks that target these types of datasets are classified as follows:

- Adversarial attacks entail the subtle modification of input data to deceive the model, employing methods such as noise introduction and word substitutions.
- Backdoor attacks embed a trigger inside the model that induces anomalous behavior upon the reception of a specific input.

- Poisoning attacks compromise the training process by introducing detrimental data or inaccurately labeling training samples, resulting in misclassification.
- Evasion attacks alter inputs to evade detection, whereas model inversion seeks to extract sensitive information from the model.
- White-box, which vary according to the extent of access an attacker possesses to the model's internals.
- Black box attack varies according to the level the attacker possesses.
- Robustness assessment attacks assess a model's resilience to adversarial conditions.
- Transferability attacks entail creating adversarial examples on one model and deploying them on another.

These categories emphasize various ways for evaluating the security and resilience of models, particularly against adversarial manipulation, aiming to enhance model performance and guarantee reliability in practical applications. Table II depicts these types of attacks and their corresponding classes [22–25].

TABLE II. THE CATEGORIES OF ATTACKS ON INDUSTRIAL DATASETS AND THEIR CORRESPONDING CLASS

Attack Category	Attack Class
Adversarial	Manipulation of input to cause misclassification
Backdoor	Implanting a hidden trigger that modifies behavior
Poisoning	Malicious modification of training data to corrupt learning
Evasion	Manipulating inputs to bypass detection or misclassification
Model Inversion	Inferring sensitive information from the model
White box	Attacks based on full knowledge of model internals
Black box	Attacks based on access only to the model's
Robustness Evaluation	Tests to measure the model's resilience
Transferability	Creating adversarial examples that transfer to different models

A. System Architecture

The proposed design aims to provide real-time threat analysis and detection, ensure data integrity, and provide adequate management over the smart grid. The proposed design leverages the combination between Machine Learning (ML) and blockchain techniques. The proposed architecture incorporates three primary layers to enhance the security of IoT-based smart grids. Each layer ensures the security of the design by assigning a unique role for each one of them. These rules ensure threat detection, data integrity, and maintain the system's resilience.

Fig. 2 represents the proposed design architecture and the integration between the three layers of the design. IoT devices, blockchain, and machine learning layers are used to enhance security, ensure data integrity, and maintain system scrutiny. The figure shows the integration between the three layers and the collaboration between them according to the following steps:

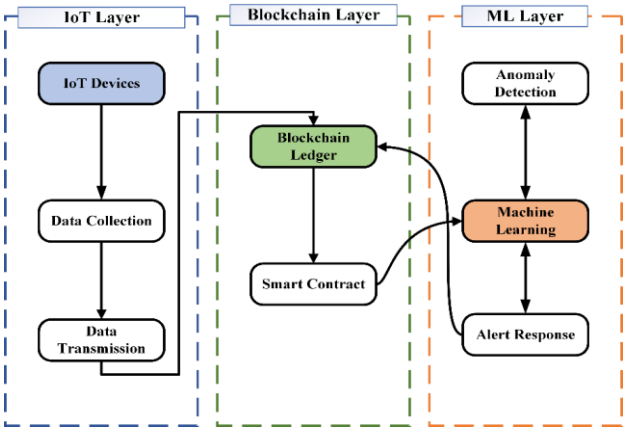

Fig. 2. System architecture for IoT-based smart grid security framework.

1) IoT layer

This layer contains the IoT devices, which are sensors, actuators, and devices collecting data from the surrounding environment. The functionality of this layer is incorporated in the following:

- Data Collection: This layer collects raw data from the IoT devices for further processing. The collected data needs a preprocessing stage before it is passed to the next stage.
- Data Transmission: The collected data is securely transmitted to the next layer for validation and storage.

2) Blockchain layer

This layer works as an intermediary between the IoT Layer and the Machine Learning (ML) Layer. It adopts the following functionalities:

- Blockchain Ledger: The ledger encodes data from the previous layer. The data is segmented and cryptographically linked to blocks. The main purpose of the ledger is to ensure non-repudiation of records that are collected from different IoT devices.
- Smart Contract: The smart contract expedites and automates the verification process of transmissions between the IoT devices, which in turn enhances procedural efficiency by enforcing predefined rules.

3) Machine learning layer

This layer works as a computational intelligence paradigm. It analyzes and detects possible threats that might harm the system. The ML layer plays a major role in the proposed architecture. It involves three complex functionalities.

- Machine Learning: The proposed design employs eXtreme Gradient Boosting (XGBoost) classification paradigms. The XGBooST approach supports the decentralized nature of the blockchain, which ensures decentralized training and scalability. This layer learns from the data stored in the blockchain layer, recognizing anomaly patterns.
- Anomaly Detection: This functionality ensures higher accuracy in detecting anomalous actions.

This includes the deviation of the predicted pattern caused by operational faults, which minimize false positive behaviors.

- Alert Response: This functionality responds to anomaly actions based on the severity of the detected threat. The low-risk anomalies notify the operator to investigate or reconsider, while the high-risk actions execute an immediate countermeasure. The list of countermeasure actions can be specified according to the operational environment. In our case, it triggers node isolation and reconfiguration.

B. Integration Between Machine Learning and Blockchain

The integration between the components of the proposed design ensures that the ecosystem of the IoT-based smart grid is secure and has a prophylactic real-time threat detection mechanism. All three layers work together to accomplish the prospective task. Algorithm 1 shows the integration and operation of the proposed design. The process is summarized as follows:

1) XGBoost-based model for classification

The selection of XGBoost comes from the fact that it can handle large datasets effectively and efficiently because of the fast training that can be accomplished in parallel. On the other hand, XGBoost supports the ability to understand the feature importance, which helps in the selection of the features according to their importance. Moreover, XGBoost reduces overfitting compared to SVM [26].

2) Data validation and preprocessing

The data are collected from the IoT devices and passed through the preprocessing phase to formalize data for further steps. The data is divided into features and labels. Then the features are normalized to ensure zero mean and unit variance.

3) Data division

The dataset is divided into training and testing with a ratio of *t*. In our experiment, the ratio is 70% training and 30% testing.

C. XGBoost and SHAP Integration

SHapley Additive exPlanations (SHAP) is a method for interpreting machine learning models by assigning importance scores to input features. This helps identify which features contribute the most to a model's decisions by providing an additive tree paradigm [26, 27]. The SHAP explanation output is used by the XGBoost to minimize the perpetual objective function, as shown in Eq. (1).

$$\mathcal{L}(\phi) = \sum_{i=1}^{n} l(y_i, \hat{y_i}) + \sum_{t=1}^{T} \Omega(f_t) \quad (1)$$

Here,

- $l(y_i, \hat{y_i})$ loss function,
- $\hat{y_i} = \sum_{t=1}^{T} f_t(x_i)$ represent the prediction for data point i with respect to t^{th} tree,
- $\Omega(f_t) = \gamma T + \frac{1}{2}\lambda|w|^2$ represent the complexity of the perpetual terms for the tree,
- T: number of leaves,
- w: leaf weights and
- γ, λ are the perpetual parameters.

The equation represents the objective function that XGBoost minimizes during training, combining both prediction accuracy and model simplicity. With every iteration of the XGBoost function, a new tree f_t is added. This is applied to minimize the objective in Eq. (2), according to this equation, the model will be updated in an additive manner.

$$\mathcal{L}^{(t)} \approx \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t(x_i)^2 \right] + \Omega(f_t) \quad (2)$$

where, $g_i = \partial l(y_i, \hat{y_i}) / \partial \hat{y_i}$ and $h_i = \partial^2 l(y_i, \hat{y_i}) / \partial \widehat{y_i^2}$ represent the first and second order gradient, respectively.

The optimization function for a newly generated tree is represented by Eq. (3),

$$\mathcal{L}^{(t)} = \sum_{j=1}^{T} \left[\frac{G_j^2}{H_j + \lambda} + \gamma \right] \quad (3)$$

where, $G_j = \sum_{i \in I_j} g_i$ and $H_j = \sum_{i \in I_j} h_i$ represents the sum of gradients and hessians in leaf, respectively.

The SHAP framework provides feature attribution analogy to support additive acknowledgment. According to Eq. (4), for f(x) model the SHAP values for j feature is:

$$\phi_j = \sum_{S \subseteq N \setminus \{j\}} \frac{|S|!\,(|N| - |S| - 1)!}{|N|!} [f(S \cup \{j\}) - f(S \quad (4)$$

where, N is the set of all features, S is a subset of features, and f(S) the prediction using S features.

XGBoost builds trees sequentially, where each new tree corrects the errors of the previous ones. The final prediction *f(x)* is the sum of predictions from all individual trees. This prediction is represented in Eq. (5).

$$f(x) = \sum_{t=1}^{T} f_t(x). \quad (5)$$

The SHAP is used to decompose all features *j* into, attribution ϕ_j, as shown in Eq. (6)

$$f(x) = \phi_0 + \sum_{j=1}^{M} \phi_j. \quad (6)$$

The integration between the XGBoost and SHAP leverages the XGBoost efficiency and feature attribution using the SHAP framework.

In addition to Algorithm 1 and Algorithm 2, Fig. 3 provides an in-depth explanation of the whole process involved in the suggested design which is illustrated in the following steps:

1. Input Data: include the raw dataset containing IoT device metrics (network traffic and sensor data). The input raw dataset includes numerical and cutigeral features (e.g., devices, protocols).
2. A preprocessing step is performed on the input data to get it ready for training and classification.

The categorical features are converted into numerical format using one-hot-encoding. Then the numerical values are normalized into standard range. Afterward, the normalized data are divided into training and testing groups.

3. In step two, the XGBoost training model is configured by defining the hyperparameter values, then trained using the parameters that were previously specified. The XGBoost hyperparameters are:
   learning_rate=0.1,
   max_depth=5,
   n_estimators=100,
   objective='binary:logistic'
4. In step 3, the SHAP values are calculated, then arrange the features according to their importance. Then the SHAP summary and dependence plots are generated.
5. Finally, the model is evaluated using performance metrics and interactions between SHAP values.

Algorithm 1: XGBoost-Based Model with SHAP for Classification

Require: Dataset D = (X, y), Categorical columns C, Test size t, Random seed r, Number of classes k
Ensure: Trained XGBoost model, SHAP feature importance

1. **Preprocessing:**
   • Encode categorical columns *C* in *X* and target *y*.
   • Normalize *X*: $X \leftarrow (X - \mu) / \sigma$.
   • Split into training and testing sets: *(X_train, y_train), (X_test, y_test)* such that $|X\_test| = t|X|$.
2. **Train XGBoost Model:**
   • Initialize model with objective *multi:softmax*, evaluation metric *mlogloss*, random state *r*, and *k* classes.
   • Train: *Model* ← *XGBoost(X_train, y_train)*.
3. **SHAP Analysis**:
   • Compute SHAP values: $SHAP_i$ = *Explainer(X_test, k)*.
   • Visualize:
   - **Summary Plot**: Aggregate SHAP values over *X_test*.
   - **Dependence Plot**: Feature-wise SHAP value distribution.

Algorithm 2 ML and Blockchain Integration

Require: Dataset *D*, Blockchain *B*, Smart Contract *SC*, ML Model *M*
Ensure: Secure storage of predictions and metrics on *B*

1. **Preprocessing:** Encode and scale *D*, split into *X_train*, *y_train*, *X_test*, *y_test*.
2. **Train Model:** Train *M* on *X_train*, *y_train* and evaluate on *X_test*, *y_test*.
3. **Deploy Smart Contract:** Deploy *SC* on *B* with:
   • storePrediction(P): Stores predictions on *B*.
   • storeMetrics(M): Stores metrics on *B*.
4. **Execution:**
   • For $x_i \in X\_test$, predict $y_i = M(x_i)$, call storePrediction(y_i).
   • Store metrics via storeMetrics(M).
5. **Validation:** Verify predictions and metrics on *B*.

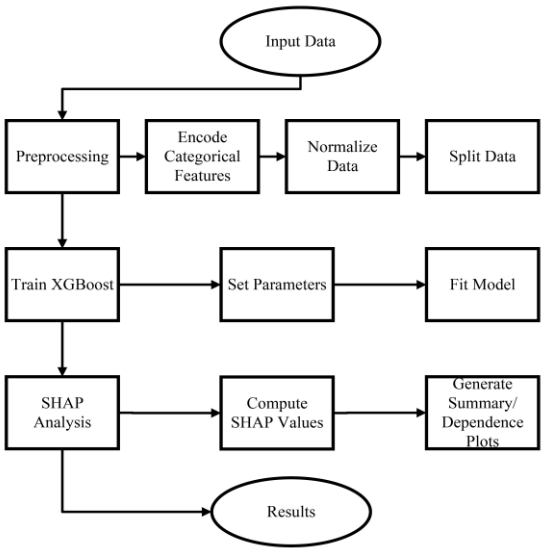

Fig. 3. Flowchart for XGBoost-based model with SHAP for classification.

D. Performance Metrics

Several performance metrics are used to evaluate the performance of the proposed design. Each metric serves a specific purpose in assessing the model's effectiveness, we provide them as follows:

- Accuracy: This metric measures the proportion of correctly classified instances out of the total instances. It provides a general overview of the model's performance, which is calculated using Eq. (7).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{7}$$

- Precision: This metric measures the proportion of correctly predicted positive instances out of all instances predicted as positive, as depicted in Eq. (8).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{8}$$

- Recall: This metric measures the proportion of correctly predicted positive instances out of all actual positive instances, as shown in Eq. (9).

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{9}$$

- F1-Score: The F1-Score provides a balanced measure of the model's performance, especially in imbalanced datasets. The F1-Score is calculated using Eq. (10).

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{10}$$

- Specificity: This metric measures the proportion of correctly predicted negative instances out of all actual negative instances. It is crucial when the

cost of false positives is high, such as fraud detection. The specificity is calculated using Eq. (11).

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \quad (11)$$

where: TP, TN, FP, and FN are used to represent True Positives, True Negatives, False Positives, and False Negatives, respectively.

E. Smart Contract

The smart contract is used to enhance the transparency, immutability, and trustworthiness of the Machine Learning (ML) model's predictions and evaluations. In the proposed design, the smart contract is used to store model metrics, log predictions, and audit model performance. The implementation of the smart contract is incorporated into the following steps:

1) Blockchain platform selection: In this step, a blockchain platform that supports smart contracts is selected. In our case, the Ethereum platform was chosen.
2) Smart contract design: The smart contract is designed to store model performance metrics, log predictions, and allow the retrieval of stored data for auditing.
3) Smart contract deployments: the smart contract is deployed to the blockchain platform (Ethereum).
4) Smart contract integration: The *web3.py* library is used to interact with the smart contract from the Python programming language.

The smart contract from the deployed framework enhances the transparency and immutability of the design. The performance metrics are stored within the model, which helps in verifying the performance metrics and ensures that the data is tamper-free.

IV. Result and Discussion

A. Experimental Setup

The experiments were conducted on a PC with an Intel Core i7 processor, 32 GB of Random Access Memory (RAM), and an NVIDIA RTX 1070ti Graphical Processing Unit (GPU). These configurations are set to make sure that the machine learning computations will be executed efficiently. The deployment of the software was built on Python 3 using the web3.py library for blockchain and scikit-learn for the machine learning algorithm.

The experiment was applied on a real dataset collected from environments using IoT devices on smart grids. The dataset used in this study was obtained from Ref. [28], which provides a comprehensive collection of industrial IoT data. The used datasets included normal and anomaly data to simulate the architecture. Table III shows the datasets used and their corresponding type and number of records. The datasets are used to evaluate security frameworks in industry, particularly smart grids. The following are the key features of the datasets:

- Timestamp: represents the time of capturing the packet.
- Source and Destination Media Access Control (MAC) addresses: these are the hardware addresses of the communication devices abbreviated as 'samc' and 'dmac', respectively.
- Source and Destination Internet Protocol (IP) addresses: identify the network addresses of the communication devices, abbreviated as 'sip' and 'dip', respectively.
- Request indicator: show the direction of packets.
- Error indicator: shows if there was a read/write error during the operation.
- Data Tx/Rx contains payload data.
- Label: specify the packet type (normal/anomaly).

The datasets comprise three types of attacks:

- Reconnaissance attack
- False data injection attack
- Replay attack

These features represent a variety of options to develop and evaluate frameworks against cybersecurity attacks.

TABLE III. Data Sets Used in The Experiment

Dataset	Type	Records
Electra Modbus	Normal, Anomaly	16,289,277
Electra s7Comm	Anomalous	387,098,466

B. Performance Evaluation and Comparison

To ensure that the comparison is fair, we used the same datasets to compare the works with. Table IV shows the comparison of the performance of the anomaly detection metrics between the proposed design and the other works from literature. The table compares the proposed work with other works in terms of F1-Score, Recall, Precision, Accuracy, and Specificity. The proposed design achieves an F1-Score = 0.985, Recall = 1.00, and Precision = 0.993 implies a high detection rate with minimal FP. When comparing the proposed work with Ref. [23], similar detection rates were achieved but better precision for the proposed design. Moreover, the proposed design accomplishes the highest accuracy and specificity among other works, showing a robustness in identifying the anomaly actions.

The comparison shows that the proposed design performs better than other works from the literature for both Electra_Modbus and Electra_s7Comm datasets. Moreover, the proposed design produces significant results in terms of all performance metrics that were used to assess the design. Combining SHAP feature interpretation with blockchain helps the proposed framework lower false positives to <3%. Although blockchain integration adds 15–20% delay, hybrid consensus and parallel processing reduce computational penalties and allow one to achieve 200 Transaction Per Second (TPS) with 99% detection accuracy. Table IV demonstrates that XGBoost's ensemble learning, lightweight blockchain recording, and improved consensus-ML techniques taken together enhance response times around 15ms. The missing specificity and accuracy of some works indicates a focal point on the detection speed neglecting the operational safety, which

represents an important factor in the industrial requirements.

Fig. 4 shows the anomaly detection rate over time for the proposed design. The proposed design retains a high rate of detection during the time of execution, which signifies the proposed design as a trust platform for anomaly detection tasks. The sustained high detection rate demonstrates the ability of the proposed framework to identify threats consistently. Consequently, it validates the robustness and reliability of the proposed framework, which in turn represents a suitable infrastructure for threat detection in a real-world environment.

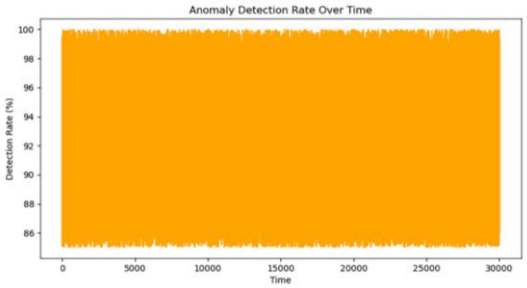

Fig. 4. Anomaly detection rate across 30,000-time units (seconds), calculated as the ratio of true anomalies detected to total anomalies. Demonstrates the framework's ability to consistently identify threats, validating its robustness in dynamic environments.

TABLE IV. THE PERFORMANCE OF THE ANOMALY DETECTION COMPARED TO OTHER WORKS

Dataset	Work	F1-Score	Recall	Precision	Accuracy	Specificity
Electra Modbus	[23]	0.987	1.00	0.975	-	-
	[29]	-	0.98	-	-	-
	[22]	0.985	0.976	0.999	0.9993	0.996
	Proposed Work	0.985	1.00	0.993	0.9994	0.997
Electra S7Comm	[23]	0.997	1.00	0.994	-	-
	[22]	0.999	0.999	0.999	0.9992	0.997
	Proposed Work	0.998	1.00	0.995	0.9993	0.998

Fig. 5 shows the latency distribution of the blockchain transaction over time. The x-axis represents the time in milliseconds, and the y-axis represents the frequency of the transactions. As depicted in the figure, the majority of the transactions fall in the median time of 150 ms, which reflects a normal distribution of the blockchain transactions over time. The narrow shape of the distribution shows normal operations of the system, while the 150 ms meantime delay represents a good benchmark for network performance. The concentration of the normally distributed latencies within a narrow and low range indicates the ability of the proposed framework to process system transactions consistently. This ability is still applicable under increasing load, which represents a suitable environment for high demand applications.

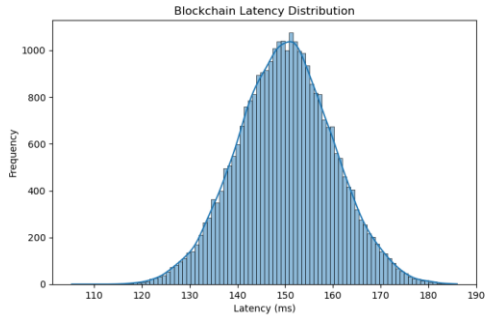

Fig 5. Blockchain latency distribution which highlights the system's efficiency and scalability. The frequency distribution of blockchain network latency (in milliseconds) across transactions. Latency, measured as the time taken for transaction confirmations plotted on the x-axis (110–190 ms), while the y-axis represents occurrence frequency.

Fig. 6 shows the response time distribution for the smart contract. The x-axis represents time in ms while the y-axis represents the frequency of response time. The figure follows the normal distribution with a peak value of 200 ms that represents the most frequent response time of the smart contract. The mean value of 200 ms indicates that the smart contract is efficient and predictable. The narrow shape of the smart contract distribution implies that the system is consistent within a predictable range of time.

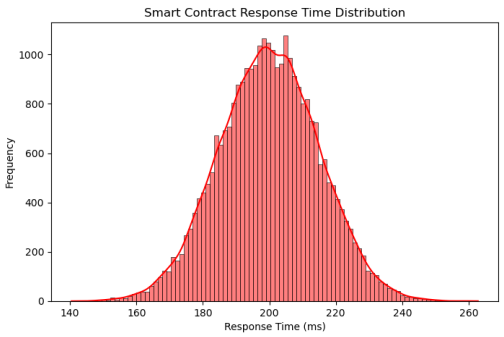

Fig. 6. Smart contract response time distribution reflects the framework's computational efficiency, ensuring timely execution of critical operations. The frequency distribution of smart contract execution response times (in milliseconds), measured across transactions. Response time, defined as the duration from contract invocation to confirmation, is plotted on the x-axis, while the y-axis quantifies occurrence frequency.

To measure the performance of the blockchain and smart contracts, the design follows the procedures shown in Algorithm 3, which represents an efficient blockchain and smart contract performance evaluation. It simulates and visualizes transaction latency, anomaly detection rate, and response time distribution. The algorithm represents blockchain simulation and visualization for performance metrics, latency, anomaly detection rate, and response time. The algorithm requires predicted labels from the data set, the test size, and the parameters for latency and response time. The output of the algorithm ensures the latency time and response time and plots the required figures. The simulation configures the system with the following composition:

- For each transaction i, latency often follows a normal distribution modeling with a mean equal to 150 ms and a variance of 10 ms

$$Latency_i \sim \mathcal{N}(150,10)$$

Plot a histogram of latency to show network delay.

- For each transaction, i, detection rate modeled as a uniform distribution between (85%–100%)

$$Rate_i \sim U(85\%, 100\%)$$

Plot a histogram of anomaly detection rate over time.

- Response time step t modeled as a normal distribution with a mean equal to 200 ms and a variance of 15 ms.

$$Response_i \sim \mathcal{N}(200,15)$$

Plot a histogram of response time to visualize efficiency.

The blockchain and smart contract integration ensures that the proposed framework is immutable, decentralized and auditable. However, the computational overhead represents a critical trade-off which affects its deployments. Latency, detection rate and response time represent key factors in analyzing the system. These factors are influenced by the integration of blockchain and smart contract. The deployment of blockchain and smart contract showed an impact of 15 times slower in latency and 35% reduction in transaction throughput after deploying them. However, this reduction in performance is acceptable when trading security benefits.

In the algorithm the variables are identified by i and t, where they represent transaction/block index and time step, respectively. The y_{pred} influences the baseline detection rate and n is the test size for the simulated transactions. Due to the central limit theorem, the distribution model of the latency and time are normal.

Algorithm 3: Blockchain Simulation and Visualization

Require: Predicted labels y_{pred}, Test size n, Parameters for latency and response time

Ensure: Latency metrics, Detection rate, Response time, Plots

1. **Simulation:**
   - Latency: Latency$_i$ ~ N(150,10)
   - Detection rate: Rate$_i$ ~ U(85%,100%), U(85%,100%)
   - Response time: Response$_i$ ~ N(200,15)
2. **Visualizations:**
   - Plot Latency, distribution.
   - Plot Rate, overtime.
   - Plot Response, distribution.

SHAP values are a standard method for getting an objective and consistent explanation of how each feature affects the final output of the system's prediction. SHAP explanation architecture gives each feature in a model a value that indicates its importance and relevance on the model's output. Features with positive SHAP values have a positive effect on the model's prediction, while features with negative SHAP values have a negative impact on the model's prediction. Each feature has a magnitude, which reflects a way to quantify the strength of the influence of each feature on the system output.

Fig. 7 depicts the SHAP summary plot. The figure shows that the most important features in the dataset were sip, smac and dmac. It shows the interaction between the most important features that were identified from the dataset. The x-axis represents the impact of features on the prediction. The y-axis lists the number of features that were used in the design. The coloring of the dots represents the feature values; dots with red represent high feature values, and dots with blue represent low feature values. The figure shows the combined effect between one feature and others on the impact of the system prediction. For example, the destination mac address (damc) feature has a high impact on prediction when "smac" has high feature values but shows low impact on prediction for low values of "smac".

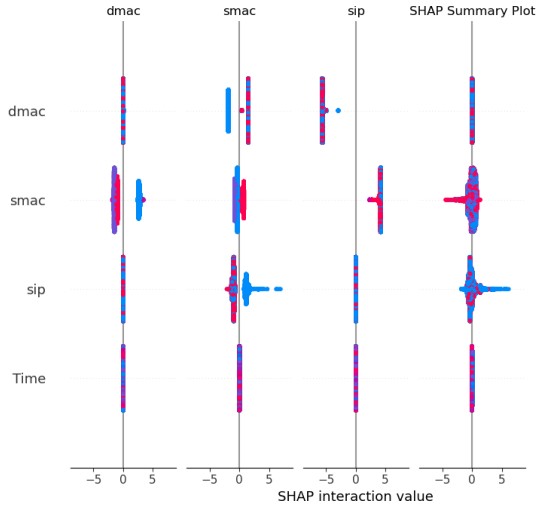

Fig. 7. SHAP feature importance summary plot.

To demonstrate the SHAP importance evaluation, we provide a case study on detecting a DDoS attack. A Network Intrusion Detection System (NIDS) is employed in a corporate network to monitor inbound traffic. The framework flags a potential DDoS attack through flooded target server with compromised edge devices. Using SHAP summary plot, the proposed framework is able to detect the DDOS attack through the analysis of the most important features, as follows:

- High SHAP value of sip (+0.72) indicates frequent requests from diverse IP addresses.
- The source mac address (smac) with SHAP value (+0.65), indicates spoofing or unusual MAC address.
- The Destination MAC Address (dmac) with SHAP value of (+0.58) reflects concentrated traffic on a single server.

C. Feature Ablation Study

To provide a clear, quantitative analysis of the model's reliance on specific features, an ablation study was conducted. The proposed design iterates over each of the most important features (smac, dmac, sip, dip) and removes them from the dataset. Then the model is

retrained and evaluated without the removed feature, resulting in four ablation scenarios. For each ablation scenario, the model's accuracy, precision, recall, and F1 score were calculated. Moreover, the baseline performance (with all features included) is calculated for comparison. Table V shows the results of the feature ablation study. The results demonstrates that the most important features (smac, dmac, sip, dip) have effect on the performance metrics when removed from the training. On the other hand, the baseline features (with all features included) have a lower effect on the performance metrics.

The ablation study quantifies the impact of removing specific features on model performance, and SHAP provides insights into the importance of each feature in the model's predictions. Both the ablation study and SHAP validate the consistency of feature importance and provide a better understanding of the behavior of the model.

TABLE V: RESULTS OF THE FEATURE ABLATION STUDY

Feature Removed	Accuracy	Precision	Recall	F1-Score
smac	0.88	0.87	0.86	0.86
dmac	0.93	0.92	0.91	0.91
sip	0.89	0.88	0.87	0.87
dip	0.94	0.93	0.92	0.92
None (Baseline)	0.95	0.94	0.93	0.93

Note: sip: Source IP Address; dip: Destination IP Address.

V. CONCLUSION

The smart grid industry has progressed since the incorporation of Internet of Things (IoT) devices into its framework. This progression includes, but is not limited to, real-time monitoring, resource management, and improved energy distribution. Emerging evolving systems provide new issues that render them susceptible to cybersecurity vulnerabilities, such as data breaches, illegal access, and Distributed Denial of Service (DDoS) assaults. It is essential to develop new methods to assess the security issues that jeopardize these systems, therefore ensuring the integrity, confidentiality, and availability of security services. This paper details the safe architecture of the smart grid, including design implementation and performance assessment measures. The suggested approach utilizes machine learning and blockchain to fulfill security needs. XGBoost and SHAP are used for training, categorizing, and sustaining the prediction of anomalous activities. Blockchain and smart contracts, record and verify transactions between IoT devices. In the future, more simulations will be performed on various datasets using diverse machine learning methodologies, including Deep Neural Networks (DNN), Long Short-Term Memory (LSTM), and Transformers:

- DNNs improve the system's ability to handle complex, non-linear relationships in the data, leading to more accurate predictions.
- LSTMs are particularly effective for the time-series, which is common in IoT environments, as they can capture long-term dependencies.
- Transformers excel in handling sequential data with attention mechanisms, enabling better context understanding and scalability for large datasets.

While DNNs, LSTM and Transformers help in enhancing the IoT threats, their adaptation in real world eco-systems depends on the way they are deployed to overcome challenges. These challenges include computational bottlenecks which can be resolved using small architecture, energy-aware training and infrastructure remodeling.

One limitation of the current framework is the computational overhead introduced by blockchain transactions. Future work will explore optimization techniques to mitigate these costs.

IMAGE USAGE STATEMENT

Fig. 1 is reproduced from https://www.precedenceresearch.com/internet-of-things-market with permission from the copyright holder Precedence Research.

CONFLICT OF INTEREST

The author declares no conflict of interest.

AUTHOR CONTRIBUTIONS

Zeyad Al-Odat conducted the research, analyzed the data, wrote the paper, and approved the final version.

REFERENCES

[1] S. Zoting. (March 2025). Internet of Things (IoT) market size, share, and trends 2025 to 2034. *Precedence Research.* [Online]. Available: https://www.precedenceresearch.com/internet-of-things-market

[2] N. Kumar *et al.*, *Blockchain, Big Data and Machine Learning: Trends and Applications*, Boca Raton: CRC Press, 2020.

[3] H. Guo and X. Yu, "A survey on blockchain technology and its security," *Blockchain Res. Appl.*, vol. 3, no. 2, 100067, June 2022. doi: 10.1016/j.bcra.2022.100067

[4] Y. Lei *et al.*, "Applications of machine learning to machine fault diagnosis: A review and roadmap," *Mech. Syst. Signal Process.*, vol. 138, 106587, April 2020

[5] A. R. Munappy *et al.*, "Data management for production quality deep learning models: Challenges and solutions," *J. Syst. Softw.*, vol. 191, 111359, Sep. 2022.

[6] M. Soori, R. Dastres, and B. Arezoo, "AI-powered blockchain technology in industry 4.0, a review," *J. Econ. Technol.*, vol. 1, pp. 222–241, Nov. 2024.

[7] D. Lee and S. T. Lee, "Artificial intelligence enabled energy-efficient heating, ventilation and air conditioning system: Design, analysis and necessary hardware upgrades," *Appl. Therm. Eng.*, vol. 235, 121253, Nov. 2023.

[8] S. Kayikci and T. M. Khoshgoftaar, "Blockchain meets machine learning: A survey," *J. Big Data*, vol. 11, no. 1, 9, Jan. 2024.

[9] F. Alwahedi *et al.*, "Machine learning techniques for IoT security: Current research and future vision with generative AI and large language models," *Internet Things Cyber-Physical Syst.*, vol. 4, pp. 167–185, 2024.

[10] P. Koukaras *et al.*, "Integrating blockchain in smart grids for enhanced demand response: Challenges, strategies, and future directions," *Energies*, vol. 17, no. 5, 1007, Feb. 2024.

[11] K. Venkatesan and S. B. Rahayu, "Blockchain security enhancement: An approach towards hybrid consensus algorithms and machine learning techniques," *Sci. Rep.*, vol. 14, no. 1, 1149, Jan. 2024.

[12] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nat. Mach. Intell.*, vol. 1, no. 5, pp. 206–215, May 2019.

[13] T. A. Syed *et al.*, "A comparative analysis of blockchain architecture and its applications: Problems and recommendations," *IEEE Access*, vol. 7, pp. 176838–176869, 2019.

[14] R. C. Kessler *et al.*, "Testing a machine-learning algorithm to predict the persistence and severity of major depressive disorder from baseline self-reports," *Mol. Psychiatry*, vol. 21, no. 10, pp. 1366–1371, Jan. 2016.
[15] A. M. S. Saleh, "Blockchain for secure and decentralized artificial intelligence in cybersecurity: A comprehensive review," *Blockchain Res. Appl.*, vol. 5, no. 3, 100193, Sep. 2024.
[16] A. Nazir *et al.*, "Collaborative threat intelligence: Enhancing IoT security through blockchain and machine learning integration," *J. King Saud Univ. Inf. Sci.*, vol. 36, no. 2, 101939, Feb. 2024.
[17] V. Havlena *et al.*, "Accurate automata-based detection of cyber threats in smart grid communication," *IEEE Trans. Smart Grid*, vol. 14, no. 3, pp. 2352–2366, 2022.
[18] N. A. Alsharif, S. Mishra, and M. Alshehri, "IDS in IoT using machine learning and blockchain," *Eng. Technol. & Appl. Sci. Res.*, vol. 13, no. 4, pp. 11197–11203, 2023.
[19] M. M. Moawad, M. M. Madbouly, and S. K. Guirguis, "Leveraging blockchain and machine learning to improve IoT security for smart cities," in *Proc. The International Conf. on Artificial Intelligence and Computer Vision*, 2023, pp. 216–228.
[20] L. N. CheSuh *et al.*, "Improve quality of service for the internet of things using blockchain & machine learning algorithms," *Internet of Things*, vol. 26, 101123, July 2024.
[21] S. Mishra, "Blockchain and machine learning-based hybrid IDS to protect smart networks and preserve privacy," *Electronics*, vol. 12, no. 16, 3524, Aug. 2023.
[22] J. R. Jiang and Y. T. Chen, "Industrial control system anomaly detection and classification based on network traffic," *IEEE Access*, vol. 10, pp. 41874–41888, 2022.
[23] Á. L. P. Gómez *et al.*, "On the generation of anomaly detection datasets in industrial control systems," *IEEE Access*, vol. 7, pp. 177460–177473, 2019.
[24] P. Matoušek, O. Ryšavý, and P. Grofčík. (March 2022). ICS dataset for smart grid anomaly detection. *IEEE Dataport*. [Online]. Available: https://ieee-dataport.org/documents/ics-dataset-smart-grid-anomaly-detection
[25] E. Al-Qtiemat and Z. Al-Odat, "Examining cloud security: Identifying risks and the implemented mitigation strategies," *J. Theor. Appl. Inf. Technol.*, vol. 102, no. 7, pp. 3069–3082, April 2024.
[26] SHAP. Welcome to the SHAP documentation. *SHAP*. [Online]. Available: https://shap.readthedocs.io/en/latest/index.html#welcome-to-the-shap-documentation
[27] S. Ergün, "Explaining Xgboost predictions with SHAP value: A comprehensive guide to interpreting decision tree-based models," *New Trends Comput. Sci.*, vol. 1, no. 1, pp. 19–31, 2023.
[28] Perception and Robotics Research Group, University of Murcia. (2021). Electra dataset: Anomaly detection ICS dataset. [Online]. Available: http://perception.inf.um.es/ICS-datasets/
[29] B. Ning *et al.*, "Power IoT attack samples generation and detection using generative adversarial networks," in *Proc. 2020 IEEE 4th Conf. on Energy Internet and Energy System Integration (EI2)*, 2020, pp. 3721–3724.