Received 19 October 2023; revised 17 November 2023; accepted 21 November 2023. Date of publication 23 November 2023; date of current version 8 December 2023. The review of this paper was arranged by Associate Editor Thomas I. Strasser.

Digital Object Identifier 10.1109/OJIES.2023.3336234

Double-Edged Defense: Thwarting Cyber Attacks and Adversarial Machine Learning in IEC 60870-5-104 Smart Grids

HADIR TERYAK, ABDULLATIF ALBASEER (Member, IEEE), MOHAMED ABDALLAH (Senior Member, IEEE), SAIF AL-KUWARI (Senior Member, IEEE), AND MARWA QARAQE (Senior Member, IEEE)

Division of Information and Computing Technology, College of Science and Engineering, Hamad Bin Khlifa University, Doha 34110, Qatar

CORRESPONDING AUTHOR: ABDULLATIF ALBASEER (e-mail: aalbaseer@hbku.edu.qa)

This work was supported by NPRP Cluster Project (NPRP-C) Twelve (12th) Cycle under Grant NPRP12C-33905-SP-67 through the Qatar National Research Fund (a member of Qatar Foundation).

ABSTRACT Smart grids (SGs), a cornerstone of modern power systems, facilitate efficient management and distribution of electricity. Despite their advantages, increased connectivity and reliance on communication networks expand their susceptibility to cyber threats. Machine learning (ML) can radically transform cyber security in SGs and secure protocols as in IEC 60870 standard, an international standard for electric power system communication. Notwithstanding, cyber adversaries are now exploiting ML-based intrusion detection systems (IDS) using adversarial ML attacks, potentially undermining SG security. This article addresses cyber attacks on the communication network of SGs, specifically targeting the IEC 60870-5-104 protocol. We introduce a novel ML-based IDS framework for the IEC 60870-5-104 protocol. Specifically, we employ an artificial neural network (ANN) to analyze a new and realistically representative dataset of IEC 60870-5-104 traffic data, unlike previous research that relies on simulated or unrelated data. This approach assists in identifying anomalies indicative of cyber attacks more accurately. Furthermore, we evaluate the resilience of our ANN model against adversarial attacks, including the fast gradient sign method, projected gradient descent, and Carlini and Wagner attacks. Our results demonstrate that the proposed framework can accurately detect cyber attacks and remains robust to adversarial attacks. This offers efficient and resilient IDS capabilities to detect and mitigate cyber attacks in real-world ML-based adversarial environments.

INDEX TERMS Adversarial attacks, deep learning, IEC 60870-5-104 protocol, intrusion detection systems (IDS), machine learning (ML), smart grids (SGs).

I. INTRODUCTION

The increasing electricity demand is exhausting the current power systems, which lack reliability, efficiency, and automation. This motivates the move toward the smart grid (SG), which can potentially solve many of the challenges in the existing power grid by providing bidirectional information flow between different power system components [1]. SG is designed to improve the efficiency and reliability of the traditional power grid by integrating advanced technologies for power generation. Unlike conventional power grids, the bidirectional flow of power and information enables power generation at the consumer level using renewable energy sources and storing excess energy for future use. Thus, communication technologies play a pivotal role in the functioning of the SGs. They interconnect various grid components and Internet of Things devices, such as smart meters, which generate vast volumes of data for automated decision-making and dynamic energy management. In effect, the communication infrastructure serves as the backbone of the SG, enabling all the data exchange and processing that are vital for SG operations [2].

In this context, the International Electrotechnical Commission (IEC) 60870 standard is particularly important to streamline integration and enhance interoperability between different devices and systems within the SG [3], [4]. This facilitates the use of various networking technologies in the grid, such as Ethernet, Wi-Fi, and cellular networks [5]. However, IEC 60870, and specifically the IEC 60870-5-104 protocol, introduces several vulnerabilities that must be addressed. Adversaries can access to numerous vulnerabilities, including ways to launch destructive attacks and access sensitive information. As an example of such attacks, the 2015 Ukrainian power outage illustrates the critical vulnerabilities to cyber attacks in both the control center and the smart devices employed for managing and observing the electrical system [6]. Specifically, IEC 60870-5-104 is susceptible to standard network attacks, such as denial of service (DoS), man-in-the-middle (MITM), and other forms of cyber attacks. In addition, using IP networks in IEC 60870-5-104 inadvertently expands the attack surface that cyber attackers often target. Notably, the utilization of widely recognized protocols, such as IEC 60870-5-104, could simplify unauthorized access for attackers who are familiar with these insecure protocols. Moreover, the presence of older devices lacking robust security features within the grid could constitute a potential weak point. Consequently, it is critical to focus on enhancing the security mechanisms within IEC 60870 to strengthen the SG's resilience against cyber threats [7], [8].

Machine learning (ML), as a rapidly growing technology, can play a crucial role in detecting cyber attacks on SGs [2], [9], [10], [11]. ML algorithms can be trained to identify patterns and anomalies in large volumes of data generated by the grid using either anomaly-based or signature-based detection methods [12]. These algorithms can analyze normal network traffic and energy consumption patterns, creating a baseline of "normal" behavior [13], [14], [15]. When an attacker tries to infiltrate the network or conduct a disruptive operation, the ML algorithm can quickly recognize this unusual activity as it deviates from the established baseline. Once an anomaly is detected, alerts can be triggered for further investigation and prompt response. This proactive approach enables real-time detection and prevention of cyber threats, significantly enhancing the security of overall SG systems [16].

A. CONTRIBUTIONS

Despite the increasing use of the IEC 60870-5-104 protocol in SG and its crucial role in several industries, there is a shortage of efficient and resilient intrusion detection systems (IDS) capable of identifying and counteracting cyber attacks directed toward IEC 60870-5-104-based systems. In addition, the lack of realistic, publicly available datasets for this protocol hampers the development and testing of ML models tailored for IDSof IEC 60870-5-104 communication. Furthermore, robust ML models that can resist adversarial attacks—a key consideration in cyber security—are scarcely explored in the existing literature. These gaps highlight a critical need for thorough studies and approaches to harness ML's potential in enhancing IEC 60870-5-104's security, fueling the development of robust IDS, and reinforcing the resilience of SGs against sophisticated cyber threats. This motivates the need for robust IDS that can secure the IEC 60870-5-104 communication between different components within SGs. This article contributes to these challenges, and our contributions can be summarized as follows.

1) Addressing the lack of a real IEC 60870-5-104 dataset, which limits the research community's ability to develop effective security measures for the IEC 60870-5-104 protocol. We provide a clean and labeled dataset that captures information from the IEC 60870-5-104 header, enabling the training of ML-based IDS.
2) Proposing a novel approach for IDS in IEC 60870-5-104-based systems using advanced ML techniques. The proposed framework can detect and mitigate up to 11 cyber attacks using a hierarchical approach to differentiate between legitimate and malicious network traffic and, providing an efficient familial analysis of the attack type. This approach enables a rapid and effective response to cyber attacks.
3) Handling the issue of the lack of a robust IEC 60870-5-104 IDS that can maintain a high positive detection rate in the presence of ML-based adversarial attacks. The proposed IEC 60870-5-104 IDS is tested against several ML adversarial attacks, such as the Fast Gradient Sign Method (FGSM), Projected Gradient Descent (PGD), and Carlini and Wagner (C&W).
4) Contributing to the open-source research community by making the dataset and all the associated training codes available for public access. This enables other researchers to validate, reproduce, and build upon our work. The resources can be accessed with a detailed description via the following GitHub link.[1]

B. ORGANIZATION

The rest of this article is organized as follows. In Section II, we provide a brief overview of the existing related work. In Section III, we overview the dataset. In Section IV, we describe our system architecture and explain the methodology used. Moreover, in Section VI, we illustrate our experimental results and summarize. Finally, Section VII concludes this article.

II. RELATED WORK

In the literature, several studies have explored the use of ML algorithms for IDS in SGs. Some effort in the literature was devoted to generating security datasets for SG. For example, Babu et al. [17] presented a framework called Melody that mitigates the security risk associated with SG datasets through simulation and emulation. The melody framework emulates the normal traffic and the intrusion attacks on a simulated SG network to mimic a real SG network dataset. Melody uses power world to replace the electrical features of the SG and Mininet as the communication network between emulated SG components, such as supervisory control and data acquisition (SCADA), remote terminal unit, and intelligent

[1] https://github.com/Abdullatif2/Double-Edged-Defense

electronic device (IED). The framework was evaluated regarding scalability, temporal accuracy, and traffic reply capability. In [18], combined deep neural networks (NNs) with honeypots were proposed to generate Modbus data traffic. Their Neuralpot used generative adversarial networks (GANs) and autoencoders, with the Conpot, a honeypot widely used for industrial applications. Neuralpot could generate data with high similarity in a short time. Ahmed and Kandasamy [19] built an electrical grid testbed called EPIC to generate datasets for ML. The testbed consisted of four zones: generation, MicroGrid, transmission, and smart Home. They distributed IEDs across all four zones and collected current, voltage, and frequency measurements. They simulated the electrical layout and communication network between SG components. The data was collected using a different configuration of the EPIC and by implementing different attack scenarios.

Considering the ML algorithms to secure the SG, the work in [20], for instance, used the dataset (KDDCUP99) to train ML-based IDS. They used principal component analysis (PCA) for dimensionality reduction and compared ML algorithms, such as support vector machine (SVM), decision trees (DT), and Naive Bayesian. According to [21], particle swarm optimization (PSO) can be used to optimize feature selection. They performed feature selection optimization and preprocessing on the published datasets KD99 and NSLKDD to improve the detection accuracy of the trained ML models. Using PSO, they selected seven out of 41 features in the NSLKDD dataset and 8 out of 41 in the KDD99 dataset. Khan et al. [21] trained binary and multiclass classifiers using K-nearest neighbor (k-NN), NNs, DTs, and random forest (RF) trained models. They evaluated their models based on metrics, such as f1-Score, accuracy, recall, and precision, to prove the efficiency of using PSO on these datasets. Likewise, Ustun et al. [22] focused on intrusion detection within SGs, utilizing a range of ML algorithms, including DT, RF, SVM, k-NN, and adaptive boost (AdaBoost). In addition, Aziz et al. [23] claimed that hybrid models can help with false data injection (FDI) attacks. They used a public SG dataset and trained hybrid classifiers. such as SVM with extreme gradient boosting (XGBoost) classifier, gradient boosting classifier, categorical boosting (CatBoost) classifier, AdaBoost classifier, and SVM with the light-gradient and histogram boosting classifiers. Their results show that SVM combined with the CatBoost classifier gives the highest accuracy among all the tested hybrid models. On the other hand, Nowroozi et al. [24] tested the transferability of adversarial attacks between models. They showed that the malicious attack could be transferred between ML models even if they are trained on different datasets or have different architectures. The work in [25] offered a guide on model selection based on the confidentiality, integrity, and availability (CIA) triad, covering both traditional and deep learning models. A survey by Alkuwari et al. [26] listed some existing anomaly detection methods, particularly against FDI, DoS, and load drop attacks, while also discussing data privacy and the advantages of hybrid methods. Wang et al. [27] introduced a stacked deep learning approach and highlighted the effectiveness of XGBoost for SCADA systems. The work in [28] proposed the binary gray wolf optimization evolutionary computation (BGWO-EC) scheme, demonstrating its applicability across diverse attack scenarios. Lastly, Yu et al. [29] presented the gray wolf algorithm (GWA) artificial neural network (ANN) model, which employs GWA and outperforms conventional algorithms when tested on the MSU/ORNL datasets. Table 1 summarizes the advantages and disadvantages of the various methodologies discussed in this section.

In summary, despite the considerable efforts in the literature, there is a clear shortfall in addressing the vulnerabilities of IEC 60870-5-104 communication. Furthermore, there is a significant gap in applying realistic IEC-specific datasets for testing and developing these models. Also, none of these studies substantively test or prove the resilience of their proposed models against adversarial intrusions. To address these pressing concerns, our study focuses on developing a robust IDS specifically tailored for IEC 60870-5-104. Our approach includes the use of a realistic, labeled IEC 60870-5-104 dataset and validating the IDS's resilience against adversarial intrusions, offering a thorough solution to IEC 60870-5-104 communication security.

III. INDUSTRIAL CONTROL SYSTEM (ICS)INTRUSION DETECTION DATASET PREPARATION AND ANALYSIS

In IDS, the utilization of real-world datasets is instrumental to the development, testing, and evaluation of proposed models. In March 2022, Brno University published a streaming ICS dataset capturing information from two headers, IEC 60870-5-104 and IEC 61850 MMS, which are commonly used for anomaly detection and security monitoring [30]. In this work, we mainly focus on IEC 60870-5-104 related data. The data was either captured from real ICS devices' communication or virtualized from ICS applications. The data spanned across one or multiple days, including benign and malicious samples with attacks, such as switching, scanning, and communication interruption. However, this dataset has no labels and needs to be preprocessed to be appropriate for commercial or research purposes. Labeling, cleaning, and preprocessing datasets is a crucial step in the data preparation process for many types of analysis. This section details our process of preparing this dataset, including assigning appropriate labels to data samples, eliminating irrelevant or inaccurate data, and transforming data to a format conducive to subsequent analysis (i.e., encoding).

A. OVERVIEW OF THE UTILIZED IDS DATASET AND PREPROCESSING STEPS

The IDS dataset exhibits a well-structured folder hierarchy, complemented by informative readme.txt files outlining data types and attack timestamps. Table 2 provides an overview of the dataset's folder arrangement, while Table 3 delineates the attributes extracted from the IEC 60870-5-104 header. Notably, this dataset encompasses 11 distinct types of attacks, each capable of exerting diverse effects on the SG system.

TABLE 1. Summary of Advantages and Disadvantages of Related Works

Related work	Advantages	Disadvantages
Melody Framework [17]	Emulation and simulation of SG networks. Evaluate scalability, temporal accuracy, and traffic reply capability.	May lack real-world applicability. No countermeasure against adversarial attacks.
Neuralpot [18]	Generates high-similarity data quickly using GANs and Autoencoders. Integrates with Conpot honeypot for industrial applications.	Specialized for Modbus data traffic. No countermeasure against adversarial attacks.
EPIC Testbed [19]	Coverage of SG zones for data collection. Real-world scenarios simulated.	Limited validation details. No countermeasure against adversarial attacks.
Using KDDCUP99 with PCA [20]	Uses PCA and multiple ML algorithms.	Relies on older datasets (KDDCUP99). No countermeasure against adversarial attacks.
PSO [21]	Optimize feature selection.	Uses older datasets. No countermeasure against adversarial attacks.
Hybrid Models [23]	Tackles FDI attacks. Multiple hybrid classifiers.	Specificity limits general applicability. No countermeasure against adversarial attacks.
Transferability of Attacks [24]	Insights into attack transferability between ML models.	Uses irrelevant datasets. No countermeasure against adversarial attacks.
Berghout et al. [25]	Guide on model selection based on CIA triad. Covers traditional and deep learning models.	Consider a simple attack scenario. No countermeasure against adversarial attacks.
Wang et al. [27]	Stacked deep learning approach. Highlights XGBoost effectiveness.	irrelevant datasets. No countermeasure against adversarial attacks.
Panthi et al. [28]	Proposes BGWO-EC scheme. Effective across diverse attack scenarios.	High-complexity. No countermeasure against adversarial attacks.
Yu et al. [29]	Presents GWA-ANN model. Uses grey wolf optimization. Outperforms conventional algorithms on MSU/ORNL datasets.	No consideration for limited-resource devices. No countermeasure against adversarial attacks.

1) *DoS attack:* Grid components may become unresponsive, leading to potential power outages, delays in response to grid events, and reduced overall system reliability.
2) *Injection attack:* It can lead to unauthorized control over SG components, potentially affecting grid stability and security. For example, changing setpoints or control commands could result in equipment damage or grid instability.
3) *Connection-loss attack:* It can lead to a lack of situational awareness, delayed response to grid events, and potentially cascading failures as grid components cannot coordinate effectively.
4) *Switching attack:* It can lead to erratic device behavior, potentially causing grid instability, overloads, or unsafe operating conditions.
5) *Scanning attack:* Attackers may gain insight into the SG's structure and vulnerabilities, potentially enabling future attacks or unauthorized access to critical components.
6) *Rogue device attack:* It can lead to false data being injected into the SG, potentially causing incorrect decisions and actions by control systems.
7) *MITM attack:* It can lead to unauthorized manipulation of data and control commands, potentially compromising grid stability and security.
8) *Value change attack:* Altered values may damage equipment, grid instability, or unsafe operating conditions.
9) *Masquerading attack:* It can introduce false data into the system, leading to incorrect decisions and actions by control systems. This could impact grid stability, reliability, and the overall integrity of data.
10) *Report-block attack:* These attacks can disrupt the flow of critical information, potentially leading to a lack of situational awareness and delayed responses to grid events.
11) *Replay attack:* It can deceive the SG's systems into making incorrect decisions or taking actions based on duplicated data.

B. DATASET LABELING

The Raw data from Brno University was not labeled. However, the timestamp for the attack occurrence was mentioned in the dataset description. Hence, based on the time stamp, we performed the needed preprocessing to label the data, which included two main tasks: timestamp processing and adding the day feature.

1) *Timestamp processing:* We converted the timestamp from HH:MM: SS format to HH.MMSS Format. This facilitates the numerical comparison between timestamps and allows accurate labeling of the exact time when the attack occurred.
2) *Day feature:* The captured IEC 60870-5-104 data are spanned across multiple days, meaning that the timestamp will be repeated. Hence, if we only consider the timestamp as our only reference for the attack timing, an error will occur in labeling. To avoid this, we calculated the day feature from the relative time, which is the logical time that starts with the traffic capturing. Day $= \frac{\text{relative time}}{86\,000\,\text{s}}$.

TABLE 2. **Description of IDS Dataset Folders**

Folder name	Description
but-iec104-i	This folder contains 7 CSV files. The first file contains regular traffic communication data, while the others contain six attack scenarios: label=. • **DoS Attack:** Overwhelms the host with legitimate IEC 60870-5-104 packets, employing a spoofed IP address and specific ASDU characteristics. • **Injection Attack:** Compromises one host and sends unusual requests, including Single Command (ASDU TypeID=45) and File Access (ASDU TypeID=122) attacks. • **Connection-Loss Attack:** Temporarily interrupts the connection, causing packet loss. • **Switching Attack:** Consists of switching the device on/off using specific IEC 60870-5-104 packets. • **Scanning Attack:** Includes horizontal and vertical scanning, probing for IEC 60870-5-104 objects and responses. • **Rogue-Device Attack:** Involves a rogue device sending legitimate IEC 60870-5-104 packets with measured values.
but-iec104-ii	This folder contains benign communication data.
rts-iec104	The traces in this folder reflect normal SCADA network communication.
vrt-iec104	This folder contains data collected from an IEC virtual testbed, including benign communication and five attacks: label=. • **MITM Attack:** Alters command types and values. • **Value Change Attack:** Manipulates packet contents. • **Masquerading Attack:** Pretends to be a legitimate device. • **Report-block Attack:** Blocks information and sends duplicate packets. • **Replay Attack:** Replays packets to deceive the system.

Table 4 gives the labels and the count of samples in each class after labeling.

C. DATASET PREPROCESSING

Some of the data in the IEC 60870-5-104 dataset included the feature (IOA), and some did not. Hence, the IEC 60870-5-104 data was split into two data frames, IECwithIOA and IECwithoutIOA, and then preprocessed as follows.

1) *IP address encoding:* The IP addresses were encoded and converted into integers by converting the IP address into binary, concatenating all four octets together, then converting them into decimals.
Step 1:192.168.0.1 = 11000000101010000000000000000001
Step 2:11000000101010000000000000000001 = 3232235521.
2) *Categorical values encoding:* The IEC 60870-5-104 data included three categorical features: APDU Format type, u-format type, and IOA. In this step, we represented each unique categorical feature value as an integer value. After conversion, the APDU format type had a value of 0–2, the u-format type ranged between 0 and 3, and IOA was equal to 0–138.
3) *Clear the dataset from null values:* Many of the features included null values. To mitigate this, we replaced all the nulls with the calculated median of each feature.
4) *Data sampling:* As given in Table 4, there is a huge imbalance between classes, and the benign data represent the majority in the dataset. This imbalance can limit the

TABLE 3. **IEC 60870-5-104 Dataset Features**

Feature	Description
Timestamp	The absolute current time
Relative time	The time from the start of capturing in seconds
Source IP address	From the IP header
Destination IP address	From the IP header
Source port	From the TCP header
Destination port	From the TCP header
IP length	from the IP header
APDU length	from the IEC 60870-5-104 header
APDU format type	i-format:0×0, s-format:0×1, u-format:0×3
u-format type	start data transfer:0×01, stop data transfer:0×02, test frame activation:0×10, test frame confirmation: 0×20, stop data transfer action: 0×04, stop data transfer confirmation: 0×08
ASDU type identification	single point information M_SP_NA_1:1, interrogation command: C_IC_NA_1:100, etc.
Number of Information objects	Number of Information objects within an ASDU packet
Cause of Transmission	periodic:1, spontaneous:3, activation:6, confirmation activation:7, etc
Originator address	A unique identifier that specifies the address of the device that originated the communication (optional)
ASDU address field	The address of the target station or the broadcast address
Information Object Address (IOA)	a list of addresses of Information Objects present in the ASDU

TABLE 4. **IEC 60870-5-104 Labels**

Data type	Label	Number of samples
Benign Data	0	2715768
Connection-Loss Attack	1	1564
DoS Attack	2	4362
Switching Attack	3	342
Scanning Attack	4	954
Rogue-device Attack	5	889
Injection Attack	6	671
MITM attack	7	30334
Replay attack	8	30740
Report-block attack	9	11668
Value-change attack	10	19274
Masquerading attack	11	26224

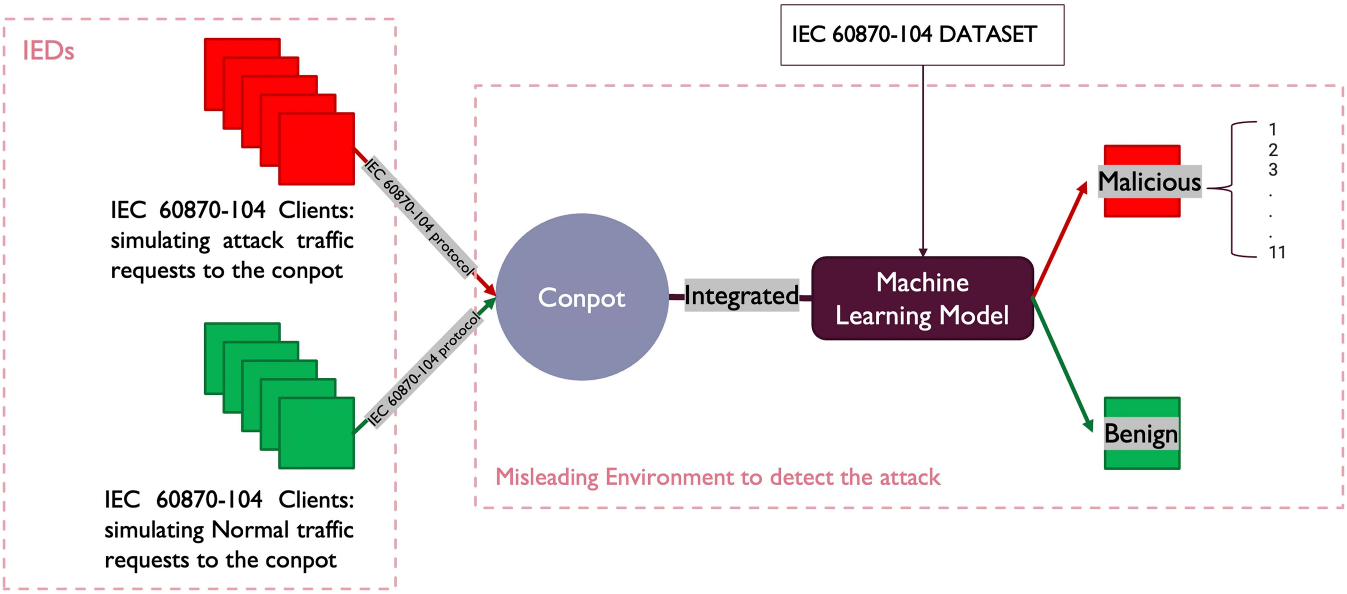
IEDs
IEC 60870-104 Clients: simulating attack traffic requests to the conpot
IEC 60870-104 Clients: simulating Normal traffic requests to the conpot
IEC 60870-104 protocol
IEC 60870-104 protocol
IEC 60870-104 DATASET
Conpot
Integrated
Machine Learning Model
Malicious
1
2
3
.
.
.
11
Benign
Misleading Environment to detect the attack

FIGURE 1. System architecture.

accuracy of the trained model since the majority classes will overpower the minority, causing an imbalance bias in the model. To overcome this, each class was either oversampled/undersampled as needed.

5) *Normalization:* The last step involved applying min–max normalization to preserve the correlation between the original data points while minimizing the impact of outliers.

After preprocessing (steps 1 to 3), we extract 2 CSV files; the first file includes IEC 60870-5-104 with the IOA feature and contains 1 573 737 data samples: 1 564 955 benign, and 8782 malicious samples. The second extracted file has IEC 60870-5-104 without the IOA feature and contains 1 150 813 normal traffic samples and 11 840 attack samples.

IV. METHODOLOGY

In this section, we go through the detailed methodology to build a robust model that can secure the IEC 60870-5-104 communication and detect various attacks with high accuracy, even in the presence of adversarial intrusions. The proposed approach involves training a hierarchical multilayer perceptron (HMLP) and utilizing defensive distillation to enhance the model's resilience.

A. SYSTEM ARCHITECTURE

Our system consists of two main components: the ML classifier and the Conpot. The Conpot will be running with the IEC 60870-5-104 template to mimic the behavior of the IEC 60870-5-104 control and monitoring communication flow. The ML model will be trained on the IEC 60870-5-104 dataset, allowing it to detect up to 11 various attacks. After training the model, it could be integrated on top of the Conpot to enable real-time detection of malicious traffic. This architecture will be able to detect intrusions in real-time. Fig. 1 illustrates the architecture of our proposed IDS, and Algorithm 1 outlines the steps of our methodology.

Algorithm 1 begins with an initialization step, where the model's weights and biases are randomly initialized. The training phase follows, spanning multiple epochs (controlled by the parameter E), during which the training dataset D_{train} is shuffled, and for each batch B_i, a forward pass is performed through the NNs with L hidden layers and H neurons in each layer. This forward pass involves applying activation functions (σ) to compute layer-wise activations. The loss is computed, incorporating both the model's predictions ($M(x)$) and true labels (y), along with a regularization term to prevent overfitting. The backward pass computes gradients using backpropagation, and the weights and biases are updated using a learning rate (Lr) and gradient descent. The algorithm then transitions to the adversarial attack phase, generating adversarial examples for each training example in D_{train}. Three attack methods are outlined: FGSM, PGD, and C&W. These attacks introduce controlled perturbations to input examples to deceive the model. Following the adversarial attack phase, the algorithm proceeds to defensive distillation. It trains a teacher model (M_T) on a distilled dataset D_{dist} using a temperature parameter T_s. For each training example in D_{train}, it softens the logits using the teacher model and trains a student model (M_S) on these softened logits and true labels, incorporating a temperature parameter T_d.

B. BASELINE ML

To get an insight into the performance of the preprocessed dataset and study the importance of the IOA feature, we use SVM, DTs, and multilayer perceptron (MLP) to train binary

Algorithm 1: HMLP Training With Adversarial Attacks and Defensive Distillation.

```
Input      : Training dataset $D_{train}$, learning rate $Lr$, number of epochs $E$, attack strength $\epsilon$, number of iterations $T$, step size for each iteration $\alpha$, temperature parameter $T_d$, distilled dataset $D_{dist}$
Parameters : Number of hidden layers $L$, number of neurons in each hidden layer $H$, batch size $B$, weight decay $\lambda$, temperature parameter $T_s$
Output     : Trained model $M$
Training: Initialize weights $W$ and biases $b$ randomly
for $e = 1$ to $E$ do
    Shuffle training dataset $D_{\text{train}}$
    for each batch $B_i$ in $D_{train}$ do
        Forward pass: $a^{(0)} = B_i$, $a^{(l)} = \sigma(W^{(l)}a^{(l-1)} + b^{(l)})$ for $l = 1, \ldots, L$
        Compute loss: $J = \frac{1}{|B_i|}\sum_{(x,y)\in B_i}\mathcal{L}(M(x),y) + \frac{Lr}{2}\sum_{l=1}^{L}\|W^{(l)}\|_2^2$
        Backward pass: Compute gradients $\nabla_W J$ and $\nabla_b J$ using backpropagation
        Update weights and biases: $W^{(l)} \leftarrow W^{(l)} - \alpha\nabla_{W^{(l)}}J$ and $b^{(l)} \leftarrow b^{(l)} - Lr\nabla_{b^{(l)}}J$ for $l = 1, \ldots, L$
    end
end
Adversarial Attack: for each example $(x, y)$ in $D_{train}$ do
    Generate adversarial example $x_{adv}$
    FGSM Attack:
    Compute gradient: $g = \nabla_x\mathcal{L}(M(x), y)$
    Compute perturbation: $\delta = \epsilon \cdot sign(g)$
    Compute adversarial example: $x_{adv} = \text{Clip}(x + \delta, 0, 1)$ where $\text{Clip}(x, a, b)$ clips $x$ to the range $[a, b]$
    PGD Attack:
    Initialize perturbation: $\delta = 0$
    for $t = 1$ to $T$ do
        Compute gradient: $g = \nabla_x\mathcal{L}(M(x + \delta), y)$
        Compute perturbation: $\delta' = \text{Clip}(\delta + \alpha \cdot sign(g), -\epsilon, \epsilon)$
        Compute adversarial example: $x_{adv} = \text{Clip}(x + \delta', 0, 1)$
        Update perturbation: $\delta \leftarrow \delta'$
    end
    C&W Attack:
    Initialize perturbation: $\delta = 0$
    for $t = 1$ to $T$ do
        Compute perturbation update: $\delta' = \text{Clip}(\delta + \alpha \cdot \text{sign}(\nabla_x J(M(x + \delta), y) - \nabla_x J(M(x), y)), -\epsilon, \epsilon)$
        Compute adversarial example: $x_{adv} = \text{Clip}(x + \delta', 0, 1)$
        Update perturbation: $\delta \leftarrow \delta'$
    end
    Add adversarial example to training dataset: $D_{train} \leftarrow D_{train} \cup \{(x_{adv}, y)\}$
end
Defensive Distillation:
Train teacher model $M_T$ on distilled dataset $D_{dist}$ using temperature parameter $T_s$
for each example $(x, y)$ in $D_{train}$ do
    Soften logits using teacher model: $z = \frac{1}{T_s} \cdot M_T(x)$
    Train student model $M_S$ on softened logits and true labels $(z, y)$ using temperature parameter $T_d$
end
```

and multiclass classifiers. We use the accuracy and F1-score as the performance metrics.

1) BINARY CLASSIFICATION

We train a binary classifier using DT, SVM, and MLP. For each algorithm, we train two classifiers, one for the data with the IOA feature and one for the data without the IOA feature. For the MLP, we use a trial-and-error approach to tune the hyperparameters. The architecture and final parameters of the binary MLP are detailed in Table 5.

TABLE 5. HMLP Models Parameters

	Binary model	Attack model
Parameters	Learning Rate: 0.006 Epochs: 20 Batch size: 1000	Learning Rate: 0.006 Epochs: 300 Batchsize: 1000
Inputs	14	14
Outputs	2	11
Layers	Linear (14×512) Batchnorm (512) ReLU Dropout (0.2) Linear (512×128) Batchnorm (128) ReLU Dropout (0.2) Linear (128×64) Batchnorm (64) ReLU Dropout (0.2) Linear (64×2)	Linear (14×512) Batchnorm (512) ReLU Dropout (0.2) Linear (512×256) Batchnorm (256) ReLU Dropout (0.2) Linear (256×128) Batchnorm (128) ReLU Dropout (0.2) Linear (128×32) Batchnorm (32) ReLU Dropout (0.2) Linear (32×11)

2) MULTICLASS CLASSIFICATION

Similarly, we train two classifiers per algorithm for multiclassification, one for IOA and one for non-IOA data. We sample the data classes before the training to overcome the bias caused by data imbalance. The architecture and final parameter of the multiclass MLP are similar to the attack model in Table 5.

3) IOA FEATURE SIGNIFICANCE

We utilize a RF classifier to determine the importance of the IOA feature. Following this, we exported and organized the significance of each feature. The significance of each feature in the classifier is illustrated in Fig. 2.

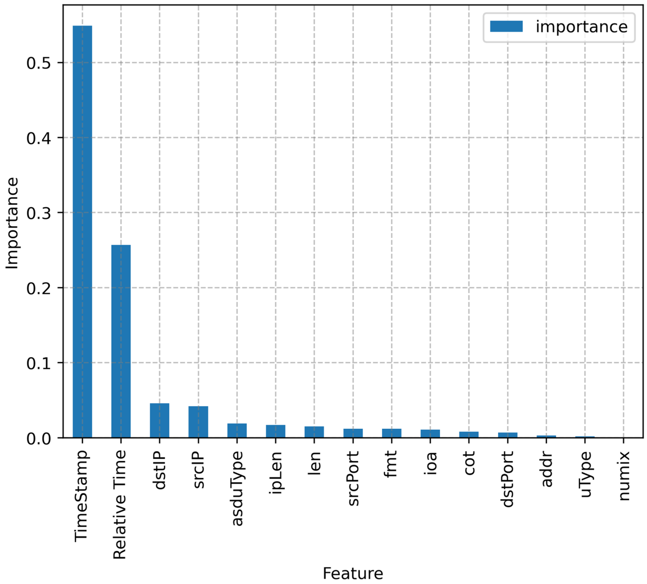

FIGURE 2. **Features significance.**

C. HIERARCHICAL MULTILAYER PERCEPTRON

The data imbalance between benign and malicious samples is evident in Table 4. This imbalance is causing multiclass classifiers to perform poorly. This motivates us to use a hierarchical approach to train the model. The first layer decides if the data are benign or malicious, and the second layer determines the attack type. This approach isolates the attack samples from the benign samples when training the multiclass classifier in the second layer, which helps to reduce the impact of data imbalance drastically, allowing the model to perform better and achieve a higher detection rate, especially for multiclass classification.

After observing the performance of both binary and multiclass classifiers and determining the significance of the IOA feature, we adopted a hierarchical classification approach to increase the granularity of the model. As shown in Fig. 3, the trained classifier includes two models: the first model decides whether a given sample is benign or malicious, and if the sample is malicious, it is forwarded to the second model, which determines the exact type of attack.

1) PREPROCESSING FOR THE HMLP

Before training the HMLP, the IOA feature was dropped, and all the IEC 60870-5-104 data with the 11 attacks were merged. We unify the label of all attack samples to train the binary model. In contrast, we keep the labels as it is to train the attack model, but we remove all the benign samples from the dataset.

V. ADVERSARIAL TRAINING FOR ROBUST ML-BASED IDS

With the wide use of ML in the security of SGs, attackers started employing their attacks on ML models to mislead IDS and evade detection. Such attacks are called adversarial attacks, which generate malicious inputs using small perturbations to fool ML models and force them to misclassify a given input. After training our HMLP, which consists of the binary and the attack models, we evaluate the robustness of both models against adversarial attacks to study the effect of such attacks on the model's accuracy and strengthen the robustness of the model.

A. ADVERSARIAL ATTACKS

In this work, we evaluated our model robustness against FGSM, PGD, and C&W attacks.

1) FGSM

A simple attack method that adds a small perturbation to each feature of the input in the direction of the gradient of the loss function with respect to the input.It is called "fast" because it only requires one forward and one backward pass through the model to generate the adversarial example

$$\boldsymbol{x}' = \boldsymbol{x} + \epsilon \cdot \operatorname{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y)) \tag{1}$$

where $\boldsymbol{x}$ is the original input, $\boldsymbol{x}'$ is the perturbed input, ϵ is the magnitude of the perturbation, $J(\boldsymbol{\theta}, \boldsymbol{x}, y)$ is the loss function of the NNs with parameters $\boldsymbol{\theta}$ evaluated on input $\boldsymbol{x}$ and true label y, and $\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y)$ is the gradient of the loss with respect to the input.

2) PGD

PGD is a more powerful attack method that performs multiple steps of gradient descent with a small step size to iteratively generate an adversarial example that maximizes the model's loss. PGD can be seen as an extension of FGSM and is considered one of the most effective white-box attacks

$$\boldsymbol{x}^{(t+1)} = \operatorname{clip}_{\boldsymbol{x}+\epsilon} \boldsymbol{x}^{(t)} + \alpha \cdot \operatorname{sign}(\nabla \boldsymbol{x}^{(t)} J(\boldsymbol{\theta}, \boldsymbol{x}^{(t)}, y)) \tag{2}$$

where $\boldsymbol{x}$ is the original input, ϵ is the maximum perturbation allowed, t is the iteration index, α is the step size of the update, clip is a function that clips the values of its argument to the given range, and $\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}^{(t)}, y)$ is the gradient of the loss with respect to the input evaluated at the current perturbed input $\boldsymbol{x}^{(t)}$.

3) CARLINI AND WAGNER

C&W attack uses an optimization-based approach to generate adversarial examples. It minimizes a custom loss function that considers the distance between the original input and the adversarial example, as well as the confidence of the model's prediction on the adversarial example. This makes the C&W attack one of the most effective black-box attack methods, as it does not require knowledge of the model's parameters or gradients

$$\min_{\boldsymbol{\delta}, \boldsymbol{\omega}} |\boldsymbol{\delta}|_p + c \cdot f(\boldsymbol{x} + \boldsymbol{\delta}) - f(\boldsymbol{x}) + \alpha \cdot |\boldsymbol{\omega}|_q \tag{3}$$

where $\boldsymbol{\delta}$ is the perturbation vector added to the input $\boldsymbol{x}$, $\boldsymbol{\omega}$ is an auxiliary variable used to enforce a constraint on the perturbation magnitude, f is the NN, p and q are the L_p

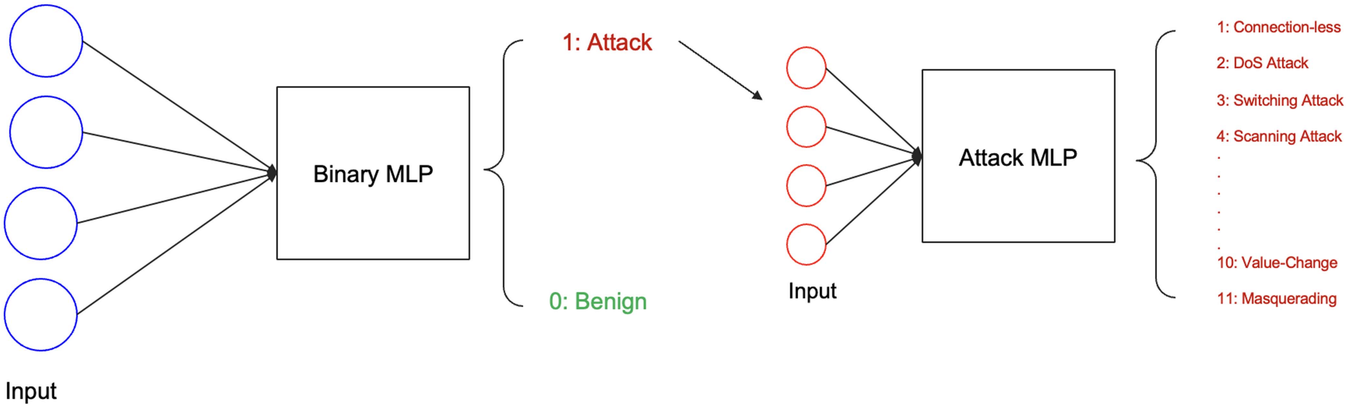

FIGURE 3. **Hierarchical MLP architecture.**

and L_q norms used to measure the perturbation and auxiliary variables, c is a hyperparameter that controls the tradeoff between the perturbation magnitude and the classification loss, and α is a hyperparameter that controls the tradeoff between the perturbation magnitude and the constraint on the auxiliary variable.

B. DEFENSE MECHANISMS

To increase the robustness of the model against the aforementioned attacks, we use two well-known defense mechanisms: adversarial training and defensive distillation.

1) ADVERSARIAL TRAINING

We train the model on a mixture of original and adversarial examples. The adversarial examples are generated by perturbing the original input data using FGSM, PGD, or C&W attacks. We perform the adversarial training for each attack separately. The model is expected to be more resilient to adversarial attacks by training on both original and adversarial examples. Fig. 4 illustrates the adversarial training steps.

2) DEFENSIVE DISTILLATION

We trained the final model using a two-stage process. In the first stage, a model is trained on the original data using a standard supervised learning algorithm. In the second stage, a new model is trained using the output of the first model as input but with a different temperature parameter during the Softmax activation function. The temperature parameter controls the smoothness of the probabilities assigned to each class, with higher temperatures resulting in softer probabilities. We train a single model for all three attacks, and each iteration involves three optimization steps. The first step calculated the loss based on the original data, while the second and third steps computed it based on FGSM adversarial examples and PGD, respectively. The approach that was followed is illustrated in Fig. 5.

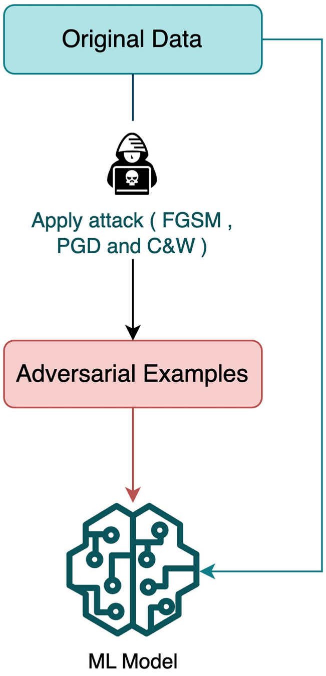

FIGURE 4. **Adversarial training.**

VI. PERFORMANCE EVALUATION

In this section, we provide details about the implementation of our system model and later present the results we obtained.

A. EXPERIMENTAL SETUP

To construct our robust intrusion detector, we employ the architecture and parameters outlined in Table 5. We observe that the time complexity of the binary model is primarily governed by operations within its linear layers, estimated to involve approximately $14 \times 512 + 512 \times 128 + 128 \times 64 + 64 \times 2$ computations. Meanwhile, the complexity of the attack model, also driven by its linear layers, is summarized by the equation $14 \times 512 + 512 \times 256 + 256 \times 128 + 128 \times 32 + 32 \times 11$ computations. Among different NN architectures, our proposed models, namely the binary model,

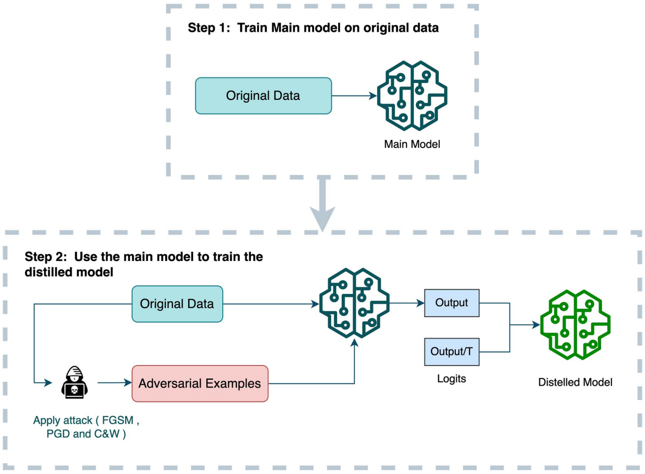
Step 1: Train Main model on original data
Original Data
Main Model
Step 2: Use the main model to train the distilled model
Original Data
Output
Output/T
Logits
Distelled Model
Adversarial Examples
Apply attack (FGSM , PGD and C&W)

FIGURE 5. Defensive distillation.

and the attack model, are noted for their comparative simplicity and enhanced efficiency. Compared to more complex structures such as convolutional NNs, long short-term memory networks, and gated recurrent units, both models are characterized by their minimal computational requirements and lower resource consumption. This aspect renders them particularly suitable for applications in environments constrained by limited computational resources, such as in SG devices. We use Pytorch to train and test the model and our detector was built by training the binary model on a dataset of 200 000 samples, equally divided between benign and malicious data. For the attack model, we train it on 65 000 samples, with a nearly equal distribution across all 11 classes. The split ratio between the training and testing data is 70:30. Specifically, the testing dataset was constructed as a subset of the original dataset but kept distinct from the training dataset to ensure model generalization. It comprises 60 000 samples for the binary model and 27 500 samples for the attack model. Both datasets contain a balanced distribution of benign and malicious data classes but differ in their temporal aspects, ensuring that the model is tested on different scenarios than those it was trained on. The data in the testing set was also subjected to separate preprocessing steps, including normalization and the introduction of simulated noise to more closely simulate real-world variability. For software, we utilized PyTorch for model training and evaluation, running in a Python 3.8 environment. Additional libraries, such as Scikit-learn and NumPy, were used for data preprocessing and manipulation. On the hardware front, all experiments were conducted on a computing cluster featuring Nvidia GeForce RTX 3080 GPUs and Intel Xeon CPUs. Finally, we incorporate the Torchattacks library from [31] to execute FGSM, PGD, and C&W attacks using the specified parameters in Table 6.

B. EVALUATION METRICS

We conduct extensive simulation experiments to determine the effectiveness of our proposed approach. We assess the performance of the detection scheme using standard classification metrics, specifically accuracy and F1-score. In addition, we analyze the per-class performance by examining the confusion matrix

TABLE 6. Adversarial Attacks Parameters

Attack	Parameter
FGSM	$\epsilon = 0.05$
PGD	$\epsilon = 0.05, \alpha = 0.01, \text{steps} = 10$
C&W	$\text{c} = 1, \text{kappa} = 0, \text{steps} = 100, \text{lr} = 0.01$

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP + TN + FP + FN}} \tag{4}$$

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

$$\text{Precision} = \frac{\text{TP}}{\text{TP + FP}} \tag{6}$$

where TP, TN, FP, and FN are the number of true positives, true negatives, false positives, and false negatives, respectively. The F_1 is defined as follows:

$$\text{F}_1^{\text{macro}} = \frac{1}{n}\sum_{i=1}^{n} \frac{2 \cdot \text{precision}_i \cdot \text{recall}_i}{\text{precision}_i + \text{recall}_i} \tag{7}$$

where n is the number of classes, precision_i is the precision of class i, and recall_i is the recall of class i.

C. BASELINE ML RESULTS

To understand the performance and create a baseline benchmark, we compare several ML models, includingDTs, SVM, and NNs. During this phase, separate models are trained for IOA and non-IOA data, and we experiment with both binary and multiclass classification approaches. The performance of these models is evaluated and summarized in Table 7.

Following our analysis of the results presented in Table 7, we concluded that implementing a hierarchical approach could potentially enhance the overall performance of our models. Therefore, we proceeded to train two separate models (the attack model and the binary model) utilizing the same model architectures and training parameters as detailed in Table 7. The performance of these models was as follows.

1) BINARY MODEL

The binary model achieved 100% accuracy and 100% macro average F1 Score. The confusion matrix in Fig. 6 shows that the model can accurately classify all the test samples. This performance suggests that the binary model has learned the feature space for benign and malicious data quite effectively, making it highly reliable for the classification tasks at hand.

2) ATTACK MODEL

The attack model is evaluated on a test dataset containing 19 500 instances. The confusion matrix in Fig. 8 shows the

TABLE 7. ML Models Results

Algorithm	Metric	Multiclass classification-IOA	Multiclass classification-NoIOA	Binary classification-IOA	Binary classification-NoIOA
SVM	Macro Average F1-Score	14%	10%	48%	48%
	Accuracy	81.8%	6%	82%	90.7%
DT	Macro Average F1-Score	34%	98%	54%	99%
	Accuracy	95.5%	99.5%	97%	99.7%
MLP	Macro Average F1-Score	94%	99%	100%	100%
	Accuracy	94.7%	98.7%	100%	100%

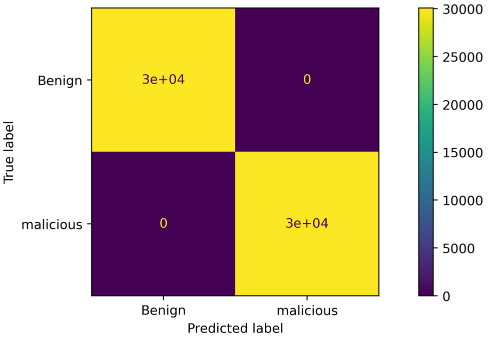

FIGURE 6. Binary model confusion matrix.

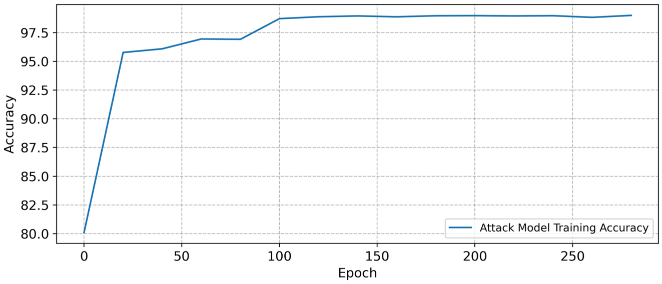

FIGURE 7. Attack model training accuracy.

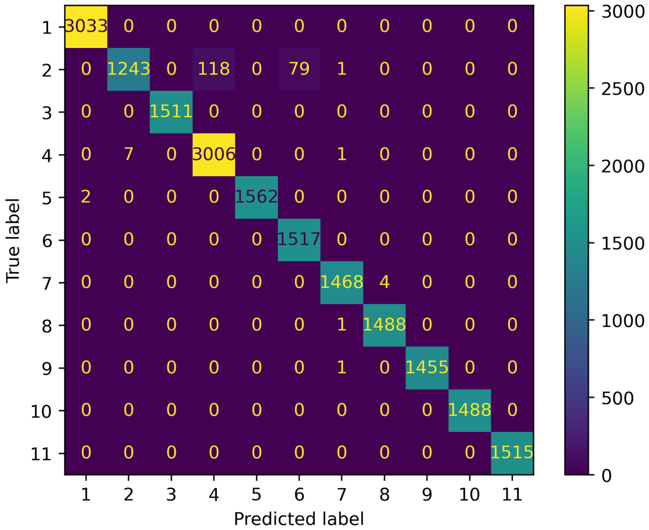

FIGURE 8. Attack model confusion matrix.

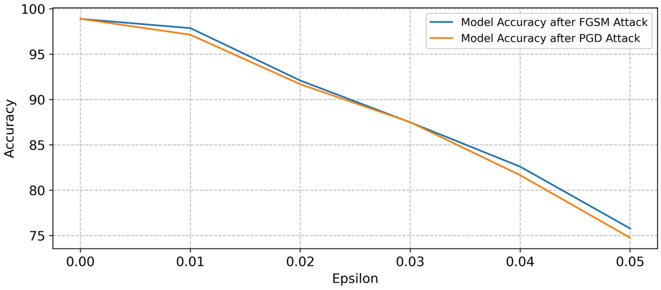

FIGURE 9. FGSM and PGD attack effect using different epsilon values.

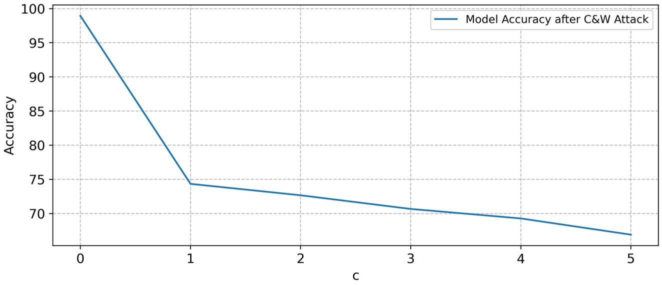

FIGURE 10. C&W attack effect using different C values.

correct classification count for each class along the diagonal line. A total of 220 test samples were misclassified. The model's accuracy is 98.87%, with a macro average F1 score of 99%. The accuracy trend of the attack model throughout the training process is shown in Fig. 7. It is worth noting that the misclassified instances mainly belong to specific classes, thereby providing a route for further fine-tuning.

D. ADVERSARIAL ATTACK EFFECT

To test the resilience of our models, we conducted a series of adversarial attacks. This section elaborates on the susceptibility of our models to these attacks and the effectiveness of the implemented defenses. Figs. 9 and 10 illustrate the impact of applying adversarial attacks (FGSM, PGD, and C&W) on the attack Model while varying the attack parameters. Similarly, Table 8 presents the effect of these attacks on both models using the parameters described in Section VI-A.

Table 8 tabulates that the binary model is inherently resistant to all three attacks. This implies that even when introducing a perturbation to the input, the binary model can

TABLE 8. Adversarial Attacks Effect on Model Accuracy

	Original	FGSM	PGD	C&W
Binary Model	100%	100%	100%	100%
Attack Model	98.9%	75.7%	74.9%	73.7%

TABLE 9. Adversarial Training Effect on Model Accuracy

	Original	FGSM	PGD	C&W
Accuracy Before	98.9%	75.7%	74.9%	73.7%
Accuracy After	96%	95.4%	95%	73.7%

TABLE 10. Defensive Distillation Effect on Model Accuracy

	Original	FGSM	PGD	C&W
Accuracy Before	98.9%	75.7%	74.9%	73.7%
Accuracy After	99.4%	97.5%	92%	98.9%

still accurately classify the given input's class. However, as shown in the table, the accuracy of the attack model declines by almost 30% when any of the three attacks are employed, leading to misclassification of the attack type. Consequently, it is imperative to implement a defense mechanism to enhance the attack model's robustness against adversarial attacks.

E. ADVERSARIAL DEFENSE RESULTS

1) ADVERSARIAL TRAINING

To enhance the robustness of our model, we adopt adversarial training. This involves optimizing the model parameters at each iteration twice: once using the original data and then using the generated adversarial examples. The model can learn to handle perturbations and improve its overall robustness by incorporating adversarial examples into the training process. Table 10 illustrates how adversarial training affects the model's accuracy against different attacks during the adversarial training process. It is clear from the Table that even though the adversarial training helped increase the robustness of the model against FGSM and PGD, it did not provide any improvement for the C&W attack. This iterative training process has shown to be particularly effective for countering FGSM and PGD attacks, as confirmed by the significantly improved post-training accuracy.

2) DEFENSIVE DISTILLATION

While adversarial training enhanced our model's robustness, we sought an additional layer of security through defensive distillation to protect against a broader range of attacks. Adversarial training reduced the impact of FGSM and PGD attacks but impacted the model's original accuracy and did not enhance its resistance against C&W attacks. Furthermore, since adversarial training was only effective against each attack individually, we trained a separate model for each attack. To address this limitation, we use defensive distillation, which involves training a student model using the original model's output to improve its generalization and make it more robust against adversarial attacks. Fig. 11 shows how defensive distillation improved the model's accuracy in classifying all three types of attacks during the training process. The results demonstrate that the distillation process enhanced the model's ability to accurately identify the adversarial examples generated by the attacks.

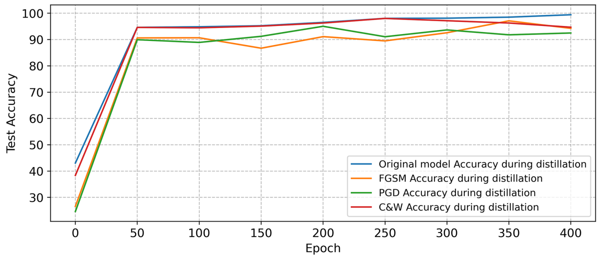

FIGURE 11. Model accuracy during defensive distillation training.

Implementing defensive distillation significantly enhanced the model's accuracy against various attacks. Specifically, the accuracy improved from 77.5% to 96% against FGSM, from 73.8% to 92% for PGD, and from 70.3% to 99% for C&W attacks while boosting the model's original accuracy to 99.4%.

VII. CONCLUSION

In this article, we presented a novel approach for building a model for the IEC 60870-5-104 protocol that is robust to adversarial attacks and can be used to enhance the security of SG systems. Our contributions include creating and utilizing new datasets for the IEC 60870-5-104 protocol, a significant asset for the open-source community. The proposed model uses a hierarchical approach to classify the input, where the first layer gives a binary decision about whether the sample is benign or malicious, and the second layer determines the type of attack. Our models achieved exceptionally high accuracy rates, with the binary model reaching 100% accuracy and the attack model scoring 98.87%, confirming the effectiveness of the hierarchical approach. We used defensive distillation to increase the model's resilience against various kinds of attacks, including FGSM, PGD, and C&W attacks. Specifically, implementing defensive distillation enhanced the model's accuracy against FGSM from 77.5% to 96%, against PGD from 73.8% to 92%, and against C&W from 70.3% to 99%. This substantial improvement in robustness underscores the efficacy of our adversarial defense strategies. We also quantified the efficiency of our proposed ML-based IDS, demonstrating its suitability for real-time applications of ICS. We also demonstrated how our developed model can be integrated with Conpot, a popular open-source tool for simulating ICSs, to detect and prevent attacks on the SG. Our experimental results showed that the proposed approach achieved high accuracy in classifying normal and adversarial examples while maintaining highly efficient intrusion detection for eleven types of attacks. These results demonstrate the potential of the proposed approach to improve the security and reliability of SG systems against cyber threats.

ACKNOWLEDGMENT

The findings herein reflect the work, and are solely the responsibility, of the authors. Open Access funding provided by the Qatar National Library.

REFERENCES

[1] P. Kumar, Y. Lin, G. Bai, A. Paverd, J. S. Dong, and A. Martin, "Smart grid metering networks: A survey on security, privacy and open research issues," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2886–2927, Third Quarter 2019.

[2] P. H. Mirzaee, M. Shojafar, H. Cruickshank, and R. Tafazolli, "Smart grid security and privacy: From conventional to machine learning issues (threats and countermeasures)," *IEEE Access*, vol. 10, pp. 52922–52954, 2022.

[3] A. Elmasry, A. Albaseer, and M. M. Abdallah, "OpenPLC and lib61850 smart grid testbed: Performance evaluation and analysis of GOOSE communication," in *Proc. Int. Symp. Netw. Comput. Commun. Trust Secur. Privacy*, 2023, Art. no. 6.

[4] S. P. Sathar, S. Al-Kuwari, A. Albaseer, M. Qaraqe, and M. M. Abdallah, "Mitigating IEC-60870-5-104 vulnerabilities: Anomaly detection in smart grid based on LSTM autoencoder," in *Proc. Int. Symp. Netw. Comput. Commun. Trust Secur. Privacy*, 2023, Art. no. 6.

[5] H. Zhang, T. Shen, F. Wu, M. Yin, H. Yang, and C. Wu, "Federated graph learning–A position paper," 2021, *arXiv:2105.11099*.

[6] G. Liang, S. R. Weller, J. Zhao, F. Luo, and Z. Y. Dong, "The 2015 Ukraine blackout: Implications for false data injection attacks," *IEEE Trans. Power Syst.*, vol. 32, no. 4, pp. 3317–3318, Jul. 2017.

[7] A. Albarakati et al., "Security monitoring of IEC 61850 substations using IEC 62351-7 network and system management," *IEEE Trans. Ind. Informat.*, vol. 18, no. 3, pp. 1641–1653, Mar. 2022.

[8] L. Erdődi, P. Kaliyar, S. H. Houmb, A. Akbarzadeh, and A. J. Waltoft-Olsen, "Attacking power grid substations: An experiment demonstrating how to attack the SCADA protocol IEC 60870-5-104," in *Proc. 17th Int. Conf. Availability Rel. Secur.*, 2022, pp. 1–10.

[9] O. Boyaci, M. R. Narimani, K. R. Davis, M. Ismail, T. J. Overbye, and E. Serpedin, "Joint detection and localization of stealth false data injection attacks in smart grids using graph neural networks," *IEEE Trans. Smart Grid*, vol. 13, no. 1, pp. 807–819, Jan. 2022.

[10] M. E. Eddin et al., "Fine-tuned RNN-based detector for electricity theft attacks in smart grid generation domain," *IEEE Open J. Ind. Electron. Soc.*, vol. 3, no. 1, pp. 733–750, Dec. 2022.

[11] A. Albaseer and M. Abdallah, "Fine-tuned LSTM-based model for efficient honeypot-based network intrusion detection system in smart grid networks," in *Proc. 5th Int. Conf. Commun. Signal Process. Appl.*, 2022, pp. 1–6.

[12] T. N. Nguyen, B.-H. Liu, N. P. Nguyen, and J.-T. Chou, "Cyber security of smart grid: Attacks and defenses," in *Proc. IEEE Int. Conf. Commun.*, 2020, pp. 1–6.

[13] M. Ezeddin, A. Albaseer, M. Abdallah, S. Bayhan, M. Qaraqe, and S. Al-Kuwari, "Efficient deep learning based detector for electricity theft generation system attacks in smart grid," in *Proc. 3rd Int. Conf. Smart Grid Renewable Energy*, 2022, pp. 1–6.

[14] K. Nagaraj, A. Starke, and J. McNair, "GLASS: A graph learning approach for software defined network based smart grid DDoS security," in *Proc. IEEE Int. Conf. Commun.*, 2021, pp. 1–6.

[15] A. Albaseer and M. Abdallah, "Privacy-preserving honeypot-based detector in smart grid networks: A new design for quality-assurance and fair incentives federated learning framework," in *Proc. IEEE 20th Consum. Commun. Netw. Conf.*, 2023, pp. 722–727.

[16] M. Zhang, X. Fan, R. Lu, C. Shen, and X. Guan, "Extended moving target defense for AC state estimation in smart grids," *IEEE Trans. Smart Grid*, vol. 14, no. 3, pp. 2313–2325, May 2023.

[17] V. Babu, R. Kumar, H. H. Nguyen, D. M. Nicol, K. Palani, and E. Reed, "Melody: Synthesized datasets for evaluating intrusion detection systems for the smart grid," in *Proc. Winter Simul. Conf.*, 2017, pp. 1061–1072.

[18] I. Siniosoglou et al., "NeuralPot: An industrial honeypot implementation based on deep neural networks," in *Proc. IEEE Symp. Comput. Commun.*, 2020, pp. 1–7.

[19] C. M. Ahmed and N. K. Kandasamy, "A comprehensive dataset from a smart grid testbed for machine learning based CPS security research," in *Proc. Int. Workshop Cyber- Phys. Secur. Crit. Infrastructures Protection*, Springer, 2021, pp. 123–135.

[20] W. Zhe, C. Wei, and L. Chunlin, "Dos attack detection model of smart grid based on machine learning method," in *Proc. IEEE Int. Conf. Power Intell. Comput. Syst.*, 2020, pp. 735–738.

[21] S. Khan, K. Kifayat, A. K. Bashir, A. Gurtov, and M. Hassan, "Intelligent intrusion detection system in smart grid using computational intelligence and machine learning," *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 6, 2021, Art. no. e4062.

[22] T. S. Ustun, S. 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, 2021, Art. no. 826.

[23] S. Aziz, M. Irshad, S. A. Haider, J. Wu, D. N. Deng, and S. Ahmad, "Protection of a smart grid with the detection of cyber-malware attacks using efficient and novel machine learning models," *Front. Energy Res.*, vol. 10, 2022, Art. no. 1102.

[24] E. Nowroozi, Y. Mekdad, M. H. Berenjestanaki, M. Conti, and A. E. Fergougui, "Demystifying the transferability of adversarial attacks in computer networks," *IEEE Trans. Netw. Service Manag.*, vol. 19, no. 3, pp. 3387–3400, Sep. 2022.

[25] T. Berghout, M. Benbouzid, and S. Muyeen, "Machine learning for cybersecurity in smart grids: A comprehensive review-based study on methods, solutions, and prospects," *Int. J. Crit. Infrastructure Protection*, vol. 38, 2022, Art. no. 100547.

[26] A. N. Alkuwari, S. Al-Kuwari, and M. Qaraqe, "Anomaly detection in smart grids: A survey from cybersecurity perspective," in *Proc. 3rd Int. Conf. Smart Grid Renewable Energy*, 2022, pp. 1–7.

[27] W. Wang, F. Harrou, B. Bouyeddou, S.-M. Senouci, and Y. Sun, "Cyber-attacks detection in industrial systems using artificial intelligence-driven methods," *Int. J. Crit. Infrastructure Protection*, vol. 38, 2022, Art. no. 100542.

[28] M. Panthi and T. K. Das, "Intelligent intrusion detection scheme for smart power-grid using optimized ensemble learning on selected features," *Int. J. Crit. Infrastructure Protection*, vol. 39, 2022, Art. no. 100567.

[29] T. Yu et al., "An advanced accurate intrusion detection system for smart grid cybersecurity based on evolving machine learning," *Front. Energy Res.*, vol. 10, 2022, Art. no. 903370.

[30] Sep. 2022. [Online]. Available: https://www.fit.vut.cz/research/project/1303/.en

[31] H. Kim, "Torchattacks: A PyTorch repository for adversarial attacks," Sep. 2020. [Online]. Available: http://arxiv.org/abs/2010.01950

HADIR TERYAK received the B.Sc. degree in computer engineering from Qatar University, Doha, Qatar, in 2021 and the M.Sc. degree in data science and engineering from Hamad Bin Khalifa University, Ar-Rayyan, Qatar, in 2023.

Her research focuses on the use of machine learning in smart grid security.

ABDULLATIF ALBASEER (Member, IEEE) received the M.Sc. degree in computer networks from the King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia, in 2017 and the Ph.D. degree in computer science and engineering from Hamad Bin Khalifa University, Doha, Qatar, in 2022.

He is currently a Postdoctoral Research Fellow with the Smart Cities and IoT Lab, Hamad Bin Khalifa University. He has authored and co-authored more than twenty conference and journal papers in IEEE ICC, IEEE Globecom, IEEE CCNC, and IEEE Transactions. He also has six US patents in the area of the wireless network edge. His current research interests include AI for networking, AI for cybersecurity, distributed AI, and edge intelligence.

MOHAMED ABDALLAH (Senior Member, IEEE) received the B.Sc. degree in electrical and computer engineering from Cairo University, Giza, Egypt, in 1996, and the M.Sc. and Ph.D. degrees in electrical and computer engineering from the University of Maryland at College Park, College Park, MD, USA, in 2001 and 2006, respectively.

From 2006 to 2016, he held academic and research positions with Cairo University and Texas A&M University in Qatar, Doha, Qatar. He is currently a Founding Faculty Member with the rank of Associate Professor with the College of Science and Engineering, Hamad Bin Khalifa University, Doha. He has authored or co-authored more than 150 journals and conferences and four book chapters and co-invented four patents. His current research interests include wireless networks, wireless security, smart grids, optical wireless communication, and blockchain applications for emerging networks.

Dr. Abdallah is a recipient of the Research Fellow Excellence Award at Texas A& M University in 2016, the Best Paper Award in multiple IEEE conferences, including IEEE BlackSeaCom 2019 and the IEEE First Workshop on Smart Grid and Renewable Energy in 2015, and the Nortel Networks Industrial Fellowship for five consecutive years, 1999–2003. His professional activities include an Associate Editor of IEEE TRANSACTIONS ON COMMUNICATIONS and IEEE OPEN ACCESS JOURNAL OF COMMUNICATIONS, *Track Co-Chair of IEEE VTC Fall 2019 Conference*, *Technical Program Chair of the 10th International Conference on Cognitive Radio-Oriented Wireless Networks*, and a *Technical Program Committee Member of several major IEEE conferences*.

SAIF AL-KUWARI (Senior Member, IEEE) received the Bachelor of Engineering degree in computers and networks from the University of Essex, Colchester, CO4 3SQ, U.K., in 2006 and two PhD's from the University of Bath, Bath, BA2 7AY, U.K., and Royal Holloway, University of London, London, TW20 0EX, U.K., in computer science, both in 2012.

He is currently an Assistant Professor with the College of Science and Engineering, Hamad Bin Khalifa University, Doha, Qatar. His research interests include applied cryptography, Quantum Computing, Computational Forensics, and their connections with Machine Learning.

Dr. Kuwari is IET and BCS Fellow, and ACM Senior Member.

MARWA QARAQE (Senior Member, IEEE) received the bachelor's degree in electrical engineering from Texas A&M University at Qatar, Doha, Qatar, in 2010, and the M.S. and Ph.D. degrees in electrical engineering from Texas A&M University, College Station, TX, USA, in 2012 and 2016, respectively.

She is currently an Associate Professor at Hamad Bin Khalifa University in Qatar. Her research interests include wireless communication, signal processing, and machine learning, and their application in multidisciplinary fields, including but not limited to security, IoT, and health, and in physical layer security, federated learning over wireless networks, and machine learning for wireless communication, security, and health.