This article has been accepted for publication in IEEE Transactions on Network Science and Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2025.3567340

Trojan Attack and Defense for Deep Learning based Power Quality Disturbances Classification

Sultan Uddin Khan, Mahmoud Nabil, Mohamed M. E. A. Mahmoud, Maazen AlSabaan, Tariq Alshawi,

***Abstract*—Accurate classification of power quality disturbances (PQD) is essential for ensuring the reliability and safety of modern power systems. However, deep learning (DL) models used for PQD classification can be compromised by trojan attacks—malicious modifications that alter model behavior only in the presence of specific triggers. Motivated by the urgent need to safeguard modern power grids, we, for the first time, propose trojan attack for DL-based PQD classification by introducing a novel trojan attack algorithm called Sneaky Spectral Strike (S^3). Key features of S^3 include balancing the signal-to-noise (SNR) ratio to maintain imperceptibility and optimizing the fooling rate (FR) for maximum effectiveness. Leveraging the Fast Fourier Transform (FFT) and trigger optimization techniques to embed a stealthy trigger, S^3 achieves an impressive fooling rate of 99.9% with minimal impact on clean-data accuracy across diverse DL architectures. Unlike prior time series data (TSD) based trojan attack studies that use datasets with limited sample diversity and time-step variations, we utilize a comprehensive PQD dataset encompassing a wide range of events and varied time steps, thereby exposing vulnerabilities in more diverse and realistic scenarios. S^3 outperforms state-of-the-art methods, improving the average fooling rate by 7.4% over TimeTrojanDE, 0.83% over TSBA, and 0.25% over TrojanFlow. To assess generalizability, S^3 was also evaluated on two additional time-series datasets, achieving fooling rate of 99.89% on the Online Retail Dataset and a perfect 100% on the Household Electric Power Consumption Dataset. To counter such sophisticated attacks, we also propose an innovative defense mechanism that detects trojan attacks by analyzing decision boundary discrepancies resulting from trojan insertion and injecting universal adversarial perturbations. Our defense strategy demonstrates effectiveness in identifying compromised models across various class scenarios, with the capability to detect infected models across 14 of the 17 class scenarios with trojan infection probability peaks at 0.94076.**

***Index Terms*—Trojan attack, targeted attack, security, smart grid, power quality disturbance, deep learning**

I. Introduction

The modern power system is an interconnected system that encompasses energy generation, transmission, and distribution, facilitating two-way communication between the utility and its customers. This allows consumers to more efficiently manage their energy use and even sell surplus power back to the grid [1]. Secure communication technologies underpin this bidirectional flow of electricity, with the goal of improving customer satisfaction, optimizing utility operations, and promoting environmental sustainability by using more renewable energy sources [2]. It also possesses the potential to autonomously restore the power flow to the load through its inherent self-healing capabilities in the event of a distribution feeder or transformer failure [3]. The system employs advanced technologies, such as sensors, automation, and communication networks, to monitor and control power distribution in real-time [4]. Smart meters, which are part of the Advanced Metering Infrastructure (AMI), are installed at the consumer end to measure power consumption and deliver real-time data to the utility [5]. Consequently, it can deliver energy more efficiently, enabling improved consumer utility engagement, facilitating widespread voltage control, ensuring dependable frequency control, implementing modern management techniques, and effectively responding to a wide range of events inside the system [6]. Renewable energy sources and energy storage systems offer significant potential for decarbonizing metropolitan areas, regulating frequency and voltage deviations, and addressing periods of high demand surpassing generation capacity [7], [8].

Despite the integration of advanced communication technologies, there remains a potential for power quality to be degraded. Various factors contribute to PQD, which exhibit similarities to those observed in traditional power systems. PQD are any deviations in voltage, current, or frequency that can impact the functioning of electrical equipment [9]. The increasing reliance on renewable energy sources and other important variables can cause PQD into the grid [10]. Moreover, the growing use of non-linear equipment, such as personal computers, light-emitting diode lamps, and numerous electronic gadgets, can produce harmonics and pose new concerns with PQD [11]. In addition, small cybersecurity threats can potentially disrupt communication lines and control mechanisms, hence increasing PQD [12]. The occurrence of PQD can result in a wide range of implications, varying from small inconveniences to substantial economic and safety issues. PQD has the potential to adversely affect power system-related equipment, leading to various consequences such as production interruptions, escalated maintenance expenses, diminished device longevity, heightened safety hazards, and increased energy inefficiency. To address this serious issue, anomalies in the grid, such as PQD or equipment failures, can be quickly identified by employing DL models [13].

While DL models possess significant potential in PQD classification, they are vulnerable to trojan attacks [14]. A trojan attack maliciously alters a neural network so that it behaves normally with clean inputs but reacts in a specific, predetermined way when exposed to certain trigger inputs. The complexities associated with trojan signals, which exhibit a notable similarity to clean signals, pose a significant threat, resulting in jeopardizing the integrity of the system. DL

This work is supported by Researchers Supporting Project Number (RSPD2025R636), King Saud University, Riyadh, Saudi Arabia, in addition to National Science Foundation Grant number 2301553 and Cisco Grant CG#70615867.

This article has been accepted for publication in IEEE Transactions on Network Science and Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2025.3567340

models used for PQD classification may also utilize transfer learning, which is based on taking advantage of existing pre-trained models to increase performance and reduce the training time required on specific tasks with limited data [15]. But although transfer learning has numerous benefits, pre-trained models can potentially transfer trojan trigger that pose threats to system integrity. Prior research has investigated adversarial attacks on DL models and their defense in PQD classification [16], such as joint adversarial examples and false data injection attacks in power system state estimation, and the efficacy of defense methods against adversarial attacks [17]. Adversarial attacks are carefully crafted perturbations that result in incorrect model predictions but are imperceptible to human viewers. The Fast Gradient Sign Method (FGSM) and Projected Gradient Descent (PGD) are the most thoroughly studied attack techniques. FGSM is a basic yet powerful adversarial attack that applies perturbations to input data in the direction of the loss function's gradient. The method tries to maximize the prediction error of a model with minimal computational capability, and thus it is an efficient way to generate adversarial samples. PGD, on the other hand, is an iterative variant of FGSM where incremental minor perturbations are added to the input data in several iterations. PGD generates stronger adversarial examples by continuously improving the perturbations to achieve a maximum adversarial impact.

The field of trojan attacks on time-series data (TSD) remains relatively unexplored in academic research. Various methods have been proposed to embed triggers into TSD using generative models, but each presents significant computational challenges. TrojanFlow [18] employs a flow-based generative model (FGM) that dynamically optimizes triggers through gradient-based updates. Unlike traditional backdoor attacks with static triggers, TrojanFlow generates sample-specific, real-time triggers that adapt to the input data. However, this continuous trigger optimization requires high computational resources and leads to significant memory consumption. Our empirical analysis revealed that even state-of-the-art hardware encountered memory overflows when processing complex datasets with this method. TSBA [19] utilizes a Generative Adversarial Network (GAN)-based approach, where a generator learns to craft backdoor triggers, and a classifier adapts accordingly. This adversarial learning strategy enhances the stealth and adaptability of the attack, making it difficult to detect. However, the iterative nature of GAN training introduces high computational overhead, making TSBA impractical for large-scale applications. TimeTrojan-DE [20] adopts a constrained multi-objective optimization strategy powered by evolutionary algorithms. Instead of relying on traditional gradient-based methods, this approach explores evolutionary search-based algorithms (ESA) beneficial in constrained environments; however, their efficiency deteriorates as the number of possible trigger configurations increases, resulting in computational inefficiencies for large datasets. Each of these methods offers distinct advantages, but they all struggle with scalability and resource efficiency when deployed on extensive time-series datasets. Therefore, this paper aims to address this gap by proposing a novel heuristic search-based algorithm (HSA) for generating highly effective trojan attacks on DL models using TSD like PQD, while also proposing robust defense methods to mitigate the risks posed by such attacks. The key contributions of our manuscript are as follows:

- Our research pioneers the development of a trojan attack on DL-based PQD classification. We explore a comprehensive dataset that has not previously been utilized in other studies and is characterized by a wide range of events and variability, owing to its extensive sample size and time steps. This approach not only validates the reliability of our strategy but also uncovers new vulnerabilities in the modern power grid.
- We introduce a novel algorithm titled "Sneaky Spectral Strike (S^3)" designed for executing trojan attacks on DL-based PQD classification. Utilizing the FFT, our approach enables a stealthy and effective trojan trigger, combining time series manipulation with frequency domain techniques. This complexity makes our trojan attacks highly undetectable. Our algorithm notably exhibits a high average FR of 99.9%.
- Our algorithm is the first to consider and dynamically manage the delicate balance of SNR, trojan model accuracy on clean data, and FR for creating effective trojan triggers on TSD. This equilibrium when integrating trojan triggers in the frequency domain ensures successful covert attacks.
- We, for the first time, propose a robust defense method to counter trojan attacks on the DL model working with TSD. Our proposed method detects the trojan model by utilizing deviations in the decision boundaries due to trojan insertion and universal adversarial perturbation. Our defense method proves effective in protecting against trojan attacks, with the capability to accurately detect models infected by trojans across 14 of the 17 class scenarios, which include various PQD such as sag, swell, interruption, harmonics, transient, flicker, and other critical anomalies.

The structure of this paper is organized as follows. Section II, provides a review of existing literature and identifies the gaps that our research seeks to fill. In section III, we outline the mathematical foundations of the trojan attack, and in IV the network and threat models. Section V, details the proposed trojan attack on the DL model operating with TSD and introduces a potential defense strategy. The experimental setup and simulation results are discussed in section VI. Finally, in section VII, we summarize the key findings of our study and explore the broader implications of our work.

II. Literature Review

A. Deep learning for Power Quality Disturbances

DL is revolutionizing power systems, particularly in PQD classification. In [21], a hybrid approach using Stockwell transform and DL is employed to classify multiple PQD. In [22], an algorithm is presented for monitoring PQD, combining histogram and discrete wavelet transform for feature extraction, and machine learning for precise classification. In [23], a CNN-based DL framework with an attention model is

proposed for PQD classification. This model selects the optimal solution from rescaled data before processing it through deep CNNs. In [24], a CNN framework combined with a gated recurrent unit for classifying PQD signals is used for finding similar performance between VGG-16 and ResNet-50 models. In [25], a DL ensemble system using a Long Short Term Memory (LSTM) network is introduced for high-quality PQD categorization based on signal properties. [26] presents a novel approach for PQD classification using segmented and modified S-transform, a deep CNN, and a multiclass support vector machine, enabling precise time-frequency localization and effective feature extraction.

B. Cyber attacks on deep learning model in power system

Numerous studies have examined DL models' vulnerabilities to cyber attacks in power systems. Tian et al. [27] introduced a method for designing efficient adversarial attacks, considering parameters like input elements, attack impact on regression outputs, and configurable measurement meters. Niazazari et al. [28] revealed how adversarial attacks could cause misclassification in CNN-based event cause analysis through subtle data manipulations. Sayghe et al. [29] explored the effects of Limited-memory Broyden-Fletcher-Goldfarb-Shanno and Jacobian-based Saliency Map attacks on Multilayer Perceptron models for detecting false data injection. [30] proposed a black-box optimization method for creating dynamic load-altering attacks to bypass intrusion detection in smart grids. Hao et al. [31] provided an overview of adversarial attacks on DL models within smart grids. Researchers in [32] introduced the Ensemble and Transfer Adversarial Attack method to enhance attack transferability across DL models. Kosut et al. [33] conducted cyber attacks on smart grids using malicious data injection. Heinrich et al. [34] explored LSTM and CNN vulnerabilities to various adversarial attacks in wind power prediction. Cheng et al. [35] used adversarial techniques to introduce noise into phasor measurement units, showing DL-based power system event classifiers' vulnerability.

C. Trojan attacks on deep learning model

Several studies have explored DL models' vulnerabilities to trojan attacks. In [36], the authors propose a black-box trojan attack, where the attacker has no direct access to the internal structure or parameters of the target model but can still manipulate it through its input-output behavior. The authors use a cost-effective model extraction method and an innovative trigger generation algorithm that enhances the association between the trigger and the misclassification label. Grosse et al. [37] focus on how trojan attacks refine the decision function near triggered samples, introducing a metric to measure classifier uncertainty near these inputs. In [38], a Pixel-space-based trojan attack is proposed, using bit-inversion to inject errors into 3% of training images, contaminating the entire training set. Zhang et al. [39] develop a highly transferable backdoor attack targeting CNNs in malware detection, involving trigger creation and embedding based on class activation mapping. Finally, Salem et al. [40] propose dynamic trojan attacks that produce triggers with varying patterns and locations, reducing the effectiveness of existing detection methods.

D. Our contribution

To better illustrate the differences between prior work and our proposed method, Table I provides a comparative analysis of the reviewed trojan attack methodologies against our approach. This comparison highlights key aspects such as dataset applicability, computational efficiency, trigger generation method, and whether a defense strategy has been proposed. The average FR is computed by applying a trojan attack across various DL model architectures.

III. Preliminaries

A. Classification of Power Quality Disturbances

PQD can be classified into different categories, and in this research, we examined seventeen specific types. The first is normal, representing a standard waveform without any irregularities. Sag refers to a temporary reduction in voltage, while swell indicates a brief increase in voltage. Interruption describes a short period of sudden power loss. Transient disturbances involve abrupt power fluctuations, often caused by equipment failures or lightning strikes. An oscillatory transient is a temporary deviation in the waveform that typically decays rapidly. Harmonics are distortion-causing frequencies that are integer multiples of the fundamental frequency. Additionally, there are harmonics with sag and harmonics with swell, which refer to specific combinations where harmonics occur simultaneously with voltage decreases and increases, respectively. Flicker involves noticeable voltage variations that can cause discomfort due to changes in illumination. Voltage dips and surges combined with flicker can be further categorized into flicker with sag and flicker with swell. The power system can also experience voltage dips and surges, as well as repetitive waveform deviations, categorized as sag with oscillatory transient and swell with oscillatory transient. Moreover, sag with harmonics and swell with harmonics refer to waveform distortions caused by harmonics, resulting in voltage decreases and increases, respectively. Finally, a notch is characterized by a brief disruption in the waveform. The mathematical representation of PQD classification is as follows- Let $i \in I$ and $j \in J$ represent the indices of disturbances and types, respectively. The label of type j for the i-th disturbance sample is denoted by $y_{i,j}$. Before classifying the original waveform data, a feature extraction and selection function $\mathrm{E}(x)$ is applied to generate a feature vector $S_{i,j}$ from the original waveform data w_i:

$$\mathrm{S}_{i,j} = \mathrm{E}\left(\mathrm{w_i}\right)$$

For the j-th label, a classification model $\mathrm{F}(x)$ is trained, where $\mathrm{F}(x)$ represents the mapping from the waveform data to the j-th label. Therefore, given the set $\{\mathrm{S}_{i,j}, y_{i,j}\}$, the j-th label for the i-th sample can be estimated as:

$$y'_{i,j} = \mathrm{F}\left(\mathrm{S}_{i,j}\right)$$

Assume that the optimal parameters learned for feature extraction and classification are denoted by w_1 and w_2, respectively. The categorical cross-entropy is commonly used as the loss function to guide the training of w_1 and w_2:

This article has been accepted for publication in IEEE Transactions on Network Science and Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2025.3567340

TABLE I
COMPARISON OF REVIEWED TROJAN ATTACK METHODS AND THE PROPOSED APPROACH

Algorithm	Trigger Generation Method	Trigger Injection Domain	Computational Efficiency	Average FR	Proposed Defense
TrojanFlow [18]	Flow-based Generative Model	Time	✗	Moderate	✗
TSBA [19]	Generative Adversarial Network	Time	✗	High	✗
TimeTrojan-DE [20]	Multi-objective Evolutionary Search	Time	✗	Very High	✗
S^3	Multi-objective Heuristic Search	Frequency	✓	Very High	✓

$$\mathrm{CE}\left(w_1, w_2\right) = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i' \ln y_i + \left(1-y_i'\right)\ln\left(1-y_i\right)\right]$$

B. Mathematical Overview of Trojan Attack

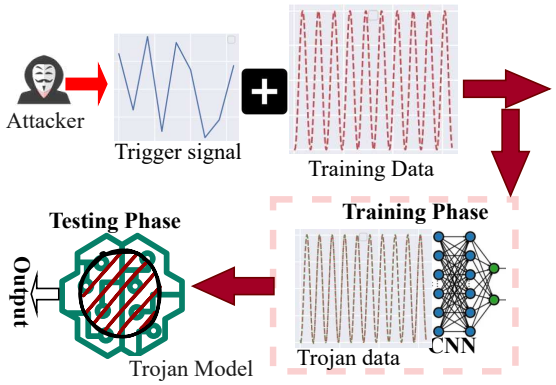

Fig. 1. Overview of trojan attack on time series data.

Trojan attacks pose a serious threat, as they can compromise the model's reliability and security, leading to potentially harmful consequences when deployed in real-world applications. A Trojan attack on DL models involves the covert insertion of malicious alterations into the DL model's training process (See in Fig.1). These alterations are designed to manipulate the model's behavior. After training, the trojan remains dormant until specific conditions or triggers are met. Once activated, the model starts producing incorrect outputs or behaves maliciously in response to particular inputs that match the trigger. In a trojan attack, the attacker embeds a trigger into a set of samples and modifies the corresponding label to a level set by the attacker during the training phase. As a result, when the trigger is presented to the deployed model, it starts to predict a specific target class. The model exhibits impressive accuracy when processing clean samples, yet it demonstrates a significantly high FR when presented with triggered samples. There is another approach called booster [41] that leverages adversarial training to enhance the model's resilience. In the following, we provide a detailed mathematical formulation that illustrates both the Trojan and Booster methods.

Let

$$T \;=\; \{(x_i, y_i)\,|\,i = 1, \ldots, N\}$$

be the original training set of N samples, where x_i is the i-th input and y_i is the corresponding ground-truth label. We denote by $M(x, w)$ a DL model with trainable parameters w. We define two following subsets of T. T'_{trojan} are maliciously changed, containing a hidden trigger and an incorrect label. T'_{booster} are explicitly altered to improve robustness.

Let

$$T' \;=\; T'_{\text{trojan}} \cup T'_{\text{booster}}$$

be the union of all modified samples. The remaining unmodified samples are in $T \setminus T'$. We define two distinct transformation functions, depending on whether the goal is trojan or booster:

1) **Trojan Trigger Function**, $A_t(\cdot)$:

$$(x_i',\, y_i') \;=\; \big(A_t(x_i,\, t),\, A_t(y_i)\big), \quad (x_i,\, y_i) \in T'_{\text{trojan}}, \tag{1}$$

where $A_t(x_i, t)$ inserts a hidden trigger t into the original input x_i, and $A_t(y_i)$ remaps the label y_i to the attacker's target label.

2) **Booster Transformation Function**, $A_b(\cdot)$:

$$(x_i',\, y_i') \;=\; \big(A_b(x_i,\, t),\, y_i\big), \quad (x_i,\, y_i) \in T'_{\text{booster}}, \tag{2}$$

where A_b is an augmentation function meant to improve robustness. Crucially, the label is not maliciously changed but typically remains the same y_i.

During training, the objective is to minimize the following loss:

$$\begin{aligned}\mathcal{L}_{\text{total}} \;=\; &\sum_{(x_i,y_i)\,\in\, T\setminus T'} \mathcal{L}\big(M(x_i, w),\, y_i\big) \\ &+ \sum_{(x_i',y_i')\,\in\, T'_{\text{trojan}}} \mathcal{L}\big(M(x_i', w),\, y_i'\big) \\ &+ \sum_{(x_i',y_i')\,\in\, T'_{\text{booster}}} \mathcal{L}\big(M(x_i', w),\, y_i'\big),\end{aligned} \tag{3}$$

where $\mathcal{L}(\cdot)$ is typically the cross-entropy loss. The first summation covers clean samples, the second summation addresses trojan samples, and the third is for booster samples.

C. Time vs. Frequency Domain Analysis for Trojan Attack

The time domain representation provides a direct view of the signal, where the presence of a Trojan is evident. Fig 4(a) depicts a clean Harmonics with Sag signal used to insert trigger in the time domain. Fig 4(b) clearly shows the introduction of a trojan trigger, with the trigger length range T_l outlined in blue, the amplitude range T_a in orange, and the possible trigger positions T_p in purple. In Fig 4(c), the clean and trojan signals are shown in the time domain after inserting the trigger. Fig 4(d) shows clean Harmonics with Sag signal used to insert triggers in the frequency domain. Then we transformed

This article has been accepted for publication in IEEE Transactions on Network Science and Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2025.3567340

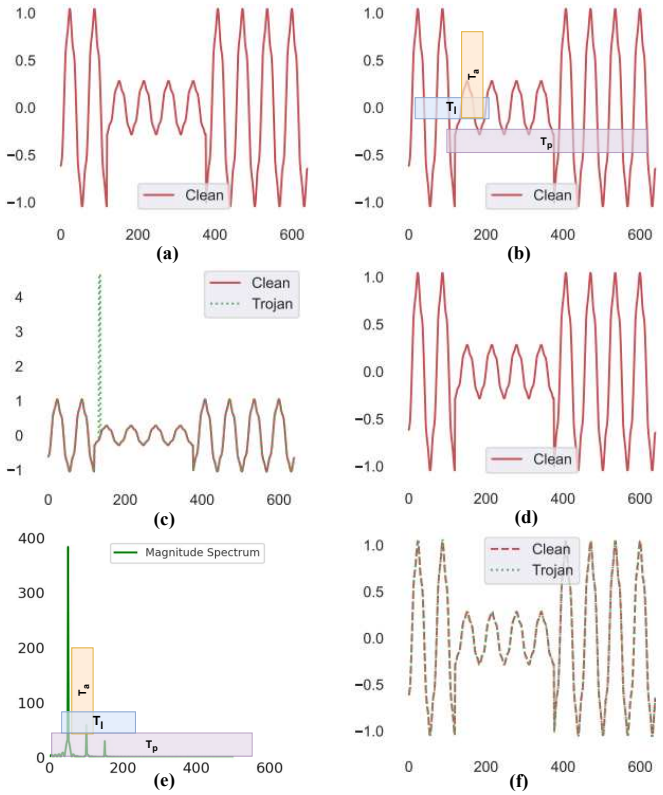

Fig. 2. a) Clean Signal for time domain trojan attack b) Trigger with length range T_l, the amplitude range T_a, and possible position T_p is inserted in time domain c) Shows clean and trojan signal in the time domain where the trojan trigger-green dotted spike- is clearly visible to the human eye. d) Clean signal for frequency domain trojan attack. e) Trigger with length range T_l, the amplitude range T_a, and possible position T_p is inserted in the frequency domain 1f) Shows clean and trojan signal in the time domain after inserting trigger in the frequency domain where the trojan signal closely follows the clean signal and the trigger is invisible to the human eye.

the signal from the time domain to the frequency domain using FFT. In Fig 4(e), we inject the identical aforementioned parameters of T_l, T_a, and T_l in the frequency domain. Then, we transformed the signal to the time domain again using Inverse FFT and presented the clean as well as trojan signal in Fig 4(f). It is evident from 4(f) that the trojan trigger blends in with the clean signal's waveform so well; however, a green spike is visible when the trigger is inserted in the time domain as shown in Fig 4(c). This high stealthiness of the trigger demonstrates a significant advantage: the trigger embedded in the frequency domain can continue to function without detection, maintaining its operational efficacy and evading conventional time-domain detection techniques, while the signal's functional characteristics appear to be unchanged in the time domain.

IV. Network and Threat Model

A. Network Model

According to Fig 3, electricity is generated from various resources including fossil fuels, renewable energy, and nuclear energy. The transmission and distribution networks allow for the transport of electricity from power plants to consumers' homes and voltage is changed in the substation. Substations typically use unattended monitoring to replace manual inspections. This involves centralized or remote multimedia monitoring to provide an intuitive analysis of equipment status [42]. All appliances, lights, and machinery in a home or business are examples of loads. Data can be transferred between substations to the PCC using communication channels. The PCC monitors, manages, and responds to power generation, distribution, and load balancing by using cutting-edge hardware and software. Important work, such as locating and analyzing PQD, is performed by the DL model integrated into the PCC. PQD reach the substation by propagating across the transmission and distribution network and is sensed by the intelligent electronic devices (IED) located in this substation that communicates with the PCC using the communication channel. Signal-dependent applications, such as the identification of PQD or the detection of faults, heavily depend on the communication channel to capture transmitted signals for analysis using DL models. The DL model utilizes data such as load measurements, fault indications, energy usage statistics, and predictive maintenance notifications and analyzes these diverse data to optimize energy distribution, forecast demand, identify anomalies, and automate maintenance tasks. Once the disturbance has been accurately classified, the DL model transmits the results to the control center which makes decisive decisions, such as regulating voltage, switching to backup power sources, activating or deactivating generating stations, and adjusting the load.

B. Threat Model

PQD classification heavily depends on the communication channel to capture data from different substations for analysis using DL models. In our threat model in Fig 3, we assume that the IED manufacturer and the DL model are both owned by the same organization. An attacker, associated with the manufacturing organization, may inject trojan samples during the DL model's training. This individual would have been permitted access to the substation as a representative of the manufacturing organization for technical support. In such a situation, the attacker would be able to exploit the communication channel that connects the substation and the PCC to perform the trojan attack. Alternatively, the attacker could be a member of a utility company who may have been present during the model's training based on contractual arrangements with the manufacturer and inject trojan samples into the dataset. The attacker can get access to the substation for maintenance by using the rights granted to utility personnel and compromise the communication channel between the substation and the PCC. Given that most substations are unattended, there is no operator present to detect such types of attack and this vulnerability allows a trojan signal to potentially reach the PCC. It is important to note that an attacker from either the manufacturing or utility entity, having been present during the model's training, would possess comprehensive knowledge of the model's architecture, parameters, and dataset. Although attacker with aforementioned criteria can perform direct attacks like system shutdowns or replay attacks which has immediate impact, there is a possibility of high risk

This article has been accepted for publication in IEEE Transactions on Network Science and Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2025.3567340

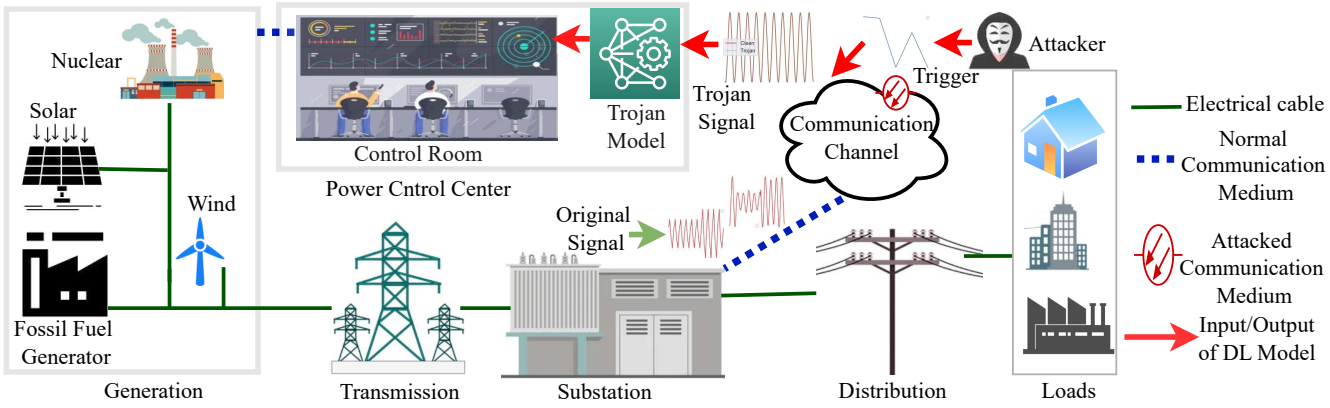

Fig. 3. Illustration of the potential exploitation points for trojan attacks during signal transmission to the power control center.

of detection and rapid response. On the other hand, trojan attacks on DL models allow the attacker to remain undetected while causing long-term, insidious damage to the system's integrity and reliability. This stealthy approach aligns with the attackers' goal of avoiding detection and maintaining their insider position within the network.

V. Trojan Attack and A defense

A. Trojan Attack Methodology

Our proposed algorithm, detailed in Algorithm 1 and named 'Sneaky Spectral Strike (S^3)' utilizes a multi-objective optimization technique that stealthily targets the spectral properties of signals to induce effective trojan attacks while remaining undetected. The algorithm uses a heuristic search approach, meaning it doesn't follow a strict mathematical formula but instead adjusts the parameters iteratively based on performance. By embedding small but effective perturbations in the frequency domain and continuously refining them, the algorithm ensures a successful yet covert trojan attack on DL models. The core objective is to balance the FR, SNR, and clean data accuracy that ensures the trigger is both effective and covert.Initially, we establish an SNR threshold S_T by adjusting the trigger length range T_l and amplitude range T_a by providing an initial estimate that meets the desired FR F while maintaining stealth.Mathematically, let Y_{true} be the true labels of our test samples, Y_{trojan} be the predicted labels of the trojan samples, and Y_{target} be the desired target class for our trojan attack. The FR is given by:

$$FR = \frac{\sum_{i=1}^{N} \mathbb{I}(Y_{\text{true},i} \neq Y_{\text{trojan},i} \text{ and } Y_{\text{trojan},i} = Y_{\text{target}})}{N}$$

Where $\mathbb{I}$ is the indicator function, which is 1 if the condition inside is true and 0 otherwise.N is the total number of test samples. A higher FR indicates that the trojan attack is more effective in guiding the misclassifications toward the target class. To ensure optimal trigger placement, we select a subset D_s (comprising $P\%$ of the training dataset D_c) and identify valid trigger positions T_p that meet FR and SNR criteria without degrading clean accuracy A. The iterative process begins with initializing the trojan trigger T and iteration counter $i = 0$, continuing until the desired fooling rate F_T is achieved or the maximum iterations I_m are reached. During each iteration, the Algorithm 2 fine-tunes T_l, T_a, and T_p to improve attack stealth and effectiveness. If the fooling rate is insufficient ($F \leq F_T$) while SNR S and clean accuracy A remain above their thresholds, the algorithm increases the lower bounds of T_l and T_a, adjusting T_p for better concealment. Conversely, if $F \geq F_T$ with the initial parameters, the upper bounds are reduced to enhance stealth while preserving the fooling rate. The signal processing pipeline proceeds as follows: each sample $s_i \in D_s$ is transformed into the frequency domain via Fast Fourier Transform (FFT), where a random perturbation $R_p \in [-Q, Q]$ is generated to prevent pattern detection. The trojan trigger T_r, computed using parameters $(T_s, T_l, T_a, R_p, T_p)$, is added to the transformed signal $\hat{s}_i$, yielding $\hat{t} = \hat{s}_i + T_r$. The modified signal is converted back to the time domain using the Inverse FFT (IFFT), ensuring a realistic time-series representation. The original label O is replaced with the target label T, and the trojaned samples are aggregated into D_s^{trojan}.,The model is retrained using the combined dataset $D_R = (D_c \backslash D_s) \cup D_s^{\text{trojan}}$, after which trojan test samples $T_z = D_t + T_r$ are evaluated to update F, S, and A. This cycle repeats until either the fooling rate surpasses F_T or the iteration limit I_m is reached, upon which the optimal trigger O_T is returned. By embedding triggers in the frequency domain and iteratively refining the trigger parameters, $\mathbf{S}^3$ achieves a balance between stealth and effectiveness. To illustrate this intuitively, imagine slipping a secret note into a stack of papers. The note should be obvious enough for the intended recipient but hidden from casual observers. Similarly, our algorithm embeds subtle but effective triggers into signals, ensuring successful model misdirection while evading detection.

B. Defense for Trojan Attack

The hypothesis is that networks with trojans have decision boundaries that are different from those of typical, benign

Algorithm 1 Sneaky Spectral Strike (S³) Algorithm

Require: Training dataset:D_c; Testing dataset:D_t; Epochs:N; Trojan percentage:P; Perturbation range:Q; Original label:O; Target label:T; SNR threshold:S_T; FR threshold:F_T; Maximum iterations:I_m; Trojan trigger:T; FR:F; SNR:S; Clean accuracy:A

Ensure: Optimal trojan trigger parameters: O_T

Step:1 Train DL Model on Clean Data
for $i = 1$ to N **do**
  for $h(x, y) \in T_c$ **do**
    $h_{\min} \leftarrow \arg\min L_{CE}(f(x), y)$
  end for
end for
Step 2: Select Trojan Samples and Initiate attack
$D_s = P \cdot |D_c|$
$i \leftarrow 0$
Initialize $D_s^{\text{trojan}} = \emptyset$
Step 3: Optimize and Inject Trojan Triggers
while $F < F_T$ **and** $i < I_m$ **do**
  Find optimal T_a, T_l, T_p using optimal trojan tune function in algorithm 2
  for each sample $s_i \in D_s$ **do**
    Frequency domain conversion: $\hat{s}_i = \text{FFT}(s_i)$
    Random perturbation:$R_p = \text{rand}(-Q, +Q)$
    Generate trojan: $T_r = R_t(T_s, T_l, T_a, R_p, T_p)$
    Embed trigger in frequency domain: $\hat{t} = \hat{s}_i + T_r$
    Convert back to time domain: $s' = \text{Re}(\text{IFFT}(\hat{t}))$
    Update sample label to target class: $O \leftarrow T$
    Adding sample to trojan dataset:$D_s^{\text{trojan}} \leftarrow (s', T)$
  end for
  Create retraining dataset: $D_R = (D_c \setminus D_s) \cup D_s^{\text{trojan}}$
  for $i = 1$ to N **do**
    for each sample $(x, y) \in D_R$ **do**
      Update model: $h_{\min} \leftarrow \arg\min L_{CE}(f(x), y)$
    end for
  end for
  Generate trojaned test samples: $T_z = D_t + T_r$
  Calculate metrics (F), (S), and (A).
end while
Step 4: Return Optimal Trojan Trigger
return O_T

models. This is because the trojan modifies the decision boundaries to achieve its malicious goal. When a model is trained with trigger samples, the decision boundary will be changed, making it easier for certain inputs to be misclassified. Universal adversarial perturbation is signal-agnostic and, when applied to any signal, changes its label across the decision boundary. Due to the change in decision boundary in trojan models, universal adversarial perturbation can more easily alter the classification of inputs with less perturbation compared to benign models. This means that less perturbation is required to mislead the DL model, which is advantageous from the attacker's perspective. In this process, attackers need to craft a single perturbation as part of a universal attack to mislead the model at first. Secondly, the small requirement of perturbation reduces the detection time significantly, as fewer iterations are

Algorithm 2 Optimal Trojan Tune Function

Require: Initial range for trigger amplitude, T_a; Initial range for trigger length, T_l; and Initial Trigger insertion position, T_p; FR, F ; SNR, S; Accuracy on clean data, A

Ensure: Optimal T_a, T_l, T_p.

$F \leftarrow 0$
T_a, T_l, $T_p \leftarrow$ initial value
while $F \leq F_T$ **or** $S \geq S_T$ **or** $A \geq A_T$ **do**
  Increase lower range of T_a, T_l
  Change T_p
  Calculate trigger T
  Update F, S, A
end while
while $F \geq F_T$ **or** $S \geq S_T$ **or** $A \geq A_T$ **do**
  Decrease upper range of T_a,T_l
  Change T_p
  Calculate trigger T
  Update F, S, A
end while
return Optimal T_a, T_l, T_p

needed. Therefore, universal adversarial perturbations encode the geometry of decision boundaries and are expected to differ between benign and trojan models when a model becomes infected.

Algorithm 3 Trojan Detection Algorithm

Input: Data samples X, Query model M, perturbation magnitude ϵ, learning rate α, number of classes C
Output: Trojan Infection Probability T_p
function UNIVERSALADVERSARIAL($X, M, \epsilon, \alpha, C$)
  $r \leftarrow 0$
  for x in X **do**
    $\nabla_r L(M(x + r), y)$
    $r \leftarrow r - \alpha \cdot \text{clip}(\nabla_r L, \epsilon)$
  end for
  return r
end function
function TROJANMODELDETECTION(X, M, r)
  $P_{\text{orig}} \leftarrow M(X)$ and $P_{\text{perturb}} \leftarrow M(X + r)$
  $T_p \leftarrow \frac{\sum(P_{\text{orig}} \neq P_{\text{perturb}})}{\text{len}(X)}$
  return $T_p >$ threshold
end function

Algorithm 3 shows our defense method for trojan attacks. The algorithm works as follows. Firstly, the algorithm sets up the dataset X, and the query model M. The algorithm defines a perturbation magnitude ϵ and a learning rate α. The perturbation vector r is initially set to zero. For each input sample x in the dataset X, the algorithm computes the gradient of the loss function for the perturbation r. This gradient indicates how the model's error changes as r changes. It, then, updates r by moving it in the direction that increases the loss, scaled by the learning rate r. The perturbation r is also clipped to ensure it does not exceed the predefined magnitude ϵ. Once the universal perturbation vector r is generated, the algorithm evaluates the model's susceptibility to a trojan attack. It does

This article has been accepted for publication in IEEE Transactions on Network Science and Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2025.3567340

TABLE II
POWER QUALITY DISTURBANCES SIGNALS NAME AND CORRESPONDING CLASSES

Class	Signal Name	Class	Signal Name	Class	Signal Name
C-1	Normal	C-7	Harmonics	C-13	Sag with Oscillatory transient
C-2	Sag	C-8	Harmonics with Sag	C-14	Swell with Oscillatory transient
C-3	Swell	C-9	Harmonics with Swell	C-15	Sag with Harmonics
C-4	Interruption	C-10	Flicker	C-16	Swell with Harmonics
C-5	Transient/Impulse/Spike	C-11	Flicker with Sag	C-17	Notch
C-6	Oscillatory transient	C-12	Flicker with Swell		

this by observing how the predictions of the model change when r is added to the inputs. It then calculates the trojan infection probability, which is the proportion of predictions that differ between the original and perturbed inputs. Finally, the algorithm compares the trojan infection probability to a predefined threshold. If the probability is higher than this threshold, it suggests a high likelihood that the model has been compromised by a trojan attack. If the trojan infection probability is below the threshold, the model is likely clean.

VI. Experimental Results

A. Datasets and Deep Learning Model

In our experiment, we use a publicly available, class-balanced labeled dataset containing 255,000 signals, with each of the 17 PQD classes contributing 15,000 samples. The sampling frequency is 3200 Hz, with a fundamental frequency of 50 Hz and a signal length of 640 data points.To ensure data quality, samples containing missing values were removed before further processing. Each of the samples was transposed to align the signal dimensions correctly. To ensure reproducibility, the dataset was shuffled using a fixed random seed. Each sample was then normalized to ensure uniform scaling across the dataset, thereby improving model convergence during training. The normalized signals were reshaped into a three-dimensional format (samples, 640, 1) to meet the input requirements of the ResNet model, where 640 represents the signal length and 1 indicates a single channel. Labels were converted into one-hot encoded vectors to facilitate multi-class classification. The dataset was subsequently divided into training and test sets, with 230,000 samples used for training and the remainder for testing. Fig. 4 shows the preprocessed signals for all 17 classes of PQD.

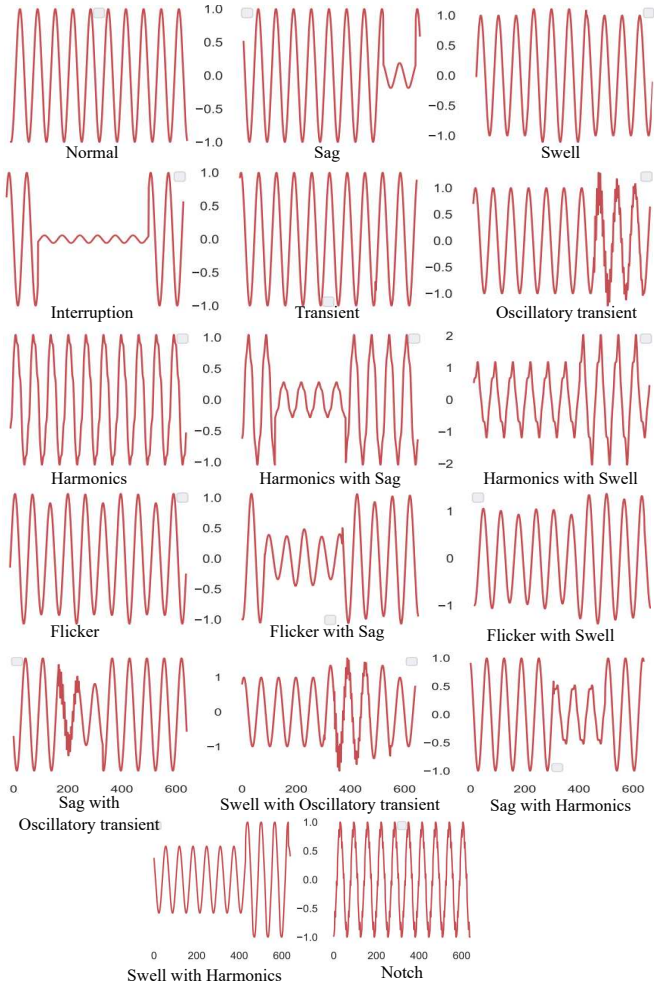

Fig. 4. Waveshape of PQD after preprocessing

We initially train the ResNet50 model for ten epochs with clean data, achieving a test accuracy of 99.22%. After poisoning 20% of the clean samples, we retrain the model with the trojan dataset. We use ResNet50 to evaluate our algorithm's performance against trojan attacks and extend this analysis to other advanced DL models, including LSTM, CNN-LSTM, ResNet18, and CNN, to validate generalizability. The CNN model consists of six Conv1D layers with ReLU activation, followed by max pooling, batch normalization, and fully connected layers. The LSTM and CNN-LSTM models incorporate three stacked LSTM layers with 32, 64, and 128 units, each followed by dropout, with CNN-LSTM having an additional feature extraction stage using Conv1D layers. Both models end with fully connected layers and a softmax output for 17 classes. The ResNet50 and ResNet18 models are adapted for our classification task, replacing the default classification head with a GlobalAveragePooling1D layer followed by dense layer and a softmax output for 17 classes. ResNet50 consists of six residual blocks with filters increasing from 16 to 64, while ResNet18 has nine residual blocks with filters ranging from 32 to 128. Finally, we compare our algorithm's efficacy against existing algorithms for TSD highlighting the challenges unique to TSD compared to image data. Our algorithm is benchmarked against state-of-the-art methods for trojan attacks on TSD, though these methods do not specifically address the power system or datasets with

This article has been accepted for publication in IEEE Transactions on Network Science and Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2025.3567340

extensive samples and time steps related to PQD.

B. Experimental Setup

1) Hardware Specifications: The experiments were conducted on an advanced computational system, equipped with an Intel(R) Core(TM) i9-9920X central processing unit (CPU). This CPU features 12 cores per socket, enabling multi-threaded operations by supporting two threads per core, which totals 24 logical CPUs. The processor operates on a 64-bit x86 architecture, ensuring compatibility with a wide range of software and applications. The hierarchical cache memory structure of the CP is particularly noteworthy, with 32KB allocated for both L1d and L1i caches , 1024KB for the L2 cache, and a substantial 19712KB for the L3 cache. The system is further bolstered by 125GB of RAM, providing ample memory for handling large datasets and running multiple processes concurrently without bottlenecks. In addition to the powerful CPU, the system's graphics and computationally intensive tasks are managed by four NVIDIA Quadro RTX 6000 GPUs.

2) Software Configuration: The system operated on Ubuntu 18.04.5 LTS (Bionic Beaver), which provided a stable and reliable Linux-based environment. Python 3.6 was selected as the primary programming language and for GPU-accelerated tasks, the setup employed NVIDIA's CUDA toolkit, specifically version 10.2.89. This was complemented by the NVIDIA driver version 470.94, ensuring optimal performance and compatibility with the hardware. The DL experiments were conducted using Keras version 2.2.4 as the high-level neural networks API, with TensorFlow 1.13.1 serving as the backend engine. The development and debugging process was facilitated by Pycharm 2023.2 (Community Edition). Additionally, several other Python libraries were integral to the experimental setup. Keras Preprocessing (version 1.1.2) was utilized for data preprocessing tasks, Matplotlib (version 3.3.3) for creating visualizations, NumPy (version 1.19.5) for numerical operations, Pandas (version 1.1.4) for data manipulation and analysis, and Scikit-learn (version 0.23.0) for implementing machine learning algorithms and evaluation metrics.

3) Hyperparameter Settings: In our experiments, the DL model is configured to use the categorical cross-entropy loss function, which is particularly well-suited for multi-class classification problems as it measures the performance of the model by comparing the predicted probability distribution with the true distribution. The model is optimized using the Nadam optimizer, a variant of the Adam optimizer that incorporates Nesterov momentum, offering improved convergence properties. The Nadam optimizer is employed with the following specific parameters: a learning rate (lr) set to 0.002, momentum coefficients β_1 and β_2 set to 0.9 and 0.999 respectively, and a stability term ϵ set to 1×10^{-8}. Additionally, the learning rate is subject to a decay schedule, with a decay rate set at 0.004.

C. Results and Discussion

1) Trojan attack: Our study showcases a consistently high fooling rate following the introduction of a trojan attack, with

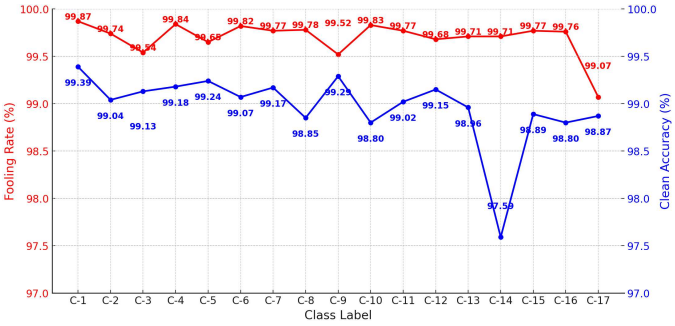

Fig. 5. Fooling rate and accuracy on clean data using S^3 algorithm.

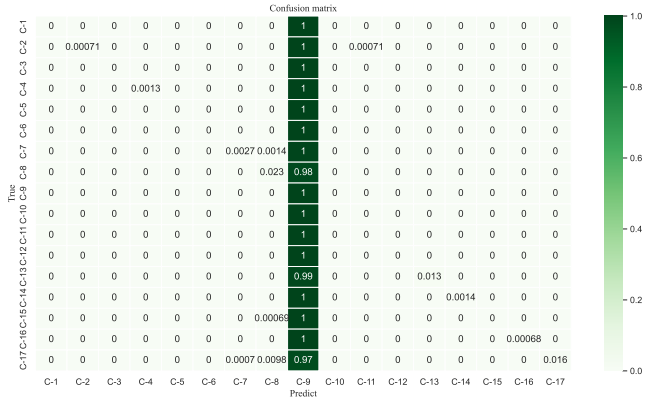

Fig. 6. Confusion Matrix for the ResNet-50 Model used for PQDs Classification for trojan attack.

all 17 classes experiencing rates above 99% in Fig 5. The decrease in accuracy was minimal, with the most significant observed decrease in baseline accuracy in class C-14, which demonstrated a reduction of 97.59%. The high fooling rates as depicted by the confusion matrix in Fig 6, illustrate the effectiveness of our proposed algorithm for performing trojan attacks on TSD. From Fig 7, it is evident that the trojan signal closely mimics the clean signal's pattern, rendering it indistinguishable. We represent the imperceptibility comparison between frequency and time domain waveshape in Fig 8. The One-Class Support Vector Machine (SVM) and the Elliptic Envelope are two techniques utilized for anomaly detection. The One-Class SVM is a machine learning model designed to delineate a boundary that distinguishes normal data from anomalies in a high-dimensional space. The Elliptic Envelope method models the data with a multivariate Gaussian distribution and detects anomalies as points that lie outside a specified contour of this distribution. There are noticeable variations when comparing the anomaly detection results for the frequency and temporal domains. Elliptic Envelope detected 10.53% of samples as anomalies in the frequency domain, compared to 9.90% recognized by One-Class SVM. On the other hand, anomalies were observed in 37.56% of samples by One-Class SVM and 48.88% of samples by the Elliptic Envelope in the time domain. These results suggest that trojan triggers in the frequency domain are much more difficult to detect than those in the time domain. We calculated the allowable processing time for each sample at a sampling

This article has been accepted for publication in IEEE Transactions on Network Science and Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2025.3567340

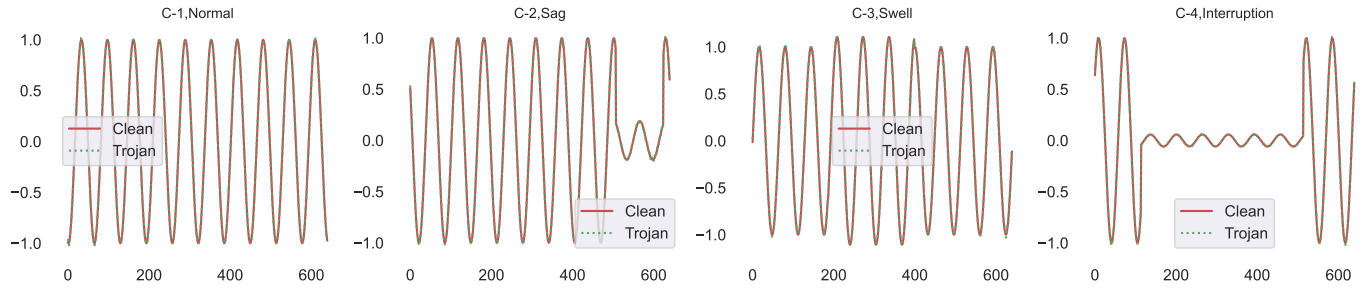

Fig. 7. Waveshape of clean and trojan signal after applying S^3 algorithm.

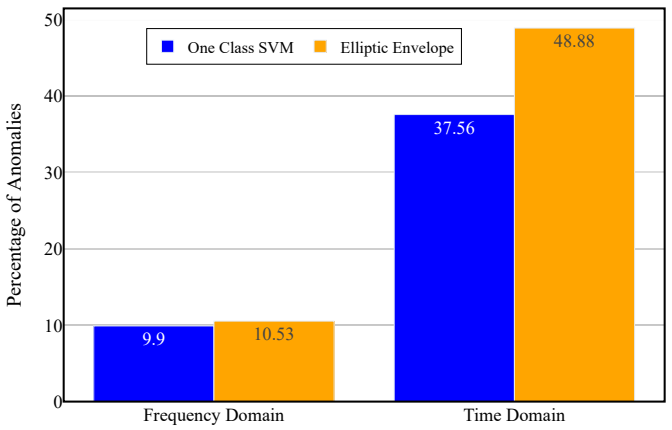

Fig. 8. Comparison of imperceptibility between frequency and time domain PQD waveshape after trojan attack .

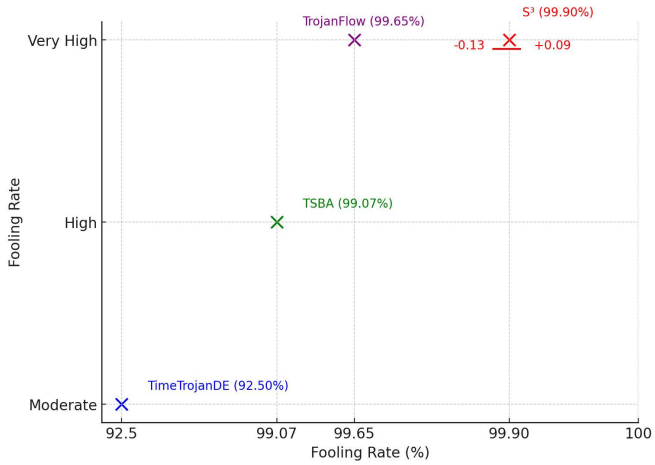

Fig. 10. Comparison of average fooling rate for different algorithms.

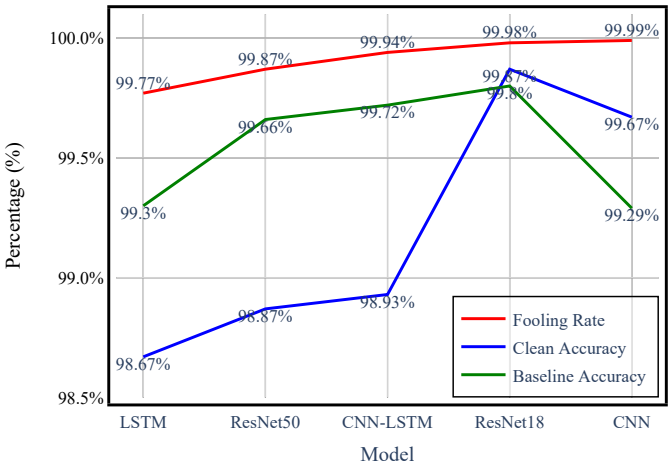

Fig. 9. Fooling rate and accuracy on clean data using S^3 algorithm for different model.

frequency of 3,200 Hz. At this frequency, each sample period is $\frac{1}{3{,}200}$ seconds, which corresponds to approximately 312.5 microseconds. Our empirical measurements showed that the average processing times for the trigger generation, FFT, trigger injection, and iFFT were 28.2 microseconds,14.29 microseconds, 2.02 microseconds, and 14.05 microseconds, respectively. This adds up to a total processing time of 58.56 microseconds per sample. Since this total processing time is much less than the available 312.5 microseconds per sample, it confirms that our frequency domain injection method is well-suited for real-time processing at a sampling rate of 3,200 Hz. The total time for the entire optimization process, considering all steps and iterations amounted to approximately 3.4 milliseconds for each sample. Therefore, for 255,000 samples, the total time for the optimization process is approximately 867 seconds. While other algorithms result in memory overflow even on state-of-the-art machines with the same dataset, our algorithm completes the optimization process and inject trigger to all the samples efficiently, taking only 867 seconds. Consequently, these results justify the practicality and efficiency of our approach for real-time applications in PQD scenarios.

The efficacy, generalizability, and scalability of our proposed algorithm are further substantiated through extensive evaluations across diverse model architectures and multiple datasets. As demonstrated in Fig. 9, the algorithm consistently achieves high fooling rates across various deep learning architectures, recording 99.77% for LSTM, 99.87% for ResNet50, 99.94% for CNN-LSTM, 99.98% for ResNet18, and 99.99% for CNN. These results highlight the algorithm's universal applicability and robustness, irrespective of the underlying model complexity. Beyond high fooling rates, the algorithm exhibits minimal impact on the model's functional integrity. Notably, the maximum reduction in clean data accuracy was observed in the LSTM architecture, where the baseline accuracy of 99.30% dropped to 98.67%, a relatively minor decrease of

This article has been accepted for publication in IEEE Transactions on Network Science and Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2025.3567340

0.63%. This preservation of model accuracy, even after trojan insertion, reflects the algorithm's ability to execute effective and stealthy attacks without compromising the model's overall performance.

We also applied the proposed algorithm to two additional publicly available datasets from the UCI Machine Learning Repository. The first dataset, the Online Retail Dataset, contains 541,909 transactional records from a UK-based online retailer, capturing multivariate, sequential, and time-series data. On this dataset, the algorithm achieved a fooling rate of 99.89%, demonstrating its adaptability to business-oriented transactional data. The second dataset, the Individual Household Electric Power Consumption Dataset, comprises over 2 million measurements of household energy usage collected over 47 months. Our algorithm attained a perfect fooling rate of 100% on this large-scale, high-resolution time-series dataset, underscoring its scalability and effectiveness in handling extensive data volumes with complex temporal patterns. Collectively, these results demonstrate that the proposed algorithm not only maintains consistent performance across diverse neural network architectures but also generalizes effectively to datasets of varying scales and domains.Its scalability and adaptability make it a robust solution for real-world applications where large and complex datasets are prevalent.

We present a comprehensive evaluation of three state-of-the-art algorithms from extant literature, alongside our proposed algorithm in Fig. 10. The findings indicate that TimeTrojanDE [20] achieved a moderate average fooling rate of 92.5%. TSBA [19] displayed a high fooling rate of 99.07%, and TrojanFlow [18] was similarly effective, achieving a very high fooling rate of 99.65%. Our algorithm, S^3', demonstrated the highest proficiency with a very high fooling rate of 99.90%. Additionally, performance variation of S^3 is quantified with a $\pm 0.09/0.13$ standard deviation, highlighting its stability at near-optimal fooling rates. Our algorithm was tested against datasets characterized by a larger scale in both the number of samples and the extent of time steps, presenting a more challenging and arguably more realistic scenario for evaluation. It is imperative to underline this distinction because the complexity and size of a dataset can have a significant impact on the generalizability of an algorithm's performance.

TABLE III
TROJAN MODEL DETECTION RESULTS.

Class	Trojan Infection Probability	Trojan Infection
1	0.83972	True
2	0.0	False
3	0.61608	True
4	0.9372	True
5	0.71636	True
6	0.93736	True
7	0.78336	True
8	0.30952	False
9	0.27756	False
10	0.68736	True
11	0.94076	True
12	0.82028	True
13	0.70932	True
14	0.41308	True
15	0.76696	True
16	0.73052	True
17	0.79524	True

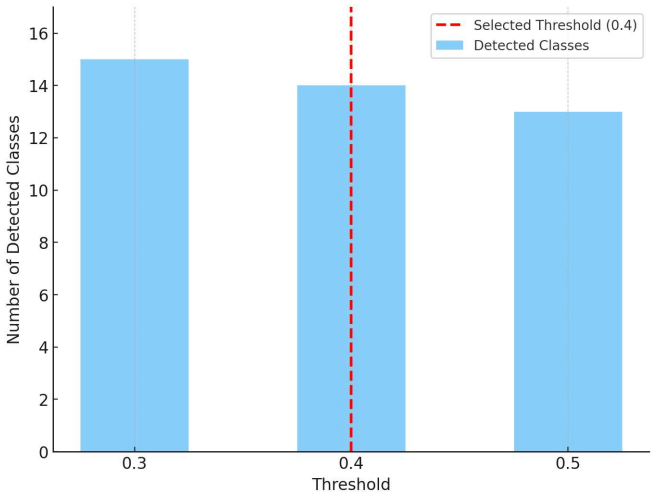

Fig. 11. Number of Detected Trojan Classes at Different Thresholds.

2) Defense for trojan attack: Table III provides the results of how the trojan infection probability varies across different classes and its association with the trojan infection. We have considered complementary performance metrics including the false positive rate (FPR) and FR in our discussion to offer a more holistic view of detection performance. The FPR quantifies how often a trojan infected model is incorrectly flagged as clean, and the FR highlights the degree to which an attacker's trojan remains undetected. We set a detection threshold for the trojan infection probability at 0.4, meaning that any model exceeding this probability is classified as trojan-infected. This threshold choice is selected by empirical observations showing that a clean model typically has a trojan infection probability of around 25%, leaving a 15% margin to account for variability.

If the trojan infection probability is equal to or above 0.9, it correlates with lower FR values, indicating that the attack is more likely to be detected by our defense. For example, classes 4, 6, and 11 exhibit the highest trojan infection probabilities—0.9372, 0.93736, and 0.94076, respectively—and also show confirmed trojan infections and lower FR. While the defense method effectively detects most trojan-infected classes, it shows reduced effectiveness for classes 2, 8, and 9, where the mechanism fails to identify the compromise. These undetected infections contribute to the overall FPR and highlight areas for future refinement. Despite these cases, the defense remains generally robust, detecting trojan infections in 14 out of 17 classes and demonstrating a strong balance between detection sensitivity and minimizing the misclassification of infected models. To assess how varying the detection threshold affects the performance of our defense, we analyzed how different thresholds influence the number of detected trojan-infected classes. As illustrated in Fig. 11, lowering the threshold to 0.3 increases the detection rate, capturing 15 out of 17 infected classes. This reduction in the threshold also leads to a lower FPR, as fewer infected models are incorrectly classified as clean. Conversely, increasing the threshold to 0.5 reduces the detection rate to 13 classes, which increases the FPR because more infected models evade detection and are mistakenly labeled as clean. Ultimately, selecting a threshold of 0.4 strikes

an effective balance, successfully detecting 14 out of the 17 infected classes while keeping the FPR at an acceptable level. When considering the FR, the relationship aligns with FPR trends. As the threshold increases, more trojan-infected models go undetected, allowing the trojan to remain effective and increasing the FR. Lowering the threshold, on the other hand, reduces these rates by ensuring more infected models are correctly identified.

VII. Conclusion

In this paper, we propose a novel algorithm for conducting highly effective trojan attacks on DL models that operate with PQD data. Our proposed algorithm, titled *Sneaky Spectral Strike* (S^3), emphasizes the significant vulnerability of DL models to trojan attacks. This vulnerability is clearly demonstrated using a ResNet50 model, which initially achieved an accuracy rate of 99.22% on clean, untainted data. However, after the application of the trojan attack, the model exhibited a dramatically increased fooling rate of 99.87%. The efficacy of our algorithm is further validated by its ability to produce high fooling rates across various models, reaching a maximum of 99.99% for a CNN, thereby showcasing its effectiveness and adaptability across different architectures. In a comprehensive evaluation against three state-of-the-art algorithms, our algorithm proved to be the most proficient, achieving the highest fooling rate of 99.90% on a more complex dataset that included larger sample sizes and extended time steps. These exceptionally high fooling rates highlight the significant risks that trojan attacks pose within power systems, where even minor misclassification rates could potentially lead to catastrophic outcomes. Furthermore, our study reveals that the triggers generated by our attack are exceptionally subtle, making them difficult to detect. To counteract these threats, our proposed defense method has demonstrated high effectiveness, successfully detecting trojan-infected models across 14 out of 17 class scenarios. However, our algorithm still struggled to detect trojans in the remaining 3 classes, likely subtle trigger patterns. Looking ahead, future research should focus on developing robust defense mechanisms capable of accurately identifying the trojan model for all classes. Expanding the algorithm's adaptability to various real-world scenarios beyond power systems, particularly in other time-series data applications, remains a key research direction. The practical implications of this work are significant for power system security, where trojan attacks pose a serious threat to infrastructure stability and operational reliability. Recent advancements, such as the work [43] demonstrate the growing importance of machine learning in improving detection rates while minimizing false positives. Building on such developments, our future research aims to further enhance detection robustness and ensure broader applicability in diverse critical systems.

References

[1] A. Faizan, "Difference between traditional power grid and smart grid," *Electrical Academia*, 2017.

[2] T. Khan, M. Waseem, H. A. Muqeet, M. M. Hussain, M. Yu, and A. Annuk, "3E analyses of battery-assisted photovoltaic-fuel cell energy system: Step towards green community," *Energy Reports*, vol. 8, pp. 184–191, 2022.

[3] A. Ali, R. Shakoor, A. Raheem, H. A. u. Muqeet, Q. Awais, A. A. Khan, and M. Jamil, "Latest energy storage trends in multi-energy standalone electric vehicle charging stations: A comprehensive study," *Energies*, vol. 15, no. 13, 2022.

[4] G. Dileep, "A survey on smart grid technologies and applications," *Renewable energy*, vol. 146, pp. 2589–2625, 2020.

[5] D. Bian, M. Kuzlu, M. Pipattanasomporn, and S. Rahman, "Analysis of communication schemes for advanced metering infrastructure (AMI)," in *2014 IEEE PES General Meeting— Conference & Exposition*, pp. 1–5, IEEE, 2014.

[6] M. Shahab, S. Wang, and H. Abd ul Muqeet, "Advanced optimal design of the IoT based university campus microgrid considering environmental concerns and demand response," in *2021 6th International Conference on Power and Renewable Energy (ICPRE)*, pp. 798–802, 2021.

[7] I. Worighi, A. Maach, A. Hafid, O. Hegazy, and J. Van Mierlo, "Integrating renewable energy in smart grid system: Architecture, virtualization and analysis," *Sustainable Energy, Grids and Networks*, vol. 18, p. 100226, 2019.

[8] Q. Hassan, A. A. Khadom, S. Algburi, A. K. Al-Jiboory, A. Z. Sameen, M. A. Alkhafaji, H. A. Mahmoud, E. M. Awwad, H. B. Mahood, H. A. Kazem, *et al.*, "Implications of a smart grid-integrated renewable distributed generation capacity expansion strategy: The case of iraq," *Renewable Energy*, vol. 221, p. 119753, 2024.

[9] J. C. Smith, G. Hensley, and L. Ray, "Ieee recommended practice for monitoring electric power quality," *IEEE std*, pp. 1159–1995, 1995.

[10] S. Shah, A. Hellany, M. Nagrial, and J. Rizk, "Impact of wind speed on total harmonic distortion in a hybrid renewable energy system," in *2022 IEEE International Conference on Power Systems Technology (POWERCON)*, pp. 1–5, 2022.

[11] Ł. Michalec, M. Jasiński, T. Sikorski, Z. Leonowicz, Ł. Jasiński, and V. Suresh, "Impact of harmonic currents of nonlinear loads on power quality of a low voltage network–review and case study," *Energies*, vol. 14, no. 12, p. 3665, 2021.

[12] C.-C. Sun, A. Hahn, and C.-C. Liu, "Cyber security of a power grid: State-of-the-art," *International Journal of Electrical Power & Energy Systems*, vol. 99, pp. 45–56, 2018.

[13] R. A. D. Oliveira and M. H. Bollen, "Deep learning for power quality," *Electric Power Systems Research*, vol. 214, p. 108887, 2023.

[14] Y. Wang, K. Chen, Y.-a. Tan, S. Huang, W. Ma, and Y. Li, "Stealthy and flexible trojan in deep learning framework," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 3, pp. 1789–1798, 2023.

[15] Rahul, M. Gangadharappa, and A. Kharola, "Power signal disturbances analysis based on transfer learning in deep architecture," in *2022 IEEE International Conference on Current Development in Engineering and Technology (CCET)*, pp. 1–5, 2022.

[16] J. Tian, B. Wang, Z. Wang, K. Cao, J. Li, and M. Ozay, "Joint adversarial example and false data injection attacks for state estimation in power systems," *IEEE Transactions on Cybernetics*, vol. 52, no. 12, pp. 13699–13713, 2021.

[17] R. Huang and Y. Li, "Adversarial attack mitigation strategy for machine learning-based network attack detection model in power system," *IEEE Transactions on Smart Grid*, vol. 14, pp. 2367–2376, 2023.

[18] R. Ning, C. Xin, and H. Wu, "Trojanflow: A neural backdoor attack to deep learning-based network traffic classifiers," in *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications*, pp. 1429–1438, 2022.

[19] Y. Jiang, X. Ma, S. M. Erfani, and J. Bailey, "Backdoor attacks on time series: A generative approach," in *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pp. 392–403, 2023.

[20] D. Ding, M. Zhang, Y. Huang, X. Pan, F. Feng, E. Jiang, and M. Yang, "Towards backdoor attack on deep learning based time series classification," in *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pp. 1274–1287, 2022.

[21] C. Cui, Y. Duan, H. Hu, L. Wang, and Q. Liu, "Detection and classification of multiple power quality disturbances using stockwell transform and deep learning," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–12, 2022.

[22] F. Ucar, O. F. Alcin, B. Dandil, and F. Ata, "Power quality event detection using a fast extreme learning machine," *Energies*, vol. 11, no. 1, p. 145, 2018.

[23] I. Topaloglu, "Deep learning based a new approach for power quality disturbances classification in power transmission system," *Journal of Electrical Engineering & Technology*, vol. 18, 2023.

[24] E. Yiğit, U. Özkaya, Ş. Öztürk, D. Singh, and H. Gritli, "Automatic detection of power quality disturbance using convolutional neural network structure with gated recurrent unit," *Mobile Information Systems*, vol. 2021, pp. 1–11, 2021.

[25] J. Wang, D. Zhang, and Y. Zhou, "Ensemble deep learning for automated classification of power quality disturbances signals," *Electric Power Systems Research*, vol. 213, p. 108695, 2022.

[26] M. Liu, Y. Chen, Z. Zhang, and S. Deng, "Classification of power quality disturbance using segmented and modified s-transform and dcnn-msvm hybrid model," *IEEE Access*, vol. 11, pp. 890–899, 2023.

[27] J. Tian, B. Wang, J. Li, and C. Konstantinou, "Adversarial attack and defense methods for neural network based state estimation in smart grid," *IET Renewable Power Generation*, vol. 16, no. 16, pp. 3507–3518, 2022.

[28] I. Niazazari and H. Livani, "Attack on grid event cause analysis: An adversarial machine learning approach," in *2020 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, pp. 1–5, IEEE, 2020.

[29] A. Sayghe, J. Zhao, and C. Konstantinou, "Evasion attacks with adversarial deep learning against power system state estimation," in *2020 IEEE Power & Energy Society General Meeting (PESGM)*, pp. 1–5, IEEE, 2020.

[30] E.-N. S. Youssef, F. Labeau, and M. Kassouf, "Adversarial dynamic load-altering cyberattacks against peak shaving using residential electric water heaters," *IEEE Transactions on Smart Grid*, vol. 15, pp. 2073–2088, 2023.

[31] J. Hao and Y. Tao, "Adversarial attacks on deep learning models in smart grids," *Energy Reports*, vol. 8, pp. 123–129, 2022. 2021 6th International Conference on Clean Energy and Power Generation Technology.

[32] G. Zhang and B. Sikdar, "Ensemble and transfer adversarial attack on smart grid demand-response mechanisms," in *2022 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pp. 53–58, 2022.

[33] O. Kosut, L. Jia, R. J. Thomas, and L. Tong, "Malicious data attacks on smart grid state estimation: Attack strategies and countermeasures," in *2010 first IEEE international conference on smart grid communications*, pp. 220–225, IEEE, 2010.

[34] R. Heinrich, C. Scholz, S. Vogt, and M. Lehna, "Targeted adversarial attacks on wind power forecasts," *arXiv preprint arXiv:2303.16633*, 2023.

[35] Y. Cheng, K. Yamashita, and N. Yu, "Adversarial attacks on deep neural network-based power system event classification models," in *2022 IEEE PES Innovative Smart Grid Technologies-Asia (ISGT Asia)*, pp. 66–70, IEEE, 2022.

[36] X. Gong, Y. Chen, W. Yang, H. Huang, and Q. Wang, "B3: Backdoor attacks against black-box machine learning models," *ACM Trans. Priv. Secur.*, vol. 26, aug 2023.

[37] K. Grosse, T. Lee, B. Biggio, Y. Park, M. Backes, and I. Molloy, "Backdoor smoothing: Demystifying backdoor attacks on deep neural networks," *Computers & Security*, vol. 120, p. 102814, 2022.

[38] I. Arshad, M. N. Asghar, Y. Qiao, B. Lee, and Y. Ye, "Pixdoor: A pixel-space backdoor attack on deep learning models," in *2021 29th European Signal Processing Conference (EUSIPCO)*, pp. 681–685, 2021.

[39] Y. Zhang, F. Feng, Z. Liao, Z. Li, and S. Yao, "Universal backdoor attack on deep neural networks for malware detection," *Applied Soft Computing*, vol. 143, p. 110389, 2023.

[40] A. Salem, R. Wen, M. Backes, S. Ma, and Y. Zhang, "Dynamic backdoor attacks against machine learning models," *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*, pp. 703–718, 2022.

[41] H. Kheddar, Y. Himeur, S. Al-Maadeed, A. Amira, and F. Bensaali, "Deep transfer learning for automatic speech recognition: Towards better generalization," *Knowledge-Based Systems*, vol. 277, p. 110851, 2023.

[42] Z. Sun, X. Liu, B. Xu, S. Zhang, C. Fu, G. Yang, J. Li, G. Shao, and C. Zhang, "Equipment failure detection method of substation based on tunnel robot," *Procedia Computer Science*, vol. 166, pp. 305–309, 2020. Proceedings of the 3rd International Conference on Mechatronics and Intelligent Robotics (ICMIR-2020).

[43] N. P. Bhatta and F. Amsaad, "Ml assisted techniques in power side channel analysis for trojan classification," *Knowledge-Based Systems*, vol. 28, 2025.

Sultan Uddin Khan received his B.Sc. degree in electrical and electronic engineering from the Chittagong University of Engineering and Technology, Chittagong, Bangladesh, in 2011. He received his M.Sc degree from the Department of Electrical and Computer Engineering, North Carolina A & T State University, USA. Before pursuing a Master of Science degree, he developed his skills in the field of power systems by working at Dhaka Power Distribution Company Ltd., one of the largest power distribution companies in Bangladesh, as a Power System Protection and Automation Engineer. He became profoundly involved in large-scale projects involving the modernization, protection, and smart grid technologies of power systems. His career spans more than seven years and is distinguished by extensive experience and practical knowledge that bridges the divide between power systems engineering and contemporary automation processes. His research interests include cyber security, smart grid, the application of deep learning in power systems, unmanned aerial vehicles, machine learning, and deep learning for cyber security.

Mahmoud Nabil Dr. Mahmoud received his Bachelor of Science (BS) and Master of Science (MS) degrees, with honors, in computer engineering from Cairo University, Egypt, in 2012 and 2016, respectively. He completed his PhD in electrical and computer engineering from Tennessee Tech University, Cookeville, Tennessee, in August 2019. Currently, he holds the position of associate professor at the Department of Electrical and Computer Engineering, North Carolina A & T State University. Dr. Mahmoud is an accomplished researcher and has authored and coauthored numerous publications in prestigious venues. His research work has been published in renowned journals such as IEEE Internet of Things, IEEE Transactions of Dependable and Secure Computing, IEEE Transactions on Human-Machine Systems, and IEEE Transactions of Mobile Computing. He has also contributed to leading conferences including the International Conference on Communication, International Conference on Pattern Recognition, and International Conference on Wireless Communication. With diverse research interests, Dr. Mahmoud's areas of expertise include security and privacy in unmanned aerial systems, smart grids, machine learning applications, vehicular Ad Hoc networks, and blockchain applications. He has received significant funding for his research projects from esteemed national agencies and organizations including the National Science Foundation (NSF), Department of Transportation (DOT), Air Force Research Laboratory (AFRL), NASA, Intel, Cisco, and Lockheed Martin.

Mohamed M. E. A. Mahmoud received the PhD degree from the University of Waterloo in April 2011. Currently, Dr. Mahmoud is a professor in the Department of Electrical and Computer Engineering at Tennessee Tech University, USA. His research interests include security and privacy-preserving schemes for smart grid, e-health, and intelligent transportation systems. Dr. Mahmoud has received the NSERC-PDF award and won the Best Paper Award at the IEEE International Conference on Communications (ICC'09) in Dresden, Germany, 2009. He is the author of more than 100 papers published in IEEE conferences and journals. Dr. Mahmoud serves as an Associate Editor in the IEEE Internet of Things Journal and the Springer journal of peer-to-peer networking and applications. He has also served as a technical program committee member for several IEEE conferences.

This article has been accepted for publication in IEEE Transactions on Network Science and Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2025.3567340

MAAZEN ALSABAAN received the B.S. degree in electrical engineering from King Saud University (KSU), Saudi Arabia, in 2004, and the M.A.Sc. and Ph.D. degrees in electrical and computer engineering from the University of Waterloo, Canada, in 2007 and 2013, respectively. He is currently an Associate Professor with the Department of Computer Engineering at KSU. From 2015 to 2018, he served as the Chairperson of the Department. He is a consultant for various agencies and has received numerous grants from KSU and King Abdulaziz City for Science and Technology (KACST). His research interests encompass wireless communications and networking, surveillance systems, vehicular networks, green communications, intelligent transportation systems, and cybersecurity.

Tariq Alshawi (Senior Member, IEEE) is a faculty in the Electrical Engineering Department at King Saud University, Riyadh, Saudi Arabia. He has over 15 years of teaching and research experience in multi-dimensional signal processing, computer vision, and machine learning. He was the emerging technology advisor to his excellency the governor of Saudi telecom regulator, a Researcher with the Prince Sultan Advanced Technologies Research Institute, and the Center of Excellency in Optics & Radio at King Saud University. He received his M.S. from the University of Michigan, Ann Arbor, and his Ph.D. from the Georgia Institute of Technology.