Graph Autoencoder-Based Power Attacks Detection for Resilient Electrified Transportation Systems

Shahriar Rahman Fahim, *Member, IEEE*, Rachad Atat, *Senior Member, IEEE*, Cihat Kececi, Abdulrahman Takiddin, *Member, IEEE*, Muhammad Ismail, *Senior Member, IEEE*, Katherine R. Davis, *Senior Member, IEEE*, and Erchin Serpedin, *Fellow, IEEE*

***Abstract*—The interdependence of power and electrified transportation systems introduces new challenges to the reliability and resilience of charging infrastructure. With the increasing prevalence of electric vehicles (EVs), power system attacks that can lower customers charging satisfaction rates are on the rise. The existing false data injection attacks (FDIAs) detection strategies are not suitable for protecting the power-dependent transportation infrastructure since (a) these detectors are primarily optimized for power grids alone, and (b) they overlook the impact of attacks on the quality-of-service of EVs and charging stations (CSs). In response to these challenges, this paper aims to develop an FDIA detection strategy that takes advantage of the data correlations between power and transportation systems, ultimately enhancing the charging satisfaction rate. To achieve this goal, we propose a graph autoencoder-based FDIA detection scheme capable of extracting spatio-temporal features from both power and transportation data. The input features of power systems are active and reactive power while those for transportation systems are the hourly traffic volume in CSs. The proposed model undergoes comprehensive training and testing on various types of FDIAs, showcasing improved generalization abilities. Simulations are conducted on the 2,000-bus power grid of the state of Texas, featuring 360 active CSs. Our investigations reveal an average detection rate of 98.3%, representing a substantial improvement of 15-25% compared to state-of-the-art detectors. This underscores the effectiveness of our proposed approach in addressing the unique challenges posed by power-dependent electrified transportation systems.**

***Index Terms*—Cybersecurity, smart grids, electric vehicles, false data injection attacks, graph autoencoder, and graph neural networks.**

NOMENCLATURE

Acronyms

ACC	Overall accuracy.
AHP	Analytical hierarchical process
CS	Charging station
DNN	Deep neural network
DR	Detection rate.
ERCOT	Electric reliability council of texas
FAR	False alarm rate.
FDIA	False data injection attack
GAE	Graph autoencoder
GCNN	Graph convolutional neural network
GNN	Graph neural network
LSTM	Long short-term memory
ReLU	Rectified linear unit.

Parameters

$\mathcal{D}$	Denial of service rate.
$\mathcal{S}$	EV power satisfaction rate.
ψ_θ	Convolutional filter.
σ	Trainable parameters.
$C(\tilde{z}, \sigma)$	Cross entropy function.
E_E	Kullback–Leibler divergence.
O^k	Output of each hidden layer.
P_j	Chebyshev polynomial.
$T_{(v_i,t)}$	Historical time series data for each node of graph v_i at time step t.
V_P, E_P	Set of nodes and set of vertices.
X, X^*	Original and regenerated time series data.
$z, \tilde{z}$	True and predicted levels.
$\mathbf{W}$	Adjacency matrix
G	Constructed graph

Variables

ΔP_i^t	Maliciously added power on bus i at time stamp t.
γ_{ω_j}	j^{th} Chebyshev coefficient.
$\gamma_{\max}$	Maximum eigenvalue of matrix L.
θ^*	Optimal parameter configuration.
$C_{l_E}^t, H_{l_E}^t$	Cell state and hidden state of an LSTM cell.
E_{DP}	Reconstruction error.
$i_{l_E}^t, o_{l_E}^t, f_{l_E}^t$	Input, output and forget gate of an LSTM cell.
L	Graph laplacian matrix.
n_{ev}	Number of EV in a CS.
n_{fev}	False number of EV in a CS.
$P(X)$	Probability distribution of X.
$P_{\text{true},i}^t, P_{\text{false},i}^t$	True and false power on bus i at time stamp t respectively.
P_{cs}	Available power on a CS.
P_{EV}	Power of each EV.
U	Eigenvector matrix.

I. INTRODUCTION AND MOTIVATION

IN response to the pressing demand for environmental protection, nations globally are dedicated to crafting clean energy solutions aimed at minimizing carbon emissions. Notably, electrified transportation systems have garnered significant global attention in this context. This is largely due to their capacity to diminish carbon footprints, simultaneously enhancing environmental sustainability and bolstering energy

S. R. Fahim, C. Kececi, K. R. Davis, and E. Serpedin are with the Electrical & Computer Engineering Department, Texas A&M University, College Station, TX 77843, USA (e-mail: {sr-fahim, kececi, katedavis, eserpedin}@tamu.edu).

R. Atat is with the Electrical & Computer Engineering Department, Texas A&M University at Qatar, Doha, Qatar (email: rachad.atat@qatar.tamu.edu).

A. Takiddin is with the Department of Electrical and Computer Engineering, FAMU-FSU College of Engineering, Florida State University, Tallahassee, FL 32310, USA (e-mail: a.takiddin@fsu.edu).

M. Ismail is with the Department of Computer Science, Tennessee Tech University, Cookeville, TN 38505, USA (e-mail: mismail@tntech.edu).

This work is supported by NSF EPCN Awards 2220346 and 2220347.

security [1], [2]. Electric vehicles (EVs) have become a pivotal means of attaining environmentally conscious and sustainable transportation. The dependence between the power grid and the transportation infrastructure increases as the usage of EVs keeps growing [3]. Meanwhile, modern power systems, being cyber-physical in nature, rely on a huge amount of measurement data for decision-making and situational awareness. Therefore, safeguarding the authenticity of the collected data is a critical necessity in ensuring the stability and dependability of the coupled system. Unfortunately, the data dependency makes power systems more vulnerable to false data injection attacks (FDIAs), where the integrity of the data can be tampered by malicious entities [4], [5]. Such actions falsify operational decisions, which can in turn degrade the EVs charging satisfaction rate.

Modern electricity and transportation networks represent a paradigm shift in the realm of urban infrastructures. These two intricate networks, deployed on a vast scale, are interdependent and difficult to control and secure due to multiple factors that may influence their operation. In particular, charging stations (CSs) draw power from a generator substation to enable efficient and convenient charging of EVs. The satisfaction rate of EV charging is directly shaped by the availability of power at CSs. Sufficient charging power translates to enhanced services with faster charging rates, resulting in heightened customer satisfaction. Conversely, inadequate power can lead to CS congestion, blocking charging requests, thereby extending charging times and dissatisfying users. In the context of FDIAs, attackers can falsify the power measurements by making them appear as high (additive attacks), low (deductive attacks), or a combination of both (camouflage attacks), all negatively influencing the EVs charging satisfaction rate. In cases of falsified high power scenarios, attackers manipulate power measurements to create a misleading perception of abundant charging capacity. Hence, the charging demand increases at the target connected CSs, leading to congestion, rise in the number of blocked charging requests, and a significant drop in charging efficiency (longer charging periods). Similarly, in a falsified low power scenario or in the presence of a deductive attack, attackers manipulate power measurements to portray insufficient available power. In this case, EVs generally face the challenge of finding available CSs. In both cases, falsified data generate a feeling of uncertainty in customers and consequently a significant drop in charging requests satisfaction rates. Camouflage attacks normally lead to fluctuations in the available CSs charging power, and thus, are deemed harder to detect. To address the challenges posed by these cyber attacks, this paper proposes a novel FDIAs detector fit for interconnected power-transportation systems and assesses the effects of such attacks on EVs charging satisfaction rate.

A. Related Work

In recent years, the field of cyber attack detection has witnessed a dynamic shift from traditional model-based approaches to more agile and adaptable data-driven approaches. This transition has been driven by the increasing complexity of modern power systems. The majority of existing works used artificial intelligence-based approaches to detect FDIAs [6]. At present, the proposed detection schemes rely on two main approaches: i) machine learning (ML)-based models employing either shallow or deep neural network (DNN) architectures, and ii) graph neural networks (GNNs)-based models utilizing graph signal processing filters. Despite the favorable detection performance achieved by these approaches, they are subject to limitations, as will be elaborated upon shortly.

1) Traditional Model-based Approaches: Model-based anomaly detection approaches assume that the behavior of a system can be accurately represented and predicted through precise modeling. Within the context of power systems, the model-based techniques relied on static equations and optimization frameworks to detect anomalies and intrusions. Under such a framework, state estimation-based anomaly detection schemes were proposed in [7], [8]. These approaches look for differences between the estimated states and the actual measurements by using mathematical models of the power system. In reference [9], an extended Kalman filter interval state estimation technique was proposed to avoid erroneous measurements. Moreover, a decentralized model-based approach was proposed in [10] that uses a maximum likelihood estimation approach.

Nevertheless, employing model-driven approaches in this domain presents numerous challenges, particularly when dealing with coupled power and transportation systems. In these interconnected systems, various components and variables interact, forming an intricate network of relationships and dependencies. The complexity arises from the diverse behaviors that these interconnections can represent, making it difficult to formulate a comprehensive mathematical model that accurately captures all the dynamics of such systems. Furthermore, model-driven methods usually hinge on developing precise mathematical equations and models to describe the system's behavior [11].

Efforts to encapsulate the intricate interactions within a unified optimization framework using equations are generally highly challenging and frequently impractical. Therefore, a more viable and efficient alternative is to leverage data-driven deep-learning neural networks. The primary advantage of data-driven deep learning methods lies in their ability to model complex interactions within systems without requiring explicit knowledge of all the underlying parameters.

2) ML- and DNN-based Detection Schemes: Various degrees of success have been reported by the ML-based FDIA detection schemes. For instance, [12] proposed a support vector machine (SVM)-based stealthy FDIA detection scheme that reached an F1-score of 82%. In [13], the FDIA detection task is formulated as a multi-label classification problem and a decision tree is employed to detect the location of attacks. A random forest-based algorithm with 93% detection rate was proposed in [14]. The main limitation of these works is that they fail to achieve a good generalization performance as they lack interpretability due to i) not optimizing the hyperparameters and ii) not extracting the underlying complex feature data of power systems [15].

Due to the inherent complex feature extraction capability, deep learning (DL)-based techniques have gained widespread

prominence in detecting cyber-physical attacks in power systems. Following this popularity, feed-forward neural network (FNN)-based implementations reported more than 90% accuracy [16], [17]. In [18], the Kalman filter is combined with a recurrent neural network (RNN) to reach 96% detection accuracy. A detection performance of 96.2% was reported by an autoencoder model integrated with a generative adversarial network (GAN) in [15]. Aiming for a reinforcing solution, [19] proposed a denoising variational autoencoder to detect faults in power systems. Reference [20] showed that the deep belief network (DBN) framework outperforms the residual-based detector and the extreme learning machine (ELM)-based detector [21]. In addition, convolutional neural network (CNN)-based solutions attained 93% [22] and 99% [23] detection accuracy. In [23], the CNN is combined with Kalman's filter to process temporal and spatial data correlations. Although the aforementioned works achieve high detection rates, they fail to extract the spatial relationships present in the measurement data. This is because these approaches ignore the topological characteristics of power grids [24], [25].

B. Advancing the Adoption of GAE

A fundamental requirement in the realm of interconnected power systems is the requirement to efficiently address the power flow challenges. Modeling the power flow necessitates estimating two parameters from a set that includes active power, reactive power, node voltage, and node angle, based on the values of other two given parameters. For large interconnected power systems, this represents a challenging task due to the presence of a non-linear set of $2\cdot(n-1)$ equations [26]. This system may be decomposed into two sets of equations, one set of equations for real and another for reactive powers, respectively. Regarding the load buses, the considered power system may contain thousands of nodes.

The inherent graph nature of the interconnected power systems has prompted researchers to explore graph theory. Specifically, distribution systems can be modeled as graphs, with a bus associated to a node and a power line represented as an edge or branch [27], [28]. The graph representation proves useful for modeling and analyzing complex topologies of power systems. Moreover, graph representation of power systems can capture the spatial dependencies of different nodes (e.g., buses) in the power grid. They are also capable of handling temporal dependencies, which are crucial for analyzing the dynamic behavior of power systems. Motivated by these considerations, researchers have employed graph neural networks to address challenges in various aspects of power systems, including power system state estimation [29], fault analysis [30], load prediction [31], and cybersecurity [32]. The Graph Autoencoders (GAEs) offer a powerful mechanism for feature extraction from graph-structured data. GAEs represent complex graph structures in lower-dimensional representations while retaining essential structural information [33]. This latent space representation extracts abstract features that are often challenging to discern in the original data. These features represent critical information about the graph's topology and connectivity, allowing for more effective downstream analysis. In summary, GAEs offer a powerful framework for modeling and analyzing large interconnected power systems, making them essential tools in the field of electrical engineering and power grid management. In summary, GAEs offer a powerful framework for modeling and analyzing large interconnected power systems, establishing themselves as essential tools in the field of electrical engineering and power grid management.

1) Graph-Centric Detection Approaches: Graph-based FDIA detection schemes present a great capability to overcome the intrinsic limitations of traditional DL-based algorithms. The most obvious benefit of utilizing graph-based techniques is their capability of capturing spatial relationships and topological characteristics from the graph-structured data of power systems [34]. Currently, graph-based signal processing approaches have been adopted with success in several power system applications such as power flow analysis [35], fault detection and localization [36], time-series prediction [31] and FDIA detection [37]. The GNN-based detection schemes for FDIAs employ graph signal processing operations, which enable adaptive aggregation and transmission of the information throughout the graph. The superiority of GNN-based detectors was probed by [24], which showed an improvement of the F1 score by 4%. Detection of unobservable attacks was handled via an auto-regressive moving average (ARIMA) graph filter in [38], which allowed the detector to adapt better against sharp changes in the spectral domain. A modified temporal multi-graph convolutional network was proposed in [39] by fusing the training stages of graph convolutions and multi-layer perceptions to represent the node attributes simultaneously. The proposed model achieved 96% accuracy against multiple power system topologies. Another work [15] adopted a hybrid approach where the graph convolution methodology was equipped with the long short-term memory (LSTM) unit to attain 96% accuracy. The graph autoencoder (GAE) neural network strategy was proposed by [40] to detect attacks in the presence of unseen topologies. An ensemble detector based on GAE showed 12% performance improvement relative to shallow detectors [41]. Reference [42] compared the performances of simple autoencoder (SAE), variational autoencoder (VAE) and autoencoder equipped with attention mechanism (AAM) in terms of detecting FDIAs. The outcomes revealed that AAM improved the network's resilience to cyber attacks and exhibited enhanced performance. Despite the advantages offered by the aforementioned GNN-based detectors, they are trained and tested only on power systems, and therefore, we label them as single graph detectors. In practice, the power system is coupled and interacts with other systems [43]. For instance, in the coupled power-transportation system, the FDIAs on the power grid may influence the satisfaction rate of EVs. Therefore, investigating the effect of FDIAs by considering the tight couplings and dependencies between the power substations and CSs is required.

2) Thwarting Attacks in Dependent Systems: The dependence of the electrified transportation infrastructure on power systems is vital for smart cities management. Yet, only a few studies investigated the impact of malicious attacks on such dependent systems. For instance, in [44], a case study was performed on the coupled transportation-energy system to in-

vestigate the effect of FDIAs with the aim of enhancing cyber resiliency in smart city management. Another work adopted a static Bayesian game theory-based approach to resist attacks on coupled systems [1]. In-vehicle bus security issues were studied in [45]. FDIA detectors in vehicular communication networks were addressed in [46] to avoid traffic congestion and to increase user satisfaction rates.

It is worth mentioning that existing literature does not address the effect of FDIAs on EVs charging satisfaction rates. Moreover, the existing approaches reported dissimilar performance metrics for different FDIAs and system types, which imposes extra challenges when comparing them. This paper fills up this gap by investigating the impact of power system FDIAs on electrified transportation systems, which at its turn directly impacts the EVs market growth.

C. Contributions and Organization

This paper contributions are next summarized.

- First, we address the limitations of current FDIA detectors by proposing a GAE-based detection strategy for the power grid-dependent electrified transportation network. The proposed detector captures the topological relationships present in the data of both systems through its Chebyshev graph convolutional layers.
- Second, we show that the proposed model detects FDIAs in the presence of unseen topological configurations, i.e., the testing dataset is not part of the training dataset, unlike existing works. Thus, the proposed approach portrays practical real-world scenarios.
- Third, to demonstrate the effectiveness of our detector, we conduct extensive simulations against different types of attacks on the power system, mainly additive attacks, deductive attacks, and camouflage attacks for the following cases: i) when the attackers have no knowledge of the system and they randomly hack power substations, and ii) when the attackers have full knowledge of the system and hack the most vulnerable power substations. The latter case helps to develop robust defense strategies by identifying weaknesses in the network.
- Fourth, we study the impact of the aforementioned attacks on EVs satisfaction rate for the 2,000-bus power system of the state of Texas with 360 allocated CSs.

The remainder of the paper is organized as follows. Section II presents the model of the power-transportation system. Section III describes the threat model, attack strategies, and generation of benign and malicious samples. Section IV presents the architecture of the proposed GAE detector. Section V introduces the benchmark detectors, hyperparameters selection, and evaluation metrics. Section VI describes the simulation results, and Section VII concludes the paper.

II. Power-Transportation System Model

This section describes how the power system is modeled and coupled with the transportation system.

A. Modeling of Power System

Since the inherent structure of a power grid can naturally be represented as a graph, GNN-based strategies can be employed to develop efficient FDIA detection strategies. However, the asymmetric adjacency matrix of a directed graph restricts the free flow of information especially at the peripheral regions of the power grid, thereby constraining the GNN's learning efficacy [47]. Thus, we model power systems as weighted undirected connected graphs [24], [38], [48]. Fig. 1 illustrates the graph representation of the power grid of Texas along with the directions of power flow at each bus. The undirected graph is represented as $G = (V_{\mathrm{P}}, E_{\mathrm{P}}, \mathbf{W})$, where $V_{\mathrm{P}} = \{1, 2, .., B\}$, E_{P}, and $\mathbf{W} \in \mathbb{R}^{B \times B}$ model the set of vertices or buses (B denotes the total number of buses), power lines (connecting edges), and adjacency (line admittance) matrix, respectively. The weight of the edge between the interconnected buses i and j is $\mathbf{W}_{ij}$. Each grid node presents unique power and voltage characteristics. The power lines present unique active/reactive power flows determined by the corresponding line impedance. Learning the specific patterns in the power system

Fig. 1: Graph model of the 2,000-bus power grid of Texas with red nodes indicating the buses connected to CSs.

helps to distinguish between normal operating conditions and attack conditions [24]. GNNs present spatial-temporal learning abilities that facilitate processing of spatial topological (node connectivity and their layout) information and temporal (power flows) features and identification whether the system is in normal or abnormal (attack) state. The detection performance of GNN-based models depends on how well they are trained to the different attacks. Thus, the GNN-based models should be trained on a comprehensive set of different attacks, ensuring the detection remains versatile and robust.

B. Coupling of Power and Transportation Systems

The transportation system under consideration consists of 360 charging stations, each with a unique geographical location that is precisely identified using specific geographic coordinates. To establish a meaningful and efficient coupling between the power grid and transportation network, we considered the geographical coordinates of CSs. The coupling between the two networks is established by employing the

haversine distance formula that calculates the shortest distance between two points of the Earth while considering the Earth curvature into account. By applying the haversine formula, the shortest route from each CS to the nearest power substation is inferred. The approach of minimizing haversine distances aids in the selection of the optimal path, ensuring a systematic and meaningful integration of the two systems. In addition, the proposed geolocation aware methodology represents a valuable strategy for modeling complex infrastructures, as it considers the geographical aspects of both power and transportation networks. As a result, it not only facilitates modeling the coupling of these systems but also it contributes to a more informed and efficient methodology to account for the infrastructure expansion and development. An illustration of the coupled power and transportation system is presented in Fig. 2.

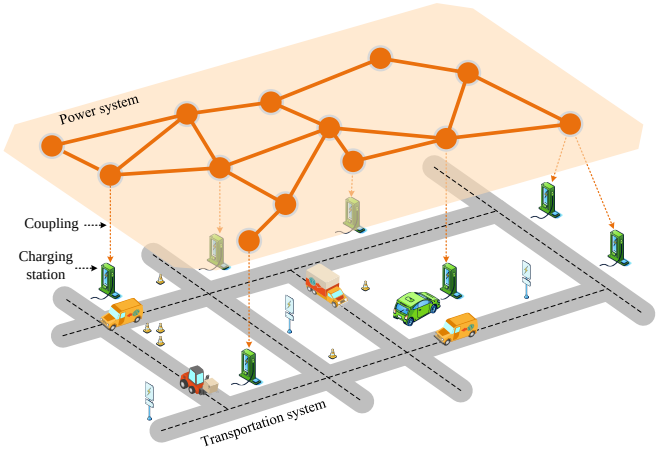

Fig. 2: Illustration of an interconnected power-transportation system.

III. Data Generation

A. Threat Modeling

In an FDIA threat model, attackers stealthily manipulate power measurement data to falsify system states, while ensuring that they bypass the traditional bad data detectors employed by power systems. In this paper, we consider three different types of attacks: 1) additive attacks, 2) deductive attacks, and c) a combination of both additive and deductive, a.k.a. camouflage attacks.

If P_i^t represents the power measurement at bus i at timestamp t, then under normal conditions, the true power reading $P^t_{\text{true},i}$ should match the measured power at the controller (i.e., $P^t_{\text{true},i} = P^t_{\text{measured},i}$. However, tampered measurements might convey falsified data values as will be described next.

1) Additive attacks: In this attack scenario, the attacker reports the current data such that

$$P^t_{\text{false},i} = P^t_{\text{true},i} + \Delta P_i^t, \tag{1}$$

where ΔP_i^t indicates the maliciously added data by the attacker. An adversary may gain access to the communication network and report incorrect values at some buses. The manipulation is performed to make the readings appear within the permitted range, while in reality, the true measurements might be beneath this threshold. Additive attacks can lead to situations where the system believes there's more power at some buses than there actually is. This can lead to unnecessary and potentially harmful corrective actions.

2) Deductive attacks: The mechanism of a deductive power attack involves subtracting a certain value from the original measurement:

$$P^t_{\text{false},i} = P^t_{\text{true},i} - \Delta P_i^t. \tag{2}$$

In this scenario, an attacker may gain access to the data communication channel and under-represent power values at certain buses. Deductive attacks can mislead systems into believing there is less power than there actually is. This can result in under-utilization or missed opportunities for distributing power efficiently.

3) Camouflage attack: The camouflage or combined attack is a sophisticated blend of both additive and deductive attacks. The camouflage attack is modeled through this equation:

$$P_i^t = P^t_{\text{true},i} + b \cdot \Delta P_i^t - (1-b) \cdot \Delta P_i^t, \tag{3}$$

where b is a binary variable which is 1 for the additive attack and 0 for the deductive attack. For example, half of the attacked measurements can be additive, while the other half can be deductive. By adopting a combination of both attacks, attackers aim to create a more complex attack pattern that is harder to detect and in turn maximize the potential damage. Such simultaneous upsurge and downswing in measured values can lead to conflicting operational decisions, potentially causing system malfunctions and inefficiencies.

Fig. 3 and Fig. 4 illustrates the active and reactive power during normal and malicious conditions respectively. In Fig. 3 and Fig. 4, the attack occurs at $t = 2, 3, 4, 5, 6, 8$ time stamps.

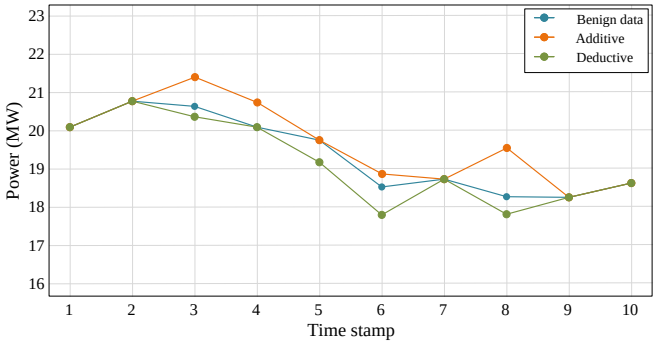

Fig. 3: Active power of benign and malicious samples.

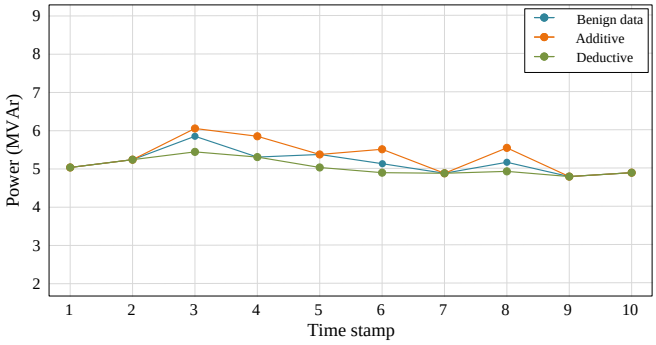

Fig. 4: Reactive power of benign and malicious samples.

B. Attack Strategies

1) Random Bus Attacks: Random bus attacks are defined as non-discriminative random selection of buses as targets. They reduce to random selection of r buses from B ($r \leq B$), i.e., selection of a subset out of $B!/\left(r!(B-r)!\right)$ alternatives. Such attacks can degrade the network's reliability and introduce service interruptions, especially if the attacked buses are not restored quickly.

2) Most Vulnerable Target Buses: Vulnerability refers to the capacity of a bus to serve as a potential point of failure; an attack on such a vulnerable point could result in substantial damage to the overall system. In this section, we assess the vulnerability of each bus in the coupled system. The aim of this assessment is to allocate vulnerability scores to buses, which will later be utilized to design the most vulnerable bus attack strategy.

The vulnerability of buses not only depends on the topological characteristics but also on the power flow dynamics. Thus, we consider a detailed set of metrics, combining topological and electrical metrics. The topological metrics include i) the CSs neighborhood density, which measures the importance of a bus based on the density of CSs in the neighborhood area, ii) connectivity impact, which refers to the number of buses that remain connected after a failure [49], iii) connectivity loss, which captures the average decrease in the number of generators after a failure event [50], iv) betweenness centrality, which indicates the extent to which a bus lies on the shortest paths connecting other pairs of buses [51], v) clustering coefficient, which models the extent to which buses tend to cluster together [52], and vi) degree centrality, which represents the number of buses and lines that directly influence a bus [53]. As for the electrical metrics, they include i) load shedding, which measures the total apparent power after a bus failure [49], ii) effective graph resistance, which captures the total cost to transfer power between a pair of buses [54], iii) electrical degree centrality, which indicates the number of power flows that directly influence the status of a bus, and iv) electrical betweenness centrality, which indicates the extent to which a bus lies on the path connecting other pairs of buses under the assumption that power flows over the shortest paths between them [55].

The weight factor of each of these metrics is obtained by employing the analytical hierarchical process (AHP) methodology [56], where pairwise comparisons are performed in order to determine the relative importance of each metric with respect to the other. Once we obtain the corresponding weights of the topological and electrical metrics, we compute the topological vulnerability score as the weighted summation of topological metrics. Similarly, we compute the electrical vulnerability score as the weighted summation of electrical metrics. Finally, we run AHP analysis again to obtain the corresponding weights of the calculated topological and electrical vulnerability scores in order to compute the final overall vulnerability score.

C. Generating Benign and Malicious Samples

To generate the normal time-series active and reactive power values, we conduct power flow analysis using Newton's method in MATLAB MATPOWER toolbox [57]. This toolbox facilitates the calculation of system voltages, currents, and of real and reactive power flows. In this regard, a scalar vector, F, is first created by normalizing the load data from the Electric Reliability Council of Texas (ERCOT) [58]. The scaling factor is then applied to the active and reactive power values from the preceding timestamp using a normal distribution with mean and standard deviation $1 + 0.025F$ and 0.01, respectively. The dynamic range of charging load values is increased by this operation, which generates dynamic changes in the time-series data. This facilitates generation of the time-series power datasets. Regarding the transportation system, the dataset records the hourly traffic volume at each charging station to monitor the influx of EVs. The charging power of each EV is used to determine the hourly demand for EV charging at a particular CS. Moreover, in addition to the regular load, the power request at the bus level represents the total power demands of all CSs linked to that specific bus.

Using the above-mentioned approach, extensive spatio-temporal datasets can be generated for power systems operating in normal (standard) and under-attack conditions. These datasets play an important role in developing effective detection mechanisms for coping with different attacks. For the considered power system, 96 snapshots of power dynamics per day are recorded, equating to data points collected at every hour. Over a span of 6 months, this sums up to 17280 recorded timestamps. By simulating the power flows across the power system, benign or normal data are generated. Subsequently, bad data are injected following the attack strategy described earlier in this section. The generated data are then used for training and testing the model.

The datasets encompass crucial input features and output labels, ensuring the model's ability to effectively differentiate between normal and anomalous behavior. The input features encompass node features, capturing information related to active and reactive power values of buses as well as edge features that model the power flow dynamics between these nodes. These features provide a comprehensive representation of the power system's operational characteristics. The output labels serve as binary indicators, enabling the model to classify each data sample as either "normal" or "anomalous". The number of data samples within the datasets is influenced by the time series data, typically consisting of 17280 timestamps over a 6 month period for each scenario (normal and under attack, respectively). The generated structured dataset facilitates the model's training and assessment, leveraging node and edge attributes to enhance the security of the interconnected power system against potential cyber threats.

IV. GAE-Based FDIA Detection Model

This section introduces the FDIA detector, which will be shown to be robust against the random attacks as well as attacks targeting the most vulnerable buses. The proposed FDIA detection scheme leverages an autoencoder architecture

that utilizes an unsupervised training dataset, relying solely on benign data samples, X_b, for training [59]. In addition, it employs a generalized training strategy that enables the model to detect unseen FDIAs. The proposed GAE model incorporates Chebyshev graph convolutional recurrent layers, which facilitate the extraction of complex spatio-temporal patterns from the power measurements [60]. This in turn allows the model to strengthen its resilience against FDIAs. We show in the experimental results section (Section VI) the potential of the proposed model in identifying FDIAs with high detection accuracy and adaptability, making it valuable for ensuring power system security and reliability.

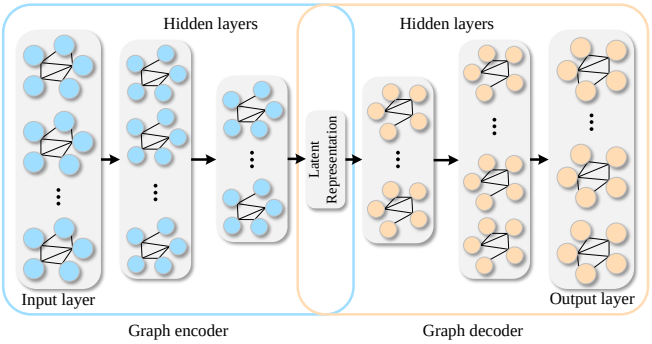

Fig. 5: Architecture of GAE model.

A. Generative Learning Problem Formulation

FDIAs can manifest in different forms, making it challenging to characterize them with precise criteria and patterns for accurate detection. GAE detects FDIAs based on deviations from normal behavior. GAE learns a low-dimensional representation of the power network graph while preserving critical information about the system's topology. In addition, the generative nature of GAEs facilitates data augmentation, which in turn allows to construct synthetic attack samples to address the uncertainty of FDIAs. The architecture of the proposed GAE-based FDIA detector is illustrated in Fig. 5. The proposed model consists of three layers: i) a graph feature extraction layer, ii) an encoder layer, and ii) a decoder layer. The first layer extracts the features from graph-structured data, while the second and third layers enable acquisition of information at graph level about the benign samples observed during normal operation.

For each time step t, each node of graph v_i holds a historical time series record $T(v_i, t)$, which is used as an input to the model. The goal of this model is to obtain a conditional probability distribution $P^*(V^*(t') \mid \pi)$ for the predicted tensor $V^*(t') =< v_1^*(t'), v_2^*(t'), \ldots, v_n^*(t') >$ given the historical tensor $\pi =< T(v_1, t), T(v_2, t), \ldots, T(v_n, t) >$. The autoencoder captures the probability distribution $P(X)$ for X data points over the n-dimensional vector space $\mathcal{X} \subseteq \mathbb{R}^n$. Using this procedure, we can regenerate sample X^* that closely resembles X. As the interdependencies between variables in $\mathcal{X}$ increase, learning the true probability distribution $P(X)$ becomes more challenging. To address this challenge, a "latent variable"-based architecture is adopted in this paper, wherein a hidden random vector, $z \in Z$, incorporates the key characteristics of $P(X)$ (such as anomalous patterns in the data). In fact, the variable z is sampled from a probability distribution $P(z)$ that is not known or specified. To ensure that the model is generative and capable of constructing samples X^*, we confirm the presence of at least one configuration $\hat{z} \in Z$ that forces the model to generate some data samples $\hat{X}$ in $\mathcal{X}$. The model contains deterministic functions $f(z;\theta)$ with parameters $\theta \in \Theta$, and the function $f : Z \times \Theta \to \mathcal{X}$ maps each "latent variable-parameter" pair to $\mathcal{X}$. The objective is to find an optimal parameter configuration $\theta^* \in \Theta$ such that when $z \sim P(z)$ the value of $X^* = f(z; \theta = \theta^*)$ closely corresponds to some $X \in \mathcal{X}$. In other words, the optimization aims to maximize the probability of f by constructing an output X^* that is similar to the original data X. Herein, the generative learning optimization problem is formulated as

$$\theta^* = \arg\max_{\theta} \left[P(X) = \int f(z;\theta) P(z) dz \right]. \tag{4}$$

Upon converging to the optimal solution θ^*, the GAE-based generative model is expected to regenerate X^*. The function $f(z;\theta)$ is represented as a Gaussian distribution $P(X \mid z;\theta) = N\left(X \mid f(z;\theta), \sigma^2 I\right)$, where $f(z;\theta)$ indicates the mean and σ denotes the hyperparameter used to calculate the standard deviation.

B. Spectral Graph Filtering and Graph Encoder

To apply convolutional filters to graph data structures, it is essential to represent the data in a suitable manner. Our objective is to express the data in the spectral domain, facilitating the implementation of graph filtering operations. In pursuit of this objective, we consider the graph Laplacian matrix, denoted as L, which encapsulates the graph's features, accounting for diverse node connections and enabling the computation of spectral graph filters. These spectral filters empower graph convolution to analyze complex graph-structured data with efficacy. With U denoting the eigenvector matrix of the normalized Laplacian $L = U\Omega U^T$, the spectral graph convolution on graph $\mathcal{G}$ is performed via $\psi_\theta * \pi = U\psi_\theta U^T \pi$. The convolutional filter $\psi_\theta = \mathrm{diag}(\theta)$ parameter vector is represented by $\theta \in \mathrm{R}^n$. The graph Laplacian is computed in the Fourier domain as the Fourier transform diagonalizes the graph Laplacian and reveals its eigenvalues and eigenvectors, which are crucial for spectral analysis and filtering. The Fourier transformation of π is expressed as $U^T\pi$. Since ψ_θ is intricately linked to the eigenvalues of L, the filter is defined as $\psi_\theta(\Omega)$. To approximate $\psi_\theta(\Omega)$, we use Chebyshev Polynomials P_j. By employing P_j, we obtain the approximation $\psi_\omega \approx \sum_{j=0}^{J} \omega_j P_j \left(\frac{2}{\gamma_{\max}}\Omega - I\right)$, where $\gamma_{\max}$ yields the maximum eigenvalue of matrix L, with j^{th} Chebyshev coefficient represented by ω_j. Consequently, the spectral graph convolution on graph $\mathcal{G}$ is expressed as

$$\psi_\omega * \pi \approx \sum_{j=0}^{J} \omega_j P_j \left(\frac{2}{\gamma_{\max}}\Omega - I\right) \pi. \tag{5}$$

The convolution operation is simplified further through $\delta = \omega_0 = -\omega_1$. The aim of this simplification is to reduce the size

of the hyper-parameters while assuming $\gamma_{\max} = 2$ for $J = 1$. Therefore, Eq. (5) is revised as

$$\begin{aligned} \psi_\omega * \pi &\approx \omega_0 P_0(L-I)\pi + \omega_1 P_1(L-I)\pi \\ &= \delta\left(I + D^{-\frac{1}{2}} A D^{-\frac{1}{2}}\right)\pi. \end{aligned} \tag{6}$$

Following the convolution operation in Eq. (6), the graph feature extraction is performed through $L_{\mathcal{G}}$ hidden layers across all nodes of $\mathcal{G}$. The output of each hidden layer is expressed as

$$O^k = \mathrm{ReLU}\left(MO^{k-1}\ \mathrm{W}^k\right) \text{ s.t. } M = \tilde{D}^{-\frac{1}{2}}(A+I)\tilde{D}^{-\frac{1}{2}}, \tag{7}$$

where $\tilde{D}_{ii} = \sum_j (A+I)_{ij}$ and ReLU stands for the rectified linear unit. The feature extraction layers take the input $O^0 = \pi$ and return the output $O^{L_{\mathcal{G}}} = R(\mathcal{G})$, which denotes the temporal representation of $\mathcal{G}$.

C. Graph Encoder

As the graph feature extraction layer captures the spatio-temporal features from π, the proposed GAE model can be interpreted as approximating $P^*(V^* \mid R(G))$ instead of $P^*(V^* \mid \pi)$. The encoder layer takes $X = R(\mathcal{G})$ as input, obtained from the previous layer, and assumes L_E hidden layers and ReLU as activation function at each layer. The objective of the encoder layer is to codify V^* into a latent vector representation $z \in Z$ in a way that the resultant z can be decoded back to V^*. The error function of the encoder is expressed using Kullback–Leibler (KL) divergence as follows:

$$\begin{aligned} \mathrm{E}_E &= KL\left[Q\left(z \mid \pi, V^*\right) \| N(0,1)\right] \\ &= KL\left[Q\left(z \mid R(\mathcal{G}), V^*\right) \| N(0,1)\right]. \end{aligned} \tag{8}$$

To capture the temporal correlations within the time-series data, an LSTM module is integrated to control the recurrent flow of information. The memory module also exhibits remarkable adaptability in addressing the vanishing/exploding gradient problem encountered during the learning process of the time-series data, particularly in scenarios involving extended time intervals. The LSTM cell is equipped with three gates: input gate $i^t_{l_E}$, output gate $o^t_{l_E}$, and forget gate $f^t_{l_E}$. Each LSTM cell presents two states: cell state $C^t_{l_E}$, which retains information over extended time steps, and hidden state $H^t_{l_E}$ (a.k.a. LSTM output). In particular, the following relationships hold:

- $\boldsymbol{i}^t_{l_E} = \varphi\left(\boldsymbol{W}^i_{l_E}\boldsymbol{X}^i_{l_E} + \boldsymbol{U}^i_{l_E}\boldsymbol{h}^{t-1}_{l_E} + \boldsymbol{V}^i_{l_E}\boldsymbol{s}^t_{l_E} + \boldsymbol{b}^i_{l_E}\right)$
- $\boldsymbol{o}^t_{l_E} = \varphi\left(\boldsymbol{W}^o_{l_E}\boldsymbol{X}^o_{l_E} + \boldsymbol{U}^o_{l_E}\boldsymbol{h}^{t-1}_{l_E} + \boldsymbol{V}^o_{l_E}\boldsymbol{s}^t_{l_E} + \boldsymbol{b}^o_{l_E}\right)$
- $\boldsymbol{f}^t_{l_E} = \varphi\left(\boldsymbol{W}^f_{l_E}\boldsymbol{X}^f_{l_E} + \boldsymbol{U}^f_{l_E}\boldsymbol{h}^{t-1}_{l_E} + \boldsymbol{V}^f_{l_E}\boldsymbol{s}^t_{l_E} + \boldsymbol{b}^f_{l_E}\right)$
- $\boldsymbol{C}^t_{l_E} = f^t_{l_E} C^{t-1}_{l_E} + i^t_{l_E} \tanh\left(\boldsymbol{W}^C_{l_E}\boldsymbol{X}^t_{l_E} + \boldsymbol{U}^C_{l_E}\boldsymbol{H}^{t-1}_{l_E} + \boldsymbol{b}^C_{l_E}\right)$
- $\boldsymbol{H}^t_{l_E} = \boldsymbol{o}^t_{l_E} \tanh\left(\boldsymbol{C}^t_{l_E}\right)$,

where $C^{t-1}_{l_E}$ and $H^{t-1}_{l_E}$ represent the cell and hidden state of the previous cell, respectively; $\varphi(\cdot)$ denotes the activation function; $\boldsymbol{W}$ and $\boldsymbol{b}$ refer to the weight matrix and bias, respectively.

D. Attention mechanism

The attention layer dynamically assigns higher significance to timestamps that present a greater impact on generating a specific output [41]. To achieve this, the attention layer receives inputs in the form of the GAE's hidden states, S_h and S_{h-1} at time stage t and $t-1$, respectively. The attention process involves the computation of an alignment score $\boldsymbol{\nu}$ and Softmax function Ω. The alignment score ν is calculated as:

$$\boldsymbol{\nu} = \kappa\left(\boldsymbol{h}^{L_{\mathrm{E}}/2}_t, \boldsymbol{h}^{L_{\mathrm{D}}}_{t-1}\right). \tag{9}$$

The alignment function (κ) represents an FNN trained with both S_h and S_{h-1}. The attention weight is obtained by performing the Softmax transformation on the alignment scores:

$$\boldsymbol{\Omega} = \frac{\exp(\boldsymbol{\nu})}{\sum_{|\boldsymbol{\nu}|}\exp(\boldsymbol{\nu})}, \tag{10}$$

where $|\boldsymbol{\nu}|$ denotes the cardinality of ν. The attention layer returns a context vector $\mathcal{C}$ at time stage t that it is expressed as a weighted sum of GAE's hidden state vectors:

$$\mathcal{C} = \sum_T \Omega \times S_h. \tag{11}$$

E. Graph Decoder

Similar to the encoder, the decoder adopts ReLU as activation function. The objective of the decoder is to generate $\hat{V}$ as close as possible to the output of the previous layer V^*. The reconstruction error is defined as

$$\mathrm{E}_{\mathrm{D}P} = \left\| V^* - \hat{V} \right\|^2. \tag{12}$$

Similar to the graph encoder, the LSTM memory module is also integrated with the graph decoder. The state of each cell in the decoder-LSTM is controlled by input $i^t_{l_D}$, output $o^t_{l_D}$, and forget gate $f^t_{l_D}$ and is governed by these relationships:

- $\boldsymbol{i}^t_{l_D} = \varphi\left(\boldsymbol{W}^i_{l_D}\boldsymbol{X}^i_{l_D} + \boldsymbol{U}^i_{l_D}\boldsymbol{h}^{t-1}_{l_D} + \boldsymbol{V}^i_{l_D}\boldsymbol{s}^t_{l_D} + \boldsymbol{b}^i_{l_D}\right)$
- $\boldsymbol{o}^t_{l_D} = \varphi\left(\boldsymbol{W}^o_{l_D}\boldsymbol{X}^o_{l_D} + \boldsymbol{U}^o_{l_D}\boldsymbol{h}^{t-1}_{l_D} + \boldsymbol{V}^o_{l_D}\boldsymbol{s}^t_{l_D} + \boldsymbol{b}^o_{l_D}\right)$
- $\boldsymbol{f}^t_{l_D} = \varphi\left(\boldsymbol{W}^f_{l_D}\boldsymbol{X}^f_{l_D} + \boldsymbol{U}^f_{l_D}\boldsymbol{h}^{t-1}_{l_D} + \boldsymbol{V}^f_{l_D}\boldsymbol{s}^t_{l_D} + \boldsymbol{b}^f_{l_D}\right)$
- $\boldsymbol{C}^t_{l_D} = f^t_{l_D} C^{t-1}_{l_D} + i^t_{l_D} \tanh\left(\boldsymbol{W}^C_{l_D}\boldsymbol{X}^t_{l_D} + \boldsymbol{U}^C_{l_D}\boldsymbol{H}^{t-1}_{l_D} + \boldsymbol{b}^C_{l_D}\right)$
- $\boldsymbol{H}^t_{l_D} = \boldsymbol{o}^t_{l_D} \tanh\left(\boldsymbol{C}^t_{l_D}\right)$.

F. Training and Testing

The GAE model is developed to differentiate between normal operation (benign samples) and abnormal operation (malicious samples) by looking at the variations from the learned benign samples. This differentiation is accomplished by analyzing the error throughout the reconstruction procedure. The model learns the data patterns from benign samples and detects the samples of abnormality by comparing the reconstructed samples with the original ones. When the reconstruction error surpasses a certain threshold value, the GAE model flags the event of FDIAs. The overall objective function of the proposed model is formulated as:

$$\min_{\Phi} C\left(\boldsymbol{X}, E_E + E_D\right), \tag{13}$$

where Φ indicates the model parameters.

The training procedure of the proposed GAE is outlined in Algorithm 1, where the main aim is to optimize the model parameters: $\phi(\boldsymbol{W}, \boldsymbol{b}, \boldsymbol{U}, \boldsymbol{V})$. The optimization procedure involves an iterative gradient descent algorithm, which is executed via a stochastic gradient descent approach. To expedite the training procedure, we divided the training samples, $X \in X_n$, into equally sized mini-batches, which we feed them into the model over 128 epochs with the learning rate η. In addition, we incorporate convolutional spectral graph layers to facilitate sample differentiation. This allows GAE to be proficient at distinguishing benign from malicious samples. The importance ranking of the nodes during the training process is illustrated in Fig. 6. Once trained, the proposed GAE model applies its learned knowledge to analyze new and previously unseen data. This means that its performance remains stable and reliable even when tested on data that differ from those on which they were initially trained.

Fig. 6: Graph structure of the considered power system with node color indicating the importance of nodes.

V. Experimental Setup

We herein introduce the benchmark detectors and optimize their hyperparameters. The selection of optimum hyperparameters enables them perform best and ensures a balanced comparison between them. Later in this section, we define the evaluation metrics and obtain the AHP outputs for the considered metrics.

A. Benchmark Detectors

We evaluate and compare the detection performance of the proposed strategy with two primary categories of detection models: graph-based detection model (presented in [24]) and traditional ML-based detection models. Later in this section, we briefly introduce a graph CNN (GCNN)-based detection scheme as well as other benchmark schemes.

1) GCNN based Detector: The input layer of GCNN-based detector indicates the graph representation of considered power system. Following the input layer, there are $\mathcal{L}_v$ hidden layers which perform Chebyshev graph convolution on the input data. Let c_{l_v} denote the channel number in a hidden layer l_v, which takes $\boldsymbol{X}^{l_{\mathrm{v}}-1} \in \mathbb{R}^{n \times c l_{\mathrm{v}-1}}$ as input and outputs $\boldsymbol{X}^{l_{\mathrm{v}}} \in \mathbb{R}^{n \times c_{l_{\mathrm{v}}}}$. Each l_v captures the spatial features from the graph structured data by employing graph convolutional operation and ReLU activation function. The ReLU function produces the output tensor X^{l_v} of hidden layer l_v. The subsequent dense layer determines the presence probability of an attacked sample. The output layer conveys the final decision.

For training of the GCNN-based detection model and estimation of its free parameters, we adopt the cross-entropy function:

$$C(\tilde{z}, \sigma) = \frac{-1}{|\boldsymbol{X}_{\mathrm{T}}|} \sum_{\boldsymbol{X}_{\mathrm{T}}} \{z \log(\tilde{z}) + (1-z) \log(1-\tilde{z})\}, \quad (14)$$

where $\boldsymbol{X}_{\mathrm{T}}$ and σ indicate the number of training samples and trainable parameters, respectively; z and $\tilde{z}$ represent the true (benign) and predicted (malicious) labels, respectively. The model is trained using an iterative optimization process based on gradient descent. During this process, the samples X from the original training dataset $\boldsymbol{X}_{\mathrm{T}}$ are divided into evenly sized mini-batches, which are then fed into the model.

2) Traditional ML-based Benchmark Detectors: The traditional ML-based detectors exhibit a range of features, includ-

Algorithm 1 Training procedure of GAE model

function TRAIN_GAE(X_n : *optimal* $\phi(\boldsymbol{W}, \boldsymbol{b}, \boldsymbol{U}, \boldsymbol{V})$)
  ▷ This function aims to minimize training parameters
  while *not converged to minimum* **do**
    for every training instances **X** **do**
      Forward propagation:
      Graph encoding:
      for *every graph encoder layers* $l_E \in \mathcal{L}_E$ **do**
        obtain X^{l_E} through ReLU activation
        for *each time stamp* **do**
          Obtain:
          $\boldsymbol{i}^t_{l_{\mathrm{E}}}, \boldsymbol{o}^t_{l_{\mathrm{E}}}, \boldsymbol{f}^t_{l_{\mathrm{E}}}, \boldsymbol{s}^t_{l_{\mathrm{E}}}$, and $\boldsymbol{h}^t_{l_{\mathrm{E}}}$
        end for
      end for
      for *every graph decoder layers* $l_D \in \mathcal{L}_D$ **do**
        obtain X^{l_D} through ReLU activation
        for *each time stamp* **do**
          Obtain:
          $\boldsymbol{i}^t_{l_{\mathrm{D}}}, \boldsymbol{o}^t_{l_{\mathrm{D}}}, \boldsymbol{f}^t_{l_{\mathrm{D}}}, \boldsymbol{s}^t_{l_{\mathrm{D}}}$, and $\boldsymbol{h}^t_{l_{\mathrm{D}}}$
        end for
      end for
      Back propagation:
      Compute:
      $\min_{\Phi} C(\boldsymbol{X}, E_E + E_D)$
      Obtain the derivatives:
      $\nabla_{\boldsymbol{\mu}^{l(\cdot)}} C, \nabla_{\boldsymbol{b}^{l\cdot()}} C, \nabla_{\boldsymbol{W}^{l(\cdot)}_{(\cdot)}} C,$
      $\nabla_{\boldsymbol{U}^{l(\cdot)}_{(\cdot)}} C$, and $\nabla_{\boldsymbol{V}^{l}_{(\cdot)}(\cdot)} C$
    end for
    Parameter updating rule:
    $\boldsymbol{\mu}^{l(\cdot)} \leftarrow \boldsymbol{\mu}^{l(\cdot)} - \frac{\eta}{|\boldsymbol{X}_{\mathrm{TR}}|} \sum_x \nabla_{\boldsymbol{\mu}^{l(\cdot)}} C$
    $\boldsymbol{b}^{l(\cdot)} \leftarrow \boldsymbol{b}^{l(\cdot)} - \frac{\eta}{|\boldsymbol{X}_{\mathrm{TR}}|} \sum_x \nabla_{\boldsymbol{b}^{l}_{(\cdot)}} C$
    $\boldsymbol{W}^{l_{(\cdot)}}_{(\cdot)} \leftarrow \boldsymbol{W}^{l_{(\cdot)}}_{(\cdot)} - \frac{\eta}{|\boldsymbol{X}_{\mathrm{TR}}|} \sum_x \nabla_{\boldsymbol{W}^{l(\cdot)}_{(\cdot)}} C$
    $\boldsymbol{U}^{l_{(\cdot)}}_{(\cdot)} \leftarrow \boldsymbol{U}^{l_{(\cdot)}}_{(\cdot)} - \frac{\eta}{|\boldsymbol{X}_{\mathrm{TR}}|} \sum_x \nabla_{\boldsymbol{U}^{l(\cdot)}_{(\cdot)}} C$
    $\boldsymbol{V}^{l(\cdot)}_{(\cdot)} \leftarrow \boldsymbol{V}^{l_{(\cdot)}}_{(\cdot)} - \frac{\eta}{|\boldsymbol{X}_{\mathrm{TR}}|} \sum_x \nabla_{\boldsymbol{V}^{l(\cdot)}_{(\cdot)}} C$
  end while
end function

ing shallow or deep architectures, supervised or unsupervised training approaches.

- ARIMA, a shallow unsupervised dynamic model that takes the normal condition data for training and predicts the future pattern by minimizing the MSE. During testing, it flags a data point as an anomaly if the data exceeds a predefined threshold value.
- LSTM, a deep dynamic variation of traditional RNN model, is trained on both normal and attack data in a supervised manner. It operates by holding the previous knowledge through recurrent cycles of information flow.
- FNN, a supervised deep static model, is trained on both normal and attack condition data. It extracts the patterns through stacked hidden layers characterized by fully-connected neurons where information propagates in a feed-forward fashion.
- CNN learns the features adaptively in a supervised manner by employing convolution operations.
- SVM assumes a supervised learning framework by determining the optimum hyperplane that maximizes the margin between classes.

B. Hyperparameters Selection

To maximize the detection performance of the proposed and benchmark models we employ a sequential grid search strategy to fine-tune the models' hyperparameters. The sequential grid search strategy is a straightforward and transparent optimization technique. Given the complexity of our coupled power and transportation network model, we prioritize simplicity to ensure ease. Moreover, it systematically explores a broad range of hyperparameter combinations, ensuring that we do not overlook critical configurations. Lastly, opting for a well-established optimizer, we aim to provide a fair and consistent benchmarking process across all considered models. The sequential hyperparameter selection strategy assumes systematic exploration of the hyperparameter space. The hyperparameters yielding the highest performance during the validation step are selected. The search for the optimal hyperparameter values considers that $\beta = \{\mathcal{L}, \mathcal{P}, \mathcal{O}, \mathcal{U}, \mathcal{A}, \mathcal{K}\}$ is constrained within a predefined search space: number of layers $\mathcal{L}$ =$\{2, 3, 4, 5, 6, 8\}$, rate of dropout $\mathcal{P} = \{0, 0.2, 0.4, 0.5\}$, optimizer $\mathcal{O}$ =$\{$Adam, Rmsprop, SGD, Adamax$\}$, unit number $\mathcal{U} = \{4, 8, 16, 32, 64\}$, order of neighborhood $\mathcal{K} = \{2, 3, 4, 5\}$, and the activation function $\mathcal{A} = \{$Sigmoid, ReLU, Tanh, Elu$\}$. The optimal hyperparameters for LSTM, FNN, CNN, GCNN, and the proposed approach, are $\beta_{\text{LSTM}} = \{3, 0.2, \text{Adam}, 32, \text{Relu}, \text{N/A}\}$, $\beta_{\text{FNN}} = \{4, 0, \text{Adam}, 32, \text{Relu}, \text{N/A}\}$, $\beta_{\text{CNN}} = \{4, 0.4, \text{Rmsprop}, 32, \text{Relu}, 5\}$, $\beta_{\text{GCNN}} = \{5, 0.2, \text{Rmsprop}, 32, \text{RelU}, 4\}$, and $\beta_{\text{GAN}} = \{6, 0.2, \text{Adam}, 64, \text{Relu}, 5\}$, respectively. For the SVM, the optimum kernel, gamma, and regularization are selected as Sigmoid, auto, and 1, respectively. For ARIMA model, after exploring the search space $\{0, 1, 2, 3\}$, we selected the best values for the moving average and differencing degree as 0 and 1, respectively.

C. Evaluation Metrics

To assess the effectiveness of the FDIA detectors under study, we make use of the following performance metrics. First, the detection rate: $\text{DR} = \frac{\text{TP}}{\text{TP+FN}}$, which indicates how many actual malicious samples are detected. Second, the false alarm rate: $\text{FAR} = \frac{\text{FP}}{\text{FP+TN}}$, which is indicative of how often a model erroneously detected non-malicious samples as threats. Lastly, the accuracy metric, $\text{ACC} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}}$, which provides a holistic perspective on the model's performance, revealing its efficacy in detecting both malicious and benign samples. Here, TP (True Positive) and TN (True Negative) denote the correctly identified malicious and benign samples, respectively. In contrast, FP (False Positive) refers to benign samples misidentified as malicious, while FN (False Negative) indicates the malicious samples that the detector overlooked.

The vulnerability of buses not only depends on the topological characteristics but also on the power flow dynamics. Thus, we consider a detailed set of metrics, combining topological and electrical metrics. The topological metrics include i) the CSs neighborhood density, which measures the importance of a bus based on the density of CSs in the neighborhood area, ii) connectivity impact, which refers to the number of buses that remain connected after a failure [49], iii) connectivity loss, which captures the average decrease in the number of generators after a failure event [50], iv) betweenness centrality, which indicates the extent to which a bus lies on the shortest paths connecting other pairs of buses [51]. As for the electrical metrics, they include i) load shedding, which measures the total apparent power after a bus failure [49], ii) effective graph resistance, which captures the total cost to transfer power between a pair of buses [54], iii) electrical degree centrality, which indicates the number of power flows that directly influence the status of a bus.

D. AHP Outputs

The AHP outputs obtained for topological metrics are: CSs neighborhood density: 0.3637, connectivity impact: 0.1612, connectivity loss: 0.2458, and betweenness centrality: 0.2294. For the electrical metrics, the AHP outputs are: Load shedding: 0.7555, Effective Graph Resistance: 0.0300, and electrical degree centrality: 0.2145. After obtaining the weights through AHP analysis, each normalized metric is multiplied by its respective weight. Thereafter, the weighted metric scores are summed up to obtain the vulnerability score of each node. After obtaining the weighted sum of topological as well as electrical metrics we perform AHP analysis again to obtain weight for Topological: 0.8713 and the weight for Electrical: 0.1287 metric.

E. Metrics of User Experience

During the considered attack scenarios, the original power measurements of some buses are tempered, causing more users to be unsatisfied with the service than in normal conditions. Thus, to assess the user experience we considered two metrics: the user satisfaction rate and the percentage of service denied. The user satisfaction rate is defined as: $\mathcal{S} = \frac{P_{cs}}{n_{ev} \times P_{ev}} \times 100\%$,

where $P_{cs} = P_{total} - P_{res}$ signifies the available charging power at a bus with P_{total} and P_{res} representing the total power and total residential load of a bus, respectively. During the considered attack scenarios, the actual power measurement of a CS P_{cs} is tempered causing more users to be unsatisfied with the service than in normal conditions. The residential load is assumed to be variable over the day and modeled following the IEEE reliability test system load pattern [61]. Variable n_{ev} denotes the number of EVs being charged at a bus.

Under the purview of FDIAs, n_{ev} transforms to n_{fev}, indicating the false number of EVs to be served. The denial of service metric is defined as $\mathcal{D} = \frac{|n_{fev} - n_{ev}|}{n_{ev}} \times 100\%$. By employing the user experience metrics we can better anticipate challenges and devise resilient strategies to maintain high user satisfaction rate.

VI. Experimental results and insights

Tables I and II depict the detection performance of the proposed and benchmark models for the three different attacks and four different attack percentage levels. For the combined attack scenario, the additive and deductive attacks are selected with 1:1 ratio.

A. Detection Performance Against Random Attacks

Table I presents the overall detection performance of the considered models against random buses attacks. In this experiment, the models are tested with varying levels of random attack percentage. The findings indicate that as the attack level increases, the detection performance decreases. Such decline is due to the increase in false positives. Specifically, testing the models on additive attacks offers 3-4% performance improvement compared to the combined attack scenarios. Moreover, all the models show similar performance for both the additive and deductive attacks. Overall, the proposed strategy offers about 4-38% performance improvement compared to the benchmark detectors over the different attack types. The improvements in the detection performance of the proposed approach are listed next:

- In the case of additive attacks, the proposed approach improves the performance by 2.63-32.52% in DR, 3.85-40.86% in FAR, and 3.22-33.38% in ACC. GCNN-based detector exhibits slightly lower performance. While still considerable, there might be some data points where GCNN detector does not perform adequately.
- In the case of deductive attacks, the proposed approach achieves 98.19-94.67% in DR, 8.20-12.29% in FAR, and 97.04-94.74% in ACC. GCNN-based detector exhibits slightly lower performance which is still considerable.
- In the case of camouflage attacks, the proposed approach outperforms the other detectors and exhibits 97.28-93.04% in DR, 10.27-13.71% in FAR, and 95.28-93.78% in ACC. The performance gap between the GCNN-based detector and the proposed approach is higher in this scenario. This may due to the fact that the GCNN detector struggles to capture the underlying features set from the complex attack dataset.

TABLE I: Performance of benchmark detectors against different types of attacks and attack injection levels during random node attacks.

Attack type	Detector	Metric	Attack data percentage			
			5%	10%	20%	30%
Additive attacks on random nodes	ARIMA	DR	66.67	59.41	51.13	40.74
		FAR	47.78	53.98	62.99	73.71
		ACC	65.44	58.33	50.40	40.40
	SVM	DR	69.72	63.68	55.58	45.29
		FAR	40.17	46.35	54.36	64.17
		ACC	68.54	62.39	54.17	44.06
	FNN	DR	74.05	70.02	63.08	54.95
		FAR	32.34	37.59	43.14	51.38
		ACC	72.69	68.60	61.50	53.52
	LSTM	DR	80.27	75.09	68.84	61.66
		FAR	26.79	30.15	36.00	44.11
		ACC	78.57	73.54	67.47	60.35
	CNN	DR	83.04	80.29	75.40	68.52
		FAR	20.51	24.52	29.74	36.11
		ACC	82.79	79.98	75.09	68.16
	GCNN	DR	96.56	93.67	90.72	85.96
		FAR	10.77	12.55	16.59	20.92
		ACC	95.60	92.60	89.63	84.78
	GAE	DR	99.19	99.11	98.33	96.33
		FAR	6.92	8.20	9.56	10.78
		ACC	98.82	98.74	97.87	95.80
	SAT-GAE	DR	99.20	99.08	98.18	96.42
		FAR	6.50	8.25	10.01	10.83
		ACC	98.85	98.76	97.90	95.78
Deductive attacks on random nodes	ARIMA	DR	65.33	57.55	49.47	39.20
		FAR	49.00	55.30	64.45	75.29
		ACC	63.90	56.69	49.38	38.53
	SVM	DR	67.97	62.49	54.42	44.12
		FAR	41.40	47.70	55.85	65.69
		ACC	66.99	61.26	52.91	42.73
	FNN	DR	72.64	68.31	61.90	53.49
		FAR	33.60	38.90	44.65	52.89
		ACC	71.63	66.75	59.63	51.79
	LSTM	DR	79.00	74.00	67.22	60.26
		FAR	28.00	31.50	37.45	45.69
		ACC	77.34	71.94	66.19	58.88
	CNN	DR	81.55	79.12	73.53	66.82
		FAR	21.80	25.90	31.25	37.69
		ACC	81.71	78.80	74.05	66.87
	GCNN	DR	95.21	92.43	88.97	84.64
		FAR	12.00	13.90	18.05	22.49
		ACC	94.53	90.71	87.93	83.60
	GAE	DR	98.19	97.38	96.69	94.67
		FAR	8.20	9.50	11.05	12.29
		ACC	97.04	97.18	96.57	94.74
	SAT-GAE	DR	98.07	97.31	96.59	94.63
		FAR	8.24	9.55	11.08	12.36
		ACC	97.00	97.12	96.48	94.61
Combined Attacks on random nodes	ARIMA	DR	64.06	55.70	47.84	37.70
		FAR	50.46	56.76	65.81	77.46
		ACC	62.40	55.08	48.46	36.66
	SVM	DR	66.24	61.38	53.34	43.04
		FAR	43.00	49.52	57.58	67.28
		ACC	65.48	60.22	51.72	41.46
	FNN	DR	71.28	66.62	60.80	52.08
		FAR	35.49	40.25	46.56	54.36
		ACC	70.66	64.90	57.76	50.08
	LSTM	DR	77.80	73.00	65.64	58.92
		FAR	29.42	33.30	38.78	47.90
		ACC	76.18	70.38	64.98	57.46
	CNN	DR	80.10	78.04	71.66	65.14
		FAR	24.04	28.09	33.15	39.91
		ACC	80.72	77.70	73.10	65.64
	GCNN	DR	93.92	91.26	87.24	83.38
		FAR	13.58	15.74	19.49	24.59
		ACC	93.56	88.82	86.26	82.50
	GAE	DR	97.28	95.66	95.08	93.04
		FAR	10.27	10.87	12.71	13.71
		ACC	95.28	95.66	95.34	93.78
	SAT-GAE	DR	97.17	95.54	94.95	92.93
		FAR	10.33	10.94	12.78	13.75
		ACC	95.24	95.60	95.28	93.71

TABLE II: Performance of the benchmark detectors against different types of attacks and attack injection levels during most vulnerable node attacks.

Attack type	Detector	Metric	Attack data percentage			
			5%	10%	20%	30%
Additive attacks on most vulnerable nodes	ARIMA	DR	62.27	52.99	45.48	35.6
		FAR	51.92	58.22	67.11	80.06
		ACC	60.25	52.76	47.24	33.92
	SVM	DR	63.52	59.65	51.66	41.36
		FAR	44.9	51.77	59.68	69.16
		ACC	63.11	58.61	49.86	39.48
	FNN	DR	69.36	64.17	59.29	50.07
		FAR	37.65	41.53	48.74	55.84
		ACC	69.37	62.19	55.02	47.6
	LSTM	DR	75.92	71.45	63.16	56.83
		FAR	31.03	35.5	40.25	50.76
		ACC	74.38	67.93	63.09	55.24
	CNN	DR	78.04	76.55	68.92	62.71
		FAR	26.75	30.73	35.32	42.59
		ACC	79.39	76.2	71.84	63.93
	GCNN	DR	91.91	89.44	84.52	81.42
		FAR	15.44	18.02	21.13	27.28
		ACC	92.05	85.85	83.63	80.79
	GAE	DR	96.08	93.17	92.76	90.68
		FAR	12.71	12.2	14.49	15.11
		ACC	92.71	93.47	93.62	92.61
	SAT-GAE	DR	96.03	93.12	92.71	90.63
		FAR	12.74	12.23	14.52	15.14
		ACC	92.65	93.41	93.56	92.55
Deductive attacks on most vulnerable nodes	ARIMA	DR	61.28	51.25	44.03	34.27
		FAR	52.62	58.92	67.69	81.69
		ACC	58.97	51.34	46.71	32.16
	SVM	DR	61.94	58.87	50.92	40.62
		FAR	45.8	52.95	60.74	70.04
		ACC	61.81	57.93	48.98	38.5
	FNN	DR	68.27	62.65	58.53	48.9
		FAR	38.92	42.09	50.03	56.56
		ACC	68.79	60.46	53.26	46.05
	LSTM	DR	75.02	70.82	61.77	55.76
		FAR	31.69	36.64	40.79	52.44
		ACC	73.54	66.57	62.18	54.06
	CNN	DR	76.83	75.81	67.16	61.2
		FAR	28.47	32.39	36.6	44.29
		ACC	78.77	75.45	71.28	63.01
	GCNN	DR	90.9	88.59	82.94	80.46
		FAR	16.31	19.23	21.81	28.82
		ACC	91.45	84.07	82.13	80.03
	GAE	DR	95.57	91.61	91.34	89.23
		FAR	14.21	12.8	15.46	15.76
		ACC	91.09	92.16	92.69	91.91
	SAT-GAE	DR	95.52	91.56	91.29	89.18
		FAR	14.24	12.83	15.49	15.79
		ACC	91.03	92.1	92.63	91.85
Combined Attacks on most vulnerable nodes	ARIMA	DR	59.15	49.63	40.84	32.66
		FAR	54.41	61.93	69.67	84.86
		ACC	55.18	48.79	44.38	29.95
	SVM	DR	59.97	56.58	48.22	38.95
		FAR	49.06	55.24	64.1	72.54
		ACC	58.79	54.61	46.43	36.02
	FNN	DR	66.19	60.05	56.1	45.94
		FAR	40.67	45.44	51.74	58.38
		ACC	66.48	57.35	49.26	43.85
	LSTM	DR	72.06	67.14	58.3	53.71
		FAR	34.06	39.94	43.17	55.09
		ACC	71.43	64.36	58.61	50.37
	CNN	DR	74.97	73.61	64.35	57.45
		FAR	31.48	35.62	38.81	46.48
		ACC	76.88	72.29	67.42	59.81
	GCNN	DR	87.24	84.81	79.4	76.89
		FAR	17.88	21.33	23.98	31.27
		ACC	88.75	81.77	79.54	77.13
	GAE	DR	93.09	88.66	88.89	87.22
		FAR	16.62	16.06	18.8	18.97
		ACC	89.34	90.19	89.9	88.27
	SAT-GAE	DR	93.04	88.61	88.84	87.17
		FAR	16.65	16.09	18.83	19
		ACC	89.28	90.13	89.84	88.21

B. Detection Performance Against Most Vulnerable Buses Attacks

Table II shows the detection performance of the models while testing against the most vulnerable buses attacks. Similar to the random buses attacks, as the attack percentage increases, their performance decreases. Overall, the proposed detector yields superior performance in all of the three attack strategies. Moreover, the overall performance of the detectors decreases compared to the random buses attack conditions. In summary, the proposed approach for the most vulnerable buses attack strategy attains the following performance metrics.

- In the case of additive attacks, the proposed approach achieves 90.57-95.97% in DR, 12.80-15.20% in FAR, and 92.6-92.38% in ACC. GCNN achieves a comparable performance while ARIMA model performs the worst with 35.44% accuracy
- In the case of deductive attacks, the proposed approach improves the performance by 2.5-30.52% in DR, 3.0-42.86% in FAR, and 3.89-36.8% in ACC. GCNN performs almost similarly to the additive attack scenarios.
- In the case of camouflage attacks, the proposed approach outperforms the other detectors and exhibits 92.98-87.11% in DR, 16.71-19.06% in FAR, and 89.23-88.16% in ACC. The performance gap between the GCNN detector and the proposed approach is more pronounced in this scenario. This confirms the superiority of the proposed approach over the other models.

Our experiments were conducted in an experimental environment where the training process of the proposed and benchmark models was carried out offline. Specifically, we employed an NVIDIA GeForce RTX 3080 hardware accelerator, and the training was implemented using the Keras sequential API. It is worth mentioning that offline training typically takes between three to four hours to complete. The proposed framework was primarily developed and trained offline where it learns features of a normal system as well as potential attack scenarios. Once the model is trained, we deploy it in a real-time environment where it takes 3 miliseconds to detect each testing sample. Research referenced in [62] has shown that at the initial stages, power disturbances tend to evolve slowly, often taking minutes or even hours to manifest significant effects. This finding indicates that our model proves to be sufficiently quick in identifying attacks during the earlier phases of a cascading failure.

However, the challenges associated with real-time decision-making in a dynamic environment include considerations related to data acquisition and preprocessing for real-time applications. Ensuring that data can be collected, processed, and fed into the model with minimal delay is essential for timely decision-making. Network conditions, data transmission delays, and data quality all play a role in this aspect. In our specific case, where the model was trained offline, the transition to real-time deployment would require careful consideration and adaptation of the model architecture, as well as optimization of the inference process. Future work may explore the adaptation of our framework for real-time decision-making, taking into account the specific challenges of dynamic

operational environments in the context of coupled power and transportation networks.

To deal with the wrong detections in the proposed approach, several strategies can be employed. Firstly, data augmentation techniques can be used to increase the diversity and volume of training data which helps the model generalize and reduce errors. Secondly, multiple models can be used in conjunction to make decisions that can improve accuracy and reduce the likelihood of wrong detections. Additionally, fine-tuning the model with a more relevant dataset can significantly enhance its performance. In summary, by enhancing the data and the learning process, we can reduce the errors and improve the overall robustness of the system.

C. Detection Performance Against Noisy Data

In a realistic system setting, the different components of power systems are exposed to different weather conditions and are operated under various conditions. As a result, the data collected from the smart meters often contain interfering signals, for instance, corona noise, jet flyovers, insulator noise, wind-induced noise, or noise due to human intervention. These factors suggest that the signals measured from power systems exhibit irregular and dynamic properties, and crucial fault-related information may be obscured by strong interfering signals. To mimic this scenario, we included Additive White Gaussian Noise (AWGN) to the original data with the signal-to-noise ratio (SNR) varying from 10 to 20 dB (as per [63]). We then tested our proposed system using the generated noisy data. The results presented in Fig. 7 reveal that within the considered noise range, the proposed model maintains a good classification performance. Only 2% decrease in detection performance is observed when referencing with respect to the low noise condition at 10 dB SNR. Throughout the robustness analysis, we maintained the training data devoid of noise and evaluated the system using data injected with noise that it had not encountered previously. The conducted analysis confirms the noise-immune performance of the proposed method.

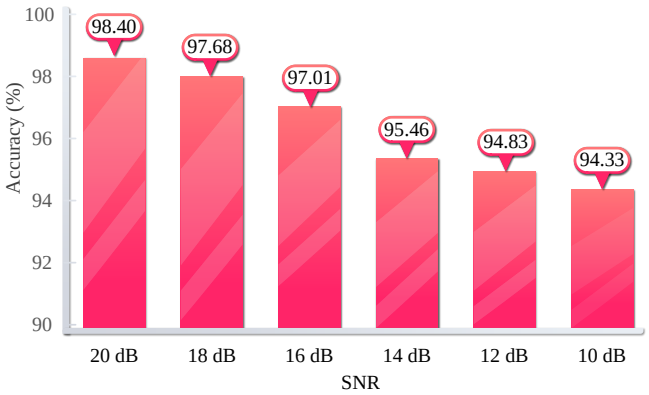

Fig. 7: Performance of the proposed model against different levels of noise.

D. Impact of Attack Detection on User Satisfaction Metrics

This section explores how the aforementioned detection mechanism improves the EV power satisfaction rate and lessens the percentage of service denied. Tables III and IV illustrate how the post-implementation of the proposed detection mechanism improves the EV power satisfaction rate and lowers the denial of services, respectively. In a normal setting and with all the initially integrated CSs, the EV power satisfaction rate observed is 89.36%, and the service denial rate is 11.28%. Later in this section, we will investigate the change in user experience if corrective measures against the FDIAs were taken based on the detection performance of the proposed detector.

- Table III lists the EV power satisfaction rate when the attack is active and after mitigating it. Overall, we can see an average of 11-30% improvement in EV power satisfaction rate over the considered attack scenarios. Specifically, in the presence of 5% and 10% attack levels, the detector offers the most stable performance with only 1.5-2.5% degradation of EV power satisfaction rate relative to the normal condition. On the other hand, a slight degradation in EV power satisfaction rate is observed for the higher percentage of attack level, yet the detector improves the performance by 30-33% compared to the active attack conditions. Such superior performance is due to the model's generalization ability against different attack conditions.
- Table IV presents the percentage of services denied during attacks and after detecting and mitigating them. Across all types and levels of attacks, the service denial rate decreases by 3.84 to 36.59%. With the increase in attack levels, a higher denial of service is observed. The highest deviation in denial rate is observed for the most vulnerable node attacks with an increment of only 6%.

TABLE III: $\mathcal{S}$ (%) during attacks and after detection.

		Before detection				After detection			
Attack level		5%	10%	20%	30%	5%	10%	20%	30%
RNA	Additive	76.98	71.97	67.46	59.80	87.54	87.29	85.61	83.29
	Deductive	73.79	70.63	62.13	51.23	85.77	84.92	82.39	80.91
	Combined	68.23	59.63	53.17	47.32	82.21	81.79	79.43	77.85
VNA	Additive	73.42	68.53	65.33	54.74	87.04	86.16	85.11	82.18
	Deductive	71.63	67.21	59.97	46.22	85.12	84.48	82.01	79.94
	Combined	66.88	56.55	50.28	43.87	81.73	80.71	78.33	77.24

TABLE IV: $\mathcal{D}$ (%) during attacks and after detection.

		Before detection				After detection			
Attack level		5%	10%	20%	30%	5%	10%	20%	30%
RNA	Additive	17.48	25.87	36.21	43.29	13.64	15.55	16.30	17.20
	Deductive	18.48	29.47	39.22	49.65	14.03	15.91	16.37	17.73
	Combined	20.36	32.42	42.87	51.20	14.78	16.29	16.72	17.80
VNA	Additive	20.63	28.01	40.71	48.85	14.57	15.58	16.54	17.61
	Deductive	21.03	30.87	43.01	51.56	15.18	16.06	17.08	17.84
	Combined	24.46	39.20	46.87	55.39	15.42	16.35	17.46	18.80

E. Computational Cost and Accuracy Analysis

We investigated the costs and benefits associated with reducing the model size of the Graph Autoencoder (GAE) equipped with the attention mechanism. In this regard, we compared

the reconstruction Accuracy and the computational efficiency of the compromised models with a baseline model. Table V summarizes the comparative analysis of different models in terms of computational cost and autoencoder reconstruction accuracy. We started with the original GAE model, which has 6 layers and 64 neurons per layer. The original model is optimized with a sequential grid search algorithm. The detail of the hyperparameter optimization is discussed in Section V. This optimized model serves as our baseline for performance and computational efficiency. We experimented by reducing the number of layers in the GAE while keeping the number of neurons per layer constant . This model performed with slightly lower accuracy though with faster training time. Later, we kept the number of GNN layers constant but reduced the number of neurons in each layer. This leads to faster training and inference times with a slight reduction in performance. Lastly, we performed a combined reduction in both the number of layers and neurons per layer. From the aforementioned analysis, it is observed that the larger models require comparatively longer training time as they extract more feature information from the data. However, the training time and computational ability of the proposed model do not create a significant concern as the system operators can conduct offline training on available datasets periodically (weekly or monthly). Testing can be done in real time as per the reported decision time which is 3 millisecond.

TABLE V: Model comparison based on training time and reconstruction accuracy.

Model description	Reconstruction accuracy (%)	Training time
Baseline model ($\mathcal{L}=6,\mathcal{U}=64$)	97.45	2h 30m
Reduced layer model ($\mathcal{L}=4,\mathcal{U}=64$)	94.20	2h 10m
Reduced neuron model ($\mathcal{L}=6,\mathcal{U}=32$)	92.31	1h 50m
Reduced neuron and layer model ($\mathcal{L}=4,\mathcal{U}=32$)	87.49	1h 25m

VII. Conclusions

This paper proposed a GAN-based FDIA detection framework and investigated its performance against different attack types and attack injection levels. FDIAs on coupled power and transportation networks can cause up to 45.73% degradation in EV power satisfaction rate and 44.11% escalation in services denial rate. Our extensive simulation studies have revealed that the proposed detector performed with as high as 95.80% accuracy while improving the EV power satisfaction rate by 10-33%, and decreasing the services denial rate by 4-36%. Moreover, the comparative performance analysis against the benchmark detectors showed an average of 30% improvement in DR for random node attacks and 35% improvement in DR for the most vulnerable attack strategy. The proposed GAE-based detector employs an autoencoder combined with Chebyshev graph convolution recurrent layers that facilitate extraction of the spatial and temporal correlations from the measurement data. Localization and mitigation of the attacks at node level and in the presence of errors and dynamic changes in the network represent open research problems.

Appendix A
Parameters of the proposed model

Number of layers, and neurons per layers: $\mathcal{L}=6$, $\mathcal{U}=64$
Dropout rate, $\mathcal{P}=0.2$
Optimizer, $\mathcal{O}=Adam$
order of neighborhood, $\mathcal{K}=5$
Number of nodes= 2000
Number of edges= 2667

References

[1] Z. Yang, Y. Xiang, K. Liao, and J. Yang, "Research on security defense of coupled transportation and cyber-physical power system based on the static bayesian game," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 3, pp. 3571–3583, 2022.

[2] A. Zahedmanesh, K. M. Muttaqi, and D. Sutanto, "A cooperative energy management in a virtual energy hub of an electric transportation system powered by pv generation and energy storage," *IEEE Transactions on Transportation Electrification*, vol. 7, no. 3, pp. 1123–1133, 2021.

[3] W. Gan, M. Shahidehpour, J. Guo, W. Yao, S. Pandey, E. A. Paaso, A. Vukojevic, and J. Wen, "A tri-level planning approach to resilient expansion and hardening of coupled power distribution and transportation systems," *IEEE Trans. Power Syst.*, vol. 37, no. 2, pp. 1495–1507, 2021.

[4] K. Xiahou, Y. Liu, and Q. Wu, "Decentralized detection and mitigation of multiple false data injection attacks in multiarea power systems," *IEEE J. of Emerg. and Sel. Topics in Ind. Electron.*, vol. 3, no. 1, pp. 101–112, 2021.

[5] K.-D. Lu and Z.-G. Wu, "Multi-objective false data injection attacks of cyber–physical power systems," *IEEE Trans. Circuits and Syst. II: Express Briefs*, vol. 69, no. 9, pp. 3924–3928, 2022.

[6] A. Sayghe, Y. Hu, I. Zografopoulos, X. Liu, R. G. Dutta, Y. Jin, and C. Konstantinou, "Survey of machine learning methods for detecting false data injection attacks in power systems," *IET Smart Grid*, vol. 3, no. 5, pp. 581–595, 2020.

[7] H. Long, Z. Wu, C. Fang, W. Gu, X. Wei, and H. Zhan, "Cyber-attack detection strategy based on distribution system state estimation," *Journal of Modern Power Systems and Clean Energy*, vol. 8, no. 4, pp. 669–678, 2020.

[8] P. Zhuang, R. Deng, and H. Liang, "False data injection attacks against state estimation in multiphase and unbalanced smart distribution systems," *IEEE Transactions on Smart Grid*, vol. 10, no. 6, pp. 6000–6013, 2019.

[9] A. Meng, H. Wang, S. Aziz, J. Peng, and H. Jiang, "Kalman filtering based interval state estimation for attack detection," *Energy Procedia*, vol. 158, pp. 6589–6594, 2019.

[10] R. Moslemi, A. Mesbahi, and J. M. Velni, "A fast, decentralized covariance selection-based approach to detect cyber attacks in smart grids," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 4930–4941, 2017.

[11] F. Mohammadi, "Emerging challenges in smart grid cybersecurity enhancement: A review," *Energies*, vol. 14, no. 5, p. 1380, 2021.

[12] M. Esmalifalak, L. Liu, N. Nguyen, R. Zheng, and Z. Han, "Detecting stealthy false data injection using machine learning in smart grid," *IEEE Syst. J.*, vol. 11, no. 3, pp. 1644–1652, 2014.

[13] X. Lu, J. Jing, and Y. Wu, "False data injection attack location detection based on classification method in smart grid," in *2020 2nd International Conference on Artificial Intelligence and Advanced Manufacture (AIAM)*. IEEE, 2020, pp. 133–136.

[14] D. Wang, X. Wang, Y. Zhang, and L. Jin, "Detection of power grid disturbances and cyber-attacks based on machine learning," *Journal of Information Security and Applications*, vol. 46, pp. 42–52, 2019.

[15] Y. Zhang, J. Wang, and B. Chen, "Detecting false data injection attacks in smart grids: A semi-supervised deep learning approach," *IEEE Trans. Smart Grid*, vol. 12, no. 1, pp. 623–634, 2020.

[16] D. Xue, X. Jing, and H. Liu, "Detection of false data injection attacks in smart grid utilizing elm-based ocon framework," *IEEE Access*, vol. 7, pp. 31 762–31 773, 2019.

[17] E. M. Ferragut, J. Laska, M. M. Olama, and O. Ozmen, "Real-time cyber-physical false data attack detection in smart grids using neural networks," in *2017 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE, 2017, pp. 1–6.

[18] Y. Wang, Z. Zhang, J. Ma, and Q. Jin, "Kfrnn: An effective false data injection attack detection in smart grid based on kalman filter and recurrent neural network," *IEEE Internet Things J.*, vol. 9, no. 9, pp. 6893–6904, 2021.

[19] R. Zheng, J. Gu, Z. Jin, H. Peng, and Y. Zhu, "Load forecasting under data corruption based on anomaly detection and combined robust regression," *International Transactions on Electrical Energy Systems*, vol. 30, no. 7, p. e12103, 2020.

[20] Y. Li and Y. Wang, "False data injection attacks with incomplete network topology information in smart grid," *IEEE Access*, vol. 7, pp. 3656–3664, 2018.

[21] L. Yang, Y. Li, and Z. Li, "Improved-elm method for detecting false data attack in smart grid," *International Journal of Electrical Power & Energy Systems*, vol. 91, pp. 183–191, 2017.

[22] S. Wang, S. Bi, and Y.-J. A. Zhang, "Locational detection of the false data injection attack in a smart grid: A multilabel classification approach," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8218–8227, 2020.

[23] G. Zhang, J. Li, O. Bamisile, D. Cai, W. Hu, and Q. Huang, "Spatio-temporal correlation-based false data injection attack detection using deep convolutional neural network," *IEEE Trans. Smart Grid*, vol. 13, no. 1, pp. 750–761, 2021.

[24] O. Boyaci, A. Umunnakwe, A. Sahu, M. R. Narimani, M. Ismail, K. R. Davis, and E. Serpedin, "Graph neural networks based detection of stealth false data injection attacks in smart grids," *IEEE Syst. J.*, vol. 16, no. 2, pp. 2946–2957, 2021.

[25] A. S. Musleh, G. Chen, and Z. Y. Dong, "A survey on the detection algorithms for false data injection attacks in smart grids," *IEEE Trans. Smart Grid*, vol. 11, no. 3, pp. 2218–2234, 2019.

[26] A. Yaniv, P. Kumar, and Y. Beck, "Towards adoption of gnns for power flow applications in distribution systems," *Electric Power Systems Research*, vol. 216, p. 109005, 2023.

[27] T. Ishizaki, A. Chakrabortty, and J.-I. Imura, "Graph-theoretic analysis of power systems," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 931–952, 2018.

[28] T. Werho, V. Vittal, S. Kolluri, and S. M. Wong, "Power system connectivity monitoring using a graph theory network flow algorithm," *IEEE Transactions on Power Systems*, vol. 31, no. 6, pp. 4945–4952, 2016.

[29] C. Yuan, Y. Zhou, G. Zhang, G. Liu, R. Dai, X. Chen, and Z. Wang, "Exploration of graph computing in power system state estimation," in *2018 IEEE Power & Energy Society General Meeting (PESGM)*. IEEE, 2018, pp. 1–5.

[30] H. Tong, R. C. Qiu, D. Zhang, H. Yang, Q. Ding, and X. Shi, "Detection and classification of transmission line transient faults based on graph convolutional neural network," *CSEE Journal of Power and Energy Systems*, vol. 7, no. 3, pp. 456–471, 2021.

[31] W. Lin, D. Wu, and B. Boulet, "Spatial-temporal residential short-term load forecasting via graph neural networks," *IEEE Trans. Smart Grid*, vol. 12, no. 6, pp. 5373–5384, 2021.

[32] S. H. Haghshenas, M. A. Hasnat, and M. Naeini, "A temporal graph neural network for cyber attack detection and localization in smart grids," in *2023 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*. IEEE, 2023, pp. 1–5.

[33] X. Du, J. Yu, Z. Chu, L. Jin, and J. Chen, "Graph autoencoder-based unsupervised outlier detection," *Information Sciences*, vol. 608, pp. 532–550, 2022.

[34] B. L. Nguyen, T. V. Vu, T.-T. Nguyen, M. Panwar, and R. Hovsapian, "Spatial-temporal recurrent graph neural networks for fault diagnostics in power distribution systems," *IEEE Access*, 2023.

[35] N. Retiére, D. T. Ha, and J.-G. Caputo, "Spectral graph analysis of the geometry of power flows in transmission networks," *IEEE Syst. J.*, vol. 14, no. 2, pp. 2736–2747, 2019.

[36] B. A. ugli Olimov, K. C. Veluvolu, A. Paul, and J. Kim, "Uzadl: Anomaly detection and localization using graph laplacian matrix-based unsupervised learning method," *Computers & Industrial Engineering*, vol. 171, p. 108313, 2022.

[37] Q. Wang, W. Tai, Y. Tang, and M. Ni, "Review of the false data injection attack against the cyber-physical power system," *IET Cyber-Physical Systems: Theory & Applications*, vol. 4, no. 2, pp. 101–107, 2019.

[38] 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, 2021.

[39] Y. Han, H. Feng, K. Li, and Q. Zhao, "False data injection attacks detection with modified temporal multi-graph convolutional network in smart grids," *Computers & Security*, vol. 124, p. 103016, 2023.

[40] A. Takiddin, R. Atat, M. Ismail, O. Boyaci, K. R. Davis, and E. Serpedin, "Generalized graph neural network-based detection of false data injection attacks in smart grids," *IEEE Trans. Emerg. Topics in Comput. Intell.*, 2023.

[41] A. Takiddin, M. Ismail, U. Zafar, and E. Serpedin, "Robust electricity theft detection against data poisoning attacks in smart grids," *IEEE Transactions on Smart Grid*, vol. 12, no. 3, pp. 2675–2684, 2020.

[42] ——, "Deep autoencoder-based anomaly detection of electricity theft cyberattacks in smart grids," *IEEE Systems Journal*, vol. 16, no. 3, pp. 4106–4117, 2022.

[43] W. Liao, B. Bak-Jensen, J. R. Pillai, Y. Wang, and Y. Wang, "A review of graph neural networks and their applications in power systems," *Journal of Modern Power Systems and Clean Energy*, vol. 10, no. 2, pp. 345–360, 2021.

[44] P. Zhao, S. Li, P. J.-H. Hu, Z. Cao, C. Gu, D. Xie, and D. D. Zeng, "Coordinated cyber security enhancement for grid-transportation systems with social engagement," *IEEE Trans. Emerg. Topics in Comput. Intell.*, 2022.

[45] M. Wolf, A. Weimerskirch, and C. Paar, "Security in automotive bus systems," in *Workshop on Embedded Security in Cars*. Bochum, 2004, pp. 1–13.

[46] R. A. Biroon, Z. A. Biron, and P. Pisu, "False data injection attack in a platoon of cacc: real-time detection and isolation with a pde approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 8692–8703, 2021.

[47] J. B. Hansen, S. N. Anfinsen, and F. M. Bianchi, "Power flow balancing with decentralized graph neural networks," *IEEE Trans. Power Syst.*, 2022.

[48] O. Boyaci, M. R. Narimani, K. Davis, and E. Serpedin, "Cyberattack detection in large-scale smart grids using chebyshev graph convolutional networks," in *2022 9th International Conference on Electrical and Electronics Engineering (ICEEE)*. IEEE, 2022, pp. 217–221.

[49] G. J. Correa and J. M. Yusta, "Grid vulnerability analysis based on scale-free graphs versus power flow models," *Electr. Power Syst. Res.*, vol. 101, pp. 71–79, 2013.

[50] R. Kinney, P. Crucitti, R. Albert, and V. Latora, "Modeling cascading failures in the north american power grid," *Eur. Phys. J. B*, vol. 46, pp. 101–107, 07 2005.

[51] F. Jamour, S. Skiadopoulos, and P. Kalnis, "Parallel algorithm for incremental betweenness centrality on large graphs," *IEEE Trans. Parallel Distrib. Syst.*, vol. 29, no. 3, pp. 659–672, 2018.

[52] D. Deka, S. Vishwanath, and R. Baldick, "Analytical models for power networks: The case of the western U.S. and ERCOT grids," *IEEE Trans. Smart Grid*, vol. 8, no. 6, pp. 2794–2802, Nov 2017.

[53] E. I. Bilis, W. Kröger, and C. Nan, "Performance of electric power systems under physical malicious attacks," *IEEE Syst. Journal*, vol. 7, no. 4, pp. 854–865, 2013.

[54] X. Wang *et al.*, "A network approach for power grid robustness against cascading failures," in *2015 7th Int. Workshop on Reliable Netw. Design and Modeling (RNDM)*, 2015, pp. 208–214.

[55] A. B. M. Nasiruzzaman, H. R. Pota, and M. A. Mahmud, "Application of centrality measures of complex network framework in power grid," in *IECON 2011 - 37th Annual Conf. of the IEEE Ind. Electron. Soc.*, 2011, pp. 4660–4665.

[56] S. Chanda and A. K. Srivastava, "Defining and enabling resiliency of electric distribution systems with multiple microgrids," *IEEE Trans. Smart Grid*, vol. 7, no. 6, pp. 2859–2868, 2016.

[57] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "Matpower: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 12–19, 2010.

[58] "ercot 2020," Nov 2020. [Online]. Available: https://www.ercot.com/mktinfo/loadprofile/alp

[59] C. Stamile, A. Marzullo, and E. Deusebio, *Graph Machine Learning: Take graph data to the next level by applying machine learning techniques and algorithms*. Packt Publishing Ltd, 2021.

[60] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," *arXiv preprint arXiv:1707.01926*, 2017.

[61] M. F. Shaaban *et al.*, "Joint planning of smart ev charging stations and dgs in eco-friendly remote hybrid microgrids," *IEEE Trans. Smart Grid*, vol. 10, no. 5, pp. 5819–5830, 2019.

[62] M. Almassalkhi and I. Hiskens, "Chapter 5 - impact of energy storage on cascade mitigation in multi-energy systems," in *Energy Storage for Smart Grids*. Academic Press, 2015, pp. 115–169.

[63] K. Chen, J. Hu, and J. He, "Detection and classification of transmission line faults based on unsupervised feature learning and convolutional sparse autoencoder," *IEEE Transactions on Smart Grid*, vol. 9, no. 3, pp. 1748–1758, 2016.