Spatio-Temporal Graph-Based Generation and Detection of Adversarial False Data Injection Evasion Attacks in Smart Grids

Abdulrahman Takiddin, *Member, IEEE*, Muhammad Ismail, *Senior Member, IEEE*, Rachad Atat, *Senior Member, IEEE*, and Erchin Serpedin, *Fellow, IEEE*

***Abstract*—Smart power grids are vulnerable to security threats due to their cyber-physical nature. Existing data-driven detectors aim to address simple traditional false data injection attacks (FDIAs). However, adversarial false data injection evasion attacks (FDIEAs) present a more serious threat as adversaries, with different levels of knowledge about the system, inject adversarial samples to circumvent the grid's attack detection system. The robustness of state-of-the-art graph-based detectors has not been investigated against sophisticated FDIEAs. Hence, this paper answers three research questions: (a) What is the impact of utilizing spatio-temporal features to craft adversarial samples and how to select attack nodes? (b) How can adversaries generate surrogate spatio-temporal data when they lack knowledge about the system topology? (c) What are the required model characteristics for a robust detection against adversarial FDIEAs? To answer the questions, we examine the robustness of several detectors against five attack cases and conclude the following: (a) Attack generation with full knowledge using spatio-temporal features leads to $5-26\%$ and $2-5\%$ higher degradation in detection rate (DR) compared to traditional FDIAs and using temporal features, respectively, whereas centrality analysis-based attack node selection leads to $3-11\%$ higher degradation in DR compared to a random selection; (b) Stochastic geometry-based graph generation to create surrogate adversarial topologies and samples leads to $3-13\%$ higher degradation in DR compared to traditional FDIAs; (c) Adopting an unsupervised spatio-temporal graph autoencoder (STGAE)-based detector enhances the DR by $5-53\%$ compared to benchmark detectors against FDIEAs.**

***Impact Statement*—Cyber-physical systems, such as smart power grids, generate and exchange lots of data among their components, which makes them vulnerable to various attack types. To detect such attacks, tools based on artificial intelligence (AI) are being proposed since they can learn the system behavior. Simple attack types present obvious deviations in the data patterns and can be spotted using classical AI-based attack detectors. However, complex attack types can be injected in a stealthy way and bypass the detectors, especially when attackers have access to leaked information about the system. Such cases lead to further degradation in the attack detection rates of existing detectors by up to 26% compared to simple attacks, causing system instability. This paper answers questions on what is required for an AI-based detector to be robust against such complex attacks when attackers have full, partial, and no knowledge about the system. It turns out that detectors that capture information about the grid connectivity and temporal aspects lead to enhanced attack detection rates of up to 53% compared to classical AI-based detectors. The provided answers in this paper are relevant to wider applications in cyber-physical systems, industries, and internet of things since they follow similar environments.**

***Index Terms*—cyber-physical systems security, cyberattacks, evasion attacks, false data injection attacks, graph autoencoder, graph neural networks, machine learning, smart grids.**

Abdulrahman 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).

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

Rachad Atat is with the Computer Science and Mathematics Department, Lebanese American University, Beirut, Lebanon (email: rachad.atat@lau.edu.lb).

Erchin Serpedin is with the Electrical and Computer Engineering Department, Texas A&M University, College Station, TX 77843 USA (e-mail: eserpedin@tamu.edu).

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

I. Introduction

THE integration of cyber and physical elements within a smart power grid renders it a highly complex system, characterized by the continuous exchange of measurement data among these components [1]. Smart power grids present cyber-physical systems that necessitate accurate measurement data for the attainment of optimal operation, decision-making, and situational awareness [2]. Thus, ensuring data integrity is essential to maintain power system reliability [3]. Nevertheless, the occurrence of false data injection attacks (FDIAs) can compromise the integrity of measurement data [4] due to the manipulation of sensor data by malicious entities (adversaries). Such actions can lead to system overload, owing to the emergence of incorrect operational decisions. FDIAs pose a significant challenge when executed in a stealthy manner [5], especially adversarial false data injection evasion attacks (FDIEAs) that circumvent traditional bad data detection (BDD) systems.

A. Related Work and Limitations

Existing data-driven approaches mostly focus on detecting traditional FDIAs such as replay attacks on smart power grids. However, while evasion attacks present a major threat to cyber-physical systems [6], the detection of adversarial FDIEAs is limited in smart grid literature. We review existing traditional and adversarial detectors next.

1) Detection of Traditional FDIAs: We classify data-driven approaches based on their ability of capturing spatial (i.e., grid power topology) and temporal (i.e., time-series correlations) aspects as follows.

a) Spatially-Unaware Detection: The relevant detectors rely on classical machine learning-based approaches with shallow or deep structures that do not capture the topological aspect of the grid. These detectors aim to detect traditional FDIAs only. Among the most notable results, shallow support vector machine (SVM) [7] and decision tree [8] detection schemes offered F1-Scores of 82% and 88%, respectively. A

shallow random forest-based scheme reported a detection rate (DR) of 93% against basic attacks [9]. Such shallow detectors offer limited detection capabilities since they fail to fully capture the complex patterns present within the measurement data of different grid components [10]. Therefore, machine learning-based detectors with shallow structures present unreliable solutions. Detectors that are built using artificial neural networks overcome the pattern complexity limitation by employing deep structures with stacked layers as follows. Different feedforward neural network (FNN)-based approaches offered accuracy scores of $90-99\%$ [11], [12]. Autoencoder and recurrent neural network (RNN)-based approaches provided DRs of 96% [13], [14]. A convolutional neural network (CNN)-based approach yielded a row accuracy of 97% [15]. Despite the improved reported detection performance due to capturing complex patterns, such models still do not capture the spatial aspect of the grid topology [5] and hence we refer to them as "spatially-unaware" detectors.

b) Spatially-Aware Detection: Spatially-aware detectors based on graph signal processing (GSP) [16] and graph neural networks (GNNs) were proposed to capture the spatial topological aspects. The GSP-based tools offered 90% in DR while requiring manual designing of custom filters [17], [18]. Such a limitation restricts the scalability of the model to a specific network size. Hence, GSP-based approaches are also not considered reliable despite capturing the spatial features [5]. A GNN-based approach that utilizes undirected graphs to model the power system provided DRs of $83-96\%$ using a convolutional GNN (CGNN) structure [3]. Such a detector employs a supervised learning model trained on labeled data with a finite number of attack types. Thus, supervised GNNs are vulnerable to unseen (zero-day) attacks that are not part of their training sets [19]. Also, due to the lack of recurrent layers within the model, the CGNN-based detector does not capture the temporal correlations within the grid's time-series datasets. Besides, such a detector fails to generalize a detection strategy in case of system seasonal topological reconfigurations as it is trained on a specific topological configuration [5]. To overcome the generalization limitation, a state-of-the-art generalized graph autoencoder (GAE)-based detector trained on multiple topological configurations reported $94-99\%$ in DRs against only simple traditional FDIAs in unseen topologies [5], yet it lacks apprehending the temporal aspects due to its feedforward structure.

c) Spatio-Temporal-Aware Detection: Since smart grids rely on temporal (time-series) data with spatial aspects, spatio-temporal-aware FDIA detectors capturing both aspects were proposed. A spatio-temporal RNN (STRNN) model offered an accuracy of 91% on IEEE 118 and 300-bus systems [20]. A spatio-temporal graph neural network (STGNN) approach that classifies cyber events using supervised learning offered an accuracy of 96% on a test distribution system [21]. A spatio-temporal autoencoder (STAE) model showed an accuracy of 98% on an IEEE 39-bus system [22]. A spatio-temporal CNN (STCNN) model offered an accuracy of 99% on an IEEE 39-bus system [23]. A spatio-temporal fully-connected neural network (STFNN) approach offered a precision of 99% on 95 and 255-bus systems [24]. A spectrum-based neural network approach offered an accuracy of 99% on IEEE 9, 39, and 118-bus systems [25]. A spatio-temporal federated learning approach provided an accuracy of 99% on IEEE 14 and 118-bus systems [26]. Despite reporting high detection performance, such studies lack at least one of the following aspects. First, they offer supervised and static detection that is limited to the attack types and system configuration that they are trained on. Thus, such detectors present vulnerability to new unseen attacks as well as system reconfigurations. Second, they offer detection against simple FDIAs, without investigating the robustness against adversarial FDIEAs. Third, they implement non-graph approaches, which limits their detection performance compared to graph-based detectors that capture the grid's topological configurations via the Chebyshev graph convolution operator. Fourth, they are tested against specific system sizes, which limits their generalization and scalability abilities [5]. Such drawbacks lead to limited detection performance (as will be discussed in Section V-C2) compared to other spatio-temporal graph-based detectors.

The detection performance of the aforementioned detectors [3], [5], [7] - [9], [11] - [15], [17], [18], [23] - [26] is reported only against simple traditional FDAIs with the aim of deceiving the grid's control systems and operations. This means that such studies neglect an important cybersecurity aspect, which is adversarial FDIEAs that circumvent the machine learning-based detectors via injecting adversarial attack samples that have similar patterns to benign ones [6]. Specifically, the aforementioned detectors have not been tested against complex adversarial FDIEAs that are more challenging to detect due to the presented similarity between adversarial and benign samples. Next, we review studies that attempt to capture adversarial attacks.

2) Detection of Adversarial FDIEAs: Such attacks may take place in two forms; data poisoning [27] or evasion attacks [6]. Data poisoning denotes incorrectly labeled malicious samples in the training set [28]. Evasion attacks present cases where malicious entities inject adversarial samples into the system to fool the adopted detectors [6]. In evasion attacks, the injected adversarial samples present similar patterns to benign samples and hence circumvent the detector, which is the focus of this paper. Next, we review adversarial attack detectors in power grids and common adversarial detectors in different domains.

a) Adversarial-Specific Detectors in Power Grids: Several studies concluded that evasion attacks severely deteriorate the detection performance in power systems [29], [30], [31]. Therefore, the following adversarial-specific detectors (i.e., detectors that are designed specifically to only detect adversarial attacks) have been proposed. An integrated generative adversarial network (GAN) approach offered 96% in accuracy on IEEE 13 and 123-bus systems [32]. A spatio-temporal generative adversarial network (STGAN) approach provided 97% in accuracy on IEEE 13 and 33-bus systems [33]. An adversarial machine learning approach offered 99% in accuracy on IEEE 14, 30 118-bus systems [34]. Other approaches based on adversarial training (AT) and sequential ensemble learning (SEL) offered 91% and 95% in accuracy, respectively, against electricity theft evasion attacks [35] [6]. However, such

solutions present at least one of the following limitations. First, they offer adversarial attack-specific detection solutions, which means that they are designed specifically to detect adversarial attacks, whereas in reality, power systems could encounter other attack types (e.g., traditional FDIAs) [36]. Second, such solutions require implementing an additional block to the detection mechanism specifically to detect adversarial attacks, which increases computational complexity. Third, they present supervised learning that requires including adversarial samples in the training set of the models, which limits the detection to seen attack types only. Fourth, they do not capture the spatio-temporal aspects of smart power grids.

b) Adversarial Detectors in Other Domains: Other defense approaches against evasion attacks have been proposed in different domains. First, an adversarial training-based approach [37] that incorporates adversarial samples into the training set was proposed to familiarize the model with such samples. However, malicious entities may launch new adversarial attack samples that are not part of the training set. Second, a certified defense-based approach [38] was proposed to issue a robustness certificate for a fixed network size. However, network sizes vary according to the application. Third, an approach utilizing a reformer model was proposed to bring adversarial samples closer in proximity to a manifold of benign samples [39]. However, such a solution is limited to one attack setting where adversaries have full knowledge about the detection details, but in reality, adversaries may have different knowledge levels, as will be discussed in Section II-B6.

3) Additional Remarks: The reviewed studies lack a common comparison ground as they report multiple detection performance metrics from various domains based on different experiments conducted on multiple system sizes with various injection levels, which makes it challenging to draw conclusions using only the reported detection performance. A high reported detection performance (e.g., 99% in DR) in the literature does not necessarily reflect the effectiveness and reliability of the model since the results are mostly limited to detecting simple FDIA types (included during training) using relatively small datasets while being trained and tested on the same topological configurations, which limits the model's scalability. It is also worth noting that the reviewed studies resorted to utilizing computer software and simulation tools to generate normal operation and attack samples due to the lack of readily available benign and attack datasets.

B. Research Questions

To the best of our knowledge, generalized state-of-the-art spatially-aware FDIA detectors have been only tested against simple attacks on random nodes [5]. Thus, it is worth investigating the robustness of such detectors against sophisticated adversarial FDIEAs with multiple attack cases, which prompts the following research questions that this paper addresses:

- Since existing studies only utilized temporal features to craft adversarial samples, *what is the impact of utilizing spatio-temporal features to craft adversarial samples and how to select attack nodes?*
- In case utilizing spatio-temporal features increases the adversarial FDIEAs impact, *how can adversaries generate surrogate spatio-temporal data when they lack knowledge about the system topology?*
- From a defense perspective, *what are the required model characteristics for a robust detection against adversarial FDIEAs?*

C. Contributions

The aforementioned limitations motivate the goal of investigating the robustness of spatially-unaware, spatially-aware, spatio-temporal-aware, and adversarial-specific detectors against FDIA and adversarial FDIEAs in smart power grids in different attack settings. Toward this objective, this paper provides the following contributions:

- We introduce highly damaging adversarial FDIEAs using three dynamic attack functions that lead to higher degradation in DR by $5-12\%$ compared to static benchmark ones. The generated adversarial samples are injected using multiple levels and settings based on the adversary's knowledge about the adopted detection system, data (features), and grid topology. Knowledge levels include full, partial, and no knowledge reflecting white, gray, and black-box attack settings, respectively. We found out that adversarial FDIAEs generated utilizing spatio-temporal features lead to $5-26\%$ and $2-5\%$ higher degradation in DR compared to traditional FDIAs and utilizing only temporal features, respectively, in white-box settings.
- We propose a node selection strategy to increase the strength of evasion attacks in white-box settings based on the betweenness and degree centrality analysis. We found out that injecting attacks into nodes with high centrality leads to $3-11\%$ higher DR degradation compared to randomly selecting attack nodes.
- We propose a graph-generation process in gray and black-box settings to craft attack vectors where adversaries lack knowledge about the grid topology and data used by operators. The process involves generating surrogate spatio-temporal features based on a stochastic geometry approach where adversaries create an adversarial environment with realistic spatial data (graphs and connectivity) and conduct a power flow analysis to generate temporal data to mimic a realistic system. The crafted adversarial samples using the adversarial environment are then used to fool the operator's detector, leading to $3-13\%$ higher degradation in DR compared to traditional FDIAs.
- Our experiments revealed that adopting a spatio-temporal graph autoencoder (STGAE)-based detector offers a robust detection against adversarial FDIEAs. The STGAE model presents an unsupervised model (autoencoder with attention) that identifies unseen adversarial samples, a recurrent mechanism (LSTM cells) that models temporal aspects, and spatial layers (graph convolution) that capture topological configurations. Such characteristics lead to $5-53\%$ improved DR compared to spatially-unaware, spatially-aware graph, and spatio-temporal-aware non-graph benchmark detectors against adversarial FDIAEs.

The remainder of the paper is organized as follows. Section II describes the data preparation, threat model, and benchmark

detectors. Section III explores the impact of utilizing spatio-temporal features for generating adversarial samples in white-box settings and introduces the proposed attack node selection strategy. Section IV presents the stochastic geometry-based generative model to generate surrogate adversarial environments in gray and black-box settings. Section V describes the STGAE model, examines the robustness of the detectors, and presents the characteristics of the robust STGAE-based detector. Section VI presents the conclusions.

II. Preliminaries

This section presents the data preparation for the conducted experiments, threat model in terms of generating traditional FDIAs along with adversarial FDIEAs, and benchmark detectors.

A. Data Preparation

From a defense (system operator) perspective, building a data-driven attack detector requires comprehensive datasets with different system features for training, validation, and testing. The ideal approach would be utilizing huge amounts of real historical spatio-temporal datasets from real power systems of different sizes and configurations. However, such dynamic datasets are not readily available to use due to regional legal agreement policies and security reasons [40]. Therefore, as discussed in Section I-A, existing research has been utilizing IEEE test systems and computer software to simulate power flow to generate normal operation and attack data. Thus, getting realistic and reliable results requires performing three tasks. The first task is adopting realistic power systems, where we consider the IEEE 14, 39, and 118-bus systems. The second task is introducing dynamicity to such systems for generalization abilities, where we utilize an approach based on stochastic geometry [41] to build multiple topologies/graphs out of the IEEE 14, 39, and 118-bus systems with nodes and edges mimicking real power grids topological characteristics. The third task is obtaining temporal data, where we utilize Newton's method to generate power systems measurement data for each topology [27]. Details on the three tasks are described next.

1) IEEE Bus Systems: To perform the first task, we utilize IEEE standard bus systems that are widely adopted in power systems applications as seen in Section I-A. To examine the scalability of the detectors, we utilize three IEEE standard bus systems with small, medium, and large sizes, namely, the IEEE 14, 39, and 118-bus systems, respectively. The IEEE 14-bus represents an American Electric Power system with 14 buses, 5 generators, 11 loads, and 20 lines [42]. The IEEE 39-bus system represents the New-England Power System with 39 buses, 10 generators, 19 loads, and 46 lines. [43]. The IEEE 118-bus system represents a Midwest American Electric Power system with 118 buses, 19 generators, 91 loads, 35 synchronous condensers, 9 transformers, and 177 lines [44].

2) Spatial Data: To perform the second task, we utilize stochastic geometry, which is a powerful tool when it comes to considering the physical constraints of connecting the power elements as well as capturing the spatial coupling and correlations of electrical elements [5], [27], [45], [46]. The stochastic geometric morphogenesis of cities utilizing iterated Poisson tessellations matches the aforementioned IEEE 14,39, and 118-bus systems, presenting real world systems [47], which has been validated against real power grids and IEEE test systems [41]. In this work, the generative stochastic geometry approach serves two purposes; to construct multiple topological configurations to build generalized data-driven attack detectors (graph generation is discussed in Section IV-C1c), and to construct additional surrogate topological configurations for adversaries that do not have access to the operator data (discussed in Section IV-C). In both cases, ten different topological configurations are generated for 14, 39, and 118-bus systems. From a defense perspective, the investigated detectors are trained on several topological configurations of 14, 39, and 118-bus systems and are then tested against unseen topological configurations. The configurations are represented as $\mathbf{\Gamma} = [1, 2, ..., 10]$ for each system size, where training, testing, and validating topologies are selected according to a leave-one-out method [48]. Specifically, we perform eight separate generalized experiments for each system size. Each experiment utilizes seven, one, and two topologies presenting a training set $\boldsymbol{X}_{\text{TR}}$, validation set $\boldsymbol{X}_{\text{VAL}}$, and test set $\boldsymbol{X}_{\text{TST}}$, respectively. The average detection performance of the experiments is then reported against the remaining unseen topological configurations. The reason behind such an experimental setting is to achieve a generalization ability that captures the dynamic aspect of power grids where topological reconfigurations take place [5], [27].

3) Temporal Data: To perform the third task, we adopt a power flow analysis under steady-state conditions using Newton's method via MATLAB's MATPOWER toolbox [49] to determine the temporal features for the IEEE 14, 39, and 118 bus systems along with the constructed graphs/topologies of these systems [5]. The temporal features present time-series data in the form of active power P_i in megawatt (MW) and reactive power Q_i in megavolt amperes (MVAr). Specifically, the load data profile from the Electric Reliability Council of Texas (ERCOT) [50] is utilized [45], [51] and normalized with a zero mean and unit standard deviation scalar vector. We multiply the power values by a scaling sample from a normal distribution (i.e., $1+0.025*\digamma_t$ mean and 0.01 standard deviation) such that $\digamma_t$ denotes the scalar value at timestamp t. This way, we achieve a dynamic variation in the measurement values, resulting in a dynamic range of load values with respect to the properties of normal distribution. In this work, the power flow analysis is used to serve two purposes; to construct the benign samples to build generalized data-driven attack detectors, and to construct additional surrogate temporal features for adversaries that lack knowledge about the operator data (discussed in Section IV-C2). To build the data-driven attack detectors, for each topology, we include 4 power dynamics timestamps per hour, resulting in 96 daily power dynamics timestamps during a period of 180 days, yielding a total of around $17,000$ timestamps. The resulting data represent the benign samples (under normal operation) where a benign sample is expressed as $\boldsymbol{X}_{\text{b}}(t, i) \in \boldsymbol{\mathcal{X}}_{\text{b}}$ such that t and i present a given timestamp and bus, respectively,

This article has been accepted for publication in IEEE Transactions on Artificial Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAI.2024.3464511

TABLE I
TRADITIONAL FDIA FUNCTIONS

Attack	Attack function
Random attack	$f_1(\boldsymbol{X}_{\mathrm{m}}(t,i)) = \boldsymbol{X}_{\mathrm{b}}(t,i) + \alpha.\boldsymbol{X}_{\mathrm{b}}(t,i)$
General attack	$f_2(\boldsymbol{X}_{\mathrm{m}}(t,i)) = \boldsymbol{X}_{\mathrm{b}}(t,i) + (-1)^{\beta}\alpha.\gamma.\mathrm{Range}(\boldsymbol{X}_{\mathrm{b}}(t,i))$
One-step replay	$f_3(\boldsymbol{X}_{\mathrm{m}}(t,i)) = \boldsymbol{X}_{\mathrm{b}}(t-1,i)$
Random replay	$f_4(\boldsymbol{X}_{\mathrm{m}}(t,i)) = \boldsymbol{X}_{\mathrm{b}}(t-\hat{t},i)$
Interval replay	$f_5([\boldsymbol{X}_{\mathrm{m}}(t_n,i),..,\boldsymbol{X}_{\mathrm{m}}(t_m,i)]) = [\boldsymbol{X}_{\mathrm{b}}(t_{\hat{n}},i),..,\boldsymbol{X}_{\mathrm{b}}(t_{\hat{m}},i)]$
Target replay	$f_6([\boldsymbol{X}_{\mathrm{m}}(t_n,i),..,\boldsymbol{X}_{\mathrm{m}}(t_m,i)]) = [\boldsymbol{X}_{\mathrm{b}}(t_{\dot{n}},i),..,\boldsymbol{X}_{\mathrm{b}}(t_{\dot{m}},i)]$

TABLE II
ADVERSARIAL FDIEA FUNCTIONS

Attack	Attack function
FGSM	$f_7(\boldsymbol{X}_{\mathrm{a}}(t,i)) = \boldsymbol{X}_{\mathrm{b}}(t,i) + \varepsilon\ \mathrm{sign}\nabla_{\boldsymbol{X}_{\mathrm{b}}(t,i)} J(\phi, \boldsymbol{X}_{\mathrm{b}}(t,i), \boldsymbol{y})$
BIM	$f_8(\boldsymbol{X}_{\mathrm{a}}(t+1,i)) =$ $\mathrm{Clip}_{\boldsymbol{X}_{\mathrm{b}}(t,i),\bar{\varepsilon}}\{\boldsymbol{X}_{\mathrm{a}}(t,i) + \varepsilon\ \mathrm{sign}\nabla_{\boldsymbol{X}_{\mathrm{b}}(t,i)} J(\phi, \boldsymbol{X}_{\mathrm{a}}(t,i), \boldsymbol{y}\}$
C&W	$f_9(\boldsymbol{X}_{\mathrm{a}}(t,i)) = \min_{\varepsilon}\ \omega(\boldsymbol{X}_{\mathrm{b}}(t,i), \boldsymbol{X}_{\mathrm{b}}(t,i) + \varepsilon)$
DMP	$f_{10}(\boldsymbol{X}_{\mathrm{a}}(t+1,i)) =$ $\mathrm{Clip}_{\boldsymbol{X}_{\mathrm{b}}(t,i),k}\{\boldsymbol{X}_{\mathrm{a}}(t,i) + \varepsilon\ \mathrm{sign}\nabla_{\boldsymbol{X}_{\mathrm{b}}(t,i)} J(\phi, \boldsymbol{X}_{\mathrm{a}}(t,i), \boldsymbol{y}\}$
DMD	$f_{11}(\boldsymbol{X}_{\mathrm{a}}(t,i)) = \min_{\varepsilon}\ \omega(\boldsymbol{X}_{\mathrm{b}}(t,i), \varepsilon)$ where $\varepsilon = \bar{\varepsilon}\ \boldsymbol{X}_{\mathrm{b}}(t,i)$
DMA	$f_{12}(\boldsymbol{X}_{\mathrm{a}}(t,i)) = \min_{\varepsilon}\ \|\varepsilon\|_p + c \cdot f(\boldsymbol{X}_{\mathrm{b}}(t,i) + \varepsilon, \boldsymbol{y})$ where $\varepsilon = \bar{\varepsilon}\ \boldsymbol{X}_{\mathrm{b}}(t,i)$

and $\boldsymbol{\mathcal{X}}_{\mathrm{b}}$ denotes all benign samples. All samples $\boldsymbol{\mathcal{X}}_{\mathrm{b}}$ have correct benign labeling information of $y = 0$. Samples $\boldsymbol{\mathcal{X}}_{\mathrm{b}}$ are then manipulated using the attack functions described next to construct traditional FDIA and adversarial FDIEA samples.

B. Threat Model

We investigate the impact of two cyber threats on power grids, namely, traditional FDIAs and adversarial FDIEAs. We examine the robustness of these threats in five attack cases (discussed in II-B6). In traditional FDIAs, malicious entities manipulate the exchanged data with the aim of deceiving the grid's control systems and operations, resulting in inaccurate decision making and blackouts in severe cases. The limitation of traditional FDIAs is that they present simple attacks that may bypass traditional BDD systems, but may be spotted using machine learning-based detectors [27]. Adversarial FDIEAs present a more serious threat as adversaries inject adversarial samples with the aim of circumventing the grid's attack detection system itself since they present similar patterns to benign ones and hence fool the machine learning-based detector [6]. Thus, detecting adversarial FDIEAs is more challenging than traditional FDIAs. Next, we describe how both threat types may take place in real life along with the system vulnerability. We then describe the attack implementation and how we end up with the generated attack and adversarial samples.

1) Smart Grid Vulnerability: As cyber-physical systems, smart power grids are susceptible to cyberattacks due to the interconnected nature of the grid and reliance on communication networks. Since data communication takes place, attackers (whether they have full, partial, or no system knowledge), in a real life scenario, could monitor and intercept the data traffic using network traffic monitoring tools such as Wireshark, tcpdump, or Nmap. With such tools, attackers can identify vulnerabilities in network protocols or devices within the grid infrastructure. By analyzing captured traffic, attackers may discover weaknesses. The utilized vulnerabilities (i.e., system entry points) such as buffer overflows or authentication bypasses could be exploited to compromise the communication. After identifying the vulnerability, the attacker acts as a man-in-the-middle where intercepted signals are manipulated using the attacks described in Tables I and II, resulting in false data. Utilizing the identified vulnerability, the false data is then injected back into the system via tools such as Ettercap [52] that can facilitate the man-in-the-middle attack scenario.

2) Traditional FDIA Samples: The traditional FDIA samples are generated using the six FDIA functions presented in Table I. In such traditional attacks, attackers do not require any knowledge about the system, topology, or used detector where network traffic analysis software may be used to collect readings to manipulate and falsely inject them into the system. To capture such a case, we consider the following traditional random, general, and replay FDIA functions.

As shown in Table I, a malicious sample $\boldsymbol{X}_{\mathrm{m}}(t,i) \in \boldsymbol{\mathcal{X}}_{\mathrm{m}}$ is produced using traditional FDIA functions $(f_1(\cdot) - f_6(\cdot))$. In the random attack $(f_1(\cdot))$, a benign sample is randomly selected, where a bounded noise α with a magnitude of $-0.05 \leq \alpha \leq 0.05$ is applied to that sample. In the general attack $(f_2(\cdot))$, β and γ depict a binary random variable and uniform random variable between 0 and 1, respectively, where the Range term depicts the range of true measurements at timestamp t and bus i. The replay attacks falsely repeat benign samples from a previous timestamp or a series of timestamps. In the one-step replay attack $(f_3(\cdot))$, a sample from one previous timestamp $(t-1)$ is repeated, whereas a sample from a randomly selected previous timestamp $2 \leq \hat{t} \leq 5$ is repeated in the random replay attack $(f_4(\cdot))$. In the interval replay $(f_5(\cdot))$ and target replay $(f_6(\cdot))$ attacks, a series of benign readings $[\boldsymbol{X}_{\mathrm{b}}(t_n,i), \cdots, \boldsymbol{X}_{\mathrm{b}}(t_m,i)]$ from previous consecutive timestamps is repeated where the time interval length is randomly selected between 2 and 5 to present similarities in readings without being spotted by the detectors. In $f_5(\cdot)$, the repeated malicious samples are presented as $[\boldsymbol{X}_{\mathrm{b}}(t_{\hat{n}},i), \cdots, \boldsymbol{X}_{\mathrm{b}}(t_{\hat{m}},i)]$. In $f_6(\cdot)$, the repeated malicious samples are presented as $[\boldsymbol{X}_{\mathrm{b}}(t_{\dot{n}},i), \cdots, \boldsymbol{X}_{\mathrm{b}}(t_{\dot{m}},i)]$, where such samples must present measurements with higher/lower values than $[\boldsymbol{X}_{\mathrm{b}}(t_n,i), \cdots, \boldsymbol{X}_{\mathrm{b}}(t_m,i)]$ so that measurements with lower values get replaced with higher values from a series of previous consecutive timestamps and vice versa. Attack functions $f_5(\cdot)$ and $f_6(\cdot)$ present stronger attacks than $f_1(\cdot) - f_4(\cdot)$ since they repeat intervals of benign readings and hence introduce benign patterns in a stealthy manner.

3) Adversarial FDIEA Samples: Generating adversarial samples requires introducing sophisticated adversarial mea-

TABLE III
CASES OF EVASION ATTACK SETTINGS DEPENDING ON THE ADVERSARIES KNOWLEDGE ON THE DETECTION SYSTEM DETAILS

Setting	Case	Knowledge	Used detector	Used topology	Used features	Node selection
White-box	**Case 1**	Full	Same as opr.	Same as opr.	Temporal	Random
	Case 2	Full	Same as opr.	Same as opr.	Spatio-temporal	Random
	Case 3	Full	Same as opr.	Same as opr.	Spatio-temporal	Centrality analysis
Gray-box	**Case 4**	Partial	Same as opr.	Different from opr.	Spatio-temporal	Random
Black-box	**Case 5**	None	Different from opr.	Different from opr.	Spatio-temporal	Random

surement data in a stealthy way that fools the implemented detector to be falsely identified as benign. Hence, we apply static and dynamic adversarial FDIEA functions to mimic a system that encounters adversarial FDIEAs. As shown in Table II, an adversarial sample $\boldsymbol{X}_{\mathrm{a}}(t,i) \in \boldsymbol{\mathcal{X}}_{\mathrm{a}}$ is crafted using the adversarial FDIEA functions $f_7(\cdot) - f_{12}(\cdot)$ discussed next.

a) Static Evasion Attack Functions: Adversarial FDIEA functions $f_7(\cdot)$ - $f_9(\cdot)$ present benchmark static attacks since they generate one adversarial sample $\boldsymbol{X}_{\mathrm{a}}(t,i)$ based on one benign sample $\boldsymbol{X}_{\mathrm{b}}(t,i)$ utilizing bounded perturbations. The fast gradient sign method (FGSM) [37] attack function $f_7(\cdot)$ applies a perturbation value ε to a benign sample to fool the detector. Specifically, $f_7(\cdot)$ uses the model's loss function gradient with respect to $\boldsymbol{X}_{\mathrm{b}}(t,i)$ such that an adversarial sample $\boldsymbol{X}_{\mathrm{a}}(t,i)$ is created while maximizing the loss. The process of maximizing the model's loss is carried out through a one-step gradient update along the direction of the gradient's sign at a given t. In $f_7(\cdot)$, ε, sign, and $\boldsymbol{y}$ denote the perturbation magnitude, signum function, and true label, respectively, whereas ∇, J, and ϕ depict the model's gradient, loss function, and parameters, respectively. The basic iterative method (BIM) [53] attack function $f_8(\cdot)$ applies $f_7(\cdot)$ iteratively over several timestamps and uses the clip function to clip the obtained elements after each timestamp, which guarantees that $\boldsymbol{X}_{\mathrm{a}}(t,i)$ and $\boldsymbol{X}_{\mathrm{b}}(t,i)$ present similar patterns. In $f_8(\cdot)$, ε depicts a small perturbation value at each t. $\hat{\varepsilon} = 0.1$ is tuned and denotes the maximum perturbation magnitude to increase the chance of fooling the detector while maximizing the loss. The Carlini & Wagner (C&W) [54] attack function $f_9(\cdot)$ operates based on the minimization of the Euclidean distance (root-mean-square) ω between $\boldsymbol{X}_{\mathrm{b}}(t,i)$ and $\boldsymbol{X}_{\mathrm{b}}(t,i) - \varepsilon$ [6].

b) Dynamic Evasion Attack Functions: For a stronger evasion impact, adversarial FDIEA functions $f_{10}(\cdot)$ - $f_{12}(\cdot)$ present dynamic attack functions where ε varies for each adversarial sample $\boldsymbol{X}_{\mathrm{a}}(t,i)$ based on the dynamic mean $\bar{\varepsilon}$ of $\boldsymbol{X}_{\mathrm{b}}(t,i)$ and k neighboring readings. The dynamic mean perturbation (DMP) attack function $f_{10}(\cdot)$ applies $\bar{\varepsilon}$ such that ε varies for each reading and perturbation values are not bounded by $\hat{\varepsilon}$. Similarly, in the dynamic mean distance (DMD) attack function $f_{11}(\cdot)$, ω varies based on the mean of $\boldsymbol{X}_{\mathrm{b}}(t,i)$ and k neighboring readings. Therefore, ε also varies for each generated $\boldsymbol{X}_{\mathrm{a}}(t,i)$, where ω denotes the Euclidean distance between $\boldsymbol{X}_{\mathrm{b}}(t,i)$ and ε, where $\varepsilon = \bar{\varepsilon}\ \boldsymbol{X}_{\mathrm{b}}(t,i)$. In the dynamic mean elastic-net (DME) attack function $f_{12}(\cdot)$, we apply a modified version of the elastic-net attacks on deep neural networks [55], which aims to minimize the combined perturbation norm $\|\varepsilon\|$ of ε with respect to norm p that determines the magnitude of ε. In $f_{12}(\cdot)$, parameter c is used to control the trade-off between ε and the loss function value while scaling the regularization importance with respect to the loss term.

4) Attack Magnitude and Signals: The attack magnitude and number of attack signals rely on the values of ε and k, which are crucial when it comes to the stealthiness and detectability of the attacks. Determining ε and k vary based on the level of knowledge that adversaries have. When adversaries have access to the details of the operator's detector, we assume that adversaries will adopt a trial-and-error approach to find the largest ε and k values that fool the model. Thus, adversaries tend to apply smaller ε and k values and report the false negative rate (FNR) as ε and k increase. The largest ε and k that report the highest FNR are then applied, which turns out to be $0.4 \leq \varepsilon < 0.9$ (with 0.01 increments) and $2 \leq k \leq 16$. In cases where adversaries do not have knowledge about the adopted detector, they tend to maintain ε and k values below certain thresholds to avoid being detected, which turn out to be $\varepsilon = 0.65$ and $k = 2$.

5) Attack Levels: We examine the robustness of the detectors against a combination of traditional FDIAs and adversarial FDIEAs in multiple attack injection levels. To avoid biased results, half of the test samples are benign samples, whereas the other half is split equally among the attack functions in multiple injection levels. The first injection level contains 0% and 100% adversarial FDIEA and traditional FDIA samples, respectively. The second injection level contains 25% and 75% adversarial FDIEA and traditional FDIA samples, respectively. The third injection level contains 50% from each adversarial FDIEA and traditional FDIA. The fourth injection level contains 75% and 25% adversarial FDIEA and traditional FDIA samples, respectively. The fifth injection level contains 100% and 0% adversarial FDIEA and traditional FDIA samples, respectively.

6) Attack Cases: For a comprehensive analysis on the robustness of the investigated detectors, we consider several attack cases based on the level of knowledge an adversary has. Table III summarizes the attack cases where an adversary has full (white-box), partial (gray-box), and no (black-box) knowledge about the operator (opr.) system. Further details will be discussed in Sections III and IV.

C. Benchmark Detectors

We provide a comprehensive robustness analysis of detectors equipped with multiple characteristics presenting spatially-unaware, spatially-aware, spatio-temporal-aware, and adversarial-specific models with feedforward or temporal mechanisms employing supervised or unsupervised detection

with shallow, deep, or graph structures. Due to their nature, supervised models are trained on labeled benign and malicious samples, whereas unsupervised models are only trained on benign samples. All detector types are tested on benign, malicious, and adversarial samples. All sample types have equal number of samples. We adopt a sequential grid-search hyperparameter selection process [56] to achieve the best outcomes in terms of the offered DR against the validation set. The hyperparameters are selected from a list of possible values as follows. For the shallow models, the differencing degree and moving average parameters are selected from $\{0, 1, 2, 3\}$. The kernel, gamma, and regularization parameters are selected from {Linear, Sigmoid, RBF}, {scale, auto}, and $\{1, 10, 100\}$, respectively. For the deep and graph models, the number of layers and units is selected from $\mathcal{L} = \{2, 3, 4, 5, 6, 8\}$ and $\mathcal{U} = \{4, 8, 16, 32, 64\}$, respectively. The dropout rate, neighborhood order, optimizer, and activation function are selected from $\mathcal{D} = \{0, 0.2, 0.4\}$, $\mathcal{K} = \{2, 3, 4, 5\}$, $\mathcal{O} = \{$Adam, Adamax, SGD, RMSprop$\}$, and $\mathcal{A} = \{$Sigmoid, Tanh, ReLu, Elu$\}$, respectively. For unsupervised models, test samples are labeled using a detection threshold ψ (discussed in Section V-B).

1) Spatially-Unaware Benchmark Detection: The unsupervised autoregressive integrated moving average (ARIMA) [57] detector is based on a shallow temporal model that predicts future measurements using benign samples and detects attack samples based on deviations from such predictions with differencing degree and moving average optimal parameters of 1 and 0, respectively, and $\psi = 0.42$. The supervised SVM [7] detector is based on a shallow static model that separates benign from attack samples using a decision boundary with kernel, gamma, and regularization optimal parameters of Sigmoid, auto, and 1, respectively. The supervised FNN [11] detector is based on a deep static model that captures features using stacked feedforward layers with $L = 5$, $U = 32$, $D = 0$, $O =$ Adam, and $A =$ ReLu optimal parameters. The supervised RNN [14] detector is based on a deep temporal model that captures time-series dependencies using recurrent layers while holding past states with $L = 3$, $U = 32$, $D = 0.2$, $O =$ Adam, and $A =$ ReLu optimal parameters. The supervised CNN [15] detector is based on a deep model that employs convolutions to extract features from data with $L = 4$, $U = 32$, $K = 5$, $O =$ RMSprop, and $A =$ ReLu optimal parameters. The unsupervised SEL [28] detector is based on a three-stage model that handles data sequentially through autoencoder, temporal, and feedforward neural networks with $L = 8$, $U = 32$, $D = 0.2$, $O =$ SGD, and $A =$ Sigmoid optimal parameters and $\psi = 0.53$.

2) Spatially-Aware Benchmark Detection: Such detectors apprehend the spatial aspects of the power grid by employing graph-based approaches. The supervised CGNN [3] detector is based on a feedforward static GNN model that employs the graph convolution operation [58] with $L = 5$, $U = 16$, $K = 3$, $O =$ RMSprop, and $A =$ ReLu optimal parameters. The unsupervised GAE [5] detector is based on a graph model that employs the graph convolution operation as part of an autoencoder with $L = 6$, $U = 64$, $K = 5$, $O =$ Adam, and $A =$ ReLu optimal parameters.

3) Spatio-Temporal-Aware Benchmark Detection: Such detectors apprehend the spatial and temporal aspects of the power grid. We examine six spatio-temporal-aware benchmark detectors including supervised STFNN [24], STRNN [20], STCNN [23], and STGNN [21] models as well as unsupervised STAE [22] and STGAN [33] models using the aforementioned optimal hyperparameters.

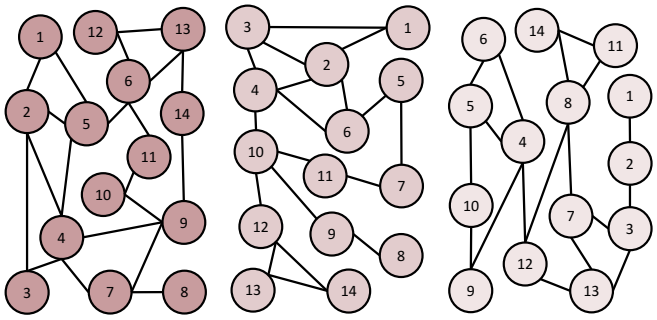
4) Adversarial-Specific Benchmark Detection: Such detectors are designed to detect adversarial samples using supervised learning. AT [35] introduces adversarial samples in the training stage to familiarize the model with such patterns. GAN-based detection [32], consisting of a generator and discriminator, generates data to fool a discriminator refining the generator's output.

5) Evaluation Metrics: We report how well the models detect attack samples using detection rate (DR = TP/(TP + FN)), where TP and FN denote the number of true positive and false negative samples, respectively. We also report the percentage of incorrectly marked benign samples as attack samples using false alarm rate (FAR = FP/(FP + TN)), where FP and TN stand for the number of false positive and true negative samples, respectively. The reported results are based on the average detection performance of generalized detectors built using the topological configurations of the 14, 39, and 118-bus systems and tested against levels of traditional FDIA and adversarial FDIEA samples on unseen topological configurations. This way, the generalization ability of the detectors against dynamic environments is also examined.

6) Model Complexity: The experiments are conducted offline on an NVIDIA GeForce RTX 4060 hardware accelerator using Python. Spatially-unaware, spatially-aware, and spatio-temporal-aware models take $3.5 - 5$, $7 - 7.5$, and $8 - 9$ hours, respectively, for offline training while adversarial-specific models require $1.5 - 2$ additional hours. The online (real-time) testing takes $2 - 5$ milliseconds to label a reading, which meets grid systems latency requirements. The size of spatially-unaware, spatially-aware, and spatio-temporal-aware models is roughly $10 - 35$, $70 - 120$, and $180 - 210$ megabytes, respectively. The additional time (i.e., training period) and space (i.e., model size) overhead of the robust STGAE model are associated with capturing more distinctive features from the data, requiring more training parameters. From the operator side, both aspects (time and space) do not pose further constraints since operators can perform offline training periodically and store the models at the control center.

III. Spatio-Temporal and Node Selection Attacks

This section answers the research question: *What is the impact of utilizing spatio-temporal features to craft adversarial samples and how to select attack nodes?* To address this question, we first introduce three attack cases in the white-box setting, where adversaries are considered insiders with full knowledge about the adopted detector, data, and topological configurations, which means that adversaries could utilize the IEEE 14, 39, and 118-bus test system configurations since they are publicly available. The rationale behind considering such attack cases is that information about the system could be



(a) IEEE 14-bus system topology (b) Sample benign topology (c) Sample surrogate topology

Fig. 1. Sample topological configurations of a 14-bus system.

accessed or leaked by insiders (i.e., system operators), which results in white-box attacks with full system knowledge. Thus, adversaries utilize the same detection type, model parameters, data, and topological configurations as the operator. We then propose an attack node selection strategy to achieve a higher evasion damaging impact. The white-box attack cases as described next.

A. White-Box Attack Cases

Case 1 (Temporal): In this attack case, although the adversaries have full knowledge about the temporal features and topological configurations, they opt to utilize the temporal features in the evasion attack vector. This case acts as a benchmark evasion attack mimicking existing studies that only adopt temporal features for adversarial samples generation. This means that adversarial samples crafted using temporal features in bus i are only injected into the same bus i. The adversaries herein utilize the same detection type and temporal data adopted by the operator to craft adversarial samples.

Case 2 (Spatio-Temporal): In this attack case, the adversaries have full knowledge about the temporal features and topological configurations. Besides utilizing the temporal features, they also include the spatial features in the evasion attack vector. Consequently, adversarial samples crafted using spatio-temporal features in bus i are injected into a randomly selected bus j. The adversaries herein utilize the same detection type and data as the operator to craft adversarial samples.

Case 3 (Centrality Analysis): In this attack case, the adversaries have full knowledge about the temporal features and topological configurations and utilize spatio-temporal features in the evasion attack vector. However, adversarial samples crafted using spatio-temporal features in bus i are injected into bus j that the adversaries strategically select. In particular, we propose an attack node selection strategy where an adversary selects nodes to attack based on the calculated betweenness centrality C_B, which measures the extent to which that node lies on the shortest paths between other pairs of nodes in the network [59]. The adversary's motivation in selecting such nodes is that nodes with higher C_B are considered the most influential hence vulnerable in the network. Thus, injecting attacks into such nodes would lead to higher performance deterioration. For example, in the graph representation of the IEEE 14-bus system illustrated in Fig. 1(a), nodes 4 and 5 are selected to be attacked since they present the highest C_B. The betweenness centrality $C_B(v)$ of node v is expressed as follows

$$C_B(v) = \sum_{i,j \in \mathcal{V}} \frac{\sigma(i,j|v)}{\sigma(i,j)}, \tag{1}$$

where $\sigma(i,j|v)$ presents the number of shortest paths among nodes i and j that pass through node v, and $\sigma(i,j)$ denotes the number of shortest paths between nodes i and j. If nodes have the same C_B, the adversary resolves this using the degree centrality C_D score, which quantifies the importance or influence of the node based on its degree [59]. The degree centrality $C_D(v)$ of node v is expressed as follows

$$C_D(v) = \frac{\deg(v)}{\max(\deg)}, \tag{2}$$

such that $\deg(v)$ depicts the number of edges connected to node v, whereas $\max(\deg)$ denotes the maximum degree reported of any given node in graph $\mathcal{G}$. The aforementioned node selection strategy is only applicable to the white-box setting where adversaries have full system access, and hence, have the capability to strategically select nodes to attack.

B. Impact of Adversarial FDIEAs in White-Box Settings

This section presents the detection performance of the benchmark detectors against static and dynamic adversarial FDIEAs in white-box settings. Table IV reports the average detection performance of the detectors against all investigated attack types and injection levels. On average, adversarial FDIAEs in white-box settings lead to higher degradation of $4.3-26.4\%$ in DR compared to traditional FDIAs since they present samples similar to benign ones and hence fool the detector. Next, we present the impact of adversarial FDIAEs generated using temporal features, spatio-temporal features, and centrality analysis-based node selection.

1) Impact of Utilizing Spatio-Temporal Features: In Table IV, we compare the performance degradation of the temporal and spatio-temporal white-box attack cases against the benchmark detectors. Specifically, higher degradation of $1.5-5.2\%$ in DR is reported when adversaries utilize spatio-temporal features (case 2) compared to utilizing temporal features (case 1) to craft adversarial samples using spatially-unaware detectors. Such a higher performance degradation is due to capturing the additional spatial aspect of the topological configurations and data adopted by the operator in generating adversarial samples that are then injected into a randomly selected node, compared to injecting them into the same node.

2) Impact of Attack Node Selection: In Table IV, we also compare the performance degradation of the spatio-temporal and centrality analysis-based node selection white-box attack cases against the detectors. Specifically, higher degradation of $3-6.1\%$ and $3.8-11.2\%$ in DR are reported when adversaries utilize spatio-temporal centrality analysis-based node selection (attack case 3) compared to random node selection using spatio-temporal features (case 2) and temporal features (case 1), respectively, to craft adversarial samples. Thus, injecting adversarial samples into the most influential nodes leads to

TABLE IV
IMPACT OF TRADITIONAL FDIAS AND ADVERSARIAL FDIEAS ON THE INVESTIGATED DETECTORS (%)

Detection type	Model type	Model	Metric	Traditional FDIAs	Adversarial FDIEA case setting				
					White-box			Gray-box	Black-box
					Case 1	Case 2	Case 3	Case 4	Case 5
Spatially-unaware	Non-graph	ARIMA	DR	68.6	53.5	48.3	42.3	55.8	57.7
			FA	40.3	54.0	59.2	65.2	51.5	49.6
		SVM	DR	71.6	57.1	51.9	46.0	59.4	61.4
			FA	32.3	44.5	49.6	55.4	42.4	40.6
		FNN	DR	76.0	62.3	57.2	51.4	65.2	66.9
			FA	25.5	37.7	42.7	48.6	34.6	33.0
		RNN	DR	81.3	68.0	63.2	57.5	70.7	72.2
			FA	19.4	32.2	37.0	42.7	28.9	28.2
		CNN	DR	84.9	72.5	67.9	62.3	75.8	77.4
			FA	13.8	24.8	29.4	35.1	22.0	20.1
		SEL	DR	89.4	78.2	73.7	68.1	80.6	82.3
			FA	9.1	19.3	23.6	29.3	19.6	16.5
Spatially-aware	Graph	GNN	DR	94.1	90.7	89.3	87.8	91.5	92.2
			FA	5.2	9.3	9.9	11.5	7.7	6.9
		GAE	DR	95.2	91.9	90.7	89.1	92.5	93.4
			FA	3.5	6.5	7.6	9.7	6.1	5.3
Spatio-temporal-aware	Non-graph	STNN	DR	90.5	83.8	83.3	82	85.1	85.5
			FA	12.4	18.9	19.8	20.6	17.8	17.1
		STRNN	DR	91.2	86	85.1	83.7	87	88
			FA	11.7	17.2	18.3	18.8	15.9	15.1
		STGAN	DR	91.7	87.4	86.8	84.4	88.2	88.5
			FA	10.3	14.4	15.8	17.9	13.8	13.3
		STCNN	DR	92.2	88.1	87.1	86	89	89.7
			FA	9.4	13.6	14.6	15.4	12.7	11.9
		STAE	DR	92.8	89.4	88.2	87.2	90.4	90.8
			FA	8.5	11.8	13	13.7	10.9	10.5
	Graph	STGNN	DR	94.7	91.1	89.9	88.2	91.9	92.8
			FA	4.8	9	9.5	11.1	7.2	6.7
		Robust STGAE	DR	97.5	**96.3**	**95.6**	**95.4**	**96.6**	**97.1**
			FA	1.9	**3.3**	**3.6**	**4.0**	**2.9**	**2.4**

the highest performance degradation since the impact of the adversarial sample becomes higher in the network, resulting in circumventing the detectors.

3) Remarks: This section answered the research question: *What is the impact of utilizing spatio-temporal features to craft adversarial samples and how to select attack nodes?*

- Utilizing spatio-temporal features in attack generation (cases 2&3) led to $4.5 - 26.4\%$ higher degradation in DR compared to traditional FDIAs since the generated samples present similar patterns to benign ones, which fool the detector.
- Utilizing spatio-temporal features in attack generation (case 2) led to $1.5 - 5.2\%$ higher degradation in DR compared to utilizing temporal features (case 1) due to capturing topological aspects to craft adversarial samples that are then injected into a randomly selected node.
- Strategically selecting attack nodes via centrality analysis (case 3) led to $3 - 11.2\%$ higher degradation in DR compared to a random selection (cases 1&2) due to injecting attacks into the most influential nodes (using C_B and C_D) in the topology.

IV. DESIGNING SPATIO-TEMPORAL SURROGATE DATA

This section answers the research question: *How can adversaries generate surrogate spatio-temporal data when they lack knowledge about the system topology?* To address this question, we first introduce gray and black-box attack cases. The rationale behind considering such attack cases is that partial information about the system could be accessed or leaked, which results in gray-box attacks carried out by either insiders or outsiders. A black-box attack case could take place by outsider adversaries that lack knowledge about the system information. We then propose utilizing a generative model based on stochastic geometry and Newton's method to construct surrogate adversarial data with realistic graphs to craft adversarial samples to be injected into real systems. Adversaries in gray and black-box attack cases could utilize the IEEE 14, 39, and 118-bus test system configurations since they are publicly available. The attack cases are described next.

A. Gray-Box Attack Case

The gray-box attack case assumes that adversaries have partial knowledge. This includes knowledge about the operator detector. Thus, adversaries use the same detection type as

the operator, but utilize surrogate datasets and topological configurations. The gray-box attack case is described next.

Case 4 (Spatio-Temporal): In this attack case, we assume that adversaries use the same detector and model parameters as the operator. However, they lack knowledge about the adopted data and topological configurations by the operator. Thus, we propose utilizing surrogate spatio-temporal datasets to create attacks on and generate adversarial samples in bus i in the adversarial environment, then injecting them into a randomly selected bus j in the real system since they lack knowledge about the real system's topology. The proposed generation process of the surrogate adversarial spatio-temporal datasets is discussed in Section IV-C.

B. Black-Box Attack Case

In the black-box attack case, we assume that adversaries lack knowledge about the adopted detector and topologies. Thus, adversaries in such a setting also resort to utilizing surrogate data and different detection schemes than the adopted ones to craft adversarial samples. Since adversaries in this case utilize a different detection type than the operator, the reported results herein present the average detection performance of all possible combinations of the investigated detectors. An example of a combination is that an adversary utilizes an SVM model while the operator adopts an FNN-based detector. The black-box attack case is described next.

Case 5 (Spatio-Temporal): In this attack case, we assume that the adversaries utilize different detector types and data than adopted by the operator. We also propose utilizing surrogate spatio-temporal datasets to craft adversarial samples while using different detection types than the operator.

C. Adversarial Surrogate Data

In attack cases 4 and 5, adversaries lack knowledge about the system's topological configuration, which makes it challenging to investigate the robustness of the detectors against such attack cases. Therefore, we propose an attack strategy where adversaries build surrogate datasets (adversarial environments) with adversarial spatial and temporal features. Adversarial spatial features are created using stochastic geometry to generate surrogate topological configurations that follow matching spatial distributions to real systems. Adversarial temporal features are simulated by performing a power flow analysis through Newton's method [49]. Specifically, we utilize ten surrogate topological configurations of 14, 39, and 118-bus systems to mimic the behavior of an adversary designing evasion attacks on datasets that are different from the ones adopted by the operators. The reason behind adopting a separate dataset for adversaries is that evasion attacks rely on benign samples and model parameters to be crafted, and in reality, adversaries might not have access to the adopted detectors, model parameters, topologies, or data. We capture the spatial aspects (i.e., spatial distribution of buses and connectivity data) along with the temporal aspects (i.e., power injections and flow data) of 14, 39, and 118-bus systems. To create a realistic adversarial environment, we generate multiple topological configurations of these systems using stochastic geometry [41]. In Fig. 1, the graph representations illustrated in (b) and (c) reflect two sample configurations derived from the IEEE 14-bus system illustrated in (a) using stochastic geometry. We resort to utilizing stochastic geometry since we require large-scale datasets with spatial and temporal features, which are not publicly available as discussed in Section II-A.

1) Adversarial Spatial Features: The stochastic geometry approach is adopted in gray and black-box settings to construct multiple topological configurations since it offers the following advantages. First, it effectively captures necessary physical constraints in interconnecting power elements [45]. Second, it incorporates spatial coupling and correlations of the electrical elements [46]. Such an approach of modeling cities using iterated Poisson structures exhibits a high degree of similarity to real-world systems [47]. Such an approach is validated against real power grids and IEEE test systems [41].

a) Generating an Adversarial Topological Configuration: Towards the objective of constructing a topological configuration for a given system (i.e., 14-bus system), we represent the geographical area by utilizing the Poisson line process with a disk radius R. The disk encompasses N lines formed as per the Poisson line density λ_n. A line Υ_n presents an angle direction θ_n and length υ_n subject to $0 \leq \theta_n < 2\pi$ and $0 \leq \upsilon_n < R$, respectively. $|\mathcal{V}|$ buses are also created by utilizing the process of one-dimensional homogeneous Poisson point (HPP) with density represented as $\lambda_{|\mathcal{V}|} = \sum_{\ell=1}^{N} \lambda_{|\mathcal{V}|_n}$. This process was proved to accurately capture the distribution of bus nodes in realistic power systems [45]. To connect the buses within the system, we use the near-geodesic route approach according to the physical paths along with the degree shifted sum of exponential distributions [45]. To make sure that power is delivered to the loads, buses are linked using the shortest pathways. Load capacities are assigned to buses via a probabilistic matching technique between the generated topology and an IEEE bus test system of equivalent size to ensure system similarity with real systems [60]. Line impedance values are allocated through the IEEE bus systems and New York independent system operator (NYISO) empirical data [61] that models impedance according to different system-based distributions. Then, the acquired values are statistically linked to real values of a same-size real or test system [27].

b) Generating Multiple Adversarial Topological Configurations: To generate such configurations, we reproduce the aforementioned processes using a constant $\lambda_{|\mathcal{V}|}$. This is carried out to ensure that the generated topological configurations present high similarities to real ones in terms of the spatial aspects (i.e., eigenvalue spread and nodal degree). In Fig. 1, the graph representations illustrated in (b) and (c) present high similarity to the IEEE 14-bus system illustrated in (a) in terms of spatial aspects to ensure the reliability of the conducted experiments, which allows generating multiple topological configurations of multiple system sizes. To mimic an adversary behavior that lacks knowledge about the grid's topology, we utilize such an approach to create an adversary (surrogate) environment that is used to craft adversarial samples to be injected into the real environment.

c) Adversarial Graph Structure: To model the power system using a graph $\mathcal{G}$, buses and power lines are represented

as nodes $\mathcal{V}$ and edges $\mathcal{E}$. An undirected graph is expressed as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{W})$, where $\boldsymbol{W} \in \mathbb{R}^{n \times n}$ denotes the weighted adjacency matrix (i.e., the line admittance). For instance, within a graph $\mathcal{G}$, when buses i and j are connected, a weight W_{ij} is associated to the edge $e = (i, j)$. Specifically, we construct 30 different configurations; ten for each system size (14, 39, and 118-bus systems) presenting the surrogate adversarial spatial features.

2) Adversarial Temporal Features: Nodes $\mathcal{V}$ of a given graph $\mathcal{G}$ of the constructed configurations are associated with time-series data (temporal features) where we generate active power P_i (MW) and reactive power Q_i (MVAr). Within each topology, the temporal features are simulated by performing a power flow analysis through Newton's method with the use of the MATLAB MATPOWER toolbox [49]. For each generated configuration, we include 4 power dynamics timestamps per hour, resulting in 96 daily timestamps during a period of 180 days. This yields around $17,000$ measurement timestamps presenting the surrogate adversarial temporal features.

D. Impact of FDIEAs in gray and black-Box Settings

This section reports the performance of benchmark detectors against adversarial FDIEAs crafted using surrogate spatio-temporal features built via the stochastic geometry model.

1) Impact of Utilizing Surrogate Environments in Gray-Box Settings: In Table IV, we report the impact of the gray-box attack setting for benchmark detectors. Using the stochastic geometry model to generate adversarial environments with surrogate spatio-temporal features leads to a higher degradation of $3.4 - 12.8\%$ in DR compared to traditional FDIAs. Such a higher degradation reflects the vulnerability of spatially-unaware detectors even in cases where adversaries utilize surrogate features to generate adversarial samples.

2) Impact of Utilizing Surrogate Environments in Black-Box Settings: In Table IV, we also report the impact of the black-box attack setting for benchmark detectors. Using the stochastic geometry model to generate adversarial environments with surrogate spatio-temporal features leads to a higher degradation of $2.8 - 11\%$ in DR compared to traditional FDIAs. This means that spatially-unaware detectors are still vulnerable to adversarial FDIEAs even in cases where adversaries lack knowledge about the system topology and detection type. This is because adversaries are able to generate adversarial environments with surrogate datasets that present high similarity to real systems using stochastic geometry.

3) Remarks: This section answered the research question: *How can adversaries generate surrogate spatio-temporal data when they lack knowledge about the system topology?*

- Adopting a generative spatio-temporal model using stochastic geometry led to generating adversarial samples that are capable of deceiving benchmark detectors.
- Generating adversarial samples using surrogate spatio-temporal features led to higher degradation of $2.8 - 12.8\%$ in DR compared to traditional FDIAs, reflecting the detectors' vulnerability, due to the high similarity present between surrogate adversarial and benign samples.

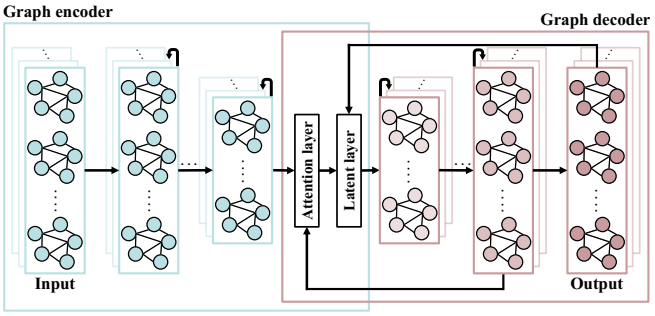


Fig. 2. Structure of the STGAE-based detector.

V. Spatio-Temporal Graph-Based Detection

This section answers the research question: *What are the required model characteristics for a robust detection against adversarial FDIEAs?* To address this question, we examine the robustness of a STGAE-based detector, that overcomes the limitations of benchmark detectors, against traditional FDIAs and adversarial FDIAEs.

A. Spatio-Temporal Graph Autoencoder-Based Detection

To capture the complex, spatial, and temporal aspects of smart power grids' data, we implement an STGAE-based detector that employs stacked graph convolutions to capture the spatial aspects as well as a recurrent mechanism to capture the temporal aspects. The STGAE model also employs an attention mechanism to focus on most relevant information during data processing. The detector also achieves unsupervised generalization abilities as it is trained on multiple graph representations of normal (benign) operation. The structure of the STGAE model is illustrated in Fig. 2 and expressed in Algorithm 1. The model adopts a graph-reconstruction process that learns the graph representations of benign data only. Overall, the model consists of an input ($\boldsymbol{\mathcal{X}}_{\mathrm{b}}$), a graph encoder ($\boldsymbol{E}$), an attention layer ($L_A$), a latent layer ($L_S$), a graph decoder ($\boldsymbol{D}$), and an output $\tilde{\boldsymbol{X}}$ as follows.

1) Input Block: The first stage in the STGAE model is the input stage. As inputs, the STGAE model takes the graph representations of benign samples $\boldsymbol{\mathcal{X}}_{\mathrm{b}}$ during normal operation. This includes the P_i and Q_i measurement values at a given node i expressed as $[P_i, Q_i] \in \mathbb{R}^{n \times 2}$.

2) Graph Encoder Block: The second stage is the graph encoder $\boldsymbol{E}$, which consists of stacked spatio-temporal graph Chebyshev encoding layers $\mathcal{L}_{\mathrm{E}}$, each with $c_{l_{\mathrm{E}}}$ channels. A spatio-temporal graph encoding layer $l_{\mathrm{E}} \in \mathcal{L}_{\mathrm{E}}$ outputs $\boldsymbol{X}^{l_{\mathrm{E}}} \in \mathbb{R}^{n \times c_{l_{\mathrm{E}}}}$ while taking the input of $\boldsymbol{X}^{l_{\mathrm{E}}-1} \in \mathbb{R}^{n \times c_{l_{\mathrm{E}}-1}}$. Afterwards, for a nonlinear capability, bias and ReLu activation are incorporated [62], resulting in an output tensor expressed in line 8 of Algorithm 1, where $\boldsymbol{\mu}^{l_{\mathrm{E}}} \in \mathbb{R}^{K \times c_{l_{\mathrm{E}}-1} \times c_l}$, $*_{\mathcal{G}}$, and $\boldsymbol{b}^{l_{\mathrm{E}}} \in \mathbb{R}^{c_{l_{\mathrm{E}}}}$ denote the Chebyshev coefficients (K order), graph convolution operator, and bias, respectively. The operator $*_{\mathcal{G}}$ handles the spatial aspects of the data. To handle the temporal aspect along with the issue of vanishing/exploding gradient with lengthy intervals, we implement long short-term memory (LSTM) units with $\boldsymbol{s}_t^{l_{\mathrm{E}}}$ and $\boldsymbol{h}_t^{l_{\mathrm{E}}}$ cell state and hidden state,

Algorithm 1: Training of the STGAE model.

1 **Input Block:** $\boldsymbol{X}_{\text{TR}}$
2 **Initialization:** Φ, $\boldsymbol{h}_{t-1}^{L_{\text{D}}}$, and $\tilde{\boldsymbol{X}}$
3 **for** *each topology* $\boldsymbol{\Gamma}$ **do**
4 **for** $\boldsymbol{X} \in \boldsymbol{X}_{\text{TR}}$ **do**
5 **Feed forward:**
6 **Graph Encoder Block ($\boldsymbol{E}$):**
7 **for** $l_{\text{E}} \in \mathcal{L}_{\text{E}}$ **do**
8 $\boldsymbol{X}^{l_{\text{E}}} = \text{ReLU}(\boldsymbol{\mu}^{l_{\text{E}}} *_{\mathcal{G}} \boldsymbol{X}^{l_{\text{E}}-1} + \boldsymbol{b}^{l_{\text{E}}})$
9 **for** $t \in T$ **do**
10 $\boldsymbol{i}_t^{l_{\text{E}}} = \varphi(\boldsymbol{W}_i^{l_{\text{E}}} \boldsymbol{X}_t^{l_{\text{E}}} + \boldsymbol{U}_i^{l_{\text{E}}} \boldsymbol{h}_{t-1}^{l_{\text{E}}} + \boldsymbol{V}_i^{l_{\text{E}}} \boldsymbol{s}_{t-1}^{l_{\text{E}}} + \boldsymbol{b}_i^{l_{\text{E}}})$
11 $\boldsymbol{o}_t^{l_{\text{E}}} = \varphi(\boldsymbol{W}_o^{l_{\text{E}}} \boldsymbol{X}_t^{l_{\text{E}}} + \boldsymbol{U}_o^{l_{\text{E}}} \boldsymbol{h}_{t-1}^{l_{\text{E}}} + \boldsymbol{V}_o^{l_{\text{E}}} \boldsymbol{s}_t^{l_{\text{E}}} + \boldsymbol{b}_o^{l_{\text{E}}})$
12 $\boldsymbol{f}_t^{l_{\text{E}}} = \varphi(\boldsymbol{W}_f^{l_{\text{E}}} \boldsymbol{X}_t^{l_{\text{E}}} + \boldsymbol{U}_f^{l_{\text{E}}} \boldsymbol{h}_{t-1}^{l_{\text{E}}} + \boldsymbol{V}_f^{l_{\text{E}}} \boldsymbol{s}_{t-1}^{l_{\text{E}}} + \boldsymbol{b}_f^{l_{\text{E}}})$
13 $\boldsymbol{s}_t^{l_{\text{E}}} = \boldsymbol{f}_t^{l_{\text{E}}} s_{t-1}^{l_{\text{E}}} + \boldsymbol{i}_t^{l_{\text{E}}} \tanh(\boldsymbol{W}_s^{l_{\text{E}}} \boldsymbol{X}_t^{l_{\text{E}}} + \boldsymbol{U}_s^{l_{\text{E}}} \boldsymbol{h}_{t-1}^{l_{\text{E}}} + \boldsymbol{b}_s^{l_{\text{E}}})$
14 $\boldsymbol{h}_t^{l_{\text{E}}} = \boldsymbol{o}_t^{l_{\text{E}}} \tanh(\boldsymbol{s}_t^{l_{\text{E}}})$.
15 **end**
16 **end**
17 **Attention Mechanism Block (L_{A}):**
18 $\boldsymbol{\kappa} = \xi(\boldsymbol{h}_t^{L_{\text{E}}}, \boldsymbol{h}_{t-1}^{L_{\text{D}}})$
19 $\boldsymbol{\Omega} = \frac{\exp(\boldsymbol{\kappa})}{\sum_{|\boldsymbol{\kappa}|} \exp(\boldsymbol{\kappa})}$
20 $\Lambda_t = \sum_T \Omega \times \boldsymbol{h}_t^{L_{\text{E}}}$
21 **Latent Block (L_{S}):**
22 $\breve{\boldsymbol{X}} = \sum(\Lambda_t, \tilde{\boldsymbol{X}})$
23 **Graph Decoder Block ($\boldsymbol{D}$):**
24 **for** $l_{\text{D}} \in \mathcal{L}_{\text{D}}$ **do**
25 $\breve{\boldsymbol{X}}^{l_{\text{D}}} = \text{ReLU}(\boldsymbol{\mu}^{l_{\text{D}}} *_{\mathcal{G}} \breve{\boldsymbol{X}}^{l_{\text{D}}-1} + \boldsymbol{b}^{l_{\text{D}}})$
26 **for** $t \in T$ **do**
27 Compute $\boldsymbol{i}_t^{l_{\text{D}}}$, $\boldsymbol{o}_t^{l_{\text{D}}}$, $\boldsymbol{f}_t^{l_{\text{D}}}$, $\boldsymbol{s}_t^{l_{\text{D}}}$, and $\boldsymbol{h}_t^{l_{\text{D}}}$
28 **end**
29 $\tilde{\boldsymbol{X}}^{l_{\text{D}}} = \text{ReLU}(\boldsymbol{\mu}^{l_{\text{D}}} *_{\mathcal{G}} \breve{\boldsymbol{X}}^{l_{\text{D}}-1} + \boldsymbol{b}^{l_{\text{D}}})$
30 **end**
31 **Back propagation:**
32 Compute $\min_{\Phi} C(\boldsymbol{X}, g_\Phi(f_\Phi(\boldsymbol{X})))$, $\nabla_{\boldsymbol{\mu}^{l_{(\cdot)}}} C$, $\nabla_{\boldsymbol{b}^{l_{(\cdot)}}} C$, $\nabla_{\boldsymbol{W}_{(\cdot)}^{l_{(\cdot)}}} C$, $\nabla_{\boldsymbol{U}_{(\cdot)}^{l_{(\cdot)}}} C$, and $\nabla_{\boldsymbol{V}_{(\cdot)}^{l_{(\cdot)}}} C$
33 **end**
34 **Parameter update:**
35 $\boldsymbol{\mu}^{l_{(\cdot)}} = \boldsymbol{\mu}^{l_{(\cdot)}} - \frac{\eta}{|\boldsymbol{X}_{\text{TR}}|} \sum_x \nabla_{\boldsymbol{\mu}^{l_{(\cdot)}}} C$
36 $\boldsymbol{b}^{l_{(\cdot)}} = \boldsymbol{b}^{l_{(\cdot)}} - \frac{\eta}{|\boldsymbol{X}_{\text{TR}}|} \sum_x \nabla_{\boldsymbol{b}^{l_{(\cdot)}}} C$
37 $\boldsymbol{W}_{(\cdot)}^{l_{(\cdot)}} = \boldsymbol{W}_{(\cdot)}^{l_{(\cdot)}} - \frac{\eta}{|\boldsymbol{X}_{\text{TR}}|} \sum_x \nabla_{\boldsymbol{W}_{(\cdot)}^{l_{(\cdot)}}} C$
38 $\boldsymbol{U}_{(\cdot)}^{l_{(\cdot)}} = \boldsymbol{U}_{(\cdot)}^{l_{(\cdot)}} - \frac{\eta}{|\boldsymbol{X}_{\text{TR}}|} \sum_x \nabla_{\boldsymbol{U}_{(\cdot)}^{l_{(\cdot)}}} C$
39 $\boldsymbol{V}_{(\cdot)}^{l_{(\cdot)}} = \boldsymbol{V}_{(\cdot)}^{l_{(\cdot)}} - \frac{\eta}{|\boldsymbol{X}_{\text{TR}}|} \sum_x \nabla_{\boldsymbol{V}_{(\cdot)}^{l_{(\cdot)}}} C$
40 **end**
41 **Output:** Optimal $\boldsymbol{\mu}^{l_{(\cdot)}}$, $\boldsymbol{b}^{l_{(\cdot)}}$, $\boldsymbol{W}_{(\cdot)}^{l_{(\cdot)}}$, $\boldsymbol{U}_{(\cdot)}^{l_{(\cdot)}}$, and $\boldsymbol{V}_{(\cdot)}^{l_{(\cdot)}}$

respectively, at timestamp t. Specifically, input $\boldsymbol{i}_t^{l_{\text{E}}}$, output $\boldsymbol{o}_t^{l_{\text{E}}}$, and forget $\boldsymbol{f}_t^{l_{\text{E}}}$ gates control the flow of information within the LSTM cells. The calculations of $\boldsymbol{i}_t^{l_{\text{E}}}$,$\boldsymbol{o}_t^{l_{\text{E}}}$, $\boldsymbol{f}_t^{l_{\text{E}}}$, $\boldsymbol{s}_t^{l_{\text{E}}}$, and $\boldsymbol{h}_t^{l_{\text{E}}}$ are expressed in lines 10 - 14 of Algorithm 1 with activation functions $\varphi(\cdot)$ and learnable weights $\boldsymbol{W}_{(\cdot)}$, $\boldsymbol{U}_{(\cdot)}$, and $\boldsymbol{V}_{(\cdot)}$.

3) Attention Mechanism Block: The third stage is an attention mechanism where we place an attention layer L_{A} to selectively focus on relevant timestamps within a sequence of measurements [19]. To achieve this, hidden states $\boldsymbol{h}_t^{L_{\text{E}}}$ and $\boldsymbol{h}_{t-1}^{L_{\text{D}}}$ of the last encoding and decoding layers, respectively, are passed to L_{A}. The calculations of the alignment score $\boldsymbol{\kappa}$, Softmax $\boldsymbol{\Omega}$, and context vector Λ_t of the attention mechanism are shown in lines 18 - 20 of Algorithm 1 with ξ depicting an alignment function that is trained jointly using $\boldsymbol{h}_t^{L_{\text{E}}}$ and $\boldsymbol{h}_{t-1}^{L_{\text{D}}}$. Then, the attention weight is expressed as a Softmax function applied to alignment scores [27].

4) Latent Block: The fourth stage represents the latent layer L_{S} that incorporates the compressed data from $\boldsymbol{D}$ with simpler representations to enhance the learning process. The concatenation $\breve{\boldsymbol{X}}$ depicted in line 22 of Algorithm 1 occurs with $\tilde{\boldsymbol{X}}$ denoting the reconstructed output from $\boldsymbol{D}$.

5) Graph Decoder Block: The fifth block presents the graph decoder $\boldsymbol{D}$ that is responsible for reconstructing the data via stacked spatio-temporal graph Chebyshev decoding layers $\mathcal{L}_{\text{D}}$, each with $c_{l_{\text{D}}}$ channels. The graph decoder takes $\breve{\boldsymbol{X}}$ as an input then produces $\tilde{\boldsymbol{X}}$. The flow of information is controlled using input $\boldsymbol{i}_t^{l_{\text{D}}}$, output $\boldsymbol{o}_t^{l_{\text{D}}}$, and forget $\boldsymbol{f}_t^{l_{\text{D}}}$ gates with $\boldsymbol{s}_t^{l_{\text{D}}}$ and $\boldsymbol{h}_t^{l_{\text{D}}}$ cell state and hidden state, respectively. A spatio-temporal graph decoding layer $l_{\text{D}} \in \mathcal{L}_{\text{D}}$ outputs $\breve{\boldsymbol{X}}^{l_{\text{D}}} \in \mathbb{R}^{n \times c_{l_{\text{D}}}}$ while taking the input of $\breve{\boldsymbol{X}}^{l_{\text{D}}-1} \in \mathbb{R}^{n \times c_{l_{\text{D}}-1}}$. Bias and ReLu activation are also incorporated, resulting in an output tensor expressed in line 25 of Algorithm 1.

6) Output Block: The sixth block is the reconstructed output $\tilde{\boldsymbol{X}}$ through the graph decoding layers, which is present at the output layer of the STGAE model. The formulation of $\tilde{\boldsymbol{X}}^{l_{\text{D}}}$ is expressed in line 29 of Algorithm 1.

B. Model Training and Testing

During training, the STGAE model learns the graph representations of the benign samples only. During testing, the model flags attack (traditional FDIA and adversarial FDIEA) samples based on the presented dissimilarities between their graph representations and benign ones. Assessing the dissimilarities is carried out according to the reconstruction error ζ presented during the graph autoencoder's reconstruction process. Let $\boldsymbol{E} = f_\Phi(\boldsymbol{X})$ and $\boldsymbol{D} = g_\Phi(\boldsymbol{X})$, with Φ denoting the model parameters. Then, we define a cost function C that penalizes $g_\Phi(f_\Phi(\boldsymbol{X})))$ for the introduced deviation from $\boldsymbol{X}$ as depicted in line 32 of Algorithm 1. The training objective is obtaining Φ (including $\boldsymbol{\mu}^{l_{(\cdot)}}$, $\boldsymbol{b}^{l_{(\cdot)}}$, $\boldsymbol{W}_{(\cdot)}$, $\boldsymbol{U}_{(\cdot)}$, and $\boldsymbol{V}_{(\cdot)}$) that optimize the cost function with $L = 6$, $U = 64$, $K = 5$, $O =$ Adam, and $A =$ ReLu optimal parameters. We adopt an iterative gradient descent algorithm employing a stochastic gradient to achieve such minimization. In lines 32 - 39 of Algorithm 1, we denote the partial derivative, learning rate, and number of training samples using ∇, η, and $|\boldsymbol{X}_{\text{TR}}|$, respectively. In the training stage, the training (benign) samples $\boldsymbol{X} \in \boldsymbol{X}_{\text{TR}}$ are divided into batches with equal sizes and fed into the model in epochs. In the testing stage, the decision whether a test sample $\boldsymbol{X} \in \boldsymbol{X}_{\text{TST}}$ is benign or malicious is made using a comparison among the reconstruction error ζ and a threshold ψ. A small ζ value reflects the model's familiarity with a test sample. We determine the value of ψ based on the median of the interquartile range of the receiver operating characteristic curve. Specifically, we use the cost function to determine ζ between $\boldsymbol{X}$ and $\tilde{\boldsymbol{X}}$. Then, if ζ exceeds ψ, an attack sample is detected and assigned a "$y = 1$" label. Otherwise, the sample is considered benign and assigned a "$y = 0$" label.

C. Model Robustness

This section analyzes the robustness of the STGAE-based detector compared to the investigated benchmark detectors against traditional FDIA and adversarial FDIEAs with multiple injection levels in five attack cases as follows.

1) Spatially-Aware Vs. Spatially-Unaware Benchmarks: Table IV reports the detection performance of the investigated spatially-aware graph-based detectors against adversarial FDIEAs. Spatially-aware graph-based detectors detectors enhance the DR by $12.6-46.9\%$, $10.9-36.7$, and $9.9-35.8\%$ in white, gray, and black-box settings, respectively, compared to spatially-unaware ones. Such an enhancement is because graph-based detectors are able to capture an additional aspect of the data, which is the topological configurations including the node connectivity. Hence, higher robustness is reported with graph-based models against adversarial FDIEAs even in white-box attack cases where adversaries have full knowledge and adopt the same detector type and topology as the operator.

2) Spatio-Temporal-Aware Vs. Spatially-Aware Benchmarks: Table IV also reports the detection performance of spatio-temporal-aware benchmark detectors compared to spatially-aware benchmark detectors. The spatially-aware graph-based benchmarks outperform the DR of the non-graph spatio-temporal-aware benchmarks by $0.6-8.1\%$, $1.1-7.4$, and $1.4-7.9\%$ in white, gray, and black-box settings, respectively. This is because the graph-based detectors implement the Chebyshev graph convolution operator that helps in apprehending the grid's topological configurations. The unsupervised spatially-aware GAE benchmark detector outperforms the supervised spatio-temporal-aware graph benchmark detector by around 1% in DR since it offers unsupervised detection that is not limited to predefined lists of attacks.

3) Classical Vs. Adversarial Detection: Table V compares classic detection types (classical machine learning) to adversarial-specific detectors (AT and GAN). AT and GAN enhance the DR by $1-4.4\%$ and $1.6-5.7\%$, respectively, compared to classical benchmark supervised training. Although AT and GAN enhance the detection compared to classical training, they introduce malicious samples during training, which limits the detection to attacks that are part of their training sets.

4) Dynamic Vs. Static Attacks: Table VI reports the DR of the best performing spatially-unaware (SEL), spatially-aware (GAE), spatio-temporal-aware (STGNN) benchmark detectors compared to the robust detector (STGAE) against the considered FDIAEs separately. The proposed dynamic evasion attacks lead to $5.2-12.2\%$ higher DR degradation compared to static ones. Such a degradation is because adversarial samples generated using dynamic attacks are designed using a series of benign samples with unbounded perturbation values, whereas adversarial samples generated using static attacks are designed using one benign sample with bounded perturbation values. Hence, dynamic attacks generate samples that present similar patterns to benign ones, making them harder to detect.

5) STGAE Vs. Benchmarks: Table IV shows that the STGAE-based detector offers the highest robustness against adversarial FDIEAs as it enhances the DR by $35.8-53.2\%$ and $14.8-44\%$ compared to shallow and deep spatially-unaware benchmarks, respectively. It also offers enhanced

TABLE V
IMPACT OF ADVERSARIAL-BASED DETECTION ON SUPERVISED MODELS COMPARED TO THE UNSUPERVISED ROBUST STGAE MODEL (%)

Attack case	Model	Detection type		
		Classic	AT	GAN
Case 1	FNN	62.3	65.9	67.1
	RNN	68.0	71.9	72.9
	CNN	72.5	76.6	77.7
	GNN	90.7	91.9	92.5
	STGAE	**96.3**	-	-
Case 2	FNN	57.2	60.3	61.5
	RNN	63.2	66.6	67.6
	CNN	67.9	71.5	71.6
	GNN	89.3	90.4	91.0
	STGAE	**95.6**	-	-
Case 3	FNN	51.4	54.3	55.6
	RNN	57.5	60.7	61.7
	CNN	62.3	65.8	65.9
	GNN	87.8	88.8	89.4
	STGAE	**95.4**	-	-
Case 4	FNN	65.2	69.0	70.3
	RNN	70.7	74.6	75.8
	CNN	75.8	80.0	81.2
	GNN	91.5	92.9	93.4
	STGAE	**96.6**	-	-
Case 5	FNN	66.9	70.9	72.3
	RNN	72.2	76.5	77.8
	CNN	77.4	81.8	83.1
	GNN	92.2	93.7	94.1
	STGAE	**97.1**	-	-

TABLE VI
IMPACT OF ADVERSARIAL FDIEAS ON THE DRS OF THE DETECTORS (%)

Adversarial FDIEA fucntion		Model	Adversarial FDIEA case setting				
			1	2	3	4	5
Static	FGSM	SEL	84.3	79.8	74.1	86.8	88.3
		GAE	94.9	93.7	92.1	95.0	95.4
		STGNN	94.1	92.9	91.2	94.3	94.3
		STGAE	**97.0**	**96.3**	**96.1**	**97.3**	**97.5**
	BIM	SEL	83.1	78.7	73.0	85.7	87.2
		GAE	94.4	93.2	91.6	94.8	95.2
		STGNN	93.6	92.4	90.7	94.1	94.1
		STGAE	**96.8**	**96.1**	**95.9**	**97.1**	**97.5**
	C&W	SEL	81.9	77.6	71.9	84.6	86.1
		GAE	93.9	92.7	91.1	94.5	94.9
		STGNN	93.1	91.9	90.2	93.8	93.8
		STGAE	**96.6**	**95.9**	**95.7**	**96.9**	**97.4**
Dynamic	DMP	SEL	74.4	69.9	64.2	76.8	78.6
		GAE	88.4	87.2	85.6	89.0	90.9
		STGNN	87.6	86.4	84.7	88.3	89.8
		STGAE	**96.0**	**95.3**	**95.1**	**96.3**	**96.8**
	DMD	SEL	73.2	68.8	63.1	75.6	77.4
		GAE	88.2	87.0	85.4	88.8	90.7
		STGNN	87.4	86.2	84.5	88.1	89.6
		STGAE	**95.8**	**95.1**	**94.9**	**96.1**	**96.6**
	DMA	SEL	72.0	67.6	62.0	74.5	76.3
		GAE	87.9	86.7	85.1	88.5	90.4
		STGNN	87.1	85.9	84.2	87.8	89.3
		STGAE	**95.6**	**94.9**	**94.7**	**95.9**	**96.4**

DR by $4.7-7.6\%$ compared to spatially-aware graph-based benchmarks due to its robust graph recurrent structure that captures spatial and temporal features. The STGAE-based detector also enhances the DR by $4.8-13.4\%$ compared to spatio-temporal-aware benchmarks since it implements an unsupervised recurrent graph structure that captures the topological configurations besides capturing the temporal correlations within measurement data. Table V shows the superiority of the unsupervised STGAE-based detection as it offers improved DR by $3-41.1\%$ compared to other models with AT and GAN-based detection. This is due to its deep graph unsupervised AE structure that distinguishes between benign and attack samples by learning benign patterns well during training without the need of including attack samples in the training set.

D. Advantages of the STGAE Model

The aforementioned performance enhancements of the STAGE model are due to the characteristics that the STGAE model offers. First, it presents an autoencoder that learns the representations of benign samples during the reconstruction process while assigning higher importance to higher timestamps using the attention layer. Second, the recurrent nature of the model allows it to capture the temporal features. Third, the graph convolution layers allows it to capture the spatial features. Fourth, due to its generalized unsupervised training nature, the STGAE leads to a robust generalized detection of unseen attack samples in unseen topological configurations. All of these characteristics resulted in a robust detector against adversarial FDIEAs regardless of the level of knowledge that adversaries have. A major advantage of the STGAE-based detector is that it is not designed specifically to detect adversarial FDIEAs only. Instead, its unsupervised nature, being trained only on benign samples, makes it robust against other attack types as well since it makes decisions based on the reconstruction error. Thus, unseen attack samples, regardless of their type, will present high reconstruction error and hence will be marked as attack samples.

E. Remarks

This section answered the research question: *What are the required model characteristics for a robust detection against adversarial FDIEAs?*

- Adopting spatially-aware detectors improved the DR by $9.9-46.9\%$ and $0.6-8.1\%$ compared to spatially-unaware and spatio-temporal-aware non-graph ones, respectively, as they capture more data aspects (i.e., topological configurations including the node connectivity) via the Chebyshev graph convolution operator.
- Adopting the STGAE-based detector led to a superior robust detection against adversarial FDIEAs. The required model characteristics to achieve a robust detection are: unsupervised training nature (autoencoder with attention) to capture unseen adversarial samples, recurrent mechanism (LSTM cells) to capture the temporal aspect, and spatial layers (graph convolution) to capture the topological configurations. Incorporating these characteristics led to an improved DR by $4.7-53.2\%$ compared to benchmark detectors.

VI. Conclusions and Future Work

This paper provided a comprehensive analysis on the impact of traditional FDIAs and adversarial FDIEAs on various detectors with different attack cases based on adversarial knowledge levels. Our experiments showed that adversarial FDIEA cases 1, 2, 3, 4, and 5 led to degradation in DR by $3.7-15.2\%$, $4.5-20.3\%$, $7.5-26.4\%$, $3.4-12.8\%$, and $2.8-11\%$, respectively, compared to traditional FDIAs, where we reached the following conclusions. First, adversarial FDIEAs led to higher degradation in DR than traditional FDIAs by $2.8-26.4\%$ due to presenting samples with similar patterns as benign ones that fool the detectors. Second, with adversarial full knowledge, adversarial samples generation using spatio-temporal features leads to around $1.5-5.2\%$ higher degradation in DR compared to utilizing temporal features only due to capturing the grid's topology and hence producing similar samples to real ones, which circumvent the detector. Third, centrality analysis-based attack node selection leads to $3-11.2\%$ higher degradation in DR compared to a random selection due to selecting the most influential (vulnerable) node in the system, leading to a more damaging impact. Fourth, when adversaries lack knowledge about the system topology, using stochastic geometry to create surrogate adversarial topologies led to $2.8-12.8\%$ higher degradation in DR compared to traditional FDIAs. This means that benchmark detectors are vulnerable to FDIEAs even when adversaries use surrogate environments to generate adversarial samples. Fifth, adopting an unsupervised spatio-temporal graph autoencoder (STGAE)-based detector enhances the DR by $4.7-53.2\%$ compared to benchmark detectors due to the presence of an autoencoder with attention, LSTM cells, and graph convolution that capture the benign patterns, temporal features, and spatial topological aspects. Future work will focus on joint detection and localization of adversarial samples.

References

[1] Z. Zhang *et al.*, "Cyber-physical coordinated risk mitigation in smart grids based on attack-defense game," *IEEE Trans. on Power Systems*, vol. 37, no. 1, pp. 530–542, Jan. 2022.

[2] Y. Li *et al.*, "Detection of false data injection attacks in smart grid: A secure federated deep learning approach," *IEEE Trans. on Smart Grid*, vol. 13, no. 6, pp. 4862–4872, Nov. 2022.

[3] O. Boyaci *et al.*, "Graph neural networks based detection of stealth false data injection attacks in smart grids," *IEEE Systems Journal*, vol. 16, no. 2, pp. 2946–2957, Jun. 2022.

[4] K. Hamedani *et al.*, "Detecting dynamic attacks in smart grids using reservoir computing: A spiking delayed feedback reservoir based approach," *IEEE Trans. on Emerging Topics in Computational Intelligence*, vol. 4, no. 3, pp. 253–264, Jun. 2020.

[5] 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. on Emerging Topics in Computational Intelligence*, vol. 7, no. 3, pp. 618–630, Jun. 2023.

[6] A. Takiddin, M. Ismail, and E. Serpedin, "Robust data-driven detection of electricity theft adversarial evasion attacks in smart grids," *IEEE Trans. on Smart Grid*, vol. 14, no. 1, pp. 663–676, Jan. 2023.

[7] M. Esmalifalak *et al.*, "Detecting stealthy false data injection using machine learning in smart grid," *IEEE Systems Journal*, vol. 11, no. 3, pp. 1644–1652, Sept. 2017.

[8] X. Lu *et al.*, "False data injection attack location detection based on classification method in smart grid," in *Int. Conf. on AI and Advc Manfct. (AIAM)*. Manchester, United Kingdom, 15–17 Oct. 2020, pp. 133–136.

[9] D. Wang *et al.*, "Detection of power grid disturbances and cyber-attacks based on machine learning," *Journal of Information Security and Applications*, vol. 46, pp. 42–52, Jun. 2019.

[10] A. S. Musleh *et al.*, "A survey on the detection algorithms for false data injection attacks in smart grids," *IEEE Trans. on Smart Grid*, vol. 11, no. 3, pp. 2218–2234, May 2020.

[11] D. Xue *et al.*, "Detection of false data injection attacks in smart grid utilizing ELM-Based OCON framework," *IEEE Access*, vol. 7, pp. 31 762–31 773, Mar. 2019.

[12] E. M. Ferragut *et al.*, "Real-time cyber-physical false data attack detection in smart grids using neural networks," in *Int. Conf. on Comp. Sci. and Com. Intel (CSCI)*. Las Vegas, NV, USA, 14–16 Dec. 2017.

[13] Y. Zhang *et al.*, "Detecting false data injection attacks in smart grids: A semi-supervised deep learning approach," *IEEE Trans. on Smart Grid*, vol. 12, no. 1, pp. 623–634, Jan. 2021.

[14] Y. Wang *et al.*, "Kfrnn: An effective false data injection attack detection in smart grid based on kalman filter and recurrent neural network," *IEEE Internet of Things Journal*, vol. 9, no. 9, pp. 6893–6904, May 2022.

[15] S. Wang *et al.*, "Locational detection of the false data injection attack in a smart grid: A multilabel classification approach," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8218–8227, Sept. 2020.

[16] F. Xia *et al.*, "Graph learning: A survey," *IEEE Trans. on Artificial Intelligence*, vol. 2, no. 2, pp. 109–127, Apr. 2021.

[17] R. Ramakrishna *et al.*, "Detection of false data injection attack using graph signal processing for the power grid," in *IEEE Glbl. Conf. on Sgnl. and Info. Proc. (GSIP)*. Ottawa, ON, Canada, 11–14 Nov. 2019.

[18] E. Drayer and T. Routtenberg, "Detection of false data injection attacks in smart grids based on graph signal processing," *IEEE Systems Journal*, vol. 14, no. 2, pp. 1886–1896, Jun. 2020.

[19] A. Takiddin, M. Ismail, U. Zafar, and E. Serpedin, "Deep autoencoder-based anomaly detection of electricity theft cyberattacks in smart grids," *IEEE Systems Journal*, vol. 16, no. 3, pp. 4106–4117, Sept. 2022.

[20] J. J. Q. Yu *et al.*, "Online false data injection attack detection with wavelet transform and deep neural networks," *IEEE Trans. on Industrial Informatics*, vol. 14, no. 7, pp. 3271–3280, Jul. 2018.

[21] A. Ahmed *et al.*, "Spatio-temporal deep graph network for event detection, localization, and classification in cyber-physical electric distribution system," *IEEE Trans. on Industrial Informatics*, vol. 20, no. 2, pp. 2397–2407, Feb. 2024.

[22] A. S. Musleh *et al.*, "Attack detection in automatic generation control systems using lstm-based stacked autoencoders," *IEEE Trans. on Industrial Informatics*, vol. 19, no. 1, pp. 153–165, Jan. 2023.

[23] G. Zhang *et al.*, "Spatio-temporal correlation-based false data injection attack detection using deep convolutional neural network," *IEEE Trans. on Smart Grid*, vol. 13, no. 1, pp. 750–761, Jan. 2022.

[24] X. Yin *et al.*, "A subgrid-oriented privacy-preserving microservice framework based on deep neural network for false data injection attack detection in smart grids," *IEEE Trans. on Industrial Informatics*, vol. 18, no. 3, pp. 1957–1967, Mar. 2022.

[25] S. Peng *et al.*, "Localizing false data injection attacks in smart grid: A spectrum-based neural network approach," *IEEE Trans. on Smart Grid*, vol. 14, no. 6, pp. 4827–4838, Nov. 2023.

[26] H. Li *et al.*, "End-edge-cloud collaboration-based false data injection attack detection in distribution networks," *IEEE Trans. on Industrial Informatics*, vol. 20, no. 2, pp. 1786–1797, Feb. 2024.

[27] A. Takiddin, M. Ismail, R. Atat, K. R. Davis, and E. Serpedin, "Robust graph autoencoder-based detection of false data injection attacks against data poisoning in smart grids," *IEEE Trans. on Artificial Intelligence*, vol. 5, no. 3, pp. 1287–1301, Mar. 2024.

[28] A. Takiddin, M. Ismail, U. Zafar, and E. Serpedin, "Robust electricity theft detection against data poisoning attacks in smart grids," *IEEE Trans. on Smart Grid*, vol. 12, no. 3, pp. 2675–2684, May 2021.

[29] H. Rao *et al.*, "Adversarial example attack on electric power network security situation awareness," in *Info. Tech. Network Elec. and Auto. Control Conf. (ITNEC)*. Xi'an, China, Nov. 2021, pp. 1394–1398.

[30] J. Tian *et al.*, "Joint adversarial example and false data injection attacks for state estimation in power systems," *IEEE Trans. on Cybernetics*, pp. 1–15, Nov. 2021.

[31] H. Rao *et al.*, "Adversarial examples in deep learning for multivariate time series regression," in *IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*. Washington, DC, USA, Oct. 2020, pp. 1–10.

[32] Y. Zhang *et al.*, "Detecting false data injection attacks in smart grids: A semi-supervised deep learning approach," *IEEE Trans. on Smart Grid*, vol. 12, no. 1, pp. 623–634, Jan. 2021.

[33] Z. Liu *et al.*, "Tscw-gan based fdias defense for state-of-charge estimation of battery energy storage systems in smart distribution networks," *IEEE Trans. on Industrial Informatics*, pp. 1–12, Nov. 2023.

[34] J. Tian *et al.*, "Exploring targeted and stealthy false data injection attacks via adversarial machine learning," *IEEE Internet of Things Journal*, vol. 9, no. 15, pp. 14 116–14 125, Aug. 2022.

[35] A. H. Bondok *et al.*, "Novel evasion attacks against adversarial training defense for smart grid federated learning," *IEEE Access*, vol. 11, pp. 112 953–112 972, May 2023.

[36] A. Takiddin, M. Ismail, and E. Serpedin, "Detection of electricity theft false data injection attacks in smart grids," in *Europ. Signal Proc. Conf. (EUSIPCO)*. Belgrade, Serbia, 29 Aug.–2 Sept. 2022, pp. 1541–1545.

[37] I. J. Goodfellow *et al.*, "Explaining and harnessing adversarial examples," in *Int. C. Lrn. Rep. (ICLR)*. San Diego, CA, USA, 7 May. 2015.

[38] A. Raghunathan *et al.*, "Certified defenses against adversarial examples," *arXiv preprint arXiv:1801.09344v2*, Oct. 2020.

[39] D. Meng and H. Chen, "Magnet: A two-pronged defense against adversarial examples," in *ACM Conf. Comput. Commun. Secur. (CCS)*. Dallas, TX, USA, Oct. 2017, pp. 135–147.

[40] F. Baouche *et al.*, "Efficient allocation of electric vehicles charging stations: Optimization model and application to a dense urban network," *IEEE Intel. Transportation Sys. Mag.*, vol. 6, no. 3, pp. 33–43, Jul. 2014.

[41] R. Atat *et al.*, "Stochastic geometry-based model for dynamic allocation of metering equipment in spatio-temporal expanding power grids," *IEEE Trans. on Smart Grid*, vol. 11, no. 3, pp. 2080–2091, May. 2020.

[42] R. Christie, "Power systems test case archive. 14 bus power flow test case," *Uni. of Washington, [Online] Available at http://www.ee.washington.edu/research/pstca/pf14/pg_tca14bus.htm*, Feb. 1962.

[43] T. Athay *et al.*, "A practical method for the direct analysis of transient stability," *IEEE Trans. on Power Apparatus and Systems*, vol. PAS-98, no. 2, pp. 573–584, Mar. 1979.

[44] R. Christie, "Power systems test case archive: 118 bus power flow test case," *Uni. of Washington [Online], Available: http://labs.ece.uw.edu/pstca/pf118/pg_tca118bus.htm*, Dec. 1962.

[45] D. Deka *et al.*, "Analytical models for power networks: The case of the western u.s. and ercot grids," *IEEE Trans. on Smart Grid*, vol. 8, no. 6, pp. 2794–2802, Nov. 2017.

[46] G. Rolim *et al.*, "Modelling the data aggregator positioning problem in smart grids," in *IEEE Int. Conf. on Comp. and Info. Tech.* Liverpool, UK, 26–28 Oct. 2015, pp. 632–639.

[47] T. Courtat *et al.*, "Mathematics and morphogenesis of cities: A geometrical approach," *Physical Review E*, vol. 83, p. 036106, Mar. 2011.

[48] C. Sammut and G. I. Webb, Eds., *Encyclopedia of Machine Learning*. Boston, MA: Springer US, 2011, pp. 600–601.

[49] R. D. Zimmerman *et al.*, "Matpower: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Trans. on Power Systems*, vol. 26, no. 1, pp. 12–19, Feb. 2011.

[50] "The Electric Reliability Council of Texas (ERCOT). Backcasted (actual) load profiles-historical." [Online]. Available: https://www.ercot.com/mktinfo/loadprofile/alp

[51] W. Xia *et al.*, "On the pmu placement optimization for the detection of false data injection attacks," *IEEE Systems Journal*, vol. 17, no. 3, pp. 3794–3797, Sept. 2023.

[52] Y. Li and J. Yan, "Cybersecurity of smart inverters in the smart grid: A survey," *IEEE Trans. on Pwr Elec.*, vol. 38, pp. 2364–2383, Feb. 2023.

[53] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *Int. Conf. on Learning Rep. (ICLR)*. Toulon, France, 24–24 Apr. 2017, pp. 1–14.

[54] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *IEEE symposium security and privacy*, 2017, pp. 39–57.

[55] Y. Wang *et al.*, "New adversarial image detection based on sentiment analysis," *IEEE Trans. on Nrl. Nets. and Lrn. Sys.*, pp. 1–15, May 2023.

[56] A. Takiddin *et al.*, "Detecting electricity theft cyber-attacks in AMI networks using deep vector embeddings," *IEEE Systems Journal*, vol. 15, no. 3, pp. 4189–4198, Sept. 2021.

[57] V. Krishna *et al.*, "ARIMA-Based modeling and validation of consumption readings in power grids," in *Critical Information Infrastructures Security*. Springer Intern. Publishing, May 2016, pp. 199–210.

[58] J. Zhou *et al.*, "Sparsity-induced graph convolutional network for semisupervised learning," *IEEE Trans. on Artificial Intelligence*, vol. 2, no. 6, pp. 549–563, Dec. 2021.

[59] J. Zhang and Y. Luo, "Degree centrality, betweenness centrality, and closeness centrality in social network," in *Int. Conf. on Modelling, Simulation and Applied Mathematics (MSAM)*, Mar. 2017, pp. 300–303.

[60] S. H. Elyas and Z. Wang, "Improved synthetic power grid modeling with correlated bus type assignments," *IEEE Trans. on Power Systems*, vol. 32, no. 5, pp. 3391–3402, Sept. 2017.

[61] Z. Wang *et al.*, "Generating statistically correct random topologies for testing smart grid communication and control networks," *IEEE Trans. on Smart Grid*, vol. 1, no. 1, pp. 28–39, June 2010.

[62] L. Ruiz *et al.*, "Invariance-preserving localized activation functions for graph neural networks," *IEEE Trans. on Signal Processing*, vol. 68, pp. 127–141, Nov. 2020.