Joint Detection and Localization of Stealth False Data Injection Attacks in Smart Grids Using Graph Neural Networks

Osman Boyaci, *Graduate Student Member, IEEE*, Mohammad Rasoul Narimani, *Member, IEEE*, Katherine R. Davis, *Senior Member, IEEE*, Muhammad Ismail, *Senior Member, IEEE*, Thomas J. Overbye, *Fellow, IEEE*, and Erchin Serpedin, *Fellow, IEEE*

***Abstract*—False data injection attacks (FDIA) are a main category of cyber-attacks threatening the security of power systems. Contrary to the detection of these attacks, less attention has been paid to identifying the attacked units of the grid. To this end, this work jointly studies detecting and localizing the stealth FDIA in power grids. Exploiting the inherent graph topology of power systems as well as the spatial correlations of measurement data, this paper proposes an approach based on the graph neural network (GNN) to identify the presence and location of the FDIA. The proposed approach leverages the auto-regressive moving average (ARMA) type graph filters (GFs) which can better adapt to sharp changes in the spectral domain due to their rational type filter composition compared to the polynomial type GFs such as Chebyshev. To the best of our knowledge, this is the first work based on GNN that automatically detects and localizes FDIA in power systems. Extensive simulations and visualizations show that the proposed approach outperforms the available methods in both detection and localization of FDIA for different IEEE test systems. Thus, the targeted areas can be identified and preventive actions can be taken before the attack impacts the grid.**

***Index Terms*—False data injection attacks, graph neural networks, machine learning, smart grid, power system security.**

NOMENCLATURE

$P_i + jQ_i$	Complex power injection at bus i.
$P_{ij} + jQ_{ij}$	Complex power flow between bus i and j.
V_i, θ_i	Voltage magnitude and phase angle of bus i.
n, m	Number of buses, number of measurements.
$\mathcal{X} \in \mathbb{R}^n$	State space.
$\mathcal{Z} \in \mathbb{R}^m$	Measurement space.
$\boldsymbol{x} \in \mathcal{X}$	A state vector.
$\hat{\boldsymbol{x}} \in \mathcal{X}$	Original state vector without an attack.
$\check{\boldsymbol{x}} \in \mathcal{X}$	False data injected state vector.
$\boldsymbol{z} \in \mathcal{Z}$	A measurement vector.
$\boldsymbol{z_o} \in \mathcal{Z}$	Original measurement vector.
$\boldsymbol{z_a} \in \mathcal{Z}$	Attacked measurement vector.
$\boldsymbol{a} \in \mathcal{Z}$	Attack vector.
$h(\boldsymbol{x})$	Nonlinear measurement function at $\boldsymbol{x}$.
$\mathcal{T}$	Attacker's target area to perform FDI attack.
$\boldsymbol{W} \in R^{n\times n}$	Weighted adjacency matrix.
$\boldsymbol{D} \in R^{n\times n}$	$\boldsymbol{D}_{ii} = \sum_j \boldsymbol{W}_{ij}$ Diagonal degree matrix.
$\boldsymbol{\Lambda} \in \mathbb{R}^{n\times n}$	$= \text{diag}[\lambda_1, \ldots, \lambda_n]$ Graph Fourier frequencies.
$\boldsymbol{U} \in \mathbb{R}^{n\times n}$	$= [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n]$ Graph Fourier basis.
$\boldsymbol{L} \in \mathbb{R}^{n\times n}$	$= \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T$ Normalized graph Laplacian.

Manuscript received May 23, 2021; revised August 27, 2021 and September 30, 2021; accepted October 2, 2021. Date of publication October 5, 2021; date of current version December 23, 2021. This work was supported by NSF under Award 1808064. Paper no. TSG-00795-2021. *(Corresponding author: Osman Boyaci.)*

Osman Boyaci, Katherine R. Davis, Thomas J. Overbye, and Erchin Serpedin are with the Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843 USA (e-mail: osman.boyaci@tamu.edu; katedavis@tamu.edu; overbye@tamu.edu; eserpedin@tamu.edu).

Mohammad Rasoul Narimani is with the College of Engineering, Arkansas State University, Jonesboro, AR 72404 USA (e-mail: mnarimani@astate.edu).

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

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TSG.2021.3117977.

Digital Object Identifier 10.1109/TSG.2021.3117977

I. INTRODUCTION

SMART grids integrate Information and Communication Technologies (ICT) into large-scale power networks to generate, transmit, and distribute electricity more efficiently [1]. Remote Terminal Units (RTUs) and Phasor Measurement Units (PMUs) are utilized to acquire the physical measurements and deliver them to the Supervisory Control and Data Acquisition Systems (SCADAs). Then, the ICT network transfers these measurements to the application level where the power system operators process them and take the necessary actions [2]. As a direct consequence, power system reliability is determined by the accuracy of the steps along this cyber-physical pipeline [3]. Power system state estimation (PSSE) modules employ these measurements to estimate the current operating point of the grid [4] and thus the integrity and trustworthiness of the measurements are crucial for proper operation of power systems. In addition, the accuracy of power system analysis tools such as energy management, contingency and reliability analysis, load and price forecasting, and economic dispatch depends on these measurements [5]. Thus, power system operation strongly depends on the accuracy of the measurements and the integrity of their flow through the system. Therefore, metering devices represent highly attractive

targets for adversaries that try to obstruct the grid operation by corrupting the measurements.

By disrupting the integrity of measurement data, false data injection attacks (FDIAs) constitute a considerable cyber-physical threat. More specifically, an adversary injects some false data to the measurements in order to mislead the PSSE and force it to converge to another operating point. Since the state of the power system is miscalculated by using these false data, any action taken by the grid operator based on the false operating point can lead to serious physical consequences including systematic problems and failures [6]. In traditional power grids, the largest normalized residual test (LNRT) is employed within the bad data detection (BDD) module along with PSSE to detect the "bad" measurement data [4]. Nevertheless, a designed false data injected measurement can bypass the BDD. In particular, [2], [7] show that by satisfying the power flow equations, an intruder can create an unobservable (stealth) FDIA and bypass the BDD if s/he has sufficient information about the grid. Various methods have been proposed to alert the grid operator about the presence of the FDIA without providing any information about the attack location [8], [9]. Localizing the attack is crucial for power system operators since they can take preventive action such as isolating the under-attack buses and re-dispatching the system accordingly. Therefore, this paper focuses jointly on detection and localization of the FDIA in power systems.

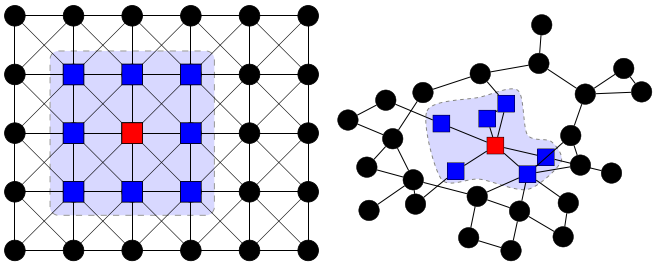
A. Related Works

In general, there are two main approaches to detect and localize the FDIAs: model-based and data driven approaches [8]. In the model-based methods [10]–[14], a model for the system is built and its parameters are estimated to detect the FDIAs. Since there is no training, these methods do not require the historical data. However, the detection delays, scalability issues and threshold tuning steps can limit the performance and usability of the model-based approaches [9]. Conversely, the data-driven methods [15] are system independent and require historical data and a training procedure. However, they provide scalability and real time compatibility due to the excessive training. Data driven methods, machine learning (ML) [16], in particular, offer superior performances to detect FDIAs in power systems as the historical datasets are growing [8], [9]. Therefore, we employ a data driven approach in this work for detecting and localizing FDIAs in power systems.

While there has been a great deal of research on detection of FDIAs, only a few attempts have been made to localize these attacks [10]–[15]. Since localization of FDIAs is relatively a newer research subject compared to detection of these attacks, the current approaches proposed in literature suffer from some limitations. A multistage localization algorithm based on graph theory results is proposed in [10] to localize the attack at cluster level. Nevertheless, the low resolution hinders the benefits of localization in cluster level algorithms. In [11], a model-driven analytical redundancy approach utilizing Kalman filters is presented for joint detection and mitigation of FDIA in AGC systems. In their model, the authors of [11] first determine a threshold using the Mahalanobis norm of the residuals of the non-attacked situation. Any residue larger than the threshold is regarded as an attacked sample. Apart from the manual threshold optimization steps, detection times are at the range of seconds in their estimation based models. A generalized modulation operator that is applied on the states of the system is presented as an ongoing work in a brief announcement in [12] to localize the FDIAs in power systems. Yet, the results are not published as of today. Authors in [13] present an internal observer-based detection and localization method for FDIA in power systems. They create and assign an interval observer to each measurement device and construct a customized logic localization judgment matrix to detect and localize the FDIA. Nevertheless, their average detection delay is more than 1.1 seconds, which can highly limit their usability in a real life scenario. Lack of scalability and the need for a custom solution requiring manual labor represent additional limitations of this method. A Graph Signal Processing (GSP) based approach is developed in [14] to detect and localize FDIAs using the Graph Fourier Transform (GFT), local smoothness, and vertex-frequency energy distribution methods. Hovewer, the random and easily detectable attacks employed to test their models do not comprehensively assess the actual performance of the models. Besides, manual threshold tuning of graph filters (GFs) brings extra effort for their proposed methods. Authors in [15] propose physics- and learning-based approaches to detect and localize the FDIAs in automatic generation control (AGC) of power systems. While the physics-based method relies on interaction variables, the learning-based approach exploits the historical Area Control Error (ACE) data, and utilizes a Long Short Term Memory (LSTM) Neural Network (NN) to generate a model for learning the data pattern. Nevertheless, [15] reports results limited to a 5-bus system and assumes training an LSTM model for each measurement. Thus, the limited number of components deeply confines the large scale attributes of the proposed method. Furthermore, training a separate detector for each bus extremely increases the overall model complexity for large systems and reduces its suitability for real world applications.

B. Motivation

Due to their graph-based topology, graph structural data such as social networks, traffic networks, and electric grid networks cannot be modeled efficiently in the Euclidean space and require graph-type architectures [17]. Processing (filtering) an image having 30 pixels and a power grid having 30 buses are demonstrated in Fig. 1. Since nodes are ordered and have the same number of neighbors for image data, it can be processed in a 2D Euclidean space. For example a sliding kernel can easily capture the spatial correlations of pixels in this Euclidean space. Conversely, neighborhood relationships are unordered and vary from node to node in a graph signal [17]. Therefore, graph signals need to be processed in non-Euclidean spaces determined by the topology of the graph. In fact, as a highly complex graph structural data, smart grid signals require graph type architectures such as GSP or GNN to exploit the spatial correlations of the grid.



(a) Processing an image having 30 pixels in Euclidean space. Nodes and and edges represent pixels and their neighbors in the image, respectively.

(b) Processing a power grid having 30 buses in non-Euclidean space. Nodes and edges represent buses and lines of the grid, respectively.

Fig. 1. Demonstration of signal processing in Euclidean 1(a) and non-Euclidean 1(b) spaces with an image and a power grid signal [17]. Neighbor nodes (blue) of a node (red) are ordered and constant in size for the image having 30 pixels in 2D Euclidean space. In contrast, they are unordered and variable in size for the IEEE 30-bus system in Non-Euclidean space. Therefore, in order to efficiently model the spatial correlation of the power grid, graph type approaches that consider the topology of the underlying systems such as GSP and GNN are necessary.

GSP has emerged in the past few years to deal with the data in non-Euclidean spaces [18]. A few researchers designed GFs to detect and localize the FDIA [12], [14] using GSP. However, manual tailoring of the filters and detection thresholds substantially limits the applicability and efficiency of GSP. Conversely, GNN, as a data-driven counterpart of the GSP, eliminates the custom design steps and provides an end-to-end design that exploits the spatial locality dictated by the historical data. Similar to the classical signal processing, a graph signal is first converted into the spectral domain by GFT, then its Fourier coefficients are multiplied with those filter weights and finally the signal transformed back into the vertex domain by the inverse GFT [19]. To circumvent this spectral decomposition and domain transformation, polynomial GFs are proposed in [20] in which localized filters are learned directly in the vertex domain [21]. For a polynomial GF, the output of each vertex v is only dependent on the K-hop neighborhood of v and its spectral response is a K-order polynomial. Polynomial GFs, which are also referred to as finite impulse response (FIR) GFs due to the local information sharing, perform a weighted moving average (MA) filtering [22], [23]. However, FIR GFs may require a high degree polynomial to capture the global structure of the graph. In fact, interpolation and extrapolation performance of high degree polynomials are unsatisfactory [22], and they are not "flexible" enough to adapt to sudden changes in the spectral domain [21]. To overcome this limitation, infinite impulse response (IIR) type GFs performing Auto-Regressive Moving Average (ARMA) are proposed in [22]. Contrary to FIR GFs, IIR GFs have rational type spectral responses. Therefore, IIR GFs can implement more complex responses with a low degree of polynomials both in the numerator and denominator since rational functions have better performance compared to polynomial ones in terms of interpolation and extrapolation capabilities [21], [22].

Detection and localization of FDIA can be a challenging task if an intruder has 'enough' information about the grid to create a stealth attack [7]. S/he can hide an attack vector into an honest sample if the topology of the grid is ignored. Moreover, s/he can design an attack vector so that a malicious sample can be indistinguishable from an honest one if the spatial correlations of grid data are not well captured or the designed GFs do not satisfy the required spectral response. Thus, we design an GNN based model by utilizing ARMA GFs to be able to fit sharp changes in the spectral domain of the grid. Filter weights are learned automatically during training by an end-to-end data-driven approach. To compare our results with the existing data-driven techniques, we utilized several models to jointly detect and localize the FDIA. Moreover, for a fair comparison, the Bayesian hyper-parameter optimization technique is employed to all models for tuning the models' hyperparameters such as number of layers, neurons, etc.

C. Contributions and Paper Organization

The contributions of this work are outlined as follows.

- To properly capture the spatial correlations of the smart grid data in a non-Euclidean space, we utilize IIR type ARMA GFs which provide more flexible frequency responses compared to FIR type Chebyshev GFs. It is demonstrated on IEEE 118- and 300-bus test systems that ARMA GFs better approximate the desired filter response compared to CHEB GFs for the same filter order by comparing their empirical frequency responses when approximating an ideal band pass filter.
- To precisely test our proposed method, we generate a dataset for each test system with 1-minute intervals using several FDIA generation algorithms in the literature as well as our optimization-based FDIA method developed in our previous paper [24].
- To automatically determine the unknown filter weights by an end-to-end data-driven approach, we propose a scalable, ARMA GF-based GNN model that jointly detects and localizes the FDIAs in a few milliseconds. The proposed architecture efficiently predicts the presence of the attack for the whole grid and for each bus separately.
- To fairly compare the proposed method with the currently available approaches, we implement the other data-driven models in the literature and compare our detection and localization results with them. Hyperparameters of the models are tuned systematically using the Bayesian hyper-parameter optimization technique.
- To adequately assess the localization performance, we evaluate the localization results, using both sample wise and node wise comparisons. For instance, although sample wise localization could yield fairly high accuracy for the entire system, the same set of nodes could be missed or falsely alarmed at each sample. If revealed, these nodes could be easily targeted by the intruders.
- To better analyze and visualize the multidimensional data processed by the implemented models, we embed them into a two dimensional (2D) space using the t-SNE algorithm [25]. By visually inspecting the output of models' intermediate layers in 2D, it is verified that the ARMA GNN based model preserves the structure of the data, and hence gives better detection performance.

The rest of this paper is organized as follows. Section II presents the problem formulation. Section III proposes the approach for the joint detection and localization of FDIA. Numerical results are presented in Section IV. Section V finally concludes the paper.

II. Problem Formulation

The system state $\boldsymbol{x}$ (V_i and θ_i at each bus i) is estimated using the PSSE module. The PSSE iteratively solves the optimization problem in (1) phrased as a weighted least squares estimation (WLSE) using the complex power measurements $\boldsymbol{z}$ collected in noisy conditions by RTUs and PMUs:

$$\hat{\boldsymbol{x}} = \min_{\boldsymbol{x}} (\boldsymbol{z} - h(\boldsymbol{x}))^T \boldsymbol{R}^{-1} (\boldsymbol{z} - h(\boldsymbol{x})), \tag{1}$$

where $\boldsymbol{R}$ represents measurements' error covariance matrix and $\boldsymbol{z}$ includes $\boldsymbol{P_i}$, $\boldsymbol{Q_i}$, $\boldsymbol{P_{ij}}$, and $\boldsymbol{Q_{ij}}$.

FDIA aim to deceive the PSSE by deliberately injecting false data $\boldsymbol{a}$ into some of the original measurements $\boldsymbol{z_o}$ in such a way that the state vector $\boldsymbol{x}$ converges to another point in the state space of the system. Formally,

$$\boldsymbol{z_o} = h(\hat{\boldsymbol{x}}), \quad \boldsymbol{z_a} = \boldsymbol{z_o} + \boldsymbol{a} = h(\check{\boldsymbol{x}}) \tag{2}$$

which means if an adversary can design $\boldsymbol{a} = h(\check{\boldsymbol{x}}) - h(\hat{\boldsymbol{x}})$, s/he can change the system state from $\hat{\boldsymbol{x}}$ to $\check{\boldsymbol{x}}$ without being detected by the LNRT based traditional BDD systems.

In general, an adversary tries to change specific measurement(s) in the power system in order to maximize the damage to the grid and at the same time minimize the probability of being detected. To this end, s/he alters some other measurement(s) connected to the targeted meter(s) since each $\boldsymbol{x}$ relates to multiple $\boldsymbol{z}$ through $\boldsymbol{z} = h(\boldsymbol{x})$. In order to reflect this constraint and to be realistic, we assume that an adversary targets a specific area of grid represented by $\mathcal{T}$ and crafts the attack vector $\boldsymbol{a}$ by changing the measurements denoted by $\mathcal{T}_z$ to spoil the state variables represented by $\mathcal{T}_x$ in this area.

The grid operator, in contrast, aims to detect those attacks and localize the attacked buses if there are any. Therefore, we formulate the FDIA detecting and localization problem as a *multi-label* classification task where each bus has a binary label indicating the presence of attack with true label 1. We also reserve an extra binary label for the whole grid to denote the attack presence with true label 1. Fig. 2 clarifies the proposed multi-label classification approach by depicting the actual and predicted under attack buses for an exemplary attack on the IEEE 14-bus test system.

III. Joint Detection and Localization of FDIA

Connected, undirected and weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{W})$ having a finite set of vertices $\mathcal{V}$ with $|\mathcal{V}| = n$, a finite set of edges $\mathcal{E}$, and a weighted adjacency matrix $\boldsymbol{W} \in \mathbb{R}^{n \times n}$ can be used to represent the topology of a smart power grid [18]. In this representation, buses correspond to vertices $\mathcal{V}$, branches and transformers corresponds to edges $\mathcal{E}$ and line admittances correspond to $\boldsymbol{W}$. Similarly, a signal or a function $f : \mathcal{V} \to \mathbb{R}$ in $\mathcal{G}$ is represented by a vector $\boldsymbol{f} \in \mathbb{R}^n$, where the element i of the vector corresponds to a scalar at the vertex $i \in \mathcal{V}$.

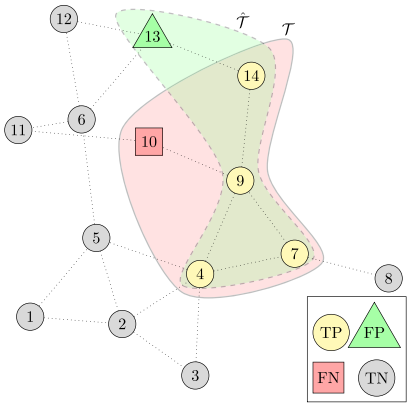


Fig. 2. Visualization of an attack and its prediction on the example IEEE 14 bus system where the actual $\mathcal{T} = \{4, 7, 9, 10, 14\}$ and predicted $\hat{\mathcal{T}} = \{4, 7, 9, 13, 14\}$ areas are enclosed with the solid red and dashed green surfaces, respectively. True positives $\mathcal{T} \cap \hat{\mathcal{T}} = \{4, 7, 9, 14\}$, false positives $\mathcal{T}' \cap \hat{\mathcal{T}} = \{13\}$, false negatives $\mathcal{T} \cap \hat{\mathcal{T}}' = \{10\}$, and true negatives $\mathcal{T}' \cap \hat{\mathcal{T}}' = \{1, 2, 3, 5, 8, 6, 11, 12\}$ are represented by yellow circles, green triangles, red squares, and black circles, respectively. In this example, the presence of the attack is correctly predicted. Nevertheless, attack to the bus 10 is missed and bus 13 is falsely alarmed even though it is not under attack.

A. Spectral Graph Filters

In spectral graph theory, the normalized Laplacian operator $\boldsymbol{L} = \boldsymbol{I_n} - \boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T \in \mathbb{R}^{n \times n}$ plays an important role for graph $\mathcal{G}$ where $\boldsymbol{D}$ and $\boldsymbol{I_n} \in \mathbb{R}^{n \times n}$ represent the degree and identity matrices, respectively. The columns $\boldsymbol{u}_i \in \mathbb{R}^{n \times 1}$ of matrix $\boldsymbol{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n] \in \mathbb{R}^{n \times n}$ store the n orthonormal eigenvector $\boldsymbol{u_i}$ and constitute the graph Fourier basis. Diagonal matrix $\boldsymbol{\Lambda} = \text{diag}([\lambda_1, \ldots, \lambda_n]) \in \mathbb{R}^{n \times n}$ captures the n eigenvalues representing the graph Fourier frequencies [18]. Analogously to the classical Fourier Transform, Graph Fourier Transform (GFT) transforms a vertex domain signal into the spectral domain: the forward and inverse GFT are defined by $\tilde{\boldsymbol{X}} = \boldsymbol{U}^T\boldsymbol{X}$, and $\boldsymbol{X} = \boldsymbol{U}\tilde{\boldsymbol{X}}$, where $\boldsymbol{X}$ and $\tilde{\boldsymbol{X}} \in \mathbb{R}^{n \times f}$ denote the vertex and spectral domain signals with f features at each node, respectively [18]. In fact, $\boldsymbol{X}$ is filtered by a GF h:

$$\boldsymbol{Y} = h * \boldsymbol{X} = h(\boldsymbol{L})\boldsymbol{X} = \boldsymbol{U}h(\boldsymbol{\Lambda})\boldsymbol{U}^T\boldsymbol{X} \tag{3}$$

by first converting the vertex domain signal $\boldsymbol{X}$ into the spectral domain using the forward GFT, then scaling the Fourier components by $h(\boldsymbol{\Lambda}) = \text{diag}\,[h(\lambda_1), \ldots, h(\lambda_n)]$, and finally reverting it back to the vertex domain by the inverse GFT [18]. For example, $\boldsymbol{X}$, h, and $\boldsymbol{Y}$ may correspond to bus injections values with high frequency noise, a low pass GF and filtered bus injections values, respectively in eq. (3). Nonetheless, this spectral filtering is not spatially localized since each λ_i is processed for each node. Besides its computational complexity is high due to eigenvalue decomposition (EVD) of $\boldsymbol{L}$ and the matrix multiplications with $\boldsymbol{U}$ and $\boldsymbol{U}^T$.

B. Polynomial Graph Filters

To localize spectral filters and reduce their complexity, polynomial spatial filters $h_{POLY}(\lambda) = \sum_{k=0}^{K-1} a_k \lambda^k$ are proposed to approximate the required filter response [20]. Since only K-hop neighbors of v are considered to calculate the filter

response at each $v \in \mathcal{V}$, they are K-localized. In fact, they implement the weighted MA filtering in the form of FIR [23].

Chebyshev polynomial approximation [26] is one of the preferred methods in signal processing due to their fast computation since they are generated via a recursion and not a convolution [27]. The Chebyshev polynomial of the first kind $T_k(x)$ can be computed recursively $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$ where $T_0(x) = 1$ and $T_1(x) = x$ [26]. Thus, a filter h can be approximated by a truncated expansion of Chebyshev polynomials T_k, up to order $K-1$. So, $\boldsymbol{X}$ can be filtered:

$$\boldsymbol{Y} = h * \boldsymbol{X} = h(\boldsymbol{L})\boldsymbol{X} = \sum_{k=0}^{K-1} a_k T_k(\tilde{\boldsymbol{L}})\boldsymbol{X} \tag{4}$$

where $T_k(\tilde{\boldsymbol{L}}) \in \mathbb{R}^{n\times n}$ is the Chebyshev polynomial of order k evaluated at the scaled Laplacian $\tilde{\boldsymbol{L}} = 2\boldsymbol{L}/\lambda_{max} - \boldsymbol{I_n}$ [20] where $\boldsymbol{a} \in \mathbb{R}^K$ is a vector of Chebyshev coefficients. Full EVD can be omitted since this operation only requires the largest eigenvalue λ_{max} which can be efficiently approximated by the power method [28]. Although the MA type Chebyshev (CHEB) GFs are fast and localized, they often require high-degree polynomials to capture the graph's global structure. In fact, it restricts their ability to adapt sharp transitions in the frequency response due to the poor interpolation and extrapolation capabilities of high degree polynomials [29].

C. Rational Graph Filters

To circumvent these problems, distributed IIR type ARMA GFs are proposed in [22], [29]. They better approximate the sudden changes in the frequency response in comparison with the FIR type MA GFs due to their rational filter composition. A potential building block of K-order ARMA GFs may start with a first order recursive ARMA_1 filter:

$$\boldsymbol{Y}^{t+1} = a\tilde{\boldsymbol{L}}\boldsymbol{Y}^t + b\boldsymbol{X}, \tag{5}$$

where $\boldsymbol{Y}^t$ is the filter output at iteration t, $\boldsymbol{X}$ is the filter input, a and b are arbitrary coefficients, and modified Laplacian $\tilde{\boldsymbol{L}} = \frac{\lambda_{max}-\lambda_{min}}{2}\boldsymbol{I_n} - \boldsymbol{L}$ is a linear translation of $\boldsymbol{L}$ with same eigenvectors as $\boldsymbol{L}$ and shifted eigenvalues $\tilde{\lambda}_n = \frac{\lambda_{max}-\lambda_{min}}{2} - \lambda_n$ relative to those of $\boldsymbol{L}$. According to [30, Th. 1], eq. (5) converges regardless of $\boldsymbol{Y}^0$ and $\boldsymbol{L}$ values and its frequency response is given by $h_{ARMA_1}(\tilde{\lambda}_n) = \frac{b}{1-a\tilde{\lambda}_n}$. In fact, eq. (5) provides a useful distributed filter realization [22]. At each iteration t, each node i revises its output $\boldsymbol{Y}_i^t \in \mathbb{R}^{n\times c_{out}}$ with a linear combination of its input $\boldsymbol{X_i} \in \mathbb{R}^{n\times c_{in}}$ and its adjacent nodes' outputs $\boldsymbol{Y}_j^{t-1}$, where c_{in} and c_{out} denote the number of channels in the input and the output tensors, respectively. It can be implemented as a NN layer if we unroll the recursion into T fixed iterations:

$$\boldsymbol{Y}^{t+1} = \tilde{\boldsymbol{L}}\boldsymbol{Y}^t\boldsymbol{\alpha} + \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\theta}, \tag{6}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^{c_{out}\times c_{out}}$, $\boldsymbol{\beta} \in \mathbb{R}^{c_{in}\times c_{out}}$, and $\boldsymbol{\theta} \in \mathbb{R}^{c_{out}}$ are trainable weights. Besides, since $0 \le \lambda_{min} \le \lambda_{max} \le 2$, the modified Laplacian can be simplified to $\tilde{\boldsymbol{L}} = \boldsymbol{I_n} - \boldsymbol{L}$ for $\lambda_{min} = 0$, and $\lambda_{max} = 2$ [21]. In Fig. 3, NN implementation of the ARMA_1 block which implements the eq. (6) in T fixed iterations is depicted. ARMA_1's K-order version ARMA_K filter

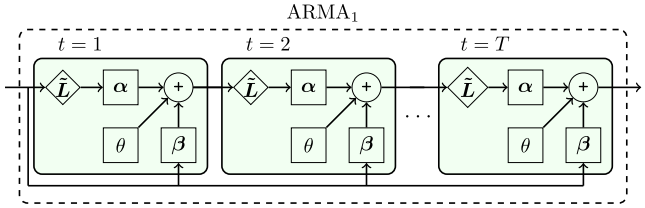


Fig. 3. NN Implementation of ARMA_1 filter as a building block of ARMA_K layer. In T fixed iterations, an ARMA_1 block realizes eq. (6).

can be realized by averaging K parallel ARMA_1 filters with $\boldsymbol{Y} = \frac{1}{K}\sum_{k=1}^{K}\boldsymbol{Y}_K^T$ which leads to an ARMA_K GF with a rational frequency response $h_{ARMA_K}(\tilde{\lambda}_n) = \sum_{k=1}^{K}\frac{b_k}{1-a_k\tilde{\lambda}_n}$ with a $K-1$ and K order polynomials in its numerator and denominator, respectively. For detailed analysis and justifications, please refer to [21], [22], [29]–[31].

D. Frequency Response of Polynomial and Rational GFs

To demonstrate the ARMA GFs better fit sharp changes in the frequency response compared to that of the CHEB GFs, we design two ideal GFs in equations (7) and (8) for IEEE 118- and 300-bus test cases, respectively.

$$h^{\dagger}(\lambda) = \begin{cases} 1, & \frac{\lambda_{max}}{3} < \lambda < \frac{2\lambda_{max}}{3} \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

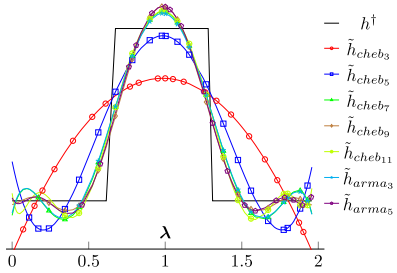
$$h^{\ddagger}(\lambda) = \begin{cases} 1, & \lambda < \frac{\lambda_{max}}{2} \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

Then, we investigate the approximating capability of the ARMA and the CHEB GFs by numerically analyzing their frequency responses. Note that similar results can be obtained by any other filters or test cases [22]. Let $\boldsymbol{x}, \boldsymbol{y} \in R^{n\times 1}$ denote the input and output of a GF $h(\lambda)$, respectively. Then, according to eq. (3), empirical frequency response $\tilde{h}$ can be calculated by $\tilde{h}(\lambda_i) = \frac{\boldsymbol{u}_i^T\boldsymbol{y}}{\boldsymbol{u}_i^T\boldsymbol{x}}$ [29]. Namely, each $\tilde{h}(\lambda_i)$ represents how $\boldsymbol{u}_i$, corresponding to λ_i, "scales" $\boldsymbol{x}$ to obtain $\boldsymbol{y}$.

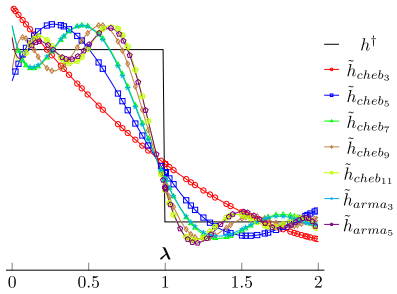
In order to obtain $\tilde{h}(\lambda_i)$ values, we first randomly generated 2^{16} $\boldsymbol{x}$s for the aforementioned systems from the normal distribution and filter them by $h^{\dagger}$ and $h^{\ddagger}$ using eq. (3) to obtain $\boldsymbol{y}$s. Then, a layer of CHEB and ARMA models with no activation function are trained in batches having 2^6 samples of $\boldsymbol{x}$ and $\boldsymbol{y}$ values until there is no improvement. Next, $\tilde{h}(\lambda_i)$ values are calculated for each $\boldsymbol{x}, \boldsymbol{y}$ tuple, averaged for smooth transitions, and plotted. As seen from Fig. 4, due to their rational type frequency responses, ARMA GFs are more flexible to fit sudden changes for a fixed K when compared to CHEB GFs having polynomial type frequency responses. This constitutes the main motivation for selecting ARMA GFs for jointly detecting and localizing the FDIA in power grids.

E. Architecture of the Proposed Joint Detector & Localizer

The proposed joint detector and localizer consists of one input layer to represent complex bus power injections, $L-1$ hidden ARMA_K layers to extract spatial features, one dense layer to predict the probability of attack at each node, and one output layer to return an extra bit to indicate the probability



(a) an ideal bandpass $h^\dagger$ for IEEE-118.



(b) an ideal lowpass $h^\ddagger$ for IEEE-300.

Fig. 4. Empirical frequency responses of CHEB and ARMA GFs when approximating ideal filters $h^\dagger$ and $h^\ddagger$ applied on IEEE 118- and 300-bus test systems, respectively. Compared to CHEB, ARMA better approximates the desired filter for the same K (e.g., $\tilde{h}_{cheb_3}$ vs $\tilde{h}_{arma_3}$) and it requires a lower K for the same level of approximation (e.g., $\tilde{h}_{cheb_{11}}$ vs $\tilde{h}_{arma_5}$).

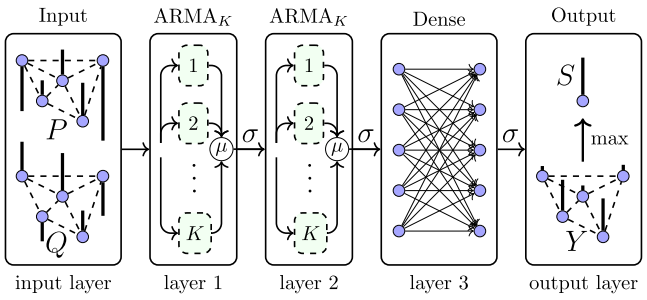


Fig. 5. Architecture of the proposed ARMA GNN based detector and localizer with three hidden layers where each ARMA$_K$ layer consists of K parallel ARMA$_1$. Each one of the K dashed blocks in an ARMA$_K$ layer corresponds to an ARMA$_1$ block depicted with a dashed block in Fig. 3. While complex power injections $\boldsymbol{P}, \boldsymbol{Q}$ and predicted attack probabilities $\boldsymbol{Y}, S$ at the node and graph level are visualized with thick bars at each node, activation and mean value functions are represented with σ and μ, respectively.

of attack at the graph level. Its architecture is demonstrated in Fig. 5 for $L = 3$ with a small graph having $n = 5$.

In this multi-layer GNN model, the input tensor $[\boldsymbol{P_i}, \boldsymbol{Q_i}]$ is given by $\boldsymbol{X^0} \in \mathbb{R}^{n\times 2}$, the output tensor of hidden layer l is denoted by $\boldsymbol{X^l} \in \mathbb{R}^{n\times c_l}$, and model outputs are denoted by $\boldsymbol{Y} \in \mathbb{R}^n$ and $S \in \mathbb{R}$ to indicate the location and the presence of the attack, respectively, where c_l represents the number of channels in layer l for $1 \leq l \leq L$. In particular, while an ARMA$_K$ layer takes $\boldsymbol{X}^{l-1} \in \mathbb{R}^{n\times c_{l-1}}$ as input and produces $\boldsymbol{X}^l \in \mathbb{R}^{n\times c_l}$ as output in layer l, the dense layer propagates the information to the whole graph and outputs the probability of the attack at the node level with $\boldsymbol{Y} \in \mathbb{R}^n$ for localization. Finally, the output layer detects the attack at the graph level by $S = \max(\boldsymbol{Y}) \in \mathbb{R}$ and outputs it with $\boldsymbol{Y}$. Note that the last ARMA$_K$ layer's output channel is selected as one in order to have one feature for each $v \in \mathcal{V}$. In addition, ReLU activation is used at the end of each ARMA$_K$ layer to increase the model's nonlinear modeling ability, whereas sigmoid is employed to transform the outputs to probabilities.

TABLE I
IMPLEMENTED FDIAS

FDIA type	formulation	used in
optimization-based (A_o)	Eq. (5) in [24]	[24]
data replay (A_r)	$z_a^i(t) = z_o^i(t-\tau)$	[32], [33]
distribution-based (A_d)	$z_a^i(t) = \mathcal{N}(\mu(z_o^i), \sigma^2(z_o^i))$	[37], [38]
data scale (A_s)	$z_a^i(t) = \mathcal{U}(0.9, 1.1) \times z_o^i(t)$	[14], [15]

IV. EXPERIMENTAL RESULTS

A. Data Generation

Due to the privacy concerns, there is no preexisting publicly available dataset to train and evaluate the proposed models against FDIA. Thus, researches use historical load profiles to mimic the timely deviations of the grids they simulate [14], [15], [32]–[35]. We take the same approach based on the historical load profile of NYISO [36] to generate our dataset. As a first step, we download 5-minute intervals of the actual load profile of NYISO for July 2021 and interpolate them to increase the resolution to 1-minute. Next, we generate a realistic dataset following the Algorithm 1 in our previous work [24] for the IEEE 57-, 118-, and 300-bus standard test cases using 1-minute interval load profile. Namely, for each timestamp, load values are distributed and scaled to buses proportional to their initial values, AC power flow algorithms are executed, and 1% noisy power measurements are saved.

To simulate the FDIA, we implement some of the frequently used FDIA generation algorithms in the literature such as data replay attacks (A_r) [32], [33], data scale attacks (A_s) [14], [15], and distribution-based (A_d) attacks [37], [38] as well as our constrained optimization based FDIA (A_o) method explained thoroughly in [24]. While A_r simply changes a measurement z_o^i with one of its previous values at τ back in time, A_s multiply it with a number sampled from a uniform distribution ($\mathcal{U}$) between 0.9 and 1.1. In contrast, A_d mimics the mean (μ) and variance (σ^2) of the measurement by sampling from a normal distribution ($\mathcal{N}$) and A_o solves a constrained optimization problem to maximize the changes in state variables while minimizing the changes in measurements.

Implemented attacks types, their formulations and some works that have utilized them are given in Table I.

We shuffled the whole data to eliminate the seasonality, standardized it with a zero mean and a standard deviation of one to have a faster and smoother learning process, and split it into three sections: 2/3 for training, 1/6 for validating and hyper-parameter tuning, and 1/6 for testing the proposed models. In order to evaluate the performance of our method under unseen attack types, we arbitrarily selected A_o and A_d and included them in the training and validation splits. Test split, in contrast, includes all of the four FDIA methods given in Table I. The number of honest samples are equalized with

TABLE II
NUMBER OF SAMPLES IN EACH SPLIT

split	non-attacked	A_o	A_r	A_d	A_s	total
train	11520	5760	0	5760	0	23040
validation	2880	1440	0	1440	0	5760
test	2880	720	720	720	720	5760

the number of malicious samples in each split to have a balanced classification problem as can be seen from Table II. The final dataset assumes 60 samples/hour × 24 hour/day × 24 day = 34560 samples which consist of complex power measurements, complex bus voltages, and $n+1$ binary labels to indicate the true labels for each bus and the whole grid at each timestamp.

B. Feature Selection, Performance Metrics, and Training

To be able to rapidly detect and localize the attacks instead of waiting for $\boldsymbol{V_i}$ and $\boldsymbol{\theta_i}$ values at the output of the PSSE process, we employ power measurements as input features in our detectors. From the power measurements, only $\boldsymbol{P_i}$ and $\boldsymbol{Q_i}$ values are fed to the models as seen from the input layer of Fig. 5 since $\boldsymbol{P_i} + j\boldsymbol{Q_i} = \sum_{k\in\Omega_i} \boldsymbol{P_{ik}} + j\boldsymbol{Q_{ik}}$, node features can represent branch features as summation in their corresponding set of buses Ω_i connected to bus i. Besides, it is experimentally verified that utilizing $\boldsymbol{V_i}$ and $\boldsymbol{\theta_i}$ values along with $\boldsymbol{P_i}$ and $\boldsymbol{Q_i}$ does not increase the model performance due to tuples' high correlation. PSSE and BDD modules, on the contrary, continue to receive every available measurement to operate. As for the weighted adjacency matrix we select $\boldsymbol{W} = |\boldsymbol{Y_{bus}}|$ to calculate $\tilde{\boldsymbol{L}}$ and feed the ARMA$_K$ layers where $\boldsymbol{Y_{bus}} \in \mathbb{R}^{n\times n}$ denotes nodal admittance matrix.

For performance evaluation we use detection rate $DR = \frac{TP}{TP+FN}$, false alarm rate $FA = \frac{FP}{FP+TN}$, and $F1$ score $F1 = \frac{2*TP}{2TP+FP+FN}$, where TP, FP, TN, and FN represent true positives, false positives, true negatives, and false negatives, respectively [16]. In addition, to overcome the division by zero problem when there is no attack at all, we assumed $DR = 1$, $FA = 0$, and $F1 = 1$ if all the labels are correctly predicted as not attacked. Otherwise, even if there is one mismatch, we assign $DR = 0$, $FA = 1$, and $F1 = 0$.

All free unknown parameters defined in the model are computed by a multi-label supervised training using the binary cross-entropy loss. Training samples are fed into the model as mini batches of 256 samples with 256 maximum number of epochs. In addition, we employ early stopping criteria where 16 epochs are tolerated without any improvement less than e^{-4} in the validation set's cross entropy loss. All the implementations were carried out in Python 3.8 using the Pandapower [39], Sklearn [40], t-SNE [25], and Tensorflow [41] libraries on Intel i9-8950 HK CPU 2.90GHz with NVIDIA GeForce RTX 2070 GPU.

C. Joint Detection and Localization Results

Since we take a data-driven approach in this work, we implement other existing data-driven approaches from the literature to compare with our method. To the best of our knowledge [15] is the only data-driven approach in the literature in which authors employ LSTM architecture to localize the FDIA. Thus, we trained an LSTM model with our dataset to compare the performances. In addition, although they are proposed for detection, we implement other available methods in the literature suitable for the multi-label classification task such as Decision Tree (DT) [42], K-Nearest Neighbor (KNN) [43], Multi Layer Perceptron (MLP) [44], Convolutional Neural Network (CNN) [43], and Chebyshev GNN (CHEB) [24]. We train, validate and test these models similarly to the proposed model using our dataset as we do not have access to the data set of corresponding works.

Model hyper-parameters are tuned with Sklearn [40] and Keras-tuner [45] Python libraries by using Bayesian optimization techniques. Models are trained on the training set and their hyper-parameters are optimized on the validation set for each IEEE test system for 250 trials. Finally, models with optimal parameters in terms of the validation set performance are saved and their results are presented for detection and localization. Table III shows the hyper-parameter set and the optimal hyper-parameters for each model and test system.

In Table IV, detection performance of the optimized models for each test system is tabulated as percentages. For all test

TABLE III
OPTIMIZED MODEL HYPER-PARAMETERS

model	param	options	IEEE-57	IEEE-118	IEEE-300
DT	criterion	{gini, entropy}	gini	entropy	entropy
	min. split	{2, 3, …, 8}	2	3	2
	max. depth	{1, 2, …, 64}	60	64	64
	features	{sqrt, log2}	log2	log2	sqrt
KNN	algorithm	{ball, kd}	kd	ball	kd
	neighbors	{3, 5, 7, 9}	3	3	3
	leaf size	{4, 5, …, 64}	29	62	58
	p	{1, 2, 3, 4}	2	1	1
MLP	layers	{1, 2, 3}	2	3	3
	units	{16, 32, 64, 128}	32	32	128
LSTM	layers	{1, 2, 3}	3	3	2
	units	{16, 32, 64, 128}	16	32	64
CNN	layers	{1, 2, 3}	3	2	3
	units	{16, 32, 64, 128}	32	128	64
	K	{2, 3, 4}	3	4	3
CHEB	layers	{1, 2, 3}	3	4	3
	units	{16, 32, 64, 128}	64	32	64
	K	{2, 3, 4}	3	4	4
ARMA	layers	{1, 2, 3}	3	2	3
	units	{16, 32, 64, 128}	16	16	32
	K	{2, 3, 4}	2	3	3
	iteration	{2, 3, 4, 5}	4	5	5

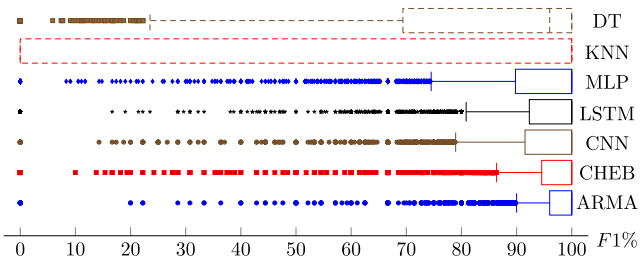
TABLE IV
DETECTION RESULTS IN *DR*, *FA*, AND *F*1 PERCENTAGES

System	IEEE-57			IEEE-118			IEEE-300		
Metric	DR	FA	$F1$	DR	FA	$F1$	DR	FA	$F1$
DT	89.55	5.45	91.84	87.40	8.72	89.13	89.69	9.38	90.11
KNN	19.41	**0.07**	32.49	30.69	**0.00**	46.97	16.67	**0.00**	28.57
MLP	95.07	0.31	97.32	89.20	1.01	93.79	82.74	1.63	89.76
LSTM	98.40	0.24	99.07	96.74	0.10	98.29	94.38	0.03	97.09
CNN	99.79	0.28	99.76	98.47	0.45	99.01	95.28	**0.00**	97.58
CHEB	99.65	0.28	99.69	97.99	0.45	98.76	99.79	0.73	99.53
ARMA	**99.90**	0.28	**99.81**	**99.13**	0.24	**99.44**	**99.97**	0.14	**99.91**

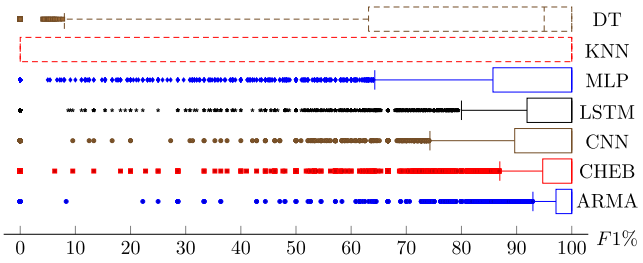
systems, although KNN yields the best *FA* rate, its $F1$ scores are not satisfactory since it gives the lowest *DR*. ARMA, in contrast, reaches the best $F1$ scores with 99.81%, 99.44%, and 99.91%, due to its high *DR* with 99.90%, 99.13%, and 99.97% and low *FA* rate with 0.28%, 0.24%, and 0.14% for 57-, 118-, and 300-bus systems, respectively. Although detection results are close to each other in terms of $F1$ scores for some models such as CHEB and ARMA, CHEB yields almost two and five times *FA* rate for IEEE 118- and 300- bus systems, respectively. Nevertheless, detection considers the attacks at the grid level and any intrusion to a bus in the grid is regarded as an attack. Thus, bus level localization is required to determine the exact place of the attack.

Since localization is a multi-label classification problem, we evaluate it in both possible ways: (i) sample-wise (SW) evaluation yields b metrics where each one of b samples at a fixed time-step is treated individually along the buses, and (ii) node-wise (NW) evaluation yields n metrics where each one of n buses is evaluated separately along the samples. Therefore, in order to better assess the models, we visualize and analyze the distributions of SW and NW localization results in $F1$ percentages by box plots and ratio of items satisfying some specified thresholds. Box plots helps us to visualize the distribution by drawing first (Q_1, 25$^{\text{th}}$ percentile), second (Q_2, 50$^{\text{th}}$ percentile or median), and third (Q_3, 75$^{\text{th}}$ percentile) quartiles, lower ($LW = Q_1 - 1.5 \times (Q_3 - Q_1)$) and upper ($UW = Q_3 + 1.5 \times (Q_3 - Q_1)$) whiskers and outliers [46]. In addition, the ratio of the samples or buses satisfying some thresholds provides quantifiable metrics to assess model performances. For instance, the percentage of samples (buses) having $F1 \leq 5\%$ or $F1 \geq 95\%$ in SW (NW) evaluation can be used to measure the ratio of "unacceptable" and "acceptable" samples in the distributions, respectively.

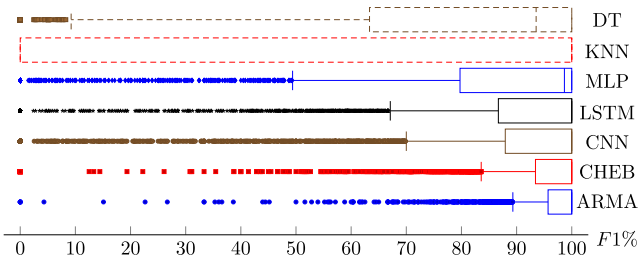
SW localization results are given in Fig. 6. Since the distributions are highly left skewed, the median (Q_2) values can overlap with Q_3. In specific, $Q_2 = Q_3 = UW = 100\%$ except the MLP for IEEE-300 and DT for all systems. This is not surprising because 50% of the samples are not attacked and it is relatively easy to predict them as not attacked for each bus. $Q1$ and LW, in contrast, vary for each model and test system. For example, $LW = Q_1 = 0$ in all test systems for KNN model which shows that $F1 = 0\%$ for more than 1/4 of the samples for that model. Although DT yields better results compared to KNN, its results are unsatisfactory: its Q_1 values are 69.39%, 63.16%, and 63.33%, for 57-, 118- and 300-bus test systems, respectively. Models from the NN family better localize the attacks compared to the classical ML approaches. For example, their $Q1$ values are greater than 79.74% for all test systems. Namely, in at least 3/4 of the samples, attacked buses are correctly labeled with $F1$ score deviating between 79.74% and 100%. To better compare the model performances, the percentages of the samples having $F1 \leq 5\%$ and $F1 \geq 95\%$ are given in Fig. 6(d) for each model and test system. ARMA gives the best results in all test cases: while the sample percentages for $F1 \leq 5\%$ are calculated as 0.21%, 0.56%, and 0.10%, sample percentages with $F1 \geq 95\%$ are measured as 79.53%, 83.00%, and 79.03% for IEEE 57-, 118-, and 300-bus test systems, respectively.



(a) IEEE-57. $Q_2 = 100\%$ except DT, $Q_3 = 100\%$ for all.



(b) IEEE-118. $Q_2 = 100\%$ except DT, $Q_3 = 100\%$ for all.



(c) IEEE-300. $Q_2 = 100\%$ except DT and MLP, $Q_3 = 100\%$ for all.

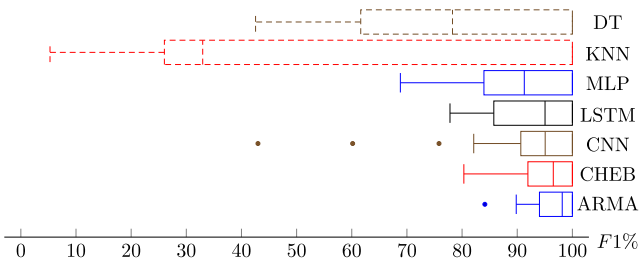
System	IEEE-57		IEEE-118		IEEE-300	
$F1 \lessgtr x$	≤ 5	≥ 95	≤ 5	≥ 95	≤ 5	≥ 95
DT	8.00	50.43	11.18	50.05	10.28	49.18
KNN	40.35	51.41	34.65	57.74	41.68	53.07
MLP	2.64	62.88	6.22	59.79	10.43	54.29
LSTM	0.94	66.82	1.72	69.57	3.00	60.59
CNN	0.24	64.95	0.99	60.82	2.57	60.38
CHEB	0.31	73.89	1.23	74.44	0.47	68.96
ARMA	**0.21**	**79.53**	**0.56**	**83.00**	**0.10**	**79.03**

(d) Sample percentages having $F1 \leq 5\%$ and $F1 \geq 95\%$.

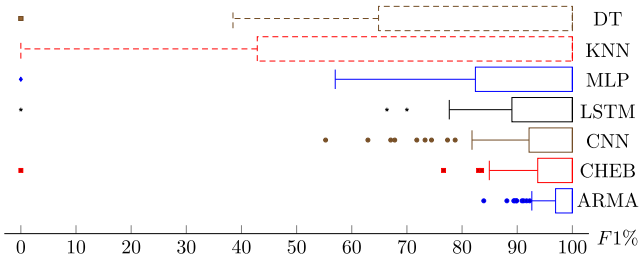
Fig. 6. Distribution of $F1$ scores for sample wise evaluation of localization.

Its "acceptable" ($F1 \geq 95\%$) percentages are 5.64%, 8.56%, and 10.07% greater than the second best model CHEB in SW localization, for IEEE 57-, 118-, and 300-bus test cases, respectively.

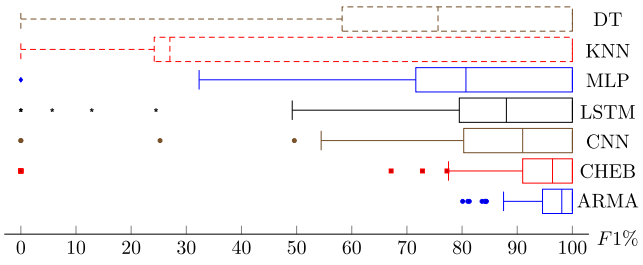
In Fig. 7, the distribution of $F1$ scores for NW evaluation is depicted. Due to the largely left skewed distributions, the median (Q_2) values may overlap with Q_3. Specifically, $Q_3 = 100\%$ for all the models and systems, and $Q2 = 100\%$ for all the models in IEEE-118. Similar to the SW evaluation, performance of DT and KNN is poor: their Q_1 values deviate between 24.19% and 64.86%. MLP, LSTM, and CNN provide better results compared to DT and KNN. Nevertheless, they are subject to some outliers at 0% which means there are some buses that are always mislabeled at each timestamp. The only model that can localize the FDIA for each bus with at least 80% $F1$ score is ARMA. Namely, for all



(a) IEEE-57. $Q_3 = 100\%$ for all.



(b) IEEE-118. $Q_2 = Q_3 = 100\%$ for all.



(c) IEEE-300. $Q_3 = 100\%$ for all.

System	IEEE-57		IEEE-118		IEEE-300	
$F1 \lessgtr x$	≤ 5	≥ 95	≤ 5	≥ 95	≤ 5	≥ 95
DT	**0.00**	38.60	1.69	52.54	0.67	44.00
KNN	**0.00**	38.60	1.69	52.54	0.67	44.00
MLP	**0.00**	40.35	0.85	53.39	0.67	44.00
LSTM	**0.00**	50.88	0.85	55.08	73.33	44.00
CNN	**0.00**	50.88	**0.00**	66.10	0.67	44.33
CHEB	**0.00**	61.40	0.85	69.49	1.67	57.00
ARMA	**0.00**	**70.18**	**0.00**	**81.36**	**0.00**	**71.67**

(d) Bus percentages having $F1 \leq 5\%$ and $F1 \geq 95\%$.

Fig. 7. Distribution of $F1$ scores for node wise evaluation of localization.

the test systems, the ARMA model can determine the location of an FDIA attack for all buses with $F1$ score greater than 80%. Fig. 7(d) presents the percentages of buses satisfying $F1 \leq 5\%$ and $F1 \geq 95\%$ levels. For all test systems, only ARMA model has 0% with $F1 \leq 5\%$ success level which means only ARMA model doesn't yield any "unacceptable" bus localization performance. In comparison, one bus in IEEE 118- and 5 buses in IEEE 300-bus systems always have $F1$ score less than 5% in all timestamps for the second best CHEB model. For the $F1 \geq 95\%$ threshold, only ARMA model can surpass the 70% level for each test systems and it outperforms the second best model CHEB by 8.78%, 11.87%, and 14.67% for the 95% $F1$ threshold level in NW localization for IEEE 57−, 118−, and 300-bus systems, respectively.

TABLE V
JOINT DETECTION AND LOCALIZATION TIMES IN MILLISECONDS

model	DT	KNN	MLP	LSTM	CNN	CHEB	ARMA
IEEE-57	0.18	147.78	1.42	16.85	2.64	2.24	2.76
IEEE-118	0.29	327.62	1.44	35.19	2.67	2.54	2.81
IEEE-300	0.69	836.52	1.50	99.78	2.73	2.71	2.94

D. Joint Detection and Localization Times

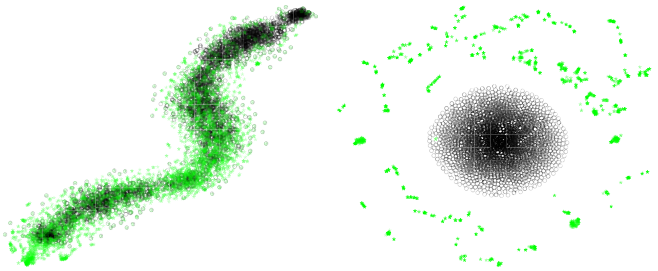
We measure the elapsed time during model's joint detection and localization process for each sample in the test set, calculate the mean values, and tabulate them in Table V.

Clearly, detection times of KNN are not satisfactory: it can take more than 0.8 second to respond. It is due to the fact that in KNN each new sample has to be compared with others for proximity calculation. LSTM, in contrast, provides better results compared to the KNN. Nevertheless, its highly complex recurrent architecture can delay its output almost 0.1 second for IEEE-300, which may limit its application in a real time scenario. All the other models including DT, MLP, CNN, CHEB, and ARMA provide reasonable detection times for a real time application: for all test system their response time is less than 3 milliseconds. Among them DT provides the best detection times with values under 0.7 milliseconds; yet, its poor detection and localization performance hinders its suitability as a reliable method.

E. Visualization of Intermediate Layers With t-SNE

To compare the proposed model with the existing approaches, we analyze and visualize the multidimensional data processed by the intermediate layers of the proposed models. Nevertheless, the high dimensionality of the data severely limits examining them. Besides, examining a specific feature of a bus does not provide enough information to fully comprehend how the model processes the grid. Thus, we transform the layer outputs by using the t-distributed stochastic neighbor embedding (t-SNE), which is a nonlinear dimensionality reduction technique to visualize the high dimensional data in two or three dimensional spaces [25]. By iteratively minimizing the Kullback-Leibler divergence between the probability distributions representing the sample similarities in the original and mapped spaces, it projects samples into the low dimensional space. Thus, it preserves the structure of the data and enables visualization of the data in a lower dimension [25].

Due to space limitations, only models trained for the IEEE-300 bus system are analyzed in two dimensions (2D) with test data having $5,760$ samples. Embedding of input data $[\boldsymbol{P}, \boldsymbol{Q}] = \boldsymbol{X} \in \mathbb{R}^{300\times 2}$ is plotted in Fig. 8(a), where a dominating daily profile can be seen from the smooth transition from the lower left to the upper right samples depicted with green stars (attacked) and black circles (non-attacked). Moreover, the close proximity between attacked and non-attacked samples indicates that the attacked samples preserve similarity to their non-attacked samples. Fig. 8(b) shows the embedding of true output $\boldsymbol{Y} \in \mathbb{R}^{300}$ where non-attacked samples clustered in the middle and attacked samples are scattered around them. This is not surprising since non-attacked samples are



(a) Embedded $\boldsymbol{X}$ in 2D (b) Embedded $\boldsymbol{Y}$ in 2D

Fig. 8. Embedded input and output data for IEEE-300 bus system where attacked and non-attacked samples are depicted with green stars and black circles, respectively.

all formed from 0s and attacked samples include 1s in their corresponding labels to indicate the attacked bus.

Fig. 9 demonstrates the embedding of layer l's output where each model takes $\boldsymbol{X}$ (Fig. 8(a)) as input, transforms it in the hidden layers, and tries to predict $\boldsymbol{Y}$ (Fig. 8(b)) as output. The number of TP (green stars), FP (blue diamonds), FN (red squares), and TN (black circles) samples are given under the model name for easy reference. MLP clearly falls behind the other approaches due to the FNs scattered all around. For instance, unlike other approaches, MLP misses easily detectable attacked samples in l_2 very close to the TP cluster and it maps many FNs nearby to the TNs placed at the lower part of its last layer. LSTM, in contrast to the MLP, reduces FN and FP samples. However, in l_2, it falsely maps many attacked samples adjacent to the non-attacked samples which yields a high number of FNs. In addition, like the MLP, it falsely predicts multiple non-attacked clusters in its final layer.

Contrarily, CNN is one of the best models in terms of FP number. Yet, it "destroys" the structure of data in l_1 which brings a significant number of FNs. We believe it is due to the fact that CNN tries to capture the correlations of non-Euclidean data in an Euclidean space and samples from different classes may look the same in that space. Due to their inherent graph architecture, CHEB and ARMA yield better results since they both consider the "structure" of the data within $\boldsymbol{X}$ in their graph convolutional layers. However, CHEB misses 5 more attacks and yields more than 5 times FP samples compared to the ARMA. For instance, many non-attacked samples in l_4 are falsely regarded as an attack due to close mapping to an attacked cluster. Conversely, our proposed model gives only 4 FP and 1 FN due to its rational graph convolutional filters that provide more flexible frequency responses. Note that no sample is mapped in the vicinity of attacked samples unlike the other models. Besides, only ARMA outputs a highly similar pattern to $\boldsymbol{Y}$: a non-attacked sample cluster in the center and attacked samples distributed around it.

F. Discussions & Theoretical Comparisons

As indicated earlier, two main approaches have been proposed for detecting and localizing the FDIAs: model-based and data driven approaches [8]. Model-based approaches such as those in [10]–[14] do not require any historical datasets. Nevertheless, scalability, manual threshold optimization process, detection lags, model complexity, and localization resolution could hinder the usability of them for real applications. For instance, results are not published in [12] and localization could only be done at the cluster level in [10]. Detection times are larger than a second in [11] and [13] for small test systems having 12 and 36 buses, respectively. In their model-based detectors, authors of [14] utilize GSP techniques such as Local Smoothness (LS) and Vertex-Frequency Energy Distribution (VFED). Nonetheless, they evaluate their method with an easily detectable attack by the classical LNRT based BDD methods which can conceal the actual performance of the model. Specifically, they simulate the FDIAs using $z_a^i(t) = z_o^i(t) + (-1)^d.a.u.range$ where $d \sim \{0, 1\}$ is a binary random variable (r.v.), $u \sim \mathcal{U}[0, 1]$ is an uniform r.v., $range = \max(z_o^i) - \min(z_o^i)$ and a is scaler for the attack. For instance, if $z_o^i \sim \mathcal{N}(\mu_o, \sigma_o^2)$, expected values of the attacked data distribution become $\mathbb{E}[\mu_a] = \mu_o$, and $\mathbb{E}[\sigma_a] = 6a\sigma_o$ due to the product properties of uniform and normal distributions, where μ_o, σ_o and μ_a, σ_a tuples represent the mean and standard deviation of original and attacked data, respectively. The accuracy of localization for IEEE 118-bus test system when $a = 4$, which makes $\mathbb{E}[\sigma_a] = 24\sigma_o$, are 85% and 91% for the LS and VFED techniques, respectively. These high accuries are not realistic since the scaler a plays a significant role in simulation process.

The data-driven methods, in contrast, present a better performance since historical datasets are growing and the modeling capability of these algorithms is being increased [8], [9]. For example, in their data-driven method in [15] researchers employ an LSTM model for each measurement in a 5-bus test system in which only one bus is under attack at a time to detect and localize the point-wise FDIAs. They report greater than 90% accuracy for detection and localization of random, ramp, and scale attacks for low, medium, and high attack levels. However, the capability of this method for detection and localization of different FDIAs in large test cases has not been investigated. Besides, assigning an independent model to each measurement has two major drawbacks: (i) overall model complexity increases severely, and (ii) spatial correlations of the measurements are ignored totally.

In data-driven approaches, compatibility between the structure of collected data and architecture of the data-driven model is the primary factor on the performance of the model. For instance, DT, KNN, or MLP architectures could be better suited for a dataset having uncorrelated features from different spaces. Similarly, an RNN architecture might be more applicable to model the recurrent relations in a natural language data. A CNN architecture, in contrast, could be more favorable than GNN for an image data where pixel locality is well modeled in 2D Euclidean space. However, as demonstrated with Fig. 1(a), spatial correlations in power measurements data can only be captured in a non-Euclidean space dictated by the topology of the grid. For instance, if we had a hypothetical power grid like in Fig. 1(a), a CNN architecture could have comparable performances with ARMA. Nonetheless, for a power grid data collected from graph type structure, a GNN

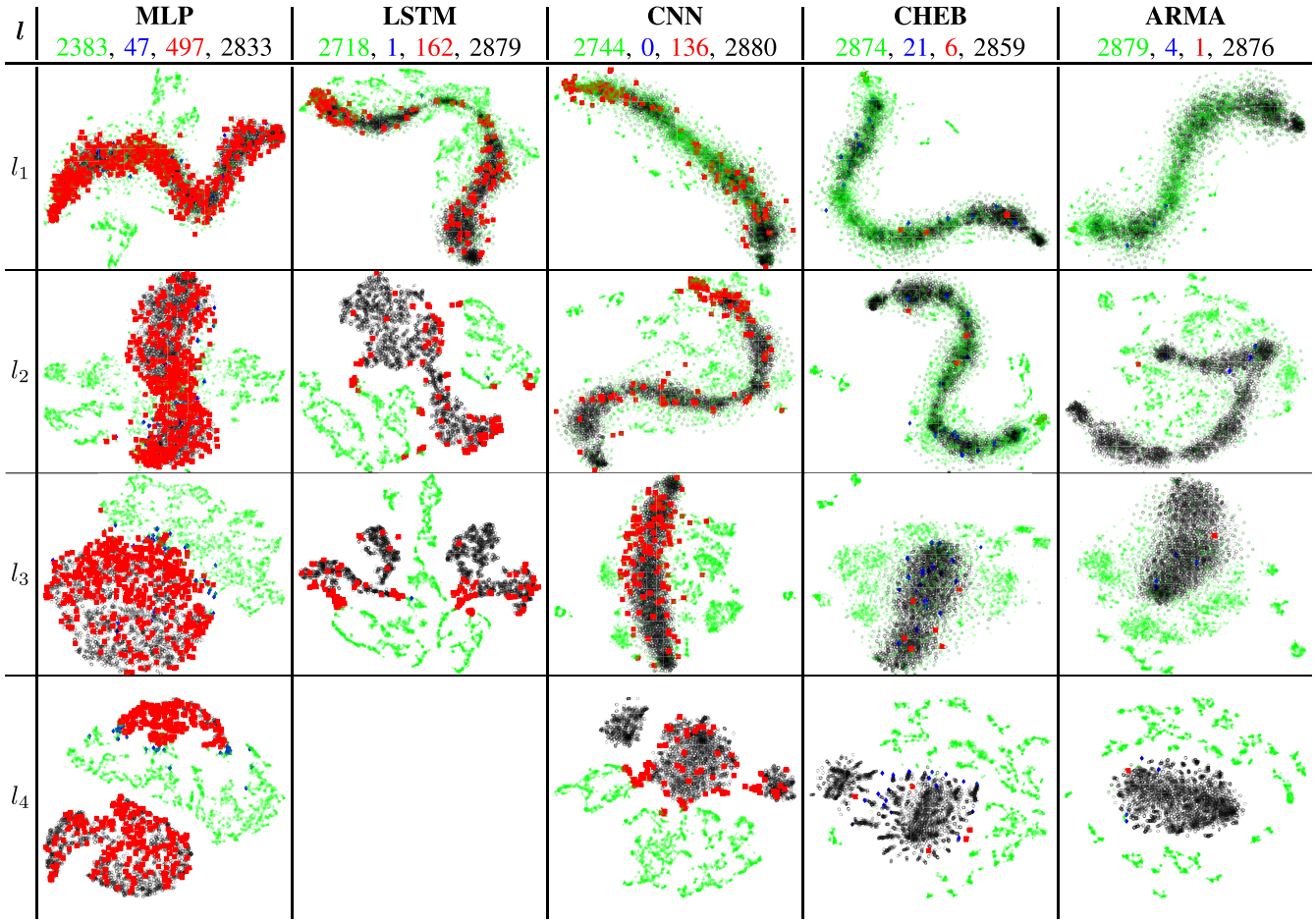


Fig. 9. t-SNE embedding of model's layers to visualize the attack detection where true input and output data are given in Fig. 8. For each model and each layer l, output of the model is embedded in 2D using t-SNE. Since t-SNE preserves the structure of the high dimensional data, models' transformation can be visualized and compared in a lower dimension, such as 2D. Note that due to its topology aware ARMA graph filtering, the proposed model better classifies samples by converging to the true output depicted in Fig. 8(b). As a consequence, it yields the minimum number of FP and FN compared to other models.

is more advantageous than other architectures as can be seen from the detection results in Table IV and the localization distributions in Figs. 6 and 7. As for the GNN family, ARMA outperforms CHEB due to the fact that rational GFs implemented using the ARMA architecture provide more flexible frequency responses compared to the polynomial filters such as CHEB [29].

It is observed from our extensive experiments that the proposed ARMA based model performs better compared to other models for larger test cases. As an illustration, for the 95% $F1$ threshold level, it outperforms the second best model CHEB by 5.64%, 8.56%, and 10.07% in SW localization and by 8.78%, 11.87%, and 14.67% in NW localization for IEEE 57-, 118-, and 300-bus systems, respectively. This difference is due to the fact that in larger and denser graphs, (i) the spatial correlation between adjacent measurements becomes more dominant compared to the global correlations and (ii) ARMA GFs better adapt to abrupt changes in the spectral domain compared to the polynomial ones.

As stated before, the output of each vertex v only depends on its K-hop neighbors for a K-order polynomial GF. In other words, the output of v is independent of the vertices beyond the K-hop neighbors for a K-order FIR GF [22]. Thus, to capture the global characteristics of the graph, an FIR GF requires "high" order spectral response as can be seen from Fig. 4. Nevertheless, due to the poor interpolation and extrapolation capabilities of the high order polynomials, it becomes sensitive to variations and may overfit to the training data [21]. To verify this characteristic, we fix the other parameters of CHEB GF at their optimal values tabulated in Table III and train a CHEB model for the IEEE 300-bus test system for each $K \in \{5, 7, 9, 11\}$. FDIA detection results in terms of $F1\%$ are depicted in Fig. 10. Clearly, increasing K beyond a certain point makes the model susceptible to variations such as noise, and thus, it can degrade the test set performance. Note that similar conclusion can also be corroborated for the localization results.

Bus level localization is a multi-label classification task and should be evaluated accordingly. Besides, performance metrics can cause inaccurate or misleading outcomes when they are not interpreted correctly. Namely, missing an attack (FN) could be much more severe than a false alarm (FP) when dealing with FDIAs due to their consequences. An example of localization results for a hypothetical model is given in Table VI with 4 samples in rows and 5 buses in columns where TP, FP, FN, and TN samples are highlighted with green, blue, red, and

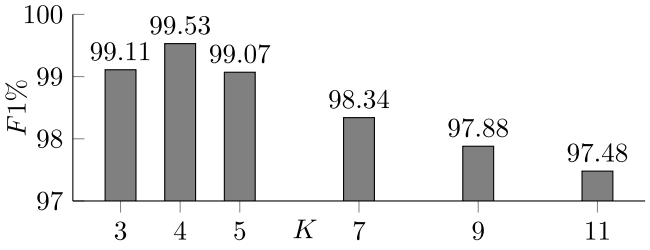


Fig. 10. Detection performance vs. filter order of CHEB models for IEEE 300-bus system. Optimal results are obtained at $K = 4$ as given in Table III.

TABLE VI
SW AND NW LOCALIZATION EXAMPLE IN *ACC*% AND $F1$%

	n_1	n_2	n_3	n_4	n_5	ACC_{sw}	$F1_{sw}$
s_1	tn	fn	tn	tn	tn	80	0
s_2	tn	fn	tn	tn	tn	80	0
s_3	tn	fn	fp	tn	tp	60	50
s_4	tp	fn	tp	fp	tp	60	75
ACC_{nw}	100	0	75	75	100		
$F1_{nw}$	100	0	66	0	100		

black colors, respectively. In addition, SW and NW localization results are given at the end of each row and columns in terms of accuracy $ACC = \frac{TP+TN}{TP+TN+FP+FN}$ and $F1$ percentages.

The ACC_{sw} is not a reliable metric since it can not properly take into account the distribution of errors. For instance, although ACC_{sw} shows high accuracy for all the samples, it does not have any mechanism to mirror the faults at n_2 which can have serious consequences for the power system. Comparing $F1$ with ACC reveals that $F1$ has a better mechanism to evaluate the accuracy of the model. For instance, the $ACC_{sw} = 60\%$ for s_3 and s_4 since they have the same number of falsely predicted samples. $F1_{sw}$ metric, in contrast, yields 50% for s_3 and 75% for s_4 since s_4 includes 1 more TP compared to the s_3. Since our focus is to determine the localization of FDIAs, then $F1$ is the proper candidate to evaluate the accuracy of the model. The result and discussion reveals the supremacy of our model compared to DT, MLP, RNN, CNN, and CHEB models in terms of detection and localization of FDIAs.

V. CONCLUSION

This work proposed a GNN based model by integrating the underlying graph topology of the grid and spatial correlations of its measurement data to jointly detect and localize the FDIAs in power systems while the full AC power flow equations are employed to address the physics of the network. Adopting IIR type ARMA GFs in its hidden layers, the proposed model is more flexible in frequency response compared to FIR type polynomial GFs, e.g., CHEB thanks to their rational type filter composition. Although our algorithm has better detection and localization performance compared to the state of the art CHEB model [24] in the literature, the improvement rate for localization is much higher than detection. Simulations performed on various standard test systems confirm that the performance of the proposed model in detecting FDIA exceeds the performance of CHEB model by 0.12%, 0.68%, and 0.38% for IEEE 57-, 118-, and 300-bus, respectively. The proposed model also outperforms the CHEB model in localizing the attacks (i.e., 95% $F1$ threshold level) by 5.64%, 8.56%, and 10.07% in SW localization and by 8.78%, 11.87%, and 14.67% in NW localization for the same above-mentioned test systems, respectively. Furthermore, visualizing the intermediate layers for different approaches including those in literature corroborates the supremacy of the proposed model in detecting FDIA.

REFERENCES

[1] X. Yu and Y. Xue, "Smart grids: A cyber–physical systems perspective," *Proc. IEEE*, vol. 104, no. 5, pp. 1058–1070, May 2016.
[2] K. R. Davis, K. L. Morrow, R. Bobba, and E. Heine, "Power flow cyber attacks and perturbation-based defense," in *Proc. IEEE 3rd Int. Conf. Smart Grid Commun. (SmartGridComm)*, Tainan, Taiwan, 2012, pp. 342–347.
[3] S. Sridhar, A. Hahn, and M. Govindarasu, "Cyber–physical system security for the electric power grid," *Proc. IEEE*, vol. 100, no. 1, pp. 210–224, Jan. 2012.
[4] A. Abur and A. G. Expósito, *Power System State Estimation: Theory and Implementation* (Power Engineering (Willis)). New York, NY, USA: Marcel Dekker, 2004. [Online].Available: https://books.google.com/books?id=NQhbtFC6_40C
[5] G. B. Giannakis, V. Kekatos, N. Gatsis, S.-J. Kim, H. Zhu, and B. F. Wollenberg, "Monitoring and optimization for power grids: A signal processing perspective," *IEEE Signal Process. Mag.*, vol. 30, no. 5, pp. 107–128, Sep. 2013.
[6] G. Liang, J. Zhao, F. Luo, S. R. Weller, and Z. Y. Dong, "A review of false data injection attacks against modern power systems," *IEEE Trans. Smart Grid*, vol. 8, no. 4, pp. 1630–1638, Jul. 2017.
[7] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," *ACM Trans. Inf. Syst. Security*, vol. 14, no. 1, pp. 1–33, 2011.
[8] A. S. Musleh, G. Chen, and Z. Y. Dong, "A survey on the detection algorithms for false data injection attacks in smart grids," *IEEE Trans. Smart Grid*, vol. 11, no. 3, pp. 2218–2234, May 2020.
[9] A. Sayghe *et al.*, "Survey of machine learning methods for detecting false data injection attacks in power systems," *IET Smart Grid*, vol. 3, no. 5, pp. 581–595, 2020.
[10] T. R. Nudell, S. Nabavi, and A. Chakrabortty, "A real-time attack localization algorithm for large power system networks using graph-theoretic techniques," *IEEE Trans. Smart Grid*, vol. 6, no. 5, pp. 2551–2559, Sep. 2015.
[11] M. Khalaf, A. Youssef, and E. El-Saadany, "Joint detection and mitigation of false data injection attacks in AGC systems," *IEEE Trans. Smart Grid*, vol. 10, no. 5, pp. 4985–4995, Sep. 2019.
[12] E. Drayer and T. Routtenberg, "Cyber attack localization in smart grids by graph modulation (brief announcement)," in *Proc. Int. Symp. Cyber Security Cryptogr. Mach. Learn.*, 2019, pp. 97–100.
[13] X. Luo, Y. Li, X. Wang, and X. Guan, "Interval observer-based detection and localization against false data injection attack in smart grids," *IEEE Internet Things J.*, vol. 8, no. 2, pp. 657–671, Jan. 2021.
[14] M. A. Hasnat and M. Rahnamay-Naeini, "Detection and locating cyber and physical stresses in smart grids using graph signal processing," 2020. [Online]. Available: arXiv:2006.06095.
[15] A. Jevtic, F. Zhang, Q. Li, and M. Ilic, "Physics- and learning-based detection and localization of false data injections in automatic generation control," *IFAC-PapersOnLine*, vol. 51, no. 28, pp. 702–707, 2018.
[16] C. Bishop, *Pattern Recognition and Machine Learning* (Information Science and Statistics). New York, NY, USA: Springer, Jan. 2006.
[17] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021.
[18] A. Ortega, P. Frossard, J. Kovačević, J. M. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proc. IEEE*, vol. 106, no. 5, pp. 808–828, May 2018.
[19] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, May 2013.

[20] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Advances in Neural Information Processing Systems (NIPS)*. Red Hook, NY, USA: Curran, 2016.

[21] F. M. Bianchi, D. Grattarola, L. Livi, and C. Alippi, "Graph neural networks with convolutional ARMA filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Jan. 26, 2021, doi: 10.1109/TPAMI.2021.3054830.

[22] X. Shi, H. Feng, M. Zhai, T. Yang, and B. Hu, "Infinite impulse response graph filters in wireless sensor networks," *IEEE Signal Process. Lett.*, vol. 22, no. 8, pp. 1113–1117, Aug. 2015.

[23] N. Tremblay, P. Gonçalves, and P. Borgnat, "Design of graph filters and filterbanks," in *Cooperative and Graph Signal Processing*. London, U.K.: Elsevier, 2018, pp. 299–324.

[24] O. Boyaci *et al.*, "Graph neural networks based detection of stealth false data injection attacks in smart grids," 2021. [Online]. Available: arXiv:2104.02012.

[25] L. V. D. Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

[26] J. C. Mason and D. C. Handscomb, *Chebyshev Polynomials*. Boca Raton, FL, USA: CRC Press, 2002.

[27] S. W. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*, vol. 14. San Diego, CA, USA: California Techn. Publ., 1997.

[28] H. Föllmer and U. Küchler, "Richard von Mises," in *Mathematics in Berlin*. Basel, Switzerland: Springer, 1998, pp. 111–116.

[29] A. Loukas, A. Simonetto, and G. Leus, "Distributed autoregressive moving average graph filters," *IEEE Signal Process. Lett.*, vol. 22, no. 11, pp. 1931–1935, Nov. 2015.

[30] A. Loukas, M. Zuniga, M. Woehrle, M. Cattani, and K. Langendoen, "Think globally, act locally: On the reshaping of information landscapes," in *Proc. 12th Int. Conf. Inf. Process. Sens. Netw.*, Philadelphia, PA, USA, 2013, pp. 265–276.

[31] E. Isufi, A. Loukas, A. Simonetto, and G. Leus, "Autoregressive moving average graph filtering," *IEEE Trans. Signal Process.*, vol. 65, no. 2, pp. 274–288, Jan. 2017.

[32] J. Zhao, J. Wang, and L. Yin, "Detection and control against replay attacks in smart grid," in *Proc. 12th Int. Conf. Comput. Intell. Security (CIS)*, Wuxi, China, 2016, pp. 624–627.

[33] G. Chaojun, P. Jirutitijaroen, and M. Motani, "Detecting false data injection attacks in AC state estimation," *IEEE Trans. Smart Grid*, vol. 6, no. 5, pp. 2476–2483, Sep. 2015.

[34] S. Tan, W.-Z. Song, M. Stewart, J. Yang, and L. Tong, "Online data integrity attacks against real-time electrical market in smart grid," *IEEE Trans. Smart Grid*, vol. 9, no. 1, pp. 313–322, Jan. 2018.

[35] J. Giraldo, A. Cárdenas, and N. Quijano, "Integrity attacks on real-time pricing in smart grids: Impact and countermeasures," *IEEE Trans. Smart Grid*, vol. 8, no. 5, pp. 2249–2257, Sep. 2017.

[36] New York Independent System Operator (NYISO). (Jul. 2021). *Actual—Historical*. [Online]. Available: https://www.nyiso.com/load-data

[37] M. Ozay, I. Esnaola, F. T. Y. Vural, S. R. Kulkarni, and H. V. Poor, "Machine learning methods for attack detection in the smart grid," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 8, pp. 1773–1786, Aug. 2016.

[38] J. Yan, B. Tang, and H. He, "Detection of false data attacks in smart grid with supervised learning," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Vancouver, BC, Canada, 2016, pp. 1395–1402.

[39] L. Thurner *et al.*, "Pandapower—An open-source python tool for convenient modeling, analysis, and optimization of electric power systems," *IEEE Trans. Power Syst.*, vol. 33, no. 6, pp. 6510–6521, Nov. 2018.

[40] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Nov. 2011.

[41] M. Abadi *et al.*, "TensorFlow: A system for large-scale machine learning," in *Proc. 12th USENIX Symp. Oper. Syst. Design Implement. (OSDI)*, 2016, pp. 265–283.

[42] A. Jindal, A. Dua, K. Kaur, M. Singh, N. Kumar, and S. Mishra, "Decision tree and svm-based data analytics for theft detection in smart grid," *IEEE Trans. Ind. Informat.*, vol. 12, no. 3, pp. 1005–1016, Jun. 2016.

[43] D. Wang, X. Wang, Y. Zhang, and L. Jin, "Detection of power grid disturbances and cyber-attacks based on machine learning," *J. Inf. Security Appl.*, vol. 46, pp. 42–52, Jun. 2019.

[44] Y. Wang, M. M. Amin, J. Fu, and H. B. Moussa, "A novel data analytical approach for false data injection cyber-physical attack mitigation in smart grids," *IEEE Access*, vol. 5, pp. 26022–26033, 2017.

[45] T. O'Malley *et al.*. (2019). *Keras Tuner*. [Online]. Available: https://github.com/keras-team/keras-tuner

[46] J. W. Tukey, *Exploratory Data Analysis*, vol. 2. Reading, MA, USA: Addison-Wesley, 1977.

Osman Boyaci (Graduate Student Member, IEEE) received the B.Sc. degree (Hons.) in electronics engineering in 2013, the B.Sc. degree (Hons.) in computer engineering in 2013, and the M.Sc. degree in computer engineering in 2017 from Istanbul Technical University, Istanbul, Turkey. He is currently pursuing the Ph.D. degree with Texas A&M University, working on graph neural network-based cybersecurity in smart grids.

His research interests include machine learning, artificial intelligence, and cybersecurity.

Mohammad Rasoul Narimani (Member, IEEE) received the B.S. degree in electrical engineering from Razi University, the M.S. degree in electrical engineering from the Shiraz University of Technology, and the Ph.D. degree in electrical engineering from the Missouri University of Science & Technology. He is an Assistant Professor with the College of Engineering, Arkansas State University. Before joining Arkansas State University, he was a Postdoctoral Researcher with Texas A&M University, College Station. His research interests are in the application of optimization techniques to electric power systems.

Katherine R. Davis (Senior Member, IEEE) received the B.S. degree in electrical engineering from the University of Texas at Austin, Austin, TX, USA, in 2007, and the M.S. and Ph.D. degrees in electrical engineering from the University of Illinois at Urbana–Champaign, Champaign, IL, USA, in 2009 and 2011, respectively. She is currently an Assistant Professor of Electrical and Computer Engineering with Texas A&M University.

Muhammad Ismail (Senior Member, IEEE) received the B.Sc. (Hons.) and M.Sc. degrees in electrical engineering (electronics and communications) from Ain Shams University, Cairo, Egypt, in 2007 and 2009, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2013. He is currently an Assistant Professor with the Department of Computer Science, Tennessee Technological University, Cookeville, TN, USA. He was a co-recipient of the Best Paper Awards in the IEEE ICC 2014, the IEEE Globecom 2014, the SGRE 2015, the Green 2016, the Best Conference Paper Award from the IEEE TCGCN at the IEEE ICC 2019, and IEEE IS 2020.

Thomas J. Overbye (Fellow, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering from the University of Wisconsin Madison, Madison, WI, USA. He is currently with Texas A&M University, where he is a Professor and holder of the O'Donnell Foundation Chair III.

Erchin Serpedin (Fellow, IEEE) is a Professor with the Electrical and Computer Engineering Department, Texas A&M University, College Station. He has authored four research monographs, one textbook, 17 book chapters, 170 journal papers, and 270 conference papers. His current research interests include signal processing, machine learning, artificial intelligence, cyber security, smart grids, and wireless communications. He served as an Associate Editor for more than 12 journals, including journals such as the IEEE TRANSACTIONS ON INFORMATION THEORY, IEEE TRANSACTIONS ON SIGNAL PROCESSING, IEEE TRANSACTIONS ON COMMUNICATIONS, IEEE SIGNAL PROCESSING LETTERS, IEEE COMMUNICATIONS LETTERS, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, *IEEE Signal Processing Magazine*, and *Signal Processing* (Elsevier), and as a technical chair for six major conferences.