Detection and Localization of Load Redistribution Attacks on Large-scale Systems

Andrea Pinceti, *Student Member*, *IEEE*, Lalitha Sankar, *Senior Member*, *IEEE*, and Oliver Kosut, *Member*, *IEEE*

***Abstract*—A nearest-neighbor-based detector against load redistribution attacks is presented. The detector is designed to scale from small-scale to very large-scale systems while guaranteeing consistent detection performance. Extensive testing is performed on a realistic large-scale system to evaluate the performance of the proposed detector against a wide range of attacks, from simple random noise attacks to sophisticated load redistribution attacks. The detection capability is analyzed against different attack parameters to evaluate its sensitivity. A statistical test that leverages the proposed detector is introduced to identify which loads are likely to have been maliciously modified, thus, localizing the attack subgraph. This test is based on ascribing to each load a risk measure (probability of being attacked) and then computing the best posterior likelihood that minimizes log-loss.**

***Index Terms*—Attack detection, cyber-security, false data injection (FDI) attack, load redistribution attack, machine learning, nearest neighbor.**

I. Introduction

THE power grid is a constantly evolving cyber-physical system, and thus it is increasingly reliant on information and communication technology. A vast research effort undertaken in the past decade in the field of cyber-security of power systems has identified some crucial vulnerabilities of the cyber layer which can be exploited to disrupt the physical system. In this context, [1] shows that state estimation (SE) and the traditional bad data detector (BDD) used in energy management systems (EMSs) can be easily spoofed and bypassed via false data injection (FDI) attacks. This finding represents the basis for the design of a wide class of the attacks called load redistribution (LR) attacks. LR attacks can be performed by injecting intelligently designed false measurements that lead to a wrong estimate of the system state. From the perspective of the operators, the attack makes it appear as if the system loads have changed from their actual values, without changing the net load.

In [2] and [3], the concept of LR attacks is formalized by developing a bi-level attacker-defender problem for targeted attacks. In this setting, the attacker can design false measurements which can cause physical consequences on the system. Specifically, [3] attempts to find an attack, which is unobservable to the EMS, to cause an overload on a target line. In a similar fashion, [4] and [5] present the examples of LR attacks on the electricity market, which show that it is possible to launch LR attacks that create system congestion, thus manipulating locational marginal prices.

We propose a new BDD that can identify LR attacks based on the analysis of load estimates, thus overcoming the limitations of SE and the traditional BDD. In [6], we develop three anomaly detectors, each based on a different machine learning technique: replicator neural network, support vector machine, and nearest neighbor. These detectors can effectively determine if the observed loads represent a normative system state or if they have been maliciously modified. The nearest neighbor-based detector works by finding the near load vector close to the real-time loads in the historical data. Based on the measured Euclidean distance, a thresholding technique is used to decide if the loads are normative or anomalous. From the tests, the nearest-neighbor-based detector demonstrates the best performance out of the three detectors. In this paper, we build this preliminary work to design an improved nearest-neighbor-based detector and an attack localization scheme. The novel contributions of this paper are as follows.

1) The basic detector is modified so that it scales to much larger power system models while preserving the good detection performance shown in [6]. This is achieved by devising a grouping strategy to organize the system loads into the clusters that can be analyzed independently.

2) Extensive testing and sensitivity analysis are performed to evaluate the performance of the detector against intelligently designed LR attacks as well as random anomalous load changes. This allows for the characterization of the strengths and limitations of the detector. Furthermore, the proposed detector is integrated within a complete EMS platform to showcase its detection performance and computation efficiency.

3) On the basis of the proposed detector, a statistical ap-

Manuscript received: February 18, 2020; revised: June 4, 2020; accepted: October 26, 2020. Date of CrossCheck: October 26, 2020. Date of online publication: February 9, 2021.

This work was supported by the National Science Foundation (No. CNS-1449080, No. OAC-1934766) and the Power System Engineering Research Center (PSERC) under projects S-72 and S-87. We would like to thank Mr. Zhigang Chu at Arizona State University (ASU) for providing access to the attack design code.

A. Pinceti (corresponding author), L. Sankar, and O. Kosut are with the School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, USA (e-mail: apinceti@asu. edu; lalitha. sankar@asu. edu; oliver. kosut@asu.edu).

DOI: 10.35833/MPCE.2020.000088

proach is presented to localize the attacks and determine the likelihood of each load being attacked. The deviation in loads is captured via a *Z*-score and the log-loss is used as a measure to find the likelihood function that minimizes the error. This represents a crucial step towards the development of decision tools that can help operators to securely manage power systems when targeted by cyber-attacks.

Related work on the design of FDI and LR attack detectors can be found in the literature. For example, in [7], multiple linear regression is used to study the voltage profiles in a system and determine if an LR attack is taking place. Unlike this paper, the method proposed in [7] is designed for distribution systems, and the attacks tested are not realistic as they simulate the changes in loads up to 100%. Other work focuses on using deep neural networks to learn the temporal correlation which exists between the real-time measurements and previous samples [8], or verifying the statistical behavior of the estimated states over time [9]. The assumption on which these detectors are built is that when an attack is injected, the false measurements are not compatible with the dynamics observed from the previous measurements, thus making it possible to flag them as attacked. On the basis of this, a slow ramping attack which only slightly changes the system state at each sampling time will not be detected to a large extent. Moreover, these detectors are tested on limited attack scenarios and their performance is not verified against multiple classes of attacks. Finally, while many attack detectors have been proposed, the idea of detecting FDI attacks by identifying patterns in the observed loads has not been explored before.

The rest of this paper is organized as follows. A description of LR attacks and how to design them is presented in Section II. The basic detection algorithm presented in [6] is summarized and its performance limitations on large-scale systems are shown in Section III. The required improvements are described in Section IV and the detection results on a wide range of LR attacks are presented in Section V. The statistical analysis that leverages the improved nearest-neighbor-based detector to determine the buses that have been attacked is illustrated in Section VI. Finally, conclusions are drawn in Section VII.

II. Attack Model and Design

For a power system, the relationship between the measurement vector $\boldsymbol{z}$ and the state vector $\boldsymbol{x}$ can be written as:

$$\boldsymbol{z} = \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{e} \tag{1}$$

where $\boldsymbol{h}(\cdot)$ is the non-linear relationship function between measurements and states (usually, complex bus voltages); and $\boldsymbol{e}$ is the vector of random measurement noises. As shown in [1], an unobservable attack can be constructed by replacing the original measurement vector $\boldsymbol{z}$ with a corrupted set of measurements $\bar{\boldsymbol{z}}$ as:

$$\bar{\boldsymbol{z}} = \boldsymbol{h}(\boldsymbol{x} + \boldsymbol{c}) \tag{2}$$

where $\boldsymbol{c}$ is the vector of attack states. Based on this fundamental result, [3] presents a bi-level optimization problem to compute $\boldsymbol{c}$ that will maximize the power flow on a specific target line. To cause such physical consequences on the power system, the false measurements must be designed in such a way that they will initiate a system response in the form of generation redispatch. This can be done by creating an unobservable attack that will lead SE to wrongly estimate the system loads, thus causing a wrong dispatching solution. The bi-level optimization problem proposed in [3] is improved in [10] to make it more efficient and scalable to large-scale systems. The first level models the attacker's choice of attack to maximize the overload on a target line; and the second level models the system response to the attack via a direct-current optimal power flow (DCOPF) to observe the resulting physical consequences. In designing the false measurements, the attacker is limited on how much the false loads can deviate from the real loads, which is represented by the load shift factor that represents the maximum percentage by which any load can be modified. This constraint comes from the fact that an operator would easily identify a large change in load over a short period as an anomaly. In the existing literature, load shift factors ranging from 10% [3], [10] to 50% [2] are considered as the maximum allowable values for an attack to remain unobservable. The attack detector presented in this paper aims at identifying in real time if the set of measured loads is genuine or if it is the result of an attack on SE. As shown below, the proposed detector is effective in identifying attacks with relatively small load shift factors and it reaches perfect detection for attacks with a load shift factor of 15% or higher.

III. Basic Detection Algorithm

A. Small-scale Systems

The proposed detector works by analyzing the correlation structure within the currently observed load values and comparing it with the attack-free historical load data. The measured load configuration to be tested is given as an input to the detector which generates a scalar value. This value is then compared against a threshold τ to label the loads as normative or attacked. To evaluate the detection performance, two metrics are used: ① detection probability, which is the ratio between the number of cases correctly labelled as attacked and the total number of attacked cases; ② false alarm rate, which is the number of normative cases labelled as attacked that is divided by the total number of normative cases. The specific value of the threshold is chosen as a tradeoff between detection probability and false alarm rate. The proposed algorithm can be considered as a semi-supervised learning problem since the detectors are trained only on normative data which are already widely available to the operators. Since no attacked data are needed in the training phase, the detectors will not be biased towards specific types of attacks. Given the almost identical detection capability of the three detectors tested in [6], in this study, the nearest-neighbor-based detector is chosen for its computation and explanatory simplicity.

Nearest-neighbor algorithms are based on the assumption that the data labelled as normative lie in limited, dense regions of space while anomalies are located further from these neighborhoods [11], [12]. Let us define $\boldsymbol{p} \in \mathbb{R}^n$ as the

vector of observed load values to be tested, where n is the number of loads in the power system. The normative data are represented by the set $\boldsymbol{P}_N^{\text{hist}} \in \mathbb{R}^{n \times n_h}$ of historical load vectors $\boldsymbol{h}_i \in \mathbb{R}^n$ that have been observed in the past, where n_h is the total number of historical vectors. The classification is done by measuring the Euclidean distance between the current load profile $\boldsymbol{p}$ and every vector $\boldsymbol{h}_i$ in the historical dataset (assumed to be attack-free). The nearest-neighbor distance d for sample $\boldsymbol{p}$ is defined as:

$$d = \min_{i=1,2,\ldots,n_h} \left\| \boldsymbol{p} - \boldsymbol{h}_i \right\|_2 \tag{3}$$

To label $\boldsymbol{p}$ as normative or attacked, d is compared against a pre-determined threshold τ.

In [6], we test the nearest-neighbor algorithm on the IEEE 30-bus system. Publicly available zonal historical load data from the PJM system [13] is mapped into the loads of the IEEE 30-bus system to create hourly load profiles for 5 consecutive years. The detector proposed in [6] shows very high detection capability with low false alarm rates. Figure 1 shows some of the results obtained on this small-scale system [6]. The blue symbol represents the minimum distance for the normative load vectors (not attacked), while the green and red symbols represent the distances corresponding to attacked cases with load shift factors of 10% and 15%, respectively. This illustrates how loads resulting from attacks lead to much longer nearest-neighbor distances compared with normative load profiles, which demonstrates that the minimum distance is an effective metric for attack detection.

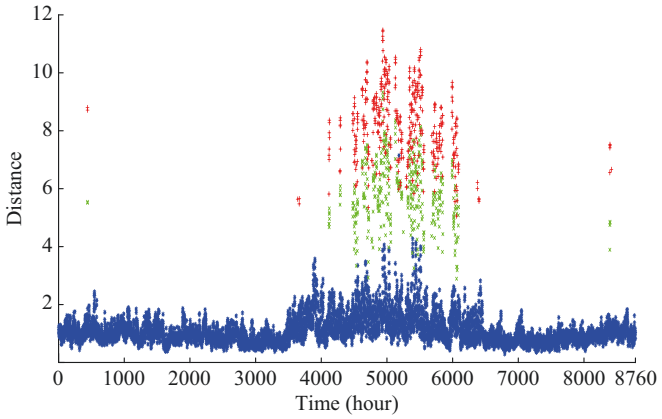

Fig. 1. Distribution of nearest-neighbor distance for normative and attacked cases in IEEE 30-bus system.

B. Large-scale Systems

While the results obtained on the IEEE 30-bus system are promising, the detector proposed in [6] needs to be tested on large-scale systems to verify its performance in a more realistic setting and to guarantee its suitability for the implementation in real system operations. To this end, the same analysis presented in [6] and summarized in the previous section is performed on the synthetic Texas system [14], [15]. This system, developed at Texas A&M University, is a synthetic power system of the state of Texas. It has 2000 buses, 3206 branches, and 1125 loads and it includes bus-level hourly load data for the year 2016. Using the attack model described in Section II, around 280 attacks with load shift factor of 15% have been designed on the most congested cases. We randomly select 90% of the 8784 normative load vectors to represent the historical data, and the remaining 10% for testing. The nearest-neighbor algorithm is used to compute the minimum distance for the test and the attacked load vectors against the historical load data.

Figure 2 shows the minimum distance for each normative load vector (blue symbol) and for the attacked cases with load shift factor of 15% (red symbol). It is easy to see that the detector proposed in [6] does not perform well, and that the attacked cases are indistinguishable from the normative ones. This can be explained by the fact that when measuring the Euclidean distance between two high-dimension vectors, the contribution of a limited subset of dimensions is small. If only a few tens of loads are attacked, the total distance measured over hundreds of loads will deviate only slightly from the distance computed on the load vector where no loads are modified. In this case, each load vector has dimensions of 1125 and the attacks modify only about 100 to 200 loads; and the effect of the attacked loads is not large enough to result in distance values significantly higher than those of the normative data.

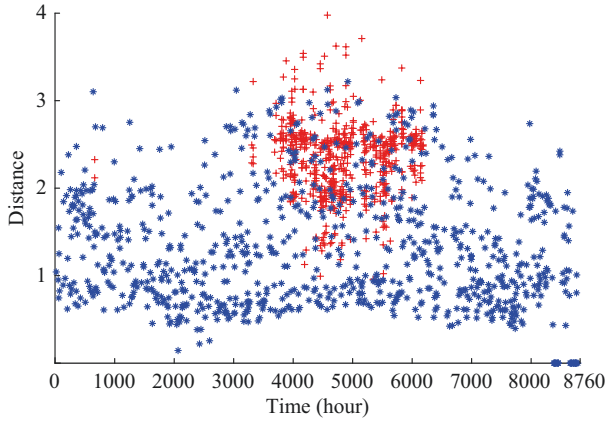

Fig. 2. Distribution of nearest-neighbor distance for normative and attacked cases in synthetic Texas system.

IV. Detection on Large-scale Systems

The simple test presented in the previous section shows that the basic nearest neighbor-based detector introduced in [6] does not perform well when applied to large-scale systems with hundreds or thousands of buses. Therefore, we need a new algorithm to improve the detection mechanism to be effective for any system, regardless of its size. The new algorithm aims to leverage the capability of the nearest-neighbor algorithm to identify anomalous loads even when only a small fraction of the total system loads is being attacked.

Previous work has shown that in a large transmission-level system, LR attacks tend to target only some portions of the network. As a consequence, the loads which are modified represent a subset of the total system loads and they are restricted to a subgraph of limited size. Based on these observations, the detection algorithm is modified so that it analyzes multiple pre-defined subsets of the system loads. In this paper, we propose a grouping strategy that can be used

to divide the loads into relatively small groups so that they can be analyzed independently and in parallel by the attack detector. It is important to notice that the presented strategy is just one example of grouping which empirically works well for the systems tested. Different grouping strategies, which may leverage specific knowledge and insights regarding the power system to be secured, can be easily implemented within the framework of the proposed detector.

A. Grouping Strategy

The first step required to define the load groups is to sort the loads based on their megawatt rating from the largest to the smallest. Starting from the largest load, the first group is created by including the load itself and all its neighboring loads within a certain radius r_g, where the radius is measured as the smallest number of branches connecting two loads. At this point, the next largest load is selected and if it is not contained in any of the previous groups, a new group is created. This process is repeated until n_g groups are created. Note that it is possible for a bus to be contained in multiple groups, or no groups. The parameters r_g and n_g have a direct effect on the detection performance and their selection will be discussed in the next sections. As our results show, this grouping strategy proves to be very effective in the detection of LR attacks because it ensures that the largest loads in a system are monitored. Prior work on FDI attacks on SE shows that, to cause significant consequences, an attacker is required to target large loads in order to create large power flow changes [2]-[5].

B. Detection Algorithm

Dividing the n system loads into groups allows us to overcome the dimensionality issue observed in Section III-B. The basic nearest-neighbor-based detector can be used on large-scale systems by running the nearest-neighbor algorithm individually on each load group. In this case, a threshold τ_j must be defined for each individual group g_j, for $j \in \{1, 2, \ldots, n_g\}$. The vector $\boldsymbol{p} \in \mathbb{R}^n$ containing the estimated loads computed by SE is divided into n_g groups according to the procedure described in the previous subsection. $\boldsymbol{p}^j$ is defined as the vector containing the real-time values of the loads in group g_j. For each group, the minimum distance between the load vector $\boldsymbol{p}^j$ and the corresponding loads in the historical dataset is calculated as:

$$d_j = \min_{r=1,2,\ldots,n_h} \left\| \boldsymbol{p}^j - \boldsymbol{h}_r^j \right\|_2 \tag{4}$$

where $\boldsymbol{h}_r^j$ is the subset of loads belonging to group g_j from the r^{th} historical load vector. The minimum distance is then compared with the threshold τ_j to determine if the loads in group g_j are normative or anomalous. Specifically, if $d_j > \tau_j$, an alarm is raised, while if $d_j < \tau_j$ the loads are considered attack-free. This process is repeated for every group and if one or more alarms are raised, the load vector $\boldsymbol{p}$ is labelled as anomalous.

V. Test of Proposed Detector

A. Experimental Procedure

The performance of the proposed detector in conjunction with the grouping strategy is tested in depth in the following subsections. The detection capability is measured both on intelligently designed attacks as well as random LR attacks; moreover, we study its sensitivity to different parameters such as the load shift factor of the attack and the number of attacked buses.

The goal of the following experiments is to analyze the quality of the detector by understanding if a load vector is normative or attacked. The primary test system used is the synthetic Texas system described in Section III-B, and all numerical results discussed below are based on this system. Additional testing performed on the 2383-bus Polish test case [16], for which we generate historical load profiles based on real data from a major US ISO [17], shows comparable results and is omitted in this paper due to space constraints.

First, the 1125 system loads in the Texas system are divided into groups following the procedure from Section IV-A. For the tests described below, the parameters chosen for the creation of the groups are $r_g = 7$ and $n_g = 35$, which ensure that more than 60% of the loads in the system are included in one or more groups and the ones that are outside of the groups have at least one monitored neighboring load. Moreover, these load groups are equally spread across the system; as a result, the system is effectively monitored in its entirety. Preliminary testing has shown that increasing the number of groups and thus of the loads considered does not improve detection performance.

In each experiment, two datasets are needed: the normative load dataset $\boldsymbol{P}_N \in \mathbb{R}^{1125 \times 8784}$ and the anomalous load dataset $\boldsymbol{P}_A \in \mathbb{R}^{1125 \times H}$, where H varies for different types of the attacks. The normative data represent one load vector for each hour of 2016 (2016 was a leap year). The dataset $\boldsymbol{P}_A$ contains attacked load vectors which are designed starting from the normative load vectors in dataset $\boldsymbol{P}_N$. Depending on the type of the attack, some of the loads are modified either intelligently or randomly, as described below.

To compute detection probability and false alarm, the load vectors of dataset $\boldsymbol{P}_N$ are first divided into three subsets: historical, training, and testing. The historical subset $\boldsymbol{P}_N^{\text{hist}}$ includes 70% of the total hours of 2016 and it represents the past loads known to the system operator and used in its nearest-neighbor algorithms. The training subset $\boldsymbol{P}_N^{\text{train}}$ represents another 20% of $\boldsymbol{P}_N$ and it is needed to determine the thresholds τ_j for each load group. The remaining 10% of normative load vectors is used as the testing subset $\boldsymbol{P}_N^{\text{test}}$ to determine the false alarm rate. To determine the threshold τ_j for group g_j, the minimum distance $d_{i,j}$ between each load vector $\boldsymbol{p}_i^j$ for time i in $\boldsymbol{P}_N^{\text{train}}$ and the historical subset is computed using (4). The threshold τ_j is defined as a fixed fraction of the maximum nearest-neighbor distance $d_{\max,j}$, which is defined for each group as:

$$d_{\max,j} = \max_{\boldsymbol{p}_i^j \in \boldsymbol{P}_N^{\text{train}}} d_{i,j} \tag{5}$$

For each load vector in $\boldsymbol{P}_N^{\text{test}}$, the minimum distance from $\boldsymbol{P}_N^{\text{hist}}$ is computed and compared with the threshold. The false alarm rate is the ratio between the number of times a load vector is labelled as attacked, e. g., at least one load group has the minimum distance greater than its corresponding threshold, and the total number of load vectors in $\boldsymbol{P}_N^{\text{test}}$. Simi-

larly, the minimum distance is calculated for every attacked case and the detection probability is computed. As we will explain in more detail in the next sections, varying the threshold about the value $d_{\max,j}$ allows to span different detection probabilities and false alarm rates in order to determine the receiver operation characteristic (ROC). The proposed detector is extremely efficient and testing a load vector only takes a fraction of a second on a normal laptop. Thus, even on large-scale systems, the detector can easily run in real time.

Since the normative load dataset is limited to one year, in order to have a more complete assessment of the performance of the detector, a k-folding technique is used to test every hour of the year by rotating through multiple sets of historical, training, and testing datasets. The hours of 2016 are randomly divided into ten equally sized partitions as illustrated in Fig. 3. The partitions are fixed throughout the testing process. For the 1st fold, the load vectors corresponding to the hours in the first partition are assigned to $\boldsymbol{P}_N^{\text{test}}$, those corresponding to the hours in the 2nd and 3rd partitions are assigned to $\boldsymbol{P}_N^{\text{train}}$ and those corresponding to the hours in the remaining partitions are assigned to $\boldsymbol{P}_N^{\text{hist}}$. Given these partitions, the number of false alarms and the number of the detected attacks are calculated on the normative and attacked load vectors in the testing partition. The subsequent folds are created by shifting the partitions assigned to the three datasets by one: for example, in the 2nd fold, $\boldsymbol{P}_N^{\text{test}}$ will coincide with the 2nd partition, $\boldsymbol{P}_N^{\text{train}}$ will coincide with the 3rd and 4th partitions, and $\boldsymbol{P}_N^{\text{hist}}$ will coincide with the remaining ones. The final detection probability is then calculated by adding up the correctly identified attacks across all folds and dividing by the total number of attacks. The false alarm rate is the total number of false alarms divided by 8784.

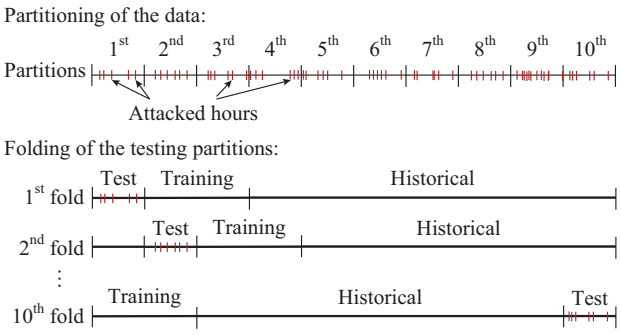

Fig. 3. Description of k-folding technique and definition of datasets.

B. Detection of Intelligently Designed Attacks

We use the bi-level problem in [3] to design the attacks that simulate specific changes in loads to cause physical overflows on a target line, while being unobservable to the system operators (and SE). The testing procedure described in the previous sections is employed here to verify the ability of the proposed detector and grouping strategy in correctly identifying malicious loads resulting from these intelligently designed attacks.

The bi-level problem in [3] is structured so that any one branch can be selected as a target, and an attack will be designed to maximize the flow on it. Depending on the specific system conditions, a successful attack (i. e., one causing the resulting power flow to exceed the branch rating) may not exist; generally, the higher the pre-attack flow is, the more likely the attack will lead to overflow. Therefore, the first step in designing the attacks is to run an AC optimal power flow (ACOPF) for every load vector in $\boldsymbol{P}_N$ to identify any congested branch. For the purpose of this study, a congested branch is any line or transformer that has a base-case power flow loading of 90% of its rating or more. The attacks are designed on each hour of 2016 for which one or more branches are congested. These branches are individually selected as the targets of the attacks. Thus, an hour will have as many different attacks as the number of branches with base-case flow above 90% in that hour. Moreover, for each target branch, the attacks are designed with a load shift factor ranging from 1% to 15% in steps of 1%. This allows us to study how the detection performance varies in relation to the attack magnitude. As a result of this process, 8861 successful attacks are computed, across every hour, target line, and load shift factors.

The resulting attacked load vectors have been tested following the k-folding procedure in Section V-A, where the threshold for each group g_j varies from $0.9d_{\max,j}$ to $1.1d_{\max,j}$. Figure 4 shows the detection probability as a function of the load shift factor and the false alarm rate.

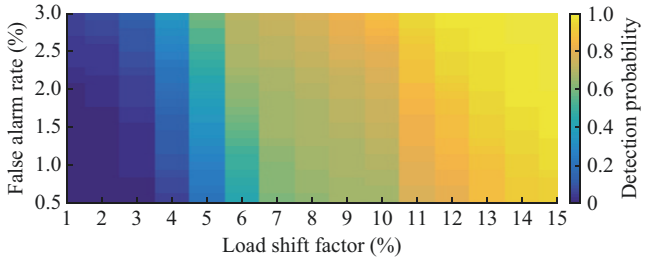

Fig. 4. Detection probability as a function of load shift factor and false alarm rate for intelligently designed attacks.

It can be observed that the detector does not perform well on the attacks with very low load shift factors, while for load shift factor between 10% and 15%, the detection probability goes from 0.8% to 1.0% with false alarm rates ranging from 0.5% to 3%. While the load shift factor is an important metric in the design phase of the attacks, from the perspective of operators, it is more meaningful to evaluate the physical consequences of the attacks. Figure 5 shows the detection probability as a function of the load shift factor and false alarm rate.

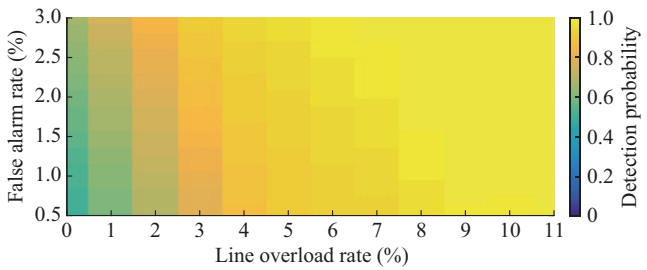

Fig. 5. Detection probability as a function of line overload and false alarm rate for intelligently designed attacks.

As shown in Fig. 5, we can easily observe that the detector has extremely high probability of detecting any attack that would cause important physical damage. Considering the safety margins built into the operation tools, an overload rate of 2% or 3% is not likely to cause any system disruption.

C. Detection of Random LR Attacks

The experiments in the previous subsection have shown that the proposed detector is effective in identifying the attacked load vectors designed to create significant overflows on specific target lines. In this subsection, the sensitivity of this algorithm is investigated to anomalous loads which have not necessarily been intelligently designed. Thus, a large number of false load vectors will be created based on the historical data. The detection performance is then computed as the number of modified loads and the amount of load change are varied across a broad spectrum.

The false load vectors are created by randomly selecting a subset of the loads in each vector of $\boldsymbol{P}_N$ and modifying them by either increasing or decreasing their values by a given load shift factor. In this study, the same load shift factors as in the previous subsection are used, while the footprint size of the attack as a percentage of the total number of system loads varies between 10% and 100% in steps of 10% for every hour. The resulting anomalous load dataset $\boldsymbol{P}_A$ has dimensions of $1125 \times H$, where $H = 8784 \times 15 \times 10 = 1317600$.

Similar to what is done in the previous subsection, all these false load vectors are fed to the proposed detector and the detection probability is computed. In this case, the detection probability is a function of three parameters: the false alarm rate, the load shift factor, and the footprint size. Figure 6 shows the detection probability as a function of the load shift factor and the footprint size with false alarm rates of 5.5% and 0.4% for random LR attacks. Clearly, for a given load shift factor and footprint size, the detection probability is higher when the false alarm rate is higher.

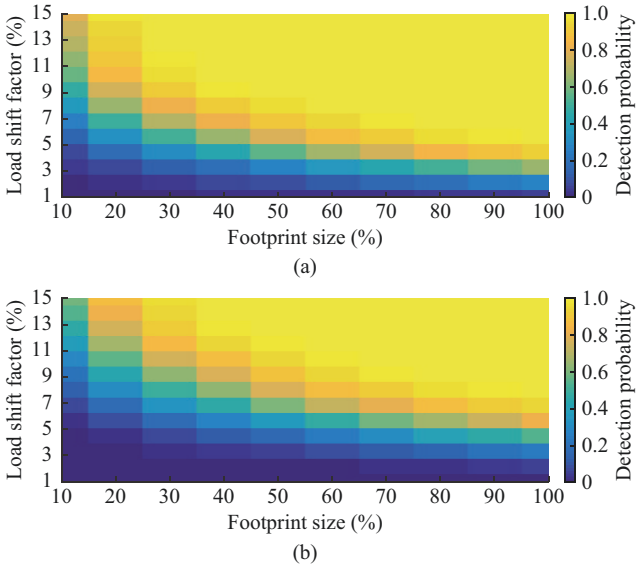

Fig. 6. Detection probability as a function of load shift factor and footprint size with false alarm rates of 5.5% and 0.4% for random LR attacks. (a) False alarm rate is 5.5%. (b) False alarm rate is 0.4%.

Overall, the proposed detector performs well, having perfect detection capability for a wide range of different attacks. Compared with the detection performance on intelligently designed attacks, the proposed detector is not as good as identifying the random LR attacks with small load shift factor and small percentages of the attacked loads. This can be explained by the fact that the intelligently designed attacks are designed in such a way that the modified loads belong to a spatially concentrated subgraph, thus it is likely that some of the load groups will include a large number of the attacked loads. In the random LR attacks, the loads are modified across the whole network, and hence distributed across a higher number of groups. Therefore, each group will experience a smaller deviation from the normative data, resulting in worse detection capability. Meanwhile, the random LR attacks are less likely to cause line overloads.

Figure 7 shows the detection probability as a function of line overload rate and false alarm rate. It can be observed that any random LR attack that would result in line overloads is easily detected, demonstrating the high effectiveness of the proposed detector in detecting anomalous and dangerous load vectors.

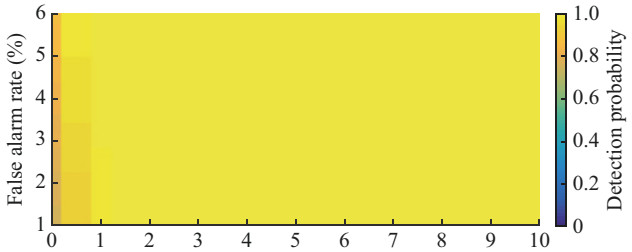

Fig. 7. Detection probability as a function of line overload rate and false alarm rate for random LR attacks.

D. Integration Within EMS

The proposed detector has been fully implemented in a state-of-the-art EMS platform developed at Arizona State University (ASU) [18]. This software was created as part of National Science Foundation (NSF) Grant 1449080 [19], [20]. The interface of the platform is shown in Appendix A Fig. A1. On the left of Fig. A1 is the network graph of the Texas system, while, on the right, the simulation page with the main blocks of the EMS is presented. In Fig. A1, the "Thr" represents the threshold of the chi-square test in the traditional bad data detector; and the "Err" represents the total state estimation error for the current set of measurements. In the example shown, the traditional residue-based BDD has easily been bypassed, while the proposed detector identifies the attack and gives information on the extent of the attack based on the number of groups that raise a flag. Overall, this platform allows for the testing of the detector in a realistic power system operation environment while showcasing its effectiveness in terms of computational efficiency and integration within EMSs. Details on the design of the software platform and its building blocks can be found in [21], while the code for the attack detection algorithm is freely available on Github [22].

VI. Attack Localization

In the previous sections, we introduce a BDD based on the nearest-neighbor algorithm and a grouping strategy which has excellent performance against both intelligently designed attacks and random LR attacks. This nearest-neighbor-based detector can be extended beyond simply determining whether a load vector contains anomalous data or not. It can be leveraged to determine which buses have been modified or are deviating from their usual behavior. Localizing the subgraph affected by an attack or load anomaly represents a step forward in terms of system operation security. Knowing which loads are likely to cause the detector to raise an alarm is an important step in the implementation of secure EMS functionalities. For example, the load values which are determined to be unreliable could be replaced by forecasted values or an uncertainty margin assigned to them so that the system could be operated in a secure state.

A similar approach for secure operations against cyber-attacks is studied in [23], where an optimal dispatching problem is presented to find a secure and cost-effective dispatching solution considering variable bus loads, and thus protecting the system from unexpected load changes. Also, in [24], a secure unit commitment (UC) problem is formulated so that in case of a cyber-attack the system operator can switch from the normal UC solution to a secure one while following all network constraints. The issue with these approaches is that it would cause the system to be operated in a too conservative, and thus less efficient state for most of the time. The advantage of being able to detect and localize an attack is that the system operator can make a better informed decision on when and how to secure the system, without impacting normal operations.

A. Likelihood Determination

The grouping strategy provides an approximate way of localizing the attacks by identifying groups of loads that deviate from their normative behavior. In this subsection, we describe a statistical approach to further analyze the values of the individual loads to identify which ones are more likely to trigger the detector. Because of many possible attack subgraphs, determining exactly which are the attacked loads would be extremely hard. Therefore, our goal is to assign to each load a probability q_l that represents the likelihood of the load being attacked. In this sense, the likelihood is a risk measure and it can be quantified using an empirical metric that relies on estimated likelihoods, namely average log-loss (also known as cross-entropy) [25]. Average log-loss is defined as:

$$l = \frac{1}{n_L}\sum_{l=1}^{n_L} -[y_l \log_2(q_l) + (1-y_l)\log_2(1-q_l)] \tag{6}$$

where n_L is the total number of samples, e.g., loads tested; q_l is the probability associated with each load; and y_l is 1 if the load is indeed attacked and 0 otherwise.

We define the values of the loads in group g_j at time i as $\boldsymbol{p}_i^j = [p_{i,1}^j, p_{i,2}^j, \ldots, p_{i,k_j}^j]^{\mathrm{T}}$, where k_j is the number of loads in group g_j. The minimum distance $d_{i,j}$ between the load vector $\boldsymbol{p}_i^j$ and the historical data are computed using (4). As explained in Section IV-B, if $d_{i,j}$ is greater than threshold τ_j, the group g_j may raise a violation at time i. Moreover, define the loads in the nearest-neighbor of $\boldsymbol{p}_i^j$ as $\boldsymbol{h}_r^j = [h_{r,1}^j, h_{r,2}^j, \ldots, h_{r,k_j}^j]^{\mathrm{T}}$. For each load in group j, the normalized difference between load l at time i and its corresponding value in the nearest-neighbor $h_{r,l}^j$ is computed as:

$$\delta_{i,l}^j = \left| \frac{p_{i,l}^j - h_{r,l}^j}{h_{r,l}^j} \right| \quad l = 1, 2, \ldots, k_j \tag{7}$$

We cannot directly know if a load is attacked through this normalized difference because different loads could have different amounts of deviation. In order to account for this variability, we determine the normative behavior of each load by computing the first- and second-order statistics of its normalized difference $\mu_{\delta_l^j}$ and $\sigma_{\delta_l^j}$:

$$\mu_{\delta_l^j} = \frac{1}{n_i}\sum_{i \in \boldsymbol{P}_N^{\text{train}}} \delta_{i,l}^j \quad \forall l, \forall j \tag{8}$$

$$\sigma_{\delta_l^j} = \sqrt{\frac{1}{n_i}\sum_{i \in \boldsymbol{P}_N^{\text{train}}} (\delta_{i,l}^j - \mu_{\delta_l^j})^2} \quad \forall l, \forall j \tag{9}$$

where n_i is the total number of time samples in the training dataset $\boldsymbol{P}_N^{\text{train}}$, i.e., the number of columns of $\boldsymbol{P}_N^{\text{train}}$.

Given a specific load vector $\boldsymbol{p}_i \in \boldsymbol{P}_N^{\text{test}}$ and its corresponding $\delta_{i,l}^j$ for all l and j, we determine how far each load deviates from the normative behavior using a Z-score, which is defined as:

$$z_{i,l}^j = \frac{\delta_{i,l}^j - \mu_{\delta_l^j}}{\sigma_{\delta_l^j}} \tag{10}$$

Intuitively, the Z-score indicates that the number of standard deviations by which $\delta_{i,l}^j$ is above (or below) the mean for load l in group j observed in the attack-free data.

On the basis of this setup, there exists a joint distribution $Q_{a,v}(z)$ for whether a load is attacked ($a = 1$) or not ($a = 0$), if it belongs to a group that raises a violation ($v = 1$) or not ($v = 0$), and its Z-score z. While $Q_{a,v}(z)$ is not known, we can empirically estimate the conditional probability $Q_{a|v}(z)$ of a load being attacked given its Z-score and whether it raises a violation or not. In other words, our goal is to define a likelihood function $\mathcal{L}_{a|v}(z)$ that takes the Z-score of a load and whether it raises a violation to determine the probability that the load is attacked as inputs.

First, we compute the Z-score (10) for all intelligently designed attacks in $\boldsymbol{P}_A$ that result in an overload rate of 3% or more. As discussed in Section V-B, those are the attacks that can cause significant damage and they are almost always detected by the nearest-neighbor algorithm. The distribution of the Z-score for the loads belonging to groups that raise a violation is shown in Fig. 8. From the curves of $\phi_{a=1,v=1}(z)$ and $\phi_{a=0,v=1}(z)$, which represent the distribution functions of Z-score for the loads that are attacked and are not attacked, respectively, we notice that if a load belongs to a group that raises a violation, it is very likely that the load is indeed being attacked. Moreover, the higher the Z-score is, the more likely the load is attacked. On the basis of these observations, we can define a function that maps the Z-score of a

load to the likelihood of the load being attacked. The estimated conditional likelihood for the loads belonging to groups that raise a violation is computed as:

$$\bar{\mathcal{L}}_{a|v=1}(z)=\frac{\phi_{a=1,v=1}(z)}{\phi_{a=1,v=1}(z)+\phi_{a=0,v=1}(z)} \tag{11}$$

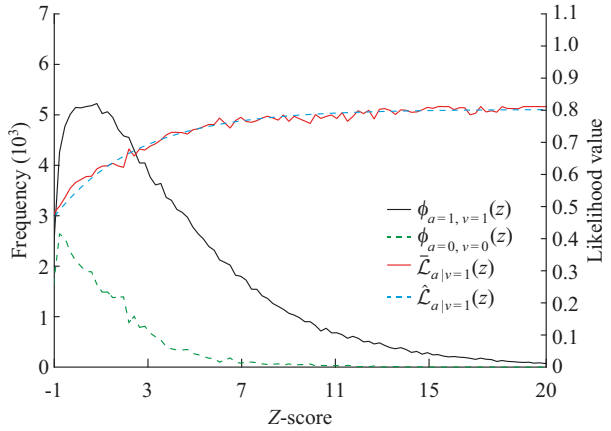

Fig. 8. Distribution of *Z*-score and likelihood function for loads in groups that raise a violation.

For the set of data points obtained by (11), we fit a smooth curve with the form of $Ae^{-Bx}+C$ to avoid overfitting, where A, B, C are constants. The corresponding likelihood function is defined as $\hat{\mathcal{L}}_{a|v=1}(z)$, which can be used to assign a probability of being attacked based on its *Z*-score to each load. The same procedure is performed on the loads in groups that do not raise a violation and the corresponding results of $\phi_{a=1,v=0}(z)$, $\phi_{a=0,v=0}(z)$, $\bar{\mathcal{L}}_{a|v=0}(z)$ and $\hat{\mathcal{L}}_{a|v=0}(z)$ are obtained, as shown in Fig. 9. Comparing the curves of $\hat{\mathcal{L}}_{a|v=1}(z)$ and $\hat{\mathcal{L}}_{a|v=0}(z)$, we notice that, for low *Z*-score values, $\hat{\mathcal{L}}_{a|v=1}(z)$ reaches a minimum likelihood value of around 0.5 while $\hat{\mathcal{L}}_{a|v=0}(z)$ reaches 0.

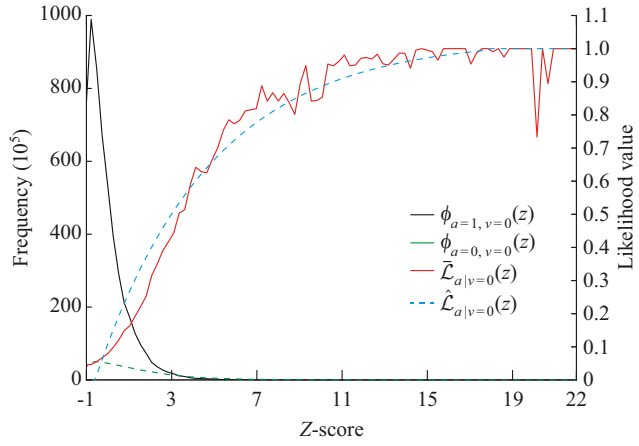

Fig. 9. Distribution of *Z*-score and likelihood function for loads in groups that do not raise a violation.

B. Numerical Results

The performance of the statistical approach proposed in Section VI-A is tested on the intelligently designed attacks from Section V-B, with $\tau_j=d_{\max,j}$. The conditional likelihood functions $\hat{\mathcal{L}}_{a|v=1}(z)$ and $\hat{\mathcal{L}}_{a|v=0}(z)$ are learned on 70% of the attacks and they are tested on the remaining 30%. The *Z*-score for every load is computed using (10) and the average log-loss is computed using (6).

For comparison, we also test two simpler approaches to assign likelihood values to each load. The first one does not rely on the *Z*-score and only considers if the load belongs to groups with violations or not. On the basis of our data, on average, in a group that raises a violation, 82% of the loads are attacked, while in the groups that does not raise violations, only 10% are actually attacked. Considering this prior knowledge, the first simple approach assigns $q_l=0.82$ to load l if the load is in a group that raises a violation and $q_l=0.10$ otherwise. The second approach is even simpler and it assigns a fixed q_l to every load regardless of which group it belong to. From our results, the optimal value of q_l for this approach is $q_l=0.15$. The average log-loss results of the proposed statistical approach (indicated as $q_{a|v}(z)$) and the two simpler ones (indicated as $q_{a|v}$ and q_a) are 0.340, 0.489, and 0.608, respectively. It can be observed that the more sophisticated the approach is, i.e., the more information is used, the smaller the average log-loss will be.

VII. Conclusion

In this paper, we propose an improved data-driven algorithm for the detection of LR attacks and a statistical approach for the localization of the attacked buses. The detector based on the nearest-neighbor algorithm and a grouping strategy is tested on a large number of attacks belonging to two different classes: intelligently designed attacks and random LR attacks. The results obtained on the synthetic Texas system show the excellent detection capability of the proposed detector, especially against the attacks that have the worst consequences on the power system. The statistical approach for attack localization assigns a likelihood value to each load indicating the probability of the load being attacked. This approach offers operators a greater insight in case of cyber-attacks allowing for more secure system operation.

As part of our future work, we intend to extend the proposed detector to the analysis of different anomalies. The model can be trained to not only detect an anomaly, but also determine the type of event, e.g., cyber-attack, natural event, and fault, that causes it. Moreover, the proposed detector can be enhanced by considering additional information about rare and sporadic events such as forecasts of extreme weather events or temporary changes in load patterns due to known causes, e.g., sporting events, holidays, etc. This could result in both improved detection probability and lower false alarm rate.

Appendix A

The implementation of the proposed detector within an EMS is depicted in Fig. A1.

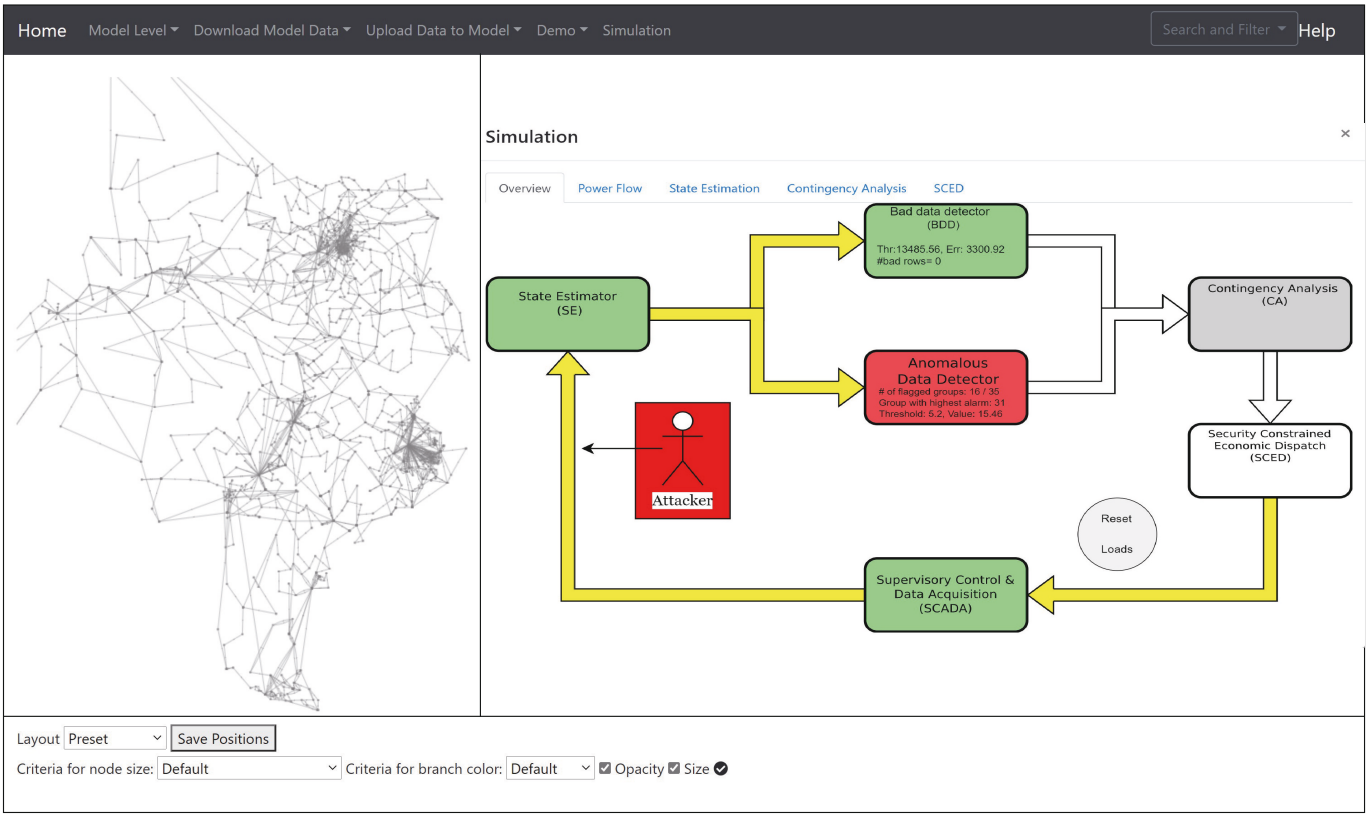

Fig. A1. Implementation of proposed detector within an EMS.

REFERENCES

[1] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," *ACM Transactions on Information and System Security*, vol. 14, no. 1, pp. 21-32, Jan. 2009.

[2] Y. Yuan, Z. Li, and K. Ren, "Modeling load redistribution attacks in power systems," *IEEE Transactions on Smart Grid*, vol. 2, no. 2, pp. 382-390, Jun. 2011.

[3] J. Zhang and L. Sankar, "Physical system consequences of unobservable state-and-topology cyber-physical attacks," *IEEE Transactions on Smart Grid*, vol. 7, no. 4, pp. 2016-2025, Jul. 2016.

[4] A. Sanjab and W. Saad, "Data injection attacks on smart grids with multiple adversaries: a game-theoretic perspective," *IEEE Transactions on Smart Grid*, vol. 7, no. 4, pp. 2038-2049, Jul. 2016.

[5] L. Xie, Y. Mo, and B. Sinopoli, "Integrity data attacks in power market operations," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 659-666, Dec. 2011.

[6] A. Pinceti, L. Sankar, and O. Kosut, "Load redistribution attack detection using machine learning: a data-driven approach," in *Proceedings of IEEE PES General Meeting*, Portland, USA, Aug. 2018, pp. 1-10.

[7] V. Joshi, J. Solanki, and S. K. Solanki, "Statistical methods for detection and mitigation of the effect of different types of cyber-attacks and parameter inconsistencies in a real world distribution system," in *Proceedings of 2017 North American Power Symposium*, Morgantown, USA, Sept. 2017, pp. 1-6.

[8] J. Yu, Y. Hou, and V. Li, "Online false data injection attack detection with wavelet transform and deep neural networks," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 7, pp. 3271-3280, Jul. 2018.

[9] Y. Huang, J. Tang, Y. Cheng *et al.*, "Real-time detection of false data injection in smart grid networks: an adaptive CUSUM method and analysis," *IEEE Systems Journal*, vol. 10, no. 2, pp. 532-543, Jun. 2016.

[10] Z. Chu, J. Zhang, O. Kosut *et al.*, "Evaluating power system vulnerability to false data injection attacks via scalable optimization," in *Proceedings of IEEE International Conference on Smart Grid Communications*, Sydney, Australia, Nov. 2016, pp. 1-10.

[11] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21-27, Jan. 1967.

[12] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: a survey," *ACM Computing Surveys*, vol. 41, no. 3, Jul. 2009, pp. 1-72.

[13] PJM. (2021, Mar.). PJM metered load data. [Online]. Available: https://dataminer2.pjm.com/list

[14] A. B. Birchfield, T. Xu, K. M. Gegner *et al.*, "Grid structural characteristics as validation criteria for synthetic networks," *IEEE Transactions on Power Systems*, vol. 32, no. 4, pp. 3258-3265, Jul. 2017.

[15] H. Li, A. L. Bornsheuer, T. Xu *et al.*, "Load modeling in synthetic electric grids," in *Proceedings of 2018 IEEE Texas Power and Energy Conference*, College Station, USA, Feb. 2018, pp. 1-8.

[16] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "MATPOWER: steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 12-19, Feb. 2011.

[17] A. Pinceti, L. Sankar, and O. Kosut, "Data-driven generation of synthetic load datasets preserving spatio-temporal features," in *Proceedings of IEEE PES General Meeting*, Atlanta, USA, Aug. 2019, pp. 1-10.

[18] L. Sankar. (2020, Mar.). A verifiable framework for cyber-physical attacks and countermeasures in a resilient electric power grid. [Online]. Available: https://sankar.engineering.asu.edu/a-verifiable-framework-for-cyber-physical-attacks-and-countermeasures-in-a-resilient-electric-power-grid/

[19] R. Podmore, "Digital computer analysis of power system networks," Ph. D. dissertation, University of Canterbury, Christchurch, New zealand, 1972.

[20] IncSys, Inc.. (2020, Apr.). IncSys–power system simulation software. [Online]. Available: https://www.incsys.com/

[21] R. Khodadadeh, "Designing a software platform for evaluating cyber-attacks on the electric power grid," M.S. thesis, Arizona State University, Phoenix, USA, 2019.

[22] A. Pinceti. (2019, May). Nearest neighbor attack detection. [Online]. Available: https://github.com/apince/EMS_FDI_NearestNeighborAttackDetection

[23] A. Abusorrah, A. Alabdulwahab, Z. Li *et al.*, "Minimax-regret robust defensive strategy against false data injection attacks," *IEEE Transactions on Smart Grid*, vol. 10, no. 2, pp. 2068-2079, Mar. 2019.

[24] H. Shayan and T. Amraee, "Network constrained unit commitment under cyber attacks driven overloads," *IEEE Transactions on Smart Grid*, vol. 10, no. 6, pp. 6449-6460, Nov. 2019.

[25] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Hoboken: Wiley-Interscience, 1991.

Andrea Pinceti received the B.E. degree in electrical engineering from the Polytechnic University of Turin, Turin, Italy, in 2015. He received the master's degree in 2019 from the School of Electrical, Computer, and Energy Engineering, Arizona State University, Tempe, USA, where he is currently pursuing the Ph.D. degree. Currently, his research interests include cyber-security and data analytics related to power system.

Lalitha Sankar received the bachelor's degree from the Indian Institute of Technology, Bombay, India, the master's degree from the University of Maryland, Maryland, USA, and the Ph.D. degree from Rutgers University, Rutgers, USA. She was an Assistant Professor at ASU from 2012 to 2016. Prior to that she was a Research Scholar in the Department of Electrical Engineering, Princeton University, Princeton, USA, working with H. Vincent Poor. She was also a Science and Technology Teaching and Research Fellow supported by the Council on Science and Technology at Princeton University. She is a recipient of the 2014 NSF CAREER award. For her doctoral work, she received the 2007-2008 Electrical Engineering Academic Achievement Award from Rutgers University. Currently, she is an Associate Professor in the School of Electrical, Computer, and Energy Engineering, Arizona State University (ASU), Tempe, USA. Her research interests include information and data sciences and its applications to power systems.

Oliver Kosut received the B.S. degree in electrical engineering and mathematics from the Massachusetts Institute of Technology (MIT), Cambridge, USA, in 2004, and the Ph.D. degree in electrical and computer engineering from Cornell University, Ithaca, USA, in 2010. Since 2012, he has been a Faculty Member in the School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, USA, where he is an Associate Professor. Previously, he was a Postdoctoral Research Associate in the Laboratory for Information and Decision Systems, MIT, from 2010 to 2012. He received the NSF CAREER award in 2015. His research interests include information theory, cyber-security, and power systems.