Detection of False Data Injection Attacks Using the Autoencoder Approach

Wang, Chenguang; Tindemans, Simon; Pan, Kaikai; Palensky, Peter

DOI
10.1109/PMAPS47429.2020.9183526

Publication date
2020

Document Version
Final published version

Published in
2020 International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)

Detection of False Data Injection Attacks Using the Autoencoder Approach

Chenguang Wang, Simon Tindemans, Kaikai Pan, Peter Palensky
Department of Electrical Sustainable Engineering
Delft University of Technology
Delft, The Netherlands
{c.wang-8, s.h.tindemans, k.pan, p.palensky}@tudelft.nl

***Abstract*—State estimation is of considerable significance for the power system operation and control. However, well-designed false data injection attacks can utilize blind spots in conventional residual-based bad data detection methods to manipulate measurements in a coordinated manner and thus affect the secure operation and economic dispatch of grids. In this paper, we propose a detection approach based on an autoencoder neural network. By training the network on the dependencies intrinsic in 'normal' operation data, it effectively overcomes the challenge of unbalanced training data that is inherent in power system attack detection. To evaluate the detection performance of the proposed mechanism, we conduct a series of experiments on the IEEE 118-bus power system. The experiments demonstrate that the proposed autoencoder detector displays robust detection performance under a variety of attack scenarios.**

***Index Terms*—Anomaly detection, autoencoder, false data injection attack, unbalanced training data, machine learning.**

I. Introduction

The power system is increasingly equipped with sensors and communication infrastructures. This enables smarter grid operations, but also makes possible novel cyber attack scenarios that manipulate power system measurements instead of directly disrupting ICT infrastructure or stealing valuable data. Although the typical bad data detection (BDD) within state estimation (SE) can detect erroneous measurements and some "basic" attacks, well-designed attacks can remain stealthy and bypass the BDD, such as the stealthy false data injection attacks (FDIAs) [1]. These stealthy measurements manipulation attacks severely threaten both the economic dispatching and security control of the power system [2], [3].

Several techniques have been proposed to deal with stealthy FDIAs. In [4], the authors have proposed a Kalman filter estimator together with a chi-square detector. Other statistical methods, such as sequential detection using Cumulative Sum (CUSUM)-type algorithms were designed in [5]. The recent work [6] has proposed a detector utilizing the statistical consistency of measurements, presuming that the system is observable by a minimal set of secure phasor measurement units. These methods, however, can be limited by the prior assumption that measurements fit specific distributions, or by restrictions on the number of manipulated measurements [7].

Moreover, it is increasingly recognised that the distribution of normal power system states is not easily characterised using standard parametric distributions [8]. The need to operate in a complex stochastic environment has led to the deployment of data-driven methods. For example, distance-based algorithms like k nearest neighbour (k-NN) were used to cluster normal and corrupted measurement states [9]. Nevertheless, the very high dimensionality of measurements (from the physical, cyber and market domains) results in data sparsity, where manipulated measurements may be masked by the noise of multiple irrelevant dimensions. This can make detection using a high-dimensional distance-based algorithm computationally inefficient or even invalid [10].

Alternative data-driven approaches to FDIA detection have been proposed in the form of support-vector machine (SVM)-based classifiers [11] and deep neural network-based classifiers [12]. Both are supervised machine learning algorithms that classify measurements into normal and manipulated data on the basis of labeled training data. However, due to the infrequent occurrence (or more likely: absence) of FDIAs in historical data, the training data set is highly unbalanced, so that it must be augmented by simulated training data. Moreover, in this way, the detector only learns to detect known attacks, which is a significant weakness in a fast-evolving field with resourceful and potentially well-equipped attackers.

This paper bridges the identified gap by proposing a detection approach based on an autoencoder neural network. The main contributions of this paper are listed below:

1) We propose an autoencoder-based detection approach for FDIAs. It learns to identify anomalous system states (and therefore possible attacks) using only SCADA-type power flow measurements for a large range of normal operating conditions. Therefore it is well-suited to the inherent data imbalance in FDIA detection applications.
2) We define a case study on the IEEE 118-bus system, including a model to generate 'normal' data. We formulate two FDIA scenarios by considering comprehensive factors of the adversaries' purpose, capacity, and knowledge and utilize indicators to evaluate the FDIA

This work is supported by the Chinese Scholarship Council.

978-1-7281-2822-1/20/$31.00 ©2020 IEEE

detection performance of our proposed mechanism. The experimental results demonstrate the mechanism has a satisfactory detection accuracy.

II. State Estimation and Data Attacks

In this section, we briefly review the state estimation and bad data detection technique and formulate the FDIA problem.

A. State estimation

The power system we consider has n_b buses and n_t transmission lines. The vector $\theta = [\theta_1, \theta_2, \ldots, \theta_{n_b}]^T$ represents n_b phase angles, excluding the angle of the reference bus. In this paper, a DC power flow model is assumed, in which the reactive power is neglected and bus voltages are assumed to be 1 (p.u.). The vector $P^I \in \mathbb{R}^{n_b}$ of active power injections is related to the angle vector θ,

$$P^I = AP^F = AR^{-1}A^T\theta, \tag{1}$$

where $P^F \in \mathbb{R}^{n_t}$ is the branch active power flow vector, $R \in \mathbb{R}^{n_t \times n_t}$ is a diagonal matrix of transmission line reactances and $A \in \mathbb{R}^{n_b \times n_t}$ is the branch-to-node incidence matrix [13]. In the following, we shall use the power injection vector P^I as the system state $x \in \mathbb{R}^{n_b}$. It is functionally equivalent to the more commonly used phase angle vector θ, but it allows for more elegant generation and detection of FDIAs.

We consider a system where the active power injections and line flows are measured with some error. Thus the system model $H \in \mathbb{R}^{(n_b+n_t)\times n_b}$ for measurement and state can be written by

$$z = \begin{bmatrix} I \\ H^F \end{bmatrix} x + e = Hx + e, \tag{2}$$

where the measurement noise vector $e \sim \mathcal{N}(0, D)$ denotes m independent zero-mean Gaussian variables with the covariance matrix $D = \text{diag}(\delta_1^2, \ldots, \delta_m^2)$ and the measurement vector $z \in \mathbb{R}^m$ indicates measured power injection and line power flow with noise. Identity matrix $I \in \mathbb{R}^{n_b \times n_b}$ and distribution factor matrix $H^F \in \mathbb{R}^{n_t \times n_b}$ are parts in H corresponding to the power injection and line power flow, respectively. According to (1), the distribution factor matrix can be described as $H^F = R^{-1}A^T(AR^{-1}A^T)^{-1}$. Given the observation of the measurements z, the state estimate $\hat{x}$ is solved by the weighted least squares (WLS) approach [14] as

$$\hat{x} = (H^T D^{-1} H)^{-1} H^T D^{-1} z := Kz. \tag{3}$$

B. Bad data detection and stealth FDIAs

The vector $\hat{x}$ is then utilized to estimate the power injection and line power flow measurements by $\hat{z} = H\hat{x}$. In bad data detection, a residual is defined to describe the difference between the actual and the estimated measurements, namely $r_o = z - \hat{z}$. This naturally gives rise to a BDD scheme that identifies bad data by comparing the 2-norm of r_o with a certain threshold τ, i.e. an alarm is triggered if $\|r_o\|_2 > \tau$.

We denote $a \in \mathbb{R}^m$ as the non-zero false data vector injected into measurement vector z. The manipulated measurement vector can be described as $z_a = z + a$. Here the vector c is defined as the deviation of the estimated state before and after the attack. The corrupted system state can be denoted as $\hat{x}_a = \hat{x} + c$. According to (3), the falsified state estimate $\hat{x}_a$ can be written by

$$\begin{aligned} \hat{x}_a &= (H^T D^{-1} H)^{-1} H^T D^{-1} z_a \\ &= (H^T D^{-1} H)^{-1} H^T D^{-1} (z + a) \\ &= \hat{x} + c, \end{aligned} \tag{4}$$

and the corresponding r_a after the attack can be expressed as

$$\begin{aligned} r_a &= z_a - H\hat{x}_a = z + a - H(\hat{x} + c) \\ &= r_o + (a - Hc). \end{aligned} \tag{5}$$

If $a = Hc$, then the manipulated residual r_a equals the original residual r_o. Thus the attacker manipulates the measurements with the residual unchanged and keeps stealthy with respect to this physics-based BDD scheme. This remains true if $a \neq Hc$, as long as $\|r_a\|_2 \leq \tau$ is still satisfied.

For our FDIA detection study, we consider one attack scenario from the perspective of an adversary that manipulates load patterns [3], for example in order to hide excessive power consumption or to reduce apparent power consumption for economic motives. The adversary needs to corrupt the power generation and power flow accordingly to avoid detection by BDD. The attack scenario will be detailed in section IV.

III. FDIA Detection Mechanism

In this section, we propose an FDIA detection mechanism based on the autoencoder algorithm. We first analyze the specific characteristics and advantages of the method for identifying FDIAs in the context of the power system. Then, we explain the attack detection principle of the autoencoder-based mechanism in detail. Finally, we describe the complete training and detection process of our proposed mechanism.

A. Autoencoder-based attack detector

FDIA detection is essentially a classification problem with the objective of distinguishing false data from data that is considered 'normal'. What the SVM-based [11] and deep neural network-based classifiers [12] have in common is to treat FDIA detection as a supervised learning task. However, supervised learning requires a training data set with representative examples of normal system operation and attacks. Such data sets are in short supply, because of the rarity of attacks, unwillingness to share data, and evolving attacks. As a result, it is difficult to learn a satisfactory discriminator of 'normal' and 'attack' scenarios on this basis [15].

Instead, we propose to approach FDIA detection as a one-class classification problem, where the detector is trained on examples of only 'normal' operation data. Observations with features that deviate substantially from those in the training data will be considered anomalies, in this case as 'potential attacks'. There are two main advantages to this approach. First, the autoencoder-based mechanism avoids the need to gather or generate attack data to create balanced data sets for training the classifiers. Second, by focusing on what is normal only,

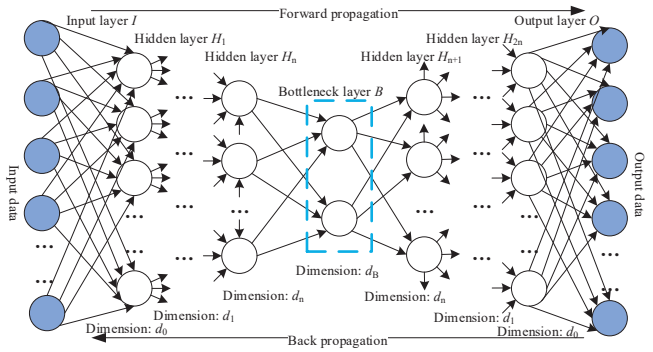

Fig. 1. The schematic of the Autoencoder.

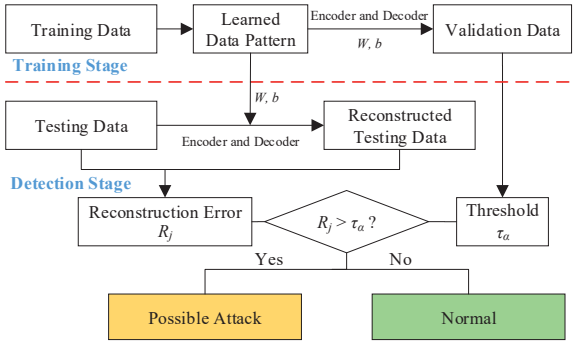

Fig. 2. The proposed training and FDIA detection mechanism.

the proposed mechanism is naturally prepared for unknown attack patterns.

Autoencoders learn the most important features of the training data (i.e. normal power system measurements) by sending the measurements through an information bottleneck while attempting to reconstruct the training data with minimal error [16]. The structure of the autoencoder algorithm is depicted in Fig. 1. The dimension reduction process of mapping the d_0-dimensional input data to the code in the bottleneck layer B through hidden layers H_1 to H_n is named the *encoder*. Afterwards, the *decoder* decompresses the code to d_0-dimensional output data. Weight matrices W and bias vectors b are utilized in the encoding and decoding process as

$$Y = \sigma(W_n^e(\dots\sigma(W_0^e Z + b_0^e)\dots) + b_n^e)\,, \tag{6a}$$

$$\hat{Z} = \sigma(W_n^d(\dots\sigma(W_0^d Y + b_0^d)\dots) + b_n^d)\,, \tag{6b}$$

where W_n^e and W_n^d denote weight matrices for encoding and decoding process respectively, b_n^e and b_n^d are bias vectors, and σ represents a nonlinear element-wise activation function. Z refers to the input data vector, Y is the data in the bottleneck layer and vector $\hat{Z}$ stands for the output data.

B. Training and detection process

The residual associated with a training observation Z_j is given by $r_j = Z_j - \hat{Z}_j$. The reconstruction error R_j is expressed as the ratio of the length of r_j to the input data dimension d_0 and the objective of the training process is to minimize the mean value of the sum of all reconstruction errors R_j as

$$\min_{W,\,b} \quad \left\{ J := \tfrac{1}{S}\sum_{1}^{S} (\|r_j\|^2/d_0) \right\}, \tag{7}$$

where S denotes the total number of the observations used for training. By training the autoencoder on training data that is considered normal, it learns to efficiently encode the features of this data in the bottleneck layer B. Data that deviates from the training data in a structural way is therefore highly likely to have a larger reconstruction error.

The training and FDIA detection process of the proposed mechanism is depicted in Fig. 2. In the training stage, the algorithm iteratively updates the value of weight matrices W and bias vectors b until the function J converges. At the end of the training process, the reconstruction errors R_j for the *validation* set are sorted in ascending order. A threshold τ_α equals to the α^{th} percentile is then chosen, for example at the value where an 'inflection point' occurs in the error distribution. A possible FDIA is detected when, for a measurement Z_j in the *test* set, the reconstruction error R_j exceeds the threshold τ_α.

IV. Case Study

In this section, we evaluate the detection performance of the proposed mechanism using a case study on the IEEE 118-bus system. First, we describe the process of modelling normal operating conditions and explain how to create anomalous attack scenarios. Then, we describe and analyse the load-targeted attack scenario. For this scenarios, we will first quantify the detection performance of our proposed detection mechanism. Specifically, the detection probability, false positive rate, false negative rate are tested. Next, the detection performance of our detector will be compared with a conventional BDD detector. To do so, we introduce "knowledge limited" attacks that both detectors can potentially detect. Notably, the "knowledge-limited" attacks are more of interest in reality as the attacker may have an inaccurate (e.g. out-dated or estimated) system model.

A. Modeling normal operating conditions

With the long-term secure and stable operation, the power system has a large number of normal operating conditions which involve a significant volume of loads, power generations and power flows data set. Trained by these data, the proposed mechanism will acquire the data pattern which represents the model of normal system operating conditions.

In the IEEE 118-bus system, electricity is supplied by $M = 54$ generators, transmitted via $Q = 186$ branches and ultimately consumed by $N = 99$ loads. We generate 'normal' (i.e. physically feasible and economically reasonable) power system states and corresponding measurements by using optimal power flow solutions.

Second order polynomial cost functions were assumed for generators, i.e., $f(P_g^G) = C_{g,2}(P_g^G)^2 + C_{g,1}P_g^G$. Hence

the economic dispatch P^{G^*} is solved with the objective to minimize the total generation cost. The solutions are implicitly parameterized by the nodal load P_l^L and generation cost parameter as

$$P^{G^*} = \arg\min_{P^G} \quad \sum_{g=1}^{M} C_{g,2}(P_g^G)^2 + C_{g,1}P_g^G \tag{8}$$
$$\text{s.t.} \quad \sum_{g=1}^{M} P_g^G - \sum_{l=1}^{N} P_l^L = 0,$$

where the injection $P^I = P^I(P^G, P^L)$ is determined by the mapping of load P^L and generation P^G onto the nodes.

Normal operating conditions are generated using a data set that contains a total of 43,717 historical hourly loads from 32 European countries between 2013 and 2017 [17]. These time series were used to generate a 99 load point time series as follows. The national load time series are first divided by 1000, to obtain reasonable magnitudes for individual buses. Then each load point is assigned a random linear combination of the 32 sources by sampling from the Dirichlet distribution with vector valued parameter $(1, \ldots, 1)^T$, which generates a uniform distribution on the 31-simplex. Additionally, a normally distributed variation with a standard deviation of $\pm 5\%$ of the measured value is added to each measurement.

An additional source of randomness was created by randomly sampling the generating cost coefficients of the 54 generators as follows. Coefficients $C_{g,2}$ were sampled uniformly in the range $[0.085, 0.1225]$ \$/MWh^2 and $C_{g,1}$ uniformly in the range $[1, 5]$ \$/MWh. These values span the range of generators included in the IEEE 9-bus system supplied with Matpower [18].

The procedure above was used to generate snapshot injections $P^I = P^I(P^{G^*}, P^L)$, which were converted into line flow measurements using $P^F = H^F P^I$. In this investigation, line transmission limits and generator capacities are not enforced, as the focus of this work is on the recognition of load, generation and power flow patterns. This results in a 339-dimensional measurement vector for training, containing 99, 54 and 186-dimensional data of loads, power generations and line power flows, respectively. Independent measurement noise e is added using a truncated Gaussian distribution with zero mean, standard deviation of 0.33% and an absolute value less than 1% of the original value [19]. The generated data set $T \in \mathbb{R}^{43717\times 339}$ was divided into a training set $T_r \in \mathbb{R}^{26197\times 339}$, a validation set $T_v \in \mathbb{R}^{8760\times 339}$ and testing set $T_e \in \mathbb{R}^{8760\times 339}$.

In this paper, the autoencoder network contains 4 hidden layers in the encoder with dimensions of 339, 256, 128 and 64, respectively. The bottleneck layer has 32 nodes, and the decoder maps the 32-dimensional data to a 339-dimensional output through 3 hidden layers with the same dimensions as the encoder. In this paper, we used the sigmoid activation function between the second and penultimate hidden layer and the Adam Optimizer [20] to iteratively optimize the value of weight matrices W and bias vectors b. The batch size and learning rate for training was 256 and 10^{-5} respectively and 2000 training epochs were used. Training and testing of the autoencoder was conducted using `tensorflow` on the Google Colab environment using the GPU option. An initial performance analysis of hyperparameter settings for the autoencoder-based FDIA detector is available in [21].

B. Creating attack scenarios

We develop feasible FDIAs from the perspective of the adversaries by adding an offset to the normal operating conditions created in the previous section. To gain economic profit, attackers inject false data into the grid by using the acquired knowledge of the targeted power system. In the context of this paper, this knowledge is represented by the incidence matrix A (topology) and the reactance matrix R of the transmission lines. Moreover, we assume that the capacity of an attacker is limited by the attackable measurement set [1] and the maximum number of the measurements that the attacker can corrupt simultaneously.

In the following, we quantify the factors described above. According to the attack capacity, the adversary selects a set of attacked loads $\mathcal{L}^A \subseteq \mathcal{L}$. The attacker then determines the change rate β_l of each selected load and calculates the total load change $\sum_{l\in\mathcal{L}^A} \beta_l P_l^L$, in which $\beta_l P_l^L$ equals the change ΔP_l^L of each load. Similarly, the attack selects a set of attacked generators $\mathcal{G}^A \subseteq \mathcal{G}$. Next, the attack determines ratios of the power generating's change amount $\lambda_1 : \lambda_2 : \ldots : \lambda_{|\mathcal{G}^A|}$ and normalizes the ratios to get the power generations' change ΔP_g^G. Here $|\mathcal{G}^A|$ represents the cardinality of $\mathcal{G}^A$.

$$\Delta P_g^G = \left[\sum_{l\in\mathcal{L}^A} \beta_l P_l^L\right] \times \frac{\lambda_g}{\sum_{g'\in\mathcal{G}^A} \lambda_{g'}} \tag{9a}$$

All load changes ΔP_l^L and generation changes ΔP_g^G, together with zeros that denote buses with unchanged injection make up the power injection change vector $\Delta P_A^I \in \mathbb{R}^{118}$. Besides, similar to (2), the attacker then utilizes the knowledge of the topology and grid parameters to coordinately calculate power flows change vector $\Delta P_A^F \in \mathbb{R}^{186}$.

$$\Delta P_A^F = H^F \cdot \Delta P_A^I, \tag{9b}$$

Afterwards, the attack vector a consists of the change vector of loads, power generations and line power flows.

The FDIA manipulates the original data of loads, power generations and line power flows. The pattern of the corrupted data may deviate from that of normal operating conditions, which enables it to be detected by the autoencoder if the reconstruction error R_j exceeds τ_α.

C. Load-targeted attack for economic profit

1) Detection effectiveness validation: We first validate the effectiveness of the trained detector. In this experiment, we observe the change of the reconstruction error R_j before and after a false data injection attack and compare it with the threshold τ_α. A common scenario for an attack happens when the adversary gets the data of a local area and utilizes it to

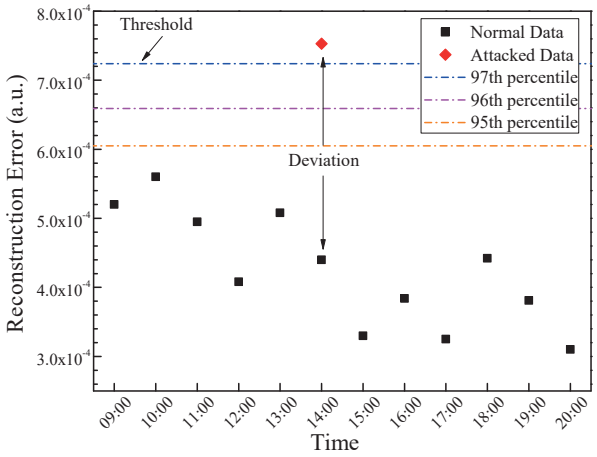

Fig. 3. Detection effectiveness validation by launching an FDIA.

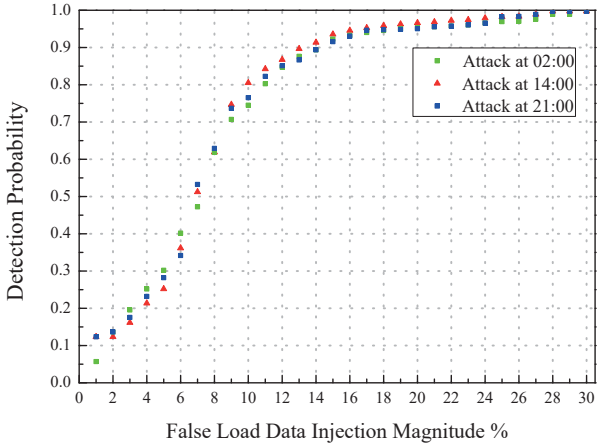

Fig. 4. Detection probability of attacks at different time with different false load data injection magnitude.

manipulate the neighboring measurements. Here, we select 12 hours' operating data from 9:00 to 20:00 on December 31st, 2017 as an example. Assuming the attacker gets the three loads' profile of bus 108, 109, 110, at 14:00, to gain economic profit, an attack is launched by injecting false data to decrease the power demand of the loads by 10% as $-7.48\,\mathrm{MW}$, $-5.69\,\mathrm{MW}$ and $-6.28\,\mathrm{MW}$ respectively. Accordingly, to balance the power of loads and generations, the attacker decreases the nearby power injection of two generators connected to bus number 110 and 111 with the ratio $\lambda_1 : \lambda_2 = 1$. Based on (9b), the corresponding transmission line power flows are obtained. The experiment result is depicted in Fig. 3.

From the result, we can observe that before the attack, the reconstruction error R_j of normal operating data is in the range of 3.10×10^{-4} and 5.60×10^{-4}, and they are lower than the threshold $\tau_{97\%} = 7.25 \times 10^{-4}$ learned in the training process shown in the subsection B of Section III. To be specific, after observing the reconstruction error distribution of the validation data, the threshold is set as 97^{th} percentile due to the occurrence of the 'inflection point' where the cumulative distribution curve of the reconstruction error flattens out from the steep rise. After manipulation by the false data injection, the reconstruction error R_j at 14:00 increases from 4.40×10^{-4} to 7.53×10^{-4}, which exceeds the threshold $\tau_{97\%}$ and triggers an alarm. The detector thus recognizes an anomaly in the corrupted measurements, which deviate from measurements taken in normal operating conditions. This result demonstrates that the autoencoder is capable of FDIA detection in at least some scenarios.

2) General detection performance: In addition to the one-off effectiveness demonstrated above, we are also interested in its statistical detection performance. This is tested by launching a larger number of FDIAs at various times and with various false load data injection magnitudes. Here the magnitude is defined as the percentage of load reduction in targeted nodes. For the sake of comparison, the attack targets remained the same as these utilized in the last experiment. In this experiment, we launch an attack at 2:00, 14:00 and 21:00 in each day of 2017 by reducing reported loads between 1% to 30% and observing the detection performance. The detection probability is the ratio of detected attacks to all the launched attacks, namely the true positive rate. The results are shown in Fig. 4.

Because the load demands at 2:00, 14:00 and 21:00 differ significantly, the resulting power system states (including flows) are also substantially different. However, the result shows, under the same false load injection magnitude, the detection probabilities differ only slightly. This demonstrates that the autoencoder learns the intrinsic relationship of the loads, power generations and power flows from different operating conditions, leading to robust detection results.

In addition, we launch 8760 attacks, one for each hour of 2017, by decreasing the power demand of the same buses by 15%. Besides, we use the hourly normal operating data in 2017 as a control group. The result is shown in Table I.

TABLE I
DETECTION PERFORMANCE EVALUATION.

	Normal Data		Attack Data
True Negative	96.5% (8453)	True Positive	93.6% (8199)
False Positive	3.5% (307)	False Negetive	6.4% (561)

From the experiment result, we can find that the detection probability (true positive rate) is 93.6%, which denotes a satisfactory detection performance. As mentioned in the first experiment, the threshold $\tau_{97\%}$ was used, corresponding to a 3% misclassification rate in the validation set. It is worth noting that the false positive rate is comparable to the 3.5% observed in Table I. This result suggests that the autoencoder has a good generalization capability and does not overfit.

3) Detection performance comparison: In the above experiments, our proposed autoencoder-based detector has succeeded in generating a diagnosis signal in the presence of FDIAs which can keep stealthy from the viewpoint of BDD. In the second experiment, we compare our detector with BDD in detection of 'unstealthy' FDIAs. Such attacks have the

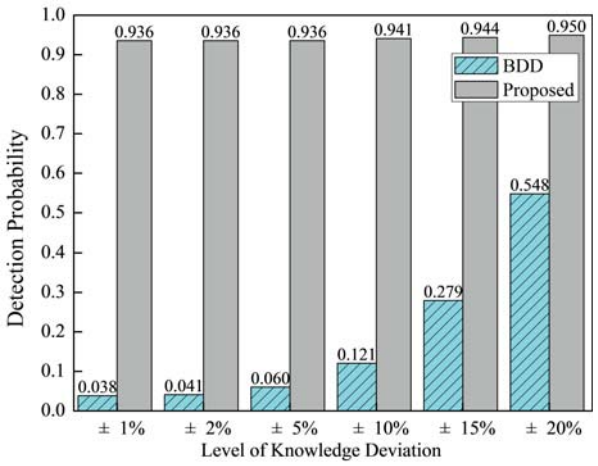

Fig. 5. Detection performance comparison between the proposed mechanism and the BDD scheme in load-targeted attack scenario.

possibility to be detected by the BDD while the detectability is intimately related to the topology or parameter errors in the construction of FDIAs by the attacker. Thus in what follows there exist knowledge deviations in the system model acquired by the attacker in computing the attack vector of (9). In particular, we explore the case that the attacker knows the exact topology of the network but inaccurate line reactance R in (1). This can be described by

$$\hat{R} = R \cdot (I^R + \gamma), \tag{10}$$

where $I^R \in \mathbb{R}^{n_t \times n_t}$ is the identity matrix and $\gamma \in \mathbb{R}^{n_t \times n_t}$ is a diagonal matrix whose elements denote the reactance deviation ratio - which we will refer to as the *knowledge deviation ratio*. In this experiment, we range the magnitude of the deviations from 0.01 to 0.20, with randomly sampled signs for each element. According to the explanation of (2), this will lead to an erroneous distribution factor matrix H^F and thus obtain inaccurate power flow values. We keep the attack target unchanged from the previous experiments and set the false load data injection magnitude on the selected three loads by decreasing them by 15%. The results are shown in Fig. 5. As the level of knowledge deviation increases from $\pm 1\%$ to $\pm 20\%$, the detection probability of BDD rises from 0.038 to 0.548, but it remains lower than the detection performance of the autoencoder.

V. Conclusion

In this paper, we propose an FDIA detection mechanism based on an autoencoder neural network. The main contribution is that, distinct from existing approaches, the approach learns the internal dependency of 'normal' operation data, which avoids the need for gathering or generating attack data for training the classifiers and thus effectively overcomes the inherent unbalanced training data set challenge in power system. The results demonstrate that the mechanism is able to robustly detect stealthy FDIAs. Moreover, it still outperforms a BDD scheme when the attacker has only approximate knowledge of the network parameters.

In future work, we aim to extend the method to analyze temporal signatures and to include contextual information.

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 (TISSEC)*, vol. 14, no. 1, p. 13, 2011.

[2] R. Liu, C. Vellaithurai, S. S. Biswas, T. T. Gamage, and A. K. Srivastava, "Analyzing the cyber-physical impact of cyber events on the power grid," *IEEE Transactions on Smart Grid*, vol. 6, no. 5, pp. 2444–2453, 2015.

[3] L. Jia, J. Kim, R. J. Thomas, and L. Tong, "Impact of data quality on real-time locational marginal price," *IEEE Transactions on Power Systems*, vol. 29, no. 2, pp. 627–636, 2013.

[4] K. Manandhar, X. Cao, F. Hu, and Y. Liu, "Detection of faults and attacks including false data injection attack in smart grid using kalman filter," *IEEE transactions on control of network systems*, vol. 1, no. 4, pp. 370–379, 2014.

[5] S. Li, Y. Yılmaz, and X. Wang, "Quickest detection of false data injection attack in wide-area smart grids," *IEEE Transactions on Smart Grid*, vol. 6, no. 6, pp. 2725–2735, 2015.

[6] J. Zhao, L. Mili, and M. Wang, "A generalized false data injection attacks against power system nonlinear state estimator and countermeasures," *IEEE Transactions on Power Systems*, p. 1, 2018.

[7] K. Pan, P. Palensky, and P. M. Esfahani, "From static to dynamic anomaly detection with application to power system cyber security," *IEEE Transactions on Power Systems*, pp. 1–1, 2019.

[8] M. Sun, I. Konstantelos, S. Tindemans, and G. Strbac, "Evaluating composite approaches to modelling high-dimensional stochastic variables in power systems," in *2016 Power Systems Computation Conference (PSCC)*. IEEE, 2016, pp. 1–8.

[9] J. Tian, M. H. Azarian, and M. Pecht, "Anomaly detection using self-organizing maps-based k-nearest neighbor algorithm," in *Proceedings of the European Conference of the Prognostics and Health Management Society*. Citeseer, 2014.

[10] C. C. Aggarwal, "Outlier analysis," in *Data mining*. Springer, 2015, pp. 237–263.

[11] Y. He, G. J. Mendis, and J. Wei, "Real-time detection of false data injection attacks in smart grid: A deep learning-based intelligent mechanism," *IEEE Transactions on Smart Grid*, vol. 8, no. 5, pp. 2505–2516, 2017.

[12] J. James, Y. Hou, and V. O. 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, 2018.

[13] F. M. Gonzalez-Longatt and J. L. Rueda, *PowerFactory applications for power system analysis*. Springer, 2014.

[14] H. Sandberg, A. Teixeira, and K. H. Johansson, "On security indices for state estimators in power networks," in *First Workshop on Secure Control Systems (SCS), Stockholm*, 2010.

[15] L. Duan, M. Xie, T. Bai, and J. Wang, "A new support vector data description method for machinery fault diagnosis with unbalanced datasets," *Expert Systems with Applications*, vol. 64, pp. 239–246, 2016.

[16] M. Sakurada and T. Yairi, "Anomaly detection using autoencoders with nonlinear dimensionality reduction," in *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*. ACM, 2014, p. 4.

[17] J. Muehlenpfordt, "Time series," *Open Power System Data*, 2019. [Online]. Available: https://data.open-power-system-data.org/time series/2019-06-05

[18] R. D. Zimmerman, C. E. Murillo-Sánchez, and D. Gan, "Matpower: A matlab power system simulation package," *Manual, Power Systems Engineering Research Center, Ithaca NY*, vol. 1, 1997.

[19] M. He, V. Vittal, and J. Zhang, "Online dynamic security assessment with missing pmu measurements: A data mining approach," *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 1969–1977, 2013.

[20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[21] C. Wang, K. Pan, S. Tindemans, and P. Palensky, "Training strategies for autoencoder-based detection of false data injection attacks," in *The 2020 IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe 2020)*, 2020, arXiv:2005.07158.