Detection of false data injection attacks against state estimation in smart grids based on a mixture Gaussian distribution learning method

S. Armina Foroutan1, Farzad R. Salmasi2

1School of Electrical and Computer Engineering, University of Tehran, North Kargar St., Tehran, Iran
2E-mail: farzad_rs@ieee.org

Abstract: One of the most addressed attacks in power networks is false data injection (FDI) which affects monitoring, fault detection, and state estimation integrity by tampering measurement data. To detect such devastating attack, the authors propose a statistical anomaly detection approach based on Gaussian mixture model, while some appropriate machine learning approaches are evaluated for detecting FDI. It should be noted that a finite mixture model is a convex combination of some probability density functions and combining the properties of several probability functions, making the mixture models capable of approximating any arbitrary distribution. Simulations results confirm superior performance of the proposed method over conventional bad data detection (BDD) tests and other learning approaches that studied in this article. It should be noted that using data which change significantly over a day can be highly clustered, and therefore, detected much easier compared with small changes in the loads. So without loss of generality, in the simulations it is assumed that the power demand follows a uniform distribution in a small range. However, the detector can be trained regularly based on the updated load profile.

1 Introduction

Smart grid is one of the most critical infrastructures and its normal operation is a key factor for security of energy generation, transmission, and distribution. In recent years, power networks have become more developed and sophisticated due to their correlation with communication devices and big data exchange. Although, this evolution provides more capabilities to these networks such as bilateral flow of electricity and information, it brings new vulnerabilities due to open standards used in communication networks [1–3]. As a result, power systems become more prone to cyber-physical attacks, a problem that had not been faced seriously before invention of such internet of energy. These attacks can target any part of the system and interfere with normal operation of the grid. Supervisory Control and Data Acquisition (SCADA) systems continuously control and supervise the power generation and power flow to ensure normal, secure, and uninterrupted operation of the grid. For this end, SCADA systems frequently collect measurements from sensors and smart meters all over the grid and feed them to Energy Management System (EMS). EMS consists of some modules such as optimal power flow, contingency analysis, and bad data detection, which subsequently provide commands to the operator. To facilitate this procedure, EMS relies on states estimation, so state estimators play an important role in secure operation of the system.

EMS/SCADA systems are mainly designed for isolated environments and do not have efficient security setups [4]. Bad data detection (BDD) systems are mostly utilised to detect random errors such as meter failures, defect in telemetry, or system noise. However, modern power networks and emerging smart grid make systems more accessible from local networks and smart meters. Existence of such entry points all over the grid makes sophisticated and potential destructive attacks feasible. However, before Liu’s paper [5], it was assumed that BDD systems and existing tests such as χ2 test could detect malicious measurement injected by attackers, since they could be considered as interacting bad measurements [6, 7]. In 2011, Liu introduced a new type of adversarial attacks, called false data injection (FDI), which could bypass current BDDs and affect estimated states [5]. This type of cyber-attack is devastating because biased estimations directly misguide the system and yield to wrong analysis and control commands.

Most of the reports in this particular area are focused on design of such attacks and there are only a few reports on FDI attack detection [8]. He and Yan reviewed the cyber-physical attacks and defence mechanisms in smart grids [9]. Cyber-attacks can be classified by the security object they have targeted, such as availability targeted attacks (ATA), integrity targeted attacks (ITA), and Confidentiality Targeted Attacks (CTA). ATAs try to delay, block, or corrupt the communications which are known as denial-service attacks. ITAs are Adversary attempts to insert some unauthorised information against operator’s will [7, 10–13], and in CTA attacker does not disrupt or block the communication, but has unauthorised access to the confidential information which usually affects electricity markets [3]. Moreover, opportunistic attackers may target the communication links between local control centers in a distrusted system like power networks [14]. Furthermore, recent advances and utilising renewable energy resources emerge new vulnerabilities to the network that occur at consumers and third-party level which impact the whole system. For this end, in [15] cybersecurity for distributed energy resources is studied. Assessing and mitigating cybersecurity risks of traffic light systems in smart cities are studied in [16].

False data may be due to unintended measurement abnormalities, topology errors, or injection by malicious attacks. An undetectable false data injection attack first introduced in [5]. Literature concerning this type of attack can be classified into two main categories, i.e. designing the attack and confronting the intrusion. There are some assumptions in [5] like attacker's full knowledge of the grid, sparsity of the attack, or attacker's resources, which limits the spread and depth of the intrusion. Some authors tried to design FDI attacks without these limitations on the attacker side [17–20]. On the other hand, confronting the intrusions are reported as protection-based, separation-based, and detection-based schemes [21]. Protection-based schemes aim to reduce the possibility of FDI attacks by protecting critical meters which are carefully selected [22–24]. Separation-based schemes try to separate attack matrix from the original measurements based on sparse nature of attack matrix and low rank of the measurements [11, 21], although it seems that these schemes only work for
attacks with a very small magnitude [25]. Moreover, detection-based schemes intend to detect and isolate the intrusions. Some authors formulated the state estimation problem under attack as a hypothesis test and suggested to use some variant of cumulative sum approach [9, 10], although such mechanisms require a prior probability distribution for the grid states. Authors in [13] utilise a graph theoretic approach and conditional covariance test to learn the network structure and trigger an alarm if two obtained graphs mismatch. Recently, few attempts have been done to use machine learning approaches to detect or predict attacks and intrusions [26–28]. In [29], authors use reinforcement learning to detect topological attacks. Authors in [30] investigated state-of-the-art machine learning algorithms including linear SVM, k-nearest neighbour, and perceptron in order to detect attacks with different degrees of sparsity, while energy of the attack vectors was not considered. In [31], supervised learning schemes were studied for false data attack detection considering attack vector magnitude.

In this paper, we investigate utilisation of a semi-supervised learning approach based on mixture Gaussian distribution for detecting FDI attacks affecting state estimation in smart grids. While prominent learning methods, classified to supervised and unsupervised, are studied to detect FDI attacks, and advantages of these methods over conventional BDD tests are shown, superiority of MGD-based approach is verified through extensive simulations. A finite mixture model is a convex combination of two or more probability density functions. By combining the properties of the individual probability functions, mixture models are capable of approximating any arbitrary distribution. Principle component analysis is utilised to overcome the curse of dimensionality in all investigated approaches. It is assumed that all attacks have the same amount of energy which result in the same amount of mean square error. We try to use minimum energy residual attacks and sparest attack which are explained thoroughly in [32, 33], respectively. In [30], linear classifiers, decision fusion, and feature fusion methods are used to model non-linearities. However, we use multi-layer perceptron and SVM with non-linear kernel along with semi-supervised algorithms based on statistical anomaly detection to investigate these learning methods. Our approach contains a three phase classification; first we use a positively labelled set to build Gaussian mixture model, then we use a mixture data set to choose proper threshold, and finally we use an unlabelled data set for evaluation. Owing to using historical data to learn threshold, in some literatures, this scheme is called semi-supervised learning. Nonetheless, if the mixture data set which used for choosing threshold as a labelled data is considered, it would be a one-class classification. Extensive numerical simulations are reported on IEEE standard 118 bus test system [34] to show the effectiveness of our proposed method, comparing with other learning-based tools. Moreover, it will be shown that the proposed MGD-based method outperforms SVM in terms of computational complexity.

This paper is organised as follows. Section 2 presents system and attack model, and BDD. Section 3 investigates the feasibility of learning techniques and introduces the proposed semi-supervised mixture Gaussian distribution approach. In Section 4 extensive and comparative numerical results are provided. Conclusions are given in Section 5.

2 System and attack models, and BDDs

2.1 System model

Considering a power network as shown in Fig. 1 with \( n + 1 \) buses and \( l \) branches, let \( Z \in \mathbb{R}^m \) denotes measurement vector and \( x \in \mathbb{R}^n \) is the state vector, \( m > n \). Hence, the measurements can be generally described as

\[
Z = h(x) + e
\]

(1)

where \( h(\cdot) \) stands for the non-linear relation between measurement \( Z \) and state \( x \), \( e \) is measurements noise with Gaussian distribution \( \mathcal{N} \sim (0_{n \times 1}, W^2) \), and \( W \in \mathbb{R}^{m \times m} \) is a diagonal matrix with diagonal elements proportional to variance of each measurement noise. Considering \( \delta \) as set of all network buses, measurement vector \( Z \) consists of active and reactive power flow of all branches, \( P, Q, i, j \in \delta \), respectively, and active and reactive power injection at each bus \( i, P_i, Q_i \). These measurements make the system observable, or in other words, make state estimation feasible. In non-linear model, bus voltage amplitude, \( V_n \), and phase angle, \( \theta_n \), are considered as state variables [35]. Aside from the reference bus, system has \( 2n + 1 \) state variables which make the state vector as follows:

\[
x = [\theta_1, \theta_2, \ldots, \theta_{n+1}, V_1, V_2, \ldots, V_{n+1}]^T
\]

(2)
Non-linear power flow equations which represent relations between state variables and measurements are as follows:

\[
P_{ij} = V_i^* r_{ij} - V_i V_j \cos(\theta_{ij}) - V_i V_j \sin(\theta_{ij})
\]

\[
Q_{ij} = -V_i^* r_{ij} + V_i V_j \sin(\theta_{ij}) - V_i V_j \cos(\theta_{ij})
\]

\[
P_i = \sum_{j \in S_i} P_{ij}, \quad Q_i = \sum_{j \in S_i} Q_{ij}
\]

(3)

where \( S_i \) is the set of all neighbour buses to the \( i \)th bus, \( r_{ij} \) and \( b_{ij} \) are conductance and susceptance of the line between bus \( i \) and \( j \), respectively, and \( \theta_{ij} = \theta_j - \theta_i \) denotes phase difference angle between bus \( i \) and \( j \).

Standard linear approximation makes the state estimation problem lot easier to solve. Linearising equations are made by some assumptions:

- All bus voltage amplitudes are fixed and equal to 1 p.u.
- All shunt susceptances and series resistances are dismissed.
- Phase difference angles are very small.

Due to these assumptions, reactive power flows become zero and consequently (3) can be rewritten as follows:

\[
P_i = \sum_{j \in S_i} P_{ij}, \quad Q_i = \sum_{j \in S_i} Q_{ij}
\]

(4)

With fixed voltage amplitude, state vector just consists of phase angle, and therefore, (1) becomes linear as follows:

\[Z = Hx\]

(5)

where \( H \in \mathbb{R}^{m \times n} \) is called Jacobian matrix and defines linear relation between measurements and states. Jacobian matrix, \( H \), depends on network topology, lines susceptance, and meter placement [36], so it can be constructed as follows:

\[H = \begin{bmatrix} T_A D A^T \\ T_D A_r \end{bmatrix}\]

(6)

where matrices \( T_A \in \mathbb{R}^{n \times n} \) and \( T_D \in \mathbb{R}^{n \times l} \) are stacked identity matrices that indicate which bus power injection and line power flow have been measured, respectively. Furthermore, \( D \) is a diagonal matrix with diagonal entries reciprocal to lines reactance, and \( A \in \mathbb{R}^{n \times l} \) is the adjacency matrix defined as:

\[A(i, j) = \begin{cases} 1 & \text{if arc } j \text{ starts at node } i \\ -1 & \text{if arc } i \text{ starts at node } j, \quad \forall \ i, j \in S \\ 0 & \text{otherwise} \end{cases}\]

(7)

State estimation problem is to find \( \hat{x} \) such that residual vector \( r(x) = Z - H \hat{x} \) is minimised considering the following weighted least square problem:

\[\min J(x) = \frac{1}{2} r(x)^T W r(x)\]

(8)

For linear model, the above problem has a closed-form solution as follows:

\[\hat{x} = (H^T H)^{-1} H^T W Z = KZ\]

(9)

### 2.2 False data injection attacks and bad data detection

Under the condition of perfect knowledge of the Jacobian matrix, an attacker can inject an attack vector, denoted by \( a \), resulting in the new measurements \( Z_e = Z + a \), where \( a \) and \( Z_e \) are \( m \times 1 \) vectors, to compromise the original measurements, \( Z = Hx + e \).

Considering \( e = [e_1, e_2, \ldots, e_m]^T \) as an arbitrary \( n \times 1 \) nonzero vector, according to [5], attack vector can be obtained by

\[a = Hc\]

(10)

With this scheme the estimated states change such that \( \hat{x} = (H^T W H)^{-1} H^T W Z_e = \hat{x} + c \) but the residual vector remains the same which leads the attack to be undetectable.

\[r_e = Z_e - H \hat{x} = Z + a - H(\hat{x} + c) = Z - H \hat{x} + (a - Hc) = Z - H \hat{x}\]

(11)

In this paper, we aim to make sure that the attack vector elements are the same in the sense of energy, so the comparison stage and simulations are valid and meaningful. As it is shown in [32], if \( e_0 = \text{Trace}(\Sigma_a - K \Sigma_e) \) presents minimum mean square error, then the MMSE in the presence of the attack would be \( e_0 + \| Ka \|_2^2 \), which means MMSE can be controlled by means of attack energy. Optimal attack in the sense of maximising the MMSE, while producing minimal residue to limit the probability of detection can be formulated as the following constraint optimisation problem

\[
\min \| Ga \|_2^2 \quad \text{subject to } \| Ka \|_2^2 \geq C
\]

(12)

where \( G \equiv I - HK \) and \( C \) is the minimum energy of the attack. It was shown in the same reference that with fixed sparsity pattern, the problem can be solved. This sparsity pattern can be obtained from procedure explained in [33], considering full measurement assumption. If the attacks are not the same in the sense of energy, the detection would be easier in some cases and harder in some others, so we use this optimisation problem to find out minimum residue energy attack.

Data which are corrupted by adversary attacks, topological errors, or faulty sensors, can be detect by calculating residual vector, \( r \) and using threshold test,

\[
\max_{i=1}^m \left| \frac{r_i}{\sigma_i} \right| \geq \gamma
\]

(13)

where \( | \cdot | \) stands for the magnitude, and hypotheses without and with FDI are shown by \( H_0 \) and \( H_a \), respectively. Moreover, \( \gamma \) stands for decision threshold, and \( r_i, i = 1, 2, \ldots, m \) denotes the elements of \( r \) and \( \sigma_i \) is the standard deviation of the \( i \)th residual error \( r_i \) [12]. In this case, it is assumed \( W^{-1} = \text{diag}[\sigma_i] \).

We may regard this test as one of the most commonly used test called largest normalised Residue which is a test on \( l_\infty \)-norm of the measurement residual, normalised so that each element has unit variance [37].

Another classic detector is the \( J(\hat{x}) \) detector, given in [22] as follows:

\[
r^T W_{\hat{x}}^T \hat{x} \geq \gamma
\]

(14)

### 3 Investigation on feasibility of learning techniques for detection of injected false data

In this section, prominent machine learning techniques for detection of injected false data are investigated. It should be noted that authors in [30, 31] showed promising performance of SVM for detection of false data injection, although they used different FDI attack schemes, addressed in Section 1. Authors in [31] showed that SVM had better overall performance comparing to the other machine learning approaches such as K-NN and extended NN.
Moreover, in [30] several supervised methods are used as well as linear and Gaussian SVM, which turned out that SVM performed better than other schemes especially in large-scale systems. Therefore, it is considered as a benchmark in our paper. Nonetheless, in this paper we also investigate MLP too and compare both methods with our proposed scheme.

Both supervised and semi-supervised learning schemes are studied. The main motivation of the supervised methods is that the normal data and tampered ones (due to the attacks) tend to be separable in certain projected space. Given class labels in the historical data, a classifier can be trained to identify attacks. In all of these methods, one of the major challenges is the obstacle of dimensionally, which renders high computation complexity. This problem is resolved by using principle component analysis (PCA) to reduce the dimension of measurements. This new data set is then fed to the learning-based detectors. For supervised methods, we divide our data set (containing both attacked and secured classes) into two sets: training and test sets. After applying PCA for feature reduction, we train supervised classifiers with labelled training set and use test set to evaluate the classifier performance. This procedure is applied to both SVM and MLP, although we chose optimal parameters of these classifiers by grid search, fully explained in Section 4.2. For semi-supervised methods, estimated normal data distribution helps to detect unclassified attacks.

3.1 Support vector machine

Generally, SVM designs a hyperplane or a set of hyperplanes which can be used for classification or regression. Given labelled data set \( \{x_q, y_q\}, q = 1, 2, \ldots, Q \) where \( y_q \in \{1, -1\} \) are labels and \( x_q \in \mathbb{R}^m \), we seek to divide the input space into two areas so that no samples with different labels lie in a same area. SVM finds \( m \) dimensional hyperplane which provides this separation so that its margins (its distance to the nearest samples at each side) are maximised [38]. Hyperplane equation is defined as:

\[
\omega \cdot z + b = 0
\]

(15)

where \( \omega \) is the normal vector to the hyperplane and \( b \) denotes offset. Maximised marginal distance constraint is formulated as follows:

\[
y_q(\omega \cdot z + b) - 1 \geq 0, \quad \forall \ q = 1, \ldots, Q.
\]

(16)

This inequality only works for linearly separable cases. To extend maximised margin constraint to the cases which are not linearly separable, (15) can be modified by defining slack variables \( \varepsilon_q \geq 0, \quad \forall \ q = 1, \ldots, Q \), so it can be rewritten as:

\[
y_q(\omega \cdot z + b) - 1 + \varepsilon_q \geq 0, \quad \forall \ q = 1, \ldots, Q.
\]

(17)

Hence, it can be proven that optimal hyperplane is obtained by:

\[
\min_{\omega, b} \frac{1}{2}\omega^T\omega + C \sum_{q=1}^{Q} \varepsilon_q
\]

Subject to:

\[
y_q(\omega^T \phi(z_q) + b) \geq 1 - \varepsilon_q, \quad \varepsilon_q \geq 0, \quad q = 1, \ldots, Q
\]

(18)

where \( C \) is a parameter to be chosen by user which assigns penalty to errors, and \( \phi(\cdot) \) is a non-linear mapping, named kernel function. This function maps data to a much higher dimension to linearise non-linearities as much as possible, so data would be linearly separable or nearly linearly separable in this new dimension. Note that in case of using kernel function, all \( Z \)'s in (15)-(17) are mapped to the kernel space \( \phi(z) \). We use Gaussian kernel for SVM because of the smoothness it creates. Given a positive definite kernel \( k \) and its corresponding space of functions \( \mathcal{F} \), it turns out if \( k \) is a Gaussian kernel, the functions in \( \mathcal{F} \) are very smooth which makes the learned functions (e.g. regression function) very smooth, as well [40].

3.2 Multi-layer perceptron

Artificial neural network (ANN) is a set of interconnected processing units (called neurons) which imitating human nervous system and often used for estimation or function approximation in machine learning and cognitive science. A widely known class of ANNs is multi-layer perceptron (MLP); a feedforward network which maps set of inputs into a set of targets [41]. The hidden layer is critical because these layers make the MLP a generalised approximation function. Mathematically, an MLP can be represented as follows:

\[
y = g \left( \sum_{q=1}^{Q} w_q \cdot z_q \right)
\]

(19)

In which \( z_q \) represents the ANN input, \( w_q \)'s are the synaptic weights, and \( y \) is the output value from the activation function, \( g \). To achieve particular desired properties (such as convexity), it is possible to define a cost function. A commonly used cost is the mean-squared error such that if the gradient descent approach is utilised for minimisation, well-known backpropagation algorithm is resulted for training MLP. In our application, we use one hidden layer and hyperbolic tangent for activation function because average input value is close to zero.

3.3 Semi-supervised learning method

Semi-supervised learning methods use normal data to find normal operational model. This will help to detect abnormal data points because of the unconformity between anomalies and the model [42, 43]. In other words, secured and attacked measurements lie in the distinct regions of feature space. In this section, we first introduce a well-known algorithm, called statistical anomaly detection, and then we develop and extend this algorithm for FDI detection.

Given normal data sets, \( z_q, q = 1, \ldots, Q \) anomaly detection fits a Gaussian distribution on data and then using a threshold, determines whether new data points are anomaly. Following equations show multivariable Gaussian distribution density function, \( P(Z) \) as a metric given by:

\[
P(Z, \mu, \Sigma) = \frac{1}{(2\pi)^{m/2}|\Sigma|^{1/2}} \exp \left[ -\frac{1}{2} (Z - \mu)^T \Sigma^{-1} (Z - \mu) \right]
\]

(20)

where \( \mu \) and \( \Sigma \) are mean vector and covariance matrix, respectively, \( Q \) is the number of samples, and \( m \) denotes the number of features.

Fig. 2 presents statistical semi-supervised learning method. First we obtain historical data from an n-bus test system which are the measurements as discussed in Section 2. Owing to the size of the system and curse of dimensionality, PCA subroutine is used to reduce the size of the features, and subsequently, only the informative features are used. Afterwards, Gaussian probability distribution is fitted on the principal component data set obtained from training data set and main parameters including \( \mu \) and \( \Sigma \) are determined. In the second phase, a new data set denoted by \( Z_{test} \), which contains both secure and attacked measurement vectors and reduced by PCA, is used to calculate proper threshold. For this end, first the probability distributions of the test data, given by \( P(Z_{test}) \), are obtained based on the determined parameters from the training phase, and sorted, and then \( \delta = P(Z_{test}(1)) = \min P(Z_{test}) \) is chosen as a threshold. Performance index \( F_1 \) score is then calculated for this threshold. Calculation of \( F_1 \) is explained later and given in (29) based on false-positive (fp), false-negative (fn), true-positive (tp), and true-negative (tn) parameters. Since \( F_1 \) score is bigger than the corresponding initial value, i.e. \( F_{1 \text{best}} = 0 \), it is...
assigned as new $P(Z_{\text{test}}(i))$ is assigned as a new threshold. The above procedure iteratively continues until all $P(Z_{\text{test}}(i))$s are used as threshold. Finally, $\delta_{\text{best}}$ will be the proper threshold. The last phase shows evaluation process. To this end, if the probability distribution of a new data point in the evaluation set $Z_{\text{val}}$ is smaller than the threshold, the measurement vector is detected to be under the attack.

### 3.4 Gaussian mixture model for BDDs

Generally, a mixture model is capable of approximating any arbitrary distribution since it is a convex combination of two or more probability density functions [44]. In our proposed method, we fit mixture Gaussian distribution (MGD) on normal data, so generalisation would be improved. MGD is composed of $K > 0$ multivariate Gaussian density components. Each component is mixing in the population with a specific proportion $\alpha_k$. MGD is presented as follows:

$$P(Z|\Theta) = \sum_{k=1}^{K} \alpha_k P_k(Z|\theta_k)$$  \hspace{1cm} (21)

where $P_k(Z|\theta_k)$ is the probability density function of $k$th component which can be obtained from (20). Furthermore, $\alpha_k \geq 0$ denotes mixture weights that satisfy the constraint $\sum_{k=1}^{K} \alpha_k = 1$, and $\Theta = (\alpha_1, \ldots, \alpha_K, \theta_1, \ldots, \theta_K)$ is the set of parameters. The problem of estimation from $Z$ can be interpreted as finding the set of parameters $\Theta$ that gives the maximum likelihood estimate solution

$$\Theta^* = \arg \max \mathcal{L}(\Theta|Z)$$

$$\mathcal{L}(\Theta|Z) = \prod_{q=1}^{Q} \sum_{k=1}^{K} \alpha_k \cdot P_k(z_q|\theta_k)$$  \hspace{1cm} (22)

where $\mathcal{L}(\Theta|Z)$ denotes the likelihood of the data, assuming that $z_{q|\theta}$ are independently distributed. The summation inside the product of likelihood function in (21) prevents the possibility of obtaining an analytical solution. There are some well-known approaches to solve this problem like iterative expectation–maximisation (EM) algorithm or maximum a posteriori estimation [45].

Considering $Y$ as the set of unobserved data or missing data where $y^{q|\theta} \in \{1, \ldots, K\}$ denotes which component of MGD generates $z_{q|\theta}$ observation, for specific number of components, $K$, the likelihood function of complete data $(Z, Y)$ can be written as...
\[ P(Z, Y | \Theta) = \mathcal{N}(Z | Y, \Theta) \]
\[ = p(Z | Y, \Theta)p(Y | \Theta) \]
\[ = \prod_{i=1}^{K} \prod_{j=1}^{Q} \left( \alpha_k \cdot p(z|y, \Theta) \right) \]  
\[ (\Theta^0) (23) \]

where \( I \) is an indicator function, i.e. \( I(y = k) = 1 \) if \( y_i = k \) holds, and \( I(y = k) = 0 \), otherwise. We define \( Q \) as conditional expectation function of complete data \( (Z, Y) \)

\[ Q(\Theta, \Theta^{(n-1)}) = E \left[ \log(p(Z, Y | \Theta)) | Z, \Theta^{(n-1)} \right] \]
\[ = \sum_{k} \sum_{l} \left( p(Y | Z, \Theta^{(n-1)}) \right) \log(p(Z | Y, \Theta)) \]  
\[ (24) \]

where \( \mathcal{Y} \) is the space of all possible values of \( Y \) which satisfy \( \sum_{k} p(Y | Z, \Theta^{(n-1)}) = 1 \) and \( p(Y | Z, \Theta^{(n-1)}) = \prod_{k=1}^{Q} p(y|z_{(k)}^{(i)}), \Theta^{(n-1)} \). It can be proven that with known \( \Theta^{(n-1)} \), maximising \( Q \) results in \( \Theta^{n} \) which also maximises log likelihood function in (23). Therefore EM algorithm performs in two iterative steps. In the first step, given current parameters \( \Theta^{(n-1)} \), expected value of the complete likelihood function is calculated. Then in the M-step, \( \Theta^{(n)} \) which maximises the expectation from E-step, is found. These steps are iterated until a convergence criterion is satisfied.

Based on above discussions, we can define the posterior probability of \( y_{(i)} = k \), given \( z_{(i)}^{(0)} \) as

\[ p(y_{(i)} = k | z_{(i)}^{(0)}, \Theta) = \frac{p(z_{(i)}^{(0)} | y_{(i)} = k, \Theta)}{p(z_{(i)}^{(0)} | \Theta)} \]
\[ = \frac{\alpha_k p(z_{(i)}^{(0)} | y_{(i)} = k, \Theta)}{\sum_{k=1}^{K} \alpha_k p(z_{(i)}^{(0)} | y_{(i)} = k, \Theta)} \]  
\[ (25) \]

For simplicity of notation, \( p(y_{(i)} = k | z_{(i)}^{(0)}, \Theta) \) is denoted by \( r_{ik} \). In mixture models, (23) can be rewritten as

\[ Q(\Theta, \Theta^{(n-1)}) = \frac{\sum_{k=1}^{K} \sum_{i=1}^{Q} r_{ik} \log(\alpha_k \cdot p(z_{(i)}^{(0)} | \Theta^{(n-1)}))}{(26) \]

Therefore, E-step can be calculated as

\[ E(y_{(i)} = k | z_{(i)}^{(0)}, \Theta^{(n-1)}) = p(y_{(i)} = k | z_{(i)}^{(0)}, \Theta^{(n-1)}) = r_{ik} \]  
\[ (27) \]

M-step for Gaussian distribution is given as follows [44]:

\[ \alpha_k = \frac{\sum_{i=1}^{Q} r_{ik}}{Q} \]
\[ \mu_{ik} = \frac{\sum_{i=1}^{Q} r_{ik} z_{(i)}^{(0)}}{\sum_{i=1}^{Q} r_{ik}} \]
\[ \Sigma_{ik} = \frac{\sum_{i=1}^{Q} r_{ik} (z_{(i)}^{(0)} - \mu_{ik})(z_{(i)}^{(0)} - \mu_{ik})^{T}}{\sum_{i=1}^{Q} r_{ik}} \]  
\[ (28) \]

After fitting Gaussian mixture model to the reduced training data, obtaining \( P(Z) \), and calculating main parameters of this distribution, the rest of the algorithm, including choosing threshold and evaluation, is the same as anomaly detection procedure which was depicted thoroughly in Fig. 2. In other words, we use fitted probability assigned to all training data, \( P(Z) \), and then new test data to find the best threshold. Finally, in the last phase, evaluation data is used to show the performance of the proposed model by means of \( F_1 \) score performance measure, which will be explained in Section 4.1.

### 3.5 Time complexity analysis

According to [46] the complexity of support vector machine depends on both \( R \) and \( n^2 \) where \( R \) is the number of free support vectors, \( n \) is the number of training samples and \( S \) is the number of support vectors. Since support vectors grow linearly with the number of samples, the computational cost of solving the SVM problem is bounded by \( O(n^2) \) and \( O(n^3) \), though this discussion is independent of computation of the kernel values. However, training complexity involves \( O(n^3) \) calculations, related to dot product, and \( O(n^2) \) calculations, related to inverse of the kernel matrix, so non-linear SVM computational cost is \( O(n^3) \) [47]. Note that the time complexity is actually \( O(dn^2) \) and \( O(dn^3) \), so dimension reduction might help reducing the training complexity and that is the reason that data dimensionality affects running time. On the other hand, an EM-based Gaussian mixture model involves \( O(dn + Kn^2) \) computations. Therefore, we expect our proposed MGD-based method outperforms SVM in terms of computational complexity.

### 4 Numerical results

#### 4.1 Data generation

To analyse learning-based approaches, IEEE 118-bus test system from MATPOWER [34] is utilised to evaluate the methods which was shown in Fig. 2. For this end, we used Monte-Carlo simulation to generate data sets from different topological networks. In [48], using historical data which changes significantly over a day is highly clustered and can be detected much more easily compared with small changes in the loads, so without loss of generality, it assumes that the power demand follows a uniform distribution in the range of \([0.9L_0 1.1L_0]\) where \( L_0 \) is the base load. Measurements consist of power flows of all branches and injected power of all buses. Hence, in a 118-bus system, measurement vector at a time interval is a 304 × 1 vector. We record measurements from Monte-Carlo simulations for 1000 instances, so measurement matrix becomes a 304 × 1000 matrix. For attacked data, some measurements have been corrupted with the same amount of minimum mean square error (MMSE). In [32], MMSE depends on covariance of the states, covariance of noise and Jacobian matrix, while in the presence of attack it is directly affected by the attack vector. The MMSE for attack is 8 dB in our simulations. These normal and corrupted data are fed to the detectors. Due to the correlated nature of measurements and their high dimensions, we apply PCA for dimension reduction. In our data set, using only four principle components retains 96% of variance. In these tests, 70% of measurements are used as train set and the rest are utilised for performance evaluation. To measure how well the detection schemes are performing, first the true positives (tp), the true negatives (tn), the false positives (fp), and the false negatives (fn) parameters are measured. Then \( F_1 \) score is calculated as follows:

\[ \text{Pr} = \frac{tp}{tp + fp} \]
\[ \text{Re} = \frac{tp}{tp + fn} \]
\[ F_1 = 2 \times \frac{\text{Pr} \times \text{Re}}{\text{Pr} + \text{Re}} \]  
\[ (29) \]

Precision value indicates the probability of how relevant retrieved data are. On the other hand, recall parameter measures the probability of how many relevant data are retrieved. However, \( F_1 \) score combines them by the harmonic mean and the closer to one, the more accurate the classifier [49].

#### 4.2 Model generation and attack detection

In this paper, Gaussian kernel function is used for SVM simulations, as

\[ K(x, x') = \exp \left( -\frac{x - x'}{2\sigma^2} \right) \]  
\[ (30) \]

in which variance \( \sigma \) defines the complexity of decision surface, and is the only parameter that can be chosen desirably, so variance \( \sigma \) alongside penalty factor \( C \) are degrees of freedom which can

---

This is an open access article published by the IET under the Creative Commons Attribution -NonCommercial License (http://creativecommons.org/licenses/by-nc/3.0/)
improve the efficiency of SVM if chosen properly. Fig. 3a shows that $F_1$ score varies with the variations in the values of $\sigma$ and $C$. This diagram is obtained by grid search and each point on the diagram surface is the result of averaging the $F$-score for 10 runs of simulations.

On the other hand, free parameter of MLP is the number of neurons in the hidden layer. Fig. 3b shows $F$-score for different number of neurons. Therefore, 11 neurons were found to be suitable for our problem. It is worth mentioning that for each number of neurons, the simulation is repeated 100 times and then $F$ scores are averaged.

Optimally chosen parameters put classifiers at their best performance and make the comparison feasible. Table 1 shows the performance of two studied supervised methods. Although MLP acts more accurately, but it is much slower than SVM, specifically in training process. However, when the models are formed, they operate in the same order of time (test time). These results show that the only drawback which may affect the MLP performance is its low speed of constructing the model.

The investigated semi-supervised method is based on statistical distribution of the normal data, so it is worth to study data distribution. As mentioned above, after applying PCA, there are four remaining features which their histograms are shown in Fig. 4. Gaussian distribution density function is fitted to the features, while simulation results show that they do not exactly portrait the features’ variations.

Following the procedures shown in Fig. 2, the tampered measurements can be detected by applying a proper threshold $\delta$, so the performance of the detection method heavily depends on the magnitude of this threshold. Large values of $\delta$ results in higher false-negative rate and subsequently lower recall. On the other hand, choosing small $\delta$ increases false alarm rate, as this consequently degrades precision. Fig. 5 presents two-dimensional effects of choosing $\delta$.

To evaluate our proposed method, we fit a Gaussian mixture density on the data set, so the measurements’ complexity is modelled better. Fig. 6 shows data points and evolution of distribution contours with increased number of components. It can be seen that with increased number of components, mixture density distribution fits better on data points. There are some information criterions which can be used to find the appropriate number of component [44, 50]. Using Akaike information criterion defined as $AIC = 2K - 2ln(\mathcal{L}(\Theta^*))$, we choose five components. Fig. 7 demonstrates two- and three-dimensional representation of data points, while the mixture model is fitted on them, which show distribution contours model the measurements complexity.

For better comparison, simulation results of semi-supervised methods are presented in Table 2. Clearly, our proposed method based on mixture model has better performance. From this table, it can be seen that threshold $\delta$ for both methods are close. Therefore,
it can be inferred that it is the model that changes the performance. Although conventional anomaly detection needs less time to train the classifier, but in terms of processing time, they both perform equally. However, the difference between training time of anomaly detection and the proposed method is not considerable. Hence, longer training time for the proposed method cannot be considered as a drawback.

For better evaluation, ROC curves are presented in Fig. 8. For ROC curves threshold varies and then false alarm rate plotted versus detection probability. As shown in Fig. 8a, proposed method outperforms anomaly detection even at lower false alarm rates. For comparing our method with conventional tests given in (12) and (13), Fig. 8b is given, which shows proposed method surpasses the conventional tests.

Combining results in Tables 1 and 2 in Table 3 give us a better vision for comparing all methods that discussed here. From this table, mixture model-based method outperforms the others in terms of accuracy. However, MLP has a good performance, too. On the other hand, supervised methods need much longer time for their calculations. Due to these results, it seems that a mixture model-based method is good enough for FDI detection.

Generally, larger data base leads to better performance for any data-driven classifier, so we compare our proposed methods in term of data set size. Fig. 9a shows that for smaller training data sets, anomaly detection performs better than the others. Although with high number of training samples, MLP outperforms SVM, but it is too sensitive to these numbers. However, SVM shows more robustness comparing to MLP. It can also be inferred that mixture model-based approach cannot be used with <70 samples. However, historical data can be used if training samples are not enough for training. Furthermore, it can be expected that increasing the size of network with the same number of samples would affect data-driven
approaches performance. Fig. 9b shows the performance of mixture model-based approach applied on power networks with different sizes. Networks that used for this analysis are IEEE standard test systems and obtained from MATPOWER package. It can be seen that although the performance declines as the size grows, but even for large networks the performance is still acceptable.

If the load profile in a day or week shows considerable changes comparing with the last day or week, the trained FDI detector with small load change is still useful. To validate this claim and in order to show robustness of our trained detector against large fluctuations, we generated new data for large load profile variations. The size of our new validation data and the number of attacked data points are the same as before. Table 4 shows the new F1 scores versus the percentage of load variations:

| Percentage of Load Variations | F1 Score | Training Time, s | Testing Time, s |
|------------------------------|----------|------------------|-----------------|
| 5%                           | 0.8620   | 0.0979190        | 0.044103        |
| 10%                          | 0.9218   | 3.213071         | 0.09045         |
| 20%                          | 0.7212   | 0.0307           | 0.0167          |
| 30%                          | 0.9565   | 0.5591           | 0.0025          |

Table 4 F1 scores versus the percentage of load variations

The training time is less than a second in our case study as discussed above. If it is necessary, a new threshold can be obtained based on new captured historical data for that region. In this case performance of the updated detector is at least the same as the one reported in this article.
5 Conclusion
While investigations on intentional tampering of the measurements in power and energy systems are gaining more attentions [51], in this paper, we studied the false data injection attack detection from machine learning viewpoint and proposed a semi-supervised approach based on mixture Gaussian distribution. Generally, in supervised learning, data points are labelled, though, in a semi-supervised approach, it is assumed that only labels for part of data (normal data) are known. In supervised models, data points are divided into training and test sets. The training set is used to train the classifier and build the model, while the test set is used to examine the trained classifier. For supervised learning, SVM and MLP, two methods that can approximate measurement non-linearities, are utilised. Although MLP outperforms SVM, but its computational complexity leads to much longer time to build the model. Moreover, SVM shows more robustness against less number of training samples. For semi-supervised method, we assume that the attacked and secured measurements are laid in distinct regions of feature space. Thus, if secured measurements’ distribution is learnt, attacked data points are isolated. With the assumption that a subset of secured measurements is known, we fit normal and Gaussian mixture density distribution on normal data points, and using a proper threshold, pinpoint attacked measurements. As numerical results show, mixture model-based approach shows very promising performance, and is superior to other learning-based approaches.

6 References
[1] Zhan, Z., Xu, M., Xu, S.: ‘Characterizing honeypot-captured cyber attacks: statistical framework and case study’, IEEE Trans. Inf. Forensics Sec., 2013, 8, pp. 1775–1789
[2] Yan, Y., Qian, Y., Sharif, H., et al.: ‘A survey on cyber security for smart grid communications’, IEEE Commun. Surv. Tutor., 2012, 14, pp. 998–1010
[3] Wang, W., Lu, Z.: ‘Cyber security in the smart grid: survey and challenges’, Comput. Netw., 2013, 57, pp. 1344–1371
[4] Ericsson, G.N.: ‘Cyber security and power system communication—essential parts of a smart grid infrastructure’, IEEE Trans. Power Deliv., 2010, 25, pp. 1501–1507
[5] Liu, Y., Ning, P., Reiter, M.K.: ‘False data injection attacks against state estimation in electric power grids’, ACM Trans. Inf. Syst. Secur., 2011, 14, pp. 13
[6] Cheniae, M., Mili, L., Rousselou, P.: ‘Identification of multiple interacting bad data via power system decomposition’, IEEE Trans. Power Syst., 1996, 11, pp. 1555–1563
[7] Huang, Y., Esmaeilifalak, M., Nguyen, H., et al.: ‘Bad data injection in smart grid: attack and defense mechanisms’, IEEE Commun. Mag., 2013, 51, pp. 27–33
[8] Liang, G., Zhao, J., Luo, F., et al.: ‘A review of false data injection attacks against modern power systems’, IEEE Trans. Smart Grid, 2017, 8, (4), pp. 1630–1638
[9] He, H., Yan, J.: ‘Cyber-physical attacks and defences in the smart grid: a survey’, IET Cyber-Physical Syst. Theory Appl., 2016, 1, pp. 13–27
[10] Li, S., Yilmaz, Y., Wang, X.: ‘Quickest detection of false data injection attack in wide-area smart grids’, IEEE Trans. Smart Grid, 2015, 6, pp. 2725–2735
[11] Liu, L., Esmaeilifalak, M., Ding, Q., et al.: ‘Detecting false data injection attacks on power grid by sparse optimisation’, IEEE Trans. Smart Grid, 2014, 5, pp. 612–621
[12] Manandhar, K., Cao, X., Hu, F., et al.: ‘Detection of faults and attacks including false data injection attack in smart grid using kalman filter’, IEEE Trans. Control Netw. Syst., 2014, 1, pp. 370–379
[13] Sedghi, H., Jonckheere, E.: ‘Statistical structure learning to ensure data integrity in smart grid’, IEEE Trans. Smart Grid, 2015, 6, pp. 1924–1933
[14] Ahmadlou, F., Salmassi, F.R.: ‘A cyber-attack on communication link in distributed systems and detection scheme based on H-infinity filtering’, 2017 IEEE Int. Conf. Industrial Technology (ICIT), Toronto, ON, 2017, pp. 698–703
[15] Qi, J., Hahn, A., Lu, X., et al.: ‘Cybersecurity for distributed energy resources and smart inverters’, IET Cyber Physical Syst. Theory Appl., 2016, 1, pp. 28–39
[16] Li, Z., Jin, D., Hannon, C., et al.: ‘Assessing and mitigating cybersecurity risks of traffic light systems in smart cities’, IET Cyber-Physical Syst. Theory Appl., 2016, 1, pp. 60–69
[17] Yu, Z.-H., Chin, W.-L.: ‘Blind false data injection attack using pca approximation method in smart grid’, IEEE Trans. Smart Grid, 2015, 6, pp. 1219–1226
[18] Liu, X., Li, Z.: ‘Local load redistribution attacks in power systems with incomplete network information’, IEEE Trans. Smart Grid, 2014, 5, pp. 1665–1676
[19] Liu, X., Bao, Z., Lu, D., et al.: ‘Modeling of local false data injection attacks with reduced network information’, IEEE Trans. Smart Grid, 2015, 6, pp. 1868–1896
[20] Ozay, M., Enaola, I., Vural, F.T.Y., et al.: ‘Distributed models for sparse attack construction and state vector estimation in the smart grid’, 2012 IEEE Third Int. Conf. Smart Grid Communications (SmartGridComm), 2012, pp. 306–311
[21] Huang, H., Yan, Q., Zhao, Y., et al.: ‘False data separation for data security in smart grids’, Knowl. Inf. Syst., 2017, 52, (3), pp. 815–834
[22] Yang, Q., Yang, J., Yu, W., et al.: ‘On false data-injection attacks against power system state estimation: modeling and countermeasures’, IEEE Trans. Parallel Distrib. Syst., 2014, 25, pp. 717–729
[23] Bobba, R.B., Rogers, K.M., Wang, Q., et al.: ‘Detecting false data injection attacks on dc state estimation’, Preprints of the First Workshop on Secure Control Systems, CPSWEEK, 2010
[24] Bi, S., Zhang, Y.J.: ‘Graphical methods for defense against false-data-injection attacks on power system state estimation’, IEEE Trans. Smart Grid, 2014, 5, pp. 1216–1227
[25] Candès, E.J., Li, X., Ma, Y., et al.: ‘Robust principal component analysis?’, J. ACM, 2011, 58, pp. 11
[26] Fadlullah, Z.M., Fouda, M.M., Kato, N., et al.: ‘An early warning system against malicious activities for smart grid communications’, IEEE Netw., 2011, 25, pp. 50–55
[27] Zhang, Y., Wang, L., Sun, W., et al.: ‘Distributed intrusion detection system in a multi-layer network architecture of smart grids’, IEEE Trans. Smart Grid, 2011, 2, pp. 796–808

[28] Hink, R.C.B., Beaver, J.M., Buckner, M.A., et al.: ‘Machine learning for power system disturbance and cyber-attack discrimination’. 2014 7th Int. Symp. Resilient Control Systems (ISyRCS), 2014, pp. 1–8

[29] Yan, J., He, H., Zhong, X., et al.: ‘Q-learning based vulnerability analysis of smart grid against sequential topology attacks’, IEEE Trans. Inf. Forensics Sec., 2017, 12, (1), pp. 200–210

[30] Ozay, M., Esnaola, I., Vural, F.T.Y., et al.: ‘Machine learning methods for attack detection in the smart grid’, IEEE Trans. Neural Netw. Learn. Syst., 2016, 27, pp. 1773–1786

[31] Yan, J., Yang, B., He, H.: ‘Detection of false data attacks in smart grid with supervised learning’. 2016 Int. Joint Conf. Neural Networks (IJCNN), 2016, pp. 1395–1402

[32] Konst, O., Jia, L., Thomas, R.J., et al.: ‘Malicious data attacks on the smart grid’, IEEE Trans. Smart Grid, 2011, 2, pp. 645–658

[33] Hendrickx, J.M., Johansson, K.H., Jungers, R.M., et al.: ‘Efficient computations of a security index for false data attacks in power networks’, IEEE Trans. Auton. Control, 2014, 59, pp. 3194–3208

[34] Zimmerman, R.D., Murillo-Sánchez, C.E., Thomas, R.J.: ‘MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education’, IEEE Trans. Power Syst., 2011, 26, pp. 12–19

[35] Grainger, J.J., Stevenson, W.D.: ‘Power system analysis’ (McGraw-Hill, 1994)

[36] Wood, A.J., Wollenberg, B.F.: ‘Power generation, operation, and control’ (John Wiley & Sons, 2012)

[37] Schewpepe, F.: ‘Power system state estimation, parts I and II’, IEEE Trans. Power Apparatus Syst., 1970, 89, pp. 120–135

[38] Cristianini, N., Shawe-Taylor, J.: ‘An introduction to support vector machines and other kernel-based learning methods’ (Cambridge University Press, 2000)

[39] Steinwart, I., Christmann, A.: ‘Support vector machines’ (Springer Science & Business Media, 2008)

[40] Williams, C.K., Rasmussen, C.E.: ‘Smoothing, weight functions and equivalent kernels’, in (Eds): ‘Gaussian processes for machine learning’ (The MIT Press, Cambridge, 2006), pp. 24–26

[41] Kulkarni, S., Harman, G.: ‘An elementary introduction to statistical learning theory’ (John Wiley & Sons, 2011), vol. 853

[42] Kuusela, M., Vatanen, T., Malini, E., et al.: ‘Semi-supervised anomaly detection-towards model-independent searches of new physics’. J. Phys. Conf. Ser., 2012, p. 912032

[43] Chandola, V., Banerjee, A., Kumar, V.: ‘Anomaly detection: a survey’, ACM computing surveys (CSUR), 2009, 41, p. 15

[44] McClachlan, G., Peel, D.: ‘Finite mixture models’ (John Wiley & Sons, 2004)

[45] Costa Filho, I.G.: ‘Mixture models for the analysis of gene expression: integration of multiple experiments and cluster validation’, Citeseer, 2008

[46] Bottou, L., Lin, C.-J.: ‘Support vector machine solvers’, in Bottou, L., Chapelle, O., DeCoste, D., et al. (Ed.), ‘Large Scale Kernel Machines’ (MIT Press, 2007), pp. 301–320

[47] Bordes, A., Ertekin, S., Weston, J., et al.: ‘Fast kernel classifiers with online and active learning’, J. Mach. Learn. Res., 2005, 6, pp. 1579–1619

[48] Weng, Y., Negi, R., Faloutsos, C., et al.: ‘Robust data-driven state estimation for smart grid’, IEEE Trans. Smart Grid, 2017, 8, (4), pp. 1956–1967

[49] Chinchor, N., Sandheim, B.: ‘MUC-5 evaluation metrics’. Proc. 5th Conf. Message Understanding, 1993, pp. 69–78

[50] Atkin, M., Yu, D., Francis, B.: ‘A new bayesian approach for determining the number of components in a finite mixture’, Metron, 2015, 73, pp. 155–176

[51] Salmasi, F.: ‘A self-healing induction motor drive with model free sensor tampering and sensor fault detection, isolation, and compensation’, IEEE Trans. Ind. Electron., 2017, 8, (64), pp. 6105–6115