Computers & Security 97 (2020) 101994

Contents lists available at ScienceDirect

Computers & Security

journal homepage: www.elsevier.com/locate/cose

Detecting stealthy false data injection attacks in the smart grid using ensemble-based machine learning

Mohammad Ashrafuzzaman[a,*], Saikat Das[b], Yacine Chakhchoukh[c], Sajjan Shiva[b], Frederick T. Sheldon[a]

[a] Computer Science, University of Idaho, Moscow, ID 83843, United States
[b] Computer Science, University of Memphis, Memphis, TN 38152, United States
[c] Electrical and Computer Engineering, University of Idaho, Moscow, ID 83843, United States

ARTICLE INFO

Article history:
Received 22 April 2020
Revised 21 July 2020
Accepted 29 July 2020
Available online 5 August 2020

Keywords:
Smart grid security
Stealthy false data injection attack
Ensemble-based machine learning
Cyber-physical system security
Critical infrastructure protection

ABSTRACT

Stealthy false data injection attacks target state estimation in energy management systems in smart power grids to adversely affect operations of the power transmission systems. This paper presents a data-driven machine learning based scheme to detect stealthy false data injection attacks on state estimation. The scheme employs ensemble learning, where multiple classifiers are used and decisions by individual classifiers are further classified. Two ensembles are used in this scheme, one uses supervised classifiers while the other uses unsupervised classifiers. The scheme is validated using simulated data on the standard IEEE 14-bus system. Experimental results show that the performance of both supervised individual and ensemble models are comparable. However, for unsupervised models, the ensembles performed better than the individual classifiers.

1. Introduction

With the integration of cyberspace for communication, control and monitoring of the physical processes of generation, transmission, distribution, and consumption of electricity, the smart grid (SG) has evolved into the largest and most complex cyber-physical system (CPS). Even though CPS industry has attempted to "air-gap" the operational technology (OT) of CPSs from their information technology (IT) networks, the OT networks are unfortunately still not fully insulated from IT networks. OT networks are in fact vulnerable to both internal and external cybersecurity threats (Byres, 2013). A study by the Ponemon Institute reports that 90% of organizations relying on OT have experienced at least one business-impacting cyber-attack within the 2 years prior to the report (Ponemon Institute, 2019). Being that SGs are critical national infrastructures, there is widespread concern about reducing the attack surface of SGs. For example, in December 2015, three power distribution companies were taken down in a coordinated cyber-attack, resulting in a power outage for about 225,000 Ukrainians (Liang et al., 2017).

Rapid advancement in machine learning algorithms have enabled them to find natural patterns in data that generate insight and enable better decisions and predictions. The use of data analytics to predict, detect, and prevent security threats is termed *security analytics* (Cybenko and Landwehr, 2012). The incorporation of cyber capabilities into SG functionality has led to the proliferation of new data sources. The availability of abundant data generated by these components has enabled investigators to better study cybersecurity threats and countermeasures in SGs using security analytics (Tan et al., 2017).

1.1. The research problem

State estimation (SE) is the principal tool used by energy management systems (EMS) at power transmission control centers. SE computes voltage magnitudes and phase angles for all of the different buses after collecting measurement data communicated to the control center from remote terminal units (RTUs) equipped with SCADA units (Abur and Gomez-Exposito, 2004). A cyber-attacker can subtly modify these measurement data after compromising RTUs or substation meters, by gaining unauthorized access to the control system or by intruding into the wireless communication networks, for example, by successfully conducting a man-in-the-middle attack. This new type of cyber-attack against SE in EMS is called *false data injection* (FDI) attacks (Liu et al., 2011). The falsely injected measurement data can affect the outcome of the

* Corresponding author.
E-mail addresses: ashr3866@vandals.uidaho.edu (M. Ashrafuzzaman), sdas1@memphis.edu (S. Das), yacinec@uidaho.edu (Y. Chakhchoukh), sshiva@memphis.edu (S. Shiva), sheldon@acm.org (F.T. Sheldon).

https://doi.org/10.1016/j.cose.2020.101994
0167-4048/

SE and can reduce the control center operators level of situational awareness. This can potentially force operators to take erroneous corrective actions against spoofed data, which may disrupt the real-time operation of the grid by impacting OT tools and methods such as contingency analysis, unit commitment, optimal power flow and the computational outcomes of locational marginal pricing for electricity markets. The FDI attacks that impact SE predictions have been presented in several publications (Hug and Giampapa, 2012; Kosut et al., 2011; Li et al., 2017; Liu et al., 2011; Xie et al., 2011). FDI is an important element of a coordinated attack on the power grid and, represents an important class of CPS attack (Xiang et al., 2017).

Numerous attempts have been undertaken to devise detection methods for FDI attacks using traditional statistical approaches and based on the physics of state estimation (Chakhchoukh et al., 2019; Li et al., 2017) and using data-driven machine learning-based security analytics (Tan et al., 2017).

1.2. Motivation

Machine learning approaches work by treating FDI attacks on the measurement data as anomalous compared to the normal data. Hence the problem is reduced to a binary classification task. Most of the related studies up to now have used stand-alone (either supervised, unsupervised or semi-supervised) classification methods. Different classifiers usually perform differently on the same data. In ensemble-based machine learning, multiple classifiers are used together and the results given by these constituent classifiers are further classified by another (second stage) classifier (Dietterich, 2000; Polikar, 2012). Ensemble-based machine learning approaches have been shown to perform well in solving other problems (Das et al., 2020; Moustafa et al., 2018; Zhang et al., 2019).

Supervised methods have the advantage of learning to differentiate the anomalous from normal data using a tagged or labeled dataset. Unsupervised methods, on the other hand, sort data into different clusters where the strength of the clustering lies within the algorithm itself. Among unsupervised methods, novelty and outlier detection strategies have shown significant promise in detecting otherwise hidden anomalies. The supervised methods require large amounts of data to be curated as labeled data (i.e., with ground truth). It is not very difficult to collect data during normal operation. However, “attack” data is very sparsely available making the process of creating labeled datasets onerous. Unsupervised methods do not require labeled data. These methods, though more prone to higher rates of false detection, are better equipped to discover zero-day attacks never before encountered. A scheme that utilizes both supervised and unsupervised classifiers can address and counterbalance the drawbacks inherent to each other.

1.3. Ensemble-based machine learning

In this paper, a security analytics approach that employs two sets of machine learning ensembles is presented to identify stealthy FDI attacks on state estimation. One set of the ensembles uses only supervised methods while the other one uses only unsupervised methods. The supervised methods used are logistic regression (LR), support vector machines (SVM), naive Bayes with Gaussian function (NB), decision tree (DT), and neural networks (NN). The unsupervised methods are one-class SVM (OCSVM), isolation forest (ISOF), elliptic envelope (EE), and local outlier factor (LOF). These individual classifiers when run together, simultaneously or one after another, form the first phase of a machine learning ensemble. The resultant classifications, into either normal or anomalous, are used as input to another classifier, called the *ensemble classifier*. Majority voting, logistic regression, support vector machines, naive Bayes with Gaussian function, decision tree, and artificial neural networks are used as the ensemble classifiers. The models are trained using historical data and the best performing supervised and unsupervised ensemble models are chosen out of six ensemble models of each kind. Both of these “best” models from these experiments would then be deployed online for real-time detection of stealthy FDI attacks on SE.

1.4. Contributions

The major technical contributions of this paper are summarized as follows.

- Design of an ensemble-based machine learning (EML) scheme that consists of ensembles of both supervised and unsupervised classification methods to leverage the advantages of these two learning classes.
- Inclusion of the random forest classifier (RFC) in the EML scheme enabling a reduced number of features in the data thereby assessing and minimizing the effect of the so-called “curse of dimensionality” (Verleysen and François, 2005).
- Generation of power transmission system measurement data using a standard IEEE 14-bus system simulated in MATPOWER (Zimmerman et al., 2011) and introduction of stealthy FDI attacks, i.e., intelligently crafted modifications, to the power flow data.
- Development of a testing and evaluation scheme using the simulated data to enable performance assessment of the different EMLs against stand-alone models using standard evaluation metrics.

1.5. Paper organization

Henceforward this paper is organized as follows. Section 2 summarizes the related works. Section 3 describes the state estimation and stealthy FDI process. Section 4 presents the EML framework. A set of experiments with this framework along with the results are presented in Section 5. Conclusions and future work are presented in Section 6, followed by an acknowledgment and the references. All the abbreviations used throughout the paper are listed in Table A.7 in Appendix A. Additional literature review is presented in Appendix B, and a brief overview of the machine learning methods used in this paper is given in Appendix C.

2. Related work

A number of machine learning based approaches have been proposed to detect false data injection attacks on the state estimation in energy management systems. Table 1 lists the related works showing the learning class, algorithms used for feature selections, algorithms used for training, and how the datasets were generated or obtained for each of the works listed. A review of these works is given in Appendix B. In this section, the works directly related to the current work are discussed.

Ozay et al. (2015) employed supervised and semi-supervised machine learning methods to detect both observable (non-stealthy) and unobservable (stealthy) FDI attacks. The supervised methods used were perceptron, k-nearest neighbors (k-NN), support vector machine (SVM) with linear and Gaussian kernels, sparse logistic regression (SLR), and AdaBoost; and the semi-supervised SVM (S3VM). They evaluated the models using MATPOWER simulations of IEEE 9-, 57- and 118-bus systems. While they used multiple individual classifiers and a boosting algorithm (i.e., AdaBoost), they did not assemble the methods in an ensemble.

Wang et al. (2017a) applied the concept of “first difference”, borrowed from economics and statistics to time-series measurement data to detect time-synchronous attacks on SE. The

Table 1
Overview of machine learning-based detection of FDI attacks on the SE in the SGs.

ML Class	Author(s)	Feature Selections	Training Models	Datasets Generator
Supervised	Esmalifalak et al. (2017)	PCA	Distributed SVM	MATPOWER
	Ozay et al. (2015)		Perceptron, k-NN, SVM & SLR	MATPOWER
	He et al. (2017)	CGB-RBM	CDBN	MATPOWER
	Wang et al. (2017b)		MSA	Simulink
	Wang et al. (2017a)		k-NN, ANN, SVM, NB & DT	MATPOWER
	Ashrafuzzaman et al. (2018)	RFC	ANN	MATPOWER
	Ayad et al. (2018)		RNN	MATPOWER
	Ahmed et al. (2018a)	GA	ED-based	MATPOWER
	Ahmed et al. (2018b)	GA	SVM	MATPOWER
	Niu et al. (2019)		RNN & CNN	MATPOWER
	Wang et al. (2019)		RF, AdaBoost	Simulated
	Alimi et al. (2019)		SVM & ANN	Nigerian Grid
	Camana-Acosta et al. (2020)	KPCA	ERT	MATPOWER
	Mohammadpourfard et al. (2020)	PCA	k-NN	MATPOWER
Unsupervised	Chaojun et al. (2015)		KLD-based	MATPOWER
	Hao et al. (2015)		PCA-approximation	MATPOWER
	Ahmed et al. (2018a)	GA	ED-based	MATPOWER
	Wang et al. (2018)	SAE	LR	MATPOWER
	Yang et al. (2018)	PCA	OCSVM, RC, ISOF & LOF	MATPOWER
	Ahmed et al. (2019)	PCA	ISOF	MATPOWER
Semi-supervised	Ozay et al. (2015)		S3VM & SLR	MATPOWER
	Chakhchoukh et al. (2016)		DRE	MATPOWER
	Foroutan and Salmasi (2017)	PCA	MGD-based	MATPOWER
Reinforcement	Kurt et al. (2018)		SARSA	MATPOWER

"first-difference aware" data is then trained using five supervised models k-NN, neural network (NN), SVM, naive Bayes (NB) and decision tree (DT). They tested their proposed approach on MATPOWER-simulated IEEE 14-bus system. In this case also the authors used the methods as stand-alone methods only.

On the unsupervised learning side, (Yang et al., 2018) used one-class SVM (OCSVM), robust covariance (RC), isolation forest (ISOF) and local outlier factor (LOF) as individual classifiers. They ran these methods using data from a simulated IEEE 14-bus system.

A few observations can be made from the review of the works above and the works listed in Table 1:

1. Different learning classes had been attempted by the researchers. However, no attempts to use adversarial learning or explanation-based learning have been undertaken up to now.
2. None of the investigations combined both supervised and unsupervised methods together in one approach.
3. A few studies mentioned here used multiple classifiers as individual methods, but none used any ensemble methods.
4. Almost all the studies used simulated datasets for validating their approaches. (Alimi et al., 2019) used power flow data from the Nigerian power grid, but they seeded synthetic attacks into the dataset later.

3. False data injection attacks on state estimation

This section describes state estimation (SE) in power transmission system, the mathematical formulation for stealthy false data injection (FDI) attacks on static SE, and how stealthy FDI attacks are carried out on SE.

3.1. State estimation in power transmission systems

State Estimation is used to provide the best estimate of the values of the system's unknown state variables, i.e., voltage magnitudes and phase angles on the system buses, based on the measurements available from the network model and sent by the supervisory control and data acquisition (SCADA) system to the control center. This is known as static state estimation which captures the quasi-static behavior of the power transmission system.

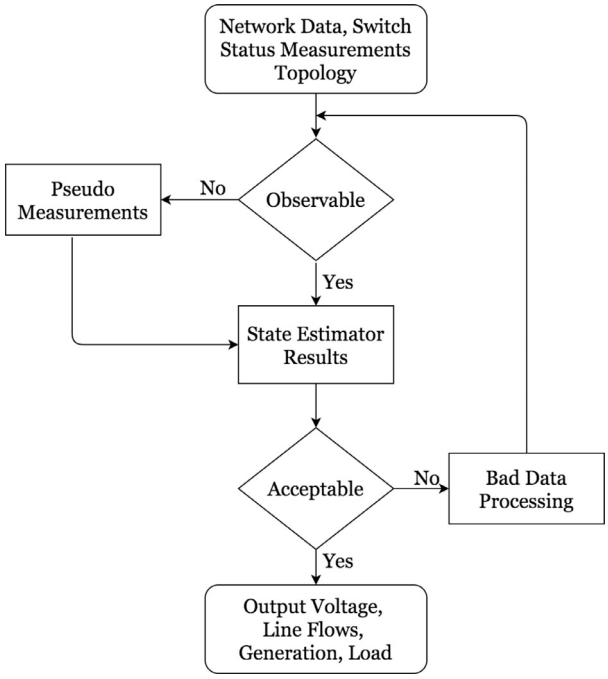


Fig. 1. The state estimation process for power transmission systems (from Thomas and McDonald (2015)).

SE is run every few seconds to few minutes. The functions of the state estimator include identifying and correcting anomalies in the data, suppressing any bad data, and refining the measurements. Finally, SE delivers a set of state variables that are acceptable to the operator and as inputs to other computational programs within the energy management system (EMS) (Thomas and McDonald, 2015). Fig. 1 gives the data flow in a typical state estimator.

These EMS tools are directly dependent on SE outputs process (Thomas and McDonald, 2015):

Contingency analysis is one of the most important tasks of the EMS. It involves performing efficient calculations of system

performance from a set of simplified system contingency conditions and is used for fast estimation of system static stability right after outages.

Unit commitment is an operational planning method used to determine the schedule called the unit commitment schedule which tells the system operators beforehand when and which units to start and shut down during the operation over a pre-specified time, such that the total operating cost for that period is minimized.

Optimal power flow is a technique used to simulate load flow through an AC power system by finding the combination of flows that is operationally and economically optimal.

Locational marginal pricing tools are used to price out the cost of electricity for the local distributors and consumers.

The SE process generates a "residual vector" which is analyzed to detect possible abnormal measurements by checking for residuals that do not obey the Gaussian assumption. While the standard residual analysis tests can identify the presence of errors, it may not detect "stealthy" FDI attacks because an attacker familiar with the target power transmission system topology can carefully craft the amount of data to be injected in such a way that the residual of the original measurement vector remains equal to the residual of the measurement vector that contains the injected (or spoofed) false data.

3.2. Formulation of SE and FDI attacks

The static SE is run after the SCADA units collect the power flows, power injections and voltage magnitudes measurements from the system buses. The static SE estimates the state vector $\boldsymbol{x} \in \mathbb{R}^n$ that contains phase angles and voltage magnitudes at the different buses, where $n = 2k - 1$ and k is the number of buses in the system. For AC static SE, the state vector $\boldsymbol{x}$ obeys the following nonlinear equation:

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

In the above equation, the vector of measurements $\boldsymbol{z} \in \mathbb{R}^m$ contains measurement readings from SCADA units, where m is the number of measurements. The nonlinear vector function $\boldsymbol{h}(\cdot)$ is computed from the grid topology and the transmission lines, transformers and other grid device parameters. The error vector $\boldsymbol{e} \in \mathbb{R}^m$ is assumed Gaussian with a covariance matrix R. The SE is run to compute and estimate the state vector $\boldsymbol{x}$ using an iterative algorithm based on the weighted least squares (WLS) (Abur and Gomez-Exposito, 2004).

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_{k-1} + H_k^{\sharp}(\boldsymbol{z}_k - \boldsymbol{h}(\boldsymbol{x}_{k-1})) \tag{2}$$

where $H_k^{\sharp} = (H_k^{\top} R^{-1} H_k)^{-1} H_k^{\top} R^{-1}$ and H_k is the Jacobian matrix of $\boldsymbol{h}$ with respect to $\boldsymbol{x}$ at step k. The WLS algorithm is optimal under Gaussian noise.

After the algorithm converges, i.e., once $\|\hat{\boldsymbol{x}}_k - \hat{\boldsymbol{x}}_{k-1}\| < \delta$ for some chosen small threshold $\delta > 0$, the obtained residuals are analyzed for possible abnormal measurements by checking for residuals that do not obey the Gaussian assumption. These abnormal or bad data could be due to natural failures such as sensor or communication error, or due to FDI attacks. The chi-square test (χ_2) and the "3σ" rejection rule are used, in most cases, to detect the bad data (Abur and Gomez-Exposito, 2004).

A DC static SE can also be computed. In this case, only phase angles are estimated. All voltages are assumed to be equal to 1pu. The line resistances are neglected and the phase angles differences between buses are assumed to be small. The DC model obeys a linear regression model.

$$\boldsymbol{z} = H\boldsymbol{x} + \boldsymbol{e} \tag{3}$$

3.3. Stealthy FDI attacks on state estimation

Conventional methods detect abnormal or bad data by analyzing the residual (i.e., the difference between the measurement vector $\boldsymbol{z}$ and the calculated value from the SE, i.e., $\boldsymbol{z} - H\hat{\boldsymbol{x}}$). If the largest absolute value of the elements in normalized residual is greater than a pre-defined threshold $\alpha > 0$, (α is generally chosen to be 3) the corresponding measurement is identified as bad data and reported to system operators. The measurement is removed and the estimation is re-executed. Therefore, if the bad data is due to FDI attacks and the injected data are large enough, the conventional residual tests can detect them: these are called *non-stealthy FDI attacks*, or simply FDI attacks. In the non-stealthy case, the attackers do not have the knowledge of the measurement matrix H and they simply manipulate the meter readings to generate random attacks. If the attackers have knowledge of the system topology or know the measurement matrix H, they can carefully and intelligently craft the false data in such a way that the residual $\boldsymbol{r}$, the true residual of the original measurement vector $\boldsymbol{z}$, remains the same as the residual $\boldsymbol{r}_a$ of the measurement vector $\boldsymbol{z}$ with the injected data $\boldsymbol{z}_a$.

$$\boldsymbol{r}_a = \boldsymbol{z}_a - H\hat{\boldsymbol{x}}_a = \boldsymbol{z} - H\hat{\boldsymbol{x}} = \boldsymbol{r} \tag{4}$$

These attacks cannot be detected using the conventional methods based on residual analysis, and are called *stealthy FDI attacks* (Liu et al., 2011).

3.4. Stealthy FDI attack process

False data injection attacks can be carried out on different parts of the power grid, e.g., transmission systems, distribution systems, advanced metering infrastructure, etc. (Liu and Li, 2017). However here, only FDI attacks on the SE in the AC power transmission system are considered.

The following are the steps an adversary may use to commit a stealthy FDI attack:

1. Intrusion into the System (the stepping-stone):
   (a) If the adversary is an outsider, they will hack into the system using any or more of the usual cyber-attacks, e.g., spear-phishing, password-cracking, cracking the cryptographic protection, or using a man-in-the-middle attack by compromising any wired or wireless communication channel, among others.
   (b) An outside adversary can also be successful in installing malware using the means described above or by using social engineering ploys. This malware may have the ability to steal system information, particularly bus topology.
   (c) If the adversary is a trusted insider, then they may already have the access and authority to the needed information.
2. Carry out stealthy FDI attacks:
   (a) After the adversary has gained access into the system and obtained the necessary system information, they can now surreptitiously change measurement data to achieve a stealthy FDI attack.
   (b) The operator and the state estimator assume that the data is correct and estimate the state variable values based on this false assumption.
   (c) Since the state variable values do not represent the actual state of the system, calculation by any of the post-SE tools will be incorrect. This will cause an erroneous situation resulting in adverse operation of the system in turn causing malfunctions or major disruptions.

The goal of the attacker is to disrupt the operation of the transmission system leading to a failure in one or more components or buses, which may even trigger a cascade of failures, i.e., tripping of

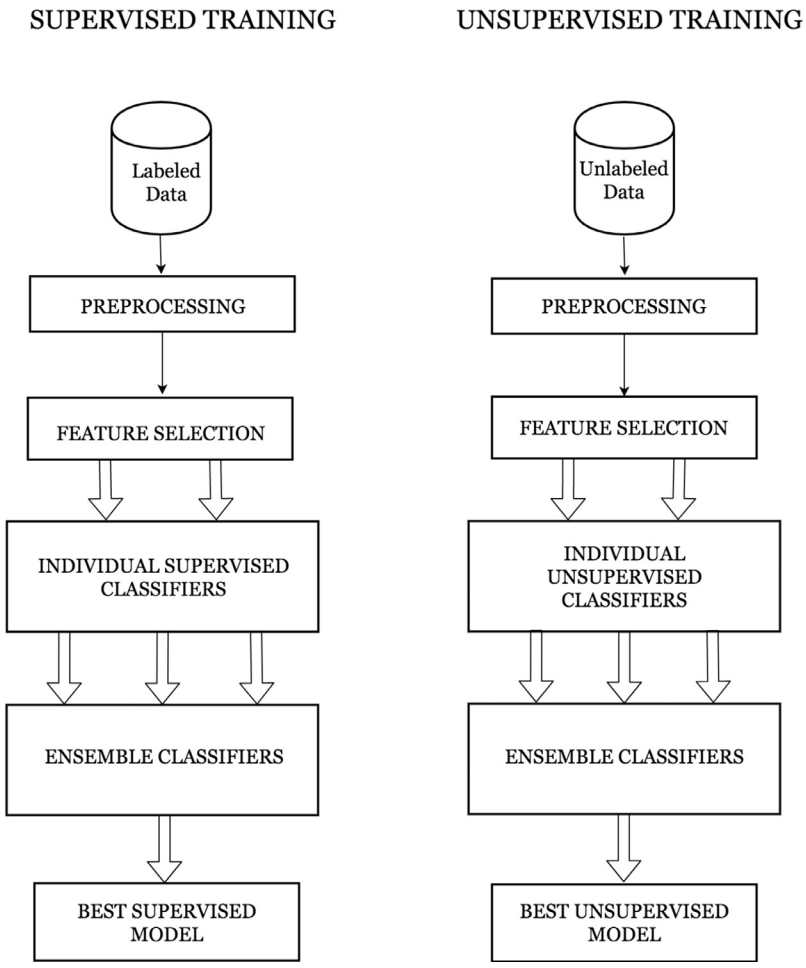


Fig. 2. High-level diagram of the training pipelines for supervised and unsupervised learning.

breakers because of power overload, and causing localized to wide-scale power disruption.

4. Ensemble-based method

This section provides an overview of the ensemble-based stealthy FDI attack detection scheme. The framework implements pipelines of individual and ensemble methods consisting of both supervised and unsupervised classifiers. The detailed schematic diagram depicting the process flow is given in Fig. 2. It shows the individual processing phases, namely (1) data preprocessing, (2) feature selection, (3) classification using individual classifiers, (4) classification using ensemble methods, and (5) building the best performing detection models. The training is performed offline with historical data, and the testing is performed online in real-time deployment mode. A review of the machine learning methods used in this framework is given in Appendix C.

4.1. Data preprocessing

Data preprocessing is the first and foremost task before starting with any machine learning application. There are various sub-tasks in a data preprocessing phase such as removing unwanted data, data conversion, scaling, removing invalid data, etc. At this time, the framework supports standard and min-max scaling.

4.2. Feature selection

The dimension of a dataset increase with the size of the power grid. For example, the number of features in the measurement data for state estimation is 27 for the IEEE 9-bus system and is 1122 for the IEEE 300-bus system. With increasing feature dimensions the complexity and elapsed time for training the models increase very steeply, causing the so-called “curse of dimensionality” (Verleysen and François, 2005). To minimize this problem, feature selection is often used to eliminate the least discriminating features from the dataset, thereby reducing the dimensionality without sacrificing much of the information. Selecting the best feature and best number of features could lead the method to achieve its optimal performance while minimizing its running time. The feature selection phase is one of the most crucial phases of model classification, which can be done by various inbuilt mechanisms or by using domain knowledge. The ensemble framework currently supports random forest classifier (RFC) as a feature selection algorithm.

4.3. Individual classifiers

The scheme employs both supervised and unsupervised classifier methods. In supervised learning, labeled data, i.e., a training set of examples with correct responses, is provided and based on this training the machine learning algorithm generalizes (i.e., learns the patterns in the training data) to classify the unlabeled input sets. In unsupervised learning, the algorithm is trained with

unlabeled data to identify similarities between the inputs that have something in common. These similar inputs are then categorized together. In other words, the algorithm attempts to learn the hidden patterns in the input data, and later predicts responses to test inputs based on the learned patterns.

If trained with well-developed labeled data, supervised learning models perform better than unsupervised models. However, the additional and often cumbersome data engineering needed to label raw data is painstaking and challenging. Moreover, attack data is very sparse. Consequently, the availability of labeled data is not always guaranteed in a timely manner. That is why most datasets are synthetic. Because supervised models are trained with labeled data, they learn the patterns associated with "attacks" very well and can detect such patterns more consistently. However, if the attack pattern is not one of the learned patterns or if the attack is a new one, then the performance of supervised models is highly degraded. In these cases, unsupervised models which flag any out-of-the-ordinary pattern more consistently must be considered. For these reasons, both supervised and unsupervised models are included in the framework.

4.3.1. Supervised classifiers

The supervised classifiers included in the framework are: decision tree (DT), logistic regression (LR), naive Bayes (NB), support vector machine (SVM), and artificial neural network (NN). C.2 provides a brief description of these methods.

4.3.2. Unsupervised classifiers

The unsupervised classifier models included in the framework are: one-class support vector machine with polynomial kernel (OCSVM_P), one-class support vector machine with linear kernel (OCSVM_L), isolation forest (ISOF), elliptic envelope (EE), and local outlier factor (LOF). C.3 provides a brief description of these methods.

4.4. Ensemble classifiers

Ensemble is a learning method where different classifiers are combined into a meta-classifier that has better generalization performance than each individual classifier alone (Dietterich, 2000; Polikar, 2012). Ensemble is a two-stage process. The first stage consists of different classification methods, for example, DT, SVM, LR, and so on, or one base classification algorithm can be used repeatedly with different subsets of the training data. These individual classifiers are run together, simultaneously or one after another. In the second stage, the decisions given by individual classifiers are fed as input to another classifier, called the *ensemble classifier*, for the final decision. Majority voting is the most popular ensemble method, which selects the class label that has been predicted by the majority of the constituent classifiers. Instead of majority voting any binary classification method can be used as the ensemble classifier.

This framework contains two ensembles: one is an ensemble of all the individual supervised classifiers used in the framework, and the other is an ensemble of all the individual unsupervised classifiers. The framework uses six classifiers as the ensemble classifier: majority voting (Ens_MV), logistic regression (Ens_LR), naive Bayes (Ens_NB), neural network (Ens_NN), decision tree (Ens_DT), and support vector machine (Ens_SVM). Fig. 3 shows the ensemble arrangements.

4.5. Best performing models

The datasets in the pipeline go through all the individual and ensemble classifiers. The performance of all the supervised classifiers and the unsupervised classifiers are then compared using standard evaluation metrics. This comparison identifies the best performing supervised ensemble model and the best performing unsupervised ensemble model. These models will be deployed in the state estimation process for real-time detection of stealthy false data injection attack in power transmission systems. Fig. 4 depicts a possible real-time detection pipeline. Any unknown FDI attack detected by the best unsupervised model, can be used to train the supervised ensemble, so that next time around those attacks are detected more quickly and more reliably.

5. Experiments and results

This section presents a set of experiments that were performed using the framework presented above and associated results.

5.1. Attack model

The attack model considered in this paper constitute FDI attacks targeting the static AC (alternating current) state estimation of the transmission system. The considered attack model assumes that the attacker is capable of changing the communicated data such as voltages, currents and power magnitudes. The adversary has only selected partial knowledge of the network topology and therefore can generate a stealthy attack on a single bus only. The attack model assumes one fixed bus is targeted throughout the entire duration of the attack, meaning that it is a *targeted* attack, not a *random* attack.

5.2. Simulation and data generation

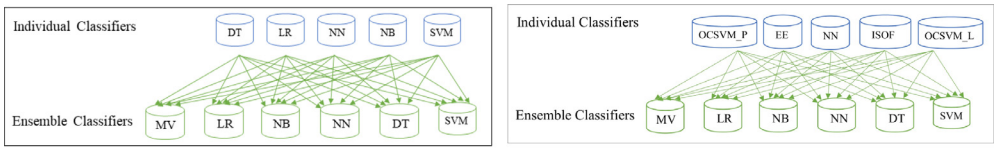
Simulation of the standard IEEE 14-bus system is used for the purpose of generating power flow data. The IEEE 14-bus system has 5 generators and 11 loads (University of Washington, 2018), as shown in Fig. 5. The measurements are obtained from solving power flows using the MATPOWER toolbox (Zimmerman et al., 2011) and adding Gaussian measurement noise. The loads are considered to vary randomly around their average values. The measurements are 40 active power-flows, 14 active power-injections, 40 reactive power flows, 14 reactive power-injections and 14 voltage magnitudes giving a total of 122 measurement features. The dataset consists of 100,000 sets of measurements.

5.3. Experimental setup

The framework has been implemented in the Python programming language and using the machine learning library scikit-learn (Pedregosa et al., 2011). The experiments were executed on a PC with x64 Intel®CORE™i5-6600K CPU @ 3.50GHz, 8GB memory and running a 64-bit Ubuntu Linux operating system.

5.4. Data preprocessing

The dataset generated contains 90% "normal" data and 10% "attack" data implying that the dataset is imbalanced. Classifiers perform poorly when trained with imbalanced datasets, especially for the minority class. In this case, the "attacks" are in the minority class, and the goal is to detect these precisely. To overcome this problem, two popular techniques to balance datasets, namely the synthetic minority over-sampling technique (SMOTE) and the edited nearest neighbor (ENN), were applied to over-sample the "attack" sets of data and under-sample the "normal" sets of data respectively (Batista et al., 2004). After this balancing act the ratio of major and minor class samples in the dataset was 3:2. Since the unsupervised models function as outlier or anomaly detectors



(a) Supervised Ensemble Models. (b) Unsupervised Ensemble Models.

Fig. 3. Five individual classifiers combining with one of six ensemble classifiers to form six ensembles.

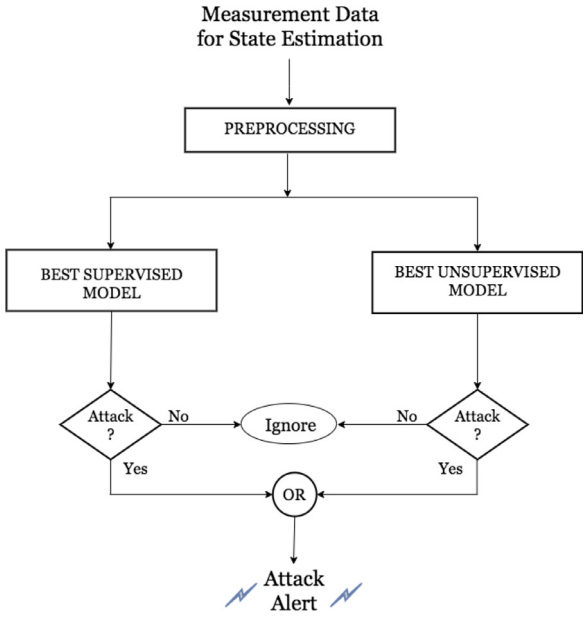


Fig. 4. High-level diagram of the deployment pipeline using the best-performing models.

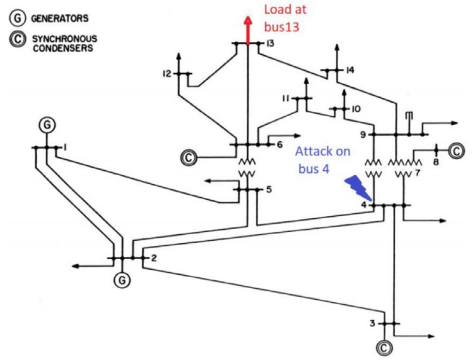


Fig. 5. Diagram of IEEE 14-bus system (adapted from University of Washington (2018)) showing the stealthy FDI attack on bus number 4.

in this paper, the dataset was not balanced for the unsupervised models.

The dataset did not have any missing data or invalid data; consequently, no data cleaning was performed. The data types of all the features in the dataset are numeric, except for the class label which is either "normal" or "attack". All the "normal"s were changed to 0s and "attack"s to 1s. Standard scaling was applied to the dataset. In standard scaling, the features are normalized by removing the mean and scaling to unit variance. Standard scaling replaces the data values in a feature by their z score. For a value x, the z score is calculated as: $z = (x - \mu)/\sigma$, where μ is the mean and σ is the standard deviation of the data values for a given feature.

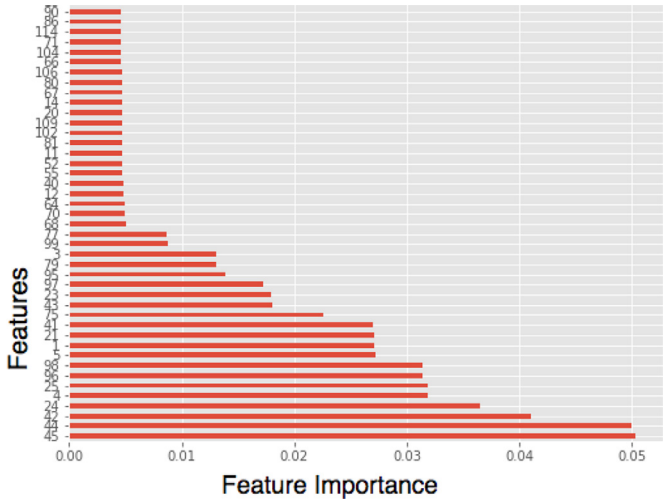


Fig. 6. Plot showing the features in the dataset in order of importance.

Table 2
The Confusion Matrix.

	Predicted Normal	Predicted Attack
Actual Normal	TN	FP
Actual Attack	FN	TP

5.5. Feature selection

The random forest algorithm was used on the dataset to obtain an ordering of the features according to their importance. A plot showing the feature importance is given in Fig. 6. The figure shows that the first 21 features have the largest variances, and therefore only these features were retained in the dataset as the predictor variables, plus the target variable.

5.6. Model training

In the model training phase, the experiments were conducted with individual classification first and then ensemble classification. The experiments ran the data through five supervised and five unsupervised learning models. Six ensemble models were run with the outcomes of the supervised models, and six ensemble models with the outcomes of the unsupervised models. The models were run using grid-search and the best values of the hyper-parameters given by grid-search were retained. The optimal hyper-parameters used for different models are given in Table 3. The dataset was split into two subsets: 70% for training and 30% for testing. To avoid over-fitting and to obtain robust models, 10-fold cross validation over randomly divided training data was used. Then the test data was used for prediction and for measuring model performance.

5.7. Evaluation metrics

A machine learning classification model predicts class labels as output for a given input data. In this case, the class labels are

Table 3
Hyper-parameter values used for different individual and ensemble classifiers.

	Classifier	Short Names	Hyper-parameter Values
Supervised Models	Logistic Regression	LR	random_state=0, solver='lbfgs', multi_class='multinomial'
	Decision Tree	DT	default parameters
	Naive Bayes	NB	alpha=1.0, binarize=0.0, fit_prior=True, class_prior=None
	Neural Network	NN	solver='lbfgs', alpha=1e-5, hidden_layer_sizes=(5, 2), random_state=1
	Support Vector Machine	SVM	C=1.0, kernel='rbf', degree=3, gamma='scale', coef0=0.0, shrinking=True, probability=True
Unsupervised Models	One Class SVM-Polynomial Kernel	OCSVM_P	nu=0.2, kernel='poly', gamma=0.1
	One Class SVM-Linear Kernel	OCSVM_L	nu=0.2, kernel='linear', gamma=0.1
	Local Outlier Factor	LOF	n_neighbors=20, contamination=0.1, novelty=True
	Isolation Forest	ISOF	behaviour='new', max_samples=100, random_state= rng, contamination=0.1
	Elliptic Envelope	EE	support_fraction=1, contamination=0.1, random_state = rng
Ensemble Models	Majority Voting	Ens_MV	none
	Decision Tree	Ens_DT	default parameters
	Naive Bayes	Ens-NB	alpha=1.0, binarize=0.0, fit_prior=True, class_prior=None
	Logistic Regresion	Ens_LR	random_state=0, solver='lbfgs', multi_class='multinomial'
	Neural Network	Ens_NN	solver='lbfgs', alpha=1e-5, novelty=True hidden_layer_sizes=(5, 2), random_state=1
	Support Vector Machine	Ens_SVM	C=1.0, kernel='rbf', degree=3, gamma='scale', coef0=0.0, shrinking=True, probability=True

Table 4
Evaluation metrics values for supervised individual and ensemble models using the test dataset.

Models	F1-Score	Accuracy	Precision	Sensitivity	FPR	Specificity	ROC AUC
LR	0.8439	0.8931	0.9991	0.7304	0.0003	0.9997	0.8639
NB	0.8439	0.8931	0.9991	0.7304	0.0003	0.9997	0.8081
NN	0.8439	0.8931	0.9991	0.7304	0.0003	0.9997	0.8650
DT	0.8438	0.8930	0.9991	0.7302	0.0003	0.9997	0.8797
SVM	0.8439	0.8931	0.9991	0.7304	0.0003	0.9997	0.8642
Ens_MV	0.8439	0.8931	0.9991	0.7304	0.0003	0.9997	–
Ens_LR	0.8472	0.8961	0.9993	0.7353	0.0003	0.9997	0.8675
En_NB	0.8472	0.8961	0.9993	0.7353	0.0003	0.9997	0.8675
Ens_NN	0.8472	0.8961	0.9993	0.7353	0.0003	0.9997	0.8675
Ens_DT	0.8472	0.8961	0.9993	0.7353	0.0003	0.9997	0.8675
Ens_SVM	0.8472	0.8961	0.9993	0.7353	0.0003	0.9997	0.8675

"normal" and "attack". Depending on the predicted class labels, the outcomes for binary classification can be categorized as: (1) True positive (TP): when the model correctly identifies an attack, (2) True negative (TN): when it correctly identifies a normal or non-attack, (3) False positive (FP): when a non-attack is incorrectly identified as an attack, and (4) False negative (FN): when an attack is incorrectly identified as a non-attack. These four categorizes constitute the so-called *confusion matrix* (Hastie et al., 2008) shown in Table 2.

To evaluate the models in this paper, the following metrics derived from the confusion matrix were used (Sokolova and Lapalme, 2009).

1. Accuracy $= (TP + TN)/TotalPopulation$
2. Precision $= TP/(FP + TP)$
3. False Positive Rate (FPR) $= FP/(FP + TN)$
4. Sensitivity $= TP/(FN + TP)$
5. Specificity $= TN/(FP + TN)$
6. F1-score $= 2TP/(2TN + FP + FN)$

Accuracy is the percentage of correct identification over total data instances. *Precision*, also known as the positive predictive value, represents how often the model correctly identifies an attack. *Sensitivity*, also known as the true positive rate (TPR), recall, or detection rate, indicates how many of the attacks the model does identify correctly. Sensitivity intuitively gives the ability of the classifier to find all the positive samples and the precision intuitively gives the ability of the classifier not to label as positive a sample that is negative. *Specificity*, also known as the true negative rate, measures the proportion of actual negatives, i.e., non-attacks, that are correctly identified as such. *F1-score*, also known as F-measure, provides the harmonic average of precision and sensitivity. The *ROC AUC score* is a measure of the diagnostic ability of binary classifier systems. To demonstrate the detection performance of different models over all possible thresholds, the *ROC curves* are plotted.

In addition, the run times (i.e., elapsed times) were measured for comparing the speed of different training models running the all-feature dataset versus the reduced-feature dataset.

5.8. Discussion of results

Table 4 and the corresponding bar-graph in Fig. 7 show the results, i.e., the values for the evaluation metrics, from running all the five supervised classifiers and six ensemble classifiers on a feature-reduced dataset with 21 features. The experiment results show that the values for individual classifiers and those for the

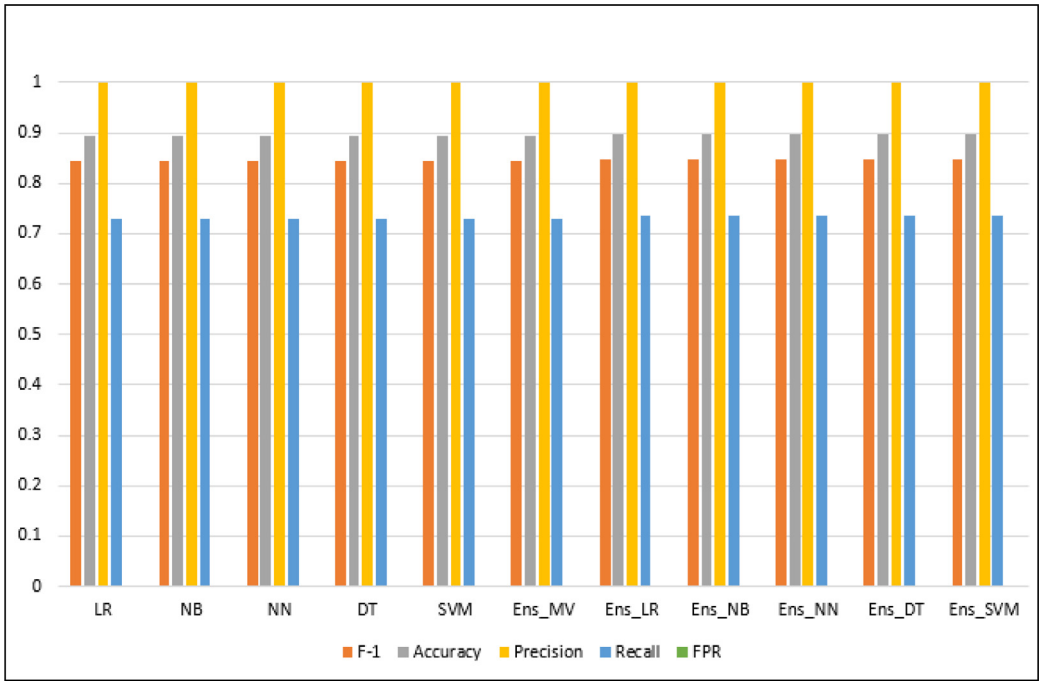


Fig. 7. Bar-graph of the evaluation metrics values for supervised individual and ensemble models.

Table 5
Training times for supervised individual and ensemble models.

Models	Elapsed Time (in s)		
	21 Features	122 Features	Without SVM
LR	0.56	1.02	
NB	0.27	1.03	
NN	0.57	0.83	
DT	1.59	114.52	
SVM	2713.82	8897.83	
Ens_MV	2718.96	9017.94	6.07
Ens_LR	2717.06	9015.59	4.15
En_NB	2717.01	9015.32	4.12
Ens_NN	2717.11	9015.91	4.33
Ens_DT	2717.02	9015.59	4.08
Ens_SVM	2733.31	9031.70	8.55

ensemble classifiers are effectively the same for all the metrics. The table shows that precision values for the models are very close to 100%, whereas accuracy values are about 90%. However, in a classification problem where the goal is to detect the minor class occurrences, the most important metrics are the sensitivity which, in this case, measures the proportion of actual "attacks" that are correctly identified as such; and the specificity which measures the proportion of actual "non-attacks" that are correctly identified as such. For supervised models, the sensitivity values for all the models are very similar, with the ensemble models having a little better number at 73.53%. This indicates that even the best model would be able to detect about 73% of the attacks and the rest 27% will go undetected. The specificity values for the models are 99.97% meaning that the models are able to identify a "non-attack" as such almost always, and will seldom raise a false alert.

Table 5 shows the elapsed time for training the different models. It is notable that not only the ensemble models do not perform any better, but they also take more time to run than the individual models. This is because the ensembles first run all the five individual models and then run the ensemble classifier, and the accumulated elapsed time is therefore higher.

The dataset with all the 122 features was also used to train the models and it was found that the performance numbers for the models do not change at all for this case. Comparison of the values in column two in Table 5 with those in column three, shows that the training times from 122 features are up to 400% more than those with 21 features.

As seen in Section 2 the SVM is a popular model among the researchers working on the problem of detecting stealthy FDI attacks on SE. However, the experiment shows that SVM performs the same as the other models. Moreover, SVM takes much more

Table 6
Evaluation metrics values for unsupervised individual and ensemble models using the test dataset.

Models	F1-Score	Accuracy	Precision	Sensitivity	FPR	Specificity	ROC	Elapsed Time (in s)	
							AUC	21 Features	122 Features
OCSVM_P	0.0834	0.4549	0.0494	0.2661	0.5257	0.4743	0.3702	284.16	7962.94
LOF	0.1388	0.8586	0.1604	0.1223	0.0657	0.9343	0.5283	841.25	3047.80
ISOF	0.3781	0.7939	0.2630	0.6722	0.1935	0.8065	0.7393	7.87	19.16
EE	0.6318	0.9214	0.5606	0.7237	0.0582	0.9418	0.8327	12.39	84.20
OCSVM_L	0.1731	0.5000	0.1023	0.5617	0.5062	0.4938	0.5277	35.14	199.62
Ens_MV	0.4375	0.8892	0.4150	0.4626	0.0669	0.9331	–	1212.97	11316.29
Ens_LR	0.6409	0.9394	0.6739	0.6111	0.0287	0.9713	0.7912	1211.11	11313.98
Ens_NB	0.5502	0.9034	0.4679	0.6676	0.0736	0.9264	0.7969	1211.03	11313.96
Ens_NN	0.6218	0.9216	0.5428	0.7278	0.0595	0.9505	0.8341	1211.35	11315.06
Ens_DT	0.6615	0.9407	0.6689	0.6544	0.0314	0.9686	0.8114	1211.04	11315.05
Ens_SVM	0.6615	0.9407	0.6689	0.6544	0.0314	0.9686	0.8114	1224.92	11316.29

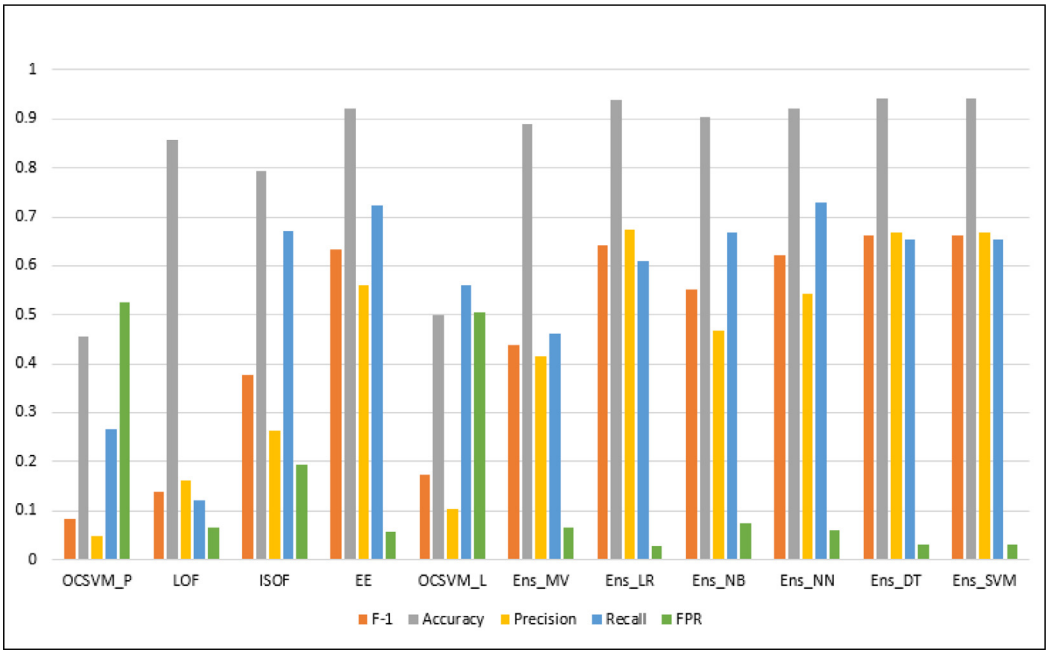
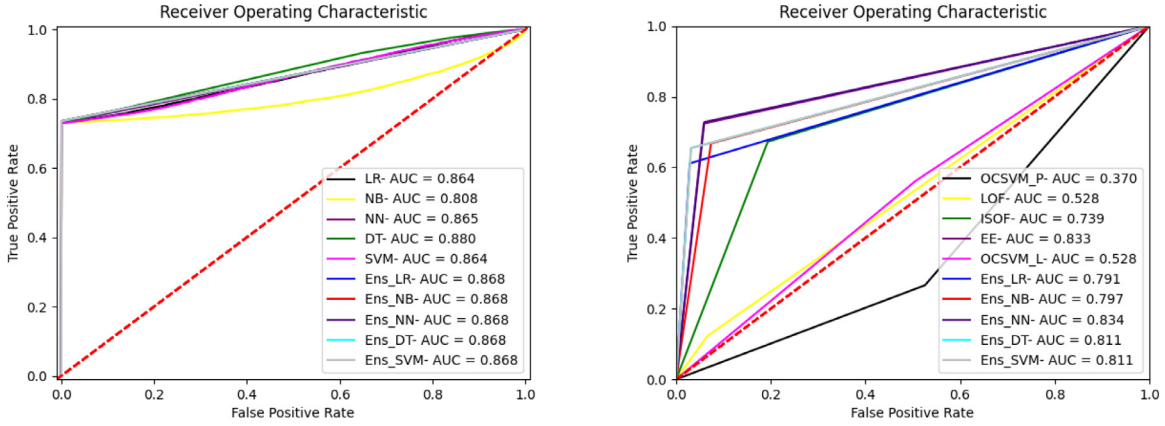


Fig. 8. Bar-graph of the evaluation metrics values for unsupervised individual and ensemble models.



(a) Supervised Models. (b) Unsupervised Models.

Fig. 9. ROC curves for the learning models. ROC curves predict probabilities for two-class problems.

time to train. Whereas the other individual models take less than 2 s to train, SVM takes more than 2700 s or 45 min on the feature-reduced dataset. On the original dataset with 122 features, SVM takes an astounding 8900 s or 2.47 h. This also exemplifies the "curse of dimensionality" and how it can be handled by reducing the feature set. It was also observed that if SVM is taken out as a constituent individual model, then the times taken by the ensemble models were reduced drastically without any reduction in performance. The last column in Table 5 shows times taken by the ensemble models when SVM is not included in the set of the individual models.

Table 6 and the corresponding bar-graph in Fig. 8 show the evaluation metrics for the unsupervised models and the ensemble models for the 21-feature dataset. For the unsupervised models, it was observed that four out of six ensemble models have better accuracy values than any of the individual classifiers. Among the individual classifiers, the elliptic envelope performs with 63.18% F1-score and 92.14% accuracy. Two of the ensemble models, namely ensemble with DT and ensemble with SVM, perform with 66.15% F1-score and 94.07% accuracy. However, considering the sensitivity, the best performing model is the ensemble with NN giving a value of 72.78%, closely followed by the elliptic envelop with 72.37%. This indicates that the best diagnostic or detection ability of either supervised, unsupervised or ensemble models is about 73%. The ensemble using the LR model has the best specificity value of all the unsupervised models at 97.13%. Among the individual models, elliptic envelope has the best specificity value at 94.18%.

For the unsupervised models, training the dataset with all 122 features takes up to 900% more time compared to the times for the 21-feature dataset. In this case also it was observed that the performance numbers do not improve when the models are trained with 122 features.

The ROC curve plots FPR on the X-axis and TPR on the Y-axis. This means that the top left corner of the plot is the "ideal" point, where the $FPR = 0$, and $TPR = 1$. The larger the area under the curve (AUC) the better. The red dotted line indicates the random classification and has an AUC of 0.5. The ROC curves for the supervised and unsupervised models are shown in Fig. 9. Among the supervised individual and ensemble models, the DT model has the largest AUC, and among the unsupervised individual and ensemble models, the ensemble with NN model has the largest AUC.

6. Conclusion

An attacker can execute stealthy false data injection attacks on the state estimation of a power grid to steal electricity, cause minor disruption in the grid, induce cascading failures and/or cause large-scale outages. Therefore, it is of utmost importance that stealthy FDI attacks are detected quickly and accurately. In this paper, a machine learning-based scheme having two ensemble pipelines, one of supervised classifiers and another of unsupervised, was described for detection of stealthy FDI attacks. The scheme also includes RFC for dimension reduction. The scheme was implemented using the Python machine-learning libraries and tested using a standard IEEE 14-bus system simulated by MATPOWER. The performance of the ensemble scheme was compared with the performance of individual classifiers. The performance of individual supervised models are the same as those of the ensemble models. However, for the unsupervised models, the ensemble models performed better than the individual models. The best models have a sensitivity metric value of 73%, meaning that the models would be able to detect 73% of the attacks. For both the supervised and unsupervised models, reducing the feature set increases the training speed many-fold without suffering any degradation in detection rates.

6.1. Future work

The next step with this research is to use these models for power systems with larger numbers of buses to ascertain whether the performance scales reasonably. It would also be interesting to find out whether the training time reduction with the reduced feature-sets is significant for large systems.

This work used simulated data generated by MATPOWER. The future plan is to run the models on realistic datasets generated by Real-time Digital Simulation (RTDS) and physical testbeds.

Another planned work is to extend the framework to incorporate additional methods, and to configure the training pipeline for different sets of methods. An extensible and configurable framework will ensure that such a technique could be standardized for systematic use in transmission smart grids.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Mohammad Ashrafuzzaman: Conceptualization, Methodology, Writing - original draft, Project administration. **Saikat Das:** Software, Validation, Methodology. **Yacine Chakhchoukh:** Data curation, Writing - review & editing. **Sajjan Shiva:** Writing - review & editing. **Frederick T. Sheldon:** Writing - review & editing, Funding acquisition, Supervision.

Acknowledgment

This research was partially supported by an Idaho Global Entrepreneurial Mission (IGEM) grant for Security Management of Cyber-Physical Control Systems, 2016 (Grant number IGEM17-001).

Appendix A. Abbreviations

Table A.7
List of abbreviations used in the paper.

Abbreviation	Term	Abbreviation	Term
AC	alternating current	LSTM	long short-term memory
ANN	artificial neural network	MGD	mixed Gaussian distribution
BDD	bad data detection	ML	machine learning
CDBN	conditional deep belief network	MLP	multi-layer perceptron
CGB	conditional Gaussian-Bernoulli	MSA	margin-setting algorithm
CNN	convolutional neural network	NB	naive Bayes
CPS	cyber-physical system	NN	neural network
DC	direct current	OCSVM	one-class SVM
DRE	density ratio estimation	OCSVM_L	OCSVM with linear kernel
DT	decision tree	OCSVM_P	OCSVM with polynomial kernel
ED	Euclidean distance	OPF	optimal power flow
EE	elliptic envelope	OT	operational technology
EMS	energy management system	PCA	principal component analysis
Ens_DT	ensemble with decision tree	PMU	phasor measurement unit
Ens_LR	ensemble with logistic regression	RBM	restricted Boltzmann machine
Ens_MV	ensemble with majority voting	RC	robust covariance
Ens_NB	ensemble with naive Bayes	RFC	random forest classifier
Ens_NN	ensemble with neural network	RLR	robust logistic regression
Ens_SVM	ensemble with SVM	RNN	recurrent neural network
ERT	extremely randomized tree	ROC	receiver operating characteristic
FDI	false data injection	RTU	remote terminal unit
FDIA	false data injection attack	S3VM	semi-supervised SVM
FN	false negative	SAE	stacked autoencoder
FP	false positive	SARSA	stateactionrewardstateaction
FPR	false positive rate	SE	state estimation
GA	genetic algorithm	SFDI	stealthy false data injection
ISOF	isolation forest	SG	smart grid
IT	information technology	SLR	sparse logistic regression
KLD	Kullback-Leibler distance	SVM	support vector machine
k-NN	k-nearest neighbors	TN	true negative
KPCA	kernel PCA	TP	true positive
LMP	locational marginal pricing	TPR	true positive rate
LR	logistic regression	WLS	weighted least square

Appendix B. Literature review

This section presents a literature review of machine learning-based approaches to detect false data injection attacks on the state estimation used in energy management systems.

The work by (Esmalifalak et al., 2017) employed a distributed support vector machine (SVM) algorithm for training and principal component analysis (PCA) for feature selection. Further, they tested their model on an IEEE 118-bus system simulated by MATPOWER.

He et al. (2017) proposed a conditional deep belief network (CDBN), one of various deep neural network architectures, to reveal the high-dimensional temporal behavior features of the stealthy FDI attacks. The CDBN they employed uses conditional Gaussian–Bernoulli restricted Boltzmann machines (CGB-RBM) for the first hidden layer to extract those features. In all other hidden layers conventional restricted Boltzmann machines (RBM) were used. An IEEE 118-bus system was simulated using MATPOWER to validate their approach.

Wang et al. (2017b) proposed a detection method using the margin setting algorithm (MSA). MSA is a new learning algorithm for spherical classification. They validated their work using phasor measurement unit (PMU) data from an IEEE 6-bus system simulated in MATLAB/Simulink.

Ashrafuzzaman et al. (2018) proposed a feed-forward neural networks (FFNN) based scheme with different configurations for detecting stealthy FDI attacks on SE. They used random forest for feature selection and compared the deep learning scheme with three other methods, namely gradient boosting machines (GBM), generalized linear models (GLM), and the distributed random forests (DRF). Like others, they conducted experiments using data generated for an IEEE 14-bus system using MATPOWER.

Ahmed et al. (2018a) proposed two Euclidean distance-based anomaly detection schemes. The first scheme utilizes unsupervised-learning over unlabeled data to detect outliers or deviations in the measurements. The second scheme employs supervised-learning over labeled data to detect deviations in the measurements. The authors used a genetic algorithm for feature selection. They tested the proposed methods on IEEE 14-, 39-, 57- and 118-bus systems using MATPOWER generated data.

Ahmed et al. (2019) utilized unsupervised learning method isolation forest (ISOF) to detect FDI attacks using simulated data generated by MATPOWER. They reduce the dimensionality of the data using PCA. To demonstrate that ISOF performs better, they compared their results with those of a few other learning methods namely SVM, k-NN, NB and MLP. They did not report how long it took to train the models. It is unexpected that their results of ISOF are better than the other models which are all supervised models. Generally, supervised models perform better in terms of accuracy than unsupervised models on the same dataset.

The proposed framework by (Niu et al., 2019) has two detectors: a network anomaly detector and an FDI attack detector. For detecting the FDI attacks, they formulated these attacks as time-series anomalies and used a long short-term memory (LSTM) based convolutional neural network (CNN). At the same time a recurrent neural network (RNN) with an LSTM cell is deployed to capture the dynamic behavior of cyber activities within an IEEE 39-bus system simulated by MATPOWER.

Wang et al. (2018) proposed a 6-layer stacked auto-encoder (SAE), one of various deep neural network architectures, to detect anomalies caused by stealthy FDI attacks. The SAE was used as an advanced feature extractor and then a classical predictor, namely logistic regression classifier, was used as the last layer. Their proposed system was tested on IEEE 9-, 14-, 30- and 118-bus systems simulated with MATPOWER.

Camana-Acosta et al. (2020) proposed a classification scheme based on the extremely randomized trees (ERT) algorithm and kernel principal component analysis (KPCA) for dimensionality reduction. They evaluated the proposed scheme using MATPOWER-simulated standard IEEE 5- and 118-bus systems.

Mohammadpourfard et al. (2020) used the idea of *concept drift*, unpredictable shifts in the underlying distribution of historical data over time, to develop a technique to improve the performance of existing supervised learning methods in detecting stealthy FDI attacks. They tested their idea using k-NN with PCA as feature extractor and evaluated the performance using a MATPOWER-simulated standard IEEE 14-bus system.

Hao et al. (2015) proposed a sparse PCA-approximation based method to detect stealthy FDI attacks. In this model, identification of real measurements with the availability of sparse data sets is achieved by using recovery functions. The recovery function's accuracy is inversely proportional to the sparsity of available data. As such, this model falls short from identifying FDI attacks when data is too sparse to produce reliably accurate recovery functions. They evaluated their approach on IEEE 9-, 14-, and 57-bus systems simulated with MATPOWER.

Chakhchoukh et al. (2016) proposed a detection method using a semi-supervised machine learning technique known as the density ratio estimation (DRE) (Sugiyama et al., 2012). They tested their model on an IEEE 118-bus system simulated by MATPOWER.

Foroutan and Salmasi (2017) proposed a four-phase classification: (1) dimension reduction using PCA, (2) mixed Gaussian model construction using a positively labeled set, (3) classification threshold selection using a mixture dataset, and finally (4) evaluation using an unlabeled dataset. An IEEE 118-bus system was simulated using MATPOWER to validate their semi-supervised approach by comparing its performance with those of SVM and MLP.

Kurt et al. (2018) proposed a detection mechanism using SARSA, a reinforcement learning algorithm. They formulated the problem of stealthy FDI attack detection as a partially observable Markov decision process (POMDP). Their approach was validated using MATPOWER-generated data for an IEEE 14-bus system.

Appendix C. Machine learning methods

This section briefly describes the machine learning methods used in the scheme presented in this paper. More information on these learning methods can be found in the books by Hastie et al. (2008) and Marsland (2015).

C1. Feature selection algorithm

C1.1. Random forest

Random forest, a popular classification and regression method, can also be used to rank the features in a dataset based on their importance. Random forest consists of a number of decision trees, where every node in the trees is a condition on a single feature, designed to split the dataset into two so that similar response values end up in the same set. At the time of this split, a measure of how much each feature contributes in making this decision is taken. This measure forms the basis of ranking the features according to their importance. Then, a number of the most important features are retained from the ordered list of features while the others are deleted from the dataset to obtain a feature-reduced dataset.

C2. Supervised learning algorithms

C2.1. Logistic regression

Logistic regression (LR) is a classification algorithm that performs very well on linearly separable classes. LR builds on ideas from the field of statistics where the logistic model is used to discern the probability of a true/false class or event. This can be extended to models with more than two classes of events. Each of

the events would be assigned a probability value between 0 and 1, where the sum of all probabilities is unity. The coefficients of the logistic regression algorithm must be estimated from the training data which is done by using maximum-likelihood estimation. The best coefficients would result in a model that would predict a value very close to 1 for the default class and value very close to 0 for the other class.

C2.2. Support vector machine

Support vector machine (SVM) is a group of supervised learning methods that identify the patterns for data classification or regression analysis based on finding a separating hyperplane in the feature space between two classes, in such a way that the distance between the hyperplane and the closest data points for each class is maximized. The SVM algorithm is based on probabilistic statistical learning theory (Cortes and Vapnik, 1995) whereby the approach favors a minimized classification risk rather than optimal classification. Various types of dividing classification surfaces can be realized by applying a kernel, such as linear, polynomial, Gaussian radial basis function (RBF), sigmoid, or hyperbolic tangent (Christianini and Shawe-Taylor, 2000).

C2.3. Naive Bayes

Naive Bayes is a simple classifier used in many machine learning problems. Based on the Bayes theorem this probabilistic classifier helps define the probability of an event based on some prior knowledge of certain conditions associated with that event. The name naive Bayes originates from the fact that the input features are assumed to be independent, even though in practice this may not always be true.

C2.4. Decision tree

Decision tree (DT) is a non-parametric supervised learning method and is used both for classification and regression. DT builds a tree in which each branch shows a probability between a number of possibilities and each leaf shows a decision. The paths from the root of the tree to the leaves represent classification rules. The algorithm collects information and applies the rules for the purpose of the decision to take a particular path. In DT, each level splits the data according to different attributes and attempts to achieve perfect classification with a minimal number of decisions.

C2.5. Artificial neural networks

The computational architecture of neural networks mimics the neural structure and function of the brain forming the interconnected groups of artificial neurons. Each of these artificial neurons is a set of input values and associated weights that trigger the neuron beyond a threshold. The neural network is organized in layers of neurons. The first layer is the input layer and the last one is the output layer. The layers in between these two are called hidden layers. The neural networks attempt to hierarchically learn deep features and correlations in input data by adjusting the weight associated with the neurons. Neural network architectures have many variants with each finding success in specific domains of applications. An extensive review of neural networks can be found in the paper by Schmidhuber (2015).

C3. Unsupervised learning algorithms

C3.1. One-class SVM

One-class classification, also known as unary classification or class-modeling, tries to identify objects of a specific class primarily by learning from an unlabeled training set containing only the objects of that class. Trained in this way, the classifier flags any object not recognized according to the learned generalization as an outlier or anomaly (Khan and Madden, 2009). OCSVM (Schölkopf et al., 2000) is a support vector machine based anomaly detector. Like the supervised SVM models, unsupervised one-class versions also work with different kernels. In this framework one-class SVMs with a linear kernel and a polynomial kernel are used.

C3.2. Isolation forest

Isolation forest (Liu et al., 2012) is an ensemble regressor consisting of a number of isolation trees. Each tree is trained on a random subset of the training data. Isolation forest is used to perform the outlier detection efficiently in high-dimensional datasets. The algorithm isolates observations by randomly selecting a feature and then randomly selecting a split value between the maximum and minimum values of the selected features using recursive partitioning. The required number of splits to isolate a sample is equivalent to the path length from the root to the leaf. The path length from the root to leaf, averaged over a forest of such random trees, is a measure of normality and the decision function. Random partitioning produces noticeably shorter paths for anomalies. Therefore, the trees that collectively produce shorter paths for particular samples are highly likely to be anomalies.

C3.3. Elliptic envelope

Elliptic envelope (EE) attempts to 'draw' an ellipse putting the normal class members inside the ellipse. Any observation outside the ellipse is then classified as an outlier or anomaly. EE models the data as a high dimensional Gaussian distribution with possible covariances between feature dimensions and uses the FAST-Minimum Covariance Determinant (Hubert and Debruyne, 2010; Rousseeuw and Driessen, 1999) to estimate the size and shape of the ellipse.

C3.4. Local outlier factor

The local outlier factor (LOF) (Breunig et al., 2000) first computes the local densities of the data objects using distances given by k-nearest neighbors (k-NN) algorithm. LOF then identifies regions of similar density by comparing the local density of a given data object to the local densities of its neighbors. The data objects that have substantially lower densities than their neighbors are identified as the anomalies.

References

Abur, A., Gomez-Exposito, A., 2004. Power System State Estimation: Theory and Implementation. CRC Press, New York doi:10.1201/9780203913673.

Ahmed, S., Lee, Y., Hyun, S.-H., Koo, I., 2018. Covert cyber assault detection in smart grid networks utilizing feature selection and Euclidean distance-based machine learning. Appl. Sci. 8 (5), 772–792. doi:10.3390/app8050772.

Ahmed, S., Lee, Y., Hyun, S.-H., Koo, I., 2018. Feature selection–based detection of covert cyber deception assaults in smart grid communications networks using machine learning. IEEE Access 6, 27518–27529. doi:10.1109/ACCESS.2018.2835527.

Ahmed, S., Lee, Y., Hyun, S.-H., Koo, I., 2019. Unsupervised machine learning-based detection of covert data integrity assault in smart grid networks utilizing isolation forest. IEEE Trans. Inf. Forens. Secur. 14 (10), 2765–2777. doi:10.1109/TIFS.2019.2902822.

Alimi, O.A., Ouahada, K., Abu-Mahfouz, A.M., 2019. Real time security assessment of the power system using a hybrid support vector machine and multilayer perceptron neural network algorithms. Sustainability 11 (13), 3586. doi:10.3390/su11133586.

Ashrafuzzaman, m., Chakhchoukh, Y., Jillepalli, A., Tosic, P., Conte de Leon, D., Sheldon, F., Johnson, B., 2018. Detecting stealthy false data injection attacks in power grids using deep learning. In: Proceedings of the Fourteenth International Wireless Communications and Mobile Computing Conference (IWCMC). IEEE, pp. 219–225. doi:10.1109/IWCMC.2018.8450487.

Ayad, A., Farag, H.E., Youssef, A., El-Saadany, E.F., 2018. Detection of false data injection attacks in smart grids using recurrent neural networks. In: 2018 Proceedings of the IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT). IEEE, pp. 1–5. doi:10.1109/ISGT.2018.8403355.

Batista, G.E., Prati, R.C., Monard, M.C., 2004. A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD Explor. Newslett. 6 (1), 20–29. doi:10.1145/1007730.1007735.

Breunig, M.M., Kriegel, H.-P., Ng, R.T., Sander, J., 2000. LOF: identifying density-based local outliers. In: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, pp. 93–104. doi:10.1145/342009.335388.
Byres, E., 2013. The air gap: SCADA's enduring security myth. Commun. ACM 56 (8), 29–31. doi:10.1145/2492007.2492018.
Camana-Acosta, M.R., Ahmed, S., Garcia, C.E., Koo, I., 2020. Extremely randomized trees-based scheme for stealthy cyber-attack detection in smart grid networks. IEEE Access 8, 19921–19933. doi:10.1109/ACCESS.2020.2968934.
Chakhchoukh, Y., Lei, H., Johnson, B.K., 2019. Diagnosis of outliers and cyber attacks in dynamic PMU-based power state estimation. IEEE Trans. Power Syst. doi:10.1109/TPWRS.2019.2939192.
Chakhchoukh, Y., Liu, S., Sugiyama, M., Ishii, H., 2016. Statistical outlier detection for diagnosis of cyber attacks in power state estimation. In: Proceedings of the 2016 IEEE Power and Energy Society General Meeting (PESGM). IEEE, pp. 1–5.
Chaojun, G., Jirutitijaroen, P., Motani, M., 2015. Detecting false data injection attacks in AC state estimation. IEEE Trans. Smart Grid 6 (5), 2476–2483. doi:10.1109/TSG.2015.2388545.
Christianini, N., Shawe-Taylor, J., 2000. Support vector machines and other kernel-based learning methods, Cambridge UP.
Cortes, C., Vapnik, V., 1995. Support-vector networks. Mach. Learn. 20 (3), 273–297. doi:10.1007/BF00994018.
Cybenko, G., Landwehr, C.E., 2012. Security analytics and measurements. IEEE Secur. Privacy 10 (3), 5–8. doi:10.1109/MSP.2012.75.
Das, S., Venugopal, D., Shiva, S., 2020. A holistic approach for detecting DDoS attacks by using ensemble unsupervised machine learning. In: Proceedings of the Future of Information and Communication Conference. Springer, pp. 721–738. doi:10.1007/978-3-030-39442-4_53.
Dietterich, T.G., 2000. Ensemble methods in machine learning. In: Proceedings of the International Workshop on Multiple Classifier Systems. Springer, pp. 1–15. doi:10.1007/3-540-45014-9_1.
Esmalifalak, M., Liu, L., Nguyen, N., Zheng, R., Han, Z., 2017. Detecting stealthy false data injection using machine learning in smart grid. IEEE Syst. J. 11 (3), 1644–1652. doi:10.1109/JSYST.2014.2341597.
Foroutan, S.A., Salmasi, F.R., 2017. Detection of false data injection attacks against state estimation in smart grids based on a mixture Gaussian distribution learning method. IET Cyber-Phys. Syst.: Theory Appl. doi:10.1049/iet-cps.2017.0013.
Hao, J., Piechocki, R.J., Kaleshi, D., Chin, W.H., Fan, Z., 2015. Sparse malicious false data injection attacks and defense mechanisms in smart grids. IEEE Transactions on Industrial Informatics 11 (5), 1–12. doi:10.1109/TII.2015.2475695.
Hastie, T., Tibshirani, R., Friedman, J., 2008. The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd New York: Springer-Verlag.
He, Y., Mendis, G.J., Wei, J., 2017. Real-time detection of false data injection attacks in smart grid: a deep learning-based intelligent mechanism. IEEE Trans. Smart Grid doi:10.1109/TSG.2017.2703842.
Hubert, M., Debruyne, M., 2010. Minimum Covariance Determinant, 2. Wiley Interdisciplinary Reviews: Computational Statistics, pp. 36–43. doi:10.1002/wics.61.
Hug, G., Giampapa, J.A., 2012. Vulnerability assessment of AC state estimation with respect to false data injection cyber-attacks. IEEE Trans. Smart Grid 3 (3), 1362–1370. doi:10.1109/TSG.2012.2195338.
Khan, S.S., Madden, M.G., 2009. A survey of recent trends in one class classification. In: Proceedings of the Irish Conference on Artificial Intelligence and Cognitive Science. Springer, pp. 188–197. doi:10.1007/978-3-642-17080-5_21.
Kosut, O., Jia, L., Thomas, R.J., Tong, L., 2011. Malicious data attacks on the smart grid. IEEE Trans. Smart Grid 2 (4), 645–658. doi:10.1109/TSG.2011.2163807.
Kurt, M.N., Ogundijo, O., Li, C., Wang, X., 2018. Online cyber-attack detection in smart grid: a reinforcement learning approach. IEEE Trans. Smart Grid 10 (5), 5174–5185. doi:10.1109/TSG.2018.2878570.
Li, B., Lu, R., Wang, W., Choo, K.-K.R., 2017. Distributed host-based collaborative detection for false data injection attacks in smart grid cyber-physical system. J. Parall. Distrib. Comput. 103, 32–41. doi:10.1016/j.jpdc.2016.12.012.
Liang, G., Weller, S.R., Zhao, J., Luo, F., Dong, Z.Y., 2017. The 2015 Ukraine blackout: implications for false data injection attacks. IEEE Trans. Power Syst. 32 (4), 3317–3318. doi:10.1109/TPWRS.2016.2631891.
Liu, F.T., Ting, K.M., Zhou, Z.-H., 2012. Isolation-based anomaly detection. ACM Trans. Knowl. Discov. Data (TKDD) 6 (1), 1–39. doi:10.1145/2133360.2133363.
Liu, X., Li, Z., 2017. False data attack models, impact analyses and defense strategies in the electricity grid. Electr. J. 30, 35–42. doi:10.1016/j.tej.2017.04.001.
Liu, Y., Ning, P., Reiter, M.K., 2011. False data injection attacks against state estimation in electric power grids. ACM Trans. Inf. Syst. Secur. 14 (1), 13:1–13:33. doi:10.1145/1952982.1952995.
Marsland, S., 2015. Machine Learning: An Algorithmic Perspective. CRC press.
Mohammadpourfard, M., Weng, Y., Pechenizkiy, M., Tajdinian, M., Mohammadi-Ivatloo, B., 2020. Ensuring cybersecurity of smart grid against data integrity attacks under concept drift. Int. J. Electr. Power Energy Syst. 119, 105947. doi:10.1016/j.ijepes.2020.105947.
Moustafa, N., Turnbull, B., Choo, K.-K.R., 2018. An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things. IEEE Internet Things J. 6 (3), 4815–4830. doi:10.1109/JIOT.2018.2871719.
Niu, X., Li, J., Sun, J., Tomsovic, K., 2019. Dynamic detection of false data injection attack in smart grid using deep learning. In: Proceeding of the 2019 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT). IEEE, pp. 1–6. doi:10.1109/ISGT.2019.8791598.
Ozay, M., Esnaola, I., Vural, F.T.Y., Kulkarni, S.R., Poor, H.V., 2015. Machine learning methods for attack detection in the smart grid. IEEE Trans. Neural Netw. Learn. Syst. 27 (8), 1773–1786. doi:10.1109/TNNLS.2015.2404803.
Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: machine learning in Python. J. Mach. Learn. Res. 12, 2825–2830.
Polikar, R., 2012. Ensemble learning. In: Ensemble Machine Learning. Springer, pp. 1–34. doi:10.1007/978-1-4419-9326-7_1.
Ponemon Institute, 2019. Cybersecurity in operational technology: 7 insights you need to know. Accessed on April 10, 2020.
Rousseeuw, P.J., Driessen, K.V., 1999. A fast algorithm for the minimum covariance determinant estimator. Technometrics 41 (3), 212–223.
Schmidhuber, J., 2015. Deep learning in neural networks: an overview. Neural Netw. 61, 85–117. doi:10.1016/j.neunet.2014.09.003.
Schölkopf, B., Williamson, R.C., Smola, A.J., Shawe-Taylor, J., Platt, J.C., 2000. Support vector method for novelty detection. In: Advances in Neural Information Processing Systems, pp. 582–588.
Sokolova, M., Lapalme, G., 2009. A systematic analysis of performance measures for classification tasks. Inf. Process. Manag. 45 (4), 427–437. doi:10.1016/j.ipm.2009.03.002.
Sugiyama, M., Suzuki, T., Kanamori, T., 2012. Density Ratio Estimation in Machine Learning. Cambridge University Press.
Tan, S., De, D., Song, W.-Z., Yang, J., Das, S.K., 2017. Survey of security advances in smart grid: A data driven approach. IEEE Commun. Surv. Tutorials 19 (1), 397–422. doi:10.1109/COMST.2016.2616442.
Thomas, M.S., McDonald, J.D., 2015. Power System SCADA and Smart Grids. CRC Press.
University of Washington, 2018. Power System Test Case Archive. [Online]. Available: http://www.ee.washington.edu/research/pstca/.
Verleysen, M., François, D., 2005. The curse of dimensionality in data mining and time series prediction. In: Proceeding of the International Work-Conference on Artificial Neural Networks. Springer, pp. 758–770. doi:10.1007/11494669_93.
Wang, D., Wang, X., Zhang, Y., Jin, L., 2019. Detection of power grid disturbances and cyber-attacks based on machine learning. J. Inf. Secur. Appl. 46, 42–52. doi:10.1016/j.jisa.2019.02.008.
Wang, H., Ruan, J., Wang, G., Zhou, B., Liu, Y., Fu, X., Peng, J.-C., 2018. Deep learning based interval state estimation of AC smart grids against sparse cyber attacks. IEEE Trans. Ind. Inf. doi:10.1109/TII.2018.2804669.
Wang, J., Tu, W., Hui, L.C., Yiu, S.-M., Wang, E.K., 2017. Detecting time synchronization attacks in cyber-physical systems with machine learning techniques. In: Proceeding of the 2017 IEEE Thirty-seventh International Conference on Distributed Computing Systems (ICDCS). IEEE, pp. 2246–2251. doi:10.1109/ICDCS.2017.25.
Wang, Y., Amin, M., Fu, J., Moussa, H., 2017. A novel data analytical approach for false data injection cyber-physical attack mitigation in smart grids. IEEE Access doi:10.1109/ACCESS.2017.2769099.
Xiang, Y., Wang, L., Liu, N., 2017. Coordinated attacks on electric power systems in a cyber-physical environment. Electr. Power Syst. Res. 149, 156–168. doi:10.1016/j.epsr.2017.04.023.
Xie, L., Mo, Y., Sinopoli, B., 2011. Integrity data attacks in power market operations. IEEE Trans. Smart Grid 2 (4), 659–666. doi:10.1109/TSG.2011.2161892.
Yang, C., Wang, Y., Zhou, Y., Ruan, J., Liu, W., et al., 2018. False data injection attacks detection in power system using machine learning method. J. Comput. Commun. 6 (11), 276. doi:10.4236/jcc.2018.611025.
Zhang, X., Zhao, Z., Zheng, Y., Li, J., 2019. Prediction of taxi destinations using a novel data embedding method and ensemble learning. IEEE Trans. Intell. Transp. Syst. doi:10.1109/TITS.2018.2888587.
Zimmerman, R.D., Murillo-Sánchez, C.E., Thomas, R.J., 2011. MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education. IEEE Trans. Power Syst. 26 (1), 12–19. doi:10.1109/TPWRS.2010.2051168.

Mohammad Ashrafuzzaman is a Ph.D. candidate of computer science at the University of Idaho. He is doing his research in cybersecurity and machine learning. He completed his M.Sc. in computer science at the University of Saskatchewan funded by a Canadian Commonwealth Scholarship. He went on to work for the software industry in companies like RSA Security and Lexmark before coming back to school. He was a recipient of the IGEM research assistantship. He is a senior member of the IEEE.

Saikat Das is a Ph.D. candidate in the Computer Science department at the University of Memphis, TN, USA. He has completed his M.Sc. degree from the same. He worked as an Instructor in Cyber Defense program at Southwest TN Community College and as his industry experience, he worked more than five years as a Senior Software Engineer in various tech companies including Samsung Electronics. His research interests include intrusion detection system with machine learning for cyber-attacks, explanation-based machine learning in cybersecurity, stealth migration protocol in cloud computing, runtime verification in software systems, etc.

Yacine Chakhchoukh received the Ph.D. degree in electrical engineering from Paris-Sud XI University, Paris, France, in 2010. His industrial experience was with the French Electrical Transmission System Operator (RTE-EDF, France). He is currently an Assistant Professor with the University of Idaho, Moscow, ID, USA. His research interests include cyber and physical security for the smart grid, power systems control, and analysis. In 2017, he was the recipient of the IEEE SPS Signal Processing Magazine Best Paper Award.

Sajjan Shiva is a Professor of computer science and served as the Director and founding chairman of the Department from 2002 to August 2015. He is the Director of Game Theory and Cyber Security laboratory. He has served on the computer science faculty at the University of Alabama in Huntsville and Alabama A&M University and has been a consultant to industry and Government. His current research spans cyber security, cloud security, secure software development and machine learning applications. He is a Life Fellow of IEEE and ACM and has authored four books (9 editions) on computer architecture used in more than 120 universities around the world.

Frederick Sheldon has 35+ years of experience from academia, industry and government in various roles working on a diverse set of computer science problems within the scope of software engineering, formal methods, information assurance and security in domains such as embedded real-time avionics/vehicular, energy delivery systems, supply chain, and cryptographic key management. He received the Sigma Xi research and UT-Battelle key contributor and significant event awards and is senior member of IEEE and ACM. He has degrees from the University of Minnesota and University of Texas at Arlington. Currently he is a professor of computer science at the University of Idaho.