Supervised Learning for Detecting Stealthy False Data Injection Attacks in the Smart Grid

Mohammad Ashrafuzzaman[1], Saikat Das[2], Yacine Chakhchoukh[3],
Salahaldeen Duraibi[1 4], Sajjan Shiva[2], and Frederick T. Sheldon[1]

[1] Department of Computer Science, University of Idaho, Moscow, ID, USA
[2] Department of Computer Science, University of Memphis, Memphis, TN, USA
[3] Department of Electrical and Computer Engineering, University of Idaho, Moscow, ID, USA
[4] Department of Computer Science, Jazan University, Jazan, KSA
email: ashr3866@vandals.uidaho.edu, sdas1@memphis.edu, yacinec@uidaho.edu, dura6540@vandals.uidaho.edu, sshiva@memphis.edu, sheldon@uidaho.edu

Abstract. The largest and the most complex cyber-physical systems, the smart grids, are under constant threat of multi-faceted cyber-attacks. The state estimation (SE) is at the heart of a series of critical control processes in the power transmission system. The false data injection (FDI) attacks against the SE can severely disrupt the power systems operationally and economically. With knowledge of the system topology, a cyber-attacker can formulate and execute stealthy FDI attacks that are very difficult to detect. Statistical, physics-based, and more recently, data-driven machine learning-based approaches have been undertaken to detect the FDI attacks. In this paper, we employ five supervised machine learning models to detect stealthy FDI attacks. We also use ensembles, where multiple classifiers are used and decisions by individual classifiers are further classified, to find out if ensembles give any better results. We also use feature selection method to reduce the number of features to investigate if it improves detection rate and speed up the testing process. We run experiments using simulated data from the standard IEEE 14-bus system. The simulation results show that the ensemble classifiers do not perform any better than the individual classifiers. However, feature reduction speeds up the training by many fold without compromising the model performance.

Keywords: Ensemble learning, feature reduction, smart grid, stealthy false data injection attack, supervised machine learning.

1 Introduction

Today's smart grids, with generators, transmission systems, distribution systems, smart meters, distributed energy resources and numerous other physical devices, are integrated with embedded computers, computation, networking and other cyber technologies; and therefore have been transformed into among the largest and most complex cyber-physical systems (CPS). With cyber capabilities, smart grids have inherited vulnerabilities and threats of cyber-attacks. Even though the industry has attempted to "air-gap" operational technology (OT) from information technology (IT) networks toward protecting valuable CPS assets critical to stable operations, OT networks are unfortunately still not fully insulated from the IT networks and are vulnerable to both internal and external threats [1].

The *false data injection* (FDI) attack is a new class of cyber-attacks against the state estimation process in the power grids [2]. The state estimation (SE) is a fundamental tool in the energy management system (EMS) at the power control center. It computes voltage magnitudes and phase angles at all of the different buses of the power system after collecting measurements that are communicated to the control center from remote terminal units (RTUs) [3]. In an FDI attack, an adversary modifies some of these measurement data with the intent of affecting the outcome of the SE, and therefore reduce the control center operators' level of situational awareness [4] forcing the operators to take erroneous corrective actions. Stealthy FDI (SFDI) attacks are those that can not be detected using traditional bad-data detection methods. An SFDI-attacked SE may disrupt the real-time operation of the grid by impacting tools such as contingency analysis, unit commitment, optimal power flow and computation of locational marginal pricing for electricity markets. The SFDI is an important element of a coordinated attack on the power grid and represents an important class of attack on cyber-physical systems [5].

Investigations to devise detection methods for FDI attacks include traditional statistical approaches and approaches based on the physics of the state estimation [6]. In the recent years, data-driven machine learning-based approaches have been gaining popularity. The machine learning approaches treat the FDI attacks on the measurement data as anomalies compared to the normal data and use well-known machine learning algorithms or some improvisations to classify or cluster the FDI attacks as anomalies. It is well-known that different classifiers may perform differently on the same data. Therefore, having an "ensemble" of classifiers may provide a wider coverage for detection of the FDI attacks. In an ensemble, the classification results given by the constituent classifiers are fed into another classifier for final decision [7]. Ensemble-based machine learning approaches have been shown to perform well in solving other problems [8–10].

In this paper, we first use five well-known supervised models, namely logistic regression (LR), naïve Bayes with Gaussian function (NB), decision tree (DT), artificial neural networks (NN) and support vector machine (SVM) as classification models to detect SFDI attacks. Then the outputs or decisions from these five classifiers are fed into seven models separately. This constructs seven ensemble models. The seven *ensemble classifiers* are: majority voting (MV), logistic regression, support vector machine, naïve Bayes with Gaussian function, decision tree, artificial neural networks, and a model that performs an OR operation of all the stand-alone model outputs. Both the stand-alone and ensemble models are trained using historical data. The performances of all the twelve models are compared using standard evaluation metrics to determine the best performing model.

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

- We design a scheme that consists of ensembles of supervised classification models. The ensembles are constructed using different classifiers with the goal to compare their performances and determine the best-performing ensemble classifier.
- We simulate the standard IEEE 14-bus system using MATPOWER [11] and introduce stealthy FDI attacks to the measurement data generated by the simulation.
- We use this simulated data to test and evaluate our proposed scheme. We compare the performances of different stand-alone and ensemble models using standard evaluation metrics and find that the performance of stand-alone models and ensemble models are same.
- We reduce the feature set using random forest and run the models using this feature-reduced dataset. We compare the performances with feature-reduced dataset with those of full-feature dataset and find that the feature-reduced dataset runs much faster while training and provides the same performance as the performance of full-feature dataset.

The remainder of the paper is organized as follows. Section 2 gives a brief review of related works. Section 3 describes the mathematical formulation for the static SE and the stealthy FDI attacks. Section 4 presents the ensemble-based machine learning scheme proposed in this paper. A set of experiments with this scheme along with the results are presented in Section 5. Conclusions are presented in Section 6, followed by the references.

2 Related Works

Esmalifalak et al. [12] employed distributed support vector machine (SVM), Ozay et al. [13] used multi-layer perceptron, k-nearest neighbors (kNN) and SVM; He et al. [14] used conditional deep belief network; Wang et al. [15] used an algorithm based on the margin setting algorithm (MSA); Wang et al. [16] used kNN, neural network, SVM, naïve Bayes and decision tree; Ashrafuzzaman et al. [17] used feed-forward neural networks, gradient boosting machines, generalized linear models and distributed random forests; Ahmed et al. [18] proposed an Euclidean distance-based anomaly detection scheme; Niu et al. [19] used an LSTM-based convolutional neural network; Wang et al. [20] used stacked auto-encoders; Camana-Acosta et al. [21] used extremely randomized trees; and Mohammadpourfard et al. [22] used kNN to detect the FDI attacks. All of these models used above are supervised learning models.

In the unsupervised category, the models used are: isolation forest by Ahmed et al. [23]; sparse principal component analysis by Hao et al. [24]; density ratio estimation by Chakhchoukh et al. [25]; and sparse logistic regression and semi-supervised SVM by Ozay et al. [13]. Kurt et al. [26] used reinforcement learning algorithm SARSA.

Most of these works mention evaluation performance metrics in terms of model accuracy and precision. However, datasets used for detecting attacks, which are sparse compared to the non-attack or normal data, are imbalanced datasets, and for this kind of classifications the more appropriate metrics are sensitivity or recall and false-positive-rate (FPR). These works don't discuss their results using these metrics. Also as shown above, many of the solutions proposed have used single classifiers, and a few used several classifiers as separate individual models, but none of the works used ensemble learning.

3 Stealthy False Data Injection Attacks on State Estimation

State estimation (SE) at the transmission system in electric power grids is a key function in supervisory control, operation and planning of the system. It is used to provide the best estimate of the values of the system's unknown state variables, *i.e.*, voltage magnitudes and phase angles of the system buses, from the measurements available from the network model and sent by the SCADA system to the control center. The functions of the state estimator include identifying and correcting contamination in the data, suppressing any bad data, and refining the measurements. Finally it gives a set of state variables that is acceptable to the operator and as inputs to other computational programs of the energy management system (EMS) [27].

3.1 Formulation of State Estimation

The static state estimation is run after the SCADA units collect the measurements of power flows, power injections and voltage magnitudes from the buses in the system. 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 devices parameters. The error vector $\boldsymbol{e} \in \mathbb{R}^m$ is assumed Gaussian with a covariance matrix R. The SE is executed to compute and estimate the state vector $\boldsymbol{x}$ using an iterative algorithm based on the weighted least squares (WLS).

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

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.

3.2 Stealthy FDI Attacks

State estimation detects abnormal or bad data by analyzing the residual vector (*i.e.*, the difference between the measurement vector $\boldsymbol{z}$ and the calculated value from the state estimation, *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. Therefore, if the bad data is due to FDI attacks and are large enough, the conventional residual tests can detect them: these are called *non-stealthy FDI attacks*, or simply FDI 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 $\boldsymbol{r}$ 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 $\boldsymbol{z}_a$.

$$\boldsymbol{r}_a = \boldsymbol{z}_a - H\hat{\boldsymbol{x}}_a = \boldsymbol{z} - H\hat{\boldsymbol{x}} = \boldsymbol{r} \tag{3}$$

These are *stealthy FDI attacks* and they cannot be detected using the conventional methods based on residual analysis.

4 Machine Learning-based Method

This section provides an overview of the proposed ensemble-based stealthy false data injection attack detection scheme. Figure 1 depicts the block diagram of the training process pipeline. It shows the processing phases, namely 1) data preprocessing, 2) feature selection, 3) classification using individual classifiers, 4) classification using ensemble methods, and 5) obtaining the best performing model. The training is performed offline with historical data, and the testing, when deployed, will be online in real-time.

4.1 Data Preprocessing

Like in any machine learning (ML) training pipeline, data preprocessing phase removes unwanted and invalid data, imputes missing data, converts data suitable for the training, performs scaling, etc. If the dataset is imbalanced, additional steps are taken to balance the dataset.

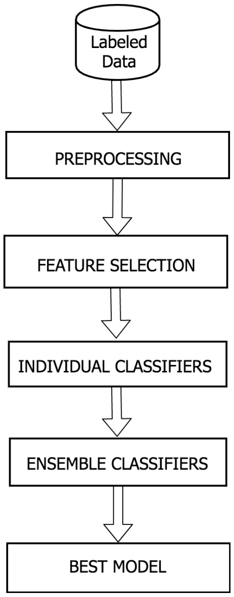

Fig. 1. The training pipeline with the ensemble-based ML framework.

4.2 Feature Selection

Feature selection is used to eliminate the least important features from the dataset, thereby reducing the dimensionality without sacrificing much of the information. Dataset with reduced features often provide better performance and minimize the running time. Our training pipeline currently supports random forest (RF) as a feature selection algorithm.

4.3 Individual Classifiers

The set of individual classifiers constitute the first part of the two-part ensemble mechanism. The classifier models included in our framework are: decision tree (DT), logistic regression (LR), naïve Bayes (NB), artificial neural network (NN) and support vector machine (SVM). The classification decisions given by the five individual models are fed as input to the ensemble classifiers.

4.4 Ensemble Classifiers

Ensemble classifiers take the classifications decisions, a set of five 0s and 1s, from the five individual classifiers as input, and classify these into 0 or 1, where 1 means attack and 0 means normal data. In order to find out the best performing ensemble classifier, we use six supervised classifiers in the pipeline. The ensemble classifiers are: majority voting (Ens_MV), logistic regression (Ens_LR), naïve Bayes (Ens_NB), artificial neural network (Ens_NN), decision tree (Ens_DT), and

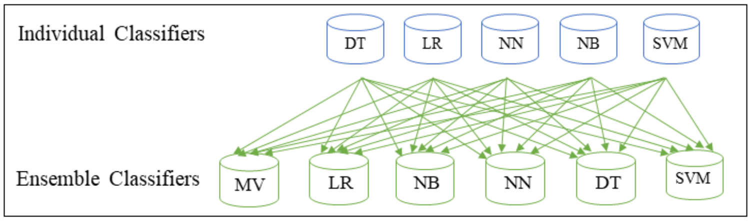

Fig. 2. Diagram showing the ensemble mechanism.

support vector machine (Ens_SVM). Figure 2 shows the ensemble mechanism. In addition to these six models we also use a model that performs OR operations on the outputs of the stand-alone models.

4.5 Best Performing Models

The datasets go through all of the individual and ensemble classifiers in the pipeline. The performance of all the classifiers are then compared using standard evaluation metrics. This comparison identifies the best performing model among the ensemble or individual models. The best model is to be deployed in the state estimation process for real-time detection of stealthy false data injection attacks.

5 Experiments and Results

This section presents an experiment with the framework proposed in this paper and discusses the results.

5.1 Attack Model

In this paper, SFDI attacks targeting the static AC state estimation of the transmission system are considered. The attacker is assumed to be capable of changing the communicated data such as voltages, currents and power magnitudes. The adversary needs only selected partial knowledge of the network topology, which allows them to generate a stealthy attack on a single bus. The considered attack model assumes that only one fixed bus is targeted throughout the entire duration of an attack.

5.2 Simulation and Data Generation

Simulation of the standard IEEE 14-bus system is considered for generating data. The system has 5 generators and 11 loads [28], as shown in Figure 3. The measurements are obtained from solving power flows using the MATPOWER toolbox [11] and adding Gaussian measurement noise. The measurements are 40

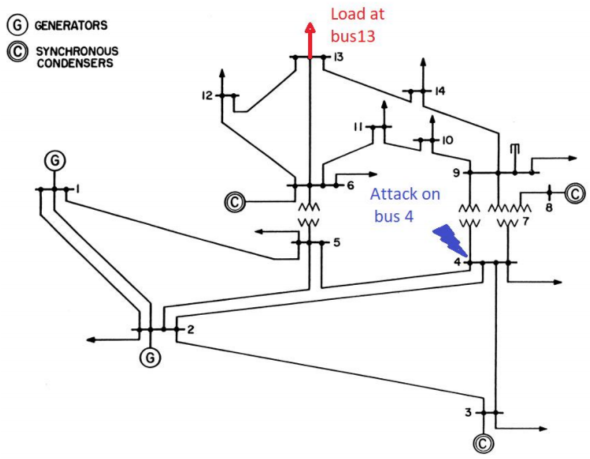

Fig. 3. Diagram of an IEEE 14-bus system (adapted from [28]) showing an attack that targets bus number 4.

active power-flows, 14 active power-injections, 40 reactive power flows, 14 reactive power-injections and 14 voltage magnitudes giving a total of 122 measurements comprising the feature set. A new measurement vector z, corresponding to one set of data, is generated every 60 seconds. The dataset consists of 100,000 sets of measurement data.

5.3 Data Preprocessing

In our dataset, 90% are "normal" data and 10% are "attack" data implying that the dataset is imbalanced. Classifiers perform poorly when trained with imbalanced datasets, specially for the minority class. In our case, the "attacks" are in the minority class, and our goal is to detect these precisely. In order to overcome this problem, we applied the synthetic minority over-sampling technique (SMOTE) and the edited nearest neighbor (ENN) to oversample the "attack" sets of data and undersample the "normal" sets of data [29]. After this balancing act the ratio of major and minor class samples in our dataset was 3:2.

The dataset did not have any missing data or invalid data; so we did not need any data cleaning to perform. However, we applied standard scaling to the data. In standard scaling, the features are normalized by scaling the values in one feature to unit variance.

5.4 Feature Reduction

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 Figure 4. 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. The final feature set includes measurement numbers 45, 44, 42, 24, 4, 25, 96, 98, 5, 1, 21, 41, 75, 43, 23, 97, 95, 79, 3, 99, and 77 (listed here ordered by their importance).

5.5 Model Training

The experiment was conducted with individual classification first and then ensemble classification. The experiment ran the data through five individual models and then the seven ensemble models were run with the outcomes of the individual models. We trained the models using grid-search and retained the best values of the hyper-parameters. We split the dataset into two sets: 70% for training and 30% for testing. To avoid over-fitting and to obtain robust models, we used 10-fold cross-validation over randomly divided training data during training of the models. Then we used the test data for prediction and for measuring model performance.

5.6 Evaluation Metrics

A machine learning model for binary classification predicts class labels as output for a given input data as: 1) True positive (TP): when the model correctly identifies an attack, 2) True negative (TN): when it correctly identifies a normal

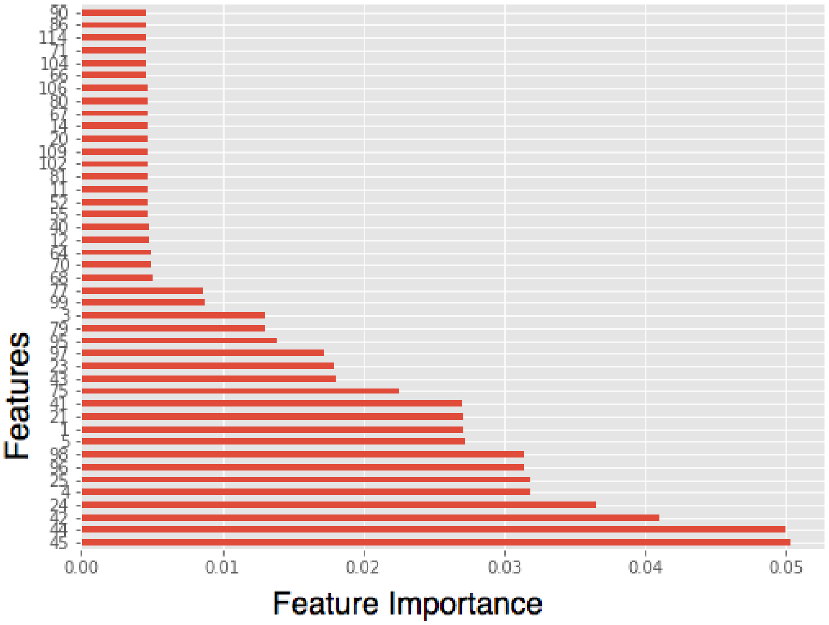

Fig. 4. Graph showing the features in order of importance.

Table 1. Evaluation metrics values for supervised individual and ensemble models using the test dataset.

						ROC	Elapsed Time (in seconds)		
Models	F1-Score	Accuracy	Precision	Recall	FPR	AUC	21 Features	122 Features	Without SVM
LR	0.8439	0.8931	0.9991	0.7304	0.0003	0.8639	0.56	1.02	–
NB	0.8439	0.8931	0.9991	0.7304	0.0003	0.8081	0.27	1.03	–
NN	0.8439	0.8931	0.9991	0.7304	0.0003	0.8650	0.57	0.83	–
DT	0.8438	0.8930	0.9991	0.7302	0.0003	0.8797	1.59	114.52	–
SVM	0.8439	0.8931	0.9991	0.7304	0.0003	0.8642	2713.82	8897.83	–
Ens_MV	0.8439	0.8931	0.9991	0.7304	0.0003	-	2718.96	9017.94	6.07
Ens_LR	0.8472	0.8961	0.9993	0.7353	0.0003	0.8675	2717.06	9015.59	4.15
En_NB	0.8472	0.8961	0.9993	0.7353	0.0003	0.8675	2717.01	9015.32	4.12
Ens_NN	0.8472	0.8961	0.9993	0.7353	0.0003	0.8675	2717.11	9015.91	4.33
Ens_DT	0.8472	0.8961	0.9993	0.7353	0.0003	0.8675	2717.02	9015.59	4.08
Ens_SVM	0.8472	0.8961	0.9993	0.7353	0.0003	0.8675	2733.31	9031.70	8.55
Ens_OR	0.8439	0.8931	0.9992	0.7304	0.0003	-	2716.21	9014.71	3.73

or non-attack, 3) False positive (FP): when a non-attack is incorrectly identified as an attack, and 4) False negatives (FN): when an attack is incorrectly identified as a non-attack. To evaluate the models in this paper, the following metrics [30] are used.

1. Accuracy $= (TP + TN)/Total$
2. Precision $= TP/(FP + TP)$
3. False Positive Rate (FPR) $= FP/(FP + TN)$
4. Recall $= TP/(FN + TP)$
5. F1-Score $= 2TP/(2TN + FP + FN)$

Accuracy is the percentage of true detection over total data instances. *Recall*, also known as the true-positive rate, sensitivity, or detection rate, indicates how many of the attacks the model does identify. *Precision*, also known as the positive predictive value, represents how often the model correctly identifies an attack. *F-Measure* provides the harmonic average of precision and recall. In addition to these five metrics, the *ROC AUC score* which is a measure of the diagnostic ability of binary classifier systems is used. To demonstrate the detection performance of different models over all possible thresholds, the *ROC curves* are plotted. The ROC curve is a graph of false positive rate (FPR) versus true positive rate (TPR). The run times (*i.e.*, elapsed times) were measured for comparing the speed of different training models.

5.7 Discussion of Results

In this section we present and discuss the results from the experiment in terms of the evaluation metrics.

Table 1 shows the results, i.e., the values for the evaluation metrics, from running all the five supervised classifiers and seven ensemble classifiers on a feature-reduced dataset with 21 features. The values for individual classifiers and those for the 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%. The high F1 scores indicate that the

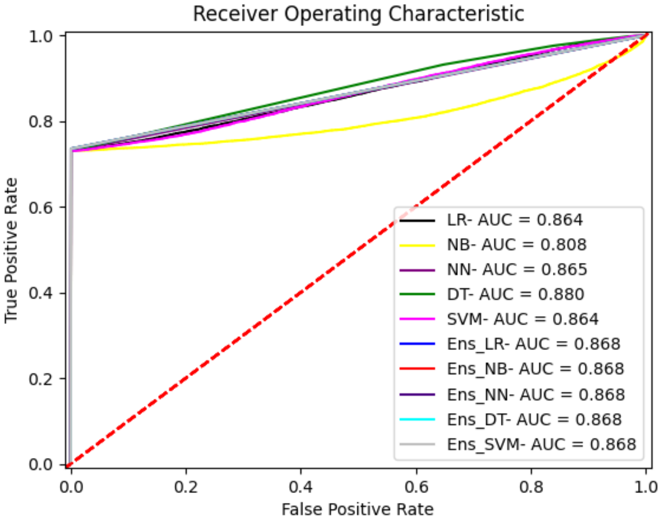

Fig. 5. ROC curves for the learning models. ROC curves predict probabilities for two-class problems.

models are quite robust. Therefore, it indicates that these models are very well-suited for precisely and reliably detecting stealthy false data injection attacks. However, in a classification problem where the goal is to detect the minor class occurrences, the most important metrics are the recall or sensitivity which, in our case, measures the proportion of "attacks" that are correctly identified as such and the FPR which measures the proportion of "non-attacks" that are incorrectly identified as "attacks" raising a false alert. 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 FPR values for the models are 0.03% meaning that the models are able to identify a "non-attack" as such almost always, and will seldom raise a false alert.

Figure 5 illustrates the ROC curves for all the models. It is not surprising that all the curves are the same within statistically negligible range.

Referring back to Table 1, we find that the elapsed time taken to train a model using the dataset having all the 122 features takes up to 400% more time than the corresponding time in the case of the feature-reduced dataset. The table also shows 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 model, and the accumulated elapsed time, therefore, is higher.

As we have seen in Section 2, the SVM is a popular model among the researchers working on the problem of detecting false data injection attacks on the static estimation in the smart grid. However, our experiment shows that SVM performs the same as the other models. Moreover SVM takes much more time to train. Whereas the other individual models take less than 2 seconds to train, SVM takes 2700 seconds or 45 minutes on the feature-reduced dataset. On the original dataset with 122 features, SVM takes an astounding 8900 seconds or 2.47 hours. If we take SVM out as an individual model, then the times taken by the ensemble models reduce drastically without any reduction in performance. The last column in Table 1 shows times taken by the ensemble models when SVM is not included in the set of the individual models.

6 Conclusion

Stealthy false data injection attacks on the state estimation of a power grid can have severe consequences, and an early and accurate detection of these are critical to prevent economic loss or outages. In this paper, we used both stand-alone supervised learning models and ensemble models for detecting stealthy FDI attacks. The ensembles are composed of five individual classifiers and seven ensemble classifiers. The scheme also includes random forest for dimension reduction. We implemented the scheme using the Python machine-learning libraries and tested it using the standard IEEE 14-bus system simulated by MATPOWER. After training the models, we compared the performance of the individual and the ensemble classifiers. The test results demonstrate that the ensemble models do not perform any better than the individual classifiers. The models show 90% accuracy and 100% precision. However, these numbers may be misleading because we are dealing with imbalanced dataset. Looking into the recall and FPR numbers, we find that the models can detect about 73% of the attacks with very low false alerts.

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).

References

1. S. McLaughlin, C. Konstantinou, X. Wang, L. Davi, A.-R. Sadeghi, M. Maniatakos, and R. Karri, "The cybersecurity landscape in industrial control systems," *Proceedings of the IEEE*, vol. 104, no. 5, pp. 1039–1057, 2016.
2. Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," *ACM Transactions on Information and System Security*, vol. 14, no. 1, pp. 13:1–13:33, 2011.

3. A. Abur and A. Gomez-Exposito, *Power System State Estimation: Theory and Implementation.* New York: CRC Press, 2004.
4. C. Alcaraz and J. Lopez, "Wide-area situational awareness for critical infrastructure protection," *Computer*, vol. 46, no. 4, pp. 30–37, 2013.
5. Y. Xiang, L. Wang, and N. Liu, "Coordinated attacks on electric power systems in a cyber-physical environment," *Electric Power Systems Research*, vol. 149, pp. 156–168, 2017.
6. X. Liu and Z. Li, "False data attack models, impact analyses and defense strategies in the electricity grid," *The Electricity Journal*, vol. 30, pp. 35–42, 2017.
7. R. Polikar, "Ensemble learning," in *Ensemble Machine Learning.* Springer, 2012, pp. 1–34.
8. N. Moustafa, B. Turnbull, and K.-K. R. Choo, "An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4815–4830, 2018.
9. X. Zhang, Z. Zhao, Y. Zheng, and J. Li, "Prediction of taxi destinations using a novel data embedding method and ensemble learning," *IEEE Transactions on Intelligent Transportation Systems*, 2019.
10. S. Das, A. M. Mahfouz, D. Venugopal, and S. Shiva, "DDoS intrusion detection through machine learning ensemble," in *2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C).* IEEE, 2019, pp. 471–477.
11. R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 12–19, 2011.
12. M. Esmalifalak, L. Liu, N. Nguyen, R. Zheng, and Z. Han, "Detecting stealthy false data injection using machine learning in smart grid," *IEEE Systems Journal*, 2014.
13. M. Ozay, I. Esnaola, F. T. Y. Vural, S. R. Kulkarni, and H. V. Poor, "Machine learning methods for attack detection in the smart grid," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 8, pp. 1773–1786, 2015.
14. Y. He, G. J. Mendis, and J. Wei, "Real-time detection of false data injection attacks in smart grid: A deep learning-based intelligent mechanism," *IEEE Transactions on Smart Grid*, 2017.
15. Y. Wang, M. Amin, J. Fu, and H. Moussa, "A novel data analytical approach for false data injection cyber-physical attack mitigation in smart grids," *IEEE Access*, 2017.
16. J. Wang, W. Tu, L. C. Hui, S.-M. Yiu, and E. K. Wang, "Detecting time synchronization attacks in cyber-physical systems with machine learning techniques," in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS).* IEEE, 2017, pp. 2246–2251.
17. M. Ashrafuzzaman, Y. Chakhchoukh, A. Jillepalli, P. Tosic, D. Conte de Leon, F. Sheldon, and B. Johnson, "Detecting stealthy false data injection attacks in power grids using deep learning," in *Wireless Communications and Mobile Computing Conference (IWCMC), 14th International.* IEEE, 2018, pp. 219–225.
18. S. Ahmed, Y. Lee, S.-H. Hyun, and I. Koo, "Covert cyber assault detection in smart grid networks utilizing feature selection and Euclidean distance-based machine learning," *Applied Sciences*, vol. 8, no. 5, pp. 772–792, 2018.

19. X. Niu, J. Li, J. Sun, and K. Tomsovic, “Dynamic detection of false data injection attack in smart grid using deep learning,” in *2019 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*. IEEE, 2019, pp. 1–6.
20. H. Wang, J. Ruan, G. Wang, B. Zhou, Y. Liu, X. Fu, and J.-C. Peng, “Deep learning based interval state estimation of AC smart grids against sparse cyber attacks,” *IEEE Transactions on Industrial Informatics*, 2018.
21. M. R. Camana-Acosta, S. Ahmed, C. E. Garcia, and I. Koo, “Extremely randomized trees-based scheme for stealthy cyber-attack detection in smart grid networks,” *IEEE Access*, vol. 8, pp. 19 921–19 933, 2020.
22. M. Mohammadpourfard, Y. Weng, M. Pechenizkiy, M. Tajdinian, and B. Mohammadi-Ivatloo, “Ensuring cybersecurity of smart grid against data integrity attacks under concept drift,” *International Journal of Electrical Power & Energy Systems*, vol. 119, p. 105947, 2020.
23. S. Ahmed, Y. Lee, S.-H. Hyun, and I. Koo, “Unsupervised machine learning-based detection of covert data integrity assault in smart grid networks utilizing isolation forest,” *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 10, pp. 2765–2777, 2019.
24. J. Hao, R. J. Piechocki, D. Kaleshi, W. H. Chin, and Z. Fan, “Sparse malicious false data injection attacks and defense mechanisms in smart grids,” *IEEE Transactions on Industrial Informatics*, vol. 11, no. 5, pp. 1–12, 2015.
25. Y. Chakhchoukh, S. Liu, M. Sugiyama, and H. Ishii, “Statistical outlier detection for diagnosis of cyber attacks in power state estimation,” in *2016 IEEE Power and Energy Society General Meeting (PESGM)*. IEEE, 2016, pp. 1–5.
26. M. N. Kurt, O. Ogundijo, C. Li, and X. Wang, “Online cyber-attack detection in smart grid: A reinforcement learning approach,” *IEEE Transactions on Smart Grid*, vol. 10, no. 5, pp. 5174–5185, 2018.
27. M. S. Thomas and J. D. McDonald, *Power System SCADA and Smart Grids*. CRC press, 2015.
28. University of Washington, *Power System Test Case Archive*. [Online]. Available: http://www.ee.washington.edu/research/pstca/, 2018.
29. G. E. Batista, R. C. Prati, and M. C. Monard, “A study of the behavior of several methods for balancing machine learning training data,” *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 20–29, 2004.
30. M. Sokolova and G. Lapalme, “A systematic analysis of performance measures for classification tasks,” *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009.