Semi-Supervised Outlier Detection and Deep Feature Extraction for Detecting Cyber-Attacks in Smart Grids Using PMU Data

67

Ruobin Qi, Craig Rasband, Jun Zheng, and Raul Longoria

Abstract

Smart grids are facing many challenges including cyber-attacks which can cause devastating damages to the grids. Existing machine learning based approaches for detecting cyber-attacks in smart grids are mainly based on supervised learning, which needs representative instances from various attack types to obtain good detection models. In this paper, we investigated semi-supervised outlier detection algorithms for this problem which only use instances of normal events for model training. Data collected by phasor measurement units (PMUs) was used for training the detection model. The semi-supervised outlier detection algorithms were augmented with deep feature extraction for enhanced detection performance. Our results show that semi-supervised outlier detection algorithms can perform better than popular supervised algorithms. Deep feature extraction can significantly improve the performance of semi-supervised algorithms for detecting cyber-attacks in smart grids.

Keywords

Smart grid · Cyber-attacks · Semi-supervised outlier detection · Deep feature extraction · Autoencoder

67.1 Introduction

Smart grids are electrical grids which manage energy using measurements from smart technologies. Phasor measurement unit (PMU) is one of such technologies, which is responsible for measuring information on power system dynamics including frequency, voltage, phases and phase angles. PMUs in smart grids are synchronized with each other via GPS clocks to produce coordinated measurements. The data from PMUs is gathered by the Phasor-Data Concentrator (PDC) to be spread to other components of the power system [1]. The PMU measurements have been widely used in smart grid applications such as wide-area monitoring, protection and control (WAMPAC) [2, 3] and dynamic state estimation [4]. On the other hand, the transmission of PMU measurements and other control information through communication networks exposes a new cyber-attack surface that could be exploited by potential adversaries to produce devastating damages to smart grids [5, 6]. The 2015 Ukraine Balckout demonstrated how cyber-attacks can directly cause the service outrages of a power grid [7]. Thus, there is a great demand of enhancing the security of smart grids against cyber-attacks.

Machine learning (ML) based approaches have shown to be a promising solution for detecting cyber-attacks in smart grids [8–12]. Majority of the researches focused on using supervised learning to build detection models which requires instances from both normal and attack events to train the detection models. However, it may be hard if not impossible to collect representative instances of various attack types which could result in poor detection models. Semi-supervised learning algorithms solve this problem by only employing instances of normal events to train the detection models. In this paper, we performed a thorough investigation of using various semi-supervised outlier detection algorithms for detecting cyber-attacks in smart grids. We also explored

R. Qi · C. Rasband · J. Zheng (✉)
Department of Computer Science and Engineering, New Mexico Institute of Mining and Technology, Socorro, NM, USA
e-mail: ruobin.qi@student.nmt.edu; craig.rasband@student.nmt.edu; jun.zheng@nmt.edu

R. Longoria
Department of Computer Science, Prairie View A&M University, Prairie, TX, USA

S. Latifi (ed.), *17th International Conference on Information Technology–New Generations (ITNG 2020)*, Advances in Intelligent Systems and Computing 1134,
https://doi.org/10.1007/978-3-030-43020-7_67

to enhance the detection performance of the semi-supervised algorithms with deep feature extraction.

67.2 Related Work

The real-time information of power system dynamics provided by PMUs has been used by a number of machine learning-based approaches for cyber-attack detection. In [8], Hink et al. explored a number of supervised learners for power system disturbance and cyber attack discrimination. In [10], supervised learning algorithms like perceptron, *k*-Nearest Neighbor (*k*-NN), support vector machines (SVMs) and sparse logistic regression (SLR) were applied to predict false data injection attacks. Ensemble learning and feature-level fusion were also investigated. The results showed that machine learning algorithms perform better than algorithms based on state vector estimation in attack detection. Wang et al. [12] proposed an ensemble of random forests combined by AdaBoost for detecting power grid disturbances and cyber-attacks. Feature construction engineering was performed to create new features that help the detection.

There were only few researches on using semi-supervised learning algorithms for attack detection in power systems. Maglaras and Jiang [13] proposed an intrusion detection module for the SCADA (Supervisory Control and Data Acquisition) system based on one-class SVM (OCSVM). The network traces collected from the SCADA system were used to detect malicious attacks. In [14], they further combined OCSVM with K-means recursive clustering for real-time intrusion detection in SCADA systems.

Unlike aforementioned work, in this paper, we explored various semi-supervised learning algorithms for cyber-attack detection in smart grids. Instead of using network traces collected from the cyber domain, the PMU data was used in our study which provides information bridging the cyber and physical domains [12].

67.3 Power System Framework and Cyber-Attacks

The dataset used in our study was generated from a power framework shown in Fig. 67.1 [8]. The framework contains smart electronic devices, supervisory control systems, and network monitoring devices. Two power generators, G1 and G2, provide the power in this system. There are four Intelligent Electronic Devices (IEDs), R1 through R4, which can be toggled to switch four breakers, BR1 through BR4, on or off respectively. Two transmission lines, L1 and L2, connect BR1 to BR2 and BR3 to BR4, respectively. For the IEDs, a distance protection scheme is used in which breakers can be automatically toggled on wherever a fault occurred. Because they don't contain any internal validation to identify any differences between the faults, breakers will be toggled on no matter the fault is a natural anomaly or an attack. The IEDs can also be manually toggled by operators performing maintenance to the power system and/or system components.

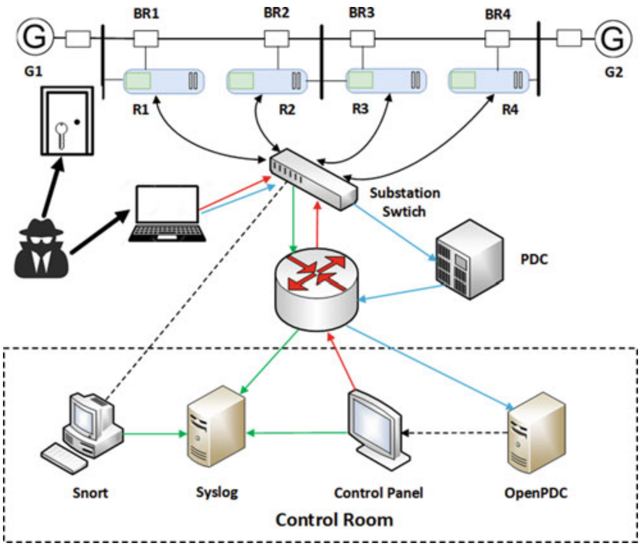


Fig. 67.1 Power system framework [8]

Table 67.1 Summary of operational scenarios

Scenario no.	Description	Event type
1–6	Short-circuit fault	Natural
13, 14	Line maintenance	Natural
7–12	Data injection	Attack
15–20	Remote tripping command injection	Attack
21–30, 35–40	Relay setting change	Attack
41	Normal readings	No event

The dataset was generated from multiple operational scenarios related to no event, natural events and cyber-attack events which are summarized in Table 67.1. Since the goal of our study is to detect cyber-attacks, both no event and natural events described in the follows are treated as normal events: (1) *Short-circuit fault*: a single line-to-ground fault occurred and can specifically be found by reading the percentage range in data; (2) *Line maintenance*: operators toggle one or more IEDs to perform maintenance on certain parts of the power system and its components; (3) *No event*: normal readings.

In addition to the normal events, there are three types of attack events generated by the framework: (1) *Remote tripping command injection attack*: attackers can send commands that toggle IEDs to switch breakers when they can penetrate to the system; (2) *Relay setting change attack*: attackers change settings, such as disabling primary functions of the settings causing the IEDs not toggle the breakers whenever a valid fault or command occurs; (3) *Data injection attack*: attackers change the PMU measurements such as voltage, current and

Table 67.2 Description of features measured by a PMU

Features	Description
PA1:VH-PA3:VH	Phase A—Phase C voltage phase angle
PM1:V-PM3:V	Phase A—Phase C voltage magnitude
PA4:IH-PA6:IH	Phase A—Phase C current phase angle
PM4:I-PM6:I	Phase A—Phase C current magnitude
PA7:VH-PA9:VH	Pos.—Neg.—Zero voltage phase angle
PM7:V-PM9:V	Pos.—Neg.—Zero voltage magnitude
PA10:VH-PA12:VH	Pos.—Neg.—Zero current phase angle
PM10:V-PM12:V	Pos.—Neg.—Zero current magnitude
F	Frequency for relays
DF	Frequency delta (dF/dt) for relays
PA:Z	Appearance impedance for relays
PA:ZH	Appearance impedance angle for relays
S	Status flag for relays

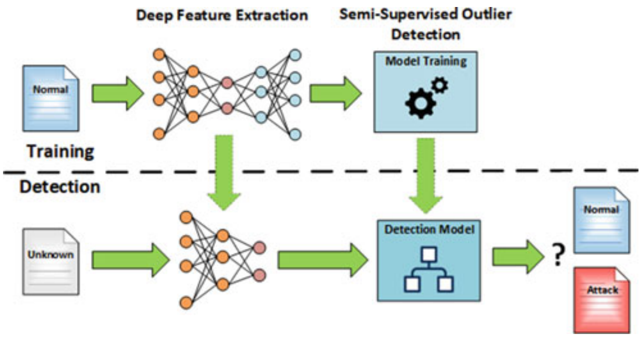


Fig. 67.2 Workflow of detecting cyber-attacks in smart grids with semi-supervised outlier detection and deep feature extraction

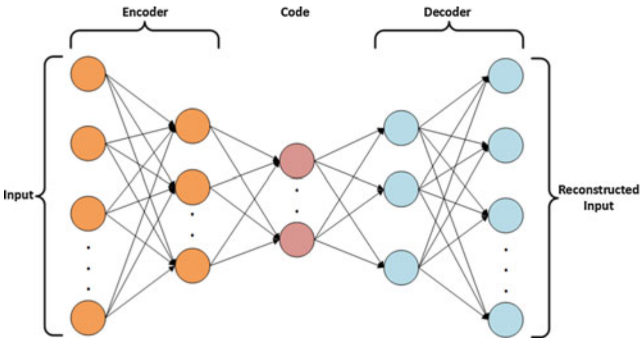


Fig. 67.3 Structure of an autoencoder

sequence components to mimic a valid fault causing the breakers to be switched off.

The power system framework contains four PMUs integrated with relays. Each PMU measures 29 features as described in Table 67.2 which results in a total of 116 features for the four PMUs. Additional features from the log information of the control room in the dataset are not considered in our study as we concentrate on using PMU data to detect cyber-attacks.

67.4 Detecting Cyber-Attacks in Smart Grids

67.4.1 Overview

Figure 67.2 shows the workflow of detecting cyber-attacks in smart grids using semi-supervised outlier detection and deep feature extraction. The training dataset for building the detection model is prepared with instances of normal events. The dimensionality of feature space is then reduced through deep feature extraction with autoencoder. Finally, a detection model is trained using a semi-supervised outlier detection algorithm with the extracted features. During the detection stage, an unknown instance is transformed to a vector of extracted features first. Then the trained detection model is applied to classify the instance as normal event or attack event.

67.4.2 Deep Feature Extraction

Reducing the dimensionality of feature space is important for better computational efficiency and improved performance of learning algorithms [15]. Deep feature extraction has shown to be a promising method for nonlinear dimensionality reduction [16].

The structure of an autoencoder for deep feature extraction is shown in Fig. 67.3 which is a multi-layer neural network consisting of an encoder, a code layer and a decoder. The encoder maps input data into the code layer which is then reconstructed by the decoder as closely as input. After training, the decoder is removed while the encoder and the code layer are retained. Since the number of nodes in the code layer is less than that of the input layer, the output of the code layer is a reduced representation of the input which will be used as the extracted features for outlier detection algorithms.

67.4.3 Semi-Supervised Outlier Detection Algorithms

We considered seven popular semi-supervised outlier detection algorithms in this study which can be categorized as liner models, proximity-based methods, and ensembles [17].

1. Linear models
   - *OCSVM*: SVM is a popular supervised machine learning method for classification. OCSVM was proposed in [18] as an extension of SVM, which is trained only

using instances of the normal class. The algorithm maps training data into a feature space using a kernel function. The mapped vectors are separated from the origin with maximum margin. The separating boundary will then be used to detect a new instance as normal observation or outlier.

2. Proximity-based methods
   - *Histogram-Based Outlier Score* (*HBOS*) is an outlier detector known for its fast computation speed [19]. HBOS works by first generating an univariate histogram for each feature and then normalizing the histograms to have the maximum height of the bins to be one. Finally, the HBOS of an instance x is calculated using Eq. (67.1):

$$HBOS(x) = \sum_{i=1}^{N} \log\left(\frac{1}{hist_i(x)}\right) \qquad (67.1)$$

where N is the number of features and $hist_i(x)$ is the density estimation of the ith feature of instance x.

- *Local Outlier Factor* (*LOF*) is a well-known outlier detector proposed in [20]. The LOF score of an instance x is measured as the degree of the instance isolating from its k nearest neighbors, which is calculated as follows:

$$LOF(x) = \frac{\sum_{o \in N_k(x)} \frac{LRD(o)}{LRD(x)}}{k} \qquad (67.2)$$

where $N_k(x)$ is the set of k nearest neighbors for the instance x, and $LRD(\bullet)$ is the local reachability density which is the inverse of the average distance of an instance from its k nearest neighbors.

- *Clustering-Based Local Outlier Factor* (*CBLOF*): Unlike LOF uses density estimation of nearest neighbors for outlier detection, CBLOF works by using density estimation of clusters [21]. The input data is clustered using a clustering algorithm such as k-Means first. Then the clusters are classified as small and large clusters. The anomaly score for an instance belonging to a large cluster is calculated based on the size of the cluster and the distance between the instance to the cluster center. If the instance belongs to a small cluster, the distance from the instance to the center of the closest large cluster is used.
- *k-Nearest-Neighbor Outlier Detection* (*KNNOD*) was proposed in [22] which uses the distance of an instance to its kth nearest neighbor as the anomaly score. The larger the distance, the more likely an instance to be anomaly. A highly efficient partition-based algorithm was developed in [22] to find the outliers. The anomaly score can also be calculated using the average distance or the median distance to k nearest neighbors [23].

3. Ensemble
   - *Feature Bagging* ensembles multiple base outlier detection algorithms for outlier detection [24]. Each base outlier detection algorithm is trained using randomly sampled subset of features from the original feature set. The outlier scores produced by the base outlier detection algorithms are then combined to generate the anomaly score for an instance. In [24], LOF was used as the base outlier detection algorithm as it was shown a good performance in network intrusion detection.
   - *Isolation Forest* (*iForest*) is an anomaly detection approach proposed by Liu et al. [25]. iForest is an ensemble of isolation trees (iTrees) which are random binary trees constructed by randomly selected data subsets, features and split values. An iTree has two types of nodes: external nodes with no children and internal nodes with two children. The anomaly score of an instance x is defined as the path length $h(x)$ which is the distance between the root node and the external node correspond to the instance in the iTree. The shorter the path length, the more likely an instance to be an anomaly as less partitions needed to isolate the instance from others. As iForst is an ensemble of iTrees, an instance will be highly likely to be anomaly if majority of the iTrees produce short path lengths for it.

67.5 Performance Evaluation and Results

The power system attack datasets are grouped as three groups: binary, three-class and multi-class [8]. The binary group adopted in our study is formed by the normal operations and attack events, which consists of 15 datasets covering the 37 scenarios of Table 67.1. The data was normalized using min-max normalization. We used python and PyOD [26], a toolkit for outlier detection, in our experiments. For all experiments, the contamination ratio, a parameter determining the decision boundary of the detection model, is set to 0.05. The metric used for performance evaluation is F_1 score which is defined as the harmonic mean of precision and recall:

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \qquad (67.3)$$

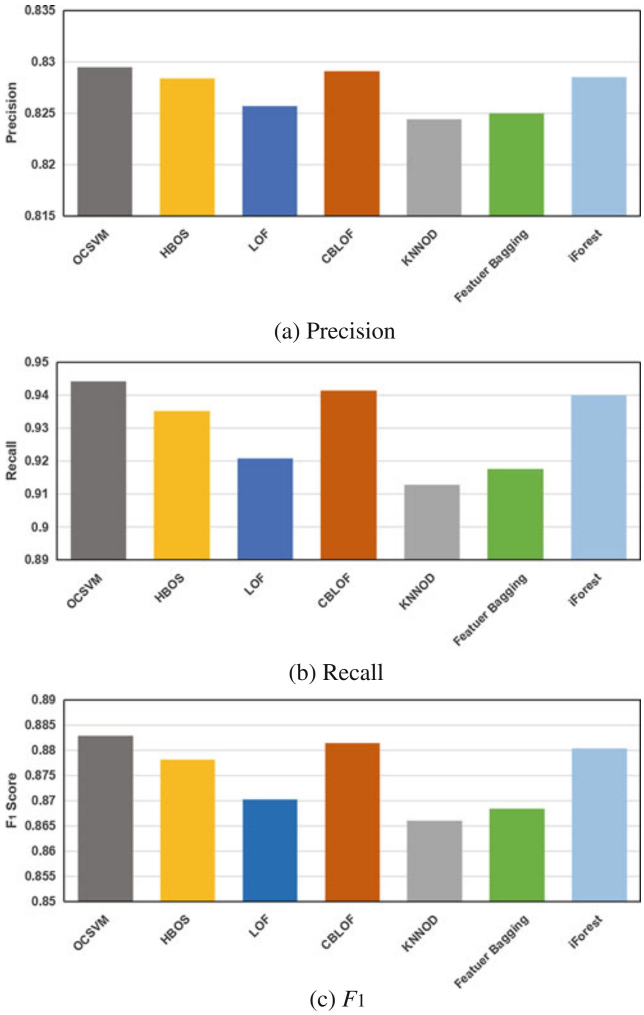
$$\text{Precision} = \frac{TP}{TP + FP} \tag{67.4}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{67.5}$$

where *TP*, *FP* and *FN* are true positives, false positives and false negatives, respectively. We treated attack events as positives and normal operations are negatives in our study. The performance of an algorithm is reported as the averaging of the results obtained from the 15 datasets.

To evaluate the performance of different semi-supervised outlier detection algorithms, we randomly selected 50% of normal instances in a dataset for training the detection model. Other instances of the dataset including normal and attack events were then used for testing. The process was repeated ten times for each dataset. Figure 67.4 shows the performance of the seven semi-supervised outlier detection algorithms using all features. It can be observed that OCSVM achieves the best performance among all algorithms. CBLOF and iForest have comparable performance to OCSVM. These three algorithms achieve significantly better performance than other four algorithms. The results also show that the semi-supervised algorithms obtain better recall than precision. This means that semi-supervised algorithms perform well on finding attack events but result in higher number of FPs.

We then compared the performance of the best three semi-supervised algorithms with supervised algorithms. Two supervised algorithms popular for detecting cyber-attacks in smart grids, SVM and k-NN [8, 10], were considered in our study for comparison. We randomly selected 50% of normal instances in a dataset for training the semi-supervised algorithms. The selected normal instances and the same number of randomly selected attack instances were used to train the supervised algorithms. The testing was done using the remaining normal and attack instances. This process repeated ten times for each of the 15 datasets. Figure 67.5 shows the performance of the algorithms. It can seen that the three semi-supervised algorithms achieve comparable performance. The semi-supervised algorithms perform significantly better than the two supervised algorithms in terms of recall demonstrating that the semi-supervised algorithms can find more attack events than the supervised algorithms. On the other hand, the two supervised algorithms achieve better precision compared with the semi-supervised algorithms due to the high *FP* rates of the semi-supervised algorithms. Overall the semi-supervised algorithms have significantly better performance than the supervised algorithms in terms of F_1 score.



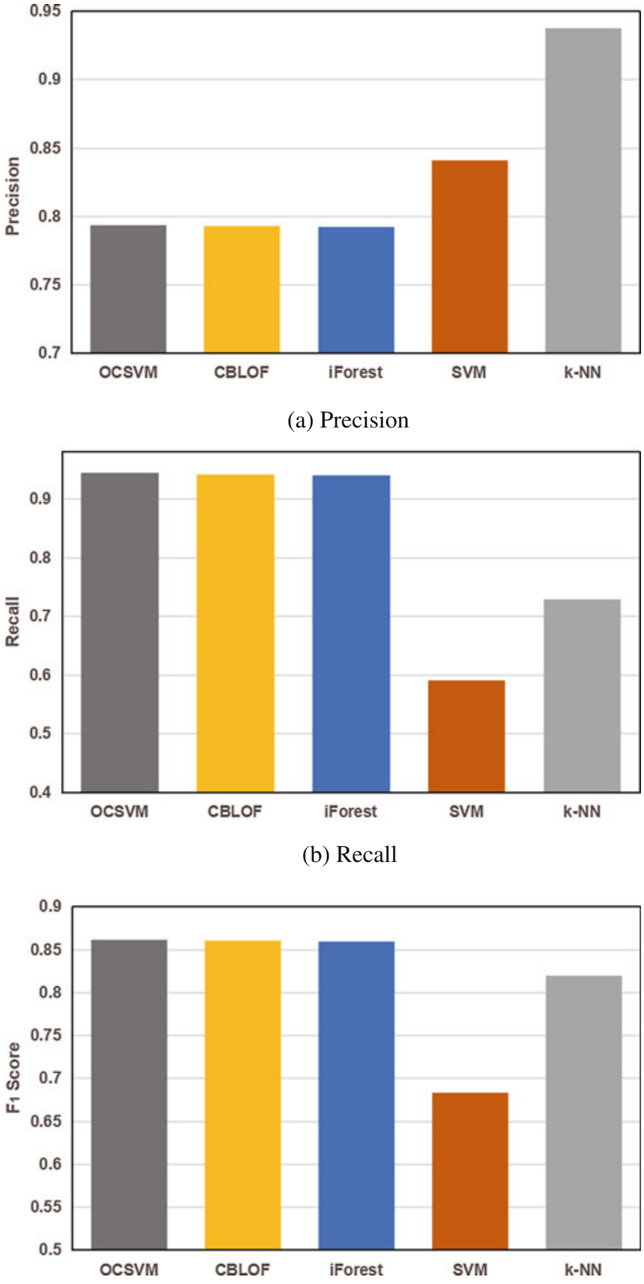
(a) Precision

(b) Recall

(c) F_1

Fig. 67.4 Performance of semi-supervised outlier detection algorithms with all features

Finally, we investigated how deep feature extraction can enhance the performance of the best three semi-supervised algorithms. The popular liner feature extraction method, principle component analysis (PCA), was used for comparison. The extracted number of features was set to 30 for both PCA and autoencoder. The autoencoder has an input layer of 116 nodes corresponding to the number of features from the PMU measurements. The hidden layer of the encoder and the code layer have 60 and 30 nodes, respectively. The results shown in Fig. 67.6 demonstrate that deep feature extraction can significantly improve the performance of all three semi-supervised algorithms in terms of the three metrics. On the other hand, PCA as a linear method works not well. Especially the features extracted by PCA result in lower recall.

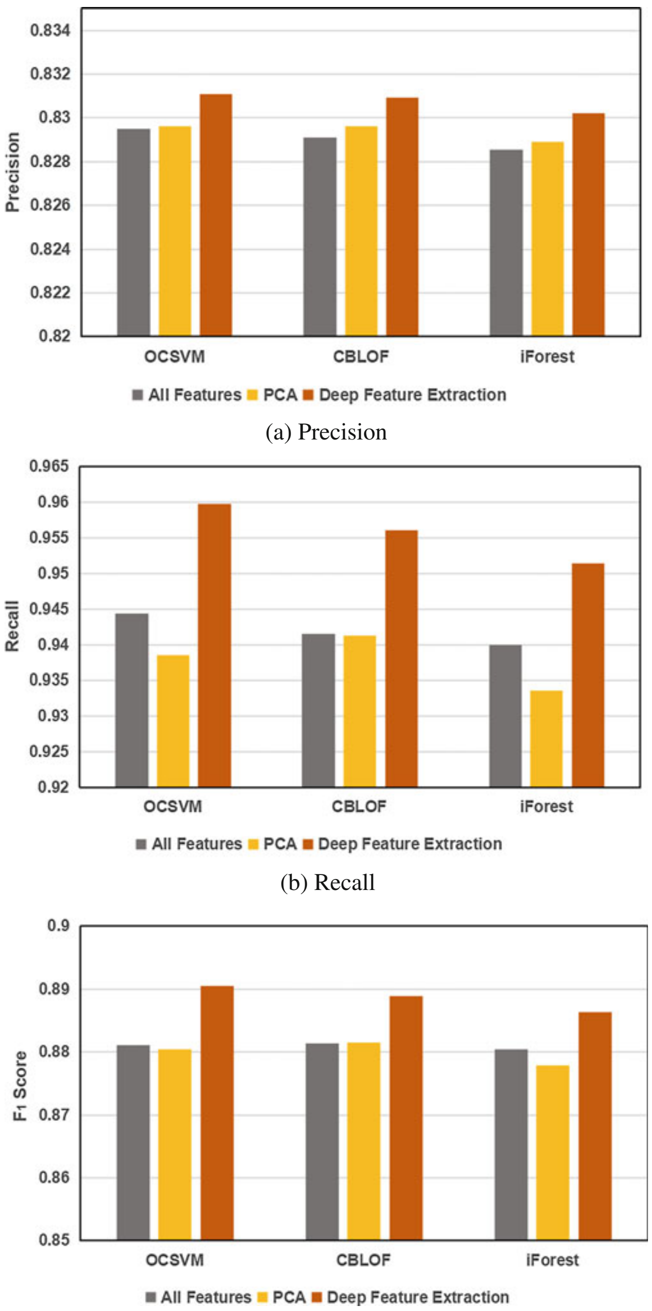


(a) Precision

(b) Recall

(c) F_1

Fig. 67.5 Performance comparison of semi-supervised algorithms with supervised algorithms



(a) Precision

(b) Recall

(c) F_1

Fig. 67.6 Performance comparison of semi-supervised outlier detection algorithms with and without feature extraction

67.6 Conclusion

Cyber-attacks are one of the major challenges faced by smart grids. In this paper, we explored the use of semi-supervised outlier detection algorithms augmented by deep feature extraction for detecting cyber-attacks in smart grids using the data collected from PMUs. Our results show that semi-supervised algorithms can achieve better detection performance than popular supervised algorithms. Nonlinear dimensionality reduction methods like deep feature extraction are better choices than liner ones like PCA for enhancing the performance of semi-supervised algorithms for detecting cyber-attacks in smart grids. In future, advanced semi-supervised learning algorithms such as deep anomaly detection [27] will be studied for better detection performance.

Acknowledgements This research was supported by the National Science Foundation under grant no. CNS-1757945 and EPSCoR Cooperative Agreement OIA-1757207.

References

1. Pignati, M., Popovic, M., Barreto, S., Cherkaoui, R., Flores, G.D., Le Boudec, J.Y., Mohiuddin, M., Paolone, M., Ramano, P., Sarri, S., Tesfay, T., Tomozei, D.C., Zanni, L.: Real-time state estimation of the EPFL campus medium-voltage grid by using PMUs. In: 2015 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT), pp. 1–5 (2015)
2. Ashok, A., Govindarasu, M., Wang, J.: Cyber-physical attack-resilient wide-area monitoring, protection, and control for the power grid. Proc. IEEE. **105**(7), 1389–1407 (2017)
3. Blair, S.M., Burt, G.M., Gordon, N., Orr, P.: Wide area protection and fault location: review and evaluation of PMU-based methods. In: 14th International Conference on Developments in Power System Protection (2018)
4. Wang, X., Bialek, J.W., Turitsyn, K.: PMU-based estimation of dynamic state Jacobian matrix and dynamic system state matrix in ambient conditions. IEEE Trans. Power Syst. **33**(1), 681–690 (2018)
5. Anu, J., Agrawal, R., Seay, C., Bhattacharya, S.: Smart grid security risks. In: 12th International Conference on Information Technology - New Generation (ITNG 2015), pp. 485–489 (2015)
6. Paudel, S., Smith, P., Zseby, T.: Stealthy attacks on smart grid PMU state estimation. In: 13th International Conference on Availability, Reliability and Security (2018)
7. Lee, R., Asante, M., Conway, T.: Analysis of the cyber attack on the Ukrainian power grid. SANS ICS Report (2016)
8. Hink, R., Beaver, J., Buckner, M., Morris, T., Adhikari, U., Pan, S.: Machine learning for power system disturbance and cyber-attack discrimination. In: 7th International Symposium on Resilient Control Systems (ISRCS), pp. 1–8 (2014)
9. Pan, S., Morris, T., Adhikari, U.: Classification of disturbances and cyber-attacks in power systems using heterogeneous time-synchronized data. IEEE Trans. Ind. Inform. **11**(3), 650–662 (2015)
10. Ozay, M., Esnaola, I., Vural, Y., Kulkarni, S., Poor, H.: Machine learning methods for attack detection in the smart grid. IEEE Trans. Neural Netw. Learn. Syst. **27**(8), 1773–1786 (2016)
11. Wu, T., Zhang, Y., Tang, X.: Isolation forest based method for low-quality synchrophasor measurements and early events detection. In: 2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids. IEEE, Piscataway (2018)
12. Wang, D., Wang, X., Zhang, Y., Jin, L.: Detection of power grid disturbances and cyber-attacks based on machine learning. J. Inf. Secur. Appl. **46**, 42–52 (2019)
13. Maglaras, L.A., Jiang, J.: Intrusion detection in SCADA systems using machine learning techniques. In: 2014 IEEE Science and Information Conference, pp. 626–631 (2014)
14. Maglaras, L.A., Jiang, J.: OCSVM model combined with K-means recursive clustering for intrusion detection in SCADA systems. In: 10th International Conference on Heterogeneous Networking for Quality Reliability Security and Robustness (QShine), pp. 133–134 (2014)
15. Song, F., Guo, Z., Mei, D.: Feature selection using principal component analysis. In: 2010 International Conference on System Science, Engineering Design and Manufacturing Informatization, pp. 27–30. IEEE, Piscataway (2010)
16. Chakraborty, D., Narayanan, V., Ghosh, A.: Integration of deep feature extraction and ensemble learning for outlier detection. Pattern Recogn. **89**, 161–171 (2019)
17. Aggarwal, C.C.: Outlier analysis. Springer, Cham (2017)
18. Scholkopf, B., Williamson, R., Smola, A., Shawe-Taylor, J., Platt, J.: Support vector method for novelty detection. In: NIPS'99, pp. 582–588 (1999)
19. Goldstein, M., Dengel, A.: Histogram-based Outlier Score (HBOS): a fast unsupervised anomaly detection algorithm. In: KI-2012, pp. 59–63 (2012)
20. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: LOF: identifying density-based local outliers. ACM SIGMOD Rec. **29**(2), 93–104 (2000)
21. He, Z., Xu, X., Deng, S.: Discovering cluster-based local outliers. Pattern Recogn. Lett. **24**(9–10), 1641–1650 (2003)
22. Ramaswamy, S., Rastogi, R., Shim, K.: Efficient algorithms for mining outliers from large data sets. ACM SIGMOD Rec. **29**(2), 427–438 (2000)
23. Angiulli, F., Pizzuti, C.: Fast outlier detection in high dimensional spaces. In: European Conference on Principles of Data Mining and Knowledge Discovery, pp. 15–27. Springer, Berlin (2002)
24. Lazarevic, A., Kumar, V.: Feature bagging for outlier detection. In: 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, pp. 157–166. ACM, New York (2005)
25. Liu, F., Ting, K., Zhou, Z.H.: Isolation based anomaly detection. ACM Trans. Knowl. Discov. Data. **6**, 1–44 (2012)
26. Zhao, Y., Nasrullah, Z., Li, Z.: PyOD: a python toolbox for scalable outlier detection. J. Mach. Learn. Res. **20**(96), 1–7 (2019)
27. Pang, G., Shen, C., van den Hengel, A.: Deep anomaly detection with deviation networks. In: 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 353–362 (2019)