Enhancing the Performance of Semi-supervised Electricity Theft Detection in Smart Grids with Feature Engineering and Ensemble Learning

Ruobin Qi
Department of Computer Science and Engineering
New Mexico Institute of Mining and Technology
Socorro, NM
ruobin.qi@student.nmt.edu

Wynter Japp
School of Computing and Informatics
University of Louisiana Lafayett
Lafayette, LA
wynter.japp1@louisiana.edu

Stephen Pan
Department of Computer Science
University of Arizona
Tucson, AZ
legosp7@arizona.edu

Jun Zheng
Department of Computer Science and Engineering
New Mexico Institute of Mining and Technology
Socorro, NM
jun.zheng@nmt.edu

Sihua Shao
Department of Electrical Engineering
New Mexico Institute of Mining and Technology
Socorro, NM
sihua.shao@nmt.edu

***Abstract*—Electricity theft is a type of cyberattack posing significant risks to the security of smart grids. Semi-supervised outlier detection (SSOD) algorithms utilize normal power usage data to build detection models, enabling them to detect unknown electricity theft attacks. In this paper, we applied feature engineering and ensemble learning to improve the detection performance of SSOD algorithms. Specifically, we extracted 22 time-series and wavelet features from load profiles, which served as inputs for the seven popular SSOD algorithms investigated in this study. Experimental results demonstrate that the proposed feature engineering greatly enhances the performance of SSOD algorithms to detect various false data injection (FDI) attacks. Furthermore, we constructed bagged ensemble models using the best-performing SSOD algorithm as the base model, with results indicating further improvements in detection performance compared to the base model alone.**

***Index Terms*—electricity theft detection (ETD), semi-supervised outlier detection (SSOD), feature engineering, ensemble learning, false data injection (FDI) attack, smart grids**

I. Introduction

Advanced metering infrastructure (AMI) enhances the efficiency and resilience of energy delivery and management in smart grids by enabling a dynamic, bidirectional flow of energy and information [1]. In the AMI network, high-frequency energy consumption data are collected from consumers by smart meters deployed by utility companies, which are used to facilitate data-driven services for energy delivery and management [1].

However, AMI also introduces a range of security vulnerabilities in smart grids [2], one of which is the threat of electricity theft [3], [4]. Rather than physically tampering with or bypassing traditional mechanical meters to lower their electricity bills, electricity thieves can alter smart meter data through cyberattacks. Substantial financial losses can be caused by electricity theft for utility companies in various countries. For instance, a 2017 study by the Northeast Group found that utilities worldwide suffer annual losses of $96 billion due to non-technical losses (NTLs) including electricity theft [5]. Therefore, advanced electricity theft detection (ETD) methods are necessary to safeguard the security and long-term sustainability of smart grids.

Physical inspections and video surveillance are traditional methods for detecting electricity theft, but they are time-consuming and costly [6]. The vast amount of high-frequency energy consumption data collected by AMI has facilitated the research of new data-driven ETD methods, primarily utilizing machine learning techniques.

Supervised learning is the most popular machine learning technique for detecting electricity theft. Traditional machine learning algorithms have been utilized to build detection models, such as decision trees [7], support vector machines (SVM) [7], and random forests [8]. Recently, the exceptional performance of deep learning across various domains has made it a leading approach for ETD. For instance, convolutional neural networks (CNNs) are a popular choice due to their effectiveness in this area [9].

Supervised ETD methods require samples of fraudulent activities of electricity thieves to build effective detection models, which can be very hard, if not impossible, to obtain in real-world applications. Unsupervised learning is another popular machine learning technique for ETD, which solely relies on unlabeled data to construct models for detecting potential fraudulent users. Clustering and correlation analysis are the two main techniques adopted by unsupervised learning-

979-8-3503-7240-3/24/$31.00 ©2024 IEEE

based methods [6], [10]. Unsupervised ETD methods come with certain limitations, including reduced effectiveness in detecting specific attack types and longer detection times.

Semi-supervised outlier detection (SSOD), on the other hand, can address the issues of supervised and unsupervised ETD methods. Since only normal usage data is employed to train detection models, there is no need to collect fraudulent usage data, and the trained models have the capability of detecting unknown attack types. Several studies in the literature have explored the use of SSOD for ETD. For instance, the study of [11] evaluated one-class SVM (OCSVM) for this purpose. Additionally, the study of [12] conducted feature engineering by extracting 20 time-series features from load profiles. They considered eight types of false data injection (FDI) attacks in their study. Their experimental results demonstrate that using these extracted time-series features significantly improves the detection performance of SSOD algorithms compared to using the original load profiles.

This paper introduces a new feature engineering technique that extracts features from both time and wavelet domains, aiming to address a broader range of FDI attacks than those considered in [12]. We explored how this technique can improve the detection performance of various SSOD algorithms. Additionally, we further improved the performance of the best-performing algorithm through ensemble learning by constructing bagged ensemble models with random subspace sampling, utilizing the features extracted through our feature engineering.

The rest of this paper is organized as follows: Section II describes the FDI attacks considered in our study, designed to simulate the fraudulent activities of electricity thieves. Section III presents the SSOD algorithms explored, our proposed feature engineering, and the bagged ensemble model with subspace sampling used to enhance detection performance. Section IV provides the results of the performance evaluation experiments. Finally, Section V concludes the paper.

II. FDI Attacks

As far as we know, there is currently no real-world high-frequency smart meter dataset that includes electricity theft data. As a result, research in this field typically employs various FDI attacks to simulate the fraudulent activities of electricity thieves. Tampered load profiles are generated by modifying normal load profiles with these FDI attacks to evaluate the detection performance of proposed ETD methods. We considered eleven FDI attacks in this study that have been adopted in other studies [4], [6], [10], [11], which are shown in Table I. These attacks fall into two categories: reduced consumption attacks (types 1 to 8) and load profile shifting attacks (types 9 to 11) [4]. To lower electricity bills, reduced consumption attacks employ various strategies to directly reduce smart meter readings. In contrast, load profile shifting attacks do not change the total daily power consumption but rather modify the timing of peaks and valleys in daily load profiles to evade higher electricity prices during specific time intervals set by utility companies. In Table I, $\mathbf{x}$ denotes the normal daily load profile, x_t denotes the reading of the normal load profile at time instance t, and $\tilde{x}_t$ denotes the altered reading of the tampered load profile at time instance t. N is the total number of readings in a daily load profile. Figure 1 illustrates the effects of different FDI attacks on altering a user's normal load profile.

TABLE I: FDI attacks

Attack Type	Modification
1	$\tilde{x}_t = \alpha x_t, \quad 0.2 < \alpha < 0.8$
2	$\tilde{x}_t = f(t) \times x_t,$ $f(t) = \begin{cases} 0 & t_1 < t < t_2 \\ 1 & \text{otherwise} \end{cases}$
3	$\tilde{x}_t \leftarrow \alpha_t x_t, \quad 0.2 < \alpha_t < 0.8$
4	$\tilde{x}_t = f(t) \times x_t,$ $f(t) = \begin{cases} \alpha & 0.2 < \alpha < 0.8,\ t_1 < t < t_2 \\ 1 & \text{otherwise} \end{cases}$
5	$\tilde{x}_t \leftarrow \alpha_t \overline{\mathbf{x}}, \quad 0.2 < \alpha_t < 0.8$
6	$\tilde{x}_t = \begin{cases} x_t & x_t \leq \gamma \\ \gamma & x_t > \gamma \end{cases} \quad \gamma < max(\mathbf{x})$
7	$\tilde{x}_t = max\{x_t - \gamma, 0\}, \quad \gamma < max(\mathbf{x})$
8	$\tilde{x}_t = (1 - f(t)) \times x_t,$ $f(t) = \begin{cases} \alpha_{max} & t \geq t_{max} \\ \beta(t - t_s) & t_s < t < t_{max} \\ 0 & t < t_s \end{cases}$
9	$\tilde{x}_t = \overline{x}$
10	$\tilde{x}_t = x_{N-t}$
11	$\tilde{x_t} = \begin{cases} x_t - \lambda x_t & t_1 < t < t_2 \\ x_t + \frac{\sigma}{N-n} & \text{otherwise} \end{cases}$

III. Methods

A. SSOD Algorithms Investigated in This Study

The performance of the following seven popular SSOD algorithms for ETD was investigated in our study.

- *Principal component analysis (PCA)*: In addition to being a popular technique for data dimensionality reduction, PCA can also be applied for outlier detection [13]. When used for outlier detection, the model is built using the major and minor principal components derived from normal samples. The outlier score of a test sample is determined by its distance in the principal component space.
- *Angle-based outlier detection (ABOD)*: ABOD draws inspiration from the notion that angles exhibit greater stability for outlier detection compared to distances [14]. When a sample is located within a cluster, the angle between it and pairs of other samples in the cluster shows significant variability. In contrast, if the sample is an outlier, the angles generally exhibit minimal variation. During the training stage, a threshold based on the variance of angles is established using normal samples, which is then utilized in the testing stage to detect outliers.
- *k-nearest neighbor (KNN)*: KNN is a simple outlier detection algorithm, which calculates the outlier score of a sample by computing the Euclidean distance from the sample to its k-th nearest neighbor [15].

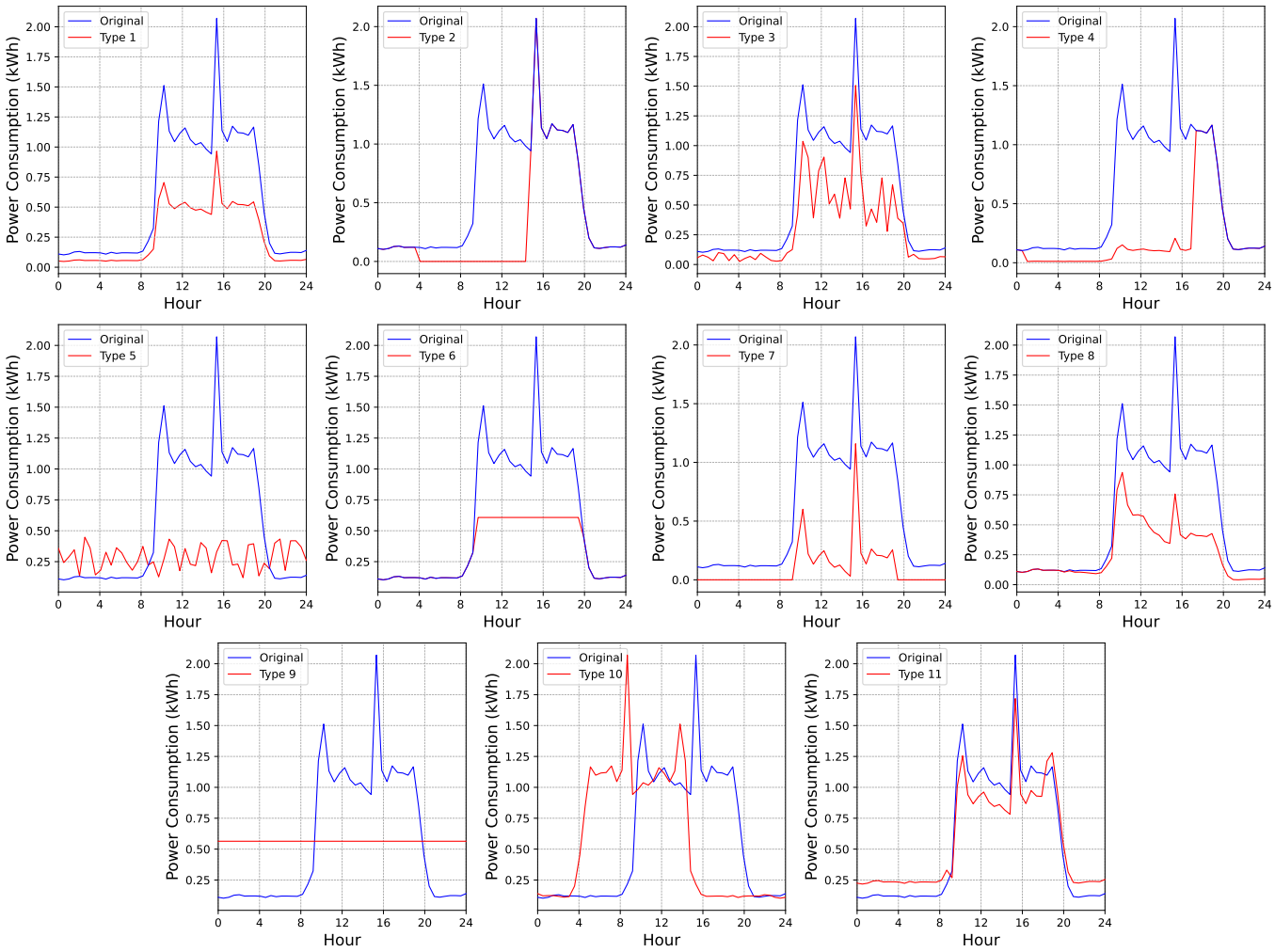

Fig. 1: An illustration of the effects of eleven FDI attacks on altering a user's normal load profile

- *Gaussian mixture model (GMM)*: GMM employs a mixture of Gaussian distributions to estimate the probability distribution of normal data [16]. The outlier score of a test sample is computed according to its probability obtained from the estimated distribution. A lower probability indicates the sample is more likely an outlier.
- *OCSVM*: OCSVM is a variation of the traditional SVM algorithm designed specifically for outlier detection [17]. In OCSVM, a kernel function is utilized to map the input data into a high-dimensional feature space. An iterative process then identifies a hyperplane that maximizes the margin between normal samples and the origin. This hyperplane is used to distinguish between normal and abnormal samples [18].
- *Local outlier factor (LOF)*: LOF compares a sample's local density to that of its neighbors for detecting outliers [19]. If a sample's local density is significantly lower than that of its neighbors, it is considered an outlier.
- *Histogram-based outlier score (HBOS)*: HBOS performs effective outlier detection by constructing histograms of input features first [20]. An input feature's histogram is utilized to compute its outlier score. The HBOS outlier score is then obtained by summing all input features' outlier scores.

B. Feature Engineering

We conducted feature engineering to extract 22 features from the time and wavelet domains to enhance the effectiveness of SSOD algorithms for ETD. The discrete wavelet transform (DWT) was performed using the db1 wavelet and four levels of decomposition. These 22 extracted features serve as the input for the SSOD algorithms, aimed at aiding in the detection of malicious changes in energy consumption amounts and/or load profile shapes. For example, the three level-4 DWT approximation coefficients not only capture the approximate shape of the load profile but also represent the user's energy consumption during different periods of the day.

- `mean`: the average value of the load profile
- `standard_deviation`: the standard deviation of the load profile

- skewness: the skewness of the load profile
- maximum: the maximum value of the load profile
- minimum: the minimum value of the load profile
- sum_values: the sum of all values of the load profile
- mean_abs_change: the averaging of the absolute differences between subsequent values of the load profile
- benford_correlation: the correlation between the first digit distribution of the load profile and the Benford's Law distribution
- last_location_of_maximum: the last location of the maximum value of the load profile
- last_location_of_minimum: the last location of the minimum value of the load profile
- first_location_of_maximum: the first location of the maximum value of the load profile
- first_location_of_minimum: the first location of the minimum value of the load profile
- percentage_of_recurring_values_to_all_values: the percentage of recurring values of the load profile
- sum_of_recurring_data_points: the sum of all recurring values in the load profile
- has_duplicate_max: a Boolean value indicating whether the maximum value of the load profile occurs more than once or not
- has_duplicate: a Boolean value indicating if any value of the load profile occurs more than once or not
- count_above_mean: the count of values of the load profile higher than the mean
- count_below_mean: the count of values of the load profile lower than the mean
- number_peaks: the count of peaks in the load profile
- DWT coefficients: three level-4 approximation coefficients of DWT

C. Bagged Ensemble with Random Subspace Sampling

To further improve the detection performance, we constructed bagged ensemble models with random subspace sampling using the best-performing SSOD algorithm as the base model. The random subspace sampling method constructs an individual detection model with a randomly sampled feature subset. To apply an ensemble model for detection, the outputs of individual models are combined with a pre-defined rule. We considered two rules in our study: MAX and AVG. The MAX rule uses the maximum posterior probability of all individual models as the output of the ensemble model, while the AVG rule uses the average posterior probability as the final output.

IV. Experiments and Results

We adopted the popular Irish CER Smart Metering Project Dataset [21] for performance evaluation. The sampling rate for the dataset is 30 minutes per sample, resulting in a daily load profile containing 48 data points. Our experiments utilized data from 100 randomly selected small and medium-sized enterprises (SMEs) over 180 days, spanning from July 15, 2009, to January 11, 2010. Each SME user's data was divided into training and testing sets with a ratio of 7:3. Subsequently, for the load profiles in the testing set, we introduced tampering in half of them using a selected FDI attack type. The proposed method was compared against two reference methods: one directly utilizing the load profile as input for SSOD algorithms, denoted as RF1, and another employing the 20 time-series features extracted with the feature engineering approach proposed in [12] as input, denoted as RF2. All methods were implemented by Python and the PyOD library [22]. The load profiles and the extracted features were normalized using min-max normalization before being inputted into SSOD algorithms. The performance metric employed in our experiments is the Area Under the Curve (AUC). The reported results represent the averages across the 100 SME users, and the result of the best-performing algorithm for each attack type is marked in bold.

The performance evaluation results of the two reference methods and the proposed method in terms of AUC are reported in Tables II to IV. The results indicate that both RF2 and the proposed feature engineering approach greatly enhance the detection performance of SSOD algorithms compared to RF1, highlighting the effectiveness of feature engineering for ETD. However, RF2, not designed for all 11 attack types, exhibits lower performance for attack type 10, as evidenced by its much lower AUC score compared to RF1. In contrast, the proposed method demonstrates comparable performance to RF1 for attack type 10 while achieving significantly better performance for other attack types. Specifically, the best-performing SSOD algorithm in the proposed method is ABOD, which consistently achieves top performance for every attack type with an average AUC score of 0.8969.

To further enhance the detection performance, we trained two bagged ensemble models using the best-performing SSOD algorithm, ABOD, as the base model. We denote the two ensemble models using the AVG and MAX combination rules as $\mathbf{EM}_{AVG}$ and $\mathbf{EM}_{MAX}$, respectively. Each ensemble model comprises 50 individual models trained with randomly selected feature subsets. Table V compares the performance of ABOD with that of the two ensemble models. The results demonstrate that both ensemble models outperform the base model ABOD, with $\mathbf{EM}_{AVG}$ achieving the highest average AUC score. Notably, the AUC scores of $\mathbf{EM}_{AVG}$ are more than 3% higher than those of ABOD for attack types 1 and 4, highlighting the effectiveness of ensemble learning in improving detection performance.

V. Conclusions

This paper aims to enhance the performance of SSOD algorithms in addressing the ETD problem. We achieved this by performing feature engineering to extract features in both time and wavelet domains from users' load profiles. We conducted a performance evaluation using the popular Irish CER smart meter dataset and applied various FDI attacks to it. Our results demonstrate that the proposed feature engineering significantly improves the detection performance of SSOD algorithms across a broad range of FDI attacks. Additionally,

TABLE II: Performance of SSOD algorithms for RF1

Attack Type	PCA	ABOD	KNN	GMM	OCSVM	LOF	HBOS
1	0.7267	0.6828	0.6407	0.5278	0.6801	**0.8097**	0.751
2	0.8836	0.8480	0.7687	0.7686	0.9072	**0.9286**	0.8062
3	0.8087	0.8487	0.7754	0.7993	0.7920	**0.9192**	0.8021
4	0.6647	0.6850	0.6321	0.6424	0.6391	**0.7586**	0.6767
5	0.8684	0.8783	0.7958	0.8243	0.8613	**0.9337**	0.8497
6	0.5369	0.5324	0.5281	0.4367	0.5072	**0.6583**	0.6143
7	0.8588	0.8134	0.7501	0.6068	0.8552	**0.8954**	0.8382
8	0.6720	0.6599	0.6123	0.6050	0.6501	**0.7177**	0.6727
9	0.7490	0.6661	0.7066	0.5110	0.7206	0.7538	**0.7921**
10	0.8291	0.8175	0.7902	0.8041	0.8239	**0.8541**	0.7330
11	0.7570	0.7628	0.7324	0.6888	0.7468	**0.7937**	0.7880
Average AUC	0.7595	0.7450	0.7029	0.6559	0.7440	**0.8203**	0.7567

TABLE III: Performance of SSOD algorithms for RF2

Attack Type	PCA	ABOD	KNN	GMM	OCSVM	LOF	HBOS
1	0.7409	**0.8536**	0.8058	0.7599	0.7105	0.8301	0.7800
2	0.9514	**0.9845**	0.9672	0.9000	0.9482	0.9710	0.9433
3	0.9405	**0.9543**	0.9503	0.8687	0.9435	0.9446	0.9100
4	0.6898	**0.7986**	0.7478	0.7312	0.6673	0.7817	0.7284
5	0.9580	0.9713	**0.9746**	0.9178	0.9594	0.9628	0.9404
6	0.9341	**0.9682**	0.9539	0.9663	0.9292	0.9572	0.8723
7	0.8985	**0.9345**	0.9135	0.8436	0.8922	0.9105	0.8825
8	0.6978	**0.7547**	0.7318	0.7105	0.6800	0.7427	0.7137
9	0.9982	**0.9997**	0.9995	0.9962	0.9988	0.9983	0.9799
10	0.5588	**0.6146**	0.5937	0.5656	0.5595	0.6036	0.5739
11	0.6995	**0.7932**	0.7423	0.7304	0.6840	0.7687	0.7327
Average AUC	0.8243	**0.8752**	0.8528	0.8173	0.8157	0.8610	0.8234

TABLE IV: Performance of SSOD algorithms for the proposed method

Attack Type	PCA	ABOD	KNN	GMM	OCSVM	LOF	HBOS
1	0.7648	**0.8536**	0.8067	0.7216	0.7255	0.8304	0.8136
2	0.9502	**0.9799**	0.9608	0.9114	0.9407	0.9594	0.9367
3	0.9157	**0.9723**	0.9373	0.8471	0.9196	0.9578	0.9039
4	0.6836	**0.7689**	0.7200	0.7105	0.6481	0.7460	0.7358
5	0.9479	**0.9763**	0.9681	0.8922	0.9481	0.9727	0.9319
6	0.9333	**0.9624**	0.9447	0.9555	0.9238	0.9501	0.8689
7	0.9150	**0.9374**	0.9189	0.8368	0.9139	0.9227	0.9002
8	0.7157	**0.7884**	0.7512	0.7169	0.7057	0.7679	0.7153
9	0.9978	**0.9996**	0.9992	0.9924	0.9987	0.9991	0.9687
10	0.7398	**0.8073**	0.7822	0.7718	0.7181	0.7827	0.7091
11	0.7570	**0.8198**	0.7893	0.7801	0.7473	0.7875	0.7685
Average AUC	0.8473	**0.8969**	0.8708	0.8306	0.8354	0.8797	0.8411

TABLE V: Performance comparison of ABOD with bagged ensemble models

Attack Type	ABOD	EM_{AVG}	EM_{MAX}
1	0.8536	**0.8850**	0.8613
2	0.9799	0.9786	**0.9805**
3	0.9723	**0.9773**	0.9730
4	0.7689	**0.7990**	0.7734
5	0.9763	**0.9830**	0.9751
6	**0.9624**	0.9421	**0.9624**
7	0.9374	**0.9391**	0.9387
8	0.7884	**0.7943**	0.7915
9	**0.9996**	0.9990	0.9995
10	0.8073	0.7961	**0.8118**
11	0.8198	**0.8322**	0.8213
Average AUC	0.8969	**0.9023**	0.8990

we developed bagged ensemble models based on the best-performing SSOD algorithm. Our findings indicate that these ensemble models offer superior detection performance compared to the base model.

ACKNOWLEDGMENT

This material is based upon work supported by the National Science Foundation EPSCoR Cooperative Agreement OIA-1757207 and Grant CNS-2150145.

REFERENCES

[1] R. R. Mohassel, A. Fung, F. Mohammadi, and K. Raahemifar, "A survey on advanced metering infrastructure," *International Journal of Electrical Power & Energy Systems*, vol. 63, pp. 473–484, 2014.

[2] R. Qi, C. Rasband, J. Zheng, and R. Longoria, "Detecting cyber attacks in smart grids using semi-supervised anomaly detection and deep representation learning," *Information*, vol. 12, no. 8, p. 328, 2021.

[3] Z. Yan and H. Wen, "Performance analysis of electricity theft detection for the smart grid: An overview," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–28, 2021.

[4] X. Xia, Y. Xiao, W. Liang, and J. Cui, "Detection methods in smart meters for electricity thefts: A survey," *Proceedings of the IEEE*, vol. 110, no. 2, pp. 273–319, 2022.

[5] A. Theron-Ord, "Electricity theft and non-technical losses total $96bn annually – report," Smart Energy International, May 12, 2017. Available: https://www.smart-energy.com/regional-news/africa-middle-east/electricity-theft-96bn-annually/.

[6] R. Qi, J. Zheng, Z. Luo, and Q. Li, "A novel unsupervised data-driven method for electricity theft detection in ami using observer meters," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–10, 2022.

[7] A. Jindal, A. Dua, K. Kaur, M. Singh, N. Kumar, and S. Mishra, "Decision tree and svm-based data analytics for theft detection in smart grid," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 3, pp. 1005–1016, 2016.

[8] S. K. Gunturi and D. Sarkar, "Ensemble machine learning models for the detection of energy theft," *Electric Power Systems Research*, vol. 192, p. 106904, 2021.

[9] Z. Zheng, Y. Yang, X. Niu, H.-N. Dai, and Y. Zhou, "Wide and deep convolutional neural networks for electricity-theft detection to secure smart grids," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1606–1615, 2017.

[10] K. Zheng, Q. Chen, Y. Wang, C. Kang, and Q. Xia, "A novel combined data-driven approach for electricity theft detection," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 3, pp. 1809–1819, 2018.

[11] P. Jokar, N. Arianpoo, and V. C. Leung, "Electricity theft detection in ami using customers' consumption patterns," *IEEE Transactions on Smart Grid*, vol. 7, no. 1, pp. 216–226, 2015.

[12] E. Orozco, R. Qi, and J. Zheng, "Feature engineering for semi-supervised electricity theft detection in AMI," in *2023 IEEE Green Technologies Conference (GreenTech)*. IEEE, 2023, pp. 128–133.

[13] M.-L. Shyu, S.-C. Chen, K. Sarinnapakorn, and L. Chang, "A novel anomaly detection scheme based on principal component classifier," in *Proceedings of the IEEE foundations and new directions of data mining workshop*. IEEE Press, 2003, pp. 172–179.

[14] H.-P. Kriegel, M. Schubert, and A. Zimek, "Angle-based outlier detection in high-dimensional data," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008, pp. 444–452.

[15] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 2000, pp. 427–438.

[16] C. C. Aggarwal and C. C. Aggarwal, *An introduction to outlier analysis*. Springer, 2017.

[17] B. Schölkopf, R. C. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt, "Support vector method for novelty detection," *Advances in neural information processing systems*, vol. 12, 1999.

[18] Y. Wang, J. Wong, and A. Miner, "Anomaly intrusion detection using one class SVM," in *Proceedings of the Fifth Annual IEEE SMC Information Assurance Workshop*, 2004, pp. 358–364.

[19] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: identifying density-based local outliers," in *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, 2000, pp. 93–104.

[20] M. Goldstein and A. Dengel, "Histogram-based outlier score (HBOS): A fast unsupervised anomaly detection algorithm," *KI-2012: poster and demo track*, vol. 9, 2012.

[21] Commission for Energy Regulation, "CER Smart Metering Project—Electricity customer behaviour trial, 2009–2010," Irish Soc. Sci. Data Arch., Dublin, Ireland, SN: 0012-00, 2012. Available: https://www.ucd.ie/issda/data/commissionforenergyregulationcer/.

[22] Y. Zhao, Z. Nasrullah, and Z. Li, "PyOD: A Python toolbox for scalable outlier detection," *Journal of Machine Learning Research*, vol. 20, no. 96, pp. 1–7, 2019.