Evaluation of Online Machine Learning Algorithms for Electricity Theft Detection in Smart Grids

Ashraf Alkhresheh
Computer Science Department
Tafila Technical University
Tafila, Jordan

Mutaz A. B. Al-Tarawneh
Computer Engineering Department
Mutah University
Karak, Jordan

Mohammad Alnawayseh
MIS Department
The University of Jordan
Amman, Jordan

***Abstract*—Electricity theft-induced power loss is a pressing issue in both traditional and smart grid environments. In smart grids, smart meters can be used to track power consumption behaviour and detect any suspicious activity. However, smart meter readings can be compromised by deploying intrusion tactics or launching cyber attacks. In this regard, machine learning models can be used to assess the daily consumption patterns of customers and detect potential electricity theft incidents. Whilst existing research efforts have extensively focused on batch learning algorithms, this paper investigates the use of online machine learning algorithms for electricity theft detection in smart grid environments, based on a recently proposed dataset. Several algorithms including Naive Bayes, K-nearest Neighbours, K-nearest Neighbours with self-adjusting memory, Hoeffding Tree, Extremely Fast Decision Tree, Adaptive Random Forest and Leveraging Bagging are considered. These algorithms are evaluated using an online machine learning platform considering both binary and multi-class theft detection scenarios. Evaluation metrics include prediction accuracy, precision, recall, F-1 score and kappa statistic. Evaluation results demonstrate the ability of the Leveraging Bagging algorithm with an Adaptive Random Forest base classifier to surpass all other algorithms in terms of all the considered metrics, for both binary and multi-class theft detection. Hence, it can be considered as a viable option for electricity theft detection in smart grid environments.**

Keywords*—Smart grid; power loss; electricity theft; online machine learning

I. Introduction

Utilizing energy resources effectively and efficiently is a crucial part of every nation's social and economic growth due to the high cost of energy acquisition and the scarcity of energy resources [1]. Future energy monitoring may now be used to its fullest potential thanks to the smart grid. The smart grid system can be characterized as a whole electrical network made up of the power system infrastructure, computers to control and monitor energy usage, and a sophisticated monitoring system that keeps track of the behavior and usage patterns of all system users [2]. Today, one of the most obvious problems affecting both traditional power grids and smart grids is electric power loss. Countries experience different levels of electricity losses. For instance, 6%, 10%, 16%, and 18% of each country's total energy production was lost in the USA, Russia, Brazil, and India, correspondingly [1]. In the transmission and distribution of electricity, there are two different categories of losses: technical and nontechnical. Energy losses in the machinery required for power transmission and distribution are referred to as technical losses. Power theft, fraud on the part of utility employees, and irregular billing practices all contribute to non-technical losses (NTL) [3]. The NTL is estimated to cost utilities around the world US$96 billion annually [4]. Power providers, engineers, and academics are working to reduce NTL in a number of creative and effective ways due to the significant economic loss [5]. One of the most effective strategies to prevent energy theft is the use of smart meter-based Energy Internet (EI) [6]. Such a technique may be used to remotely track consumption data from customers, record any suspicious activity, and quickly send the data to the utility. Despite their many benefits, smart meters are impractical for countries experiencing severe economic difficulties due to the significant costs associated with their deployment and maintenance. Before these tools are extensively deployed, it is also necessary to adequately manage the expanding cyber dangers. It is difficult to secure the information flow of the EI because of the unique characteristics of advanced metering infrastructure (AMI). By deploying intrusion tactics, the unauthorized users can alter data from smart meters. Because of this, power thefts on the EI are distinct from those that occurred on the traditional grid and were primarily the result of physically avoiding or extinguishing the mechanical [3]. The energy usage patterns of consumers may be automatically tracked by machine learning (ML) algorithms. When examining the data from smart meters, it may help to identify power thieves with greater accuracy. In other words, Machine learning technologies, such as decision trees, random forests, support vector machines, neural networks, and others, can be used to create classification models in order to assess the daily electricity usage habits of customers [7], [8]. Typically, machine learning algorithms can be applied in either offline or online scheme. In the offline (i.e. batch) learning, a dataset of electricity consumption patterns is assumed to be available offline. Thereafter, a classification model is trained and evaluated to classify users as either malicious or benign based on their consumption patterns [9], [10]. The developed model can then be deployed in a real environment to make online predictions. On the other hand, the online (a.k.a incremental) learning scheme relies on the fact that smart meters reading arrive as a continuous stream of data. Hence, a classification model needs to be incrementally constructed by examining one instance at a time. Apparently, the batch learning scheme assumes that the whole dataset is stored in memory while building a machine learning model. However, it is well recognized that the batch learning approach has a number of drawbacks. First, the training phase could take a very long time and use up a significant portion of computer resources. Second, the amount of the training dataset has an

impact on the trained model's performance. Third, after the model is trained, it cannot acquire new experience from new input instances since in a offline (i.e., batch) learning scheme, the training data are assumed to be static and unchanging over time. To put it another way, it is necessary to build a new model whenever the statistical characteristics of the model's input change (i.e., a concept drift is encountered). Online classification algorithms are advantageous over off-line (i.e. batch) classification algorithms for a number of reasons, especially given that the smart meter readings in smart grids provide a constant stream of data. First off, algorithms for online classification are built to handle infinite amounts of data and gradually pick up new information. While creating projections as necessary, they are continuously updated. Second, real-time applications that conventional (i.e. batch) learning algorithms cannot handle can be addressed by online data stream classification systems. Online classification is thus viewed as a viable technique for classifying electricity consumption patterns in smart grid systems because user behavior may change over time in an unanticipated way. Numerous techniques have been put forth for the classification of data streams [11]–[13]. To the best of the authors' knowledge, no study has ever been done on how well these algorithms perform in detecting electricity theft in smart grids, despite the fact that some of them have been studied in various fields [14]–[21]. In addition, previous research efforts have tackled electricity theft detection using batch learning algorithms [9], [22]–[25]. Hence, the contribution of this paper is threefold. First, implementing online machine learning models for electricity theft detection, based on a recent specialized dataset. Second, performing an extensive set of experiments under both binary and multi-class theft detection scenarios. Third, identifying the most viable online machine learning model for theft detection in smart grids, considering a representative set of performance metrics.

The rest of this paper is organized as follows. Section II provides background information on the considered algorithms. Section III explains the research tools and evaluation methodology. Evaluation results are shown in Section IV. Finally, Section V concludes and summarizes this paper.

II. Background

Models from static datasets have traditionally been created using ML techniques. The need for models that can handle enormous data streams is, nevertheless, expanding. This means that additional data samples might appear at any time, and it is unsuitable to store them in a static dataset.

On the one hand, learning from continuous and evolving data streams necessitates the development of the ML model and continual stream upgrades. Additionally, it is crucial to combat concept drift, in which the statistical characteristics of the evolving data change with time [26], [27]. The resultant ML model must also be immediately updated for smart grid environments, needing algorithms with appropriate levels of accuracy subject to constrained memory and processing capacity.

A. Bayes Learning Algorithms

The Naive Bayes (NB) algorithm is used in this category. The NB method uses Bayesian prediction on the presumption that each input feature included within an input instance is independent. An NB model predicts every incoming data sample's class with a high degree of certainty. The NB algorithm is distinguished by its simplicity and minimal processing demands [12].

B. Lazy Learning Algorithms

The k-Nearest Neighbors classifier (kNN) and the self-adjusting memory combined with the kNN classifier (SAM-kNN) [28], [29] are two well-known lazy learning algorithms that are taken into consideration in this work. In online learning environments, the kNN algorithm relies on maintaining track of a window with a fixed number of recently encountered input data samples. The kNN algorithm looks within the recently stored window and, using a predetermined distance metric, determines the closest neighbors whenever a new input data sample is observed. The current input sample's class label is then allocated appropriately. The SAM-kNN, on the other hand, is an improvement over the standard KNN. A self-adjusting memory (SAM) model creates an ensemble of classification models for either current or prior concepts in SAM-KNN. Depending on the needs of the present concept, several models can be used. A short-term (STM) and long-term (LTM) memory are built specifically by the SAM model. The STM is built to represent the current concept, whereas the LTM is used to represent earlier concepts. A cleaning procedure is utilized to regulate the STM's size and keep the LTM and the STM consistent.

C. Tree-based Learning Algorithms

Online machine learning applications frequently employ tree-based methods. The Hoeffding Tree (HT) [30] and the Extremely Fast Decision Tree (EFDT) from [31] are the two main tree-based algorithms employed in this work. The HT method is a decision tree induction method that, under the premise that the distribution that yields the entering data samples is constant and does not evolve over time, may learn gradually and whenever from immense online data streams.It is based on the observation that choosing the best splitting attribute may frequently be done with only a limited quantity of input samples. This statement is supported theoretically by the Hoeffding bound, which counts the number of input instances needed to estimate a particular set of statistics with a given precision. The HT technique is potentially more enticing than other incremental (i.e., online) tree-based algorithms because it provides high performance guarantees. It can be demonstrated that the outcome of an HT model is asymptotically identical to that of a batch-based learner employing infinitely many input data samples by depending on the Hoeffding bound. Additionally, the EFDT classification algorithm incrementally constructs a tree. Once it is certain that a split is useful, it looks for picking and deploying that split. Later, it reviews that split choice and replaces it if it becomes clear that a more advantageous split is there. If the distribution that generates the input instances is stable, the EFDT can quickly pick up on static distributions and finally learn the asymptotic offline tree.

D. Ensemble Learning Algorithms

Two ensemble learning methodologies are assessed in this article including Leveraging Bagging LB [32] and Adaptive

Random Forest ARF [33]. Leveraging bagging is an enhanced online bagging algorithm. In this regard, online bagging mimics conventional offline Bagging to cope with incremental learning. For offline bagging scheme, N samples are taken from an N sized training dataset with replacement creating N separate datasets for M classifiers to be trained on. Since there is no training dataset but only a stream of samples in online learning environments, drawing input samples with replacement is not an easy task. The online bagging simulates the batch based training process by training each base estimator on each incoming instance over k times, where k is drawn from the binomial distribution. Given that the input stream may be considered endless and that the binomial distribution approaches a Poisson λ = 1 distribution with infinite samples, the work in [34] has found that the procedure used by the online bagging algorithms is a good "drawing with replacement". The LB algorithm makes an effort to enhance classification outcomes when assuming an infinite input data stream by modifying Poisson distribution's parameters produced from the binomial distribution. The LB technique causes the λ value of the Poisson distribution to change from 1 to 6. The new value of λ would broaden the input space's diversity by giving the input data samples a variety of weights. In order to achieve even greater improvement, the LB approach uses output detecting codes. Each bit in the n-bit long binary code used to encode the detection codes for each class label corresponds to a particular one of the n classifiers. Every classifier is trained on its corresponding bit while a new input instance is being looked at. This helps the LB algorithm reduce linked errors to a certain extent.

The standard batch based random forest technique has been modified for the online learning scope by the ARF algorithm A weighted voting method is used in ARF to decide how to categorize each incoming data instance after many decision trees have been built. The classification choice is prioritized and the voting procedure is weighted more heavily in favor of the decision tree that performs the best in terms of Kappa or the accuracy statistic.

III. Research Tools and Methodology

A. Dataset

This work is based on the Theft Detection Dataset (TDD2022) proposed in [3]. The dataset was gathered using the Open Energy Data Initiative (OEDI) platform which is a consolidated repository for high-value energy research datasets collected from the Programs, Offices, and National Laboratories of the United States Department of Energy [35]. The information in TDD2022 stems from various domains such as private industrial parties, laboratories, institutions, etc. The dataset is composed energy consumption data for 16 different consumer types. It encloses several energy consumption measurements for distinct customer types during a one-year period. Those measurements are recorded on hourly basis during the day. This data was then used to implement a theft generator for six different types of electricity theft. Each instance in the dataset contains 11 meter readings, consumer type, and a class label as either normal consumption or one of the six theft types. Tables I, II, III and IV illustrate the dataset general statistics, feature types, customer types and instances distribution on classes, respectively.

TABLE I. General Statistics

Item	values
Number of instances	560640
Number of categorical features	1
Number of numerical features	10
Customer types	16
Instances per customer type	35040
Number of classes	7

TABLE II. Customer Types

Type	Integer code
Full service restaurant	1
Hospital	2
Large hotel	3
Large office	4
Medium office	5
Medrise apartment	6
Primary school	7
Outpatient	8
Warehouse	9
Secondary school	10
Small hotel	11
Small office	12
Stand-alone retail	13
Strip mall	14
Supermarket	15
Quic service restaurant	16

TABLE III. Feature Types

Name	Type
Electricity-Facility (KW/Hr)	Numeric
Fans-Electricity (KW/Hr)	Numeric
Cooling-Electricity (KW/Hr)	Numeric
Heating-Electricity (KW/Hr)	Numeric
Interior lights-Electricity (KW/Hr)	Numeric
Interior equipment-Electricity (KW/Hr)	Numeric
Gas-Facility (KW/Hr)	Numeric
Heating-Gas (KW/Hr)	Numeric
Interior equipment-Gas (KW/Hr)	Numeric
Water systems-Gas (KW/Hr)	Numeric
Consumer Type (KW/Hr)	Categorical

TABLE IV. Instances Distribution

Class name	Total number of instances
Normal	331824
Theft-1	51083
Theft-2	22958
Theft-3	44349
Theft-4	41460
Theft-5	33553
Theft-6	35413

The first theft type consists of a pronounced reduction of electricity consumption during the day. Such reduction is attained by multiplying the consumption by a uniformly distributed random number in the interval[0.1,0.8]. For the second theft type, electricity consumption is randomly dropped to zero throughout an arbitrary period. In addition, the third theft type resembles the first type except the fact that each hourly consumption is multiplied by a random number. Moreover, a random portion of the mean consumption is generated for the fourth theft type. Furthermore, the fifth type of theft reports the mean consumption. Finally, the sixth theft type reverses the order of the consumption values.

B. Evaluation Methodology

This section outlines the key procedures used to assess the effectiveness of the online machine learning (i.e., classification) algorithms on the TDD2022 dataset. The evaluation process using the scikit-multiflow evaluation platform [36] is shown in Fig. 1. Every online classification algorithm goes through this review process. As seen, the dataset is initially loaded as an input stream and then sent to the classification algorithm after that algorithm's initialization for online testing, incremental learning, and evaluation.

The prequential or the interleaved test then train method is used in this study to assess the classification algorithms. As each incoming input sample (i.e. instance) serves two purposes and is analyzed sequentially in order of arrival before becoming instantly inaccessible, the prequential assessment approach was created specifically for online learning environments. In prequential evaluation, each observed input instance is first employed to test the classification model (i.e. to generate a prediction), and then the same input instance is used to train that classification model. Each tested model's performance is continuously updated after each encountered instance and its capacity to handle unobserved cases is continuously monitored in real-time. As a result, a classification model that has been instantiated is constantly tested and the metrics that go with it are updated for input instances that it has not yet encountered. A number of commonly used performance metrics including accuracy, precision, recall, F-1 score and the kappa statistic derived from online learning models are used to quantify the performance of the classification algorithms. These measures are defined as follows:

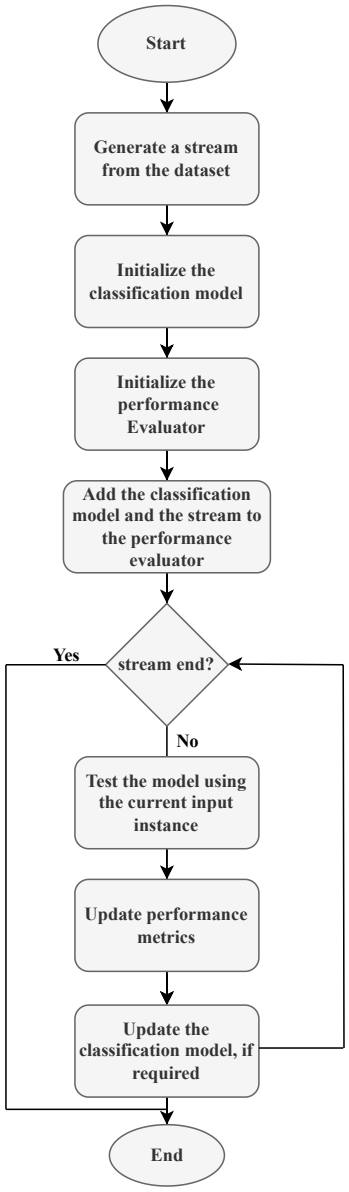


Fig. 1. Evaluation Flowchart.

- **Classification accuracy**: is the proportion of correctly classified input instances.

$$Accuracy = \frac{TN + TP}{TP + FP + FN + TN} \times 100\% \quad (1)$$

where, respectively, TP, TN, FP, and FN stand for true positive, true negative, false positive, and false negative. TP is the total number of cases that were successfully identified as positive (i.e., theft). The number of successfully identified negative (i.e., normal) events is referred to as TN. FP is the total number of positive samples that are mistakenly labeled as negative ones. The total number of negative occurrences that are mistakenly labeled as positive occurrences is known as FN.

- **Precision:** determines the proportion of predictions for the positive class that are in fact members of the positive class.

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

- **Recall:** calculates the proportion of correctly predicted classes that are positive out of all occurrences that are positive in the observed stream.

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

- **F-score:** is the precision and recall harmonic mean.

$$F - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (4)$$

- **Kappa statistic** (κ): is a reliable classification accuracy metric that takes the likelihood of agreement by chance into account. It indicates the superiority over the majority class classifier, which assume that all

incoming cases will fall within the majority class [37]. In particular for data streams with unbalanced class distribution, it is crucial in assessing classification accuracy.

$$\kappa = \frac{p_0 - p_c}{1 - p_c} \tag{5}$$

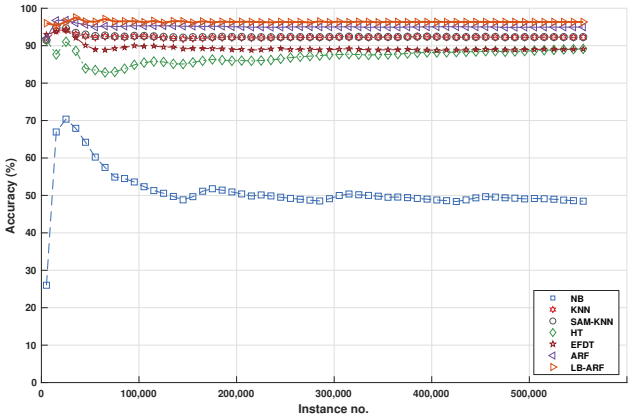
where p_0 denotes the classifier's predictive accuracy and p_c denotes the likelihood that a random classifier will produce an accurate prediction [38]. The classification procedure is always correct if $kappa = 1$.

IV. Results and Analysis

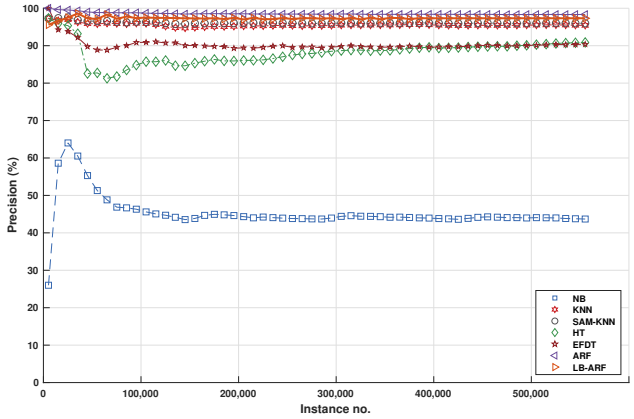
This section shows the predictive performance of the considered online machine learning algorithm on a data stream generated from the TDD2022 dataset. The considered algorithms were tested under both binary and multi-class classification settings. In binary classification, each instance in the TDD2022 dataset is labelled as either normal or theft instance regardless of the theft type. For the multi-class classification, the consumption instances retain their original labelling (i.e.,normal, Theft-1, Theft-2, Theft-3, Theft-4, Theft-5 or Theft-6). Fig. 2(a), 2(b), 2(c), 2(d) and 2(e) depict the running mean prediction accuracy, precision, recall, F-1 score and kappa statistics of the considered learning algorithms under binary classification settings. They considered algorithms were pre-trained on the first 5000 samples and then prequentially evaluated and trained on the remaining part of the consumption stream. As shown, the KNN, SAM-KNN, ARF and LB-ARF algorithms have steadily maintained highr mean values of accuracy, precision, F-1 score and kappa statistic, as compared to the NB, HT and EFDT algorithms. In addition, the NB, HT and EFDT algorithms demonstrate fluctuating performance during the first 100,000 instances. On the other hand, the NB, ARF and LB-ARF have maintained higher mean recall values, when compared to the other algorithms. However, the NB algorithm exhibits fluctuating behaviour during the 200,000 instances. Furthermore, the relatively high kappa values of the ARF and LB-ARF algorithms indicate reasonable reliability of their predictive performance. In other words, they are able to incrementally learn the statistical characteristics of the incoming normal and theft instances adapt reliably to unseen instances. Overall, the LB-ARF algorithms outperforms the other algorithm under all the considered performance metrics. Fig. 3(a), 3(b), 3(c), 3(d) and 3(e) depict the running mean prediction accuracy, precision, recall, F-1 score and kappa statistics of the considered learning algorithms under multi-class classification settings. On the one hand, Fig. 3(a) depicts that the KNN, SAM-KNN, ARF and LB-ARF have achieved relatively acceptable accuracy levels ($\geq 80\%$), as compared to the other algorithms, taking into account the complexity of multi-class classification as compared to the binary one. On the other hand, Fig. 3(b), 3(c) and 3(d) demonstrate the ability of the LB-ARF algorithm to maintain acceptable precision, recall and F-1 score, when compared to the other algorithms. Similar to the case of binary classification, the relatively high kappa value of the LB-ARF demonstrate its superior performance reliability over other algorithms. Fig. 4(a), 4(b), 4(c), 4(d) and 4(e) compare the predictive performance of the considered algorithms on binary and multi-class theft detection settings. In general, the predictive performance of all algorithms in binary classification is higher as compared to that of multi-class classification. Unlike other algorithms, the LB-ARF algorithm has maintained comparable performance levels under all metrics for both binary and multi-class classification. It is worth noting that the tree-based algorithms exhibit higher performance drop when moving from binary to multi-class classification, as compared to the other categories.

In summary, the LB-ARF (i.e., Leveraging Bagging algorithm with an Adaptive Random Forest base classifier) demonstrates consistent competence to perform theft detection under both binary and multi-class classification scenarios. This algorithm keeps a collection of n ARF base classifiers, where n in the used evaluation platform is set by default to 10 [36]. In order to classify an incoming instance, each classifier will make a prediction (i.e., a vote), and the ultimate classification result is produced by combining the individual forecasts. The Condorcet's jury theorem has a theoretical demonstration, assuming two criteria are satisfied, that the error rate of a particular ensemble tends to zero in the limit [39]–[41]. First, Individual base classifiers must outperform random guessing. This requirement is attained as the ARF algorithm achieves relatively high predictive performance that is better than random guessing as shown in Fig. 2, 3 and 4. Typically, the accuracy of a random classifier (i.e., random guess) is equal to $1/k$ where k is the total number of classes. In this work, the total number of classes is equal to 2 in case of binary classification and 7 in case of multi-class classification.

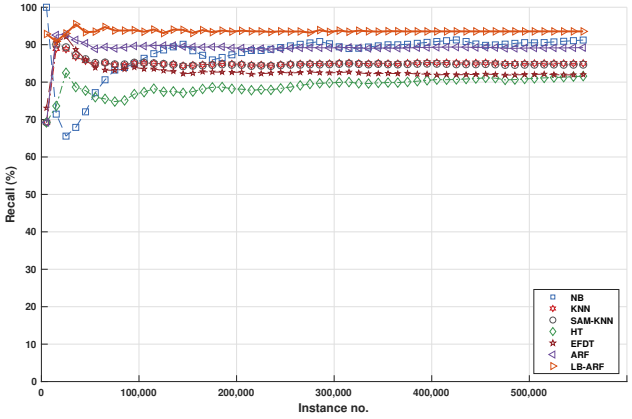
Second, each classification model must be diverse; that is, it must not generate correlated errors. For the LB-ARF method, online bagging is used by the LB algorithm to train its associated base classification models. In this context, an online re-sampling is carried out as each incoming classification case is noticed by showing that instance to every model $k\ sim$ Poisson (λ) times and updating every model in accordance. The value of k is treated as the incoming instance's weight. In order to increase online re-sampling, the λ value of the Poisson distribution is typically set to 6 in the LB algorithm. The LB ensemble algorithm is making the incoming instances weights more random with such a value of λ. As a result, it increases the diversity of the input space by giving each incoming instance a new range of weights. The LB technique further improves bagging performance by applying output codes to add randomization to the ensemble's output. As seen in Section II-D, Each prospective class label is given an n-bit binary string, where n is the total number of base classifiers in the ensemble. Each base classifier learns a single bit from the binary string. The LB algorithm utilized random output codes instead of deterministic ones, in contrast to typical ensemble approaches. To put it another way, employing output codes enables each classifier in the LB ensemble to predict a separate function, whereas the base classifiers in the traditional approaches predict the same function [32]. This would reduce the impact of correlations among base classifiers and, as a result, improve the ensemble's diversity [42], [43]. The ensemble thus partially satisfies the second criteria of the Condorcet's jury theorem by adding randomization to both the input and the output of the ensemble's base classifiers. Additionally, the LB method employs the ADWIN method to handle concept drift, employing ADWIN instance per classifier in the ensemble [28]. The poorest classifier is reset whenever a concept drift is found. As a result, the LB algorithm con-



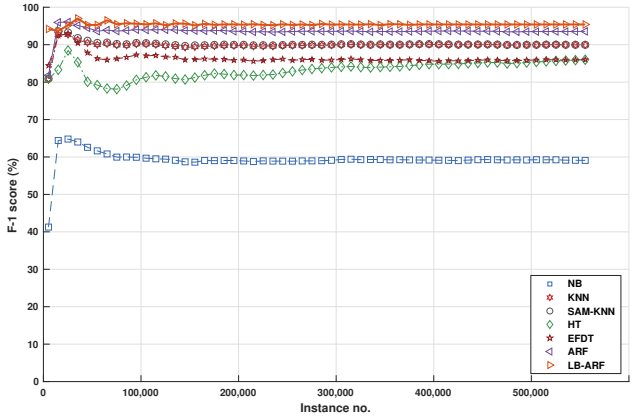
(a) Accuracy



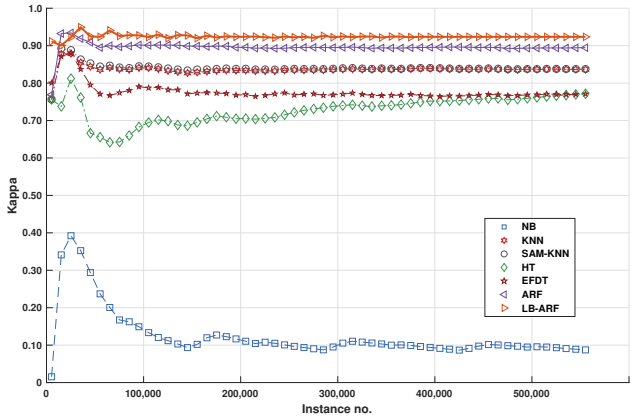
(b) Precision



(c) Recall



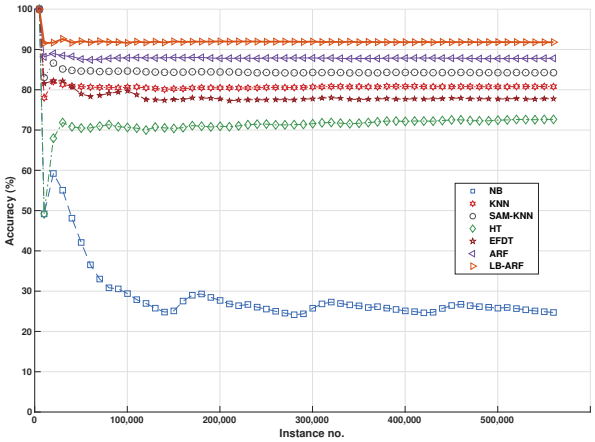
(d) F-1 score



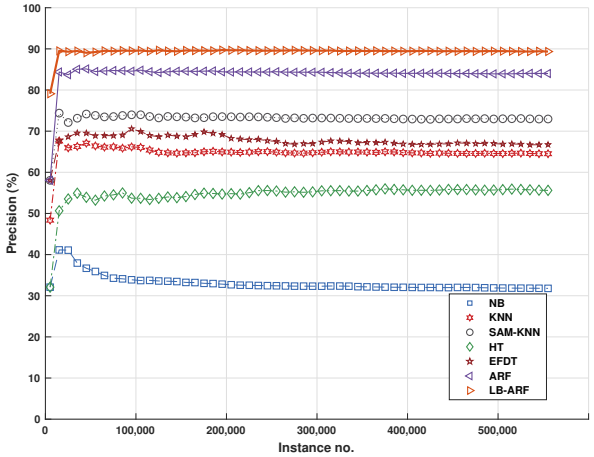
(e) Kappa

Fig. 2. Performance Results - Binary Classification

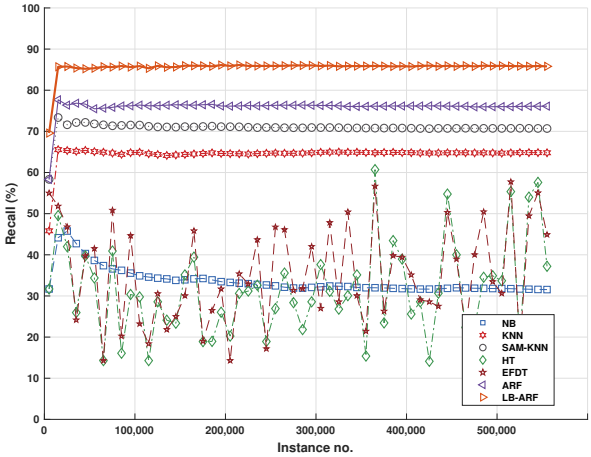
tinuously assesses the effectiveness of its learning procedure and follows the current distribution of class labels within the incoming classification examples. The classification errors caused by any given classifier would typically be offset by the LB-ARF's diversity among its basic classifiers. This can be observed in Fig. 2, 3 and 4 wherein an LB ensemble of ARF classifiers always achieve higher predictive performance than a single ARF instance. Overall, the LB-ARF algorithm has demonstrated its ability to sustain an audible performance under all taken into account performance metrics. This makes it a viable option for online theft detection (i.e. classification) in real-world smart grids.



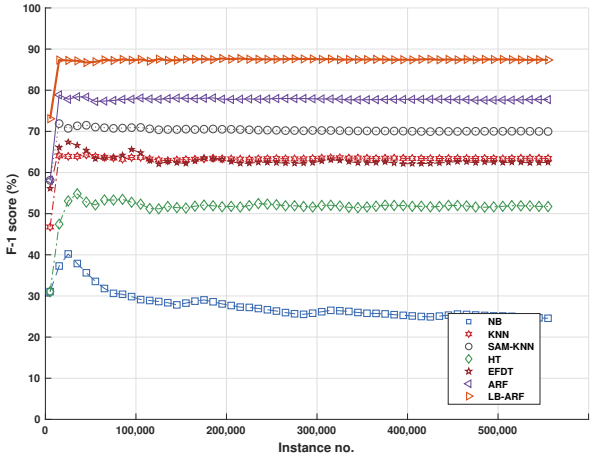
(a) Accuracy



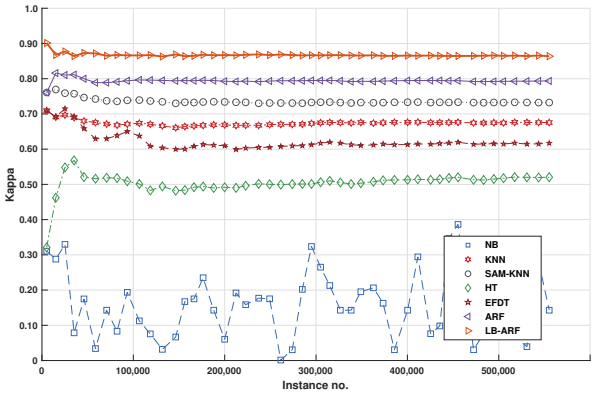
(b) Precision



(c) Recall



(d) F-1 score

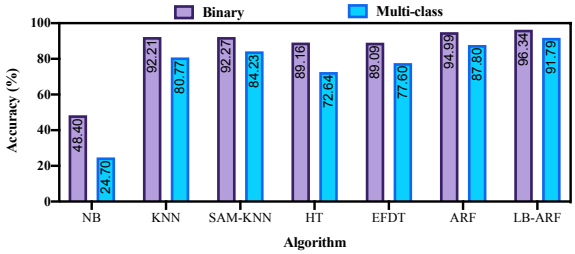


(e) Kappa

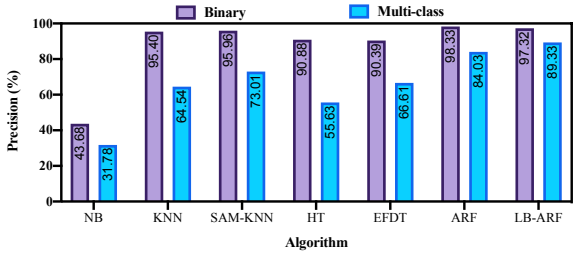
Fig. 3. Performance Results - Multi-Class Classification

V. Conclusion

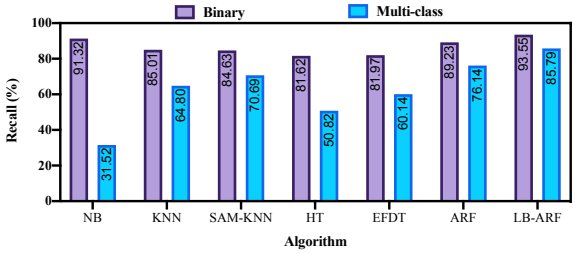
Power loss brought on by electricity theft is a critical issue in both traditional and smart grid settings. Smart meters can be used in smart grids to monitor power usage patterns and spot any questionable activities. However, using hacking techniques or cyber attacks can undermine smart meter readings. In this sense, machine learning algorithms can be employed to evaluate client daily consumption patterns and identify probable instances of electricity theft. This work studied the application of online machine learning algorithms for electricity theft detection in smart grid systems, based on a recently



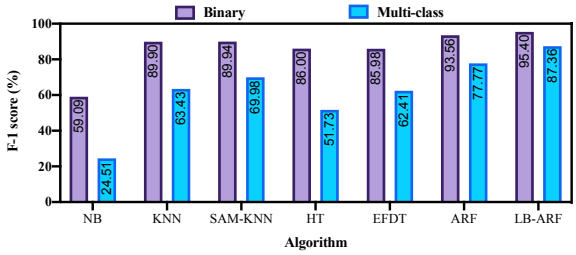
(a) Accuracy



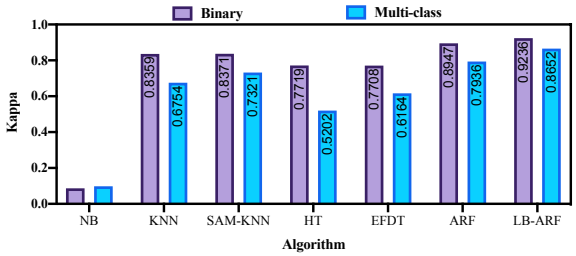
(b) Precision



(c) Recall



(d) F-1 score



(e) Kappa

Fig. 4. Performance Comparison - Binary vs. Multi-Class

proposed theft detection dataset. Evaluation results showed that leveraging bagging with an adaptive random forest base estimator surpassed its online machine learning counterparts in both binary and multi-class theft detection. Hence, it can be viewed as a promising online learning model for electricity theft detection in smart grids.

REFERENCES

[1] M. N. Hasan, R. N. Toma, A.-A. Nahid, M. M. M. Islam, and J.-M. Kim, "Electricity theft detection in smart grid systems: A cnn-lstm based approach," *Energies*, vol. 12, no. 17, 2019. [Online]. Available: https://www.mdpi.com/1996-1073/12/17/3310

[2] M. Adil, N. Javaid, U. Qasim, I. Ullah, M. Shafiq, and J.-G. Choi, "Lstm and bat-based rusboost approach for electricity theft detection," *Applied Sciences*, vol. 10, no. 12, 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/12/4378

[3] S. Zidi, A. Mihoub, S. Mian Qaisar, M. Krichen, and Q. Abu Al-Haija, "Theft detection dataset for benchmarking and machine learning based classification in a smart grid environment," *Journal of King Saud University - Computer and Information Sciences*, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1319157822001562

[4] S. Hussain, M. W. Mustafa, T. A. Jumani, S. K. Baloch, H. Alotaibi, I. Khan, and A. Khan, "A novel feature engineered-catboost-based supervised machine learning framework for electricity theft detection," *Energy Reports*, vol. 7, pp. 4425–4436, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352484721004716

[5] L. G. Arango, E. Deccache, B. D. Bonatto, H. Arango, and E. O. Pamplona, "Study of electricity theft impact on the economy of a regulated electricity company," *Journal of Control, Automation and Electrical Systems*, vol. 28, no. 4, pp. 567–575, Aug 2017. [Online]. Available: https://doi.org/10.1007/s40313-017-0325-z

[6] 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, 2019.

[7] S. S. S. R. Depuru, L. Wang, V. Devabhaktuni, and P. Nelapati, "A hybrid neural network model and encoding technique for enhanced classification of energy consumption data," in *2011 IEEE Power and Energy Society General Meeting*, 2011, pp. 1–8.

[8] 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.

[9] L. J. Lepolesa, S. Achari, and L. Cheng, "Electricity theft detection in smart grids based on deep neural network," *IEEE Access*, vol. 10, pp. 39 638–39 655, 2022.

[10] A. Ullah, N. Javaid, M. Asif, M. U. Javed, and A. S. Yahaya, "Alexnet, adaboost and artificial bee colony based hybrid model for electricity theft detection in smart grids," *IEEE Access*, vol. 10, pp. 18 681–18 694, 2022.

[11] A. Gepperth and B. Hammer, "Incremental learning algorithms and applications," in *European Symposium on Artificial Neural Networks (ESANN)*, Bruges, Belgium, 2016. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01418129

[12] Q. Yang, Y. Gu, and D. Wu, "Survey of incremental learning," in *2019 Chinese Control And Decision Conference (CCDC)*, 2019, pp. 399–404.

[13] K. K. Wankhade, S. S. Dongre, and K. C. Jondhale, "Data stream classification: a review," *Iran Journal of Computer Science*, vol. 3, no. 4, pp. 239–260, Dec 2020. [Online]. Available: https://doi.org/10.1007/s42044-020-00061-3

[14] U. Adhikari, T. H. Morris, and S. Pan, "Applying hoeffding adaptive trees for real-time cyber-power event and intrusion classification," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 4049–4060, 2018.

[15] Z. E. Mrabet, D. F. Selvaraj, and P. Ranganathan, "Adaptive hoeffding tree with transfer learning for streaming synchrophasor data sets," in *2019 IEEE International Conference on Big Data*, 2019, pp. 5697–5704.

[16] C. Nixon, M. Sedky, and M. Hassan, "Practical application of machine learning based online intrusion detection to internet of things networks," in *2019 IEEE Global Conference on Internet of Things (GCIoT)*, 2019, pp. 1–5.

[17] V. G. Turrisi da Costa, E. J. Santana, J. F. Lopes, and S. Barbon, "Evaluating the four-way performance trade-off for stream classification," in *Green, Pervasive, and Cloud Computing*. Cham: Springer International Publishing, 2019, pp. 3–17.

[18] J. Fernandes Lopes, E. J. Santana, V. G. Turrisi da Costa, B. Bogaz Zarpelão, and S. Barbon Junior, "Evaluating the four-way performance trade-off for data stream classification in edge computing," *IEEE Transactions on Network and Service Management*, vol. 17, no. 2, pp. 1013–1025, 2020.

[19] M. Al-Tarawneh, "Data stream classification algorithms for workload orchestration in vehicular edge computing: A comparative evaluation," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 21, no. 2, pp. 101–122, 2021.

[20] M. Rahouti, M. Ayyash, S. K. Jagatheesaperumal, and D. Oliveira, "Incremental learning implementations and vision for cyber risk detection in iot," *IEEE Internet of Things Magazine*, vol. 4, no. 3, pp. 114–119, 2021.

[21] M. A. B. Al-Tarawneh and S. E. Alnawayseh, "Performance assessment of context-aware online learning for task offloading in vehicular edge computing systems," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 4, 2021. [Online]. Available: http://dx.doi.org/10.14569/IJACSA.2021.0120439

[22] D. Gu, Y. Gao, K. Chen, S. Junhao, Y. Li, and Y. Cao, "Electricity theft detection in ami with low false positive rate based on deep learning and evolutionary algorithm," *IEEE Transactions on Power Systems*, pp. 1–1, 2022.

[23] A. Arif, T. A. Alghamdi, Z. A. Khan, and N. Javaid, "Towards efficient energy utilization using big data analytics in smart cities for electricity theft detection," *Big Data Research*, vol. 27, p. 100285, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214579621001027

[24] F. Shehzad, N. Javaid, S. Aslam, and M. Umar Javed, "Electricity theft detection using big data and genetic algorithm in electric power systems," *Electric Power Systems Research*, vol. 209, p. 107975, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S037877962200205X

[25] M. Ezeddin, A. Albaseer, M. Abdallah, S. Bayhan, M. Qaraqe, and S. Al-Kuwari, "Efficient deep learning based detector for electricity theft generation system attacks in smart grid," in *2022 3rd International Conference on Smart Grid and Renewable Energy (SGRE)*, 2022, pp. 1–6.

[26] G. I. Webb, R. Hyde, H. Cao, H. L. Nguyen, and F. Petitjean, "Characterizing concept drift," *Data Mining and Knowledge Discovery*, vol. 30, no. 4, pp. 964–994, Jul 2016. [Online]. Available: https://doi.org/10.1007/s10618-015-0448-4

[27] J. Demšar and Z. Bosnić, "Detecting concept drift in data streams using model explanation," *Expert Systems with Applications*, vol. 92, pp. 546 – 559, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417417306772

[28] A. Bifet and R. Gavaldà, "Learning from time-changing data with adaptive windowing," in *Proceedings of the Seventh SIAM International Conference on Data Mining*. SIAM, 2007, pp. 443–448. [Online]. Available: https://doi.org/10.1137/1.9781611972771.42

[29] V. Losing, B. Hammer, and H. Wersing, "Knn classifier with self adjusting memory for heterogeneous concept drift," in *2016 IEEE 16th International Conference on Data Mining (ICDM)*, 2016, pp. 291–300.

[30] G. Hulten, L. Spencer, and P. Domingos, "Mining time-changing data streams," in *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '01. New York, NY, USA: Association for Computing Machinery, 2001, p. 97–106. [Online]. Available: https://doi.org/10.1145/502512.502529

[31] C. Manapragada, G. I. Webb, and M. Salehi, "Extremely fast decision tree," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 1953–1962. [Online]. Available: https://doi.org/10.1145/3219819.3220005

[32] A. Bifet, G. Holmes, and B. Pfahringer, "Leveraging bagging for evolving data streams," in *Machine Learning and Knowledge Discovery in Databases*, J. L. Balcázar, F. Bonchi, A. Gionis, and M. Sebag, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 135–150.

[33] H. M. Gomes, A. Bifet, J. Read, J. P. Barddal, F. Enembreck, B. Pfharinger, G. Holmes, and T. Abdessalem, "Adaptive random forests for evolving data stream classification," *Machine Learning*, vol. 106, no. 9, pp. 1469–1495, Oct 2017. [Online]. Available: https://doi.org/10.1007/s10994-017-5642-8

[34] N. C. Oza, "Online bagging and boosting," in *2005 IEEE International Conference on Systems, Man and Cybernetics*, 2005, pp. 2340–2345.

[35] J. B. Leite and J. R. S. Mantovani, "Detecting and locating non-technical losses in modern distribution networks," *IEEE Transactions on Smart Grid*, vol. 9, no. 2, pp. 1023–1032, 2018.

[36] J. Montiel, J. Read, A. Bifet, and T. Abdessalem, "Scikit-multiflow: A multi-output streaming framework," *Journal of Machine Learning Research*, vol. 19, no. 72, pp. 1–5, 2018. [Online]. Available: http://jmlr.org/papers/v19/18-251.html

[37] T. Vasiloudis, F. Beligianni, and G. De Francisci Morales, "Boostvht: Boosting distributed streaming decision trees," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, ser. CIKM '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 899–908. [Online]. Available: https://doi.org/10.1145/3132847.3132974

[38] A. Bifet, G. de Francisci Morales, J. Read, G. Holmes, and B. Pfahringer, "Efficient online evaluation of big data stream classifiers," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 59–68. [Online]. Available: https://doi.org/10.1145/2783258.2783372

[39] L. K. Hansen and P. Salamon, "Neural network ensembles," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 10, pp. 993–1001, 1990.

[40] K. K. Ladha, "Condorcet's jury theorem in light of de finetti's theorem," *Social Choice and Welfare*, vol. 10, no. 1, pp. 69–85, Jan 1993. [Online]. Available: https://doi.org/10.1007/BF00187434

[41] J. N. van Rijn, G. Holmes, B. Pfahringer, and J. Vanschoren, "The online performance estimation framework: heterogeneous ensemble learning for data streams," *Machine Learning*, vol. 107, no. 1, pp. 149–176, Jan 2018. [Online]. Available: https://doi.org/10.1007/s10994-017-5686-9

[42] Y. Lv, S. Peng, Y. Yuan, C. Wang, P. Yin, J. Liu, and C. Wang, "A classifier using online bagging ensemble method for big data stream learning," *Tsinghua Science and Technology*, vol. 24, no. 4, pp. 379–388, 2019.

[43] M. Kolárik, M. Sarnovský, and J. Paralič, "Diversity in ensemble model for classification of data streams with concept drift," in *2021 IEEE 19th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, 2021, pp. 355–360.