The Effectiveness of Using AutoML in Electricity Theft Detection: The Impact of Data Preprocessing and Balancing Techniques

Suhad A. Yousif[1] and Venus W. Samawi[2(✉)]

[1] Computer Science Department, Al-Nahrain University, Baghdad, Iraq
[2] Department of MIS/Smart Business, Isra University, Amman, Jordan
venus.samawi@iu.edu.jo

Abstract. Electricity theft is a significant threat to the economy and society's security. Machine learning models are applied to consumption data obtained from smart meters to detect and prevent it. The consumption data must undergo preprocessing and feature selection techniques to enhance detection accuracy. Moreover, the data is imbalanced, where it contains rare malicious samples. This study uses AutoML to detect electricity theft by utilizing its useful services, such as data preprocessing, feature engineering, algorithm selection, and model hyperparameter optimization. We will showcase and evaluate two prominent platforms, TPOT and H2O, and analyze the effectiveness and accuracy of each. The study will examine whether applying additional preprocessing for outlier detection and handling impacts TPOT's and H2O's performance. Additionally, we will examine the performance of each platform on the unbalanced datasets related to electricity theft and compare it with their performance after balancing the dataset using three different techniques - Random Oversampling, Synthetic Minority Oversampling Technique (SMOTE), and adaptive synthetic (ADASYN). After analyzing the results, the Random Oversampling Accuracy technique was the best among the three techniques for balancing the dataset. The TPOT achieved the highest accuracy of 99% on the balanced dataset with Random Oversampling, while H2O reached an accuracy of 96.8%. However, H2O outperforms TPOT when applied to the unbalanced dataset. The results demonstrate that inappropriate handling of outliers reduces accuracy, particularly with imbalanced datasets. However, accuracy is improved after balancing it.

Keywords: Electricity theft detection · Data Balancing Techniques · AutoML · TPOT · H2O · Outlier Detection and Handling

1 Introduction

The smart grid (SG) is crucial in managing electricity production, distribution, and control. However, power losses often occur during the production, distribution, and transmission of electrical energy due to technical (TLs) and non-technical factors (NTLs). Electricity theft is a significant cause of (NTLs) [1, 2]. It occurs when users fraudulently

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
O. Gervasi et al. (Eds.): ICCSA 2024, LNCS 14814, pp. 68–82, 2024.
https://doi.org/10.1007/978-3-031-64608-9_5

consume energy by stealing electric energy from power grids, resulting in non-technical losses in the power sector. Three main strategies could be used to detect power theft*: network-oriented methodology*, *data-oriented technique*, and a *hybrid approach* that combines the two methods. Implementing network- and hybrid-oriented solutions is often not feasible due to security risks and high installation costs, as it requires regular modification of the network architecture [3]. Consequently, data-driven techniques have been implemented for Electricity Theft Detection (ETD). This approach uses data obtained from smart meters to forecast electricity theft. Research has demonstrated that using data-driven techniques enhances the precision of ETD, as stated by [2, 4]. Deep learning (DL) and machine learning (ML) techniques are commonly employed to predict instances of electricity theft by identifying abnormal electricity consumption patterns [5–8]. Furthermore, a combination of DL and ML techniques can be used to detect electricity theft by analyzing data collected from smart meters [1, 9]. ML algorithms analyze vast amounts of data to detect anomalous patterns. Since the data collected may contain noise, inaccuracies, and missing values, it needs to be cleaned and preprocessed first. After that, engineers apply feature engineering techniques to the data to enhance the effectiveness of ML algorithms. On the other hand, deep learning algorithms are capable of generating features from raw data; however, the data still needs to be cleaned, and data noise should be handled. Moreover, the data available for identifying electricity fraud is complex and imbalanced, making it difficult to identify fraudulent patterns through traditional machine learning and neural network techniques [10]. However, AutoML platforms can be a useful solution, as they offer valuable services such as data preprocessing, feature engineering, algorithm selection, and model hyperparameter optimization. The advantage of using Auto ML platforms is that researchers do not need to be experts in machine learning and deep learning techniques. These platforms also help to select the most appropriate technique for solving the problem [11]. Imbalanced data is biased towards the majority class, and data balancing techniques can address this issue.

Using data-oriented techniques for electricity theft detection (ETD) faces challenges in handling large electricity consumption datasets due to missing values, data variance, and nonlinear data relationship issues [4, 12]. Therefore, an effective preprocessing phase is needed to address this problem and prepare the dataset for the classification (or clustering) phase to get more accurate results [1]. Primary data preprocessing involves imputing missing values, handling outliers, and normalizing data [10, 12, 13]. Another crucial step in ML is feature engineering, including transforming raw data into new features and creating new variables by extracting them from the raw data (feature extraction). This helps train new ML models to tackle new tasks effectively, leading to more accurate predictions and insights [14]. AutoML platforms can automatically perform feature engineering to improve the handling of certain data types [15]. Finally, imbalanced data is a common issue in data science that requires data balancing techniques to overcome bias towards the majority class [16]. Resampling techniques are employed to address the imbalanced data problem. These techniques include oversampling methods such as Random oversampling, SMOTE, and ADASYN, as well as under-sampling methods like Random under-sampling, NearMiss, and Tomek-links. Choosing between under-sampling and oversampling techniques depends on the dataset characteristics and the

problem to be solved. Some studies have utilized both methods (under-sampling and oversampling) to balance the dataset [12]. According to the literature, oversampling is generally considered more effective for data balancing [17]. Based on this, we decided to investigate three oversampling techniques (Random, SMOTE, ADASYN) to balance the electricity consumption datasets and study their effect on the behaviour of the theft detection model used.

This study aims to use AutoML to tackle the problem of electricity theft prediction. We will test and assess the effectiveness and precision of two platforms, TPOT and H2O. These platforms provide multiple tools and services for data preprocessing and feature engineering, algorithm selection, and hyperparameter tuning for model building. Our study addresses challenges concerning handling outliers and unbalanced classes included in the used ETD dataset. Furthermore, the best AutoML platform (between TPOT and H2O) will be specified based on the results achieved by each of them. The best classifier in each of TPOT and H2O is specified based on the algorithm selection and model hyperparameter optimization provided by each platform. Additionally, our study intends to answer the following questions: 1) Which AutoML platform is better for identifying electricity fraud - TPOT or H2O? 2) To what extent do outlier detection and handling as a preprocessing step improve the performance of TPOT and H2O? 3) Do TPOT and H2O have similar outcomes to an imbalanced dataset? 4) Finally, we will investigate which data balancing technique (Random, SMOTE, and ADASYN) is more effective in enhancing the performance of TPOT and H2O on the used dataset.

Henceforth, this paper is organized in the following manner. Section 2 reviews AutoML platforms. Section 3 summarizes data preprocessing and feature engineering. Section 4 illustrates data balancing using resampling techniques. The system model is explained in Sect. 5. Section 6 summarizes the scenarios of experimental results of both TPOT and H2O and assesses the outcomes of the experiments. Ultimately, we stated the main findings in Sect. 6 and provided an outline for future work.

2 AutoML

AutoML (Automated Machine Learning) is an ML platform used to develop an ML model by automating the training and learning process. These platforms enable experts and non-experts to apply machine learning models to solve various problems such as regression, classification, and clustering [11, 13]. Many AutoML platforms, including Auto-Keras [21], Azure, TPOT, and H2O, are available. Each offers different features and capabilities. Some of them provide data preprocessing and feature engineering (H2O, MLBox, and TPOT), in addition to selecting the appropriate algorithms and optimizing the model's hyperparameters (H2O, MLBox, Auto-SKLearn, TPOT) [18, 19]. AutoML platforms provide a variety of tools and services for machine learning. ***Azure*** AutoML is particularly useful for supervised learning and time series forecasting. In contrast, ***Google Cloud*** AutoML uses reinforcement learning to interact with its environment and receive feedback. ***Run:ai*** is designed to automate resource management and workload orchestration for machine learning infrastructure [18, 20]. ***Auto-Keras*** uses a DL library and provides building blocks for performing architecture searches. It searches for the most suitable neural network architecture based on the dataset and task [21]. ***H2O***

utilizes ML and artificial intelligence technologies to accomplish various tasks. The tool offers model interpretation to assist users in comprehending the model's decision-making process. This is crucial for specific applications [22, 23]. Finally, ***TPOT*** is a tree-based optimization tool. It optimizes machine learning pipelines using genetic programming [24]. In this work, we are interested in studying the performance of two AutoML platforms, TPOT and H2O. TPOT and H2O use different approaches. Several research papers outline TPOT [25, 26]; however, limited information is available on H2O. Therefore, H2O will be explained in more detail.

In this work, we aim to utilize AutoML to address the problem of detecting electricity theft. The performance and accuracy of the two platforms, TPOT and H2O, will be tested and evaluated. Each platform provides its version of tools and services concerning data preprocessing and feature engineering, selecting the appropriate algorithms, and optimizing the model's hyperparameters.

2.1 H2O AutoML Platform

H2O is an AutoML platform used to handle complex models and datasets while maintaining scalability. When using H2O AutoML, users can customize the training model by specifying the runtime or choosing which models to combine. H2O AutoML provides automated tasks, including feature engineering, model selection, and hyperparameter tuning based on Bayesian optimization and Random search algorithms and others, simplifying machine learning workflows. It also selects the performance model based on validation techniques. It is worth mentioning that H2O can combine predictions from various models using stacked methods to improve accuracy, which involves a mental learner. Therefore, for data predictions, either the performing model or an ensemble of models is produced. H2O also provides interpretability tools to explain the decision-making process [22, 27, 28]. The ML stages in H2O include (see Fig. 1):

1. Data preparation: In this stage, the dataset passes through cleaning and transformation processes, where data imputation, encoding variables, and normalizing data are performed.
2. Features Engineering: H2O AutoML improves model performance by generating new features from input data and applying feature selection algorithms to reduce dimensionality and increase efficiency.
3. Exploring Models: In this stage, H2O tests different machine learning algorithms, namely linear models, tree-based methods such as Random Forests and Gradient Boosting Machines (GBMs), DL models, and ensemble methods. Each algorithm performs hyperparameter tuning to find optimal model parameters by using grid search, random search, or Bayesian optimization.
4. Models Evaluation: Cross-validation is applied to data to ensure model reliability and prevent overfitting to a subset of data. Furthermore, a leaderboard is maintained to rank models based on performance metrics such as accuracy, RMSE, AUC, and F1_mesure. The best classifier is selected from the model that tops the leaderboard.
5. Deploying in Real World Scenario: The chosen model can be easily deployed for making predictions in a production environment.

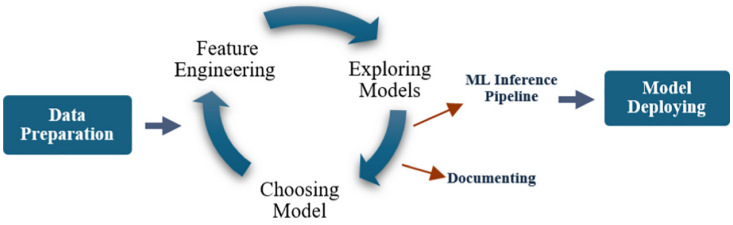


Fig. 1. The ML stages in H2O

H2O is an open-source ML platform with customized versions of algorithms for enhanced performance. AutoML automates tasks like feature engineering, model tuning, and deployment. H2O is available in multiple languages and integrates well with big data platforms. However, it is difficult to use with other frameworks like scikit learn. H2O architecture has multiple layers. The topmost layer, API, provides REST API clients for various platforms (such as Python, R, Excel, Tableau, and Flow Web UI using socket connections). The bottom layer is a big data platform that runs on a JVM process. An H2O cluster has nodes, each with a JVM process split into three layers: language, algorithms, and core infrastructure, as shown in Fig. 2 [22].

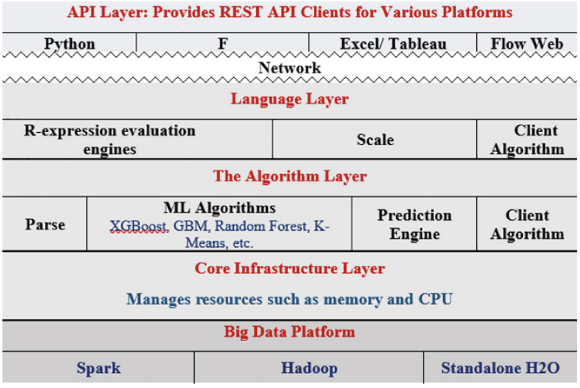


Fig. 2. H2O Architecture

2.2 TPOT AutoML Platform

TPOT is open-source software (written in Python) that automates building ML pipelines by combining expression tree representation with stochastic search algorithms. It uses the Python-based scikit-learn library as its ML menu [25, 26]. TPOT intelligently explores thousands of possible pipelines to find the best one for the data. This package uses genetic programming to help recognize the best pipeline for a specific task. Unlike H2O, TPOT discovers the best models and pipelines with preprocessors, feature selectors, and ML/DL. It provides services concerning feature engineering, model selection, and hyperparameter tuning [25, 26]. The ML Pipeline automated by TPOT is shown in Fig. 3.

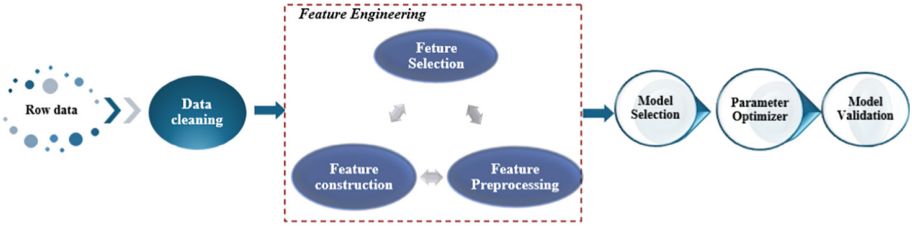


Fig. 3. ML Pipeline Automated By TPOT

3 Data Preprocessing and Feature Engineering

Machine learning often involves feature engineering and data preprocessing, which includes selecting, manipulating, and transforming raw data into features easily understood by learning algorithms. It involves handling inappropriate data, missing values, errors caused by human intervention, insufficient data sources, and other issues. Popular data preprocessing and feature engineering techniques include cleaning up datasets by fixing or removing incorrect, corrupted, duplicate, or incomplete data. The imputation technique handles irregularity in the dataset instead of removing rows or columns with missing data. It is used to fill in missing values with estimates to retain data in datasets (default values or means/medians can impute missing numerical data; for categorical data, the most frequent value in a column can replace the missing values). Datasets may contain outliers (extreme values). These outliers can significantly affect the classification accuracy. Therefore, outliers must be identified and handled before the classification phase to improve accuracy. Feature scaling (data normalization) is used to enhance model training performance and stability by adjusting the range of data. Some popular data normalization techniques include scaling to a range, clipping, log scaling, and z-score. Finally, Logarithm transformation, or log transform, is a mathematical technique used in ML to normalize skewed data and reduce the effect of outliers on the data, which makes the model more robust.

Data preprocessing and feature engineering are crucial for improving ML models and achieving more accurate predictions for real-world data. Both TPOT and H2O provide a set of services for data preprocessing and feature engineering. However, although Both H2O and TPOT integrate robust models and preprocessing techniques that can reduce the influence of the outliers, they do not explicitly detect or handle outliers as a separate step. Therefore, it is better to handle outliers manually before applying the AutoML tools. Consequently, we propose an algorithm based on Z-scores to detect and handle outliers in the original dataset.

Algorithm One: Preprocessing to filter out rows with Outliers.

1. Iterate over columns.
   a. Find Mean μ, and standard deviation δ.
   b. Let x_i be a value current column, $dx_i = |x_i - \mu|$ // deviation of each value from the mean of the column
   c. If $dx_i >$ threshold, then drop the corresponding row from the dataset # threshold is set $3 \times \delta$, which represents the cutoff for outlier

The study examines the impact of removing rows with outliers as a preprocessing step before applying TPOT and H2O and evaluates its effect on their performance.

4 Data Balancing

The dataset for detecting electricity theft is unbalanced, with a majority normal class and a minority theft class. This can cause skewness and bias issues, which can be addressed with resampling techniques. Resampling is a technique that transforms the training dataset by creating a new version with a different class distribution. The change in class distribution is only applied to the training dataset, not the test dataset. The purpose of resampling is to influence model fit, so selecting an appropriate technique for the specific problem is essential. To address the rarity of malicious samples in the used electricity consumption dataset, three oversampling data balancing techniques, namely, Random Oversampling, SMOTE, and ADASYN are experimented with to find the best method to balance the dataset, which improves the ML performance. Random oversampling is based on duplicating examples from the minority class. Although it is simple to implement and fast for large datasets, it sometimes can lead to overfitting. Random oversampling is suitable for binary and multi-class classification problems. On the other hand, SMOTE involves generating synthetic samples of the minority class by interpolating between existing minority class examples [12, 29]. Unlike random oversampling, SMOTE does not increase overfitting as it does not use copies of minority class data [17]. SMOTE and ADASYN are similar in that both of them oversample the minority class in unbalanced datasets by generating synthetic data. However, ADASYN generates synthetic samples for a given instance x_i, which are proportional to the number of neighboring samples that do not belong to the same class as x_i. This means that ADASYN focuses more on generating new examples for the minority class outliers rather than just oversampling the existing instances [16]. In this study, each AutoML platform (TPOT and H2O) is applied to both the unbalanced and balanced datasets. Their behavior will be studied by comparing their performance before and after balancing the dataset. Moreover, we will investigate which data balancing technique (Random, SMOTE, and ADASYN) is more effective in enhancing the performance of TPOT and H2O on the used Dataset.

5 Electricity Theft Predictions: The Proposed System Model

Machine learning generally comprises five phases: data preprocessing, feature engineering, Dataset balancing, learning phase, and evaluation phase. Most AutoML platforms include data preprocessing, feature engineering, algorithm selection, and model hyperparameter optimization as part of their built-in functionalities. As mentioned, this study utilizes AutoML to detect electricity theft. Two prominent platforms, TPOT and H2O, will be investigated to assess the effectiveness and accuracy of each. The study will investigate the effect of two preprocessing and feature selection algorithms, described in Sect. 3, on TPOT and H2O beyond their built-in capabilities. Furthermore, we will investigate whether TPOT and H2O have similar performance when used with imbalanced datasets related to electricity theft. Three techniques are used to understand the impact of balancing methods on TPOT and H2O's performance: Random Oversampling,

SMOTE, and ADASYN. We propose a four-phase system model to achieve the main research goals (as shown in Fig. 4).

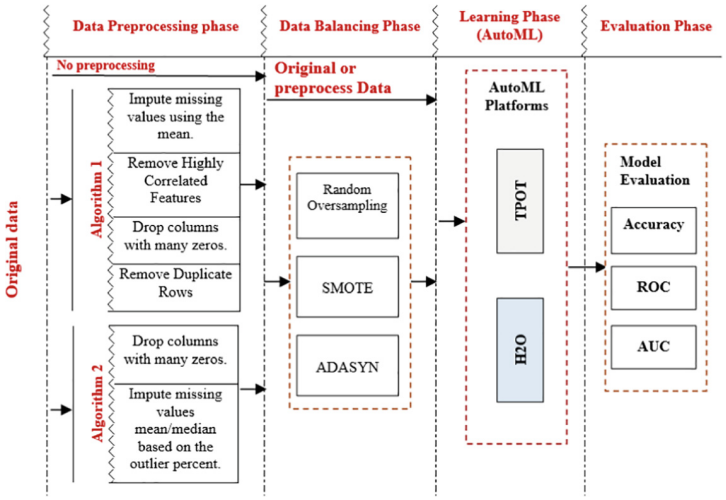


Fig. 4. ETD: The proposed system model.

The four phases are data preprocessing and feature selection, data balancing, learning, and model evaluation. Initially, the dataset is directly passed to the learning or balancing phases or undergoes two preprocessing and feature selection algorithms. These algorithms remove highly correlated features, duplicate rows, drop columns with many zeros, and impute missing values to improve the learning process of TPOT and H2O. To investigate the performance of TPOT and H2O on an imbalanced dataset and to explore whether balancing methods affect their behavior, TPOT and H2O were trained on the original data and data balanced with three methods. After completing the learning phase, the selected models from TPOT and H2O are evaluated using the testing dataset and compared based on accuracy, AUC, and ROC. The TPOT and H2O stages are illustrated in Figs. 1 and 3, respectively.

6 Experimental Results: Assessment

This research uses a real dataset containing 9956 rows and 367 columns. The first column contains alphanumeric consumer IDs, while the 2nd to 366 columns contain daily electricity consumption data. The last column, "flag," contains binary values (0 for normal and 1 indicating theft). The dataset is an unbalanced dataset that contains (8562 normal vs. 1394 theft). It also includes missing values and requires slight cleaning. The dataset can be utilized to classify electrical theft as positive or negative. Additionally, it is valuable for exploring class balancing techniques since it is an unbalanced dataset. Moreover, it can be used for time series forecasting of electrical consumption. Three scenarios were used to evaluate the developed model's performance and answer the research questions.

Scenario One (exploring whether balancing methods affect the behavior of TPOT and H2O): This investigation aims to study the impact of data balancing on the performance of H2O and TPOT. Specifically, we want to identify which method of balancing (Random, SMOTE, or ADASYN) is more effective in improving the performance of TPOT and H2O. To achieve this, we will apply TPOT and H2O to both the original and balanced datasets using the three methods.

Scenario Two (Investigating the effect of data preprocessing): This investigation aims to determine whether additional preprocessing concerning outlier handling can improve the performance of TPOT and H2O beyond their built-in capabilities. To achieve this, we will apply TPOT and H2O to the preprocessed data (using Algorithm One) before and after data balancing (using Random, SMOTE, or ADASYN).

For performance evaluation, cross-validation with 10-fold is used in both AutoML platforms (TPOT and H2O), with 30% of the dataset hold-out for testing. The best platform for detecting electricity theft will be identified based on three performance measures:

1. ***Accuracy***: is one of the metrics used in ML to measure the performance of a model (see Eq. (1)).

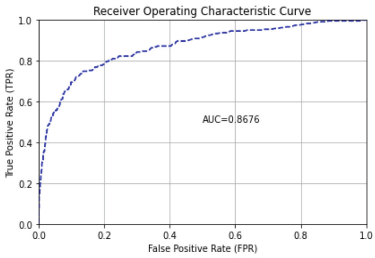
$$\text{Accuracy} = \text{Number of correct predictions}/\text{Total number of predictions} \ldots \quad (1)$$

2. ***Roc Curve***: The ROC curve plots a true positive (TP) rate against a false positive (FP) rate at different thresholds, showing a classification model's performance.
3. ***Area Under the Curve (AUC):*** is a metric that measures the effectiveness of a binary classifier in distinguishing between classes. The higher the AUC, the better the model's performance in distinguishing between the two classes.

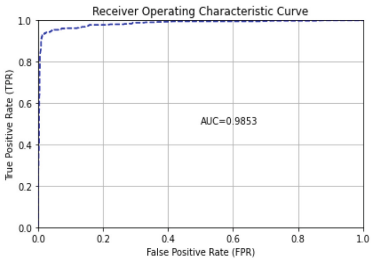
The results of the two scenarios are shown in Tables 1 and 2, while Figs. 5, 6, 7, and 8 depict the ROC and AUC results.

Tables 1 and 2 illustrate that when working with an unbalanced dataset, regardless of the scenarios, TPOT and H2O perform well with or without preprocessing. However, H2O outperforms TPOT (0.89606 and 0.7728. compared to 0.8898 and 0.7487, respectively). On the other hand, after data balancing (regarding the balancing technique), TPOT outperforms H2O in most cases. The best theft detection accuracy (0.99182 and 0.9846) is reached when applying TPOT to the original dataset balanced with Random-sampling. At the same time, the TPOT performance is comparable when applied to data balanced with SMOT and ADASYN in both scenarios. On the other hand, When Algorithm One is used to handle outliers in unbalanced data, it negatively affects the performance of H2O and TPOT, reducing accuracy to 0.7728 and 0.7487, respectively. After balancing the data, their performance returned to its average. TPOT achieves the best accuracy when applied to data balanced with random sampling (0.9846), whereas H2O performs best on data balanced with ADASYN (0.8819).

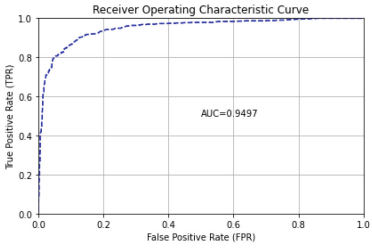
Based on the experimental results, it was discovered that the most effective pipeline is usually achieved by using ensemble machine learning. Combining multiple machine learning models using bagging, boosting, or stacking techniques. Tables 3 and 4 display the best pipeline for Scenario 1 and 2, respectively, as determined by TPOT and H2O.



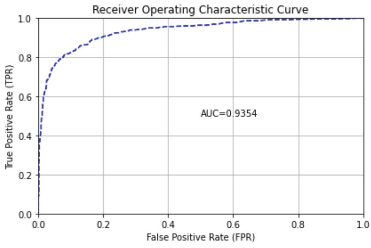
(a)Unbalanced Electricity Theft Dataset



(b)Balanced Dataset with Random OverSapmling

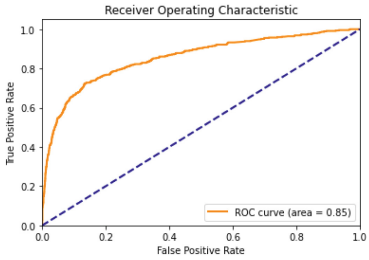


(c) Balanced Dataset with SMOTE Sampling

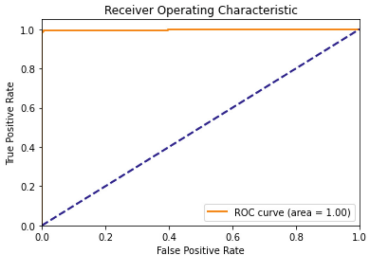


(d) Balanced Dataset with ADASYN Sampling

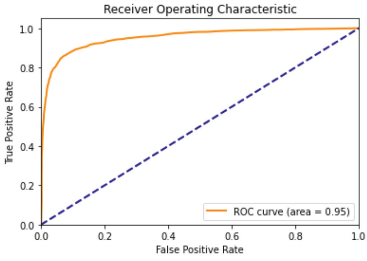
Fig. 5. ROC Curve: Apply H2O on the original dataset



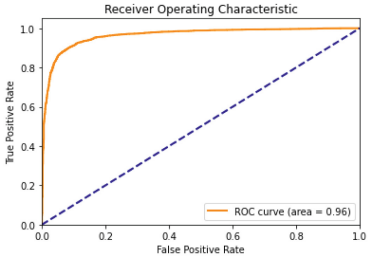
Unbalanced Electricity Theft Dataset



Balanced Dataset with Random Over Sampling

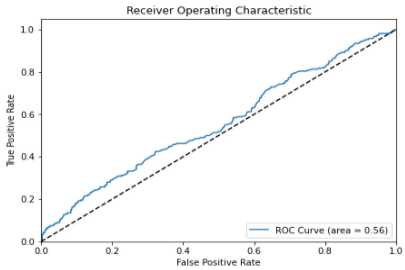


Balanced Dataset with SMOTE Sampling

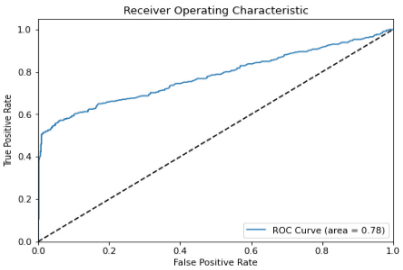


Balanced Dataset with ADASYN Sampling

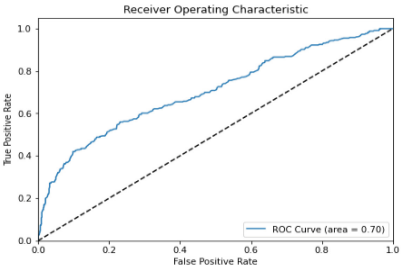
Fig. 6. ROC Curve: Apply TPOT on the original dataset.



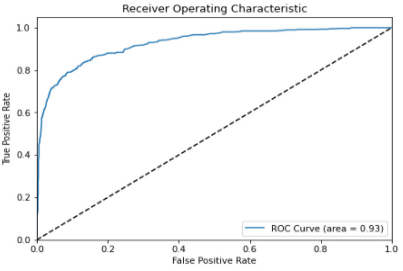
(a)Unbalanced Electricity Theft Dataset



b)Balanced with Random OverSapmling

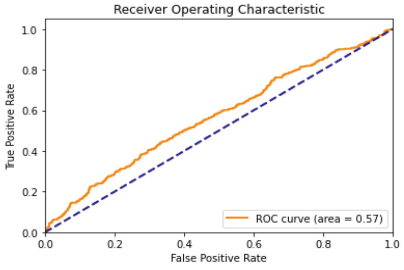


(c)Balanced with SMOTE Sampling

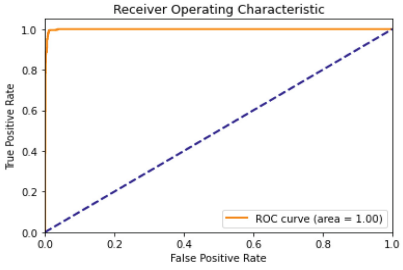


(d)Balanced with ADASYN Samplin

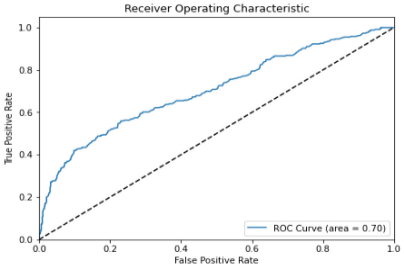
Fig. 7. ROC Curve: Apply H2O on preprocessed data with Algorithm One



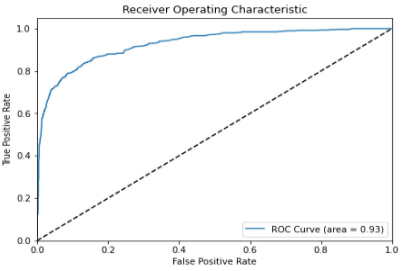
(a)Unbalanced Electricity Theft Dataset



(b)Balanced with Random OverSapmling



c)Balanced with SMOTE Sampling



(d)Balanced with ADASYN Sampling

Fig. 8. ROC Curve: Apply TPOT On preprocessed data with Algorithm One

Table 1. The accuracy of applying AutoML on the original dataset

		Balanced Sampling Methods		
AutoML	Unbalanced Data	Random Oversampling	SMOTE	ADASYN
H2O	0.8960	0.9683	0.9034	0.8918
TPOT	0.8898	0.9918	0.8923	0.90618

Table 2. The accuracy of applying AutoML to the dataset after handling the outliers with algorithm one

		Balanced Sampling Methods		
AutoML	Unbalanced Data	Random Oversampling	SMOTE	ADASYN
H2O	0.7728	0.8237	0.7433	0.8819
TPOT	0.7487	0.9846	0.9250	0.9250

Table 3. The Best pipeline For Scenario One considering the four cases

Cases	TPOT: Best pipeline	H2O
Unbalanced Dataset	GradientBoostingClassifier with GaussianNB	Stacked Ensembles All produce the best results for all cases
Random Oversampling	RandomForestClassifier with RBFSampler	
SMOTE	RandomForestClassifier with CombineDFs	
ADYSAN	GradientBoostingClassifie with PCA	

Table 4. The Best pipeline For Scenario One considering the four cases.

Cases	TPOT: Best pipeline	H2O
Unbalanced Dataset	ExtraTreesClassifier	Stacked Ensembles All produce the best results for all cases
Random Oversampling	RandomForestClassifier With PCA	
SMOTE	MLPClassifier with RobustScaler	
ADYSAN	MLPClassifier with RobustScaler	

7 Discussion and Conclusion

The objective of this study is to use two AutoML platforms, specifically H2O and TPOT, to detect electricity theft based on data collected by smart meters. The study addresses the challenges of handling outliers and unbalanced classes in the ETD dataset. The experimental results indicate that TPOT outperforms H2O in terms of accuracy for electricity theft detection on the balanced dataset, while H2O performs better on the unbalanced dataset. This is because H2O can handle complex models and datasets, while TPOT performs better in combining and selecting the best model as optimizes ML pipelines using genetic programming. Sampling methods can address the class imbalance problem, but their effectiveness depends on the dataset's nature and method. Random sampling outperforms SMOTE and ADASYN, which have shown comparable accuracy as they both create synthetic samples. However, the generated samples may not accurately represent the classes if the original data includes noise or outliers, negatively affecting classification accuracy. The variation in model performance highlights the importance of thorough data preparation in predictive modeling, especially in complex tasks like fraud or theft detection. Lastly, handling outliers by removing the data row containing at least one outlier will reduce the dataset size. Therefore, the training process will be negatively affected, leading to a reduction in classification accuracy. However, after balancing the resulting dataset, the performance of TPOT and H2O significantly improved. It's worth noting that leveraging AutoML platforms enables experts to concentrate on data analysis instead of the classification phase. As part of our upcoming work, we intend to explore different techniques to handle outliers explicitly before feeding the data to the AutoML platform.

Disclosure of Interests. The authors have no relevant competing interests to declare regarding the content of this article.

References

1. Munawar, S., et al.: Electricity theft detection in smart grids using a hybrid BiGRU–BiLSTM model with feature engineering-based preprocessing. Sensors **22**, 7818 (2022). https://doi.org/10.3390/s22207818
2. Ul Haq, E., Pei, C., Zhang, R., Jianjun, H., Ahmad, F.: Electricity-theft detection for smart grid security using smart meter data: a deep-CNN based approach. Energy Reports **9**(Supplement 1), 634–643 (2023). https://doi.org/10.1016/j.egyr.2022.11.072
3. Messinis, G.M., Hatziargyriou, N.D.: Review of non-technical loss detection methods. Electr. Power Syst. Res. **158**, 250–266 (2018)
4. Glauner, P., et al.: Identifying irregular power usage by turning predictions into holographic spatial visualizations. In: Proceedings of the 2017 IEEE International Conference on Data Mining Workshops, pp. 258–265 (2017)
5. Zhu, L., Wen, W., Li, J., Zhang, C., Zhou, B., Shuai, Z.: Deep active learning-enabled cost-effective electricity theft detection in smart grids. IEEE Trans. Indust. Inf. **20**(1), 256–268 (2024). https://doi.org/10.1109/TII.2023.3249212
6. Johncy, G., Anisha Felise, A.: An efficient power theft detection using mean-shift clustering and deep learning in smart grid. IOP Conf. Ser.: Mater. Sci. Eng. **983**, 012003 (2020). https://doi.org/10.1088/1757-899X/983/1/012003

7. Lepolesa, L.J., Achari, S., Cheng, L.: Electricity theft detection in smart grids based on deep neural network. IEEE Access **10**, 39638–39655 (2022). https://doi.org/10.1109/ACCESS.2022.3166146
8. Xia, X., Xiao, Y., Liang, W., Cui, J.: Detection methods in smart meters for electricity thefts: a survey. Proc. IEEE **110**(2), 273–319 (2022). https://doi.org/10.1109/JPROC.2021.3139754
9. Ul Haq, E., Huang, J., Xu, H., Li, K., Ahmad, F.: A hybrid approach based on deep learning and support vector machine for the detection of electricity theft in power grids. Energy Rep. **7**(6), 349–356 (2021). https://doi.org/10.1016/j.egyr.2021.08.038
10. Badr, M.M., Ibrahem, M.I., Kholidy, H.A., Fouda, M.M., Ismail, M.: Review of the data-driven methods for electricity fraud detection in smart metering systems. Energies **16**, 2852 (2023). https://doi.org/10.3390/en16062852
11. Samawi, V.W., Yousif, S.A., Al-Saidi, N.M.G.: Intrusion detection system: an automatic machine learning algorithms using auto-WEKA. In: 2022 IEEE 13th Control and System Graduate Research Colloquium (ICSGRC), Shah Alam, Malaysia, pp. 42–46 (2022). https://doi.org/10.1109/ICSGRC55096.2022.9845166
12. Khan, I.U., Javeid, N., Taylor, C.J., Gamage, K.A.A., Ma, X.: A stacked machine and deep learning-based approach for analysing electricity theft in smart grids. IEEE Trans. Smart Grid **13**(2), 1633–1644 (2022). https://doi.org/10.1109/TSG.2021.3134018
13. Yousif, S.A., Samawi, V.W., Al-Saidi, N.M.G.: Automatic machine learning classification algorithms for stability detection of smart grid. In: 2022 IEEE 5th International Conference on Big Data and Artificial Intelligence (BDAI), Fuzhou, pp. 34–39 (2022). https://doi.org/10.1109/BDAI56143.2022.9862710
14. Punmiya, R., Choe, S.: Energy theft detection using gradient boosting theft detector with eature engineering-based preprocessing. IEEE Trans. Smart Grid **10**(2), 2326–2329 (2019). https://doi.org/10.1109/TSG.2019.2892595
15. Mumuni, A., Mumuni, F.: Automated data processing and feature engineering for deep learning and big data applications: a survey. J. Inf. Intell. (2024). https://doi.org/10.1016/j.jiixd.2024.01.002
16. Ali, A., Khan, L., Javaid, N., Aslam, M., Aldegheishem, A., Alrajeh, N.: Exploiting machine learning to tackle peculiar consumption of electricity in power grids: a step towards building green smart cities. IET Gen. Transmis. Distrib. **18**(3), 413–445 (2024)
17. García, V., Sánchez, J.S., Marqués, A.I., Florencia, R., Rivera, G.: Understanding the apparent superiority of over-sampling through an analysis of local information for class-imbalanced data. Exp. Syst. Appl. **158**, 113026 (2020). https://doi.org/10.1016/j.eswa.2019.113026
18. Moatz, N., Yousif, S.A.: COVID-19 detection via blood tests using an automated machine learning tool (auto-sklearn). Iraqi J. Sci. 6013–6024 (2023)
19. Feurer, M., Klein, A., Eggensperger, K., Springenberg, J.T., Blum, M., Hutter, F.: Auto-sklearn: efficient and robust automated machine learning. In: Hutter, F., Kotthoff, L., Vanschoren, J. (eds.) Automated Machine Learning. The Springer Series on Challenges in Machine Learning. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-05318-5_6
20. What Is Automated Machine Learning (AutoML)? Run.ai (2023). https://www.run.ai/guides/automl
21. Matti, R., Yousif, S.A.: AutoKeras for fake news identification in Arabic: leveraging deep learning with an extensive dataset. Al-Nahrain J. Sci. **26**(3), 60–66 (2023)
22. H2O AutoML. Automatic Machine Learning, H2O.ai (2023). https://docs.h2o.ai/h2o/latest-stable/h2o-docs/welcome.html
23. Schmitt, M.: Automated machine learning: AI-driven decision making in business analytics. Intell. Syst. Appl. **18**, 200188 (2023). https://doi.org/10.1016/j.iswa.2023.200188

24. Sohn, A., Olson, R.S., Moore, J.H.: Toward the automated analysis of complex diseases in genome-wide association studies using genetic programming. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2017), pp: 489–496. ACM Digital Library, Berlin (2017). https://doi.org/10.1145/3071178.3071212
25. Moatz, N., Yousif, S.A.: An automated machine learning model for diagnosing coronavirus disease 2019 (COVID-19) infection. IAES Int. J. Artif. Intell. **12**(3), 1360 (2023)
26. Le, T.T., Weixuan, F., Moore, J.H.: Scaling tree-based automated machine learning to biomedical big data with a feature set selector. Bioinformatics **36**(1), 250–256 (2020)
27. LeDell, E., Poirier, S.: H2O AutoML: scalable automatic machine learning. In: 7th ICML Workshop on Automated Machine Learning (AutoML) (2020). https://www.automl.org/wp-content/uploads/2020/07/AutoML_2020_paper_61.pdf
28. Gooday, A.: Automated Machine Learning: An Introduction to AutoML. Google, Amazon & H2O.ai AutoML (2020). https://medium.com/analytics-vidhya/an-introduction-to-automl-8356b6ceb091
29. Qu, Z., Li, H., Wang, Y., Zhang, J., Abu-Siada, A., Yao, Y.: Detection of electricity theft behavior based on improved synthetic minority oversampling technique and random forest classifier. Energies **13**, 2039 (2020). https://doi.org/10.3390/en13082039