Heliyon 10 (2024) e35167

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Research article

Deep learning-based electricity theft prediction in non-smart grid environments

Sheikh Muhammad Saqib [a], Tehseen Mazhar [b,*], Muhammad Iqbal [a], Tariq Shahazad [c,**], Ahmad Almogren [d], Khmaies Ouahada [c], Habib Hamam [c,e,f,g]

[a] Department of Computing and Information Technology, Gomal University, Dera Ismail Khan, Pakistan
[b] Department of Computer Science, Virtual University of Pakistan, Lahore, 51000, Pakistan
[c] School of Electrical Engineering, Dept. of Electrical and Electronic Eng. Science, University of Johannesburg, Johannesburg, 2006, South Africa
[d] Department of Computer Science, College of Computer and Information Sciences, King Saud University, Riyadh, 11633, Saudi Arabia
[e] Faculty of Engineering, Université de Moncton , Moncton, NB, E1A3E9, Canada
[f] Hodmas University College, Taleh Area, Mogadishu, Banadir, 521376, Somalia
[g] Bridges for Academic Excellence, Tunis, Centre-Ville, 1002, Tunisia

ARTICLE INFO

Keywords:
Deep learning
Feature engineering
Principal Component Analysis (PCA)
t-distributed Stochastic Neighbor Embedding (t-SNE)
Random-Under-Sampler (RUS)
Synthetic Minority Over-Sampling Technique (SMOTE)
Random-Over-Sampler (ROS)

ABSTRACT

In developing countries, smart grids are nonexistent, and electricity theft significantly hampers power supply. This research introduces a lightweight deep-learning model using monthly customer readings as input data. By employing careful direct and indirect feature engineering techniques, including Principal Component Analysis (PCA), t-distributed Stochastic Neighbor Embedding (t-SNE), UMAP (Uniform Manifold Approximation and Projection), and resampling methods such as Random-Under-Sampler (RUS), Synthetic Minority Over-sampling Technique (SMOTE), and Random-Over-Sampler (ROS), an effective solution is proposed. Previous studies indicate that models achieve high precision, recall, and F1 score for the non-theft (0) class, but perform poorly, even achieving 0 %, for the theft (1) class. Through parameter tuning and employing Random-Over-Sampler (ROS), significant improvements in accuracy, precision (89 %), recall (94 %), and F1 score (91 %) for the theft (1) class are achieved. The results demonstrate that the proposed model outperforms existing methods, showcasing its efficacy in detecting electricity theft in non-smart grid environments.

1. Introduction

1.1. Context and motivation

Electricity theft remains a pervasive challenge for utility providers globally, resulting in substantial financial losses exceeding $96 billion annually attributed to Non-Technical Losses (NTLs), with electricity theft as the primary culprit. Sub-Saharan Africa, in particular, grapples with significant theft, where approximately 50 % of generated energy is reported stolen, according to the World Bank [1].

* Corresponding author. Department of Computer Science, Virtual University of Pakistan, Lahore, 51000, Pakistan.
** Corresponding author.
E-mail addresses: tehseenmazhar719@gmail.com (T. Mazhar), tariqshahzadd@gmail.com (T. Shahazad).

https://doi.org/10.1016/j.heliyon.2024.e35167
Received 18 May 2024; Received in revised form 23 July 2024; Accepted 24 July 2024
Available online 26 July 2024

Electricity losses are typically classified into two categories: energy delivered to customers but not paid for (termed as unpaid energy), and losses occurring within transmission and distribution lines, which are inherent to electricity transmission. Non-technical losses constitute the majority of losses in electricity networks and can exceed 40 % of the total electricity produced [2]. These losses originate from various sources, with the primary contributors being metering equipment tampering, illegal connections to the electrical grid, and energy theft [3].

Electricity theft poses several significant risks, including increased electricity demand, strain on electrical systems, and substantial revenue loss for power companies, and threats to public safety such as fires and electric shocks. For instance, according to Ref. [4], approximately 100 million Canadian dollars are lost each year due to electricity theft. This amount of lost electricity could power approximately 77,000 homes for a year.

While Smart grids may be subject to cybersecurity attacks [5], they address the issue of electricity theft by integrating power grids with intelligent devices that communicate with smart meters and sensors to manage grid operations [6]. However, in countries where smart power grids are either absent or only partially implemented, the risk of electricity theft remains high. The primary objective of the proposed model is to develop a system that can be trained using monthly consumption data from customers to identify potential instances of electricity theft at electricity distribution points where meter readings are collected.

Currently, a prevalent trend involves the utilization of machine learning and deep learning methodologies for predictive tasks, including the detection of electricity theft. While a substantial portion of research focuses on employing machine learning and deep learning techniques for this purpose, the inherent characteristics of the datasets pose challenges. Despite achieving high accuracy rates in many cases, the precision and recall rates specifically concerning the identification of instances related to theft are often notably lower [7,8].

The objective of this study is to develop a streamlined deep learning model with minimal computational complexity yet yielding significant performance outcomes. To achieve this, a lightweight deep learning framework is proposed, integrating various techniques such as direct and indirect feature engineering, Principal Component Analysis (PCA), t-distributed Stochastic Neighbor Embedding (t-SNE), and resampling methods including Random-Under-Sampler (RUS), Synthetic Minority Over-sampling Technique (SMOTE), and Random-Over-Sampler (ROS). A comprehensive evaluation of these techniques within the context of the lightweight deep learning model is conducted, aiming to identify the optimal approach for addressing the research benchmarks.

An important issue to address is the improvement in precision and recall degrees for identifying the events connected to electric theft although this area has already broad research spaces for applying machine learning and deep learning techniques to predictive tasks. This gap highlights why there is a need to come up with more efficient procedures to deal with theft detection systems challenges resulting due to intrinsic attributes of the datasets. Ultimately, the accuracy of the theft detection systems will be enhanced.

1.2. Research contribution

This research makes several significant contributions to the field of electricity theft detection in non-smart grid environments. Firstly, it introduces a unique and simple deep learning model specifically designed to recognize cases of electricity theft through customers' monthly readings. This model is tailored to be computationally efficient, making it suitable for deployment in systems with limited resources, which are common in developing nations.

Secondly, the study develops a novel approach in feature engineering by incorporating both direct and indirect methods into the model development to enhance its predictive power. This not only improves the accuracy of theft detection but also contributes to deep learning by demonstrating how different feature engineering techniques can be effectively integrated.

Furthermore, dimensionality reduction and resampling techniques, including PCA, t-SNE, RUS, SMOTE, and ROS, are examined. The results allow us to predict how much these techniques influence precision and recall values when identifying cases of illegal energy consumption. In terms of accuracy, precision, and recall, this study conducts a thorough evaluation that attests to the superiority of the designed system over current methods, making it a reliable means to reveal illegal consumption in non-smart grid environments. This is particularly relevant in addressing one of the challenging issues faced by power suppliers in many developing nations.

Finally, this research demonstrates the practical application of deep learning methods in real-life challenges. By providing an effective solution to the pervasive problem of electricity theft, this study paves the way for further research in similar areas where deep learning can address significant problems under resource limitations.

1.3. Structure of the article

This article is organized into several key sections to systematically present the research findings and contributions. Following this introduction, Section 2 presents a literature review, highlighting current research efforts and identifying the gaps that this study aims to fill. Section 3 outlines the proposed methodology, detailing the development of the lightweight deep learning model, the dataset preparation, feature engineering techniques, and the evaluation methods used. Section 4 presents the results of the study, offering a comparative analysis of the model's performance against existing methods and discussing the significance of the findings. Finally, Section 5 concludes the article with a recapitulation of the research contributions and suggests directions for future work in the area of electricity theft detection in non-smart grid environments. This structure is designed to provide a coherent flow of information, from identifying the problem and reviewing existing solutions to presenting a novel approach and discussing its implications for both theory and practice.

2. Literature review

2.1. Current research

Different machine learning techniques and deep learning work on large datasets and can draw useful conclusions. Machine learning models run on various algorithms to accurately predict the presence or absence of predicted classes. Deep learning and machine learning models are utilized across various domains for early prediction purposes, including text mining [9], health protocoling [10], disease forecasting [11], diabetic retinopathy [12], tumor classification [13], agriculture [14], smart vehicles [15], Smart Energy and Smart Buildings Management [16], and education [17].

The study of [18] addressed the problem of electricity theft in power grids using a combination of CNN and Random Forest (RF) techniques. The dataset used was from Ireland and SEAI. The model's performance was evaluated using metrics such as Precision, AUC, Recall, and F1 Score. However, the study identified privacy as a limitation, suggesting a need for further enhancement in securing user data.

The research in Ref. [19] focused on electricity theft using datasets from various random areas, employing temperature-dependent theft detection using load monitoring (TDTLM). While the approach provided insights into theft detection, the study neglected to emphasize performance metrics, making it challenging to assess its efficacy thoroughly. Despite this, the temperature-dependent method showed superior outcomes, indicating its potential effectiveness in detecting electricity theft using smart meter data.

The study in Ref. [20] employed Long Short-Term Memory (LSTM) and a bat-based random under-sampling boosting method using the SGCC dataset. Significant metrics including F1 score, precision, recall, and ROC-AUC were achieved. Despite these promising results, the study highlighted a lack of robustness in the system, indicating areas for improvement. This model aimed to enhance unbalanced data, parameter optimization, and overfitting issues, making it applicable to both commercial and residential electricity information.

In addressing electricity theft detection with concerns about the curse of dimensionality and overfitting, the study [21] used SMOTE on the SEAI dataset. The study focused on metrics such as DR, FPR, Time Complexity, and Recall but faced issues with overfitting and privacy leakage due to the high sampling rate. This approach tackled the problem of overfitting by utilizing the Synthetic Minority Over-sampling Technique (SMOTE), although it highlighted the need for improved handling of privacy issues.

Using the SGCC dataset, the study in Ref. [22] applied Tomek Links, AlexNet, and Peephole techniques to detect electricity theft in smart grids. It reported high performance in PR-AUC, accuracy, precision, recall, F1-score, and AUC. However, it mainly considered low sampling data, limiting its broader applicability. The use of synthetic monitoring samples and techniques like Tomek and Peephole helped in addressing the electricity problem in smart grids, and the model was recommended for future use to reduce power losses.

In 2018, broad CNNs were applied to analyze one-dimensional data, while deep CNNs were employed for two-dimensional data. Specifically, the one-dimensional data were transformed into two-dimensional representations of electricity consumption data [8] and achieved an accuracy is 78 % on Wide and Deep CNN. Conversely, another investigation involved the utilization of a Support Vector Machine (SVM), which utilized customer consumption data alongside the total energy distributed by the supplier. This approach facilitated the computation of errors arising from electricity meter readings [23].

In the commercial area of Brazil, the research in Ref. [24] used the Binary Hole Algorithm (BHA) and Optimal Power Flow (OPF). The study achieved mean accuracy but faced challenges with a biased dataset and inappropriate performance metrics, suggesting a need for more balanced data and relevant metrics. This study offered a comparison of deep neural network technology for electricity theft detection, supplying Recall, F1 score, and AUC, and proposed further investigation into other supervised learning algorithms.

The study of [25] focused on the SGCC dataset and applied a deep artificial neural network for electricity theft detection. It reported Recall, F1 score, and AUC as performance metrics, recommending experimentation with other supervised learning algorithms to improve results. The finite mixture model was employed alongside the gradient boosting machine method, clustering, and evolutionary genetic algorithms, enhancing the handling of attack circumstances and suggesting future use in utility corporations.

Using datasets from Ireland, the study of [26] combined gradient-boosting machine algorithms with clustering and evolutionary genetic algorithms. The reported metrics were accuracy, F1 score, AUC, and precision. However, it did not address the imbalanced nature of the data adequately, indicating a potential area for future work. The gradient boosting machine algorithm was used to enhance detection capabilities, but the study emphasized the need for better data balance management.

In the study of [27], an SVM with Kernel-PCA was employed. However, this approach necessitates manual feature engineering or selection, which can be time-intensive and may not consistently capture the most pertinent information from the data. Despite achieving an accuracy of 89 %, precision of 85 %, and recall of 88 %, it's important to note the potential limitations of this method.

A Privacy-Preserving Electricity Theft Detection Scheme was utilized in the study [28], focusing on Load Monitoring and Billing for AMI Networks. Using a real-time dataset, the study reported ROC-AUC and accuracy but indicated that the security features resulted in a slightly lower detection rate, suggesting a trade-off between security and performance. This approach was particularly notable for its emphasis on privacy-preserving techniques to detect fraudulent clients.

Employing SVM on a Malaysian dataset, the study [29] reported accuracy as the primary metric but faced limitations with appropriate metrics selection, highlighting a need for better evaluation criteria in future research. The use of SVM provided a robust framework for theft detection, though it underscored the importance of selecting suitable performance metrics.

The study in Ref. [30] used a feature-engineered CatBoost algorithm combined with SMOTETomek on the SGCC dataset. It achieved accuracy, recall, and precision but neglected improvements in system robustness, suggesting a potential area for enhancement. The SMOTETomek technique helped manage data imbalances, while CatBoost provided effective feature categorization, although further robustness improvements were needed.

In developing the Smart Energy Theft System (SETS), the study [31] combined MLP, RNN, LSTM, and GRU. The proposed technique demonstrated better accuracy and applicability in industrial and commercial sectors, indicating its potential for wider adoption. This system incorporated various machine learning models, including the Simple Moving Average (SMA) statistical model, to enhance detection accuracy and applicability.

Utilizing XG-Boost on the Endesa dataset, the study [32] focused on TPR, Recall, FPR, Precision, and AUC. The model, however, consumed high processing time on large datasets, suggesting a need for optimization in handling large-scale data. The use of XG-Boost provided a comprehensive analysis of electricity theft, though it highlighted the need for time-efficient processing methods.

The study in Ref. [33] applied XGBoost on an Irish dataset, reporting FPR, Recall, AUC, and Precision as metrics. It faced challenges with limited training data, imbalanced data, and constrained results, highlighting areas for further research. The limited dataset size affected the model's performance, indicating the need for more extensive training data.

Using the CSS for the ANN-MLP method on a Brazilian dataset, the study [34] employed PSO, SGHS, and BP. The proposed model did not adequately handle the imbalanced nature of data, suggesting a need for better balancing techniques in future studies. The application of ANN-MLP highlighted the challenges of data imbalance, requiring more effective balancing methods.

The research in Ref. [35] also used ANN-MLP on a Brazilian dataset, focusing on accuracy, precision, and recall. However, it noted that the results of the proposed model were not sufficiently accurate, indicating a need for model refinement. The study emphasized the necessity of improving model accuracy for effective theft detection.

Applying SVM to the SEAI dataset, the study [36] focused on DR and FPR but neglected accuracy, limiting its comprehensive evaluation, and suggesting a need for more balanced performance metrics. The use of SVM highlighted the importance of considering a broader range of metrics for a thorough assessment.

The study [37] used CNN-LSTM on the SGCC dataset and reported MCC and F1 scores. It faced issues with high processing time on datasets, indicating a need for more efficient algorithms to handle large data sets. The combination of CNN and LSTM provided robust theft detection capabilities but required optimization for processing efficiency.

In a Turkish shopping mall context, the research in Ref. [38] used an ensemble model combining LR, RF, and KNN. The reported metrics included TPR, FPR, F-measure, and precision but neglected the balance of TPR and FPR in the proposed work, suggesting an area for improvement. The ensemble model demonstrated potential for theft detection but required better metric balance.

Addressing issues of low accuracy, overfitting, and high FPR in electricity theft detection, the study [39] used an LSTM-based model on a self-made dataset. It achieved precision, recall, F1 score, and convergence speed but was not suitable for large datasets, indicating a need for scalability in future research. The LSTM model highlighted the challenges of overfitting and the necessity for scalable solutions.

There was a study done in 2021 that stood as a benchmark for the assessment of different classification algorithms. Among these, primary interest turned to light-GBM, an algorithm based on decision trees that has shown an 84 % accuracy rate [40]. The novel algorithm was also compared to some of the ordinary algorithms such as logistic regression, which had an accuracy of 71 %, stochastic gradient descent with an accuracy of 65 %, and the decision tree had an 86 % accuracy.

In [41], researchers evaluated 23 classifiers, employing the F1 score as the performance metric. Utilizing data from a Brazilian company focused on the electric power sector, encompassing 261,489 consumers and around 1400 attributes, they determined that ensemble methods, notably classifiers, are best suited for identifying non-technical instances of electric power loss. The gradient-boosted tree yielded an F1 score of 0.45 and achieved a 66.50 % accuracy in comparison to field inspections, outperforming the rotation forest classifier.

This study addressed in Ref. [42] the widespread issue of electricity theft, recognized as a non-technical loss, which adversely impacts electric distribution companies and consumers, leading to severe repercussions such as fires and power outages. The research focused on identifying the most effective prediction model using Machine Learning to combat electrical energy theft. Data from 42,372 consumers sourced from the State Grid Corporation of China served as the basis for analysis. Employing data imputation and feature extraction, efforts were made to enhance energy theft detection. Five Machine Learning models were evaluated, with the SVM model demonstrating the highest accuracy at 81 %, followed by K-Nearest Neighbors at 79 %, Random Forest at 80 %, Logistic Regression at 69 %, and Naive Bayes at 68 %. Thus, it is concluded that the SVM model outperforms others, offering the most reliable performance with an accuracy of 81 %.

Work [43] introduces a novel method for detecting electricity theft using ensemble learning and prototype learning techniques, demonstrating exceptional performance even on imbalanced datasets and diverse abnormal data. Leveraging convolutional neural network (CNN) and long short-term memory (LSTM) models, abstract features are extracted from electricity consumption data, resulting in an impressive accuracy of 89 %.

This study of [44] introduces a hybrid Multi-Layer Perceptron (MLP) with a Gated Recurrent Units (GRU) approach to address challenges associated with conventional Electricity Theft Detection (ETD) models, achieving notable accuracy, precision, recall, and F1-score of 81 %, 89 %, 82 %, and 85 % respectively on 25 % test data.

The author in Ref. [45] proposed an optimal scheduling model for isolated microgrids using automated reinforcement learning-based multi-period forecasting of renewable power generations and loads. This approach involved a prioritized experience replay automated reinforcement learning (PER-AutoRL) to simplify deployment, a single-step multi-period forecasting method based on PER-AutoRL, and a scheduling model considering demand response to minimize total microgrid operating costs. Simulation results demonstrated significant reductions in system operating costs by improving prediction accuracy, with privacy being a potential area for future improvement.

In [46], the proposed methodology was an ensemble machine learning (ML) model for detecting energy theft in smart grids using customers' consumption patterns. Several algorithms, including adaptive boosting, categorical boosting, extreme boosting, random

forest, and extra trees, were tested to find their false positive and detection rates. An extensive analysis based on a practical dataset of 5000 customers revealed that bagging models outperformed other algorithms, with the random forest and extra trees models achieving the highest area under the curve score of 0.90. The precision analysis showed that the proposed bagging methods perform better, indicating their effectiveness in electricity theft detection.

The paper [47] presented a hybrid deep neural network model combining a convolutional neural network, particle swarm optimization, and gated recurrent units. It aimed to perform accurate electricity theft detection and overcome issues in existing models. The proposed model was evaluated by performing simulations in terms of accuracy, the area under the curve, F1 score, recall, and precision. The results indicated that the proposed hybrid deep neural network model is more efficient in handling class imbalance issues and performing electricity theft detection.

The proposed model in Ref. [48] maintained the role of electricity theft detection considering cost-efficiency in smart grids and handling large electricity consumption datasets. Researchers used three modules: data imputation, outlier handling, normalization, and class balancing algorithms, three different machine learning (ML) methods, and a temporal convolutional network (TCN). Experimental results confirmed that the proposed framework yields a highly accurate, robust classification performance compared to other well-established machine and deep learning models.

The paper [49] compared three gradient-boosting machines for electricity theft detection: extreme gradient boosting, light gradient boosting machine, and cat boosting. It conducted experiments on a realistic dataset released by the State Grid Corporation of China with true malicious samples. Experimental results showed that gradient-boosting machines outperformed wide and deep convolutional neural networks for electricity theft detection, highlighting the effectiveness of boosting algorithms.

An ensemble model for electricity theft detection based on genetic optimization was developed in Ref. [50]. Synthetic samples were prepared through SMOTE, features of anomalous electricity consumption were extracted through PCA, and an ensemble deep learning network based on AdaBoost was established to mine implicit information in continuous time series data. The hyperparameters of the deep neural network were optimized based on a genetic algorithm. The results showed that the model is superior to other detection methods in terms of sensitivity and AUC.

The paper [51] proposed a hybrid method combining an adaptive boosting algorithm (AdaBoost) and convolutional neural networks (CNN) for electricity theft detection. Multiple CNN-based classifiers were trained to extract different features from the electricity consumption data, and AdaBoost combined them into a strong classifier based on their performance. Experimental results based on the Irish Smart Energy Trial showed the hybrid classifier had better performance than other conventional data-driven methods in electricity theft detection.

The research in Ref. [52] focused on the Irish smart energy trail utilizing the XGBoost methodology, which has great accuracy and resilience. However, the study faced limitations such as restricted data collection and outcomes. The costs sustained in Brazil due to electricity theft are significant, with commercial losses reaching $4 billion in 2011. The authors employed the Binary Black Hole Algorithm (BBHA) to address this issue. Regarding precise non-technical loss (NTL) identification and execution speed, the method outperformed current optimization strategies like genetic and particle swarm optimization techniques. However, reliable performance metrics like recall and accuracy were not used to evaluate the model.

A trustworthy assessment metric is critical for measuring model performance in an unbalanced data classification challenge. For IoT-based smart houses, the research in Ref. [53] produced a revolutionary supervised machine learning-based theft detection technique. The suggested model combined SMOTETomek with a feature-engineered CatBoost algorithm. The SMOTETomek approach, which concurrently over- and under-samples the data classes, was employed to prevent data class imbalances. The CatBoost algorithm's intelligence categorizes data into real and fraudulent consumers, and the results showed the model achieved accuracy, recall, and precision.

This research [54] highlights the significance of smart meters in a smart grid system and emphasizes the potential of machine learning and deep learning approaches for analyzing energy consumption behavior and detecting theft in smart meter data. The proposed theft detection dataset (TDD2022) and the machine learning-based solution provide valuable resources for automated theft identification in the smart grid, offering a benchmark for comparative studies and demonstrating the effectiveness of the random forest model in achieving improved performance metrics by 10 % or more compared to other models.

In this research [55], authors emphasize the significance of non-technical loss (NTL) detection in the context of electricity theft, which poses challenges for distribution network operators (DNOs) and affects the quality of the supply. The introduction of a new data set, incorporating location information of missing values, coupled with a neural network model built through neural architecture search (NAS), demonstrates promising results with an excellent AUC value of approximately 0.926. The use of NAS enables automatic model updates, making it a user-friendly tool for engineers without expertise in neural networks, as highlighted by the case study employing Density-Based Spatial Clustering of Applications with Noise (DBSCAN) for missing value pattern analysis.

The authors highlight the issue of non-technical loss (NTL) in the electricity grid system in their research paper [56], emphasizing the threat it poses to sustainability and stability. The proposed approach utilizing deep reinforcement learning (DRL) addresses the challenge of imbalanced electricity usage datasets and eliminates the need for extensive pre-processing or dataset balancing. The simulation results demonstrate the superiority of the proposed method, outperforming conventional algorithms in detecting NTL across different simulation environments. Non-technical losses (NTLs) pose significant challenges to the electricity distribution system in developing countries, impacting its quality and creating economic issues. Despite regulatory advancements in Brazil, the high levels of unbilled electricity consumption persist, affecting tariffs, investment capacity, and public policy development. This research paper [39] emphasizes the need for coordinated strategic actions, including a cultural shift in attitudes towards electricity theft, and aims to provide valuable insights to regulatory authorities, government, concessionaires, and researchers to develop practical solutions for mitigating NTLs in Brazil.

In addition, feature engineering using structured query language (SQL) analytic functions was implemented in a study by Oprea and Bâra (2022) to detect electricity fraud [57]. They proposed an extensive feature engineering approach using SQL analytic functions to enhance the detection of irregularities in consumption, highlighting the importance of combining classifiers with an anomaly detection feature obtained with an unsupervised ML algorithm—Isolation Forest [57]. Our proposed approach builds on these findings by incorporating feature engineering techniques such as SQL analytics, aggregation, merging datasets, and anomaly detection. This approach significantly improves the classification scores and offers a salient tool for utility companies to identify suspicious consumers, thereby reducing the costs related to periodic on-site investigations and non-technical losses.

2.2. Research gap

Despite the extensive application of machine learning and deep learning techniques in predicting and detecting electricity theft, as highlighted in sections 1.1 and 2.1, a significant research gap exists in developing efficient models suitable for non-smart grid environments, particularly in developing countries. The main challenges in these regions include the absence of smart grid infrastructure, limited computational resources, and unique patterns of electricity usage that differ significantly from those in more developed areas. Previous studies have focused on applying complex models that, while effective in certain scenarios, do not address these unique challenges, often requiring extensive computational power and data not available in developing countries.

Furthermore, the literature review reveals that while existing models demonstrate considerable success in electricity theft detection, they often suffer from low precision and recall rates, especially in identifying actual instances of theft. This limitation is significant because it indicates a high rate of false positives or negatives, which can lead to unnecessary investigations or missed theft instances, respectively. This inefficiency poses a critical problem for utility companies, as it directly impacts their financial stability and the reliability of electricity supply to consumers.

The absence of models that can manage imbalanced datasets, a prevalent problem in power theft detection where theft occurrences are far less frequent than authorized usage, is another serious gap that has been found. Despite having excellent overall accuracy rates, many present techniques are unable to correctly identify theft cases within these datasets, which results in models that miss a significant percentage of real stolen events.

This research addresses these gaps by introducing a lightweight deep learning model specifically designed for environments lacking smart grid technology. By focusing on monthly customer readings and employing a strategic combination of direct and indirect feature engineering, dimensionality reduction, and resampling techniques, the proposed model not only caters to the computational limitations of developing countries but also significantly improves precision and recall rates for detecting electricity theft. The incorporation of PCA, t-SNE, RUS, SMOTE, and ROS techniques into the model architecture specifically aims to tackle the challenge of imbalanced datasets, enhancing the model's ability to accurately identify theft instances without excessively increasing false positives or negatives.

Thus, the contributions outlined in section 1.2 are crucial for filling the identified research gap, presenting a novel approach that not only surpasses the performance limitations of existing models but also aligns with the practical realities faced by utility providers in non-smart grid environments. This research, therefore, not only advances the academic discourse on electricity theft detection but also provides a tangible solution with significant potential for real-world application, especially in developing countries grappling with the pervasive challenge of electricity theft.

3. Proposed methodology

The proposed methodology encompasses a lightweight deep learning model characterized by a reduced number of layers. Before delving into the model development process, it is imperative to address dataset cleansing by handling null entries and subsequent scaling to normalize the dataset, as depicted in Fig. 1. Following a comprehensive understanding and preprocessing of the dataset, the exploration of the model architecture is depicted in Fig. 3.

3.1. Dataset collection

The dataset is obtained from the [58] site which has consumption records of 42373 customers. For every client, there has been electricity use of 1034 days from year 2014–2016.

The data is based on the daily level of consumed electric energy produced by the Power Grid Corporation of China, created on December 29, 2002, and serving more than 1.1 billion people covering 88 % of the national regions [7].

The 'Flag' column contains values of either 0, indicating customers not involved in electricity theft, or 1, indicating involvement in

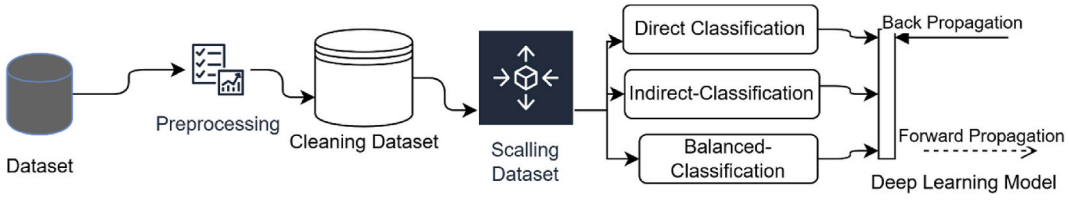

Fig. 1. Proposed work.

electricity theft. All columns are depicted in Table 1, with 'N/A' representing missing values. In total, there are 1034 records in the dataset. To provide an overview, we display the first five and last five records in Table 1.

After filling in missing values using the mean function, the consumption data of all customers involved in electricity theft and those not involved in electricity theft is depicted in Fig. 2 from 2014 to 2016.

All images in Fig. 2 illustrate that while there are 100 % reliable customers, approximately 40 % of them are classified as unreliable customers. This indicates that the data is imbalanced.

3.2. Feature engineering

Feature engineering in the context of an electricity dataset involves several crucial steps aimed at enhancing data quality and optimizing it for effective machine learning modeling. Initially, the process typically begins with identifying and handling missing values within the dataset to ensure completeness and reliability. This may involve techniques such as mean imputation for numerical features or mode imputation for categorical features. Following this, numerical features are often scaled to a consistent range using methods like StandardScaler, which standardizes data to have zero mean and unit variance. This scaling prevents features with larger numeric ranges from dominating those with smaller ranges during model training. In addition to scaling, dimensionality reduction techniques such as Principal Component Analysis (PCA), t-Distributed Stochastic Neighbor Embedding (t-SNE), and Uniform Manifold Approximation and Projection (UMAP) are applied. PCA transforms high-dimensional data into a lower-dimensional representation, facilitating easier visualization and potentially improving model performance by focusing on the most informative features. t-SNE is nonlinear and UMAP is mixed mode dimensionality reduction techniques particularly useful for capturing complex relationships within the data that may not be linearly separable. Moreover, addressing class imbalance, which is common in many real-world datasets including electricity datasets, is crucial. Techniques like Random Over-Sampling (ROS), Random Under-Sampling (RUS), and Synthetic Minority Over-sampling Technique (SMOTE) are employed to balance the class distribution. ROS randomly replicates instances from the minority class, RUS randomly removes instances from the majority class, and SMOTE generates synthetic samples for the minority class based on nearest neighbors. These techniques ensure that machine learning models trained on the dataset are not biased towards the majority class, thereby improving their ability to generalize and make accurate predictions across all classes. Collectively, these feature engineering strategies optimize the preprocessing pipeline for electricity datasets, enhancing the efficacy and robustness of subsequent machine learning models.

3.3. Deep learning model

In Fig. 3, the process begins with the data being divided into training data (75 %) and testing data (25 %). The input layer receives the data, with the number of neurons denoted as 'n-neuron', representing the input dimensions or features as shown in Eq-1 and Eq-2. Following this, there are three hidden layers: Hidden Layer-1 comprises 512 neurons with ReLU activation processed in Eq-2, Eq-3 & Eq-3a, Hidden Layer-2 consists of 256 neurons with tanh activation processed in Eq-4, Eq-5 & Eq-5a, and Hidden Layer-3 includes 128 neurons with ReLU activation processed in Eq-6, Eq-7 & Eq-7a. Dropout layers with a dropout rate of 0.3 are applied after Hidden Layer-1 and Hidden Layer-2 to mitigate overfitting. The output layer utilizes softmax activation with 2 neurons, typically for binary classification tasks. Following processing, the results are analyzed, likely involving metrics such as accuracy or loss evaluation as depicted in Eq-8, Eq-9 & Eq-10. Overall, this network architecture employs a combination of activation functions including ReLU, tanh, and softmax, as shown in Equations 11, 14, and 16.

F0B7 Input Layer

$$X, W, B = \begin{bmatrix} x1 \\ x2 \\ . \\ . \\ x1034 \end{bmatrix}, [w1 \quad w2 \quad . \quad . \quad w24], [b1 \quad b2 \quad . \quad . \quad b24] \tag{1}$$

Table 1
Missing values in features.

Features	Name	Missing Values
Feature-0	January 1, 2014	29 %
Feature-1	October 1, 2014	28 %
Feature-2	November 1, 2014	29 %
Feature-3	December 1, 2014	29 %
Feature-4	1/13/2014	34 %
–	–	–
Feature-1030	May 9, 2016	1 %
Feature-1031	June 9, 2016	1 %
Feature-1032	July 9, 2016	1 %
Feature-1033	August 9, 2016	1 %
Feature-1034	September 9, 2016	1 %

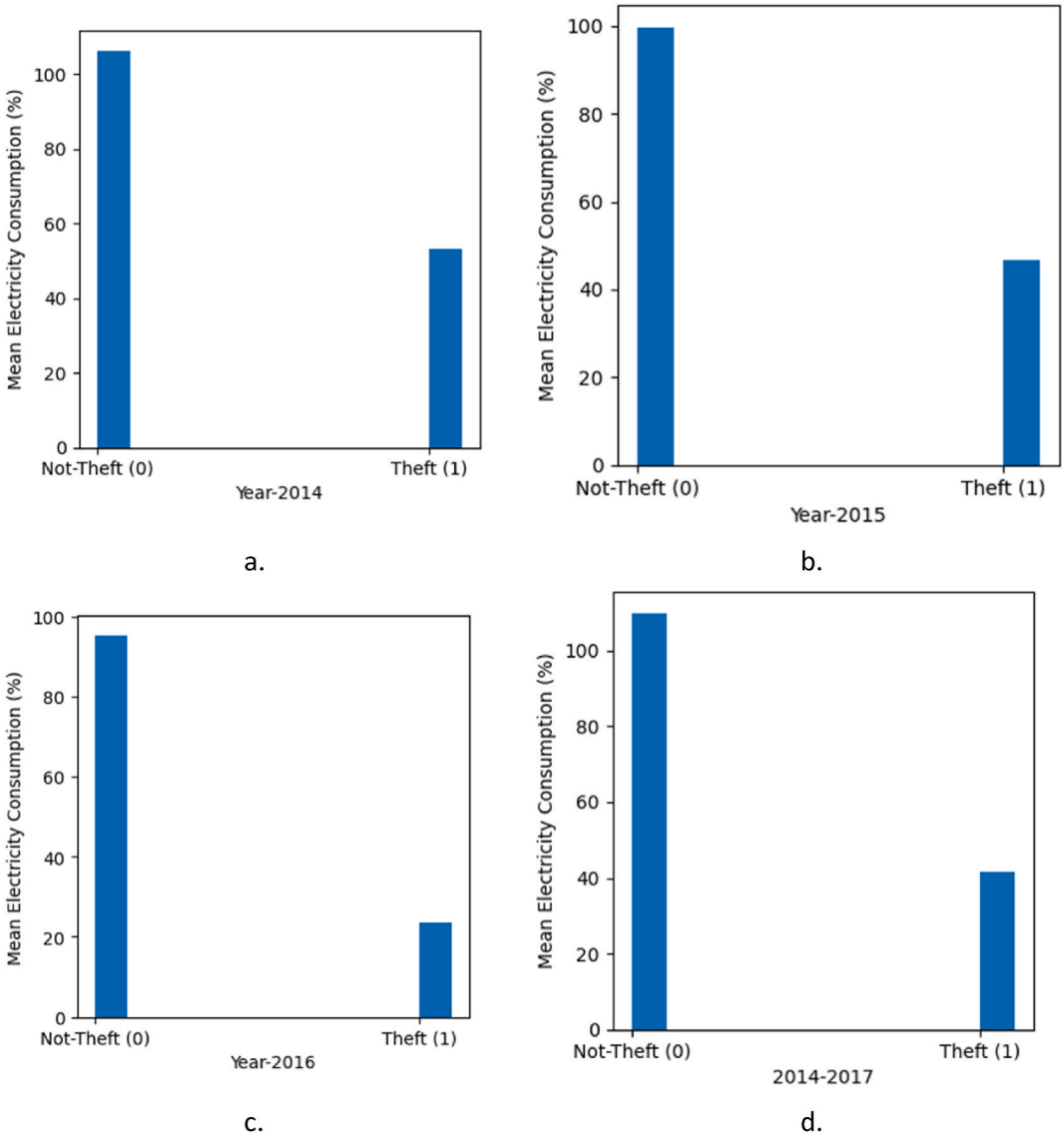

Fig. 2. Theft and not-theft electricity from 2014 to 2016.

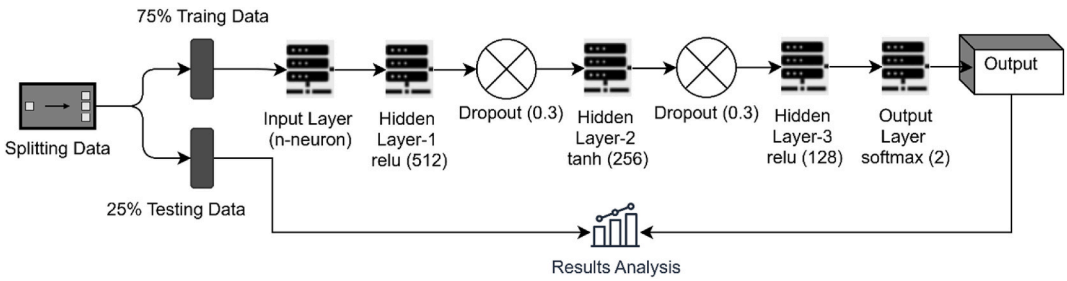

Fig. 3. Deep learning neural Network Model.

F0B7 Hidden Layer-1

$$\bigcup_{i=1}^{512} O_{1i} = \sum_{j=1}^{1034} \begin{cases} x_i * w_i + b_i, \text{Relu}(x_i * w_i + b_i) > 0 \\ 0, \text{Relu}(x_i * w_i + b_i) < 0 \end{cases} \tag{2}$$

$$\bigcup_{j=1}^{1034} New(W_j) = wj_{old} - \eta \frac{\delta L}{\delta Old(W_j)} \tag{3}$$

$$where \ \frac{\delta L}{\delta w1j_{old}} = \begin{cases} 1, x_i * w_i + b_i > 0 \\ 0, x_i * w_i + b_i < 0 \end{cases} \tag{3a}$$

F0B7 Hidden Layer-2

$$\bigcup_{i=1}^{256} O_{2i} = \sum_{j=1}^{512} \begin{cases} O_{1j} * w_{1j} + b_{1j}, \text{Relu}(O_{1j} * w_{1j} + b_{1j}) > 0 \\ 0, \text{Relu}(O_{1j} * w_{1j} + b_{1j}) < 0 \end{cases} \tag{4}$$

$$\bigcup_{j=1}^{512} W1j_{new} = w1j_{old} - \eta \frac{\delta L}{\delta w1j_{old}} \tag{5}$$

$$where \ \frac{\delta L}{\delta w1j_{old}} = \begin{cases} 1, \text{O1j} * \text{w1j} + \text{b1j} > 0 \\ 0, \text{O1j} * \text{w1j} + \text{b1j} < 0 \end{cases} \tag{5a}$$

F0B7 Hidden Layer-3

$$\bigcup_{i=1}^{128} O_{2i} = \sum_{j=1}^{256} \begin{cases} O_{1j} * w_{1j} + b_{1j}, \text{Relu}(O_{1j} * w_{1j} + b_{1j}) > 0 \\ 0, \text{Relu}(O_{1j} * w_{1j} + b_{1j}) < 0 \end{cases} \tag{6}$$

$$\bigcup_{j=1}^{256} W1j_{new} = w1j_{old} - \eta \frac{\delta L}{\delta w1j_{old}} \tag{7}$$

$$where \ \frac{\delta L}{\delta w1j_{old}} = \begin{cases} 1, \text{O1j} * \text{w1j} + \text{b1j} > 0 \\ 0, \text{O1j} * \text{w1j} + \text{b1j} < 0 \end{cases} \tag{7a}$$

F0B7 Output Layer

$$\hat{y} = \sum_{j=1}^{128} \begin{cases} 1, \text{Softmax}(O_{2j} * w_{2j} + b_{2j}) > 0.5 \\ 0, \text{Softmax}(O_{2j} * w_{2j} + b_{2j}) < 0.5 \end{cases} \tag{8}$$

$$L = \frac{(y - \hat{y})}{2} \tag{9}$$

$$\bigcup_{j=1}^{15} W2j_{new} = w2j_{old} - \eta \frac{\delta L}{\delta w2j_{old}} \tag{10}$$

The ReLU activation function is employed to activate this block, meaning that negative values of the matrix are set to 0, while positive values remain unchanged, as represented by Eq-11, Eq-12 and Eq-13.

$$Relu(z) = \max(0, z) \tag{11}$$

$$\text{Where z} = \left(\sum_{i=1}^{n} xi \, wi + bi \right) \tag{12}$$

$$Output \ at \ Hidden \ Layer1 = \begin{cases} z, Relu(z) \geq 0 \\ 0, Relu(z) < 0 \end{cases} \tag{13}$$

The hyperbolic tangent function (tanh) outputs values in the range [−1, 1], making it suitable for normalizing data and regulating the flow of information within neural networks, formula is given in Eq-14 and Eq-15.

$$\tanh(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}} \tag{14}$$

$$Output\ at\ Hidden\ Layer2 = \begin{cases} 1, tanh(z) \geq 0 \\ -1, tanh(z) < 0 \end{cases} \tag{15}$$

In this equation, e represents Euler's number (approximately equal to 2.71828), and x is the input to the function.

The softmax function takes as input a vector z of real numbers and outputs another vector of the same length. Each element in the output vector is a probability between 0 and 1, and the sum of all elements is equal to 1. The formula for the softmax function is shown in Eq-16 and Eq-17, where there are two outputs: 0.9 and 0.1. The first value represents 90 % probability of electricity not being stolen by the customer, and the second value represents 10 % probability of electricity theft by the customer.

$$f(z_i) = \frac{e^{z_i}}{\sum_j e^{zj}} \tag{16}$$

$$Output = \begin{cases} 0.9\ \%, Theft = 90\% \\ 0.1\ \%, Not - Tjeft = 10\% \end{cases} \tag{17}$$

The rest of the parameters used in the Model are shown in Table 2 About 75 % of the data from the training set was used during training [59]. All of the training data includes both the dependent variable (the result identifier) and the input factors (the predictor variables). Efficiency issues, including overfitting and underfitting, are addressed in the suggested technique by including validation data into the model. A 25 % testing subset is thus used for model evaluation [60]. When working with Keras, the parameters.

3.4. Direct classification

After preprocessing the data to handle missing entries, the dataset is directly fed into the deep learning model depicted in Fig. 3, with the value of 'n-neuron' set to 1034 at the input layer and trained for 5 epochs. Other parameters are taken from Table 2, such as the 'Training Set' comprising 31779 rows and 1034 columns, and the 'Test Set' containing 10593 rows and 1034 columns. Following training, the test data is evaluated using the model, and the predicted results are obtained.

A sample of the first five and a randomly selected record (with a prediction of False (0)) using direct classification is shown in Table 3. If the 'Softmax Predicted Value-0′ is greater than 'Softmax Predicted Value-1′, it indicates the 'Not-Theft (0)' class; otherwise, it indicates the 'Theft (1)' class.

Although the trained model achieved an overall accuracy of 91 %, its precision for the Theft class is 48 % and recall is 1 %, resulting in an F1-score of 03 %, as shown in Table 4. This suggests that many unreliable customers are also classified as reliable in this classification. While the model demonstrates high accuracy (91 %), its ability to recall instances of theft (Recall: 0.01) is relatively low, indicating a potential for improvement in correctly identifying theft cases.

3.5. Indirect classification

Indirect Classification refers to the process where the dataset is first passed through a scaler to scale all the values within the dataset to a range of 0–1, making it easily interpretable by the model. The dataset is then split into 75 % for training and 25 % for testing, and passed through a StandardScaler, which standardizes features by removing the mean and scaling to unit variance. This process ensures that the data has a mean of 0 and a standard deviation of 1. A sample of the first five records for both the training and testing data after applying StandardScaler is shown inTables 5 and 6, respectively. Due to the large size of the dataset (31779 rows with 1034 columns for the Training set and 10593 rows with 1034 columns for the Test Set), it cannot be fully displayed.

Training data from Table 5 is fed into the deep learning model depicted in Fig. 3, with the value of 'n-neuron' set to 1034 at the input layer and trained for 5 epochs. The remaining parameters such as learning rate, momentum, and batch size are taken from Table 2. A sample of the first five and last five detected classes using indirect classification on the test data from Table 6 is shown in Table 7. If the 'Softmax Predicted Value-0′ is greater than 'Softmax Predicted Value-1′, it indicates the 'Not-Theft (0)' class; otherwise, it indicates the 'Theft (1)' class.

Table 2
Used parameters in Model.

Parameters	Values
Training Set	31779 rows × 1034 columns
Test Set	10593 rows, 1034 columns
Learning Rate	0.001
Momentum	0.9
Validation_split	0.25
Batch Size	256
Total Batch	94
Loss Function	binary_crossentropy

Table 3
Prediction using direct classification.

Features	Actual	Softmax Predicted Value-0	Softmax Predicted Value-1	Predicted Result
Feature-0	Not-Theft (0)	0.995192	0.004808	True Not-Theft (0)
Feature −1	Not-Theft (0)	0.998483	0.001517	True Not-Theft (0)
Feature −2	Not-Theft (0)	0.997343	0.002657	True Not-Theft (0)
Feature −3	Not-Theft (0)	0.998777	0.001223	True Not-Theft (0)
Feature −4	Not-Theft (0)	0.997148	0.002852	True Not-Theft (0)
Feature −170	Theft (1)	0.996946	0.003054	False Not-Theft (0)

Table 4
Measures from direct classification.

Class	Precision	Recall	F1 Score	Support
Not-Theft (0)	0.91	0.1	0.95	9677
Theft (1)	0.48	0.01	0.03	916
Average	0.88	0.91	0.87	10593
Accuracy	0.91			

Table 5
Standard Vector's sample from 75 % training set.

Customers	Feature-0	Feature-1	Feature-2	Feature-3	Feature-4
Customer-0	0.058349	0.082829	0.329711	0.149208	0.286052
Customer-1	−0.00096	0.000305	0.002359	0.001404	0.11674
Customer-2	−0.16019	−0.0975	−0.0918	−0.11951	−0.18499
Customer-3	−0.00096	0.000305	0.002359	0.001404	0.11674
Customer-4	−0.00096	0.000305	0.002359	0.001404	0.11674

Table 6
Standard Vector's sample from 25 % test set.

Customers	Feature-0	Feature-1	Feature-2	Feature-3	Feature-4
Customer-0	−0.00259	0.135581	−0.30413	0.275571	0.286052
Customer-1	−0.01562	0.045043	0.17927	0.07339	−0.20071
Customer-2	−0.00096	0.000305	0.002359	0.001404	0.11674
Customer-3	−0.00096	0.000305	0.002359	0.001404	0.11674
Customer-4	−0.00096	0.000305	0.002359	0.001404	0.11674

Table 7
Prediction using indirect Classification.

Features	Actual	Softmax Predicted Value-0	Softmax Predicted Value-1	Predicted Result
Feature-0	Not-Theft (0)	0.960379	0.039621	True Not-Theft (0)
Feature −1	Not-Theft (0)	0.998239	0.001761	True Not-Theft (0)
Feature −2	Not-Theft (0)	0.989917	0.010083	True Not-Theft (0)
Feature −3	Not-Theft (0)	0.999246	0.000754	True Not-Theft (0)
Feature −4	Not-Theft (0)	0.99757	0.00243	True Not-Theft (0)
Feature −170	Theft (1)	0.990929	0.009071	False Not-Theft (0)

Table 8
Measures from indirect classification.

Class	Precision	Recall	F1 Score	Support
Not-Theft (0)	0.92	0.99	0.95	9677
Theft (1)	0.46	0.05	0.10	916
Average	0.88	0.91	0.88	10593
Accuracy	0.91			

Despite the trained model achieving an accuracy of 91 %, the precision for the Theft class is 46 %, with a recall of 05 % and an F1-score of 10 %, as indicated in Table 8. This suggests that many instances of theft may be overlooked, as a significant portion of unreliable customers are being classified as reliable in this scenario. While the model exhibits high overall accuracy, its ability to accurately identify instances of theft (Recall: 0.05) is notably lacking, indicating the need for improvement in correctly identifying theft cases.

3.5.1. PCA for feature extraction

The graph in Fig. 4 illustrates the relationship between Cumulative Explained Variance Ratio and the Number of Principal Components. It reveals an elbow point occurring approximately between 200 and 300 principal components. Beyond this threshold, the increase in explained variance ratio becomes less pronounced. Therefore, it may be advisable to consider selecting around 200 to 300 principal components for dimensionality reduction, as this allows for retaining most of the variance in the dataset.

Principal Component Analysis (PCA) is a mathematical technique used for dimensionality reduction and feature extraction. The main idea behind PCA is to transform the original data into a new coordinate system, where the axes (principal components) are orthogonal to each other and are ordered by the amount of variance they explain in the data. Given a dataset X of size n × m, where n is the number of samples and m is the number of features, the steps of PCA can be represented in following equations Eq-8 to Eq-13:

$$mean = \frac{1}{n}\sum_{i=1}^{n} xi \tag{8}$$

$$X_{centered} = X - mean \tag{9}$$

$$covariance_{matrix} = \frac{1}{n} X^{T}_{centered}.X_{centered} \tag{10}$$

$$eigenvalues, eigenvectors = eig(covariance\_matrix) \tag{11}$$

$$principal\_components = eigenvectors[:, :k] \tag{12}$$

$$transformed_{data} = X_{centered}\ .principal\_components \tag{13}$$

Using PCA, the dataset is reduced to 300 columns from its original 1034 columns. Before modelling, the dataset undergoes scaling to normalize all values between the ranges of 0–1, facilitating ease of interpretation by the model. The dataset is then split into 75 % for training and 25 % for testing and passed through a Standard Scaler to standardize features, ensuring a mean of 0 and a standard deviation of 1. Samples of the first five records for both training and testing data after StandardScaler transformation are presented in Tables 9 and 10, respectively. Due to the large size of the dataset, with 31779 rows and 300 columns for the Training set and 10593 rows and 300 columns for the Test Set, it cannot be fully displayed in the tables.

After extracting features using PCA, these transformed features are then used for the classification task. The training data from Table 9 is utilized in a deep learning model, as depicted in Fig. 3, with the input layer comprising 300 neurons and trained for 5 epochs. Other parameters such as learning rate, momentum, and batch size are obtained from Table 2. Table 11 displays samples of the first five and last five detected classes using the deep learning model on the test data from Table 10. In this classification, if 'Softmax Predicted Value-0′ is greater than 'Softmax Predicted Value-1′, it indicates the 'Not-Theft (0)' class; otherwise, it indicates the 'Theft (1)' class.

While the model achieves an overall accuracy of 91 %, its precision for identifying theft instances is only 39 %, with a recall of 03 % and an F1-score of 06 % (Table 12). This indicates that a significant number of unreliable customers are being mistakenly classified as reliable. Despite its high accuracy, the model's ability to recall theft instances is notably low (Recall: 0.03), highlighting the need for

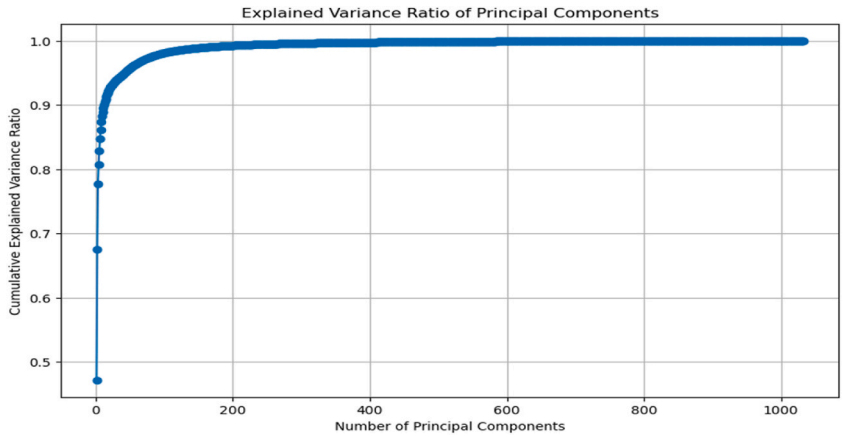

Fig. 4. Explained variance ratio of principal components.

Table 9
Standard Vector's sample from 75 % training set using PCA-Features.

Customers	Feature-0	Feature-1	Feature-2	Feature-3	Feature-4
Customer-0	3.751599	−0.33999	−0.95836	−0.15773	0.558934
Customer-1	−2.48934	−0.52361	−1.27975	−1.67704	0.063539
Customer-2	−1.9622	1.295758	−2.136	−2.04847	1.953631
Customer-3	−2.6735	2.083855	−0.13416	0.074281	1.566703
Customer-4	−6.83315	0.568223	1.585305	0.316115	−1.23705

Table 10
Standard Vector's sample from 25 % test set using PCA-Features.

Customers	Feature-0	Feature-1	Feature-2	Feature-3	Feature-4
Customer-0	−0.46064	−1.03737	−0.07175	−0.22073	−0.18469
Customer-1	1.923886	−2.23756	1.432769	−0.32944	0.941785
Customer-2	3.961314	−0.6589	−2.3892	−2.5295	1.339889
Customer-3	7.303255	−1.90127	−1.20112	−1.87013	1.768308
Customer-4	3.452378	−1.54425	−0.57962	−0.16906	0.123072

Table 11
Prediction using PCA-Features classification.

Features	Actual	Softmax Predicted Value-0	Softmax Predicted Value-1	Predicted Result
Feature-0	Theft (1)	0.951588	0.048412	False Not-Theft (0)
Feature −1	Not-Theft (0)	0.949844	0.050156	True Not-Theft (0)
Feature −2	Not-Theft (0)	0.785116	0.214884	True Not-Theft (0)
Feature −3	Not-Theft (0)	0.900009	0.099991	True Not-Theft (0)
Feature −4	Not-Theft (0)	0.883137	0.116863	True Not-Theft (0)
Feature −170	Not-Theft (0)	0.792567	0.207432	True Not-Theft (0)

Table 12
Measures from PCA-Features classification.

Class	Precision	Recall	F1 Score	Support
Not-Theft (0)	0.91	0.1	0.95	9661
Theft (1)	0.39	0.03	0.06	932
Average	0.87	0.91	0.87	10593
Accuracy	0.91			

improvement in correctly identifying theft cases.

3.5.2. T-SNE for feature extraction

T-distributed Stochastic Neighbor Embedding (T-SNE) is a nonlinear dimensionality reduction technique primarily used for data visualization. It converts high-dimensional data into two or three dimensions while preserving the relationships between data points as much as possible. T-SNE is particularly effective at visualizing clusters or groups within high-dimensional data.

Given a dataset X of size $n \times m$, where n is the number of samples and m is the number of features, the steps of T-SNE can be represented in the following simplified steps.

1. Compute pairwise similarities in the high-dimensional space.
2. Define a probability distribution over pairs of high-dimensional objects.
3. Compute pairwise similarities in the low-dimensional space.
4. Minimize the divergence between the two distributions using gradient descent.

Another scheme for reducing 1034 columns into 2 columns is t-distributed Stochastic Neighbor Embedding.

The major equation in t-SNE involves calculating the conditional probabilities Pj|I for each pair of data points i and j in the high-dimensional space. Eq-14 is a mathematical form of t-SNE.

$$P_{j|i} = \frac{\exp\left(-\|x_i - x_j\|^2 / 2\sigma_i^2\right)}{\sum_{k \neq i}\left(-\|x_i - x_k\|^2 / 2\sigma_i^2\right)} \tag{14}$$

This equation computes the similarity between data points i and j using a Gaussian kernel, where xi and xj are the high-dimensional feature vectors of the data points, and σi is a parameter controlling the variance of the Gaussian distribution for point i. This equation is fundamental in t-SNE as it establishes the conditional probabilities based on pairwise similarities, which are further used in the algorithm's optimization process to embed the data into a lower-dimensional space while preserving local structures.

Using t-SNE, the dataset is reduced to 2 columns from its original 1034 columns. Before modelling, the dataset is scaled to range between 0 and 1 to facilitate model interpretation. The dataset is then split into 75 % for training and 25 % for testing and passed through a StandardScaler to standardize features. This process ensures that the data has a mean of 0 and a standard deviation of 1. Samples of the first five records for both training and testing data after StandardScaler transformation are presented in Tables 13 and 14, respectively. Due to the large size of the dataset, with 31779 rows and 2 columns for the Training set and 10593 rows and 2 columns for the Test Set, it cannot be fully displayed in the tables.

After extracting features using T-SNE, these transformed features are then used for the classification task. The training data from Table 13 is utilized in a deep learning model, as depicted in Fig. 3, with the input layer comprising 2 neurons and trained for 5 epochs. Other parameters such as learning rate, momentum, and batch size are obtained from Table 2. Table 15 displays samples of the first five and last five detected classes using the deep learning model on the test data from Table 14. In this classification, if 'Softmax Predicted Value-0′ is greater than 'Softmax Predicted Value-1′, it indicates the 'Not-Theft (0)' class; otherwise, it indicates the 'Theft (1)' class.

Despite achieving an overall accuracy of 91 %, the trained model exhibits precision and recall scores of 0 % for the Theft class, resulting in an F1-score of 0 % (Table 16). This indicates that a significant number of unreliable customers are erroneously classified as reliable. Despite its high accuracy, the model's recall for theft instances is notably deficient (Recall: 0.0), underscoring the imperative to enhance its ability to correctly identify theft cases.

3.5.3. UMAP for feature extraction

UMAP (Uniform Manifold Approximation and Projection) is a dimensionality reduction technique designed to handle mixed data types, such as continuous and categorical data. Developed by McInnes, Healy, and Melville in 2018, UMAP is based on manifold learning and topological data analysis. It excels at preserving both local and global data structures, making it ideal for visualizing and understanding complex datasets. UMAP is particularly effective in reducing high-dimensional data to lower dimensions while maintaining essential relationships between data points. While PCA is typically used for linearly structured data and t-SNE for non-linearly structured data, UMAP is well-suited for mixed-mode data. For example, UMAP can reduce a dataset with 1034 columns to just 4 columns.

Given a dataset X of size $n \times m$, where n is the number of samples and m is the number of features, the steps of UMAP can be summarized as follows.

1. Construct a high-dimensional graph representing the data.
2. Optimize a low-dimensional graph to be as structurally similar to the high-dimensional graph as possible.
3. Minimize the divergence between the high-dimensional and low-dimensional graphs using gradient descent.

Using UMAP, the dataset is reduced to 4 columns from its original 1034 columns. Before modelling, the dataset is scaled to range between 0 and 1 to facilitate model interpretation. The dataset is then split into 75 % for training and 25 % for testing and passed through a StandardScaler to standardize features. This process ensures that the data has a mean of 0 and a standard deviation of 1. Samples of the first five records for both training and testing data after StandardScaler transformation are presented in Tables 17 and 18, respectively. Due to the large size of the dataset, with 31779 rows and 2 columns for the Training set and 10593 rows and 2 columns for the Test Set, it cannot be fully displayed in the tables.

After extracting features using UMAP, these transformed features are then used for the classification task. The training data from Table 17 is utilized in a deep learning model, as depicted in Fig. 3, with the input layer comprising 2 neurons and trained for 5 epochs. Other parameters such as learning rate, momentum, and batch size are obtained from Table 2. Table 19 displays samples of the first five and last five detected classes using the deep learning model on the test data from Table 18. In this classification, if 'Softmax Predicted Value-0′ is greater than 'Softmax Predicted Value-1′, it indicates the 'Not-Theft (0)' class; otherwise, it indicates the 'Theft (1)' class.

Despite achieving an overall accuracy of 91 %, the trained model exhibits precision and recall scores of 0 % for the Theft class, resulting in an F1 score of 0 % (Table 20). This indicates that a significant number of unreliable customers are erroneously classified as reliable. Despite its high accuracy, the model's recall for theft instances is notably deficient (Recall: 0.0), underscoring the imperative to enhance its ability to correctly identify theft cases. Results of.

The results presented in Tables 16 and 20 demonstrate an identical outcome, indicating that t-SNE and UMAP yield equivalent

Table 13
Standard Vector's Sample from 75 % Training Set Using t-SNE Features.

Customers	Feature-0	Feature-1
Customer-0	1.233045	−0.86622
Customer-1	1.139349	0.896794
Customer-2	0.544359	0.765291
Customer-3	0.890204	0.4715
Customer-4	−1.53817	0.767101

Table 14
Standard Vector's Sample from 25 % Test Set Using t- SNE Features.

Customers	Feature-0	Feature-1
Customer-0	−0.31194	−1.42367
Customer-1	0.861471	−0.31347
Customer-2	1.241249	−0.61973
Customer-3	1.552921	−0.77741
Customer-4	1.107586	−0.57651

Table 15
Prediction Using t- SNE Classification.

Features	Actual	Softmax Predicted Value-0	Softmax Predicted Value-1	Predicted Result
Feature-0	Theft (1)	0.985282	0.014717	False Not-Theft (0)
Feature −1	Not-Theft (0)	0.972073	0.027927	True Not-Theft (0)
Feature −2	Not-Theft (0)	0.982277	0.017723	True Not-Theft (0)
Feature −3	Not-Theft (0)	0.987199	0.012801	True Not-Theft (0)
Feature −4	Not-Theft (0)	0.979934	0.020066	True Not-Theft (0)
Feature −170	Not-Theft (0)	0.985282	0.014717	True Not-Theft (0)

Table 16
Measures from t- SNE Classification.

Class	Precision	Recall	F1 Score	Support
Not-Theft (0)	0.91	0.1	0.95	9661
Theft (1)	0.00	0.00	0.00	932
Average	0.83	0.91	0.87	10593
Accuracy	0.91			

Table 17
Standard Vector's sample from 75 % training set using UMAP features.

Customers	Feature-0	Feature-1	Feature-2	Feature-3
Customer-0	−0.51282	0.348598	−1.12284	0.908822
Customer-1	−0.62379	−1.73425	−0.33973	−2.24039
Customer-2	−0.66981	−1.25527	−0.02641	−0.73275
Customer-3	−0.41719	−1.1326	−0.36901	−1.18961
Customer-4	−0.91805	2.030792	0.933767	1.074828

Table 18
Standard Vector's sample from 25 % test set using UMAP features.

Customers	Feature-0	Feature-1	Feature-2	Feature-3
Customer-0	1.35992	0.317409	−0.33412	−0.50542
Customer-1	−0.14663	0.667956	−0.91096	−0.51021
Customer-2	−0.52451	0.345385	−1.05734	0.185526
Customer-3	−0.65367	0.387898	−1.32626	1.0584
Customer-4	−0.42799	0.795863	−0.98597	0.035851

Table 19
Prediction using UMAP classification.

Features	Actual	Softmax Predicted Value-0	Softmax Predicted Value-1	Predicted Result
Feature-0	Theft (1)	0.985282	0.014717	False Not-Theft (0)
Feature −1	Not-Theft (0)	0.972073	0.027927	True Not-Theft (0)
Feature −2	Not-Theft (0)	0.982277	0.017723	True Not-Theft (0)
Feature −3	Not-Theft (0)	0.987199	0.012801	True Not-Theft (0)
Feature −4	Not-Theft (0)	0.979934	0.020066	True Not-Theft (0)
Feature −170	Not-Theft (0)	0.985282	0.014717	True Not-Theft (0)

Table 20
Measures from UMAP classification.

Class	Precision	Recall	F1 Score	Support
Not-Theft (0)	0.91	0.1	0.95	9661
Theft (1)	0.00	0.00	0.00	932
Average	0.83	0.91	0.87	10593
Accuracy	0.91			

results when applied to the electricity dataset, albeit with different dimensionality reduction targets. Specifically, t-SNE reduces the dataset from 1034 columns to 2 columns, while UMAP reduces it to 4 columns. Despite the difference in the number of reduced dimensions, both techniques effectively capture the essential structure and relationships within the data, leading to comparable results. This suggests that both t-SNE and UMAP are equally effective for dimensionality reduction in this particular dataset, providing similar insights and data representation.

3.6. Balanced dataset classification

As depicted in Fig. 2, it's evident that the Theft class is significantly underrepresented compared to the Not-Theft class. Therefore, there's a necessity to balance the dataset. To address this imbalance, we employ the following techniques for the model depicted in Fig. 3.

3.6.1. Random-Under-Sampler (RUS)

The Random Under-Sampling (RUS) technique involves randomly removing samples from the majority class to balance the dataset. Following RUS, the dataset, now balanced with 1034 columns, is scaled to ensure values fall within the range of 0–1, aiding model interpretation. Subsequently, the dataset is split into 75 % for training and 25 % for testing. StandardScaler is then applied to standardize features, ensuring a mean of 0 and a standard deviation of 1. Tables 21 and 22 display samples of the first five records for both the training and testing datasets post-StandardScaler transformation. However, due to the dataset's size, with 5422 rows and 1034 columns for the Training set and 1808 rows and 1034 columns for the Test Set, complete records cannot be shown in the tables.

The training data from Table 21 is utilized in a deep learning model in Fig. 3 with the RUS-Method, featuring 1034 neurons at the input layer and trained for 200 epochs. The decision to use 200 epochs is based on the increasing accuracy observed after each epoch. Other parameters, such as learning rate, momentum, and batch size, are derived from Table 2. Table 23 showcases samples of the first five and last five detected classes using the RUS-Method classification on the test data from Table 22. In this classification, if 'Softmax Predicted Value-0′ exceeds 'Softmax Predicted Value-1′, it indicates the 'Not-Theft (0)' class; otherwise, it indicates the 'Theft (1)' class.

The trained model has an accuracy of 68 %. For the Theft class, it shows a precision of 71 % and a recall of 58 %, resulting in an F1 score of 64 % (Table 24). However, some unreliable customers are mistakenly classified as reliable in this process.

Although the model's accuracy is above average (68 %), its ability to recall theft instances (Recall: 0.58) is relatively low. This suggests there's room for improvement in accurately identifying theft cases.

3.6.2. SMOTE technique

SMOTE (Synthetic Minority Over-sampling Technique) is a method used to address class imbalance in a dataset by generating synthetic samples for the minority class. Using SMOTE, the dataset is balanced and reduced to 1034 columns. Afterwards, the dataset undergoes scaling to normalize values between 0 and 1, making it easier for the model to interpret. The dataset is then split into 75 % for training and 25 % for testing, followed by standardization using StandardScaler to ensure a mean of 0 and a standard deviation of 1. Tables 25 and 26 display samples of the first five records for both training and testing data post-StandardScaler transformation. Due to the large dataset size, with 58135 rows and 1034 columns for the Training set and 19379 rows and 1034 columns for the Test Set, the tables cannot fully display all records.

The training data from Table 25 is used in a deep learning model in Fig. 3 with the SMOTE-Method, featuring 1034 neurons at the input layer and trained for 200 epochs. We chose 200 epochs based on the increasing accuracy observed after each epoch. Other parameters, such as learning rate, momentum, and batch size, are obtained from Table 2. Table 27 displays samples of the first five and last five detected classes using the SMOTE-Method classification on the test data from Table 26. In this classification, if 'Softmax Predicted Value-0′ exceeds 'Softmax Predicted Value-1′, it indicates the 'Not-Theft (0)' class; otherwise, it indicates the 'Theft (1)'

Table 21
Standard Vector's sample from 75 % training set using RUS-Method.

Customers	Feature-0	Feature-1	Feature-2	Feature-3	Feature-4
Customer-0	0.058341	−0.0653	−0.06416	−0.0229	−0.01658
Customer-1	−0.08635	−0.04138	−0.01538	−0.16801	−0.01843
Customer-2	−0.1401	−0.12599	−0.11687	−0.14538	−0.02127
Customer-3	−0.15787	−0.12447	−0.12511	−0.16069	−0.02154
Customer-4	−0.05008	−0.05291	−0.04527	−0.05012	−0.01401

Table 22
Standard Vector's sample from 25 % test set using RUS -method.

Customers	Feature-0	Feature-1	Feature-2	Feature-3	Feature-4
Customer-0	−0.15802	−0.14139	−0.134	−0.16801	−0.02236
Customer-1	0.000525	0.011423	0.042434	0.076284	−0.0145
Customer-2	−0.03139	−0.08782	0.256279	0.179627	−0.01147
Customer-3	−0.05008	−0.05291	−0.04527	−0.05012	−0.01401
Customer-4	0.171111	0.046541	0.020984	0.041337	−0.01525

Table 23
Prediction using RUS-Method classification.

Features	Actual	Softmax Predicted Value-0	Softmax Predicted Value-1	Predicted Result
Feature-0	Theft (1)	0.59251	0.40749	False Not-Theft (0)
Feature −1	Theft (1)	0.567625	0.432375	False Not-Theft (0)
Feature −2	Theft (1)	0.089038	0.910962	True Theft (1)
Feature −3	Not-Theft (0)	0.814608	0.185392	True Not-Theft (0)
Feature −4	Not-Theft (0)	0.692272	0.307728	True Not-Theft (0)
Feature −170	Not-Theft (0)	0.792567	0.207432	True Not-Theft (0)

Table 24
Measures from RUS-Method classification.

Class	Precision	Recall	F1 Score	Support
Not-Theft (0)	0.65	0.77	0.70	908
Theft (1)	0.71	0.58	0.64	900
Average	0.68	0.68	0.67	1808
Accuracy	0.68			

Table 25
Standard Vector's sample from 75 % training set using SMOTE-Method.

Customers	Feature-0	Feature-1	Feature-2	Feature-3	Feature-4
Customer-0	−0.04927	−0.05202	−0.04693	−0.05076	−0.00843
Customer-1	−0.13265	−0.10567	−0.12429	−0.13986	−0.02292
Customer-2	−0.11607	−0.13435	−0.1395	−0.11407	−0.02242
Customer-3	−0.18032	−0.15934	−0.159	−0.19754	−0.02678
Customer-4	−0.18124	−0.15383	−0.16	−0.19859	−0.02699

Table 26
Standard Vector's sample from 25 % test set using SMOTE -method.

Customers	Feature-0	Feature-1	Feature-2	Feature-3	Feature-4
Customer-0	−0.07085	−0.06463	−0.06175	−0.06823	−0.01097
Customer-1	−0.18124	−0.07568	−0.112	−0.19859	−0.02328
Customer-2	−0.14996	−0.14503	−0.12937	−0.14883	−0.02337
Customer-3	−0.04927	−0.05202	−0.04693	−0.05076	−0.00843
Customer-4	−0.04927	−0.05202	−0.04693	−0.05076	−0.00843

Table 27
Prediction using SMOTE -method classification.

Features	Actual	Softmax Predicted Value-0	Softmax Predicted Value-1	Predicted Result
Feature-0	Theft (1)	0.000155	0.999845	True Theft (1)
Feature −1	Not-Theft (0)	0.988456	0.011544	True Not-Theft (0)
Feature −2	Theft (1)	0.000142	0.999858	True Theft (1)
Feature −3	Theft (1)	0.011812	0.988188	True Theft (1)
Feature −4	Theft (1)	0.008208	0.991791	True Theft (1)
Feature −170	Theft (1)	0.000155	0.999845	True Theft (1)

Table 28
Measures from SMOTE -method classification.

Class	Precision	Recall	F1 Score	Support
Not-Theft (0)	0.91	0.89	0.90	9627
Theft (1)	0.90	0.92	0.90	9752
Average	0.90	0.90	0.90	19379
Accuracy	0.90			

class.

The trained model achieves an accuracy of 90 %. For the Theft class, it demonstrates a precision of 90 % and a recall of 92 %, resulting in an F1 score of 90 % (Table 28). Most of the unreliable customers are accurately predicted. With its high accuracy (90 %) and strong recall for theft instances (Recall: 0.92), the model shows promise for further implementation.

3.6.3. Random Over Sampling Technique (ROS)

Using the Random Over-Sampling Technique (ROS) entails replicating samples from the minority class randomly until the class distribution achieves balance. With ROS, the dataset is balanced, consisting of 1034 columns. Afterwards, the dataset undergoes scaling to normalize values between 0 and 1, enhancing model interpretability. Following this, the dataset is divided into 75 % for training and 25 % for testing. Subsequently, it is subjected to StandardScaler to standardize features, ensuring a mean of 0 and a standard deviation of 1. Samples of the first five records for both training and testing data post-StandardScaler transformation are displayed in Tables 29 and 30. Due to the large dataset size, comprising 58135 rows and 1034 columns for the Training set and 19379 rows and 1034 columns for the Test Set, not all records can be fully displayed in the tables.

The deep learning model in Fig. 3 utilizes the ROS-Method, with 1034 neurons at the input layer trained for 200 epochs using the training data from Table 29. The decision to use 200 epochs is based on the observed increase in accuracy over each epoch. Other parameters such as learning rate, momentum, and batch size are sourced from Table 2. Table 31 showcases samples of the first five and last five detected classes using the ROS-Method classification on the test data from Table 30. In this classification, if 'Softmax Predicted Value-0′ exceeds 'Softmax Predicted Value-1′, it indicates the 'Not-Theft (0)' class; otherwise, it indicates the 'Theft (1)' class.

The trained model attains an accuracy of 91 %. Specifically for the Theft class, it demonstrates a precision of 89 % and a recall of 94 %, resulting in an F1 score of 91 % (Table 32.). The majority of unreliable customers are accurately predicted. With its high accuracy (91 %) and strong recall for theft instances (Recall: 0.94), the model shows promise for further implementation.

4. Results and discussions

All models are implemented in Python using Keras, featuring a single hidden layer with two dropout layers positioned before and after the hidden layer. The input layer and output are adjusted based on the input features using relevant methods outlined in the methodology section. Further details on the results are provided below.

4.1. Accuracy and loss during epochs on imbalanced dataset

Before addressing the imbalanced dataset, trained models achieved high accuracy. However, precision and recall for the Theft class fell short, as depicted in Fig. 5. This figure illustrates that all models trained on imbalanced data yielded unsatisfactory results on the confusion matrix concerning the Theft class. Specifically, in Fig. 5, Not Scaling, Scaling, PCA, and t-SNE detected 13, 50, 29, and 0 instances of the Theft class, respectively. Figs. 6 and 7 demonstrate an increase in accuracy and a decrease in loss throughout epochs.

4.2. Accuracy and loss during epochs on balanced dataset

Due to the necessity of balancing the dataset, the proposed model yields superior outcomes compared to the imbalanced dataset. In Fig. 8, trained models using balanced techniques exhibit high accuracy and recall for the Theft class. Notably, all models trained on balanced data demonstrate favorable results on the confusion matrix for both the Theft and Not-Theft classes. Specifically, RUS, SMOTE, and ROS exhibit the best True-Theft and True-Not-Theft outcomes in Fig. 8. Additionally, Figs. 9 and 10 depict an upward trend in accuracy and a downward trend in loss over the course of epochs.

4.3. Selection of best models based on precision, recall, and ROC Curve analysis

As the 'Not-Theft' class gains high precision and recall, but the 'Theft' class does not achieve high values, we prioritize the precision and recall of the 'Theft' class for consideration. From Fig. 11, it is clear that each model demonstrates various strengths and weaknesses across different evaluation metrics. While several models achieve high accuracy scores of 0.91, there are notable differences in precision and recall for theft detection. Among the models listed, the "ROS" (Random Over-Sampling) model stands out with an accuracy of 0.91, precision for theft of 0.89, and recall for theft of 0.94. These metrics indicate that the ROS model not only achieves high overall accuracy but also excels in accurately identifying instances of theft, with a high precision rate and a high recall rate. Comparatively, other models like "No Scaling," "Scaling," "PCA," and "t-SNE" exhibit similar accuracies of 0.91 but have lower precision and recall

Table 29
Standard Vector's sample from 75 % training set using ROS-Method.

Customers	Feature-0	Feature-1	Feature-2	Feature-3	Feature-4
Customer-0	0.312363	0.051979	0.097924	0.032764	−0.00665
Customer-1	−0.04967	−0.05311	−0.04727	−0.05113	−0.01195
Customer-2	−0.11207	−0.13101	−0.13219	−0.10725	−0.01924
Customer-3	−0.17209	−0.15465	−0.15007	−0.18124	−0.02151
Customer-4	0.051475	0.01262	0.02117	0.013343	−0.01105

Table 30
Standard Vector's sample from 25 % test set using ROS-Method.

Customers	Feature-0	Feature-1	Feature-2	Feature-3	Feature-4
Customer-0	−0.17295	−0.15524	−0.15099	−0.18216	−0.02162
Customer-1	−0.17295	−0.0755	−0.10696	−0.18216	−0.01969
Customer-2	−0.00975	−0.04524	−0.05268	−0.06749	−0.01442
Customer-3	−0.04967	−0.05311	−0.04727	−0.05113	−0.01195
Customer-4	−0.04967	−0.05311	−0.04727	−0.05113	−0.01195

Table 31
Prediction using ROS-Method classification.

Features	Actual	Softmax Predicted Value-0	Softmax Predicted Value-1	Predicted Result
Feature-0	Theft (1)	0.000884	0.999116	True Theft (1)
Feature −1	Not-Theft (0)	0.99999	9.917467e-06	True Not-Theft (0)
Feature −2	Theft (1)	0.003885	0.996115	True Theft (1)
Feature −3	Theft (1)	0.41503	0.58497	True Theft (1)
Feature −4	Theft (1)	6.6416703e-07	0.999999	True Theft (1)
Feature −170	Theft (1)	0.000884	0.999116	True Theft (1)

Table 32
Measures from ROS -method classification.

Class	Precision	Recall	F1 Score	Support
Not-Theft (0)	0.93	0.89	0.91	9627
Theft (1)	0.89	0.94	0.91	9752
Average	0.91	0.91	0.91	19379
Accuracy	0.91			

scores for theft detection. The "RUS" model shows decent precision for theft (0.71) but comparatively lower recall (0.58) and a lower overall accuracy of 0.68. On the other hand, "SMOTE" also performs well with a precision for theft of 0.9 and a recall for theft of 0.92, but its overall accuracy is slightly lower than ROS at 0.9. Considering the balance between accuracy, precision, and recall, the ROS model emerges as the most favorable choice among the listed models for theft detection, as it achieves high scores across all three key metrics.

The acronym ROC stands for Receiver Operating Characteristic, and ROC curves are commonly employed to visually illustrate the trade-off between clinical sensitivity and specificity across various cutoff points for a test or a combination of tests. Moreover, the area under the ROC curve provides insight into the overall efficacy of the test(s) within a model. Since a larger area under the ROC curve indicates a more effective test, these areas are utilized to compare the utility of different tests [50].

Fig. 12 displays the curves for all the deep learning models used on the test data. In this figure, the area under the curve for SMOTE and ROS is the highest. Therefore, based on the ROC curve analysis, SMOTE and ROS are selected as the best models.

4.4. Ablation study

The assertion regarding the superiority of the proposed model over CNN, RNN, LSTM, or Ensemble techniques is supported by several key factors. Firstly, the inclusion of dropout layers and three hidden layers in the proposed architecture contributes to reducing computational costs compared to models relying solely on CNN, RNN, LSTM, or Ensemble techniques. The utilization of fewer layers in the proposed model significantly enhances computational efficiency. By reducing the complexity of the network architecture, fewer parameters need to be trained during the learning process, thereby decreasing the computational burden. This streamlined architecture not only accelerates training times but also facilitates faster inference during deployment, making the model well-suited for real-time applications or scenarios with limited computational resources. Furthermore, the simplicity of the architecture minimizes the

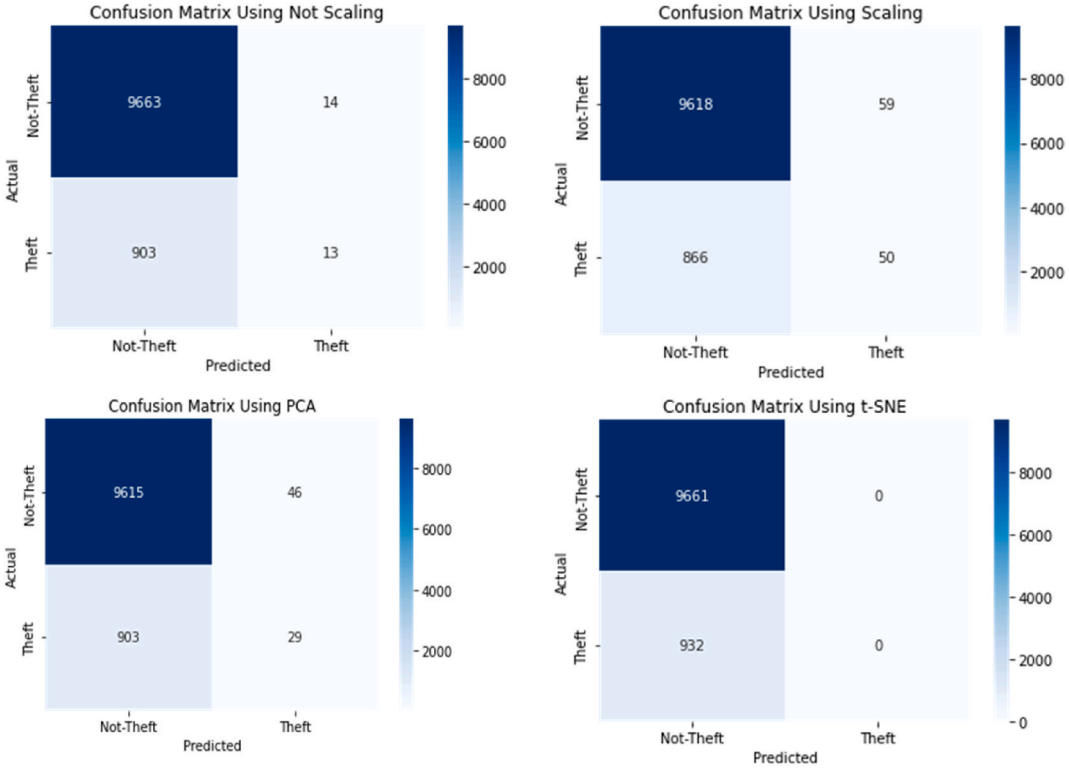

Fig. 5. Confusion matrix on imbalanced dataset.

risk of overfitting, where the model becomes overly complex and fails to generalize well to unseen data. With fewer layers, the model is less prone to memorizing noise or irrelevant patterns in the training data, resulting in improved generalization performance on unseen data.

Moreover, while existing works based on CNN, RNN, LSTM, or Ensemble techniques have demonstrated promising results, there remains room for improvement in terms of accuracy, precision, or recall. By addressing this gap and leveraging the advantages of dropout layers and multiple hidden layers, the proposed model offers an opportunity for significant performance enhancements.

In summary, the proposed model not only outperforms existing approaches in terms of computational efficiency but also presents a promising avenue for improving performance metrics such as accuracy, precision, and recall when compared similar studies as shown in Table 33.

5. Conclusion

5.1. Recapitulation

In conclusion, this study deals with the spread and the difficult challenge of the electricity theft in the non-smart grid environment via the development of the robust deep-learning model that runs on the portable hardware. Adjusting to monthly customers' readings for data reactivity, the proposed model in addition implements advocated methods such as scaling, PCA, t-SNE, and sampling methods like RUS, SMOTE, and ROS in order to maximize performance. The carefully done exam enrolled carry out of a certain level of accuracy of 91 % with this all having a precision, recall, and F1-score of 91 %. The realization of the above mentions outcomes only goes to demonstrate the validity of the model as far as it correctly figures out cases of electricity theft while at the same time curbing the cost of electricity while ensuring reliability of the supply of electricity for utility providers on the global level.

Significance of this research is that it helps in fight the serious issue – electricity theft, which is affecting mostly utility companies but also customers around the world. A deep learning model that is swift and customized for non-smart grids could be the answer to this study's quest to tackle the problem of regulatory non-compliance, and consequently enhance the safety of power supply and reduce financial losses suffered by utility providers. Besides the fact that the mixing of more improved techniques like feature engineering and resampling some of the features of the model makes it more powerful and robust, thus the possibilities of applying it in the real world are raised.

Comparing results with other similar studies demonstrates that the accuracy, precision, recall, and F1-score of the proposed system are greater than those of the other ones. The approach used by existing methods such as wide & deep CNN, machine learning pipelines, and the hybrid approach is now vastly improved by this proposed approach and has shown to perform far better, especially when it

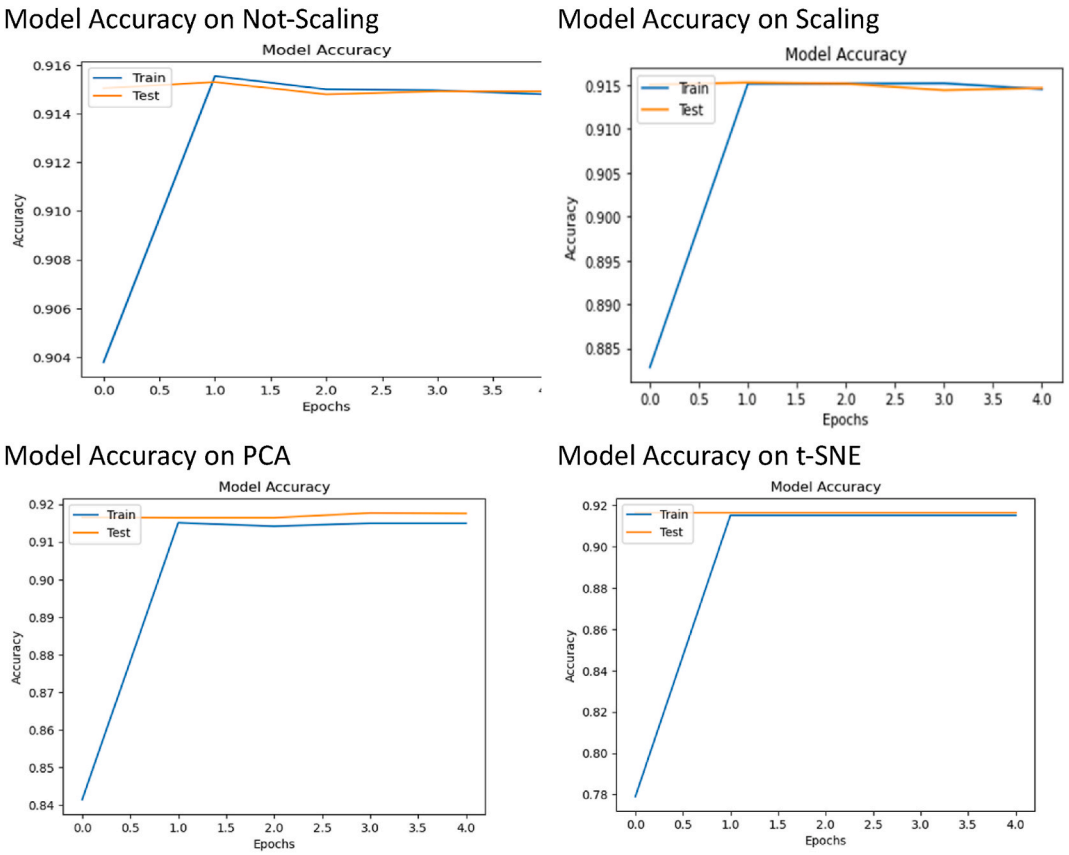

Fig. 6. Train and Test Accuracies during Epochs using Imbalanced Dataset.

balances the detection of theft as well as other crimes. For instance, the wide and deep CNN gained 78 %, a hybrid approach got 88 %, and the MLP-GRU approach ended up with 81 % precision. Nevertheless, with the suggested model's accuracy of 91 %, it has surpassed the aforementioned results and stood better regarding electricity theft detection. This ensures that the value of the model is highlighted, and the research is characterized as the model with a promising solution for utility providers who want to enhance the detection capabilities of theft in non-smart grid systems.

5.2. Future work

Consequently, we would suggest investigating certain major directions for future research where future model development can be possible thus making them more effective and applicable. Firstly, using algorithms like machine learning can make the process faster and more efficient. Also, the use of data sources like weather patterns or customer behavior can work together to further reveal the tendency of thefts. On top of that, using sophisticated anomaly detection tools and real-time electricity data from smart meters could be a plus in terms of developing an adaptive model that is capable of reacting quickly to changes in the environment. More global pilot runs would be required for this model to be validated in various geographical regions and the utility infrastructure. The results will be essential for gaining insights into its effectiveness in the real world and whether it is scalable. In the concluding part, it is recommended that we visualize the infusion of advanced technologies such as blockchain and IoT devices that would be an added shield against fraudulence. As a means to this end, future work can bring us closer to the pinnacle of innovation in electricity theft detection administration and eventually help create more robust and eco-friendly energy systems.

Data availability statement

Data is available on request.

CRediT authorship contribution statement

Sheikh Muhammad Saqib: Software, Resources, Methodology. **Tehseen Mazhar:** Writing – review & editing, Writing – original draft. **Muhammad Iqbal:** Resources, Investigation. **Tariq Shahazad:** Visualization, Validation, Investigation. **Ahmad Almogren:**

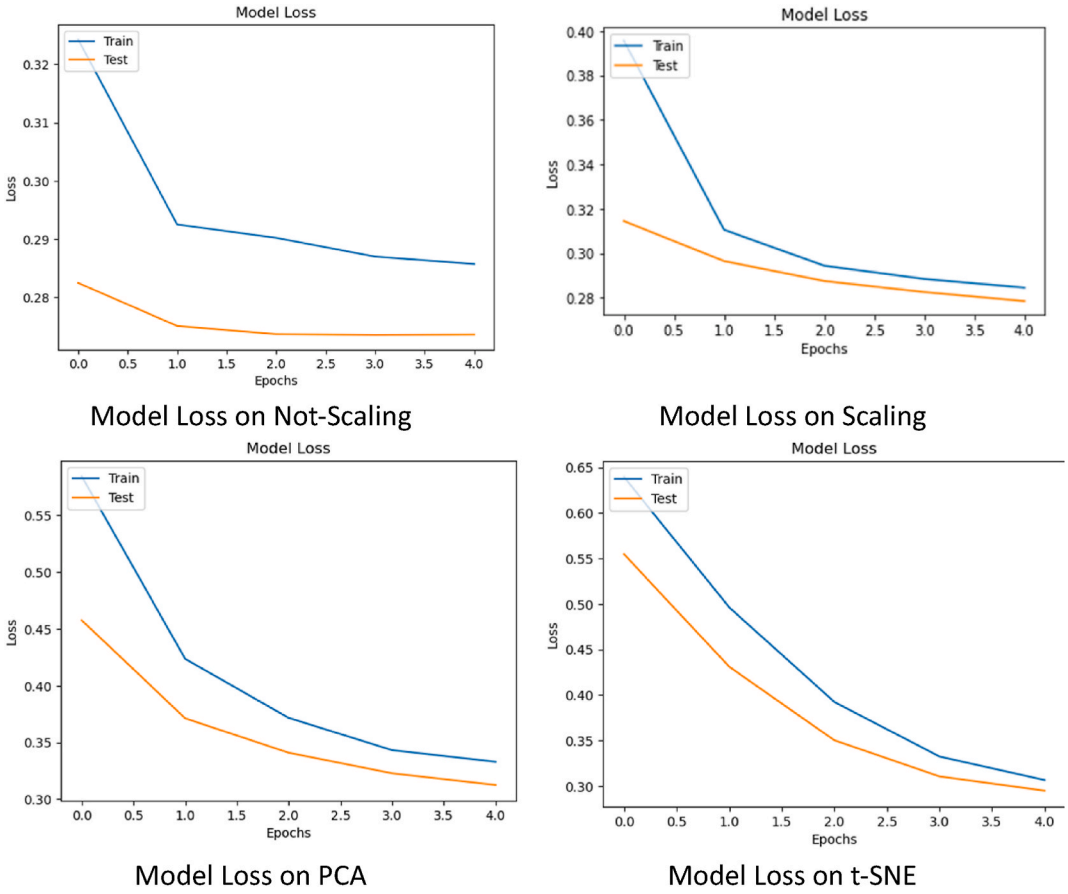

Fig. 7. Train and Test Loses during Epochs using Imbalanced Dataset.

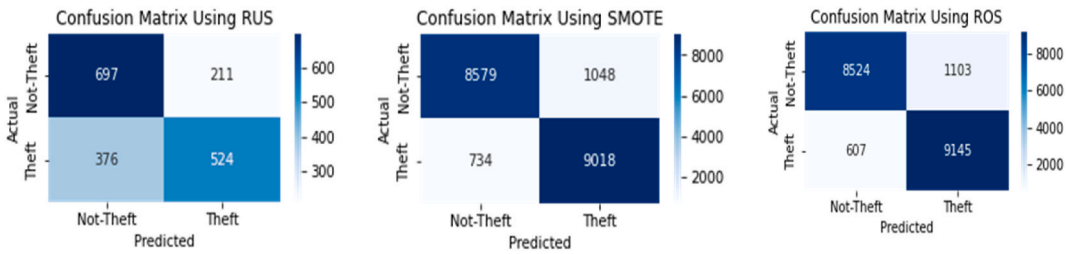

Fig. 8. Confusion matrix on Balanced Dataset.

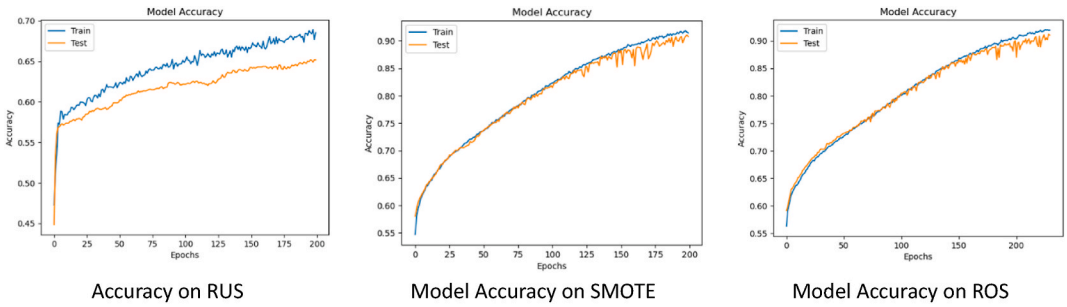

Fig. 9. Train and Test Acracies during Epochs using Balanced Dataset.

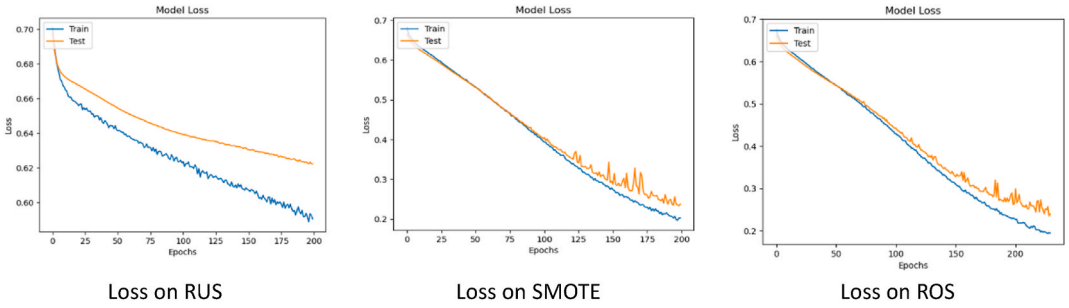

Fig. 10. Train and Test Loses during Epochs using Balanced Dataset.

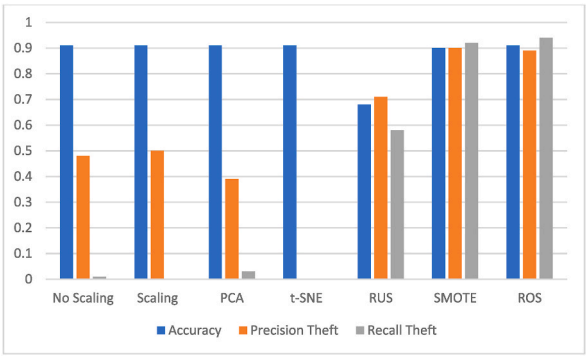

Fig. 11. Accuracy, precision and recall of theft class.

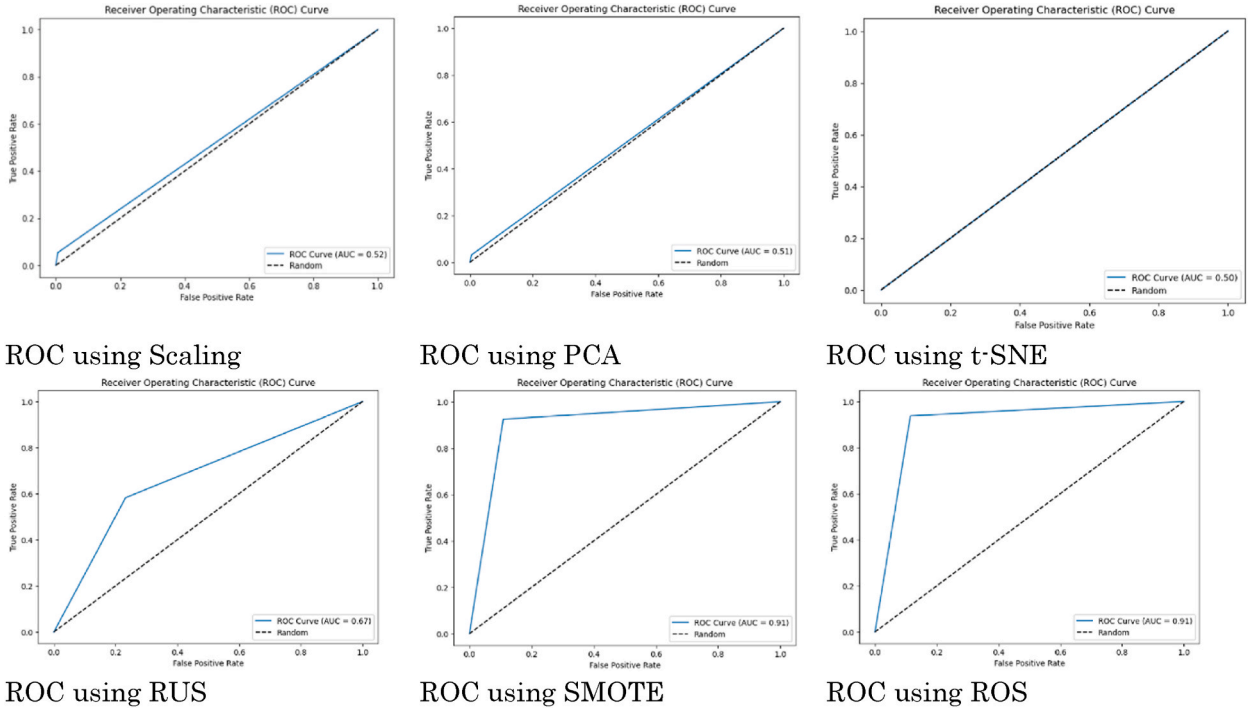

Fig. 12. Roc curves for all models.

Table 33
Comparison with similar studies.

Models	Accuracy	Precision	Recall	F1-Score
Wide & Deep CNN [8]	78 %	–	–	–
Pipeline in Machine Learning [27]	89 %	85 %	88 %	–
light-GBM [40]	84 %	–	–	–
Model [41]	66 %	–	–	45 %
Hybrid Approach [42]	88 %	–	–	–
SVM [7]	81 %	–	–	–
Ensemble Learning and Prototype Learning [43]	89 %	–	–	–
MLP-GRU [44]	81 %	89 %	82 %	85 %
Proposed Model: Random-Over-Sampling	91 %	91 %	91 %	91 %

Visualization, Resources, Investigation. **Khmaies Ouahada:** Resources, Project administration, Funding acquisition. **Habib Hamam:** Writing – review & editing, Visualization, Validation.

Declaration of competing interest

The authors have no conflict of interest.

List of Abbreviations

PCA	Principal Component Analysis,
t-SNE	t-distributed Stochastic Neighbor Embedding,
UMAP	Uniform Manifold Approximation and Projection
RUS	Random-Under-Sampler,
SMOTE	Synthetic Minority Over-sampling Technique
ROS	Random-Over-Sampler
NTLs	Non-Technical Losses
DL	Deep Learning
ML	Machine Learning
SVM	Support Vector Machine
CNN	Convolutional Neural Network
LSTM	long short-term memory
MLP	Multi-Layer Perceptron
GRU	Gated Recurrent Units
ETD	Electricity Theft Detection
SQL	structured query language
RF	Random Forest
TDTLM	Temperature-dependent theft detection using load monitoring
BHA	Binary hole Algorithm
OPF	Optimal Power Flow
SETS	Smart Energy Theft System
SMA	Simple Moving Average
PER-AutoRL	Prioritized experience replay automated reinforcement learning
TCN	Temporal convolutional network
AdaBoost	Adaptive boosting algorithm
NTL	Non-technical loss
NAS	Neural architecture search
DNOs	Distribution network operators
DBSCAN	Spatial Clustering of Applications with Noise
DRL	Deep reinforcement learning
ROC	Receiver Operating Characteristic
List of Variables	
Training Set	
Test Set	
Learning Rate	
Momentum	
Validation_split	
Batch Size	
Total Batch	
Loss Function	
Not-Theft (0)	
Theft (1)	

Acknowledgements

This work is supported by the research fund of the University of Johannesburg, South Africa, and supported by King Saud University, Riyadh, Saudi Arabia, through Researchers Supporting Project number (RSP2024R184).

References

[1] L.J. Lepolesa, S. Achari, L. Cheng, Electricity theft detection in smart grids based on deep neural network, IEEE Access 10 (2022) 39638–39655, https://doi.org/10.1109/ACCESS.2022.3166146.
[2] P. Glauner, P. Valtchev, C. Glaeser, N. Dahringer, R. State, D. Duarte, Non-Technical Losses in the 21st Century: Causes, Economic Effects, Detection and Perspectives (May) (2018) 1–9 [Online]. Available: https://www.researchgate.net/publication/325297875.
[3] B.K. Hammerschmitt, et al., Non-technical losses review and possible methodology solutions, in: Proc. - 2020 6th Int. Conf. Electr. Power Energy Convers. Syst. EPECS 2020, 2020, pp. 64–68, https://doi.org/10.1109/EPECS48981.2020.9304525.
[4] Smart meters help reduce electricity theft, increase safety [Online]. Available: https://www.bchydro.com/news/conservation/2011/smart_meters_energy_theft.html, 2011.
[5] H. Jiang, K. Wang, Y. Wang, M. Gao, Y. Zhang, Energy big data: a survey, IEEE Access 4 (2016) 3844–3861, https://doi.org/10.1109/ACCESS.2016.2580581.
[6] T. Mazhar, et al., Analysis of cyber security attacks and its solutions for the smart grid using machine learning and blockchain methods, Future Internet (2023), https://doi.org/10.3390/fi15020083.
[7] I. Petrlik, P. Lezama, C. Rodriguez, R. Inquilla, J.E. Reyna-González, R. Esparza, Electricity theft detection using machine learning, Int. J. Adv. Comput. Sci. Appl. 13 (12) (2022) 420–425, https://doi.org/10.14569/IJACSA.2022.0131251.
[8] Z. Zheng, Y. Yang, X. Niu, H.N. Dai, Y. Zhou, Wide and deep convolutional neural networks for electricity-theft detection to secure smart grids, IEEE Trans. Ind. Informatics 14 (4) (2018) 1606–1615, https://doi.org/10.1109/TII.2017.2785963.
[9] S.M. Saqib, F.M. Kundi, Semi supervised method for detection of Ambiguous Word and creation of Sense : using WordNet, Int. J. Adv. Comput. Sci. Appl. 9 (11) (2018) 353–359.
[10] M. Khadhraoui, et al., Survey of BERT-base models for Scientific text classification: Covid19 case study, Appl. Sci. (2022), https://doi.org/10.3390/app12062891.
[11] C. Mondol, et al., Early prediction of Chronic Kidney disease: a comprehensive performance analysis of deep learning models, Algorithms 15 (9) (2022), https://doi.org/10.3390/a15090308.
[12] S. Guefrachi, et al., Automated diabetic retinopathy Screening using deep learning, Multimed. Tool. Appl. 81 (2024), https://doi.org/10.1007/s11042-024-18149-4.
[13] A. Raza, et al., A hybrid deep learning-based approach for Brain tumor classification, Electronics (2022), https://doi.org/10.3390/electronics11071146.
[14] Y. Yu, K. Zhang, L. Yang, D. Zhang, Fruit detection for strawberry harvesting robot in non-structural environment based on Mask-RCNN, Comput. Electron. Agric. 163 (2019), https://doi.org/10.1016/j.compag.2019.06.001.
[15] T. Nasir, et al., Optimal scheduling of Campus microgrid considering the electric vehicle integration in smart grid, Sensors (2021), https://doi.org/10.3390/s21217133.
[16] T. Mazhar, et al., The role of ML, AI and 5G technology in smart energy and smart buildings management, Electronics (2022), https://doi.org/10.3390/electronics11233960.
[17] B.K. Yousafzai, et al., Student-performulator: Student academic performance using hybrid deep neural network, Sustainability (2021), https://doi.org/10.3390/su13179775.
[18] S. Li, et al., Electricity theft detection in power grids with deep learning and random forests, J. Electr. Comput. Eng. (2019) 1–12.
[19] S. Sahoo, et al., Electricity theft detection using smart meter data, in: 2015 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference, ISGT), Washington, DC, USA, 2015, pp. 1–5.
[20] M. Adil, et al., LSTM and batbased RUSBoost approach for electricity theft detection, Appl. Sci. 10 (12) (2020) 4378.
[21] R. Punmiya, S. Choe, Energy theft detection using gradient boosting theft detector with feature engineering-based preprocessing, IEEE Trans. Smart Grid 10 (2) (2019) 2326–2329.
[22] N. Javaid, A PLSTM, AlexNet, and ESNN-based ensemble learning model for detecting electricity theft in smart grids, IEEE Access 9 (2021) 162935–162950.
[23] S.C. Yip, W.N. Tan, C.K. Tan, M.T. Gan, K.S. Wong, An anomaly detection framework for identifying energy theft and defective meters in smart grids, Int. J. Electr. Power Energy Syst. 101 (2018) 189–203, https://doi.org/10.1016/j.ijepes.2018.03.025.
[24] M.N. Hasan, et al., Electricity theft detection in smart grid systems: a CNN-LSTM based approach, Energies 12 (17) (2019) 3310.
[25] F.A. Bohani, A. Suliman, M. Saripuddin, S.S. Sameon, N.S. Md Salleh, S. Nazeri, A comprehensive analysis of supervised learning techniques for electricity theft detection, J. Electr. Comput. Eng. (2021) 1–10, 2021.
[26] R. Razavi, A. Gharipour, M. Fleury, I.J. Akpan, A practical feature-engineering framework for electricity theft detection in smart grids, Appl. Energy 238 (2019) 481–494.
[27] M. Anwar, N. Javaid, A. Khalid, M. Imran, M. Shoaib, Electricity theft detection using pipeline in machine learning, 2020, Int. Wirel. Commun. Mob. Comput. IWCMC (2020) 2138–2142, https://doi.org/10.1109/IWCMC48107.2020.9148453, 2020, no. February 2024.
[28] M. Nabil, M. Ismail, M.M.E.A. Mahmoud, W. Alasmary, E. Serpedin, PPETD: privacy-preserving electricity theft detection scheme with load monitoring and billing for AMI networks, IEEE Access 7 (2019) 96334–96348.
[29] P. Mohassel, Y. Zhang, SecureML: a system for scalable privacy-preserving machine learning, in: 2017 IEEE Symposium on Security and Privacy (SP), 2017, pp. 19–38. San Jose, CA, USA.
[30] S. Hussain, et al., A novel feature engineered-CatBoost-based supervised machine learning framework for electricity theft detection, Energy Rep. 7 (2021) 4425–4436.
[31] W. Li, T. Logenthiran, V.-T. Phan, W.L. Woo, A novel smart energy theft system (SETS) for IoT-based smart home, IEEE IoT J 6 (3) (2019) 5531–5539.
[32] M.M. Buzau, J. Tejedor-Aguilera, P. CruzRomero, A. Gómez-Expósito, Detection of non-technical losses using smart meter data and supervised learning, IEEE Trans. Smart Grid 10 (3) (2018) 2661–2670.
[33] B. Coma-Puig, J. Carmona, Bridging the gap between energy consumption and distribution through non-technical loss detection, Energies 12 (9) (2019) 1748.
[34] L.A.M. Pereira, et al., Multilayer perceptron neural networks training through charged system search and its application for non-technical losses detection, in: 2013 IEEE PES Conference on Innovative Smart Grid Technologies (ISGT Latin America), 2013, pp. 1–6. Sao Paulo, Brazil.
[35] Z. Aydin, V.C. Gungor, A novel feature design and stacking approach for non-technical electricity loss detection, in: 2018 IEEE Innovative Smart Grid Technologies - Asia (ISGT Asia), 2018, pp. 867–872. Singapore.
[36] B.C. Costa, B.L.A. Alberto, A.M. Portela, W. Maduro, E.O. Eler, Fraud detection in electric power distribution networks using an ANN-based knowledge-discovery process, Int. J. Artif. Intell. Appl. 4 (6) (2013) 17.
[37] P. Jokar, N. Arianpoo, V.C.M. Leung, Electricity theft detection in AMI using customers' consumption patterns, IEEE Trans. Smart Grid 7 (1) (2015) 216–226.
[38] K.S.Y. Nagi, S.K. Tiong, S.K. Ahmed, M. Mohamad, Nontechnical loss detection for metered customers in power utility using support vector machines, IEEE Trans. Power Delivery 25 (2) (2009) 1162–1171.
[39] N. Ding, H. Ma, H. Gao, Y. Ma, G. Tan, Real-time anomaly detection based on long short-term memory and Gaussian mixture model, Comput. Electr. Eng. 79 (2019) 106458.

[40] S.V. Oprea, A. Bâra, Machine learning classification algorithms and anomaly detection in conventional meters and Tunisian electricity consumption large datasets, Comput. Electr. Eng. 94 (2021), https://doi.org/10.1016/j.compeleceng.2021.107329.
[41] R.M.R. Barros, E.G. da Costa, J.F. Araujo, Evaluation of classifiers for non-technical loss identification in electric power systems, Int. J. Electr. Power Energy Syst. 132 (2021), https://doi.org/10.1016/j.ijepes.2021.107173.
[42] E.U. Haq, J. Huang, H. Xu, K. Li, F. Ahmad, A hybrid approach based on deep learning and support vector machine for the detection of electricity theft in power grids, Energy Rep. 7 (2021) 349–356, https://doi.org/10.1016/j.egyr.2021.08.038.
[43] X. Sun, et al., Electricity theft detection method based on ensemble learning and prototype learning, J. Mod. Power Syst. Clean Energy 12 (1) (2024) 213–224, https://doi.org/10.35833/MPCE.2022.000680.
[44] H. Iftikhar, et al., Electricity theft detection in smart grid using machine learning, Front. Energy Res. 12 (2024), https://doi.org/10.3389/fenrg.2024.1383090.
[45] Y. Li, R. Wang, Z. Yang, Optimal scheduling of isolated microgrids using automated reinforcement learning-based multi-period forecasting, IEEE Trans. Sustainable Energy 13 (1) (2021) 159–169.
[46] S.K. Gunturi, D. Sarkar, Ensemble machine learning models for the detection of energy theft, Electr. Power Syst. Res. 192 (2021) 106904.
[47] N.J. Ullah, A.S. Yahaya, T. Sultana, F.A. Al-Zahrani, F. Zaman, A hybrid deep neural network for electricity theft detection using intelligent antenna-based smart meters, Wireless Commun. Mobile Comput. (2021) 1–19, 2021.
[48] U. Khan, et al., A stacked machine and deep learning-based approach for analyzing electricity theft in smart grids, IEEE Trans. Smart Grid 13 (2) (2021) 1633–1644.
[49] Z. Yan, H. Wen, Comparative study of electricity-theft detection based on gradient boosting machine, in: 2021 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), 2021, pp. 1–6. Glasgow, UK.
[50] Z. Qu, et al., A combined genetic optimization with AdaBoost ensemble model for anomaly detection in buildings electricity consumption, Energy Build. 248 (2021) 111193.
[51] Y. Yao, et al., A hybrid method for electricity theft detection, in: 2021 6th Asia Conference on Power and Electrical Engineering, ACPEE), Chongqing, China, 2021, pp. 436–440.
[52] P. Mohassel, Y. Zhang, SecureML: a system for scalable privacy-preserving machine learning, in: 2017 IEEE Symposium on Security and Privacy (SP), 2017, pp. 19–38. San Jose, CA, USA.
[53] S. Hussain, et al., A novel feature engineered-CatBoost-based supervised machine learning framework for electricity theft detection, Energy Rep. 7 (2021) 4425–4436.
[54] S. Zidi, et al., Theft detection dataset for benchmarking and machine learning based classification in a smart grid environment, J. King Saud Univ. Comput. Inf. Sci. 35 (1) (2023) 13–25.
[55] K. Fei, Q. Li, C. Zhu, Nontechnical losses detection using missing values' pattern and neural architecture search, Int. J. Electr. Power Energy Syst. 134 (2022) 107410.
[56] Y.G.S. Lee, I. Sim, S.H. Kim, D.I. Kim, J.Y. Kim, Non-technical loss detection using deep reinforcement learning for feature cost efficiency and imbalanced dataset, IEEE Access 10 (2022) 27084–27095, https://doi.org/10.1109/ACCESS.2022.3156948.
[57] S.V. Oprea, A. Bâra, Feature engineering solution with structured query language analytic functions in detecting electricity frauds using machine learning, Sci. Rep. 12 (2022) 3257, https://doi.org/10.1038/s41598-022-07337-7.
[58] https://www.kaggle.com/datasets/akshatrailaddha/electricity-theft-detection-dataset.
[59] A. Khattak, A. Habib, M.Z. Asghar, F. Subhan, I. Razzak, A. Habib, Applying deep neural networks for user intention identification, Soft Comput. 25 (3) (2021) 2191–2220, https://doi.org/10.1007/s00500-020-05290-z.
[60] H. Ahmad, M.U. Asghar, M.Z. Asghar, A. Khan, A.H. Mousavi, A hybrid deep learning technique for Personality Trait classification from text, IEEE Access 9 (2021) 146214–146232, https://doi.org/10.1109/ACCESS.2021.3121791.