IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 9 December 2024, accepted 28 December 2024, date of publication 10 January 2025, date of current version 17 January 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3528079

RESEARCH ARTICLE

Advanced Fraud Detection: Leveraging K-SMOTEENN and Stacking Ensemble to Tackle Data Imbalance and Extract Insights

NURAFNI DAMANIK[1] **AND CHUAN-MING LIU**[2]**, (Member, IEEE)**
[1]College of Electrical Engineering and Computer Science, National Taipei University of Technology, Taipei 10608, Taiwan
[2]Department of Computer Science and Information Science, National Taipei University of Technology, Taipei 10608, Taiwan

Corresponding author: Chuan-Ming Liu (cmliu@ntut.edu.tw)

This work was supported in part by the National Science and Technology Council, Taiwan, ROC, under Grant NSTC 113-2221-E-027-051.

ABSTRACT This study proposes an innovative solution for credit card fraud detection, utilizing a stacking ensemble of machine learning classifiers enhanced with sophisticated data resampling techniques. The model demonstrates exceptional performance, achieving an F1-score of 0.92, precision of 0.95, recall of 0.88, an AUPRC of 0.96, and a perfect ROC-AUC of 1.00. These results significantly outperform standalone models like XGBoost and Decision Tree, showcasing the strength of the proposed approach. The study addresses two critical challenges, class imbalance and overfitting, by employing K-means SMOTEENN to balance minority class representation while reducing synthetic data noise, thus lowering the risk of overfitting. Additionally, the stacking ensemble's integration of diverse classifiers produces a generalized decision boundary, enhancing model robustness in real-world scenarios. This ensures a precise fraud detection mechanism without compromising sensitivity. Furthermore, the proposed model incorporates Explainable AI (XAI) techniques to enhance interpretability and trust. The study identifies key features driving model predictions by leveraging Local Interpretable Model-Agnostic Explanations (LIME), illustrating how base learners and the meta-learner contribute to the final decision. This transparency bolsters stakeholder confidence and provides actionable insights for financial institutions. Overall, the proposed approach marks a notable step in protecting financial transactions from fraud.

INDEX TERMS K-MEANS, SMOTEENN, credit card, fraud detection, stacking ensemble, imbalance technique.

I. INTRODUCTION

Financial transactions have experienced substantial growth in recent years, driven primarily by the expansion of financial institutions and the widespread adoption of online e-commerce platforms. This increase reflects the rising integration of digital payment systems and the increased reach of internet-based economic operations [1]. The global financial landscape saw a significant disruption in 2015, with approximately $21.84 billion in damages attributed to credit card theft. By 2019, the amount had increased to $28.65 billion, indicating a notable rise of $6.81 billion in only four years [2]. According to the 2023 Nilson Report, the banking industry has experienced a substantial increase in card fraud in raw numbers and the overall amount of card transactions. Card fraud has increased significantly from $18.11 billion in 2014 to an estimated $43.47 billion by 2028. Although the total cash amount of card fraud has increased dramatically, the proportion of card fraud relative to transaction volume, measured in cents per $100, reached its highest point at 7.2 cents in 2016 and has decreased. It is anticipated to reach 6.4 cents by 2028 [1]. This amount represents a significant

The associate editor coordinating the review of this manuscript and approving it for publication was Ali Kashif Bashir.

© 2025 The Authors. This work is licensed under a Creative Commons Attribution 4.0 License.
For more information, see https://creativecommons.org/licenses/by/4.0/

financial loss that is steadily growing. Practical strategies for detecting and preventing Fraud are essential to reduce these losses. Although these approaches are frequently successful, they do not completely safeguard against credit card fraud.

Simultaneously, machine learning (ML) has been utilized to create various credit card fraud detection systems [3]. Previous and ongoing research has commonly employed several machine-learning approaches regarding credit card fraud. The mentioned techniques Support Vector Machines, Decision Trees, Bayesian Networks, Genetic Algorithms, Gradient Boosting techniques, Artificial Neural Networks, and various other methods [4]. Ensemble learning was proposed to improve the quality of the classification step. Ensemble methods in machine learning have recently received considerable attention in several domains. This is because they can create learners by combining multiple classifiers, which allows them to handle complex and high-dimensional data effectively. These methods have demonstrated great accuracy in numerous real-world problems [5]. The dataset poses a significant difficulty in detecting fraudulent transactions. The likelihood of fraudulent activities within credit card transactions is significantly lower relative to legitimate purchases. As a result, the dataset demonstrates a pronounced imbalance in label distribution [6]. One way to address this issue is by oversampling the minority class, which involves creating synthetic instances during the training process, resampling the dataset, or minimizing the amount of the majority class using the undersampling technique before applying it to the model.

Instances of ensemble models encompass bagging, boosting, and stacking. Stacking is distinct from bagging and boosting, employing a singular meta-model to amalgamate predictions derived from various base models [7].

Additionally, this work provides other significant contributions:

a. An advanced credit card fraud detection system using an ensemble model is being developed. This model combines base learners Decision Tree and Random Forest, with RF acting as the meta-learner. It compares with other machine learning models such as Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), XGBoost (XGB), and lightGBM (LGBM).
b. Resolving the class imbalance issue in credit card data by employing the K-SMOTEENN approach creatively.
c. Develop a model that effectively handles imbalanced data while minimizing overfitting on large datasets.
d. Performing a comparative study of this method compared to modern credit card fraud detection approaches using the Paysim dataset.
e. Explainable AI techniques, specifically Local Interpretable Model-Agnostic Explanations, are integrated into the framework to enhance interpretability and better understand the decision-making process of the stacking ensemble model using (LIME).

The rest of the paper is organized as follows: Section II focuses on the literature review of fraud detection, explicitly addressing the approaches for handling unbalanced data and evaluating the efficacy of imbalanced model classification—section III: Materials and Methods. Section IV focuses on the experimental results and discusses integrating several cutting-edge methods. Section V is dedicated to the paper's conclusion and future work.

II. RELATED WORK

Credit card fraud detection has been an ongoing area of interest for many researchers. Supervised learning methods, including machine learning and deep learning, have been used to identify fraudulent credit card activity. Researchers have introduced two approaches to mitigate the repercussions of imbalanced credit card data on classification outcomes. The first involves enhancing the classifier by choosing a higher-performing classification model, while the second addresses the imbalance in the data itself [8]. The paper conducts a comprehensive systematic review of credit card fraud detection, focusing on applying machine learning (ML) and deep learning (DL) techniques from 2019 to 2021. It identifies a growing need for efficient fraud detection systems due to the rise in cyberattacks targeting the banking industry. The authors systematically review 181 articles, categorizing the ML and DL techniques into supervised, unsupervised, and semi-supervised approaches. The study explores commonly used algorithms like Random Forest (RF), Support Vector Machines (SVM), and Neural Networks (NN), as well as ensemble techniques such as boosting and stacking. This shows that credit card fraud detection using machine learning is still the focus of current research [9]. The paper uses several machine learning algorithms within their soft voting ensemble, specifically logistic regression, random forest, XGBoost, and multilayer perceptron. These models are combined to create the ensemble approach, leveraging the strengths of each to improve overall prediction accuracy [10].

Furthermore, Aung et al. explored using the Random Forest (RF) algorithm to detect credit card fraud [11]. The authors outlined their research methodology, presented experimental results, and shared insights into this approach's potential advantages and limitations. One of the primary benefits of using Random Forest for imbalanced datasets, such as those encountered in credit card fraud detection, is its robustness against overfitting. This is accomplished through its ensemble learning strategy, which constructs multiple decision trees to form a ''forest.'' By synthesizing predictions from these varied trees, Random Forest mitigates the risk of overfitting the training data, which is especially advantageous when managing imbalanced datasets. Furthermore, Random Forest offers a reliable estimate of the generalization error, improving its capability to make precise predictions on new, unseen data despite the class imbalance commonly found in fraud detection tasks [12].

Moreover, resampling techniques can be utilized to address the issue of imbalanced data. Here are some resampling methods that have been studied by Zhu et al.: NUS, RUS, SMOTE, ADASYN, SMOTE + Tomek Link, and ENN [13].

K-means SMOTE effectively addresses within-class imbalance by clustering datasets into sub-clusters, balancing minority samples. This nuanced approach recognizes the varying importance of minority samples, serving as a foundation for advanced techniques. Studies have shown that K-means SMOTE outperforms traditional oversampling methods, improving classification performance on imbalanced datasets [14]. El-Naby et al. [15] proposed using Edited Nearest Neighbor (ENN) to address imbalanced datasets combined with SMOTE in credit card fraud detection. Their study demonstrated that SMOTE-ENN effectively balances the dataset by removing misclassified and eliminating noise instances from the majority class and oversampling the minority class, thereby enhancing model performance. The authors found that this approach reduced bias towards the majority class and improved the model's ability to detect fraudulent transactions.

The study presents an efficient approach for detecting credit card fraud using advanced machine learning techniques. The authors address the unbalanced data issue by adopting Bayesian optimization and propose using weight-tuning hyperparameters as a preprocessing step. They explore the effectiveness of CatBoost, XGBoost, and LightGBM and introduce a majority-voting ensemble learning approach to enhance performance. The study also incorporates deep learning for hyperparameter adjustment and fine-tuning. Extensive experiments on real-world data demonstrate that the combination of LightGBM and XGBoost achieves superior performance, with an MCC value of 0.79 and an F1-score of 0.79. The deep learning method outperforms individual algorithms and ensemble learning, achieving an MCC of 0.81 and an F1-score of 0.81. The proposed methods, including class weight tuning and Bayesian optimization, surpass existing state-of-the-art techniques in the literature [16].

Recent research in credit card fraud detection has focused on addressing the persistent challenge of class imbalance, which significantly impacts model performance. In this context, Alamri and Ykhlef [17] proposed a novel hybrid approach that combines undersampling and oversampling techniques to improve fraud detection accuracy. Their method utilizes Tomek links for undersampling to eliminate noisy and overlapping instances while employing BIRCH clustering with Borderline-SMOTE (BCBSMOTE) to achieve a more balanced distribution of legitimate and fraudulent transactions. The authors implemented a Random Forest (RF) classifier as the core detection model and extensively tested the PaySim synthetic dataset. Their results demonstrated that this hybrid sampling method significantly outperformed traditional techniques, achieving an F1-score of 85.20%, precision of 81.27%, and AUPRC of 72.77%. Notably, the model maintained a high accuracy of 99.95% while improving precision and recall, critical metrics in fraud detection scenarios [17]. Notably, the performance of the achieved metrics indicates significant room for enhancement. These relatively low values present crucial challenges in the context of fraud detection. A low AUPRC suggests the model struggles to maintain high precision across various recall levels, essential to effective fraud identification. The suboptimal recall indicates that the model fails to catch all fraudulent transactions, potentially allowing financial crimes to go undetected.

Similarly, the precision metric implies a considerable number of false positives, which could lead to legitimate transactions being flagged as fraudulent. Ensemble learning has emerged as a powerful approach to address the challenges of class imbalance in machine learning. Recent studies, such as the work by Khan et al. l [7], have demonstrated the superiority of ensemble methods over individual machine learning algorithms in handling imbalanced datasets. Ensemble techniques, including bagging, boosting, and stacking, combine multiple models to create robust classifiers that outperform standalone algorithms.

III. MATERIAL AND METHOD

This part describes the credit card dataset utilized in the study and thoroughly discusses the several algorithms and techniques that went into creating the suggested approach for detecting credit card fraud.

A. CREDIT CARD FRAUD DATASET

This research utilizes a synthetic dataset of credit card transactions created using the PaySim simulator. The scarcity of publicly accessible data on financial services is especially evident in the rapidly expanding field of mobile money transactions. As a result, the availability of such extensive simulated data is invaluable for improving fraud detection algorithms. Various research efforts have focused on financial datasets containing values for fraud detection. Given the inherent privacy of financial transactions, the current issue is not worsened by any datasets that are publicly accessible [18]. PaySim leverages aggregated information from a private dataset to create a synthetic dataset that mirrors common transactional behaviors. It subsequently incorporates fraudulent activity into the data to evaluate the effectiveness of fraud detection systems. Simulation involves utilizing a subset of actual transactions extracted from a collection of financial records spanning one month, namely from the mobile money services of an African country, to replicate mobile money transactions. The original documents came from a global company that runs a mobile money service accessible in more than 14 nations [18]. The dataset comprises over six million records and includes 11 distinct attributes. The principle component analysis (PCA) transformation of the Kaggle dataset, employed by most researchers, leaves just time and amount characteristics, which is the primary motivation behind the synthetic dataset technique. Due to this limitation, the dataset's attributes cannot fully capture consumer behavior, making it necessary to include additional attributes, such as those provided by a synthetic dataset [18]. The dataset employed in this study contains the following attributes:

Step: This variable represents a discrete unit of time, with each step equating to one hour.

Type: Specifies the category of the online transaction being executed.

Amount: Denotes the monetary value involved in the transaction.

nameOrig: Specifies the customer who initiated the transaction.

oldbalanceOrg: This represents the initiating customer's account balance before the transaction.

newbalanceOrig: Represents the initiating customer's account balance after the transaction.

nameDest: specifies the transaction's receiver.

oldbalanceDest: This is the recipient's initial account balance before the transaction.

newbalanceDest: Indicates the recipient's account balance following the transaction.

isFraud:A binary indication of fraudulent transactions, with '1' representing a fraudulent transaction and '0' representing a non-fraudulent transaction.

The ''is-Fraud'' attribute identifies instances where malicious actors attempt to gain access to customer accounts, transfer funds to different accounts, and withdraw money from the system. The ''isFlaggedFraud'' attribute, designed to flag illegal activities, explicitly marks any transaction that involves transferring more than 200,000 USD in a single instance [17].

B. DATA CLEANING

Cleaning prepares data by removing outliers, biased data, and missing values. Before training the model, the data was preprocessed to remove any unwanted noise. Random sampling was conducted in addition to addressing missing values, selecting informative features, and structuring the data [19].

Missing values: One of the data cleaning procedures is determining whether the dataset contains any null or missing values using the isnull () method. This function returns a DataFrame object with all values changed to either ''True'' or ''False'' (depending on whether the value is null or not). Null or missing values were discovered when the PaySim dataset was used.

Duplicate values: Once the null values have been verified, the duplicated () method can be used to check for duplicated values. Using this approach, duplicate rows are found throughout the dataset one by one, and the Boolean values for each row are returned. The PaySim dataset did not turn up duplicate values or false-value results.

C. EXPLORATORY DATA ANALYSIS

Exploratory Data Analysis (EDA) provides valuable insights for choosing appropriate analytical tools and preparing the data for further modeling. Exploratory Data Exploratory data analysis (EDA) is a preferred approach by skilled analysts to understand a given dataset. The objective is to elucidate the overall framework of the data, acquire concise, descriptive summaries, and potentially generate insights for a more intricate examination. This casual, investigative analysis aids in comprehending the fundamental patterns of the material [20].

There are 636,2620 transaction entries and 11 attributes in the PaySim dataset. Figure 1 illustrates that there are 6,35,4407 legitimate transactions and 8,213 fraudulent transactions. The dataset exhibits a significant imbalance, necessitating the implementation of sampling techniques to enhance the efficacy of the detection model. Figure 2 demonstrates a strong positive correlation (close to 1) between the newbalanceOrig and oldbalanceOrig columns. Hence, it is necessary to remove one of these columns during the data preprocessing step. Furthermore, it is recommended that the isFlaggedFraud column be eliminated as it does not substantially contribute to the identification of fraudulent transactions due to the inadequacy of the flagging method. The transaction quantity plays a vital role in identifying possible instances of Fraud. Accounts with more significant initial balances are more susceptible to fraudulent activity, and the timing of transactions also affects the probability of Fraud. The PaySim dataset consists of five distinct types of transactions. According to Figure 3, Cash-out and payment transactions are the most frequent, while debit and transfer transactions are the least frequent.

D. FEATURE ENGINEERING

Feature construction refers to identifying potentially beneficial relationships between existing features and creating new features based on these relationships. These new features are then used to represent training cases [21]. Following feature construction, it is possible to obtain additional features denoted as $An+1, An+2, \ldots, An+m$. For instance, a novel attribute Ak (where $n < k < n+m$) could be generated by applying a logical operation to A and A from the initial set. Representing length may be simplified and turned into a problem that exists in only one dimension, represented by B_1, which means the Area once B_1 is determined [22]. In our case, we combined the 'origin' and 'destination' features from a dataset to form a new feature, 'type2'. We will modify specific feature names to facilitate preprocessing before implementing a new feature, precisely type 2.

Here are some of the adjustments: nameOrig to be origin, oldbalanceOrg to be sender _old_balance, newbalanceOrig to be sender_new_balance, nameDest to be destination, oldbalanceDest to be receiver_old_balance, newbalanceDest to be receiver_new_balance, isFraud to be isfraud. This new feature captures the interaction between the two initial features through logical operations. Specifically, the code assigns the value 'CC' to 'type2' when both 'origin' and 'destination' contain 'C', 'CM' when 'origin' contains 'C' and 'destination' contains 'M', 'MC' when 'origin' contains 'M' and 'destination' contains 'C', and 'MM' when both contain 'M'. The feature construction process transforms the two-dimensional problem defined by 'origin' and 'destination' into a single, more informative feature 'type 2'. This

new feature combines the combined information of 'origin' and 'destination', simplifying the dataset and improving the effectiveness of machine learning models by providing a more complete and concise representation of the data. Subsequently, we remove the original and destination features from our feature set as they are already encompassed by type 2. Figure 4 and Figure 5 illustrate the distribution of data in type 2, indicating that the majority of transactions were conducted between customers (CC)

E. EVALUATION METRICS

To thoroughly evaluate and confirm the reliability of the credit card fraud detection model, we conducted a detailed analysis of the test data to ensure that it produced accurate outcomes based on the chosen evaluation metric. Machine learning models are typically assessed using a range of metrics, such as accuracy, precision, recall, F1-score, AUC-ROC, and the Area under the precision-recall curve [23]. The AUC-ROC calculation involves plotting the sensitivity (actual positive rate) against the complement of specificity (false positive rate) at various threshold levels. The AUPRC quantifies the trade-off between accuracy and recall across different thresholds without relying on any assumptions regarding class distribution. Those metrics can be defined in formulas (1, 2, 3, 4), respectively:

Where:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1 = 2 \times \frac{(Recall \times Precision}{Recall + Precision} \tag{4}$$

- TP: the count of correctly identified positive instances.
- TN: the count of correctly identified negative instances.
- FP: the count of incorrect positive results.
- FN: the count of instances that are incorrectly classified as unfavorable.

The evaluation involved conducting tests on the synthetic PaySim dataset using the algorithms mentioned earlier. The performance of these algorithms was compared based on various criteria, including increasing true positives, decreasing false positives and error rates, managing a substantially balanced dataset, and attaining superior accuracy and F1 scores [17].

F. ENSEMBLE LEARNING

Ensemble learning refers to using several inducers to decide supervised machine learning problems. It is a broad phrase that encompasses various methodologies [24]. This technique involves the results of machine learning systems, referred to as "weak learners" due to their subpar performance, from a set of learners to achieve reduced prediction errors (in regression) or lower error rates (in classification). Diversity is essential in the training process since the individual estimator needs to offer various patterns of generalization [25]. ML models sometimes exhibit limitations, such as significant bias, substantial volatility, and limited accuracy, and are not resistant to making mistakes [26]. Instead of depending on it, ensemble learning approaches utilize the advantages of two or more classifiers, resulting in improved accuracy compared to the individual basis classifiers [27].

Ensemble learning techniques can be categorized into three main groups: bagging, boosting, and stacking [28]. Stacking was employed in this paper, a widely used technique in ensemble learning. A high-level base learner aggregates lesser-level base learners to improve predicted accuracy [29]. The stacking model consists of the base layer and the meta layer. The base layer includes multiple base learners, each producing predictions based on the input features. These predictions are subsequently utilized as inputs for the meta layer, which employs a meta-model to integrate the outputs of the base learners [30].

Algorithm 1 Stacking Model

Require: The dataset has been prepared to ensure balance using K-SMOTEENN
1: **Begin**
2: Assign the resampled training set $X_{iresampled}$ to M[i], where i Is the index of the base learners.
3: Add W to $X_{iresampled}$
4: Repeat steps 2 and 3 to create an array of $X_{iresampled}$
5: Train M[i] on ($X_{iresampled}$)
6: The dataset trained using base models
7: Utilize the predictions given by the base models to train the meta-level model.
8: Derive prediction P utilizing the output of the final model.
9: **End**

In Algorithm 1, the symbol B represents the base model, R signifies the prediction created. By the base model, and P defines the final prediction. Firstly, the dataset should be divided into separate training and testing subsets. Following this phase, the base models complete training using the training subset.

The base models' predictions are provided as input to the meta-model. Ultimately, the model is trained on the complete dataset without any separate testing portion and is then utilized to make predictions on fresh, unseen data points.

IV. DISCUSS ON THE EXPERIMENTAL

A. DATA RESAMPLING

This part focuses on our research methodology, proposed framework, experimental setup, and performance evaluation metrics we have employed. We have explored Hybrid data-balancing approaches. This method uses the K-Means algorithm to partition the set of objects into clusters based on a specific set of criteria, where objects within the same cluster are more alike than those in different clusters [31].

K-Means algorithm partitions the training data into multiple clusters before applying SMOTE +ENN to each cluster. Since the distribution is a highly skewed and irregular phenomenon known as overlap, we utilize K-means clustering. Overlap often denotes the issue of having samples from various classes happening in the same data space region, making it more challenging to train a classifier capable of distinguishing samples from distinct classes within the overlapping region [32]. Therefore, this facilitates the ability of K-SMOTEENN to concentrate on sample data; this method is represented by Algorithm 2.

Algorithm 2 K-SMOTEENN

Input: training data S is a set of pairs $\{(x1, y1), \ldots, (x2, y2), \ldots, (xm, ym)\}$ where x represents the input data and y is the corresponding target vector.
n (the number of samples)
k(indicates the number of clusters)
irt (imbalance ratio threshold)
knn (quantity of nearest neighbors)
begin
// Step 1: Divide the input space into clusters *clusters* ← *kmeans*(*X*) filtered *clusters* ← empty set for c in cluster:
imbalance ratio ← $\frac{\text{majority count(c)}+1}{\text{minority count}(c)+1}$
If the imbalance ratio is less than the *irt*, add the cluster "c" to the filtered cluster set. Repeat this process until all clusters have been checked.
end
end
Step 2: Implement the SMOTE oversampling technique.
1) From the minority class, choose an instance x_i at random
2) Determine the kind of x_i and denote the samples as S_j
3) Randomly create a synthetic data point p by then selecting a sample in *Sj* called *z*, then creating a line segment in the feature space by connecting p and *z*
4) Minority class label assigned to *p*.
5) Create a series of synthetic instances by combining *p* and *z* convexly.
Step 3: Employing ENN techniques
1) Choose the arbitrary instance x_r from the set S
2) Determine the knn of x_r, with k being equal to 5
3) Remove the x_r Element if it has a more significant number of neighbors from the other class.
4) Iterate 6 to 8 steps for the entire training dataset.
End

B. ENSEMBLE TECHNIQUE

The proposed method utilizes a combination of Random Forest and Decision Tree to create a robust ensemble model based on stacking. The stacking framework consists of two.

Hierarchical levels: the base and meta layers. The primary classifiers are trained and validated in the base layer using instances excluded from the initial training set. The predictions generated by these classifiers and their genuine

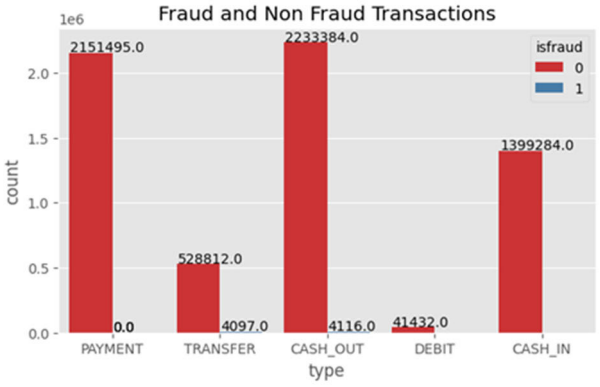

FIGURE 1. Number of legitimate transactions and fraudulent transactions for each type of transaction.

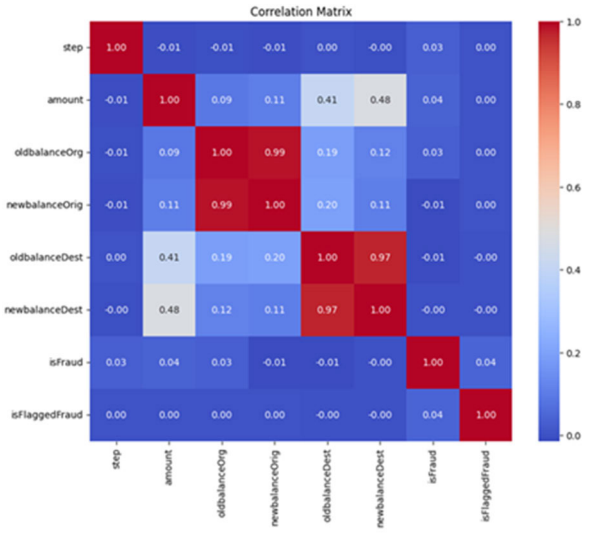

FIGURE 2. Correlation matrix for PaySim dataset.

labels represent the independent and dependent variables in an updated data set. This dataset subsequently trains the meta-classifier at the meta layer [5].

The main reasons for choosing Decision Tree and Random Forest as base learners are their resilience in managing various data patterns and exceptional predictive capabilities. Decision Trees are straightforward to understand and capture complex relationships, while Random Forests mitigate overfitting and enhance generalization by aggregating the outcomes of numerous trees. Additionally, the variations among the base models guarantee variance across the ensemble, which is crucial when different base models tend to produce distinct types of failures.

By selecting Random Forest as the meta-learner, we may exploit its capacity to effectively merge the advantages of the base learners, hence strengthening the overall model performance through variance reduction and stability improvement. Figure 11 shows the suggested methodology flowchart. The proposed approach has eight steps. First, data is cleaned

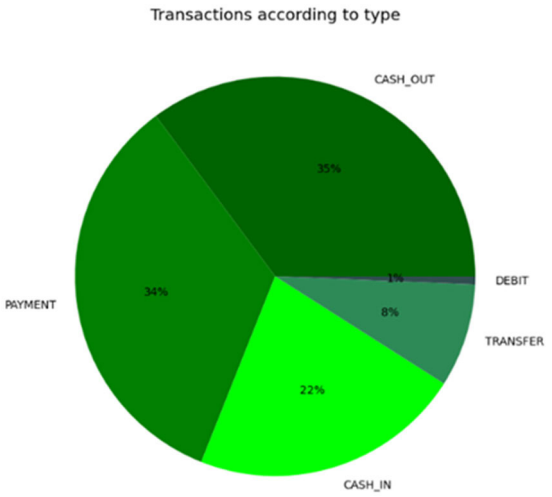

FIGURE 3. Distribution of transaction types in the PaySim dataset.

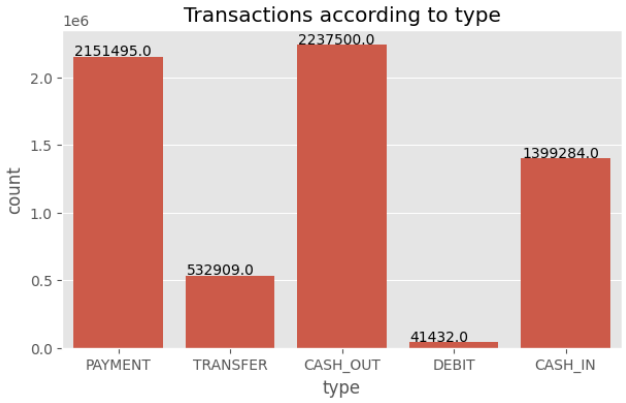

FIGURE 4. Number of legitimate transactions and fraudulent transactions for each type of transaction.

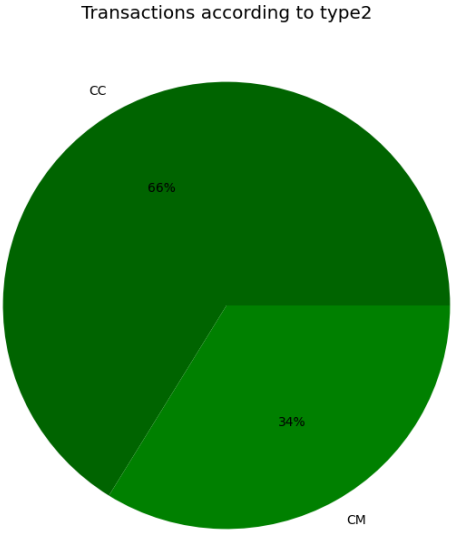

FIGURE 5. Distribution of transactions according to type 2 in the PaySim dataset.

to remove irregularities and noise, ensuring its quality and reliability. After that, feature engineering turns raw data into valuable features that help predictive models understand the situation.

This improves the model's understanding of data and learning. K-SMOTEENN addresses imbalanced datasets. K-Means clustering identifies and groups comparable instances, SMOTE generates minority class synthetic examples to balance the dataset, and ENN removes noisy and misclassified samples to improve balance and quality. After preprocessing, the data is separated into 70% training and 30% test sets. This guarantees that the model gets trained on a large percentage of the data while maintaining enough for an unbiased performance evaluation. The following stage uses Random Forest and Decision Tree classifiers as base learners. These algorithms are chosen for their durability and data versatility. The processed training data trains numerous models that capture distinct data properties. A Random Forest meta-learner classifier is then used. This meta-learner uses the strengths of each base learner to produce final predictions, improving performance and reducing overfitting. Precision, Recall, F1 Measure, and AUC/ROC assess the suggested method's performance. These measures reveal the model's accuracy, positive instance identification, precision-recall balance, and discrimination. For example, each sample is represented as xi, yi in a credit card dataset S. A generated sample is xj, yj, when xj is the out-of-sample model. $h1\,(xi)\,, h2\,(x2)\,, \ldots .hn\,(x1)\,.$

In the third phase, the dataset we generated will be utilized to train the RandomForest meta-classifier. This classifier integrates the base models. Thus, assuming a test instance x, the final ensemble prediction is obtained by applying the meta-model hn to the base model predictions. $hj(h1(x), h2(x), \ldots, hn(x))$.

In the last phase, LIME (Local Interpretable Model-Agnostic Explanations) techniques are applied to different layers of the stacking ensemble to provide a comprehensive explanation of the workflow. At the base learner level, LIME reveals how each model, such as Random Forest or Decision Tree, interprets the data and contributes to predictions. At the meta-learner level, it explains how the final ensemble prediction is derived by combining outputs from the base models. This multi-layered interpretability enhances model transparency and helps stakeholders trace decisions from input data to the final output, ensuring accountability and regulatory compliance.

V. RESULT AND DISCUSSION

This research introduces an innovative method for detecting credit card fraud. It involves creating a stacking classifier that integrates data resampling using K-SMOTEENN. The experimental results are divided into two sections: the performance of the classifiers before and after data resampling. The proposed stacking ensemble model was constructed using the

Random Forest and Decision Tree algorithms as base learners, with the Random Forest algorithm as the meta-learner. It was utilized alongside three other classifiers to evaluate and compare their performance. The classifiers comprise LGBM, Logistic Regression, and XGBoost. We constructed the models using sci-kit-learn, a widely used and respected Python package for machine learning.

The hardware utilized for model production is a Windows workstation equipped with 16 GB of RAM, showcasing an Intel(R) Core(TM) i7-10700 CPU @ 2.90GHz. The models were evaluated using the following performance metrics: Accuracy, Precision, Recall, F-1 score, Precision-Recall Curve, and Area under the ROC curve (AUC). The ROC curve is a graphical representation demonstrating the classifier's ability to distinguish between fraudulent and non-fraudulent categories. The plot is generated by graphing the rate of correctly identified positive instances against the rate of incorrectly identified negative instances at various threshold values. The AUC (Area Under the Curve) concisely represents the ROC (Receiver Operating Characteristic) curve. It is a numerical value ranging from 0 to 1, where a value of 0 signifies that all predictions made by the classifiers are incorrect, while a value of 1 means a flawless classifier.

A. PERFORMANCE OF CLASSIFIERS WITHOUT FEATURE CONSTRUCTION AND K-SMOTEENN

Table 1 illustrates the predictive accuracy of different machine learning classifiers and a stacking ensemble model. These models were trained using a balanced dataset and the k-means SMOTEENN technique. The classifiers evaluated include Decision Tree (DT), Random Forest (RF), LightGBM (LGBM), XGBoost (XGB), and Logistic Regression (LR). Additionally, a stacking ensemble that integrates these models was assessed.

TABLE 1. Performance without collaboration feature construction and K-SMOTEENN.

Classifier	Accuracy	F1-Score	Recall	Precision	AUPRCCC	ROC
RF	1.00	0.79	0.68	**0.95**	0.84	0.96
DT	1.00	0.81	0.80	0.83	0.66	0.90
XGB	1.00	0.41	**0.98**	0.26	**0.92**	**1.00**
LGBM	1.00	0.65	0.65	0.64	0.53	0.86
LR	1.00	0.59	0.45	0.89	0.58	0.93
STACKING MODEL (Proposed)	**1.00**	**0.84**	0.79	0.91	**0.91**	**0.99**

The proposed stacking model achieved higher accuracy, F1-score, precision, AUPRC, and ROC-AUC compared to the individual classifiers that make up the ensemble. Specifically, the stacking model reached an AUPRC value of 0.91 and a ROC-AUC of 0.99, indicating superior performance in distinguishing between the classes. This performance was closely followed by the XGBoost classifier, which achieved an AUPRC of 0.92 and a ROC-AUC of 1.00, and the Random Forest classifier, with an AUPRC of 0.84 and a ROC-AUC of 0.96. Additionally, the stacking model demonstrated a good balance between precision (0.91) and recall (0.79), resulting in an F1 score of 0.84. This highlights the ensemble's ability to identify positive (Fraud) and negative (non-fraud) cases more effectively than the individual classifiers. For instance, while the XGBoost model achieved the highest recall of 0.98, its precision was significantly lower at 0.47, resulting in a lower overall F1 score of 0.64. Similarly, the LightGBM model showed the lowest performance.

B. PERFORMANCE OF THE CLASSIFIER WITH FEATURE CONSTRUCTION AND K-SMOTEENN

Table 2 shows the prediction performance of various machine learning classifiers and a stacking ensemble model trained with a balanced dataset using K-SMOTEENN and feature construction techniques. The classifiers evaluated include Decision Tree (DT), Random Forest (RF), LightGBM (LGBM), XGBoost (XGB), and Stacking Ensemble models.

The proposed stacking model achieved higher accuracy, F1-score, precision, and Area under the Precision-Recall Curve (AUPRC) compared to the individual classifiers that make up the ensemble. Specifically, the stacking model reached the highest AUPRC value of 0.96, indicating superior performance in distinguishing between the classes. This performance was closely followed by the Decision Tree classifier, which achieved an AUPRC of 0.80, and the Random Forest classifier, with an AUPRC of 0.76.

TABLE 2. Performance with collaboration feature construction and K-K-SMOTEENN.

Classifier	Accuracy	F1-Score	Recall	Precision	AUPRCCC	ROC
RF	1.00	0.86	0.79	**0.96**	0.76	0.97
DT	1.00	0.90	0.91	0.89	0.80	0.94
XGB	1.00	0.64	**0.98**	0.47	0.94	**1.00**
LGBM	1.00	0.32	0.46	0.24	0.11	0.80
LR	1.00	0.51	0.41	0.92	0.38	0.99
Stacking(Proposed)	**1.00**	**0.92**	0.88	0.95	**0.96**	**1.00**

Additionally, the stacking model demonstrated an excellent balance between precision (0.95) and recall (0.88), resulting in an F1 score of 0.92. This highlights the ensemble's ability to identify positive (Fraud) and negative (non-fraud) cases more effectively than the individual classifiers. For instance, while the XGBoost model achieved the highest recall of 0.98, its precision was significantly lower at 0.47, resulting in a lower overall F1 score of 0.64. Figure 6 compares the Area under the Precision-Recall Curve (AUPRC).

Meanwhile, the ROC-AUC values further demonstrate the effectiveness of the stacking model, which achieved a perfect ROC-AUC of 1.00, indicating excellent performance in distinguishing between the positive and negative classes. The Random Forest and Decision Tree classifiers also performed

well, with ROC-AUC values of 0.97 and 0.94, respectively. These high ROC-AUC values show that the models have a solid ability to differentiate between the classes.

Figure 7 shows the different models' Receiver Operating Characteristic (ROC) curves. The stacking ensemble's superior performance in both AUPRC and ROC-AUC metrics suggests that combining the strengths of multiple classifiers leads to a more robust and accurate predictive model, effectively handling the class imbalance problem inherent in fraud detection datasets.

In conclusion, the proposed stacking model outperformed the individual classifiers regarding AUPRC, F1-score, and ROC-AUC, demonstrating its effectiveness in handling imbalanced datasets using K-SMOTEENN and feature construction techniques. The performance metrics highlight the model's effectiveness in fraud detection. F1-score balances precision and recall to minimize false positives and negatives. Precision ensures flagged transactions are truly fraudulent, while recall captures most fraud cases to reduce missed incidents. AUPRC reflects the balance between precision and recall across thresholds, which is crucial for imbalanced data. ROC-AUC indicates the model's ability to distinguish fraud from non-fraud, with a perfect score showing excellent overall performance.

C. EXPLAINABLE AI

After training the models, the test data is used to generate predictions. These predictions are then analyzed using an interpretability method such as Local Interpretable Model-Agnostic Explanations (LIME) [33]. Local Interpretable Model-Agnostic Explanations (LIME) is a technique within Explainable AI (XAI) that provides localized explanations for individual predictions made by machine learning models [34]. On the other hand, Explainable AI (XAI) provides a model that enhances interpretability, helping users better understand the decision-making process and facilitating further evaluation of result accuracy [35].

The main idea behind LIME is to identify the input features that significantly impact predicting a specific target class within a trained model. Its ease of use and quick processing make it a valuable tool for generating clear and actionable explanations [36].

In this study, LIME is used to analyze the predictions from the stacking ensemble learning model and provide interpretations of the results. Figure 10 explains the model's predictions for the features.

The stacking ensemble model identified the most influential features as Feature 0, Feature 1, Feature 5, Feature 6, Feature 8, Feature 3, Feature 4, Feature 7, Feature 9, and Feature 2. The prediction is primarily driven by the combined contributions of these features across the base learners and the meta-learner. Blue bars represent features that indicate a prediction of *No Fraud* when the feature value satisfies the specified condition. In contrast, orange bars represent features that suggest *Fraud* when their respective criteria are met.

For example:

- Feature 0 contributes to *No Fraud* when its value is less than or equal to -0.13. In this case, the feature value of Feature 0 is -0.18, which satisfies the condition for *No Fraud*.
- Feature 5, however, contributes to *Fraud* when its value exceeds or exceeds 1.36. For this prediction, the value of Feature 5 is precisely 1.36, indicating a contribution to *Fraud*.

Out of the ten features analyzed, three met the criteria for *No Fraud*, while the remaining seven aligned with the requirements for *Fraud*. Consequently, the stacking ensemble model predicts the transaction with a probability of 1.00 for No Fraud due to the more substantial cumulative contribution of features indicating *No Fraud*.

This interpretation highlights the explainability of the stacking ensemble model by demonstrating how both the base learners and the meta-learner utilized the top features to make an accurate and interpretable prediction.

D. COMPARATIVE ANALYSIS OF PERFORMANCE WITH RECENT STUDIES USING THE PAYSIM DATASET

Table 3 shows that the proposed stacking ensemble strategy outperforms other models by demonstrating outstanding performance across all evaluated metrics. It attained the highest accuracy at 100%, with Hajek et al. [40] being the closest competitor, whose random forest model using BCB_SMOTE achieved 99.95%. Our method also ranked first in the F1-Score, with a value of 0.92, followed by Hajek et al. [40] at 85.20.

The proposed method's recall score of 0.88 is comparable to the highest recall values recorded, including those of Paulraj et al. [38], with XGBoost-based frameworks that employ outlier detection (0.99) and RUS (0.978). Our method's precision was an impressive 0.95, surpassing that of Hajek et al. [40], which achieved a precision 81.27.

Hajek et al. [40] provided the closest comparable metric at 72.77, while the proposed method's AUPRCC was notably high at 0.96, indicating robust precision-recall performance. Furthermore, the ROC value for our method was a perfect 1.00, which suggests that it performed flawlessly in classification. In comparison, other high-performing models, such as Mubalaike and Adali [37] and Paulraj et al. [38], have ROC values of 0.98 and 0.995, respectively.

In conclusion, the proposed stacking ensemble method demonstrates its robustness and potential for real-world applications, such as real-time fraud detection, by demonstrating superior performance in the following areas: accuracy, F1-Score, recall, precision, AUPRCC, and ROC.

E. ENSEMBLE PERFORMANCE COMPARISON WITH RECENT SCHOLARLY WORKS

This study introduces a novel stacking ensemble approach using K-SMOTEENN for imbalanced fraud detection, evaluated on the Paysim dataset shown in Table 4. Compared

to previous research on credit card fraud detection, our method demonstrated superior performance across key metrics, achieving an accuracy of 100%, an F1-score of 0.92, recall of 0.88, precision of 0.95, an AUPRC of 0.96, and an ROC-AUC of 1.0. In contrast, prior works such as Bhakta et al. (LR-DT ensemble), Bagga et al. (Bagging), and Esenogho et al. (LSTM) reported lower performance metrics and often lacked sufficient validation metrics, mainly when dealing with imbalanced data. While others employed methods like ADASYN and SMOTE-ENN, their results, in terms of metrics like F1-score, recall, and AUPRC, were generally less robust than ours. Additionally, limitations in handling data imbalance and insufficient validation metrics were common issues in these previous studies. In contrast, our approach addressed these concerns and outperformed the alternatives in almost all respects.

F. LEARNING CURVE ANALYSIS

Figure 8 illustrates the stacking ensemble model's learning curve, with RandomForest as the meta-learner trained on the dataset. The red line represents the model's training accuracy, which remains consistently high at 1.0 across all sample sizes. This indicates that the model perfectly fits the training data, potentially signaling an overfit, as it captures the nuances of the training set without necessarily generalizing well to unseen data. In contrast, the green line shows the model's cross-validation performance, which begins at 0.85 and gradually increases to 0.92 as the number of training examples grows. The increasing cross-validation score reflects the positive impact of adding more training data, suggesting that the model benefits from a larger dataset to improve generalization. However, the diminishing slope as the curve approaches 0.92 indicates that the model's performance begins to stabilize, and further increases in data size may have a limited effect on boosting accuracy. The shaded region around the green line represents the variance in cross-validation scores, which is relatively narrow, suggesting that the model is stable across different validation folds. The gap between the training and cross-validation scores highlights a degree of overfitting, likely exacerbated by the model's complexity and the nature of the dataset. Given that the dataset is imbalanced, applying the K-SMOTEENN technique has likely mitigated some issues associated with minority class underrepresentation, as evidenced by the consistently high cross-validation performance.

G. COMPUTATIONAL COST

As depicted in Figure 9, the analysis of the computational costs associated with stacking ensemble classifiers reveals a significant increase in training time when utilizing the entire dataset, underscoring the reviewer's concern about computational expense. The graph structure, with the x-axis representing the fraction of training examples and dual y-axes for training time and memory usage, illustrates that training time remains relatively stable for smaller data subsets but dramatically increases to nearly 10 hours for the entire dataset. While peaking at around 100 MB, memory usage is more manageable than the training time. This finding suggests that while stacking ensembles offer robust predictive capabilities, their application to large datasets may require careful consideration of computational resources. To mitigate these costs, researchers could explore data sampling or dimensionality reduction techniques, which may allow for efficient model training without compromising performance. In future research, we could focus on optimizing these techniques or developing novel algorithms that maintain the benefits of ensemble methods while reducing computational demands. Additionally, investigating the trade-offs between model complexity and computational efficiency could provide valuable insights for practitioners balancing accuracy with resource constraints.

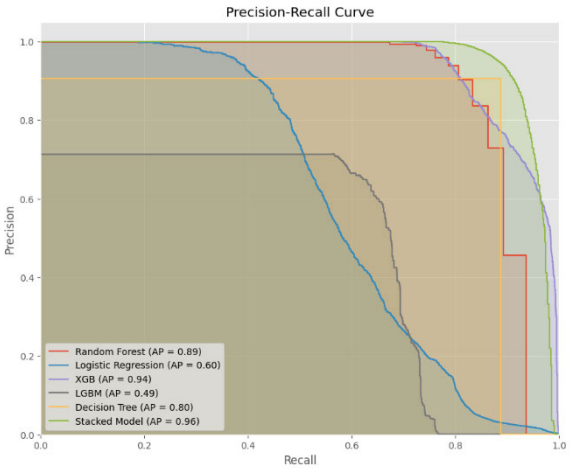

FIGURE 6. AUPRC.

H. DISCUSSION

The stacking ensemble model exhibited improved performance compared to individual classifiers when utilized for the challenge of credit card fraud detection. This was achieved by balancing the dataset using K-SMOTEENN and enhancing it through feature construction. The ensemble model, which amalgamates classifiers such as Random Forest and Decision Tree and then compared with an individual model, XGBoost, LightGBM, and Logistic Regression, surpassed each classifier in terms of crucial metrics, including F1-score, precision, recall, AUPRC, and ROC-AUC. The stacking model demonstrated an exceptional ROC-AUC of 1.00, highlighting its resilience and differentiation ability. The Random Forest classifier achieved a precision of 0.96 and an AUPRC of 0.76. The Decision Tree classifier achieved a recall of 0.91 and an AUPRC of 0.80.

On the other hand, the XGBoost classifier demonstrated the best recall rate of 0.98, while its precision was considerably lower at 0.47. As a result, the F1-score was lower at 0.64. The results highlight the benefits of the stacking ensemble,

TABLE 3. Performance comparison with recent scholarly works using the Paysim dataset.

Reference	Model	Sampling Strategy	Accuracy	F1-Score	Recall	Precision	AUPRCC	ROC
Aji et al. [37]	Stacked autoencoder	-	0.852	-	-	-	-	0.81
Paulraj et al [38]	1D-CNN	-	0.90	0.90	0.90	0.90	0.90	0.90
Schlor et al. [39]	XGB-based INALU	-	-	0.84	-	-	-	0.98
Hajek et al. [40]	XGBoost-based framework	Outlier Detection	0.999	0.873	0.99	0.779	-	0.995
Hajek et al. [40]	XGBoost-based framework	RUS	0.9760	0.489	0.978	0.326	-	0.977
Abdulwahab et al. [41]	ResNeXt-embedded Gated Recurrent Unit (GRU)	SMOTE	0.896	0.904	0.912	0.896	-	0.971
Alamri et al. [17]	Random Forest	BCB_SMOTE	99.95	85.20	89.53	81.27	72.77	-
Our proposed	**Stacking Ensemble**	**K-SMOTEENN**	**100**	**0.92**	**0.88**	**0.95**	**0.96**	**1.00**

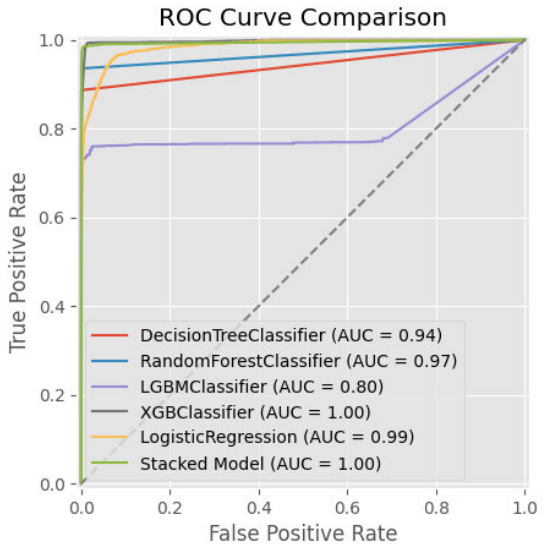

FIGURE 7. ROC.

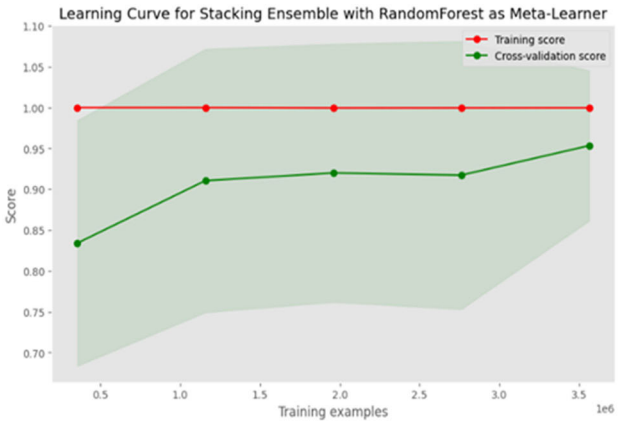

FIGURE 8. Learning curve analysis.

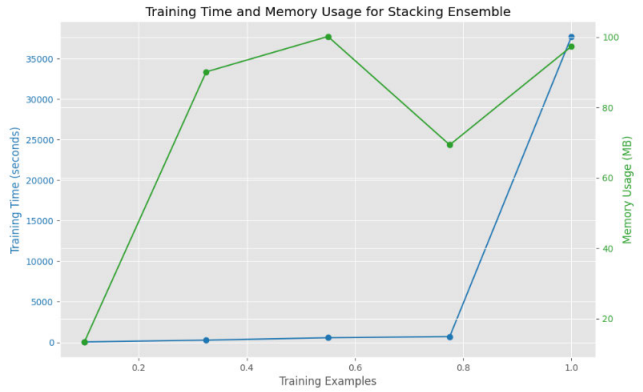

FIGURE 9. Computational cost.

which utilizes the strengths of numerous models to generate a more balanced and superior overall performance.

K-means SMOTEENN effectively addresses the dual challenges of class imbalance and noise in fraud detection datasets, significantly enhancing model performance and practical applicability. By increasing the representation of the minority class (fraudulent transactions) through K-means-guided synthetic sample generation, the technique ensures a balanced dataset while preserving feature diversity. The ENN (Edited Nearest Neighbors) component further refines the training data by eliminating noisy or misclassified instances, resulting in cleaner data that enhances model reliability. Comparative analysis shows that K-means SMOTEENN substantially improves predictive performance, achieving higher F1-scores, precision, and recall than standard resampling methods or no resampling while reducing false positives and improving the detection of fraudulent transactions. These capabilities demonstrate that K-means SMOTEENN not only overcomes the inherent challenges of imbalanced and noisy data but also addresses a significant challenge in fraud detection models with remarkable effectiveness, making it a robust solution for real-world applications.

Even though k-means clustering tries to improve the diversity of synthetic samples by generating them within local clusters, the fundamental SMOTE approach still interpolates between minority class points. Suppose the synthetic samples are too similar or overly focused on narrow feature ranges. In that case, the model might learn artificial patterns that are not representative of the minority class in real-world data.

TABLE 4. Ensemble performance comparison with recent scholarly works.

Reference	Model	Transactions Data	Sampling Strategy	Accuracy	F1-Score	Recall	Precision	AUPRC	ROC	Limitation
Bhakta et al [42]	Stacking Ensemble (LR+DT)	Kaggle Dataset	-	0.99	69.68	55.10	0.94	-	-	It did not handle imbalanced data; more validation metrics and performance needed to be better.
Bagga et al. [43]	Ensemble_Bagging Classifier	European Credit Card 2013	ADASYN	99.9	0.81	0.87	0.76	-	-	Performance is insufficient using another evaluation metric, such as MCC or BCR.
Esenogho et al. [44]	LSTM Ensemble	European Credit Card 2013	SMOTE-ENN		-	-	-	-	0.99	More validation metrics.
Rakhshaninejad et al[45]	Proposed ensemble-based method (Eclf_Best)	European Credit Card 2013	Undersampling Method	0.99	0.92	0.97	0.87	-	-	More validation metrics.
Forough et al. [46]	Ensemble the deep sequential model based on GRU-LSTM	European Credit Card 2013	-	-	0.78	0.66	0.95	0.63	0.83	The imbalanced data needed to be addressed, resulting in suboptimal performance.
Xie et al [47]	HELMDD	Kaggle Dataset	SMOTE, RUS	-	-	0.99	-	-	0.98	More validation metrics.
Our proposed	**Stacking Ensemble**	**Paysim Dataset**	**K-SMOTEENN**	**100**	**0.92**	**0.88**	**0.95**	**0.96**	**1.00**	**Mostly outperform other works.**

This can lead to overfitting, where the model performs well on the training data but fails to generalize to unseen data. The ENN component removes noisy or misclassified instances, making the training data ''cleaner'' than real-world data. If the model becomes too accustomed to this cleaned version of the data, it may be unable to handle noisy or ambiguous samples in the real-world test set, increasing the risk of overfitting. This will be one of the challenges in the future in terms of combining resampling techniques, and will also be a point of concern in using this technique to be overcome and developed in further research.

In this research, overfitting was overcome by stacking ensemble learning. Decision trees and random forests are the base learners, and random forests are metals with the maximum depth of the tree or minimum samples per leaf that can help prevent overfitting by forcing the model to generalize rather than over-learn from synthetic or noisy instances.

Overall, the findings demonstrate that the stacking ensemble model successfully detected fraudulent transactions while also achieving a high level of accuracy, hence minimizing the occurrence of false positives because one of the main targets of this research is to reduce false positives.

While the stacking ensemble model can efficiently detect suspicious transactions, the computational cost analysis reveals that training the model requires up to 10 hours on the full dataset and utilizes up to 100 MB of memory. This highlights challenges in deploying the model on large-scale data or dynamic environments requiring frequent updates. Incremental learning or model distillation could enhance scalability and reduce computational demands, enabling model updates without full retraining. Additionally, dimensionality reduction or feature selection may help decrease training complexity without compromising model

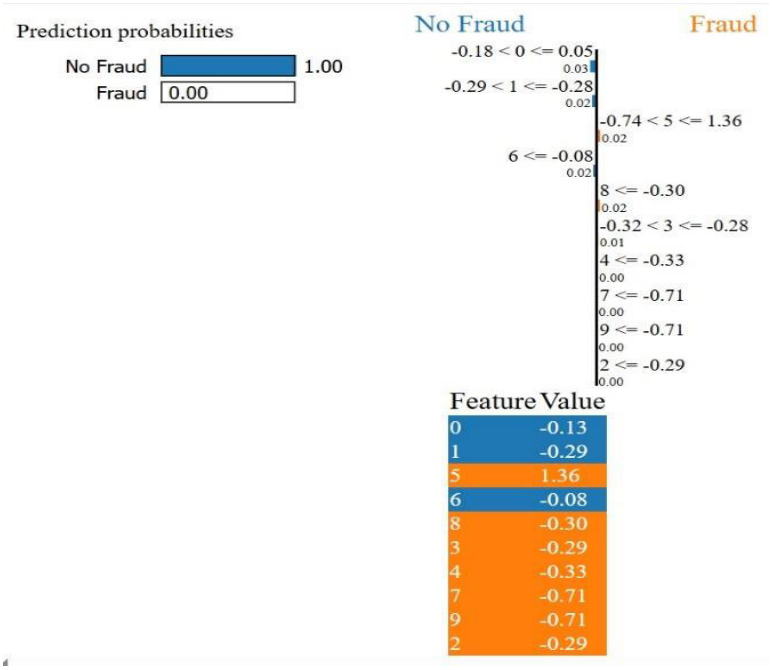

FIGURE 10. Explainable AI.

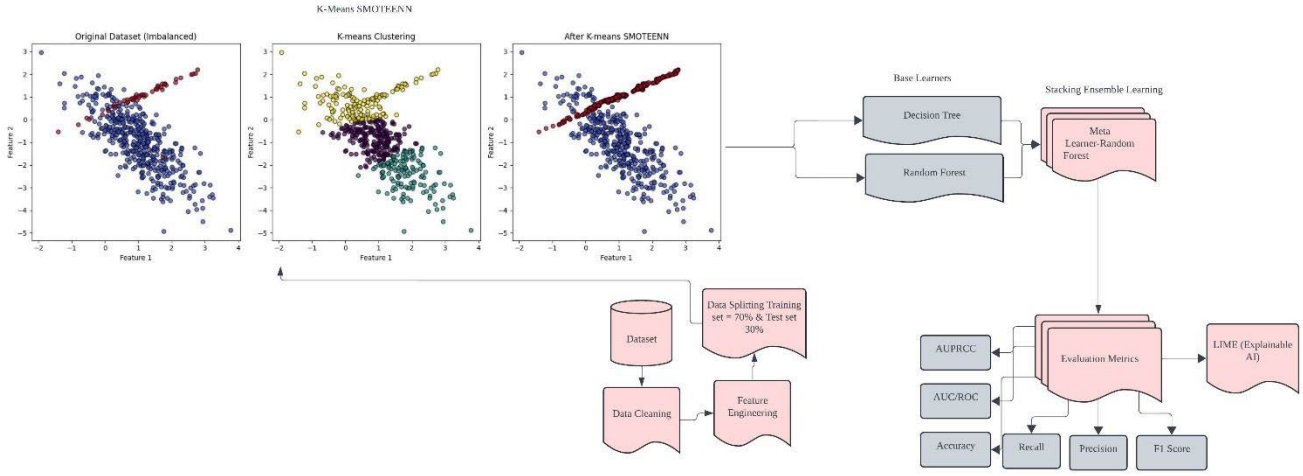

FIGURE 11. Proposed method workflow.

performance. Future research could focus on optimizing these strategies or developing novel algorithms that preserve the predictive strengths of ensemble methods while improving computational efficiency, ensuring the model's reliability for large-scale real-time applications.

VI. CONCLUSION

The surge in online transactions and the widespread use of credit cards have contributed to a notable rise in credit card fraud, posing challenges for financial institutions and their customers. Efficient identification of fraudulent activities is essential to reduce these dangers. This study enhances the current knowledge by introducing a new method that utilizes a stacking ensemble of machine learning models and data resampling techniques like K-Means SMOTEENN and feature construction. The stacking ensemble model, which combines classifiers like Random Forest and Decision Tree, then compares with individual models like XGBoost, LightGBM, and Logistic Regression, as well as Random Forest and Decision Tree, and shows more excellent performance. The stacking model obtained an F1-score of 0.92, a precision of 0.95, and a recall of 0.88.

Furthermore, it achieved performance compared to other approaches mentioned in the literature, such as 0.96 for the Area Under the Precision-Recall Curve (AUPRC) and a flawless value of 1.00 for the ROC-AUC. The results demonstrate the model's extraordinary proficiency in differentiating fraudulent and non-fraudulent transactions. When comparing several classifiers, it was found that XGBoost and Decision Tree had high recall values of 0.98 and 0.91, respectively. However, their precision and AUPRC values were considerably lower. This emphasizes the effectiveness of the ensemble technique. The Random Forest classifier demonstrated strong performance, achieving an AUPRC (Area Under the Precision-Recall Curve) of 0.76 and a ROC-AUC (Receiver Operating Characteristic - Area Under the Curve) of 0.97. The results indicate that combining many classifiers in a stacking ensemble and improved data resampling techniques can significantly improve the identification of fraudulent transactions. This leads to a more muscular and dependable approach to detecting credit card fraud. Subsequent research could investigate incorporating more variety in the foundational models by integrating classifiers with distinct training approaches and novel resampling techniques. In addition, this study successfully demonstrates the effectiveness of a stacking ensemble model in addressing key challenges in credit card fraud detection, including class imbalance and overfitting, while achieving high accuracy and interpretability. Future research will focus on advancing real-time applications by refining and enhancing our previous methodology with more innovative and adaptive approaches. Furthermore, understanding the trade-offs between model complexity, energy efficiency, and accuracy will be critical for advancing the practicality of ensemble methods in dynamic real-world environments.

COMPETING INTERESTS

The authors declare that they have no conflict of interest.

ACKNOWLEDGMENT

(Nurafni Damanik and Chuan-Ming Liu contributed equally to this work.)

The authors would like to thank the university, Taipei Tech, for all kinds of support on this study.

REFERENCES

[1] I. D. Mienye and N. Jere, "Deep learning for credit card fraud detection: A review of algorithms, challenges, and solutions," *IEEE Access*, vol. 12, pp. 96893–96910, 2024.

[2] E. Ileberi, Y. Sun, and Z. Wang, "Performance evaluation of machine learning methods for credit card fraud detection using SMOTE and AdaBoost," *IEEE Access*, vol. 9, pp. 165286–165294, 2021.

[3] N. S. Alfaiz and S. M. Fati, "Enhanced credit card fraud detection model using machine learning," *Electronics*, vol. 11, no. 4, p. 662, Feb. 2022.

[4] P. Gupta, A. Varshney, M. R. Khan, R. Ahmed, M. Shuaib, and S. Alam, "Unbalanced credit card fraud detection data: A machine learning-oriented comparative study of balancing techniques," *Proc. Comput. Sci.*, vol. 218, pp. 2575–2584, Jan. 2023.

[5] I. D. Mienye and Y. Sun, "A deep learning ensemble with data resampling for credit card fraud detection," *IEEE Access*, vol. 11, pp. 30628–30638, 2023.

[6] A. Singh, R. K. Ranjan, and A. Tiwari, "Credit card fraud detection under extreme imbalanced data: A comparative study of data-level algorithms," *J. Experim. Theor. Artif. Intell.*, vol. 34, no. 4, pp. 571–598, Jul. 2022.

[7] A. A. Khan, O. Chaudhari, and R. Chandra, "A review of ensemble learning and data augmentation models for class imbalanced problems: Combination, implementation, and evaluation," *Expert Syst. Appl.*, vol. 244, Jun. 2023, Art. no. 122778.

[8] L. Ni, J. Li, H. Xu, X. Wang, and J. Zhang, "Fraud feature boosting mechanism and spiral oversampling balancing technique for credit card fraud detection," *IEEE Trans. Comput. Soc. Syst.*, vol. 11, no. 2, pp. 1615–1630, Apr. 2023.

[9] E. A. L. Marazqah Btoush, X. Zhou, R. Gururajan, K. C. Chan, R. Genrich, and P. Sankaran, "A systematic review of literature on credit card cyber fraud detection using machine and deep learning," *PeerJ Comput. Sci.*, vol. 9, Apr. 2023, Art. no. e1278.

[10] M. A. Mim, N. Majadi, and P. Mazumder, "A soft voting ensemble learning approach for credit card fraud detection," *Heliyon*, vol. 10, no. 3, Feb. 2024, Art. no. e25466.

[11] M. H. Aung, P. T. Seluka, J. T. R. Fuata, M. J. Tikoisuva, M. S. Cabealawa, and R. Nand, "Random forest classifier for detecting credit card fraud based on performance metrics," in *Proc. IEEE Asia–Pacific Conf. Comput. Sci. Data Eng. (CSDE)*, Dec. 2020, pp. 1–6.

[12] A. M. Aburbeian and H. I. Ashqar, "Credit card fraud detection using enhanced random forest classifier for imbalanced data," in *Proc. Int. Conf. Adv. Comput. Res.*, 2023, pp. 605–616.

[13] H. Zhu, M. Zhou, G. Liu, Y. Xie, S. Liu, and C. Guo, "NUS: Noisy-sample-removed undersampling scheme for imbalanced classification and application to credit card fraud detection," *IEEE Trans. Computat. Social Syst.*, vol. 11, no. 2, pp. 1793–1804, Apr. 2023.

[14] X. Wang, J. Gong, Y. Song, and J. Hu, "Adaptively weighted three-way decision oversampling: A cluster imbalanced-ratio based approach," *Appl. Intell.*, vol. 53, no. 1, pp. 312–335, 2023.

[15] A. Abd El-Naby, E. E.-D. Hemdan, and A. El-Sayed, "An efficient fraud detection framework with credit card imbalanced data in financial services," *Multimedia Tools Appl.*, vol. 82, no. 3, pp. 4139–4160, Jan. 2023.

[16] S. K. Hashemi, S. L. Mirtaheri, and S. Greco, "Fraud detection in banking data by machine learning techniques," *IEEE Access*, vol. 11, pp. 3034–3043, 2022.

[17] M. Alamri and M. Ykhlef, "Hybrid undersampling and oversampling for handling imbalanced credit card data," *IEEE Access*, vol. 12, pp. 14050–14060, 2024.

[18] E. A. Lopez-Rojas, A. Elmir, and S. Axelsson, "Paysim: A financial mobile money simulator for fraud detection," in *Proc. 28th Eur. Model. Simulation Symp. (EMSS)*, Larnaca, Cyprus, Sep. 2016, pp. 249–255.

[19] S. García, J. Luengo, and F. Herrera, *Data Preprocessing in Data Mining*, vol. 72. Cham, Switzerland: Springer, 2015.

[20] C. Chatfield, "Exploratory data analysis," *Eur. J. Oper. Res.*, vol. 23, no. 1, pp. 5–13, 1986.

[21] S. Piramuthu, N. Raman, and M. J. Shaw, "Decision support system for scheduling a flexible flow system: Incorporation of feature construction," *Ann. Oper. Res.*, vol. 78, pp. 219–234, Jan. 1998.

[22] H. Liu and H. Motoda, *Feature Extraction, Construction and Selection: A Data Mining Perspective*, vol. 453. Cham, Switzerland: Springer, 1998.

[23] E. Ileberi, Y. Sun, and Z. Wang, "A machine learning based credit card fraud detection using the GA algorithm for feature selection," *J. Big Data*, vol. 9, no. 1, p. 24, Dec. 2022.

[24] O. Sagi and L. Rokach, "Ensemble learning: A survey," *WIREs Data Mining Knowl. Discovery*, vol. 8, no. 4, Jul. 2018, Art. no. e1249.

[25] M. Graczyk, T. Lasota, B. Trawiński, and K. Trawiński, "Comparison of bagging, boosting and stacking ensembles applied to real estate appraisal," in *Proc. Asian Conf. Intell. Inf. Database Syst. (ACIIDS)*, Hue City, Vietnam, Mar. 2010, pp. 340–350.

[26] Y. Sun, Z. Li, X. Li, and J. Zhang, "Classifier selection and ensemble model for multi-class imbalance learning in education grants prediction," *Appl. Artif. Intell.*, vol. 35, no. 4, pp. 290–303, Mar. 2021.

[27] D. Opitz and R. Maclin, "Popular ensemble methods: An empirical study," *J. Artif. Intell. Res.*, vol. 11, pp. 169–198, Aug. 1999.

[28] G. Wang, J. Hao, J. Ma, and H. Jiang, "A comparative assessment of ensemble learning for credit scoring," *Expert Syst. Appl.*, vol. 38, no. 1, pp. 223–230, Jan. 2011.

[29] D. H. Wolpert, "Stacked generalization," *Neural Netw.*, vol. 5, no. 2, pp. 241–259, Jan. 1992.

[30] N. Nayyer, N. Javaid, M. Akbar, A. Aldegheishem, N. Alrajeh, and M. Jamil, "A new framework for fraud detection in Bitcoin transactions through ensemble stacking model in smart cities," *IEEE Access*, vol. 11, pp. 90916–90938, 2023.
[31] Z. Huang, "Extensions to the k-means algorithm for clustering large data sets with categorical values," *Data Mining Knowl. Discovery*, vol. 2, no. 3, pp. 283–304, 1998.
[32] Z. Li, M. Huang, G. Liu, and C. Jiang, "A hybrid method with dynamic weighted entropy for handling the problem of class imbalance with overlap in credit card fraud detection," *Expert Syst. Appl.*, vol. 175, Aug. 2021, Art. no. 114750.
[33] B. Vihurskyi, "Credit card fraud detection with XAI: Improving interpretability and trust," in *Proc. 3rd Int. Conf. Distrib. Comput. Elect. Circuits Electron. (ICDCECE)*, Apr. 2024, pp. 1–6.
[34] G. Marvin, D. Jjingo, J. Nakatumba-Nabende, and M. G. R. Alam, "Local interpretable model-agnostic explanations for online maternal healthcare," in *Proc. 2nd Int. Conf. Smart Technol. Syst. Next Gener. Comput. (ICSTSN)*, Apr. 2023, pp. 1–6.
[35] M. Waqas, S. Abbas, U. Farooq, M. A. Khan, M. Ahmad, and N. Mahmood, "Autonomous vehicles congestion model: A transparent LSTM-based prediction model corporate with explainable artificial intelligence (EAI)," *Egyptian Informat. J.*, vol. 28, Dec. 2024, Art. no. 100582.
[36] B. Raufi, C. Finnegan, and L. Longo, "A comparative analysis of SHAP, LIME, ANCHORS, and DICE for interpreting a dense neural network in credit card fraud detection," in *Proc. World Conf. Explainable Artif. Intell.*, 2024, pp. 365–383.
[37] A. M. Mubalaike and E. Adali, "Deep learning approach for intelligent financial fraud detection system," in *Proc. 3rd Int. Conf. Comput. Sci. Eng. (UBMK)*, Sep. 2018, pp. 598–603.
[38] B. Paulraj, "Machine learning approaches for credit card fraud detection: A comparative analysis and the promise of 1D convolutional neural networks," in *Proc. 7th Int. Conf. Inf. Comput. Technol. (ICICT)*, Mar. 2024, pp. 82–92.
[39] D. Schlör, M. Ring, A. Krause, and A. Hotho, "Financial fraud detection with improved neural arithmetic logic units," in *Proc. Workshop Mining Data Financial Appl.*, vol. 2021, Ghent, Belgium, Sep. 2020, pp. 40–54.
[40] P. Hajek, M. Z. Abedin, and U. Sivarajah, "Fraud detection in mobile payment systems using an XGBoost-based framework," *Inf. Syst. Frontiers*, vol. 25, no. 5, pp. 1985–2003, Oct. 2023.
[41] A. A. Almazroi and N. Ayub, "Online payment fraud detection model using machine learning techniques," *IEEE Access*, vol. 11, pp. 137188–137203, 2023.
[42] S. S. Bhakta, S. Ghosh, and B. Sadhukhan, "Credit card fraud detection using machine learning: A comparative study of ensemble learning algorithms," in *Proc. 9th Int. Conf. Smart Comput. Commun. (ICSCC)*, Aug. 2023, pp. 296–301.
[43] S. Bagga, A. Goyal, N. Gupta, and A. Goyal, "Credit card fraud detection using pipelining and ensemble learning," *Proc. Comput. Sci.*, vol. 173, pp. 104–112, Jan. 2020.
[44] E. Esenogho, I. D. Mienye, T. G. Swart, K. Aruleba, and G. Obaido, "A neural network ensemble with feature engineering for improved credit card fraud detection," *IEEE Access*, vol. 10, pp. 16400–16407, 2022.
[45] M. Rakhshaninejad, M. Fathian, B. Amiri, and N. Yazdanjue, "An ensemble-based credit card fraud detection algorithm using an efficient voting strategy," *Comput. J.*, vol. 65, no. 8, pp. 1998–2015, Aug. 2022.
[46] J. Forough and S. Momtazi, "Ensemble of deep sequential models for credit card fraud detection," *Appl. Soft Comput.*, vol. 99, Feb. 2021, Art. no. 106883.
[47] Y. Xie, A. Li, L. Gao, and Z. Liu, "A heterogeneous ensemble learning model based on data distribution for credit card fraud detection," *Wireless Commun. Mobile Comput.*, vol. 2021, no. 1, Jan. 2021, Art. no. 2531210.

NURAFNI DAMANIK received the Master of Computer Science degree from Potensi Utama University. She is currently pursuing the Ph.D. degree with the National Taipei University of Technology, Taipei, Taiwan. She is exploring the interdisciplinary applications of artificial intelligence and data science in finance, particularly the section on fraud detection with the National Taipei University of Technology. She has presented her works at IET conferences in 2023, thereby contributing to advancing knowledge and innovation in information technology. Her research interests include machine learning, data science, deep learning, and credit card fraud.

CHUAN-MING LIU (Member, IEEE) received the Ph.D. degree in computer science from Purdue University, in 2002. He was the Head of the Extension Education Center, Department of Computer Science and Information Engineering (CSIE), National Taipei University of Technology (Taipei Tech), Taiwan, from 2018 to 2021. He joined the Department of CSIE, Taipei Tech, in the Spring of 2003. In 2010 and 2011, he visited Auburn University, Auburn, AL, USA, and Beijing Institute of Technology, Beijing, China. He is currently a Professor with the Department of CSIE, Taipei Tech, where he was the Department Chair, from 2013 to 2017. He has services in many journals, conferences, and societies and has published more than 100 papers in many prestigious journals and international conferences. His research interests include extensive data management and processing, uncertain data management, data science, spatial data processing, data streams, ad hoc and sensor networks, and location-based services. He was a co-recipient of many best paper awards, including ICUFN 2015, ICS 2016, MC 2017, WOCC 2018, MC 2019, MC 2021, and WOCC 2021.