Enhancing Credit Card Fraud Detection Using a Stacking Model Approach and Hyperparameter Optimization

El Bazi Abdelghafour[1], Chrayah Mohamed[2], Aknin Noura[3], Bouzidi Abdelhamid[4]
TIMS LABORATORY, FS Tetouan, Abdelmalek Essaadi University, Tetouan, Morocco[1,3,4]
TIMS LABORATORY, ENSA Tetouan, Abdelmalek Essaadi University, Tetouan, Morocco[2]

***Abstract*—Credit card fraud detection has emerged as a crucial area of study, especially with the rise in online transactions coupled with increased financial losses from fraudulent activities. In this regard, a refined framework for identifying credit card fraud is introduced, utilizing a stacking ensemble model along with hyperparameter optimization. This paper integrates three highly effective algorithms—XGBoost, CatBoost, and LightGBM—into a single strategy to improve predictive performance and address the issue of unbalanced datasets. To enable a more efficient search and adjustment of model parameters, Bayesian Optimization is employed for hyperparameter tuning. The proposed approach has been tested on a publicly accessible dataset. Results indicate notable enhancements over established baseline models in essential performance metrics, including ROC-AUC, precision, and recall. This method, while effective in fraud detection, holds significant promise for other fields focused on identifying rare occurrences.**

Keywords*—*Credit card fraud detection; stacking models; hyperparameter tuning; logistic regression; ensemble learning

I. Introduction

Identifying fraudulent credit card transactions has emerged as a significant challenge for the financial sector because of the swift growth of digital transactions and online shopping. Current models find it difficult to manage the inherent imbalance in datasets for fraud detection, where fraudulent transactions are infrequent yet expensive. In this regard, every day, finance institutions need to protect billions of transactions happening online. Its fraud prevention horizon, therefore, is at an unimaginable scale. This study seeks to enhance the accuracy of credit card fraud detection by employing a stacking ensemble model along with sophisticated hyperparameter optimization methods. The investigation focuses on these key inquiries: In what ways can stacking models be refined to effectively manage imbalanced datasets? What effect does hyperparameter tuning have on the performance of fraud detection?. Indeed, all estimates point to global financial losses to fraud being in excess of billions of dollars annually, which places a high burden on businesses to create much more improved systems for fraud detection [1]. Security in financial transactions has a critical role in maintaining consumer confidence and the integrity of banking as a whole [2], [3].

The traditional methods involve Logistic Regression, Decision Trees, and Random Forest, among other traditional machine learning techniques that have so far been the bedrock of most fraud detection systems [2]. The way these models work is that they learn from historical data about the patterns which can predict if a transaction is fraudulent or real. Although they have been successful in most cases, the imbalanced nature of fraud datasets, where the ratio of positive instances corresponding to fraudulent activities is much smaller compared to the negative instances corresponding to normal activities, poses a challenge that most traditional models cannot overcome. The class imbalance problem often results in biased models towards the majority class of legitimate transactions, yielding poor detection rates associated with fraudulent activities [4].

Some techniques, therefore, such as SMOTE and undersampling, are often applied to artificially balance a dataset in order to counteract this bias. Although such methods tend to slightly improve the performance of a model, their performance is not always very satisfactory, especially when the relationship in a high-dimensional dataset is complex and non-linear. These challenges have resulted in unprecedented interest in various ensemble learning methods that can achieve better predictive performance by combining multiple classifiers. Ensemble models, more specifically the stacking model, have shown capable performance even better than those of traditional methods at times by exploiting strengths of different algorithms. The mechanism for the operation of stacking models is based on the integration of several base learners, such as Logistic Regression or XGBoost, CatBoost, and LightGBM, with a meta-learner to combine these predictions. In doing so, there can be the realization of a more flexible model that captures different patterns within the data, while improving accuracy and generalization [5], [6]. The efficiency of the stacking model in fraud detection applications follows from the fact that usually, data is high-dimensional, quite complex, with subtle fraud behavior to be captured [7].

Another very important factor for success with machine learning models is hyperparameter tuning. Hyperparameters are generally the most fundamental configurations that regulate the behavior of a machine learning model. Examples include learning rate, depth of decision trees, number of estimators, and many more. Optimizing these hyper-parameter values could highly influence both model accuracy and efficiency. Advanced hyperparameter optimization techniques, especially Bayesian Optimization, Genetic Algorithms, and grid search methods, have over the past couple of years begun to make the process easier by automating and streamlining it. These methods are able to explore the most efficient hyperparameter space with the aim of optimal performance of the model on various datasets [4], [6]. Essentially, it has been found that applying advanced hyperparameter tuning to stacking models can yield enormous gains in fraud detection performance, typically benchmarked with the ROC AUC score, a popular metric used to measure the effectiveness of the model in distinguishing between fraudulent and legitimate transactions [7].

Fraud Detection Systems require scalability and high-volume data processing in real-time since financial systems generate millions of transactions each day. Since fraudsters continuously evolve their techniques, the machine learning models should be adaptive to keep pace with the detection of emerging trends and patterns in fraud. This means that models always need updates, retraining with new data and tuning generalization performance of the models to unseen scenarios [4]. In this paper the challenges identified are solved by a novel approach considering stacking models combined with an improved hyperparameter optimization inside of a set of base learners: Logistic Regression, XGBoost, CatBoost and LightGBM. This approach will tune the hyperparameters and overcome inherent issues associated with dataset imbalance, thus proving to be more efficient, scalable, and accurate in fraud detection systems by providing a robust solution against fraud transactions within the dynamic digital space.

The organization of the rest of the paper is as follows: Section II discusses related work, considering an in-depth analysis of fraud detection techniques' current status. Section III describes the methodology followed, describing the design of the stacking model along with its optimization process. The results obtained from the experimental evaluation are discussed in Section IV, while Section V concludes the paper and provides insights for future research prospects in this area.

II. Literature Review

A. Overview of Optimization Techniques

The area of optimization has become a stronghold for better performance in machine learning models in different application domains. This often includes tuning the model parameters or hyperparameters to maximize or minimize some objective functions, normally related to model accuracy or error. In large, there are two major levels where optimization techniques are used in machine learning: first, during model training, the adjustment of the parameters to fit the data; second, when doing hyper-parameter tuning, where non-learned parameters like learning rate, number of layers, and so on are tuned for optimal performance of the model [8], [9].

While the field has evolved from traditional approaches like Gradient Descent to other, more sophisticated metaheuristic techniques like Genetic Algorithms (GA) and Particle Swarm Optimization (PSO), developed to deal with various problems such as non-convexity, high-dimensional search space, multi-objective optimization problems [4]. They have been spectacularly improved in domains like fraud detection, healthcare, and financial forecasting, where the data is complex, noisy, and different models are called for based on fine-tuning to achieve high accuracy.

B. Hyperparameter Optimization

HPO is crucial to realize significant improvements in machine learning, which, in application domains such as fraud detection, relies on the accuracy of models. While the neural network parameters are learned during training, the Hyperparameters must be predefined and they limit the learning process, model complexity, and generalization ability [4]. Efficient hyperparameter tuning ensures that the model does not only fit well to training data but generalizes to unseen data, hence improving its overall robustness.

The early models of hyperparameter optimization include Grid Search and Random Search, which systematically search through predefined hyperparameter spaces. Grid Search tests all hyperparameter combinations exhaustively over a specified range, whereas Random Search picks random hyperparameter configurations [9]. While Grid Search ensures that the space is well-covered, this process is computationally expensive, especially for large, high-dimensional datasets, whereas Random Search often outperforms Grid Search in covering a wider area of the hyperparameter space with fewer iterations.

However, both approaches are not very efficient in the case of large hyperparameter spaces, since their convergence to the optimal configuration requires hundreds of evaluations. This consideration motivated the development of more advanced optimization strategies based on intelligent search mechanisms.

Recent development in hyperparameter optimization techniques has derived more sophisticated methods like Bayesian Optimization, Hyperband, and Evolutionary Algorithms [10]. Such methods have been adopted in various domains, including finance and healthcare, since they provide immense power for efficient model optimization in problems involving high stakes, such as fraud detection [11].

C. Overview of Machine Learning

Machine learning, or ML, has been described as one of the most revolutionary technologies in the modern era, deeply embedded in a wide variety of applications that range from segments of healthcare classification [12] and financial domains such as tax collection [13] and cybersecurity [1]. Its ability to process large datasets and find some meaning in them has turned it into an indispensable tool for troubleshooting different problems in reality. The evolution of machine learning models has equipped industries with the power to make data-driven decisions and also unlock predictions at an accuracy level previously unimaginable. ML algorithms are necessarily applied the world over to solve problems in need of precision and scalability, such as fraud detection, risk management, and modeling customer behaviors [7]. Different forms of machine learning have been created to address various kinds of issues.

D. Ensemble Learning

The technique of stacking models, particularly ensemble learning, has become increasingly popular in recent years for enhancing predictive accuracy. This method involves combining various base learners such as Decision Trees, XGBoost, and CatBoost with a meta-learner to achieve improved accuracy and generalization ability [14]. Ensemble learning outperforms different models that work on various applications ranging from fraud detection to credit scoring.

With the development of optimization techniques, computational power, and integration of ensemble learning techniques over the past few years, the performance and efficiency of machine learning models have improved significantly. This has enabled algorithm stacking or boosting with frameworks such as XGBoost, CatBoost, and LightGBM, allowing models to capture intrinsic, nonlinear relationships within the data that were unprecedented when compared to traditional models, including logistic regression or decision trees [4], [6]. These have also proved to be particularly efficient for high-dimensional, imbalanced datasets that come up in applications related to credit-card fraud detection [5].

E. Logistic Regression

Logistic Regression is among the simplest algorithms of machine learning; it finds its applications, especially in binary classification problems [1]. It calculates the probability of a certain input to belong to one of two classes based on the logistic function. Generally speaking, LR models are linear; that is, there is a linear relationship between input variables and the log-odds of the dependent variable. Due to its simplicity, logistic regression has been widely used in many applications, including medical diagnosis, credit scoring, and fraud detection, since interpretation at low computational cost is possible [4].

However, logistic regression has one major disadvantage: it cannot model data with non-linear trends, which limits its effectiveness on more complex datasets. Hence, logistic regression is common in ensemble models, often used as a meta-learner to combine the predictions of more potent models. It plays a leading role in synthesizing outputs in stacking ensembles, where base learners like decision trees or gradient boosting models create better final predictions. This method has been especially useful in domains like fraud detection, where the relationships among features are highly complex and data are typically imbalanced [1].

F. Gradient Boosting Models

A collection of weak learners in a boosted model collaborates to form a robust predictive model. At each iteration, the model minimizes errors by adjusting for the shortcomings of earlier models. Consequently, such models excel in applications like fraud detection and similar challenges, particularly when dealing with large and imbalanced datasets. Three popular algorithms for gradient boosting are XGBoost, CatBoost, and LightGBM.

1) XGBoost: XGBoost is one of the most efficient and effective techniques that can deal with both classification and regression challenges. It is a scalable machine learning system for faster and accurate predictions [15]. As an example, some of the useful features that make XGBoost very powerful concerns with its capabilities to handle sparse data, missing values since it can handle large datasets with high efficiency, with considerable usefulness in real-world applications such as fraud detection. In XGBoost, sequential trees are created, each of which tries to correct errors of the previously misclassified one. The boosting approach lets it generate high accuracy models and makes them less prone to overfitting, especially when combined with regularizations such as L1 or L2 [4].

XGBoost has gained extraordinary popularity because it is outstandingly competitive in many machine learning competitions, and could be easily scaled to distribute systems and GPUs [15]. Because of its capability to model complicated interaction in imbalanced data, XGBoost was the widely used base learner for fraud detection systems in multilayered models. It is the ensemble technique whose regularization and parallelization features make it particularly fitted to real-life applicative scenarios involving large-scale data in circumstances where swift fraud detection is crucial [5], [14].

2) CatBoost: CatBoost: this is a relatively new-and-awesome gradient boosting algorithm, which gained popularity quite fast, since it handles categorical features better than almost any other algorithm in machine learning. In traditional models, for instance, it was common to perform basic preprocessing either by onehot encoding or label coding so that the representations were numerical values for categorical features; this kind of transformation typically increases the dimensionality of a dataset and introduces some risk related to overfitting . CatBoost handles it through native processing of categorical data that, in turn, reduces preprocessing time and enhances the model.

Probably the most characteristic feature of CatBoost is that it can handle categorical data efficiently, which by default would otherwise need to be converted to one-hot encoding in other machine learning libraries. This makes CatBoost very useful in fields like e-commerce, finance, or fraud detection, where datasets often include naturally categorical variables. Besides this, the CatBoost library uses a lot of special tricks to reduce overfitting and improve the generalization of predictive models, which is highly recommended for large-scale datasets used in the detection of rare events such as fraudulent transactions [7], [14]. Its treatment of imbalanced data further contributed to its impressive performance and hence its popularity in several industries.

3) LightGBM: Probably the most important and challenging problem in machine learning, especially in fraud detection, is that of imbalanced datasets. Usually, fraudulent transactions are a rare class, while more than 95% of transactions are usually legitimate. This leads to biased models toward the majority class. Therefore, essentially, models fail to recognize fraud, and missed fraudulent transactions are often present that result in financial losses. In order to handle it, SMOTE - Synthetic Minority Over-sampling Technique and undersampling techniques are used in order to balance the dataset [1]. Apart from SMOTE, there are scaling techniques, such as Robust Scaler, for standardizing features and reducing the perceived effect of outliers. Robust Scaler scales the features based on interquartile range, which involves less sensitivity to extreme values . This is especially useful for fraud detection datasets, since transaction amounts may have a huge span.

G. Robust Scaler and Imbalance Handling

One of the most challenging tasks for machine learning, especially in fraud detection, is dealing with imbalanced datasets. Fraudulent transactions are generally seldom compared to the normal good ones and often result in models biased toward the majority class. The upshot is that most models thus fail in fraud detection and result in missed fraudulent transactions that translate into financial losses. To address this issue, techniques like SMOTE (Synthetic Minority Over-sampling Technique) and undersampling are used to balance the dataset [1].

III. Methodology

The sections below describe the methodology that was implemented to come up with a workable system for credit card fraud detection. It comprises major steps in methodology such as dataset preprocessing, feature engineering, handling class imbalance, building the stacking ensemble model, and tuning hyperparameters using a Bayesian Optimization Framework. Fig. 1 shows schematic representation of the credit card fraud detection system workflow.

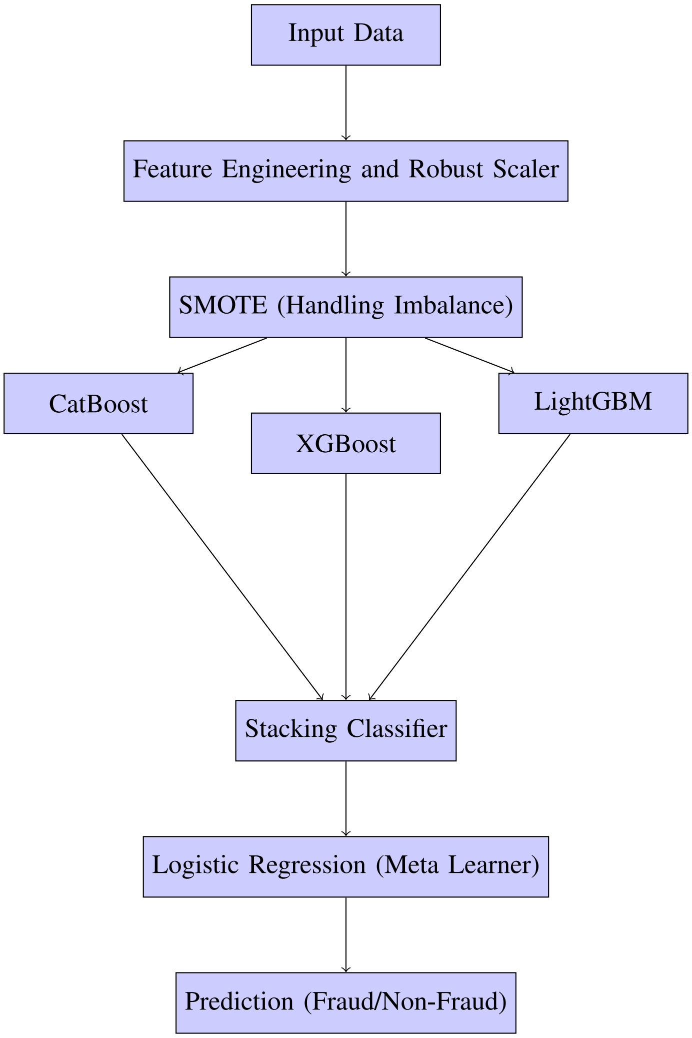
Input Data
Feature Engineering and Robust Scaler
SMOTE (Handling Imbalance)
CatBoost
XGBoost
LightGBM
Stacking Classifier
Logistic Regression (Meta Learner)
Prediction (Fraud/Non-Fraud)

Fig. 1. Schematic representation of the credit card fraud detection system workflow.

A. Dataset Description

Another popular benchmark for fraud detection, the Kaggle Credit Card Fraud Detection dataset [16], is used to benchmark an algorithm due to real-world representation of highly imbalanced datasets of financial transaction data. It contains 284,807 transactions out of which only 492 transactions (approximately **0.172%**) are classified as fraudulent. This highly imbalanced distribution represents real-world scenarios accurately, because the fraudulent transactions occur very rarely, though they bear a high financial impact [1].

In this dataset, there are in total 30 features, from which most of them have been anonymized with Principal Component Analysis - PCA so that individual privacy might not be disclosed. These anonymized features are named V1 to V28, each one corresponding to the principal components obtained from the original data. The other important features of the dataset include the following:

- Time: The time elapsed in seconds since the first transaction in the dataset. This feature can reveal temporal trends, such as specific periods with increased fraudulent activity.
- Amount: The monetary value of the transaction, which can help detect abnormal spending behavior potentially indicative of fraud.
- Class: The target variable, where 1 denotes a fraudulent transaction and 0 denotes a legitimate transaction.

This data distribution is highly imbalanced; most of the transactions here are in the non-fraudulent group. A highly imbalanced, skewed distribution sometimes results in traditional machine learning modeling that is biased toward the majority class, which may not provide the best performance for fraud detection.The distribution of fraudulent vs. non-fraudulent transactions is shown in Fig. 2. Therefore, this can be treated with different techniques such as SMOTE to balance the dataset during training of the model so that it would become more efficient in detecting fraudulent activities.

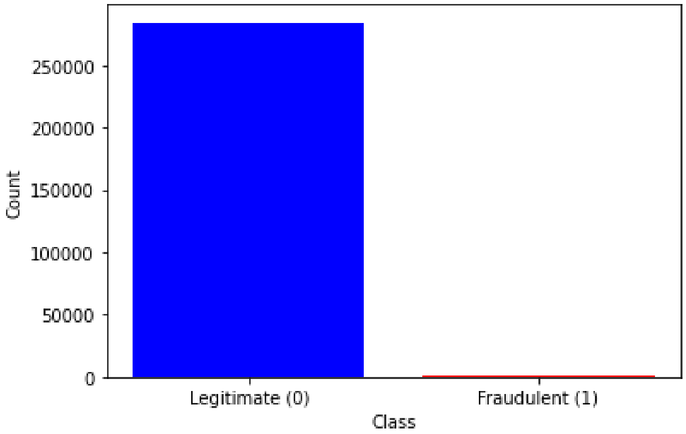

Fig. 2. Distribution of fraudulent vs. Non-fraudulent transactions.

The following is a statistical overview of the Kaggle Credit Card Fraud Detection dataset, summarized in Table I. It focuses on several critical aspects such as the total number of transactions; the number of fraudulent transactions; and the percentage of fraud within the dataset. The high degree of class imbalance may create some problems in getting truly useful models of machine learning.

To enhance the dataset for model training, feature engineering was applied. Key transformations include:

1) Time transformation: The ‘Time’ feature was converted into an ‘Hour’ feature, capturing the hour of the day each transaction occurred. This transformation can help reveal temporal patterns in fraudulent behavior.

TABLE I. Statistical Information of Kaggle Credit Card Fraud Detection Dataset

Item	Value
Total Number of Transactions	284,807
Number of Fraudulent Transactions	492
Percentage of Fraudulent Transactions	0.172%
Number of Features (Including Label)	31
PCA Principal Components	28
Transaction Amount Column	1
Time Feature Column	1
Label Column	1

2) Amount transformation: The 'Amount' feature was log-transformed into 'Amount_log' to reduce skewness in the transaction amounts, improving model performance and reducing the impact of extreme values on the learning process [1], [6].

Preprocessing and feature engineering steps are essential to improve the predictive ability of machine learning models , especially when dealing with the uneven nature of the dataset.

B. Handling Class Imbalance

The extreme class imbalance in this dataset has a number of challenges when it comes to the training of machine learning models. If it is not treated properly, the model becomes biased towards the majority class, which consists of all the legitimate transactions, thus making its performance in fraud detection extremely poor. In this work, the Synthetic Minority Over-sampling Technique (SMOTE) was used. SMOTE is an algorithm that generates synthetic examples of the minority class through interpolation. This effectively increases the number of fraudulent transactions in the training set without actually over-sampling by mere replication of existing records [4]. Moreover, the use of SMOTE is very important for improving the model's performance in correctly classifying fraudulent transactions in high imbalanced datasets [7].

C. Stacking Ensemble Model

To enhance the performance of the fraud detection system, a stacking ensemble model was implemented. Stacking is a method in which several base learner predictions combine using some meta-learner. It helps improve generalization of the model by leveraging each base learner's strengths [4].

The base learners in this stacking ensemble include:

- XGBoost: A highly efficient gradient boosting algorithm known for its speed and ability to handle sparse and large datasets [15].
- CatBoost: Another gradient boosting algorithm that handles categorical features natively without requiring extensive preprocessing, making it particularly suited for datasets with many categorical variables [17].
- LightGBM: Known for its scalability and speed, LightGBM uses a histogram-based approach to efficiently handle large datasets with low memory consumption [18].

These predictions then become the input for the Logistic Regression meta-learner, which then takes this input to give a final classification. Logistic Regression was chosen as the meta-learner due to its simplicity and how effective it is in combining outputs from multiple classifiers in stacking models [4]. The integration of base learners with the meta-learner gives us a robust fraud detection system that can enhance the identification of fraudulent transactions while reducing false positives [5].

D. Hyperparameter Tuning

In order to further enhance the performance of this model, hyperparameter optimization was done using a Bayesian Optimization Framework. It is important to realize that the learning rates, a tree's maximum depth, and the number of estimators are very important hyperparameters, having a great bearing on the effectiveness of any machine learning model. In particular, Optuna is an advanced hyperparameter optimization library that uses Bayesian optimization to dynamically adjust the search space and focus sampling efforts in the most promising areas of hyperparameter space. This allows the approach to be dynamic in its optimization, hence is computationally lighter compared to other approaches based on traditional grid search or random search methods [4], [6].

For each base learner, key hyperparameters were tuned:

- XGBoost: `n_estimators`, `max_depth`, `learning_rate`, `subsample`.
- CatBoost: `iterations`, `depth`, `learning_rate`, `l2_leaf_reg`.
- LightGBM: `n_estimators`, `num_leaves`, `learning_rate`, `feature_fraction`.

Optuna's efficient search process has focused on maximizing the ROC AUC score, a key metric in fraud detection. The ROC AUC score is particularly suitable for evaluating models in imbalanced datasets because it captures the trade-off between true positive and false positive rate across all classification thresholds. Optimized hyperparameters led to stellar improvements while ensuring that the model could cope with the stacking imbalance issue of the dataset with higher accuracy. [4], [5].

E. Model Evaluation and Metrics

Precision, recall, the F1-score, and especially the ROC AUC score were calculated on the final stacking model. In fraud detection, the ROC AUC score is particularly important given that it serves as a measure of balancing between fraudulent and legitimate varies. Better performance is reflected by higher values of AUC. Moreover, metrics such as precision and recall are used to correctly evaluate the performance of the model in terms of reducing false positives while trying to identify fraud. The F1-score offers a balance between precision and recall and is leverage to provide an overall insight into the model's performance on this imbalanced dataset [7].

The ROC curve was plotted to visualize the model's performance across various decision thresholds. The curve provides valuable insights into the trade-offs between sensitivity (recall) and specificity, allowing for an informed selection of the threshold that best meets the system's operational needs.

F. Pseudo Code

In this section, we will outline the pseudo-code for our proposed model.

Algorithm 1 Credit Card Fraud Detection Using Stacking Models and Hyperparameter Tuning

```
Input: Kaggle Credit Card Fraud Detection dataset
Output: Fraud/Non-Fraud Prediction
procedure MAIN PROCEDURE
    Load dataset creditcard.csv
    Feature Engineering:
    Extract 'Hour' from the 'Time' feature
    Apply log transformation on 'Amount' and rename to 'Amount_log'
    Drop original 'Time' and 'Amount' columns
    Data Preprocessing:
    Apply RobustScaler on all features except the target ('Class')
    Separate features (X) and target (y)
    Train-Test Split:
    Split the dataset into training and testing sets using train_test_split, stratifying by the target 'y'
    Handling Imbalance:
    Use SMOTE to oversample the minority class in the training set
    Hyperparameter Optimization:
    Define the objective function for optimization:
    Tune parameters for XGBoost, CatBoost, LightGBM using Bayesian Optimization
    Define parameter ranges: n_estimators, max_depth, learning_rate, etc.
    Evaluate model using ROC AUC score
    Model Training with Stacking:
    Initialize base models: XGBoost, CatBoost, LightGBM
    Define a Stacking Classifier with the base models and a Logistic Regression final estimator
    Perform 5-fold cross-validation with StratifiedKFold
    Model Fitting:
    Train the stacking model using the training data
    Predict fraud probabilities on the test set
    Evaluation:
    Calculate the ROC AUC score for model evaluation
    Generate a classification report with precision, recall, F1-score
    Plot and save the ROC Curve
    Error Handling:
    If an error occurs, log the error with traceback information
end procedure
```

IV. Results and Discussion

This section presents the results of our stacking ensemble model on the Kaggle Credit Card Fraud Detection dataset, followed by a detailed comparison with the results from Jiang et al. (2023) [19]. The key performance metrics considered are Precision (PR), Recall (RC), F1-Score (F1), and ROC AUC. Additionally, the ROC curve of our model is provided for visualization of its classification performance.

A. Our Model's Performance

In summary, as shown in Table II, the performance from our stacking ensemble model using Logistic Regression as the meta-learner and base learners of XGBoost, CatBoost, and LightGBM. Our model yields a performance of ROC AUC score of 0.9887 that was strong in the classification and yields good separation between fraudulent and legitimate transactions for many thresholds. This reflects a high score for the model's precision in recognizing both classes of transactions even from an imbalanced dataset where fraudulent transactions are few.

The high Recall score points to the ability of the model to detect a large proportion of fraudulent transactions, something that in fraud detection scenarios is very important because missing fraud cases results in significant financial losses. On the other hand, the model keeps a strong Precision, meaning it finds most of the fraudulent-flagged transactions to be actually fraud. This trade-off between precision and recall is important for minimum false positives, which, in turn, keeps operational costs lower for the review of legitimate transactions that have been wrongly flagged as fraud.

Furthermore, the F1-Score shows the general performance of the model in its balance of the detection of fraudulent and legitimate transactions. This is especially achieved using the stacking ensemble approach, which aggregates the strengths of several machine learning models. In such a method, it allows base learners at different layers to focus on various aspects of the data, improving their capacity for better detection of subtle patterns in fraudulent activities that may get lost with just one model. Meanwhile, it uses the integration or combination of different algorithms: XGBoost, CatBoost, and LightGBM, which then serves as a versatile and adaptive model, since the best performance will be achieved at everything related to the types of transactions.

TABLE II. Performance of Our Stacking Ensemble Model

Metric	Class 0 (Legitimate)	Class 1 (Fraudulent)	Macro Avg	Weighted Avg
Precision	1.00	0.88	0.94	1.00
Recall	1.00	0.86	0.93	1.00
F1-Score	1.00	0.87	0.94	1.00
ROC AUC		0.9887		

Fig. 3 presents the ROC curve further enforces the fact of high discriminatory power of our model. The magnitude of the AUC at a value of 0.9887 was impressive and ensures excellent classification performance over an immense variance in decision thresholds. This high score essentially means that the model is highly effective in discriminating between fraudulent and legitimate transactions by balancing the sensitivity with the specificity of the model. The fact that such high classification performance can be steadily kept with different thresholds is core in guaranteeing that the model can minimize prematurely both false positives and false negatives. For this reason, the system proves to be not only quite accurate but also reliable in real-life applications where fraudulent activities need to be detected with precision while disrupting legitimate transactions as little as possible.

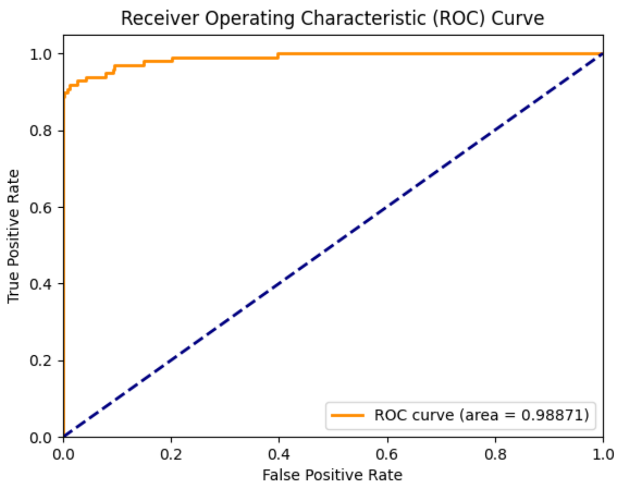

Fig. 3. ROC Curve of our stacking ensemble model

B. Comparison with Jiang et al. (2023)

Apart from the assessment of the performance of our stacking ensemble model, it was relevant to put it in comparison with other state-of-the-art methodologies within this related Research area: credit card fraud detection. Among the rela- To tively new and more sophisticated ones comes the attention Mechanism-based unsupervised anomaly detection network UAAD-FDNet, derived from the work of Jiang et al. (2023) [19]. This model leverages unsupervised learning. combined with attention mechanisms to perform anomaly detection. extend fraud detector performance.

Table III compares the performance of our proposed approach against these existing approaches, including results derived directly from the work of Jiang et al. (2023) [19] for their UAAD-FDNet model and other baseline approaches.

TABLE III. COMPARATIVE RESULTS ON KAGGLE CREDIT CARD FRAUD DETECTION DATASET (RED BOLD INDICATES OPTIMAL RESULTS)

Method	PR	RC	F1	AUC
SVM	0.8854	0.7215	0.7951	0.8586
DT	0.8837	0.7269	0.7977	0.8598
XGBoost	0.8955	0.7280	0.8031	0.8649
KNN	0.9032	0.7268	0.8055	0.8709
RF	0.9112	0.7343	0.8132	0.8827
LSTM	0.9073	0.7391	0.8146	0.8845
CNN	0.9217	0.7453	0.8242	0.9075
MLP	0.9262	0.7461	0.8265	0.9094
AE	0.9528	0.7495	0.8390	0.9279
UAAD-FDNet w/o FA (Jiang et al.)	0.9756	0.7514	0.8489	0.9437
UAAD-FDNet w/ FA (Jiang et al.)	**0.9795**	0.7553	0.8529	0.9515
OptiStack (Ours)	0.88	**0.86**	**0.87**	**0.9887**

C. Discussion of Comparative Results

Table III highlights the comparative performance of our stacking ensemble model and the various models evaluated by Jiang et al. (2023). Our model performs competitively, especially in terms of Recall, F1-Score and ROC AUC, while Jiang et al. (2023)'s UAAD-FDNet model achieves the best performance in terms of Precision.

a) Precision: The UAAD-FDNet w/ FA model in Jiang et al. (2023) was able to achieve an accuracy of 0.9795, while that of our model stood at 0.88. This shorthand form of explanation clearly states that their model returns many fewer false positives and is only suspecting a tiny fraction of legitimate transactions as fraudulent. Precision does retain its value in terms of minimum damage caused to valid users, which the model does excel at.

b) Recall: Our model further outperforms the models of UAAD-FDNet for Recall, having a value of 0.86 versus 0.7553 for UAAD-FDNet w/ FA. This is indicative of the fact that our model can capture a larger portion of fraudulent cases- a critical factor in fraud detection systems, since false negatives (cases of missed fraud) are expensive.

c) F1-Score: is a balance between precision and recall. Jiang et al. (2023)'s UAAD-FDNet w/ FA model has an F1-Score of 0.8529, slightly lower than the 0.87 from our model. That would say, while our model sacrifices some on precision, it balances capture of fraud with low false positives better.

d) ROC AUC: Our model outperformed all models on the ROC AUC score, including Jiang et al.'s UAAD-FDNet models, with the best AUC reached at 0.9515 while our best ROC AUC score was 0.9887. The higher the value of the ROC AUC score, the greater the generalizability of a model for a wide range of decision thresholds when classifying fraudulent and legitimate transactions.

D. Strengths and Areas for Improvement

Our model is very powerful in finding fraudulent transactions, as can be shown by high Recall and ROC AUC. In as far as Precision is concerned, this model has room for further improvement. It is beaten by the UAAD-FDNet model. There is an open challenge in fraud detection where reducing the number of false positives with a high recall is a challenge. The future directions could be the use of hybrid models or further tuning of hyperparameters in a way that precision improves without hurting the recall. Additionally, more sophisticated data augmentation or the usage of fraud detection systems in real time may further improve the robustness of this model. Furthermore, using extra features related to temporal or behavioral patterns may provide the key toward much better overall prediction accuracy. Continuous training of the models with updated fraud patterns can maintain the adaptability of the system against novel fraudulent activities. Last but not least, deeper neural networks or graph-based models could present opportunities for improved results.

V. CONCLUSION

This work has successfully performed a stacking ensemble model that comprises of XGBoost, CatBoost, and LightGBM; deal with Logistic Regression as the meta-model, for credit card fraud detection. Advanced hyperparameter optimization using Bayesian Optimization Framework has been performed very successfully, which yields a very big boost in performance compared to usual single-model solutions.

It shows the ability of the model to discriminate reliably between fraudulent and legitimate transactions, as can be seen by the model's performance, especially the ROC AUC score of 0.9887. The model's high recall of 0.86 and F1-score of 0.87 demonstrate a critical perspective in fraud detection systems where missed fraud cases should be as low as possible. Although the precision is somewhat lower than that of some competing methods, the overall balance of the model ensures that this can be safely deployed to environments that prioritize fraud identification without overwhelming users with false positives.

Compared to the state-of-the-art UAAD-FDNet from Jiang et al. (2023) [19], our results are competitive and mostly surpassing. Although the approach by Jiang et al. was able to show high precision, ours showed better all-around classification capabilities, especially with recall and AUC, making it highly applicable for fraud detection in real-world settings.

Besides achieving a good performance, the proposed method outperforms others by showing great potential of ensemble models complemented with effective techniques of hyperparameter tuning and handling imbalanced data like SMOTE. These methods address critical fraud detection challenges: class imbalance and precision for detecting rare fraudulent transactions.

Future work may focus on further ehancment of precision, perhaps with even newer and more complicated ensemble methods, or working towards real-time deployment in high-throughput finance systems. Another interesting aproach might be the application of this method on fraud detection problems outside this dataset for verification across different domains.

In the end, out approach of OptiStack issues out to be robust and flexible for detecting credit card fraudulent transactions. The base formed will satisfy advanced stacking methods and hyperparameter tuning, hence potentially helping enhance financial security and combat fraud.

References

[1] C. Phua *et al.*, "A comprehensive survey of data mining-based fraud detection research," *Artificial Intelligence Review*, vol. 33, no. 3, pp. 229–246, 2010.

[2] S. Xu, G. Liu, Z. Li, L. Zheng, and S. Wang, "Random forest for credit card fraud detection," *IEEE International Conference on Networking, Sensing, and Control*, pp. 1–6, 2018.

[3] M. Al-shabi, "Credit card fraud detection using autoencoder model in unbalanced datasets," *Journal of Advanced Math and Computer Science*, vol. 33, pp. 1–16, 2019.

[4] E. Esenogho, I. Mienye, T. Swart, K. Aruleba, and G. Obaido, "A neural network ensemble with feature engineering for improved credit card fraud detection," *IEEE Access*, vol. 10, pp. 16 400–16 407, 2022.

[5] J. A. Moses Ashawa, Jude Osamor, "Enhancing credit card fraud detection: An ensemble machine learning approach," *Big Data and Cognitive Computing*, vol. 8, no. 1, 2024.

[6] R. Al-Sulaiman and P. Sant, "Credit card fraud detection using stacking models with enhanced swarm optimization," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 14, no. 3, 2023.

[7] S. Makki, Z. Assaghir, Y. Taher *et al.*, "An experimental study with imbalanced classification approaches for credit card fraud detection," *IEEE Access*, vol. 7, pp. 93 010–93 022, 2019.

[8] J. Nocedal and S. J. Wright, *Numerical optimization*. Springer Science & Business Media, 2006.

[9] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, no. 2, pp. 281–305, 2012.

[10] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," *Advances in neural information processing systems*, vol. 25, 2012.

[11] A. Alshehri and R. Ahmed, "Fraud detection in financial institutions using machine learning algorithms," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 13, no. 11, 2022.

[12] H. Mustafa, C. Mohamed, O. Nabil, and Noura, "Machine learning techniques for diabetes classification: A comparative study," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 9, pp. 785–790, 2023.

[13] N. Ourdani, M. Chrayah, and N. Aknin, "Towards a new approach to maximize tax collection using machine learning algorithms," *IAES International Journal of Artificial Intelligence*, vol. 13, no. 1, pp. 737–746, 2024.

[14] F. Alomari and S. Alqarni, "Credit card fraud detection using stacked machine learning algorithms: A comparative study," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 13, no. 10, 2022.

[15] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, 2016.

[16] A. D. Pozzolo, "Credit card fraud detection dataset," https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud, 2015.

[17] A. V. Dorogush, V. Ershov, and A. Gulin, "Catboost: gradient boosting with categorical features support," *arXiv preprint arXiv:1810.11363*, 2018.

[18] G. Ke *et al.*, "Lightgbm: A highly efficient gradient boosting decision tree," *Advances in neural information processing systems*, vol. 30, pp. 3146–3154, 2017.

[19] M. Jiang, Y. Liu, J. Yin, and C. Xiao, "Credit card fraud detection based on unsupervised attentional anomaly detection network," *AI*, vol. 3, no. 1, pp. 1–15, 2023.