DETECTING REAL-TIME E-COMMERCE FRAUD WITH ADVANCED ENSEMBLE META-MODELING

Mariyam Majidha[1], Aishath Athoofa Jalal[1], Muhammad Mukhlis Amrullah[2], Muhammad Adib Mohd Akbar[1], Linda Johnson[1], Nurshakira Adriana Abu Bakar[1], and Riyaz Ahamed Ariyaluran Habeeb Mohamed[1]

[1]Faculty of Computer Science and Information Technology, Universiti Malaya, Kuala Lumpur 50603, Malaysia.

[2] Faculty of Economics and Business, Universitas Brawijaya, Malang 65300, East Java, Indonesia

Emails: 22095809@siswa.um.edu.my*, 22097847@siswa.um.edu.my, mukhlisamrullah@ub.ac.id, sim190033@siswa.um.edu.my, s2186656@siswa.um.edu.my, s2021234@siswa.um.edu.my, riyaz@um.edu.my

Abstract

As e-commerce transactions continue to surge, the threat of fraud has escalated, posing significant challenges due to class imbalances, rapidly evolving fraud tactics, and the critical need to balance false positives and negatives. This study effectively addresses these challenges through an advanced ensemble stacking approach, integrating Support Vector Machine (SVM), Neural Network, Gradient Boosting, and AdaBoost as base models, with a Random Forest as meta-model to deliver final predictions. Using an e-commerce transaction dataset, our approach achieved 99.87% accuracy, significantly outperforming individual models. The meta-model further demonstrated 0.99 precision, 0.98 recall, and 0.99 F1-score for fraud cases (Class 1), highlighting its strong ability to accurately detect fraudulent transactions while minimizing false positives and false negatives. While SVM had the longest execution time, the Neural Network was the most efficient, and AdaBoost contributed the most to the meta-model's predictions. Model validation was performed using Local Interpretable Model-Agnostic Explanations (LIME), highlighting Transaction Hour, Transaction Amount, and Account Age Days as key predictive features. The model was successfully deployed to a web-based application, demonstrating real-time fraud detection capabilities. This research offers a robust, interpretable method for e-commerce fraud prevention, potentially reducing financial losses and enhancing online transactions.

Keywords: E-commerce fraud; Ensemble stacking; Machine learning; LIME; Real-time fraud prevention.

1. Introduction

E-commerce has rapidly transformed the global marketplace, offering unparalleled convenience to consumers and new revenue streams for businesses. As online transactions grow aggressively, the risk of fraud increases. According to an article by Mastercard, global e-commerce fraud is on the rise, with losses reaching $41 million in 2022 and expected to exceed $48 billion in 2023 [1]. Because e-commerce involves such a large volume of transactions, fraud strategies are always changing, and there are major financial and reputational concerns involved, detecting fraudulent activities in this domain remains a critical challenge.

Despite the promise of machine learning-based fraud detection systems, several issues persist. A key challenge is the significant class imbalance in e-commerce fraud datasets, where fraudulent transactions represent only a small fraction of the total data [2]. Traditional resampling techniques, such as oversampling and under-sampling, can lead to model overfitting or information loss, reducing predictive performance [3]. Therefore, there is a need for approaches that address class imbalance without compromising model accuracy and generalizability.

Another major issue is the difficulty in minimizing both false negatives (undetected fraud) and false positives (misclassified legitimate transactions). Achieving this balance is essential for maintaining trust and reducing financial losses [4]. Many existing models struggle to strike this balance, resulting in suboptimal performance. Developing more sophisticated models that enhance prediction accuracy while minimizing classification errors is therefore crucial.

Additionally, while machine learning models can achieve high accuracy, their "black box" nature often prevents stakeholders from understanding how decisions are made [4]. This lack of transparency can undermine trust in the model's predictions. As a result, it is essential to validate the decision-making process and provide interpretable insights into the factors influencing fraud detection.

Most fraud detection studies also focus primarily on model evaluation, neglecting the assessment of real-world effectiveness when the model is integrated into a data product or operational environment. This oversight limits the practical applicability of these models, as the complexities of real-time data integration and performance in production are not considered. A more comprehensive approach is needed, one that not only evaluates model performance but also investigates how well the model functions when embedded in a live system.

This study makes a key contribution by tackling the challenge of class imbalance in the dataset using techniques beyond traditional resampling or under sampling. Instead, it leverages robust models inherently suited for imbalanced data, combined with careful hyperparameter tuning, to improve the detection of minority class instances. By employing an ensemble stacking technique, the study enhances model performance by reducing both false positives and false negatives, leading to more accurate fraud detection. The decision-making process is validated through Local Interpretable Model-Agnostic Explanations (LIME), ensuring transparency and insight into how the model generates predictions. Additionally, the research extends beyond evaluation by testing the model within a real-time data product, demonstrating its effectiveness in operational environments. This practical evaluation addresses a gap left by many prior studies that stop at theoretical model assessments.

The rest of this paper is organized as follows. Section 2 reviews relevant works in fraud detection, focusing on existing machine-learning techniques. Section 3 introduces the model stacking methodology, discussing the base models employed, such as Support Vector Machine, Neural Network (MLP Classifier), Gradient Boosting, and AdaBoost. This section also describes the meta-features and meta-model. Section 4 outlines the experimental study, detailing the dataset descriptions, preprocessing steps, model training, evaluation, validation, and testing within a data product environment. Section 5 presents and discusses the results of the experiments, while section 6 concludes the paper and offers directions for future work.

1.0 RELATED WORKS

This section presents critical analysis of related works of machine learning-based fraud detection technologies. The articles reviewed highlight the use of machine learning techniques for detecting financial fraud, especially in the context of fraudulent transactions.

Research by [5] applied machine learning algorithms, including Stochastic Gradient Descent (SGD), Random Forest (RF), Decision Table (DT), J48 Decision Tree, and Instance-Based k (IBk), to detect fraudulent transactions using the UCSD Data Mining Contest 2009 dataset. The Random Forest model delivered strong performance, achieving a precision of 97.60%, recall of 97.90%, F1-score of 97.60%, and a Matthews Correlation Coefficient (MCC) of 49.50%. However, the study faces key challenges. Although the dataset is highly imbalanced, the study does not implement any specific techniques to handle the class imbalance issue, potentially biasing the results toward the majority class.

A recent study by [6] applied a machine learning approach using Gaussian Naïve Bayes, K-Nearest Neighbor, and Fine Tree algorithms to detect e-commerce fraud using the BANKSIM dataset. The Fine Tree model achieved impressive performance, with an accuracy of 99.58%, precision of 99.87%, recall of 99.70%, and an F1-score of 99.79%. However, the study identified several false positive and false negative cases, highlighting the challenges of ensuring reliable fraud detection. Despite achieving a high metrics, the study lacks real-world variables like transaction behaviour, device information, and IP tracking. Additionally, it does not explicitly address class imbalance or focus on minimizing false positives and false negatives, which are critical in fraud detection.

Similarly, a convolutional neural network (CNN) model for e-commerce fraud detection was presented by [7], achieving a fraud prediction accuracy of 85.5%. The CNN model reported a precision of 42.4%, recall of 83.6%, and an F1-score of 56.2%. However, the model faces challenges like cold start problems for new users with no historical data and the use of random under-sampling, which may lead to information loss. Like the previous study, this study also lacks focus on minimizing false positives and false negatives.

In another study by [8], a multi-perspective e-commerce fraud detection method was presented. This method integrates process mining with Support Vector Machines and achieves a high AUC of 93.5%. The control and data flow approach reported a precision of 94.6%, recall of 85.2%, and an F1-score of 89.5%. Although the approach outperforms single-perspective methods, it increases computational complexity, resulting in higher overhead and reduced efficiency in real-time fraud detection. The study also does not address class imbalance or focus on minimizing false positives and negatives.

The work of [9] presented various machine learning algorithms, including Decision Tree, Naïve Bayes, Random Forest, and Neural Network, to detect fraudulent transactions in e-commerce dataset. Using SMOTE to handle class imbalance, the Neural Network with SMOTE achieved an accuracy of 85%, precision of 92.5%, recall of 76.7%, F1-score of 85.1%, and G-Mean of 84.6%. However, without SMOTE, the Neural Network had a higher accuracy of 96%, precision of 97.1%, but a lower recall of 54% and G-Mean of 73.5%. The study highlighted the trade-offs, as SMOTE improved recall and G-Mean but reduced accuracy, raising concerns about overfitting and synthetic noise. The trade-off between false positives and false negatives was not fully addressed.

Merchant fraud detection was addressed by [10] using Random Forest, Decision Tree, and Logistic Regression algorithms. The Random Forest model achieved the highest performance with an accuracy of 84.55%, precision of 87.81%, recall of 82.36%, and specificity of 87.03%. This paper includes features like merchant registration date and IP address, which could be valuable for real-world fraud detection. However, they don't address class imbalance issues or focus on minimizing false positives and negatives.

A deep reinforcement learning approach combined with artificial neural networks was proposed by [11] for fraud detection in e-commerce transactions. The model uses the Artificial Bee Colony (ABC) algorithm to initialize weights and frames the classification problem as a sequential decision-making process. This approach achieved an accuracy of 89.0%, recall of 91.0%, precision of 89.2%, F-measure of 89.0%, and G-means of 89.8%, outperformed other machine learning models like SVM, Naive Bayes, KNN, Random Forests, Logistic Regression, and Decision Trees. The study addresses class imbalance by assigning higher rewards for correctly identifying the minority class. Nevertheless, it does not provide explicit measures for minimizing false positives and negatives.

Various machine learning (ML) and deep learning (DL) algorithms, including Support Vector Machines Logistic Regression, Random Forest (RF), and Convolutional Neural Networks (CNN) were employed by [12] to detect credit card fraud with the European Card Benchmark dataset. The CNN model achieved the highest performance with an accuracy of 99.9%, precision of 93.2%, recall of 77.5%, AUC of 92.9%, and PRC of 81.6%. To address the class imbalance, the study balanced the dataset by removing non-fraudulent transactions. While this approach helped improve model performance, it also introduces challenges such as reducing data diversity. This can potentially limit the model's ability to generalize well to real-world scenarios with varying patterns.

In their framework, [13] applied machine learning algorithms, including Logistic Regression, k-NN, SVM, Decision Tree (DT), Random Forest (RF), SGD, XGBoost, and ensemble models, to detect financial fraud using a European credit card transaction dataset. The ensemble learning model achieved a precision of 89.97%, recall of 97.00%, and an F1-score of 86.02%. To address the class imbalance, they used SMOTE and random under-sampling. These methods improved metrics like recall and F1-score. However, SMOTE may lead to overfitting by replicating minority class patterns, while random under-sampling reduces data diversity, potentially affecting the model's generalization.

For credit card fraud detection, [14] presented machine learning algorithms such as Logistic Regression, Decision Tree (DT), Random Forest (RF), Artificial Neural Network (ANN), and Naive Bayes (NB), achieving the highest performance with the GA + RF model, which reached an accuracy of 99.98%, precision of 95.34%, recall of 72.56%, and F1-score of 82.41%. The study uses European credit cardholder's dataset and SMOTE was applied to manage the class imbalance. Although SMOTE enhanced key metrics like recall and F1-score, it also led to challenges like overfitting by replicating minority class patterns and reducing data diversity.

A range of machine learning algorithms, including Support Vector Machine, k-Nearest Neighbors (KNN), Random Forest (RF), Logistic Regression, Bagging, Boosting, and ensemble models, were utilized by [15] for credit card fraud detection with the European credit cardholders dataset. The RF and Boosting models achieved the highest performance with an accuracy of 99.98%, precision of 99.98%, recall of 99.98%, and F1-score of 99.98%. To address the class imbalance, they applied SMOTE and under-sampling techniques. While SMOTE improved metrics like recall and F1-score, it introduced challenges such as overfitting from replicating minority class patterns and reduced data diversity due to under-sampling legitimate transactions.

A soft voting ensemble model was proposed by [16] for detecting fraudulent credit card transactions on imbalanced data. The study addresses the class imbalance issue by comparing oversampling, undersampling, and hybrid sampling techniques. Several classifiers, including ensemble models, were developed with and without sampling methods, with the highest performance achieved without any resampling technique. The proposed soft voting ensemble outperformed individual classifiers, achieving an accuracy of 99.96%, precision of 98.70%, recall of 79.59%, and an F1-score of 87.64%. However, the resampling techniques used in study such as oversampling can

cause overfitting, undersampling reduces data diversity, and hybrid methods increase computational complexity, limiting scalability and generalizability.

The study conducted by [17], developed an ensemble stacking model that combines Decision Tree, Naive Bayes, K-Nearest Neighbor, and Random Forest to detect fraudulent Bitcoin transactions. Using the Bitcoin Heist Ransomware dataset, which had been filtered down to 381,464 instances, the study addressed class imbalance with Adaptive Synthetic (ADASYN) and Tomek Link techniques. The stacking model achieved impressive results, with an accuracy of 97%, precision of 96%, recall of 98%, F1-score of 97%, AUC-ROC of 99%, and a False Positive Rate (FPR) of 3%. Additionally, the study employed explainable AI (SHAP) to interpret and validate the model's predictions. However, the use of resampling techniques may introduce noise, leading to potential overfitting in real-world applications.

In their efforts to enhance fraud detection [18] employed stacked generalizations combined with various resampling methods. The study tested multiple machine learning algorithms, such as k-NN, Gaussian Naive Bayes, MLP, AdaBoost, GBM, SVM, EasyEnsemble, and Decision Trees, on a highly imbalanced credit card fraud dataset from the Université Libre de Bruxelles. The dataset contained 284,807 entries, with only 492 being fraudulent. The MLP model with 0SMOTE (no oversampling) achieved the best performance, with an accuracy of 99.94%, F1-score of 81.84%, and AUC of 96.0%. However, the resampling techniques explored in the study may introduce noise or lead to overfitting, which could potentially limit the model's ability to generalize to real-world applications.

Financial fraud detection in Chinese listed companies was explored by [19] through a stacking algorithm combining models such as Logistic Regression, SVM, Random Forest, RUSBoost, XGBoost, and Stacking. The stacking model achieved the highest performance with an AUC of 74.2%, and sensitivity and precision both at 76.5%. To address class imbalance, the RUSBoost algorithm was used, which involves under-sampling the majority class. This can lead to information loss and may reduce model performance in real-world applications.

Table 1 provides a summary of the related studies, highlighting common challenges in e-commerce fraud detection. A prominent issue is class imbalance, where fraudulent transactions make up only a small fraction of the dataset, often leading to overfitting or information loss when traditional resampling techniques, such as oversampling or under-sampling, are applied. Additionally, many studies lack adequate focus on minimizing both false positives and false negatives, which is essential for achieving accurate fraud detection and reducing financial losses. Another significant challenge is the "black box" nature of machine learning models, which limits stakeholders' understanding of the decision-making process and reduces trust in the system. Furthermore, most studies concentrate on model evaluation in isolation, without considering the real-world challenges of integrating these models into live systems. This oversight restricts the practical applicability and robustness of the models in operational environments. Addressing these challenges is crucial to enhancing the reliability, transparency, and effectiveness of fraud detection systems in real-world applications.

Table **1**: The Summary of all related papers

Reference	Top Performing Model	Evaluation Matrics											Dataset	Class Balance Technique
		Accuracy	Precision	Recall	F1-Score	Sensitivity	Specificity	AUR	PRC	FPR	G-Mean	MCC		
[6]	Fine Tree Model	99.58%.	99.87%.	99.70%	99.79%	-	-	-	-	-	-	-	BANKSIM dataset	Not Handled
[17]	Stacking Model	97.00%.	96.00%	98.00%	97.00%	-	-	99.00%	-	3.00%	-	-	Bitcoin Heist Ransomware dataset	Oversampling And Undersampling
[19]	Stacking Model	-	76.50%	-	-	76.50%	-	74.20%	-	-	-	-	China Listed Companies Dataset	Undersampling
[9]	Neural Network with SMOTE	85.00%.	92.50%	76.7%.	85.10%	-	-	-	-	-	84.60%	-	E-commerce fraud dataset	Oversampling
[14]	GA + RF Model	99.98%	95.34%	72.56%	82.41%	-	-	-	-	-	-	-	European credit card transaction dataset	Oversampling
[15]	RF and Boosting	99.98%	99.98%	99.98%	99.98%	-	-	-	-	-	-	-		Oversampling And Undersampling
[13]	Ensemble Learning	-	89.97%	97.00%	86.02%	-	-	-	-	-	-	-		Oversampling And Undersampling
[16]	XGBoost, MLP, and KNN)	99.96%	98.70%	79.59%	87.64%	-	-	-	-	-	-	-		Oversampling, Undersampling and Hybrid technique
[12]	CNN Model	99.90%	93.20%	77.5%.	-	-	-	92.9%	81.60%	-	-	-		Undersampling
[10]	Random Forest	84.55%	87.81%.	82.36%	-	-	87.03%	-	-	-	-	-	MasterCard exchanges dataset	Not Handled
[7]	CNN Model	85.50%	42.40%	83.60%.	56.20%	-	-	-	-	-	-	-	Open-source e-commerce service data from 2018	Undersampling
[8]	SVM Model & Control + data flow	-	94.60%	85.20%	89.50%	-	-	93.50%	-	-	-	-	Transaction log dataset	Not Handled

[5]	Random Forest Model	-	97.60%	97.90%.	97.60%	-	-	-	-	-	-	49.50%	UCSD Data Mining Contest 2009 Dataset	Not Handled
[11]	Artificial Bee Colony (ABC)	89.00%	89.20%	91.00%	89.00%	-	-	-	-	-	89.80%	-	Université Libre de Bruxelles credit card fraud dataset	Rewarding correct minority class predictions.
[18]	MLP with 0SMOTE	99.94%	-	-	81.84%	-	-	96.00%	-	-	-	-		Hybrid technique

2.0 PROPOSED FRAMEWORK

In this paper, we employed a sophisticated machine-learning technique known as Stacking. As an advanced ensemble learning approach, stacking improves model performance by strategically combining multiple simpler models into a more robust predictive system, making it an increasingly valuable tool in machine learning applications [20]. This method involves training multiple predictive models, known as base models, on the same dataset. The predictions generated using these models will be used to create a new training dataset to train higher-level models, known as Meta-Models [21]. The main advantage of stacking is its ability to consolidate the predictive capabilities of various models, which increases the system's prediction accuracy and robustness [22].

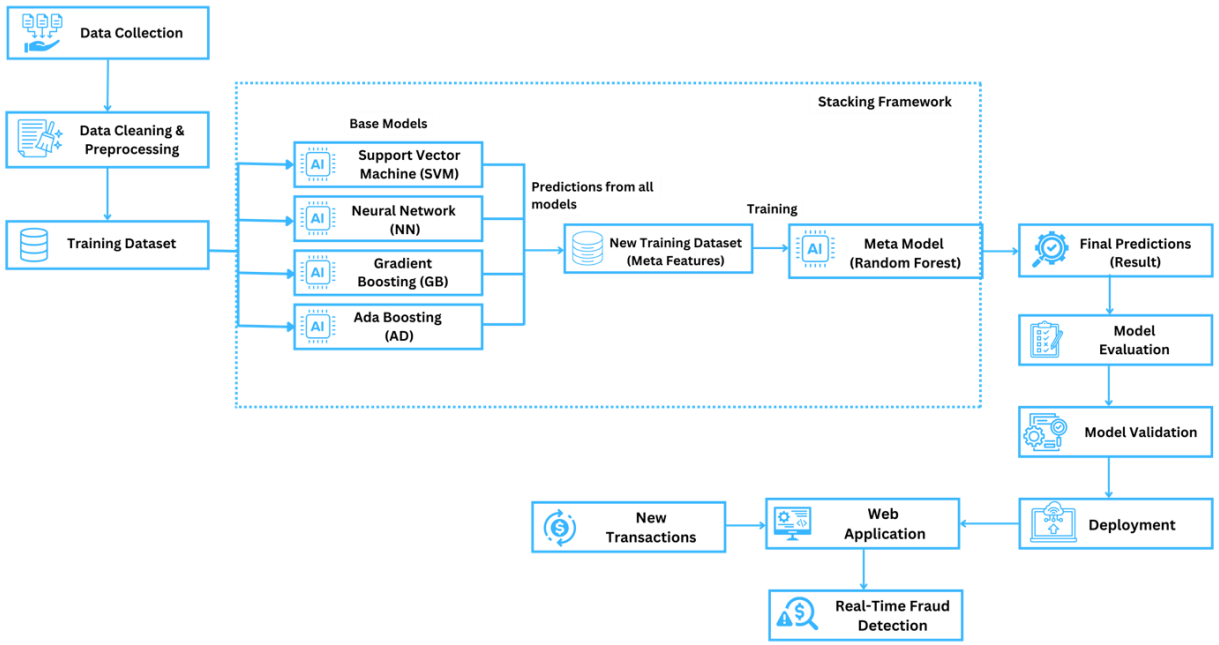

Figure 1: The flow of the model stacking framework

2.1 Base Models

The stacking process begins with the training of multiple diverse independent base models [22]. Each model is selected for its strengths and ability to capture different patterns within the dataset.

2.1.1 Support Vector Machine (SVM)

An SVM model is a versatile supervised learning method commonly used for solving binary classification problems, making it suitable for fraud detection [8]. Its ability to classify user behaviors in complex scenarios ensures reliable results, even in intricate fraud detection cases [8]. SVM works by separating data using a hyperplane or by applying kernel functions to move the data into a higher-dimensional space, enhancing the separation of classes [23]. Due to its flexibility and ease of use, SVM is frequently employed in machine learning applications, including fraud detection [24].

2.1.2 Neural Network (MLP Classifier)

Multilayer perception (MLP) is one of the simple and widely used neural network architectures [25]. In an MLP, data flows sequentially from the input layer to the output layer, with hidden layers in between processing the information [26]. The hidden layer serves as the core of the MLP, transforming input data and passing it to the output layer [27]. Neurons, which act as processing units, are fully connected between adjacent layers, enabling each neuron to pass information to every neuron in the next layer [26]. The network is trained using backpropagation, optimizing weights and biases to minimize error, a crucial factor in accurate fraud detection [28].

2.1.3 Gradient Boosting

Gradient Boosting is a well-known supervised learning algorithm that utilizes the ensemble method of "Boosting" to improve model performance [29]. It combines an ensemble of weak learners, typically decision trees, to handle

nonlinearity effectively, making it a suitable choice for e-commerce fraud [30]. One of its appealing characteristics is that it requires minimal data preprocessing, which simplifies the modelling process while maintaining high predictive accuracy [31].

2.1.4 AdaBoost

AdaBoost, also known as Adaptive Boosting, is a popular algorithm in machine learning due to its ability to quickly enhance model performance [32]. It is extensively used in various domains, particularly in transaction fraud detection [33]. The algorithm builds strong classifiers from weaker ones by iteratively improving the performance of individual classifiers, making it well-suited for both simple and complex problems [34] .

2.2 Meta Features

After training the base models, each base model generates predictions using the original dataset. These predictions, often represented as probabilities, indicate the likelihood that a given instance belongs to a specific class [35]. This process transforms the original feature space into a new, enriched dataset (Meta Features), where the predictions of the base models form the foundation for further classification.

2.3 Meta Model

In this implementation, a Random Forest Classifier is employed as a meta-model. This model is trained on a newly created dataset (Meta Features). The meta model's main function is to learn from the predictions provided by the base models rather than directly from raw input features [36]. Random Forest was selected as a meta-model due to its capacity to handle overfitting and its proficiency in integrating insights from numerous trees to make more accurate predictions [37]. The final predictions are generated using this Metal Model [22].

This paper explores the use of Stacking, a machine learning technique that combines multiple base models (SVM, Neural Network, Gradient Boosting, and AdaBoost) to improve prediction accuracy and robustness. Each base model captures unique patterns, and their predictions form a new dataset of meta-features. A Random Forest classifier, used as the meta-model, learns from these predictions to make final decisions, leveraging the strengths of individual models. This approach enhances performance, reduces overfitting, and simplifies complex fraud detection tasks.

3.0 EXPERIMENTAL STUDY

In this experiment, the effectiveness and interpretability of e-commerce fraud detection models have been evaluated by employing a stacked ensemble method. Four different algorithms, such as Support Vector Machines, Neural Networks, Gradient Boosting, and AdaBoost, are used as base models. These base models are then combined to form the meta-model. Data has been processed and analyzed to explore the distribution of the data and fraud cases, addressing challenges such as class imbalance and high false positive rates. The models were trained and tested using an e-commerce fraud dataset, which provided insights into prediction accuracy and model transparency. The following subsection will provide further details of the experiment setup.

3.1.1 Dataset Descriptions

The dataset used in this study is a synthetically generated transaction data from Kaggle, designed to simulate e-commerce transactions for the purpose of fraud detection. The dataset consists of 23,634 records and 16 columns (features) as detailed in the Table 2.

Table **2**: Shows all the features of the dataset

Features	Description
Transaction ID	A unique identifier for each transaction.
Customer ID	A unique identifier for each customer.
Transaction Amount	The amount of the transaction in the local currency.
Transaction Date	The date when the transaction took place.
Payment Method	The method used for payment (e.g., credit card, PayPal, etc.).
Product Category	The category of the purchased product.
Quantity	The quantity of the purchased product.
Customer Age	The age of the customer at the time of the transaction.
Customer Location	The location (city, state, country) of the customer.

Device Used	The device used for the transaction (e.g., smartphone, tablet).
IP Address	The IP address from which the transaction was made.
Shipping Address	The address to which the purchased product was shipped.
Billing Address	The billing address associated with the payment method.
Is Fraudulent	The target column indicates whether the transaction is fraudulent.
Account Age Days	The number of days since the customer's account was created.
Transaction Hour	The time of the day when the transaction took place.

3.1.2 Data Preprocessing

The data preprocessing involved checking for duplicates, missing values, and NaN entries. We found 116 outlier records in the 'Customer Age' column, with ages ranging from -2 to 8. Since these outliers made up less than 1% of the data and the overall age distribution was normal, they were replaced with the mean age. Figure 2 below shows that 22,412 records in our data are non-fraudulent, while only 1,222 records represent fraudulent cases, accounting for just five percent of all cases. This highlights a class imbalance in the dataset that we need to address.

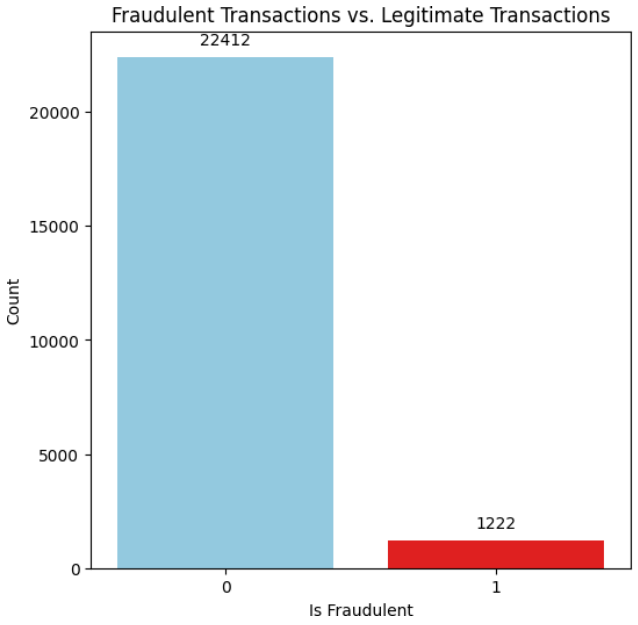

Figure **2**: Distribution of Target Class

Following that, we converted the Transaction Date' to datetime format, created 'Year' and 'Month' columns, and removed unnecessary columns like 'Transaction ID' and 'Customer ID'. Next, we encoded the 'Is Fraudulent' column with Label Encoder (0 for non-fraudulent, 1 for fraudulent) and converted other categorical variables like 'Payment Method' and 'Product Category' to numeric values. For feature selection, we applied Random Forest and used SelectFromModel to identify the most important features. The dataset was split into 80% for training and 20% for testing to prevent data leakage and ensure independent evaluation. Lastly, we standardized the selected features with Standard Scaler to enhance model performance.

3.1.3 Model Training

In this study, we employed several base models (SVM, Neural Network, Gradient Boosting, and AdaBoost) and a meta-model (Random Forest). These models were chosen for their effectiveness in handling complex classification tasks, such as e-commerce fraud detection. We carefully tuned the hyperparameters to enhance the models' performance, with a particular emphasis on addressing class imbalance and improving fraud detection accuracy.

To address the problem of class imbalance, which is often seen in fraud detection data sets, we implemented strategies to focus more on the minority class (fraud cases). For instance, we adjusted the SVM and Random Forest models to give more importance to the minority class by using class weights. This ensured that the models wouldn't overlook important fraud cases. Furthermore, we used grid search to fine-tune the parameters of all the models to

determine the most effective settings for accurate and balanced fraud detection. Table 3 summarizes the optimized parameters for each model.

Table **3**: Hyper Parameter Tunning Ranges and Best Value

Model	Parameters	Best Value	Tuning Range
SVM	C	0.1	[0.1, 1, 10]
	kernel	linear	['linear', 'rbf', 'poly']
	class_weight	{0: 1, 1: 4}	{0: 1, 1: 1}, {0: 1, 1: 4}, {0: 1, 1: 10}
	gamma	scale	['scale', 'auto']
	probability	TRUE	[True, False]
Neural Network	hidden_layer_sizes	(50,)	[(50,), (100, 50), (150, 100)]
	activation	tanh	['relu', 'tanh']
	solver	sgd	['adam', 'sgd']
	alpha	0.01	[0.0001, 0.001, 0.01]
	learning_rate_init	0.01	[0.001, 0.01, 0.1]
	max_iter	200	[200, 300, 400]
Gradient Boosting	n_estimators	300	[100, 200, 300]
	learning_rate	0.01	[0.01, 0.1, 0.2]
	max_depth	3	[3, 5, 7]
	subsample	0.6	[0.6, 0.8, 1.0]
AdaBoost	n_estimators	100	[50, 100, 200]
	learning_rate	0.1	[0.01, 0.1, 1.0, 1.5]
Random Forest (Meta Model)	n_estimators	100	[50, 150]
	max_depth	15	[5, 15]
	min_samples_split	2	[2, 10]
	min_samples_leaf	1	[1, 4]
	class_weight	balanced	['balanced', 'balanced_subsample']
	bootstrap	TRUE	[True, False]

These tuned hyperparameters allowed the models to perform better under the challenge of class imbalance while maintaining high accuracy. The meta-model, trained on the predictions of the base models, further enhanced the robustness of the overall fraud detection system.

3.1.4 Model Evaluation

Several evaluation metrics, including accuracy, precision, recall, and F1 Score, were utilized to assess the models' performance in detecting e-commerce fraud.

- **Accuracy** measures the overall correctness of the model by calculating the percentage of correct predictions (both fraud and non-fraud) out of the total predictions.
- **Precision** focuses on the proportion of correctly identified fraud cases out of all cases the model predicted as fraud, helping to minimize false positives.
- **Recall** (or sensitivity) measures the model's ability to identify all actual fraud cases, reducing false negatives correctly.
- **F1-score** balances precision and recall, offering a single score that accounts for both false positives and false negatives.

3.1.5 Model Validation

In this study, we used LIME to validate the transparency of the models and ensure their decisions were interpretable. LIME helps to provide local explanations for individual predictions by modifying the input data and analyzing the

resulting changes in the model's output [38]. This technique allowed us to examine how the model arrived at its decisions, offering a clear view of the underlying decision-making process.
When LIME was applied to the meta-model, it provided details on the contribution of each base model to the final prediction. Based on these results, we selected the top two contributing base models and then applied LIME separately to those models. This allowed us to analyze further and identify the most significant features influencing the predictions of these top models. This two-step process ensured that both the meta-model and the key base models were transparent and interpretable, addressing the need for clarity and trust in fraud detection systems.

4.0 RESULTS AND DISCUSSION

In this section, we evaluated the performance of five machine learning models, that is Support Vector Machine, Neural Network, Gradient Boosting, Ada Boosting, and a Meta Model. These models were assessed based on key metrics, execution time, and the validation of their decision-making processes. The goal was to identify the most suitable model, balancing predictive power and the ability to manage class imbalances.

4.1 Model Evaluation

To evaluate the performance of the proposed models, a comprehensive comparison of key metrics is conducted, as illustrated in Table 4. This table highlights Accuracy, False Positives, False Negatives, Precision, Recall, and F1-Score for Class 0 and Class 1, offering valuable insights into each model's ability to differentiate between majority and minority classes. This detailed analysis provides a deeper understanding of how well each model handles the classification of both the majority and minority classes.

The SVM model achieved an accuracy of 95.24%. For Class 0, it achieved a precision of 0.95, perfect recall (1.00), and an F1-score of 0.98. However, for Class 1, the precision was 0.90, the recall dropped significantly to 0.08, and the F1-score was 0.14. The model generated 2 false positives and 223 false negatives, indicating that SVM struggled with identifying the minority class. This highlights a potential limitation in the model's ability to correctly classify positive instances for the minority class, despite performing well for the majority class.

The Neural Network demonstrated a slightly higher accuracy of 95.32%. For Class 0, it achieved a precision of 0.95, perfect recall (1.00), and an F1-score of 0.98. For Class 1, the precision was 0.82, the recall was 0.11, and the F1-score improved to 0.20 compared to SVM. The model produced 6 false positives and 215 false negatives, showing a slight improvement in capturing instances from the minority class. However, there is still room for improvement in recall, as the model missed many fraud cases (false negatives).

Gradient Boosting achieved the highest accuracy among the boosting algorithms with 95.37%. The model maintained a precision of 0.95, perfect recall (1.00), and an F1-score of 0.98 for Class 0. For Class 1, the precision was 0.81, the recall improved to 0.12, and the F1-score was 0.22. The model produced 7 false positives and 212 false negatives. This performance indicates a better balance between precision and recall for the minority class, though the number of false negatives shows that further improvement is needed in detecting fraud cases.

Similarly, Ada Boosting achieved an accuracy of 95.37%. The model had a precision of 0.95, perfect recall (1.00), and an F1-score of 0.98 for Class 0. For Class 1, the precision was 0.87, the recall was 0.11, and the F1-score was 0.20. It produced 4 false positives and 215 false negatives. Although AdaBoost did not outperform Gradient Boosting, it still demonstrated reliable performance in handling both majority and minority classes, with a slight focus on reducing false positives over iterations.

The Meta Model performed remarkably well, achieving an accuracy of 99.87%, the highest among all models. This model achieved perfect scores across both classes: precision of 1.00, recall of 1.00, and an F1-score of 1.00 for Class 0. For Class 1, the precision was 0.99, recall was 0.98, and the F1-score was 0.99. With only 2 false positives and 4 false negatives, the Meta Model significantly reduced classification errors and demonstrated superior performance in handling the minority class. The superior performance of the Meta Model can be attributed to its ability to combine the strengths of various base classifiers, which helps in improving the recall for the minority class without sacrificing overall accuracy. The Meta Model's near-perfect results indicate its robustness in handling even challenging classification problems with highly imbalanced datasets.

Table **4**: Models Performance Summary

Algorithm	Accuracy (%)	False Positive	False Negative	Class – 0			Class – 1		
				Precision	Recall	F1-Score	Precision	Recall	F1-Score
SVM	95.24	2	223	0.95	1	0.98	0.90	0.08	0.14
Neural Network	95.32	6	215	0.95	1	0.98	0.82	0.11	0.20
Gradient Boosting	95.37	7	212	0.95	1	0.98	0.81	0.12	0.22
Ada Boosting	95.37	4	215	0.95	1	0.98	0.87	0.11	0.20
Meta Model	**99.87**	**2**	**4**	**1**	**1**	**1**	**0.99**	**0.98**	**0.99**

4.2 Execution time

The execution times of the models reveal notable differences in computational efficiency. SVM has the highest execution time, indicating it takes the longest to run, making it the most computationally expensive. In contrast, AdaBoost and the Meta Model have similar execution times, both significantly faster than SVM. Gradient Boosting and Neural Network exhibit relatively lower execution times, with the Neural Network being the fastest. This analysis illustrated in Figure 3 highlights the computational cost associated with each model, showing that while SVM is the most resource-intensive, the Neural Network is the most efficient.

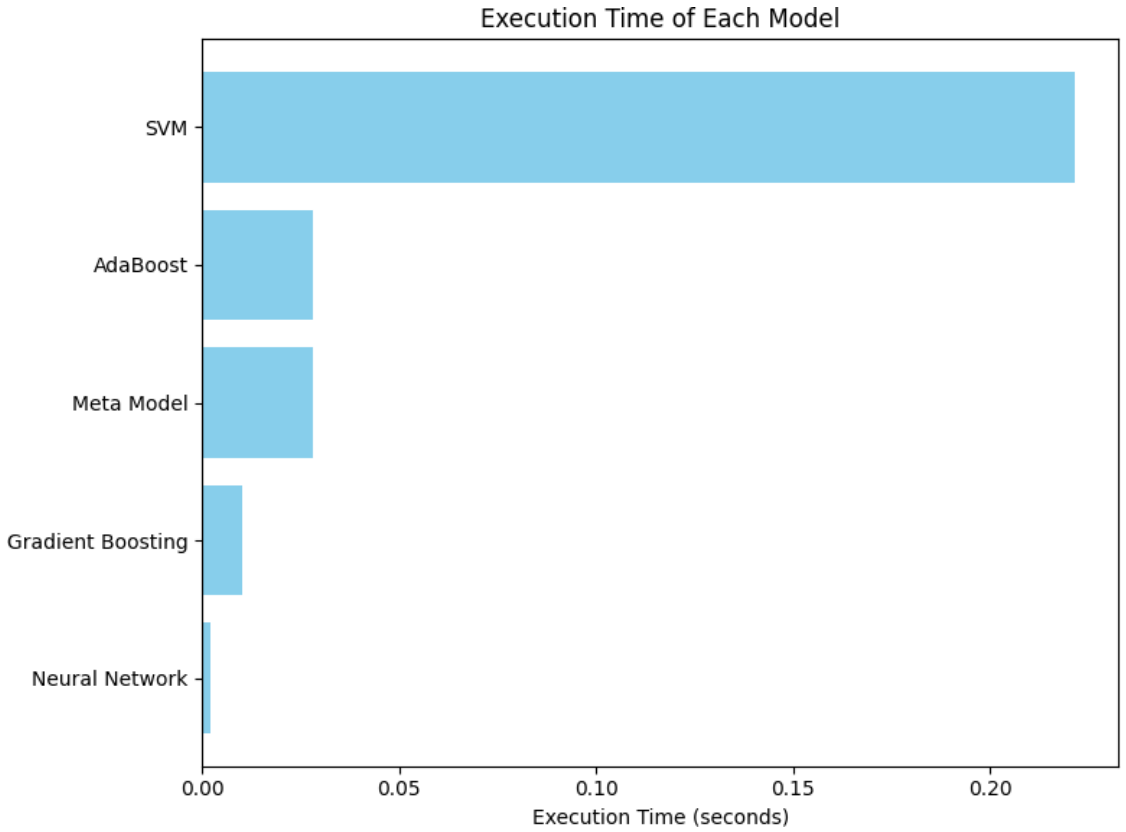

Figure **3**: Time taken by each model to make predictions

4.3 Model Validation

The relative contributions of each model within the Meta Model highlight the distinct roles in the final predictions. AdaBoost has the highest contribution, around 60%, indicating it plays the most significant role in the Meta Model's predictions. SVM follows with a contribution of about 30%, making it the second most important model. Gradient Boosting and Neural Network have smaller contributions, with Gradient Boosting at around 10% and the Neural Network being the least influential. As shown in Figure 4, this distribution reflects the weights assigned to each base model, where a higher percentage indicates a greater influence on the ensemble's final decision-making process.

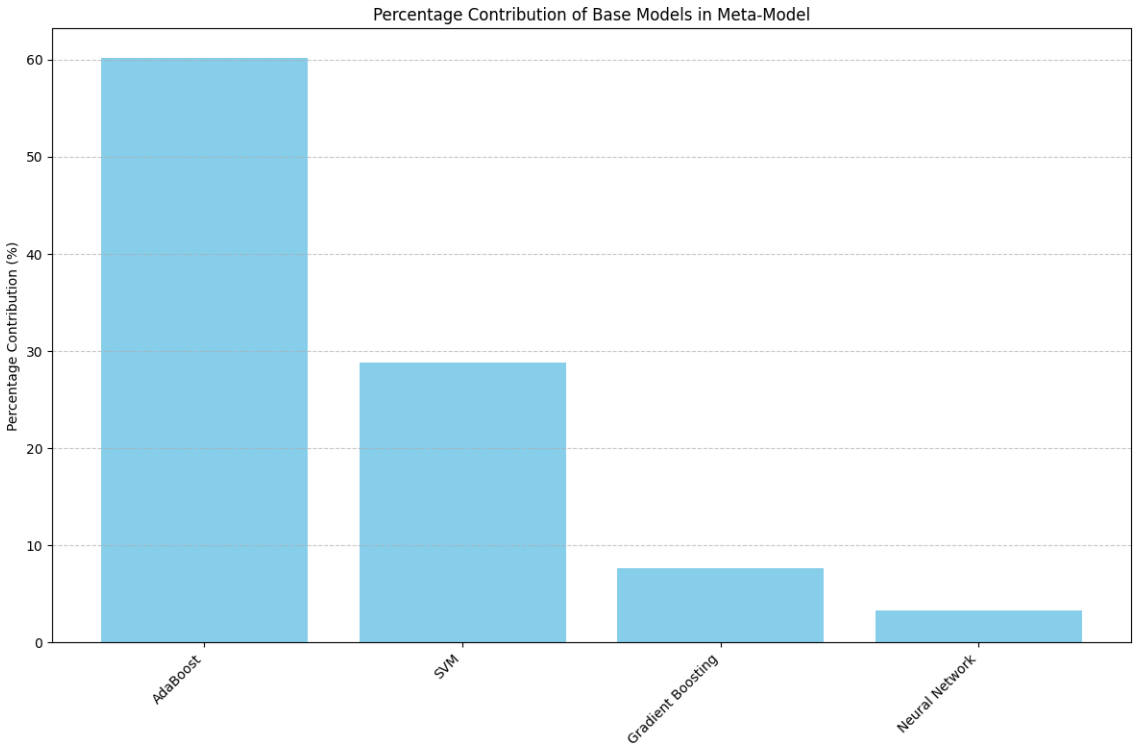

Figure **4**: Contributions of base models in meta model

The feature contributions of the top models within the meta mode reveal distinct patterns in how each model prioritizes input data. Figure 5 highlights the AdaBoost model, which assigns the highest importance to transactional behavior, such as transaction hour and amount, followed by account age. Address-related features, including billing and shipping addresses, also play a role, though with less impact on the model's predictions. Figure 6 focuses on the SVM model, where transaction amount emerges as the most influential feature. Account age and transaction hour are also significant contributors, with address-related features providing moderate influence. These visualizations offer insight into how the Meta Model leverages its base models, AdaBoost and SVM, to make accurate predictions by focusing on key transactional and account-specific data.

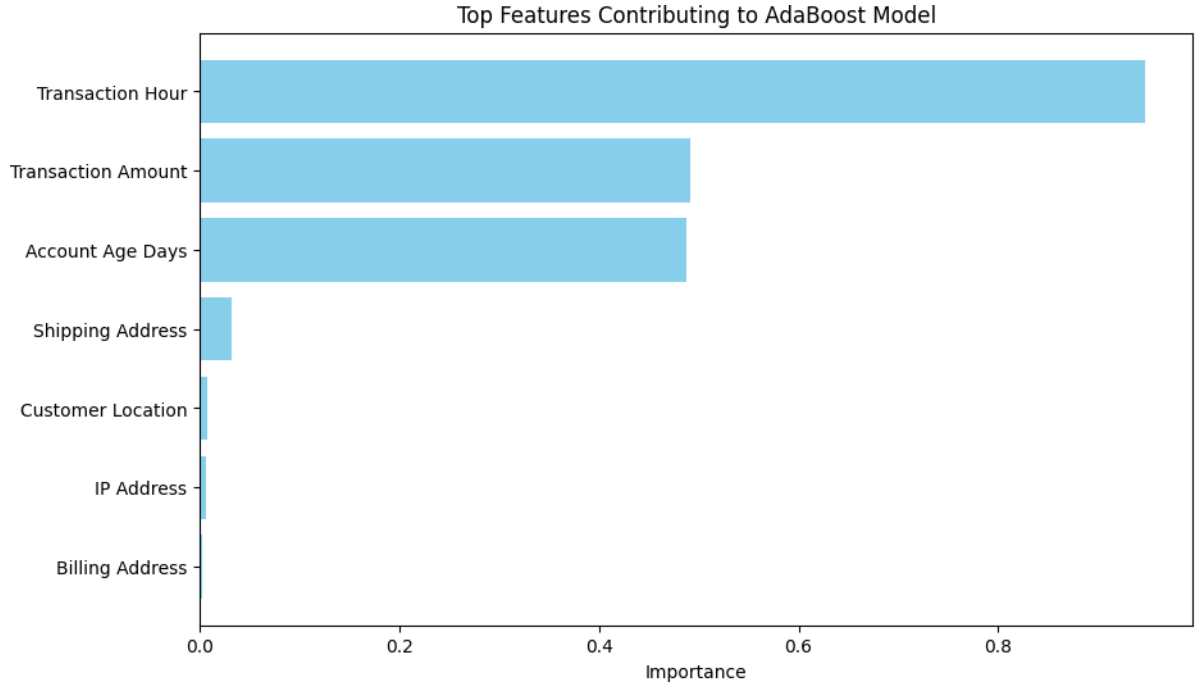

Figure **5**: Top contributing features of the AdaBoost Model

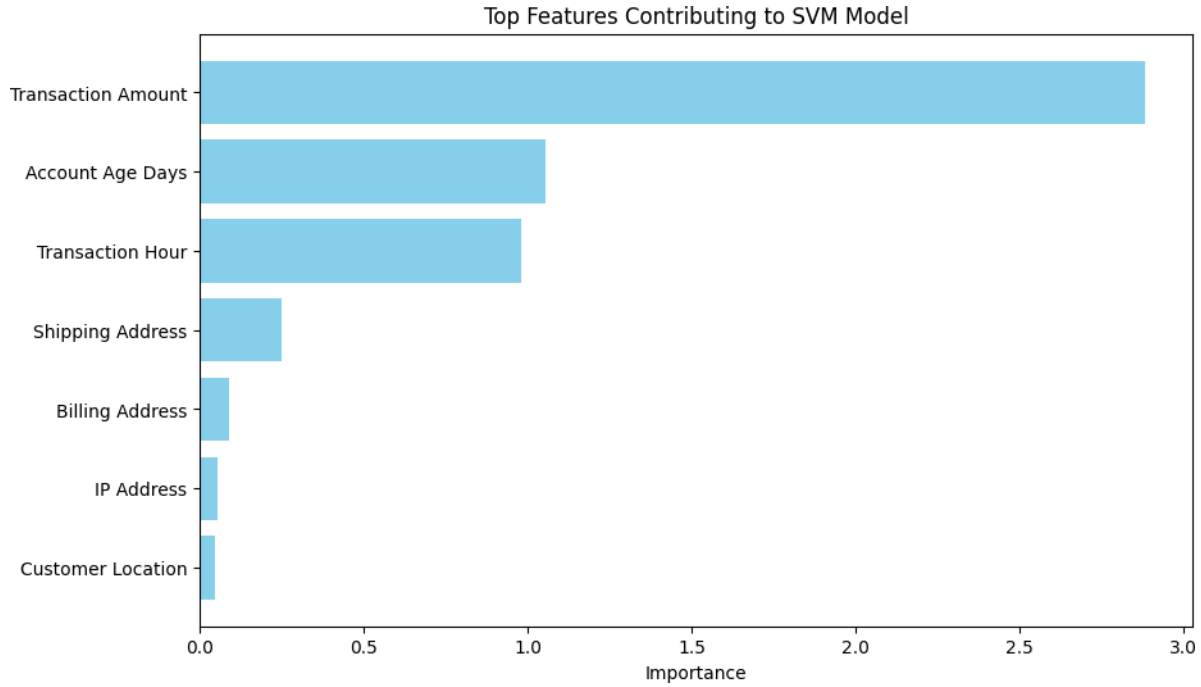

Figure **6**: Top contributing features of the SVM Model

4.4 Data Product

After incorporating the model into the data product, new transactions were created to test its effectiveness. The best-performing model, the meta-model, was integrated into a Django web application to evaluate its real-time performance. The web application featured fraud detection, alerts, transaction monitoring, and customer management, allowing users to track recent transactions and examine individual cases in detail. Transactions with large amounts, especially those made around midnight, were identified by the model as potential fraud cases. Figure 7 illustrates a new transaction recorded during the early hours of the day, highlighting the model's ability to detect unusual timing patterns. Similarly, Figure 8 presents a transaction with a high amount, further demonstrating the model's capacity to flag high-risk behaviors in real-time by leveraging key features like transaction amount and time.

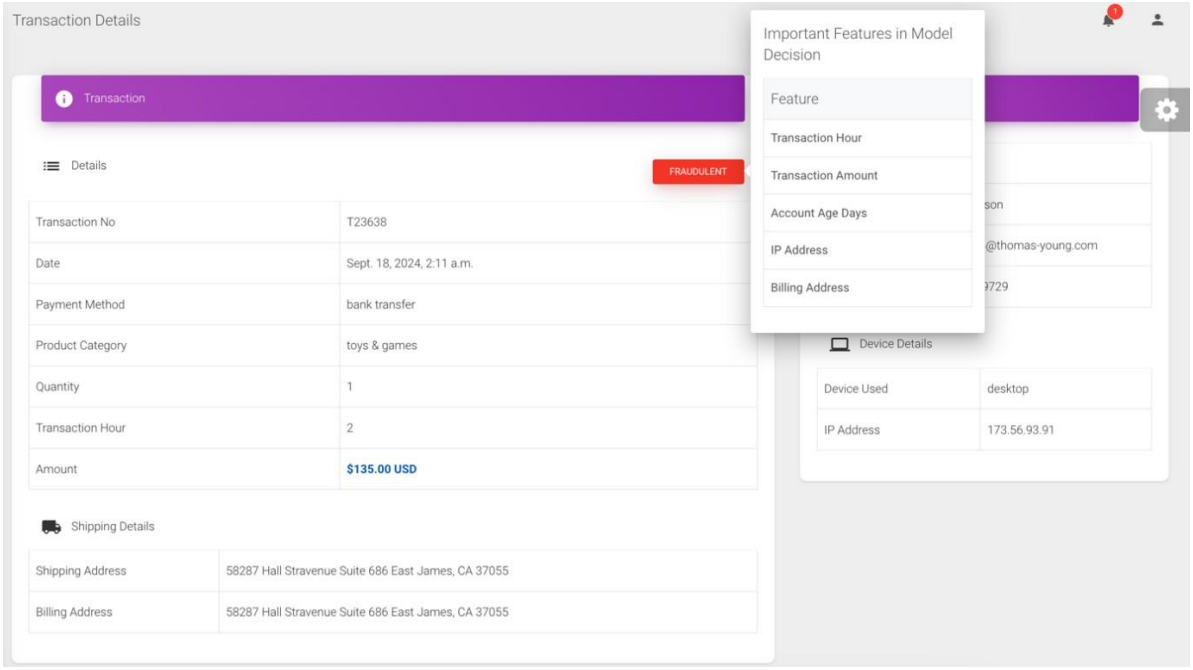

Figure 7: New Transaction Recorded During Early Hours Of the Day

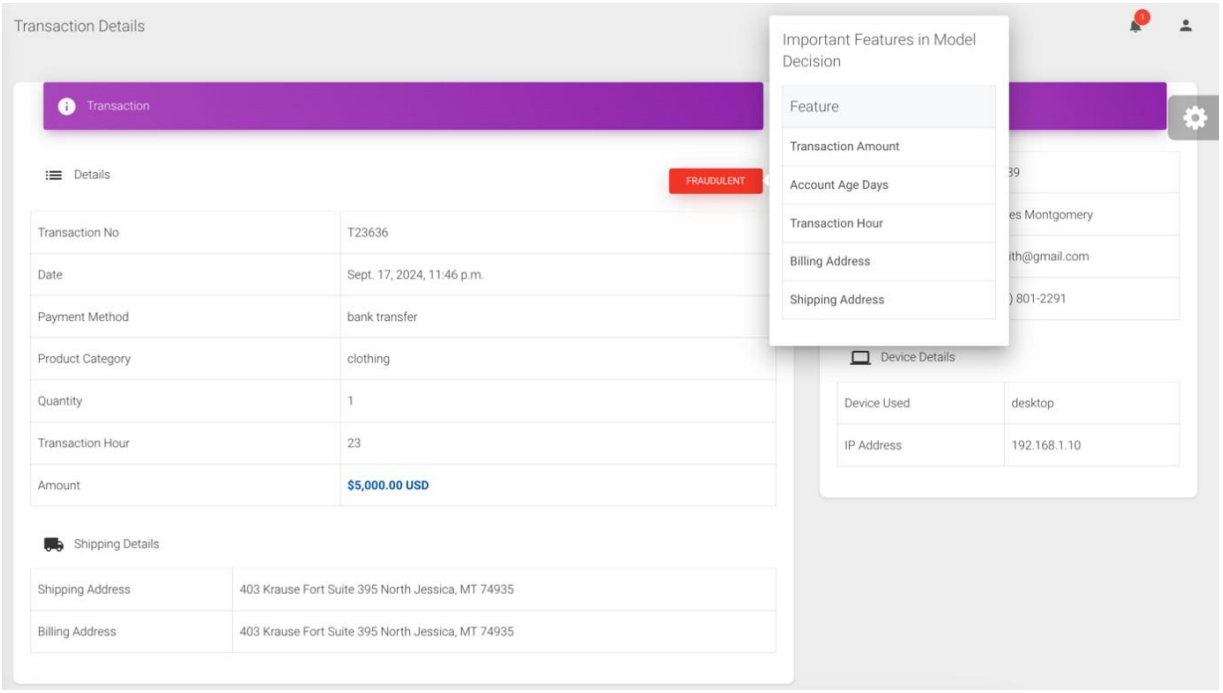

Figure 8: New transaction with a high transaction amount

This section evaluated four base models and a Meta Model, focusing on performance, class balance, and execution time. The Meta Model achieved the highest accuracy (99.87%) and effectively classified both majority and minority classes. While SVM had the longest execution time, the Neural Network was the most efficient. AdaBoost contributed the most to the Meta Model's predictions. When integrated into the data product, the model successfully flagged high-risk transactions, such as those with large amounts or made during unusual hours (Figures 7 and 8), demonstrating its effectiveness for real-time fraud detection.

5.0 CONCLUSION AND FUTURE WORKS

This study presents a significant advancement in e-commerce fraud detection by addressing class imbalance through robust model design rather than traditional resampling techniques. Our ensemble approach combines the strengths of Support Vector Machines, Neural Networks, AdaBoost, and Gradient Boosting, with a Random Forest meta-model that achieved an impressive accuracy of 99.87%, along with 0.99 precision, 0.98 recall, and 0.99 F1-score. By using carefully tuned hyperparameters and model-specific optimizations, we significantly improved minority class detection while minimizing false positives and negatives. The model's effectiveness was validated through both theoretical evaluation and practical implementation in a real-time data product environment, addressing a critical gap in existing literature.

To ensure transparency and trust in the model's decisions, Local Interpretable Model-Agnostic Explanations (LIME) was used, which identified Transaction Hour, Transaction Amount, and Account Age Days as the most significant predictors. This combination of high performance, interpretability, and practical validation demonstrates the model's readiness for real-world deployment. Future research could involve further validation using diverse, real-world datasets to enhance the model's generalizability.

REFERENCES

[1] Mastercard. "Ecommerce fraud trends and statistics merchants need to know in 2024." Mastercard. https://b2b.mastercard.com/news-and-insights/blog/ecommerce-fraud-trends-and-statistics-merchants-need-to-know-in-2024/

[2] E. Minastireanu and G. Mesnita, "Methods of Handling Unbalanced Datasets in Credit Card Fraud Detection," *BRAIN-BROAD RESEARCH IN ARTIFICIAL INTELLIGENCE AND NEUROSCIENCE*, vol. 11, pp. 131-143, 2020-03-01 2020, doi: 10.18662/brain/11.1/19.

[3] P. Gnip, L. Vokorokos, and P. Drotár, "Selective oversampling approach for strongly imbalanced data," *PEERJ COMPUTER SCIENCE*, 2021-06-18 2021, Art no. e604, doi: 10.7717/peerj-cs.604.

[4] O. Bello, A. Ogundipe, D. Mohammed, A. Folorunso, O. Alonge, and C. Bello, "AI-Driven Approaches for Real-Time Fraud Detection in US Financial Transactions: Challenges and Opportunities," pp. 84-102, 01 2023, doi: 10.37745/ejcsit.2013/vol11n684102.

[5] K. AbdulSattar and M. Hammad, "Fraudulent Transaction Detection in FinTech using Machine Learning Algorithms," 2020.

[6] N. Suardiman, Sudimanto, S. Dhanny, D. Tjahjadi, B. Permana, and K. Ukar, "E-Commerce Fraud Detection Using Generated Data From BANKSIM Using Machine Learning Approach: A Pilot Study | IEEE Conference Publication | IEEE Xplore," *2024 18th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, 2024, doi: 10.1109/IMCOM60618.2024.10418372.

[7] S. Xie, L. Liu, G. Sun, B. Pan, L. Lang, and P. Guo, "Enhanced E-commerce Fraud Prediction Based on a Convolutional Neural Network Model," *CMC-COMPUTERS MATERIALS & CONTINUA*, vol. 75, pp. 1107-1117, 2023-01-01 2023, doi: 10.32604/cmc.2023.034917.

[8] W. Yu, Y. Wang, L. Liu, Y. An, B. Yuan, and J. Panneerselvam, "A Multiperspective Fraud Detection Method for Multiparticipant E-Commerce Transactions," *IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS*, vol. 11, pp. 1564-1576, 2023-01-06 2024, doi: 10.1109/TCSS.2022.3232619.

[9] A. Saputra and Suharjito, "Fraud Detection using Machine Learning in e-Commerce," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 9, pp. 332-339, Sep 2019, doi: 10.14569/IJACSA.2019.0100943.

[10] F. Hasan, S. K. Mondal, M. R. Kabir, M. A. Al Mamun, N. S. Rahman, and M. S. Hossen, "E-commerce Merchant Fraud Detection using Machine Learning Approach," presented at the 2022 7th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2022.

[11] Y. Y. Tang, "Automatic Fraud Detection in e-Commerce Transactions using Deep Reinforcement Learning and Artificial Neural Networks," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 7, pp. 1047-1058, Jul 2023, doi: 10.14569/IJACSA.2023.01407113.

[12] F. K. Alarfaj, I. Malik, H. U. Khan, N. Almusallam, M. Ramzan, and M. Ahmed, "Credit Card Fraud Detection Using State-of-the-Art Machine Learning and Deep Learning Algorithms," *Ieee Access*, vol. 10, pp. 39700-39715, 2022, doi: 10.1109/access.2022.3166891.

[13] J. Y. Wang and C. Yang, "Financial Fraud Detection Based on Ensemble Machine Learning," in *IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, Falerna, ITALY, Sep 12-15 2022, 2022, pp. 1013-1018, doi: 10.1109/DASC/PiCom/CBDCom/Cy55231.2022.9928001. [Online]. Available: <Go to ISI>://WOS:000948109800160

[14] E. Ileberi, Y. X. Sun, and Z. H. Wang, "A machine learning based credit card fraud detection using the GA algorithm for feature selection," *Journal of Big Data*, vol. 9, no. 1, Feb 2022, Art no. 24, doi: 10.1186/s40537-022-00573-8.

[15] A. R. Khalid, N. Owoh, O. Uthmani, M. Ashawa, J. Osamor, and J. Adejoh, "Enhancing Credit Card Fraud Detection: An Ensemble Machine Learning Approach," *Big Data and Cognitive Computing*, vol. 8, no. 1, Jan 2024, Art no. 6, doi: 10.3390/bdcc8010006.

[16] M. Mim, N. Majadi, and P. Mazumder, "A soft voting ensemble learning approach for credit card fraud detection," *HELIYON*, vol. 10, 2024-02-15 2024, Art no. e25466, doi: 10.1016/j.heliyon.2024.e25466.

[17] 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, 2023-01-01 2023, doi: 10.1109/ACCESS.2023.3308298.

[18] K. Kerwin and N. Bastian, "Stacked generalizations in imbalanced fraud data sets using resampling methods," *JOURNAL OF DEFENSE MODELING AND SIMULATION-APPLICATIONS METHODOLOGY TECHNOLOGY-JDMS*, vol. 18, pp. 175-192, 2020-11-24 2021, Art no. 1548512920962219, doi: 10.1177/1548512920962219.

[19] Y. Chen and Z. Wu, "Financial Fraud Detection of Listed Companies in China: A Machine Learning Approach," *SUSTAINABILITY*, vol. 15, 2023-01-01 2023, Art no. 105, doi: 10.3390/su15010105.

[20] M. Lu *et al.*, "A Stacking Ensemble Model of Various Machine Learning Models for Daily Runoff Forecasting " *Water*, vol. 15, no. 7, 2023, doi: https://doi.org/10.3390/w15071265.

[21] R. Lazzarini, H. Tianfield, and V. Charissis, "A stacking ensemble of deep learning models for IoT intrusion detection," *KNOWLEDGE-BASED SYSTEMS*, vol. 279, 2023-09-13 2023, Art no. 110941, doi: 10.1016/j.knosys.2023.110941.

[22] M. Barton and B. Lennox, "Model stacking to improve prediction and variable importance robustness for soft sensor development," *Digital Chemical Engineering*, vol. 3, p. 100034, 2022, doi: https://doi.org/10.1016/j.dche.2022.100034.

[23] K. Lim, L. Lee, and Y. Sim, "A Review of Machine Learning Algorithms for Fraud Detection in Credit Card Transaction," *INTERNATIONAL JOURNAL OF COMPUTER SCIENCE AND NETWORK SECURITY*, vol. 21, pp. 31-40, 2021-09-30 2021, doi: 10.22937/IJCSNS.2021.21.9.4.

[24] G. Sasikala *et al.*, "An Innovative Sensing Machine Learning Technique to Detect Credit Card Frauds in Wireless Communications," *WIRELESS COMMUNICATIONS & MOBILE COMPUTING*, vol. 2022, 2022-06-23 2022, Art no. 2439205, doi: 10.1155/2022/2439205.

[25] I. Sadgali, N. Sael, and F. Benabbou, "Performance of machine learning techniques in the detection of financial frauds," in *SECOND INTERNATIONAL CONFERENCE ON INTELLIGENT COMPUTING IN DATA SCIENCES (ICDS2018)*, 2019-01-01 2019, vol. 148, pp. 45-54, doi: 10.1016/j.procs.2019.01.007.

[26] I. Mienye and Y. Sun, "A Deep Learning Ensemble With Data Resampling for Credit Card Fraud Detection," *IEEE ACCESS*, vol. 11, pp. 30628-30638, 2023-01-01 2023, doi: 10.1109/ACCESS.2023.3262020.

[27] Y. Moodi, M. Ghasemi, and S. Mousavi, "Estimating the compressive strength of rectangular fiber reinforced polymer-confined columns using multilayer perceptron, radial basis function, and support vector regression methods," *JOURNAL OF REINFORCED PLASTICS AND COMPOSITES*, vol. 41, pp. 130-146, 2021-10-19 2022, Art no. 07316844211050168, doi: 10.1177/07316844211050168.

[28] F. Anowar, S. Sadaoui, and IEEE, "Incremental Neural-Network Learning for Big Fraud Data," in *2020 IEEE INTERNATIONAL CONFERENCE ON SYSTEMS, MAN, AND CYBERNETICS (SMC)*, 2020-01-01 2020, pp. 3551-3557, doi: 10.1109/smc42975.2020.9283136.

[29] S. Jose, D. Devassy, and M. Antony, "DETECTION OF CREDIT CARD FRAUD USING RESAMPLING AND BOOSTING TECHNIQUE," in *2023 ADVANCED COMPUTING AND COMMUNICATION TECHNOLOGIES FOR HIGH PERFORMANCE APPLICATIONS, ACCTHPA*, 2023-01-01 2023, doi: 10.1109/ACCTHPA57160.2023.10083376.

[30] W. Fang, C. Chen, O. Song, L. Wang, J. Thou, and K. Thu, "Adapted Tree Boosting for Transfer Learning," in *2019 IEEE INTERNATIONAL CONFERENCE ON BIG DATA (BIG DATA)*, 2019-01-01 2019, pp. 741-750.

[31] G. Rushin, C. Stancil, M. Sun, S. Adams, P. Beling, and IEEE, "Horse Race Analysis in Credit Card Fraud-Deep Learning, Logistic Regression, and Gradient Boosted Tree," in *2017 SYSTEMS AND INFORMATION ENGINEERING DESIGN SYMPOSIUM (SIEDS)*, 2017-01-01 2017, pp. 117-121.

[32] L. Rukhsar, W. Bangyal, K. Nisar, and S. Nisar, "Prediction of Insurance Fraud Detection using Machine Learning Algorithms," *MEHRAN UNIVERSITY RESEARCH JOURNAL OF ENGINEERING AND TECHNOLOGY*, vol. 41, pp. 33-40, 2022-01-01 2022, doi: 10.22581/muet1982.2201.04.

[33] A. Petrovic, N. Bacanin, M. Zivkovic, M. Marjanovic, M. Antonijevic, and I. Strumberger, "The AdaBoost Approach Tuned by Firefly Metaheuristics for Fraud Detection | IEEE Conference Publication | IEEE Xplore," 2022, doi: 10.1109/AIC55036.2022.9848902.

[34] 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, 2021-01-01 2021, doi: 10.1109/ACCESS.2021.3134330.

[35] W. Ksiazek, M. Hammad, P. Plawiak, U. Acharya, and R. Tadeusiewicz, "Development of novel ensemble model using stacking learning and evolutionary computation techniques for automated hepatocellular carcinoma detection," *BIOCYBERNETICS AND BIOMEDICAL ENGINEERING*, vol. 40, pp. 1512-1524, 2020-10-01 2020, doi: 10.1016/j.bbe.2020.08.007.

[36] J. Brownlee. "Essence of stacking ensembles for machine learning." Machine Learning Mastery. https://machinelearningmastery.com/essence-of-stacking-ensembles-for-machine-learning/ (accessed Accessed September 15, 2024, 2024).

[37] N. S. Thomas and S. Kaliraj, "An Improved and Optimized Random Forest Based Approach to Predict the Software Faults," *SN Computer Science 2024 5:5*, vol. 5, no. 5, 2024, doi: 10.1007/s42979-024-02764-x.

[38] M. Zafar and N. Khan, "Deterministic Local Interpretable Model-Agnostic Explanations for Stable Explainability," *MACHINE LEARNING AND KNOWLEDGE EXTRACTION*, vol. 3, pp. 525-541, 2021-09-01 2021, doi: 10.3390/make3030027.