Enhancing E-commerce Security: A Comprehensive Approach to Real-Time Fraud Detection

Sara Alqethami [1†], Badriah Almutanni [1] and Walla Aleidarousr[2††],

[1] Department of Information Systems, College of Computer and Information Systems,
Umm Al-Qura University, Saudi Arabia
[2]Associate Professor, Department of Computer Engineering, College of Computer and Information Systems,
Umm Al-Qura University, Saudi Arabia

Abstract
In the era of big data, the growth of e-commerce transactions brings forth both opportunities and risks, including the threat of data theft and fraud. To address these challenges, an automated real-time fraud detection system leveraging machine learning was developed. Four algorithms (Decision Tree, Naïve Bayes, XGBoost, and Neural Network) underwent comparison using a dataset from a clothing website that encompassed both legitimate and fraudulent transactions. The dataset exhibited an imbalance, with 9.3% representing fraud and 90.07% legitimate transactions. Performance evaluation metrics, including Recall, Precision, F1 Score, and AUC ROC, were employed to assess the effectiveness of each algorithm. XGBoost emerged as the top-performing model, achieving an impressive accuracy score of 95.85%. The proposed system proves to be a robust defense mechanism against fraudulent activities in e-commerce, thereby enhancing security and instilling trust in online transactions.

Keywords:
Fraud detection, E-commerce, Real-time transactions XGBoost; Naïve Bayes, Neural Network

1. Introduction

The surge in e-commerce and electronic payments, amplified by the COVID-19 pandemic, has driven a notable increase in credit card usage for online transactions, particularly in recent years [1]. Despite economic challenges, online grocery shopping rose by over 79% in April 2020, indicating a general uptick in e-commerce. However, this growth exposes users to potential threats such as identity theft, fraudulent credit card activities, and money laundering on e-commerce platforms due to limited user background information and challenges in conducting credibility checks. Global fraud damages, projected to increase from 27.85 billion in 2018 to 40.63 billion in 10 years, highlight the urgency of developing efficient algorithms for real-time detection [2]. Academic and industrial efforts, leveraging big data, machine learning (ML), artificial neural networks (ANNs), deep learning (DL), and computational intelligence (CI) technologies, have produced various techniques differentiating between safe and fraudulent transactions [3][4][5]. However, concerns persist, notably the unbalanced distribution structure of datasets, leading to overfitting issues and poor classification efficiency [6]. In addressing these concerns, methods are proposed in this study to optimize the classification efficiency of state-of-the-art techniques, incorporating machine learning algorithms and novel feature engineering. Utilizing a Kaggle dataset from a clothing company, exploratory data analysis and extensive feature engineering were conducted, leading to the design and implementation of a more efficient machine learning technique for distinguishing fraud from legitimate transactions. The evaluation involved the original and non-flash transaction datasets to ensure reliable results.
The paper is organized as follows: Section 2 reviews recent research on e-commerce fraud detection systems. Section 3 presents exploratory data analysis and feature extraction techniques. Section 4 elaborates on the research methodology and techniques developed. Section 5 describes the experimental setup, including dataset details and evaluation measures. Section 6 illustrates the outcomes of experiments on the dataset and provides a detailed analysis of the results. Finally, Section 7 concludes the paper, discussing potential future research directions.

2. Literature Review

This section reviews contemporary challenges, prevalent approaches, and evaluation metrics, offering a comprehensive insight into recent advancements in addressing fraud detection issues. To begin, Suharjito et al. [7] introduced a method to address dataset imbalance in e-commerce fraud transactions by utilizing the Synthetic Minority Over Sampling Technique (SMOTE) and Principal Component

Manuscript received April 5, 2024
Manuscript revised April 20, 2024
https://doi.org/10.22937/IJCSNS.2024.24.4.1

Analysis (PCA) for preprocessing. The Neural Network model they employed exhibited superior accuracy at 96%, outperforming alternative models with accuracies of 95% for Random Forest, Naïve Bayes, and 91% for Decision Tree. These results highlight the effectiveness of advanced techniques in managing challenges associated with data imbalances. Shakya et al. [8] contributed a predictive analysis method for credit card fraud detection, integrating Random Forest, XGBoost, and Logistic Regression. Their hybrid resampling approach, particularly with Random Forest, outperformed alternative models, emphasizing the significance of ensemble methods in achieving robust fraud detection models. Liu et al. [9] proposed an XGBoost algorithm with SMOTE for fraudulent credit card transaction identification, addressing data imbalances in a dataset of 284,807 transactions. This approach demonstrated enhanced stability and performance, highlighting the importance of balancing techniques in preprocessing. Varmedja et al. [10] thoroughly compared various machine learning algorithms for fraud detection, where Random Forest demonstrated outstanding accuracy at 99.96%. Logistic Regression (97.46%) and Naive Bayes (99.23%) also performed well, setting a benchmark for the effectiveness of different algorithms. Xuan et al. [11] employed two Random Forest classifiers to differentiate between normal and abnormal transactions, achieving accuracy rates of 91.96% and 96.77%, respectively. This outcome emphasizes the versatility of Random Forest in effectively handling diverse transaction scenarios. Simi et al. [12] compared Random Forest, Support Vector Machine (SVM), and Artificial Neural Network (ANN) for credit card fraud detection, revealing Random Forest's accuracy falling between SVM and ANN. This nuanced comparison sheds light on the relative strengths of different machine learning approaches. Li et al. [13] enhanced SVM using a cuckoo search algorithm, demonstrating superior performance with 98% accuracy compared to SVM and other classification methods. These findings underscore the potential of metaheuristic algorithms in refining the accuracy of existing models. Trivedi et al. [14] introduced an effective feedback system for credit card fraud detection, where Random Forest exhibited a superior accuracy of 95.988% compared to other classifiers. These results emphasize the significance of feedback mechanisms in continuously improving the model. Xie et al. [15] presented a method for enhancing credit card fraud detection through the integration of feature engineering, demonstrating improved performance. This emphasizes the pivotal role that feature engineering plays in augmenting the discriminatory power of fraud detection models. Chen et al. [16] presented a graph-based system for detecting fraud in e-commerce insurance, showcasing the utility of graphs and learning methods in identifying fraudsters within a broader context. Lucas et al. [17] employed machine learning and data mining techniques for credit card fraud detection, utilizing temporal data and agglomerative clustering with a Random Forest classifier for efficient covariate shift detection. This illustrates the integration of diverse techniques for nuanced fraud detection. Nuci et al. [18][19] introduced an incremental learning approach for real-time fraud detection in online transactions, achieving an impressive accuracy of 97.2% with the Naive Bayes classifier. This contributes to the exploration of real-time fraud detection methodologies. Babu et al. [20] conducted an evaluation of Linear Regression, Decision Trees, and Random Forest for detecting credit card fraud. They emphasized the cost-effectiveness of Random Forest compared to XGBoost, albeit with a slight dip in performance. This underscores the importance of weighing both accuracy and cost-effectiveness when choosing suitable models. The collective findings from these studies significantly advance the ongoing enhancement of fraud detection methodologies across diverse domains. However, this study proposes a method aimed at improving the accuracy and efficiency of fraud detection while addressing the challenges and risks associated with it. With dedicated effort, the objective was to attain superior accuracy in identifying fraudulent transactions compared to other classification methodologies.

3. Dataset

The e-commerce fraud datasets were obtained from Kaggle, comprising 151,112 records. Of these records, 14,151 are classified as fraud, while 136,961 are classified as non-fraud transactions. The dataset entails transactions from an e-commerce clothing website, where transactions can either be safe or fraudulent. The ratio of fraudulent data is 9.3%, while legitimate transactions make up 90.07%.

3.1. Feature Engineering

The initial data encompasses details about new users and their transactions, organized into a 14-column table. Each denotes a field, including information like signup time, purchase time, user ID, age, sex, device ID, IP address, source, browser, country, and class. The raw dataset consolidates statistics related to consumer activities (such as “signup time,” “purchase time,” and “purchase value”), and personal attributes (such as “IP address,” “device ID,” “age,” “source,” “browser,” “sex,” and “country”). Among these personal attributes, two significant factors for feature recognition are briefly discussed: "time difference" and "device ID unique users, representing a user's distinct information that aids machine learning models in identifying fraudulent or legitimate transactions. However, relying solely on these features may not be adequate to identify unique patterns in the dataset. On the other hand, consumer activity-related features such as signup time, purchase time, etc. consist of continuous values that are entirely distinctive but may not effectively detect unusual transactions. To address this, additional features were derived from the provided information, enhancing the dataset's predictive capability for fraudulent transactions. A set of 10 extra features was extracted, surpassing the value of the previous ones. Table 1 presents a comprehensive list of all features derived from the original dataset.

Table 1. Feature engineering for fraud detection: enhanced dataset with extracted features

No	*Feature*	*Description*
1	time_difference	Time gap between signup and purchase times to detect fraud characteristics.
2	ip_users	Count of distinct visitors sharing the same IP address to detect potential fraud involving multiple users.
3	device_id_unique_ users	Count of distinct visitors using the same device ID for transactions to identify potential fraud.
4	day_of_the_week_ signup	Day of the week of signup or purchase to identify transaction patterns indicative of fraud.
5	week_of_the_year_ signup	Week of the year of signup or purchase to identify transaction patterns indicative of fraud.
6	total_purchase	Total purchases made using a device ID to detect fraud account for shared IDs among visitors.
7	average_purchase	Average purchase amount associated with a device ID to identify potential fraud involving shared IDs.
8	country_count	Count of occurrences for each country to identify areas with a higher likelihood of fraudulent operations.
9	day_of_the_week_ purchase_times	Day of the week of purchases to identify the number of users making transactions simultaneously.
10	week_of_the_year_ purchase_times	Week of the year of purchases to identify the number of users making transactions simultaneously.

Performing Exploratory Data Analysis (EDA) on the extracted features allows an assessment of their impact on predicting fraudulent and legitimate transactions. Before demonstrating the EDA analysis, two significant features need to be highlighted. Firstly, the statistical distribution of the "time difference" within the original data, as shown in Fig. 1 (left), clearly reveals a distinct division with a short "time difference" specifically for fraud cases. This suggests the need to split that portion of the data and recreate it "without flash transactions," as shown in Fig. 1 (right).

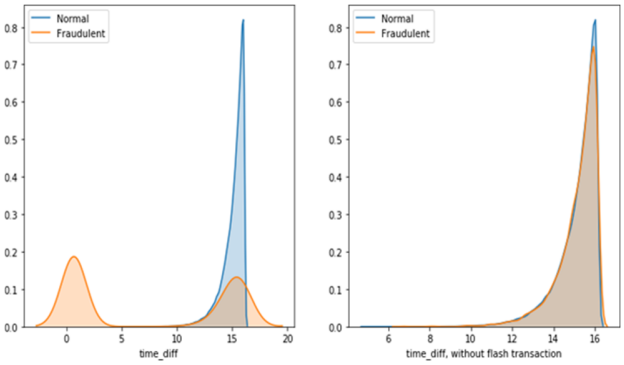

Fig. 1. Statistical distributions of “time difference” through two classes from original and without-flash-transaction datasets

Additionally, the statistical distribution of “device ID unique users” exhibits a distinct pattern unique to the fraud class. This pattern becomes pronounced after excluding observations related to the first and second transactions. Moreover, a strong linear relationship is observed between “IP users” and” device ID unique users” when the count of

"IP users" is less than 1. To avoid overfitting on these two features, they were removed, as illustrated in Fig. 2 (right).

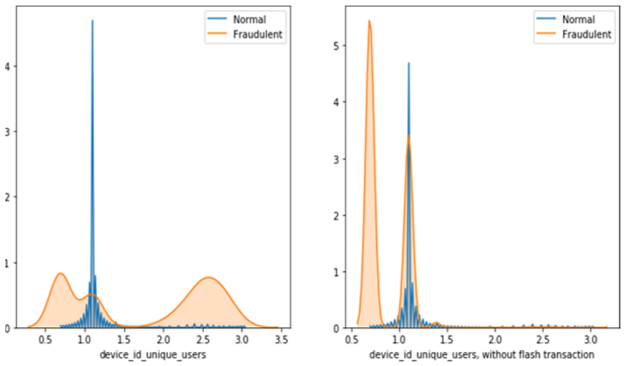

Fig.2. Statistical distributions of device id unique users through two classes from original and without-flash-transaction datasets

3.2. Processing Data

A crucial preprocessing step for machine learning is the transformation of string and object features into numeric formats. This conversion, facilitated through label encoding for categorical features, delineates the dataset into discernible input and output components. The pivotal "Class" attribute assumes the role of the output label, signifying 0 for fraudulent transactions and 1 for safe ones. The remaining attributes contribute indispensably to the input feature set. Visual representation emerges as a pivotal tool for discerning essential features, unraveling intricate relationships, and scrutinizing correlations. The emphasis is squarely on pruning highly correlated attributes to enhance model clarity and reduce variance. In the preparatory phase for modeling, a systematic removal of features with missing or duplicate values is undertaken. Subsequently, the dataset undergoes meticulous partitioning into training, testing, and validation subsets. The training set, comprising 93,027 instances, coexists with the testing set of 32,074 instances, while the validation set encompasses 13,746 instances. The initial dataset division adheres strictly to a 67:33 ratio, with the subsequent testing set division maintaining a meticulous 70:30 split between the test and validation subsets. Addressing the idiosyncrasies of flash transactions involves the creation of supplementary data through the application of natural logarithms to the "time difference" attribute. This nuanced approach results in the utilization of two distinct datasets for experimentation: one retaining the original "time difference" values and another incorporating natural logarithm values, specifically representing non-flash transactions. The dataset, characterized by its diverse data types, undergoes meticulous preparation to seamlessly align with machine learning requirements. Object-type categorical variables are deftly translated into numeric values using Scikit-Learn's encoding schemes. To ensure the fairest of comparisons, the data for non-flash transactions undergoes a parallel split, mirroring the division employed for the original dataset. This comprehensive and meticulous approach sets the stage for effective machine learning model evaluation and experimentation, laying a solid foundation for rigorous analysis and interpretation.

3.3. Data Visualizations

Several data have been analyzed and displayed, including device ID, purchase frequency, day of the week for purchase, day of the year for purchase, and country. But below are the most important basic charts and graphs that represent the main aspects of the data. Using distinct colors to distinguish between types of 2D objects, these visualizations provide valuable insights. Fig. 3 provides an illuminating perspective on the distribution of purchases across various devices, shedding light on user preferences. In Fig. 4, the analysis of purchase frequency becomes pivotal for distinguishing between legitimate and fraudulent transactions, revealing transaction patterns and potential anomalies.

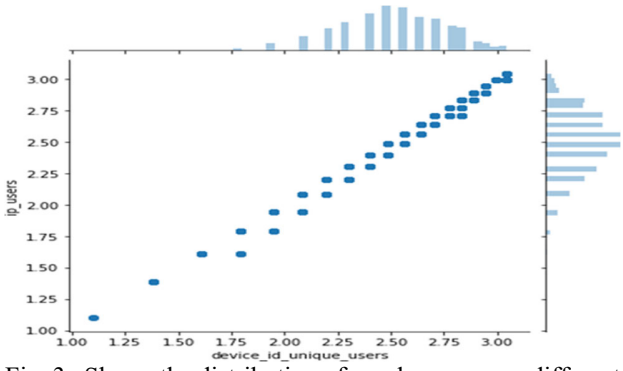

Fig. 3. Shows the distribution of purchases across different users and devices (a unique device id)

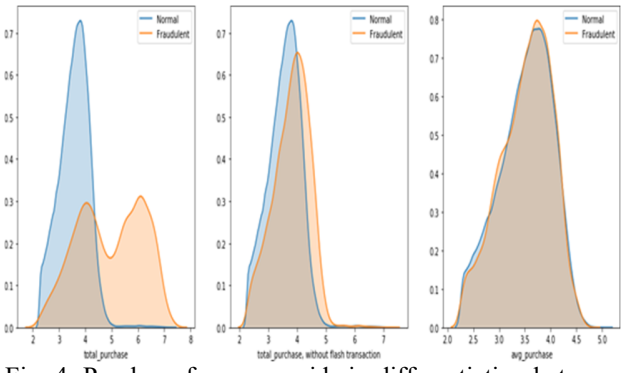

Fig. 4. Purchase frequency aids in differentiating between fraudulent and non-fraudulent transactions

4. Methodology

This paper focuses on constructing machine learning models tailored for the real-time detection of fraudulent transactions in e-commerce, employing a variety of ML algorithms. The research unfolds through distinct phases, illustrated in Fig. 5. Initiating with feature extraction via exploratory data analysis (EDA), subsequent steps involve sequential processes of data processing, modeling, and result evaluation. Key machine learning algorithms, such as XGBoost, Naïve Bayes, Decision Tree, and Neural Network, are implemented. Comparative analyses across outcomes from the e-commerce transaction dataset aim to identify the most effective model.

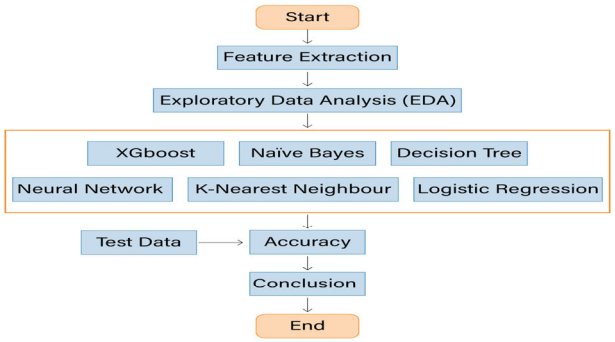

Fig. 5. Methodology steps for real-time fraudulent transaction detection in e-commerce

4.1. XGBoost

XGBoost [21], also known as "Extreme Gradient Boosting", executes Machine Learning algorithms under the GBoost framework. It provides a parallel tree boosting, enabling fast, specific solutions to a range of data science problems. Gradient boosting algorithms are primarily designed for supervised learning tasks and are well-suited for classification and regression problems. This method incorporates many weak learners to create a powerful predictive learning model. Multiple models can be trained sequentially, gradually, or cumulatively using gradient boosting algorithms. It begins by using the dataset for training a decision tree and assigning equal weights to all individual elements.

The equation below outlines the workings of the gradient boost. Discover F* (x) that maps the input values x to the output y in a way that minimizes the loss function given a training instance (x, y), where x is the input variable and y is the label output.

$$F^*(x) = \text{argmin}\ \text{argmin}_{f(x)}\ E_{x,y} \sum_{i=1}^{n} L(y, F(x)) \quad (1)$$

$$F^*(x) = \text{argmin}\ \text{argmin}_{f(x)}\ E_{x,y} \sum_{i=1}^{n} L(y, F(x)) \quad (1)$$

The algorithm performs ten times faster than typical solutions on a single computer, demonstrating scalability to millions of instances within distributed or memory-limited settings. The adaptability of XGBoost originates from numerous significant enhancements, including an inventive tree learning algorithm designed to handle sparse data and a technically justified weighted nonparametric patch method for approximate tree learning instance weight handling. The incorporation of parallel and distributed computing accelerates the learning process, enabling more rapid discovery of models. In our research, parameter settings for the XGBoost algorithm were tailored to enhance its performance. Key configurations involve, fixing the maximum depth of trees to "3", setting the learning rate to "0.1", configuring the number of estimators to "100", defining the objective function to "binary logistic", initializing the booster as "gbtree", setting regular alpha to "0," and regular lambda to "1". All other parameter values were set to default, as this generated the best results on the dataset.

4.2. Naive Bayes

The Naive Bayes algorithm (NB), rooted in Bayes' theorem, stands as a supervised learning technique commonly employed for classification tasks, particularly in scenarios featuring high-dimensional training dataset, such as text classification. As online payment trends advance, the surge in malicious activities calls for robust detection techniques in online transactions [22], where the Naive Bayes algorithm has demonstrated promising results. Operating on probability-based principles [5], NB excels at swiftly making predictions, leveraging experience to inform its decisions. The equation below elucidates the foundational working mechanism of the Naive Bayes algorithm.

$$P_r(X) = \frac{(P_r(c) * P_r(c))}{P_r X} \quad (2)$$

Where:
B: Unknown class data
A: Specific class
P (A|B): Posterior probability
P (A): Prior probability
P (B|A): Probability (conditions on the hypothesis).
P (B): Probability A

This method enables the identification of fraudulent or non-fraudulent transactions. In this study, the parameter settings selected for constructing the Naive Bayes algorithm involve setting the alpha value to "1.0," configuring a fit prior value

of “true,” and specifying the class prior as “none.” Extensive experimentation has revealed that these settings yield optimal results for the Multinomial Naive Bayes algorithm on the dataset.

4.3. Decision Tree

Decision Tree (DT) [23] is a supervised machine learning algorithm applied to tackle classification and regression challenges. DT utilizes a tree representation to solve problems, where each internal node represents the features of a dataset and has several branches based on decision rules, while each leaf node represents the outcome of those decisions. DT is a binary classification algorithm that has been used to identify transactions as either fraudulent or non-fraudulent [24]. This enables the detection of abnormal user behavior. The fundamental terminology associated with decision trees is as follows:

(a) Root Node: This signifies the entire dataset, divided into two or more subsets.
(b) Splitting: This procedure involves dividing a root node into sub-nodes.
(c) Decision Node: When a sub-node further divides into multiple sub-nodes.
(d) Leaf/ Terminal Node: This represents the outcome of a decision node and does not split further.
(e) Pruning: This is the process of removing unused sub-nodes from a tree.
(f) Branch/ Subtree: A segment of all trees is referred to as a branch or sub-tree.
(g) Parent/ Child Node: A root node is called the parent, and the sub-nodes it divides into are called child nodes.

The architecture of DT is shown in Fig. 6. It contains the root node, child node, and leaf / terminal node of the decision tree. Gini criteria were employed to split the Decision Tree at each level, as defined by the formula:

$$Gini = 1 - \sum_{i=1}^{n} (p_i)^2 \qquad (3)$$

This equation computes the Gini for each branch on a node by evaluating the class and its probability, helping identify the more probable branch. In this context, “pi” denotes the relative frequency of the examined class, while “c” represents the number of classes. In the current experiments, the DT algorithm was implemented with specific parameter settings, using the "gini" splitting criteria and setting the maximum tree depth to "20." Further configuration included requiring a minimum of "2" samples for node splitting, setting the minimum samples per leaf as “1”, and specifying ccp alpha as “0.”

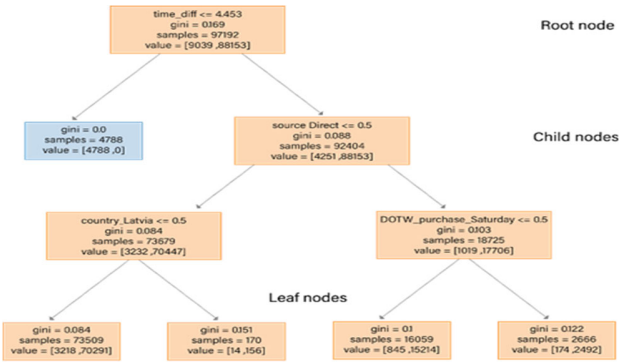

Fig. 6. Architecture of decision tree for fraud detection in e-commerce using a tree representation and decision rules

4.4. Neural Network

Artificial Neural Networks [25] are designed to mimic the neural structure of the human brain. They have emerged as versatile solutions with broad applicability across various domains. ANNs can be utilized to discover and predict new features, especially when dealing with large datasets in decision support and optimization models. Neural networks have shown promising results in tackling e-commerce fraud detection problems. In the ANN model, the number of input units equals the number of input features, which is 127. Augmenting the number of iterations decreased the training loss, but it proved to be time-consuming. Additionally, extending the size of the hidden layers improved the results to some extent, as shown in Fig. 7. After practicing on a particular dataset, an ANN can learn a predictive function.

$$f(.): R^M \rightarrow R^0 \qquad (4)$$

Where m is the input dimension and m is the output dimension.

Given an input sequence, X = x1, x2, ... xm

A multi-layer perceptron, with an input x and a label output y, can learn a nonlinear function to make predictions for both regression and classification problems. It can capture non-linear relationships between input and label output variables, in addition to linear relationships between functions. In the experiments, the following parameter settings were used to achieve sophisticated results:

(a) "Hidden units": 8
(b) "Number of hidden layers": 3
(c) "Activation function": ReLU
(d) "Solver": Adam
(e) "Maximum iteration": 500

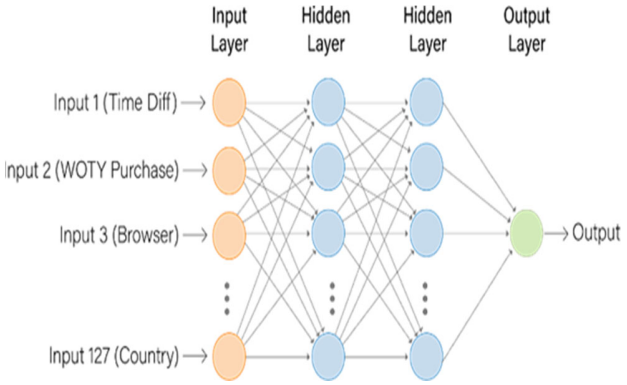

Fig. 7. Architecture of a neural network for fraud detection in e-commerce using multiple layers of artificial neurons

5. Experimental Setup

The experimental setup for fraud detection in e-commerce transactions involved two dataset settings: the original dataset and the non-flash-transaction datasets (derived from the original dataset by logarithmically transforming the time difference). The study employed specific experimental settings and evaluation methods, which will be briefly discussed.

5.1. Dataset

In these experiments, an e-commerce transaction dataset was used, covering two main categories of attributes: (1) user online transaction activity and (2) user bio information. To prepare the data suitable for predicting transactions as fraudulent or safe, additional features were derived from the original dataset. Section 3.1 offers detailed explanations of all the newly generated features. Following data cleaning, preprocessing, and feature derivation, the dataset comprises a total of 138,847 instances.

5.2. Evaluation Methodology

In the field of e-commerce fraud detection, real-time fraud detection is a task of supervised binary classification, where the goal is to distinguish between fraud (labeled as 1) and safe/non-fraud transactions (labeled as 0). Utilizing Python and Scikit-Learn libraries, an efficient online transaction fraud detection system was designed. The system was trained on a dataset encompassing training, validation, and test subsets. The training dataset facilitated model training, the validation dataset helped prevent overfitting, and the test dataset was crucial for assessing the model's performance on previously unseen data. All experiments were conducted on a Jupyter notebook with an i3 processor and 8GB of RAM.

5.3. Evaluation Measures

To evaluate and compare the models in this research, five metrics were employed, starting with accuracy, one of the most common metrics for evaluating performance. Considering the imbalanced nature of the class dataset, four additional evaluation measures were incorporated: Recall, Precision, F1 Score, and AUC ROC. The formulas for these metrics are defined as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (5)$$

$$TPR = \frac{TP}{TP + FN} \quad (6)$$

$$FPR = \frac{FP}{FP + TN} \quad (7)$$

$$Recall = \frac{TP}{TP + FN} \quad (8)$$

$$Precision = \frac{TP}{TP + FP} \quad (9)$$

$$F1\ score = \frac{2\ *\ Precision*Recall}{Precision + Recall} \quad (10)$$

Where TP, FN, TN, and FP given in the equations 5 to 10 represent the true positive prediction, and false negative prediction, true negative prediction, false positive prediction, respectively.

6. Results and Discussions

Multiple machines learning models, including the XGBoost classifier, Multinomial Naive Bayes, Decision Tree, and Neural Network, were applied to the dataset to evaluate their performance. A detailed analysis was conducted to determine which model aligned best with the dataset characteristics. The development and evaluation of these models on both the original dataset and the non-flash transaction dataset yielded nearly identical results. To analyze and compare the results of multiple machine learning models, state-of-the-art evaluation metrics were utilized. After performing feature engineering, normalization, and reducing features such as device ID, user ID, age, and IP address, which had low correlation with the output and no importance in real-time fraud detection, the dataset was left with the 127 most important features and 97,192 instances. The results on the original dataset are presented in Table 2 after parameter optimization. Table 3

depicts the results for XGBoost, Naive Bayes, Decision Tree, and Neural Network for the non-transaction dataset.

Table 2. Results of XGBOOST, NB, DT, AND NN on original dataset

Model	XGBoost	NB	DT	NN
Accuracy	95.85%	95.29%	94.78%	94.57%
Precision	96.01%	95.09%	94.38%	94.12%
Recall	95.85%	95.29%	94.78%	94.56%
F1	95.31%	94.79%	94.33%	94.13%
AUC	79.06%	77.27%	76.71%	78.17%

Table 3. Results of XGBOOST, NB, DT, AND NN ON for non-transaction dataset

Model	XGBoost	NB	DT	NN
Accuracy	95.84%	94.79%	95.29 %	95.16 %
Precision	96.01%	94.40 %	95.09 %	94.91 %
Recall	95.84%	94.79%	95.29 %	95.16 %
F1	95.31%	94.34%	94.78 %	94.68%
AUC	79.06%	76.63 %	77.26%	78.30%

XGBoost model stands out among others, achieving an impressive accuracy of 95.85%. Its success can be attributed to its flexibility, adaptability to different scenarios, and significant speed advantages. It is ten times faster than existing solutions when running on a single machine and scales effectively to handle millions of instances in distributed or memory-limited environments. On the other hand, the Decision Tree model, after thorough parameter tuning, achieved an accuracy of 94.77%. The model's parameters were carefully selected, including a maximum depth of 20, criterion set to 'gini', and minimum samples split, and leaf set to 2 and 1, respectively. The Multinomial Naive Bayes model achieved an accuracy of 95.29%, performing slightly below XGBoost. The Neural Network model achieved 94%, showing similar performance to the Decision Tree model. The close results among the models indicate their ability to capture important patterns and achieve accurate predictions. The specific algorithms and techniques used by each model contribute to slight variations in their performance, but overall, they demonstrate similar capabilities in handling the dataset. The highest accuracy of 95.85% was achieved with XGBoost. These outcomes are comparable to the highest results attained in various past studies on online transaction fraud detection systems.

Lakshmi et al. [26] achieved an accuracy of 91% using XGBoost on a highly imbalanced dataset of European bank transactions. Charleonnan et al. [27] achieved 70% accuracy using Naive Bayes with RUS sampling on a dataset from a Taiwanese bank. Mudasiru et al. [28] achieved 81% accuracy using a Decision Tree algorithm on a credit card transaction dataset. D. Cheng et al. [29] achieved 88% accuracy using a Neural Network algorithm on a dataset from a major commercial bank. Comparing the produced results to previous research, as shown in Table 4, the models for real-time fraud detection in e-commerce outperform the results from previous studies. XGBoost achieved the highest accuracy of 95.8% on the e-commerce dataset with extensive feature engineering, and it achieved an accuracy of 95.84% on the non-flash transaction dataset.

Table 4. Comparing produced results with previous works

Models	Reference	Accuracy	**Presented Accuracy**
XGBoots	Lakshmi et al. [26]	91%	**95.8%**
NB	Charleonnan [27]	70%	**95%**
DT	Mudasiru et al. [28]	81%	**94%**
NN	D. Cheng et al. [29]	88%	**94%**

7. Conclusion and Future Work

In the current era of big data, the reliance of users on online platforms has surged, leading to an increased interest in data intelligence and protection. The rapid growth of e-commerce platforms and digital payment transactions poses ongoing challenges for industry and academic researchers. The rapid growth of e-commerce platforms and digital payment transactions poses ongoing challenges for industry and academic researchers. This study aims to enhance the accuracy and efficiency of fraud detection in response to these challenges and risks. It prioritizes achieving superior accuracy in identifying fraudulent transactions compared to other classification methodologies. The intelligent fraud detection system developed for e-commerce employs various machine-learning techniques, including XGBoost, Naive Bayes, Decision Tree, and Neural Network. The research utilizes an open-source e-commerce transaction dataset for experimentation, emphasizing a real-world scenario with a significantly imbalanced dataset. Results are evaluated based on key metrics such as Accuracy, Precision, Recall, and AUC, with XGBoost demonstrating superior performance at 95.85%. The chosen dataset is large, freely accessible, and optimized to eliminate irrelevant data that could impact model performance. The outcomes highlight the effectiveness of XGBoost, particularly with feature engineering and parameter optimization, for fraud detection in e-commerce.

Future research directions include the development of a stacked ensemble of multiple machine learning models to

build a robust fraud detection system. Evaluation methods will be diversified, and data imbalances will be addressed through assimilation techniques. The focus will also extend to generating real-time fraud predictions using alternative aggregation strategies.

Reference

[1] L. Gelles, "How the Virus Transformed the Way Americans Spend Their Money," The New York Times, 2020.

[2] HSN Consultants, "The nilson report," Inc.Carpinteria, USA, 2019.

[3] S. Garg and R. Sharma, "Fraud Detection with Machine Learning and Artificial Intelligence," in Handbook of Artificial Intelligence Applications for Industrial Sustainability, CRC Press, 2024, pp. 157-166.

[4] A. R. Khalid et al., "Enhancing credit card fraud detection: An ensemble machine learning approach," Big Data and Cognitive Computing, vol. 8, no. 1, 2024, Article 6.

[5] A. Husejinovic, "Credit card fraud detection using naive Bayesian and C4.5 decision tree classifiers," International Journal of Advanced Computer Science and Applications, vol. 10, no. 9, 2019.

[6] P. Gupta et al., "Unbalanced Credit Card Fraud Detection Data: A Machine Learning-Oriented Comparative Study of Balancing Techniques," Procedia Computer Science, vol. 218, 2023, pp. 2575-2584.

[7] A. Saputra and S. Suharjito, "Fraud detection using machine learning in e-commerce," International Journal of Advanced Computer Science and Applications, vol. 10, no. 9, 2019.

[8] R. Shakya, "Application of machine learning techniques in credit card fraud detection," 2018.

[9] C. Meng, L. Zhou, and B. Liu, "A case study in credit fraud detection with SMOTE and XGBoost," Journal of Physics: Conference Series, vol. 1601, 2020, Article 052016.

[10] D. Varmedja et al., "Credit card fraud detection - machine learning methods," 2019, pp. 1-5.

[11] S. Xuan et al., "Random Forest for credit card fraud detection," 2018, pp. 1-6.

[12] M. J. Simi, "Credit Card Fraud Detection: A Comparison using Random Forest, SVM and ANN," International Research Journal of Engineering and Technology, vol. 225, 2019.

[13] M. Zareapoor and P. Shamsolmoali, "Application of credit card fraud detection: Based on bagging ensemble classifier," Procedia Computer Science, vol. 48, 2015, pp. 679-685.

[14] N. Trivedi et al., "An efficient credit card fraud detection model based on machine learning methods," MATTER: International Journal of Science and Technology, 2020.

[15] Y. Xie et al., "A feature extraction method for credit card fraud detection," 2019, pp. 70-75.

[16] C. Chen et al., "Infdetect: a large-scale graph-based fraud detection system for e-commerce insurance," CoRR, 2020, abs/2003.02833.

[17] Y. Lucas et al., "Dataset shift quantification for credit card fraud detection," CoRR, 2019, abs/1906.06977.

[18] A. Mehana and K. P. Nuci, "Fraud detection using data-driven approach," CoRR, 2020, abs/2009.06365.

[19] W. Li et al., "Regboost: a gradient boosted multivariate regression algorithm," International Journal in Computer Simulation, vol. 4, no. 1, 2020, pp. 60-72.

[20] M. G. et al. Babu, "A Machine Learning Approach for Credit Card Fraud Detection," 2020, pp. 5237-5244.

[21] B. Alshawi, "Utilizing GANs for Credit Card Fraud Detection: A Comparison of Supervised Learning Algorithms," Eng. Technol. Appl. Sci. Res., vol. 13, no. 6, 2023, pp. 12264-12270.

[22] P. Kumari and S. P. Mishra, "Analysis of credit card fraud detection using fusion classifiers," 2019, pp. 111-122.

[23] A. S. Alraddadi, "A Survey and a Credit Card Fraud Detection and Prevention Model using the Decision Tree Algorithm," Engineering, Technology & Applied Science Research, vol. 13, no. 4, 2023, pp. 11505-11510, https://doi.org/10.48084/etasr.6128.

[24] R. Jain, B. Gour, and S. Dubey, "A hybrid approach for credit card fraud detection using rough set and decision tree technique," International Journal of Computer Applications, vol. 139, no. 10, pp. 1-6, 2016.

[25] A. RB and S. K. KR, "Credit card fraud detection using artificial neural network," Global Transitions Proceedings, vol. 2, no. 1, pp. 35-41, 2021.

[26] Lakshmi and S. D. Kavilla, "Machine learning for credit card fraud detection system," International Journal of Applied Engineering Research, vol. 13, no. 24, pp. 16819-16824, 2018.

[27] A. Charleonnan, "Credit card fraud detection using RUS and MRN algorithms," MIT–73, 2016.

[28] Mudasiru and Hammed & Jumoke Soyemi, "An implementation of decision tree algorithm augmented with regression analysis for fraud detection in credit card," International Journal of Computer Science and Information Security, vol. 18, pp. 79-88, 2020.

[29] D. Cheng et al., "Spatio-temporal attention-based neural network for credit card fraud detection," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 362-369, 2020.

Sara Alqethami received her B.E. and M.E. degrees from Umm Al-Qura University in 2018 and 2020, respectively. Her studies have resulted in the publication of several scientific research papers in the field of data and artificial intelligence. She has been working as a researcher and AI Engineer with Rabigh Municipality, where she focuses on developing an artificial intelligence-based road maintenance management system. Her research interests include machine learning, deep learning, and computer vision. She is a member of the Saudi Council of Engineers.

Badriah Almutanni received the B.E. and M.E. degrees from Umm Al-Qura University in 2014 and 2022, respectively. She worked as an information technology specialist from 2015 to 2023, eventually serving as a director during that time. In 2024, she transitioned to the role of project manager at SDAIA, specifically for the Alam project. Her research interests include machine learning, computer vision, and natural language processing. She is a member of the IEEE.

Walla Al-Eidarous received her B.S. and M.S. degrees from Umm Al-Qura University (K.S.A.) and Loyola University Chicago (U.S.A.). She completed her PhD at the University of Sussex (U.K.) in 2019. She is an Assistant Professor in the Computer Networks and Engineering Department at Umm Al-Qura University. Her research interests include the Internet of Things (IoT), data visualization, localization, Mobile Ad-Hoc Networks (MANET) in extreme settings, and Delay Tolerant Networks.