ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)
https://doi.org/10.17559/TV-20241013002057
Original scientific paper

Navigating the Fraud Frontier: Machine Learning Solutions for Credit Card Security

Hye Jin KIM, Jung Soo RHEE*

Abstract: Credit cards are essential financial tools that provide convenience and flexibility to both individuals and companies, allowing safe transactions and enabling worldwide trade. However, as the number of transactions increases, the probability of fraud also rises. This presents substantial risks to financial security. It undermines customer confidence. The rapid increase in credit card transactions highlights the urgent need for advanced fraud detection systems. This study examined the efficacy of six prominent machine learning algorithms: Logistic Regression (LR), Random Forest (RF), K-Nearest Neighbours (KNN), Decision Tree (DT), Support Vector Machine (SVM), and Naive Bayes (NB) in detecting fake credit card transactions using a dataset of 568,630 transactions with 30 features collected from European cardholders in the year 2023. The data was gathered from multiple financial institutions and payment processors throughout Europe. The dataset features V1 to V28, which are anonymized transaction details like time and location. The Amount column shows the transaction value, while the Class column indicates if a transaction is fraudulent (1) or not (0). To address the inherent disparity between fake and real transactions, the research utilises thorough preprocessing approaches such as feature scaling and class balancing to ensure the model's strong performance. Thorough data analysis and visual representation were used to find significant transaction patterns and connections among various characteristics. Each algorithm is subjected to thorough training and optimization via the modification of hyperparameters and cross-validation. The evaluation of performance is carried out by analysing metrics such as accuracy, precision, recall, F1 score, and Receiver Operating Characteristic-Area under Curve (ROC-AUC). Results demonstrated that the RF model achieved the highest accuracy of 99.98% and a perfect AUC of 1.000. LR also performed well with an accuracy of 96.56% and an AUC of 0.9935. These results provide vital insights into the practical uses of these algorithms in increasing fraud detection systems and improving financial security, particularly as credit card usage continues to grow.

Keywords: class balancing; classification matrices; financial tools; fraud detection; machine learning

1 INTRODUCTION

The digitalization of the global economy has transformed money management, with credit cards playing a crucial role. They offer convenience, security, and flexibility, leading to widespread acceptance in daily life. Credit cards streamline the purchasing process, allowing quick transactions without cash and being accepted by numerous retailers worldwide. This ease of use extends to online shopping, where they are the preferred payment method due to straightforward processing and verification. Built-in security features, such as fraud detection and zero-liability standards, protect customers from fraudulent activities, which is vital in today's digital landscape. Additionally, many credit cards offer benefits like travel insurance and extended warranties, enhancing their value for cardholders.

Statistics from 2021 reveal high credit card acceptance rates. According to the World Bank, countries like Canada, Israel, and Iceland have over 74% of their populations using credit cards. This reflects economic stability and advanced financial institutions. In contrast, Afghanistan has a credit card ownership rate of just 0%, attributed to economic insecurity and poor banking infrastructure. Canada, leading with 82.74% ownership, illustrates a strong integration of credit cards into daily financial practices, supported by a robust financial system and a culture that values credit [1].

The widespread use of credit cards raises concerns about fraud, leading to financial losses for customers and organizations. This diminishes public confidence in electronic payments and may deter usage. Effective security measures are vital to maintain trust.

Credit card fraud involves unauthorized use by someone other than the cardholder for financial gain, often through stolen card details [2-4]. Fig. 1 depicts various ways of credit card fraud.

Mekterović et al. (2021) reported fraud detection in card-not-present transactions and highlighted challenges such as feature engineering and unbalanced datasets [5]. They evaluated various data mining models and compared them to existing systems. Real-world data from an international card-processing company is used to identify cost-effective improvements for fraud detection systems. Mienye and Sun (2023) presented a hybrid feature-selection method to improve credit card fraud detection [6]. It combined filter and wrapper approaches, using information gain (IG) to rank features and a genetic algorithm (GA) wrapper with extreme learning machine (ELM) for classification. The GA wrapper optimised for unbalanced data by focussing on the geometric mean (G-mean) rather than accuracy. This methodology attained sensitivity (0.997) and specificity (0.994). The study investigated credit card fraud detection utilising a variety of machine learning techniques, including SVM, KNN, and artificial neural networks. It compared supervised and deep learning algorithms for differentiating between fraudulent and non-fraudulent transactions [7].

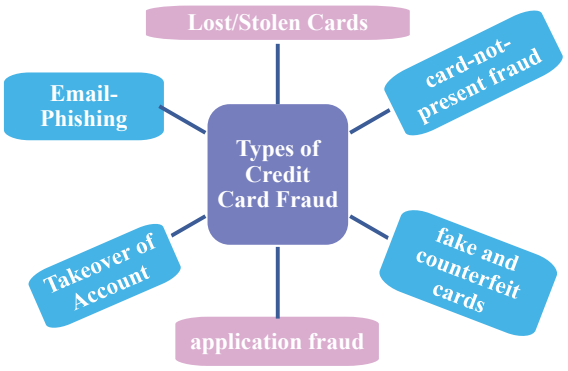

Figure 1 Different techniques of committing credit card fraud

Banks primarily use rule-based systems to detect credit card fraud by analyzing anomalies in transaction data [8-10]. The Rock Hyrax Swarm Optimization Feature Selection (RHSOFS) method improves fraud detection by

selecting key features from high-dimensional datasets. This study compared RHSOFS with other techniques, including Differential Evolutionary Feature Selection (DEFS) and Genetic Algorithm Feature Selection (GAFS) [11]. Results indicated that RHSOFS outperformed these methods, with statistical tests confirming its effectiveness. Feng and Kim (2024) examined the increase in fraud involving credit cards as a result of increasing use, projecting worldwide losses of more than USD 400 billion [12]. Their paper proposed a unique machine learning technique for fraud detection that employed compact data learning (CDL). Their approach surpassed existing feature reduction strategies because it reduces dataset size and complexity while maintaining accuracy. This study makes major theoretical and practical advances in improving fraud detection systems in the banking industry.

Various machine learning (ML) techniques have been used to detect credit card fraud, employing conditional statements to enhance sensitivity [13-17]. A rule-based approach was developed that identified fraud without resampling, achieving high accuracy and precision (0.99) [18]. Another study compared several algorithms for imbalanced datasets using a Kaggle dataset of 284,315 real transactions and 492 fraudulent ones. This research found an F1 score of 82.5% and a precision-recall area under the curve (PR AUC) of 81% when applying the neighborhood cleaning rule for undersampling [19]. Credit card transactions can be classified as real or fake using both single and hybrid ML methods [20-24]. Yan et al. presented feature analysis and preprocessing techniques to enhance fraud detection accuracy and scalability. By using logistic regression, random forest, and XGBoost with SMOTE, their adaptive model optimization improved fraud prevention and consumer trust in electronic payments [25]. Mienye and Swart introduced a hybrid framework that combines Generative Adversarial Networks (GANs) with Recurrent Neural Networks (RNNs) for credit card fraud detection [26]. The GAN generates synthetic fraudulent transactions to address data imbalance. Their GAN-GRU model achieves a sensitivity of 0.992 and specificity of 1.000 on the European credit card dataset, demonstrating its effectiveness in enhancing fraud detection. Alashwali et al. surveyed 150 U.S. debit/credit card fraud victims, revealing that psychological impacts outweighed financial losses, with no link between loss amount and psychological effects [27]. Most victims detected fraud by reviewing statements rather than through bank notifications. The study provides recommendations to improve fraud detection and reporting processes. The key contributions of this work are outlined below

- This study employs six prominent machine learning algorithms: LR, RF, KNN, DT, SVM, and NB. The research examines different algorithms and evaluates their strengths and shortcomings in identifying credit card fraud.

Hyperparameter tuning is conducted for each model using grid search to optimize their performance. This includes adjusting parameters such as the solver type and regularization strength for Logistic Regression, the number of estimators and maximum depth for Random Forest, and the number of neighbors and distance metric for K-Nearest Neighbors. For DT, key parameters include the maximum depth of the tree, the minimum number of samples required to split a node, the minimum number of samples required at a leaf node, and the criterion used for splitting (e.g., Gini impurity or entropy). For SVM, important parameters include the kernel type (e.g., linear, polynomial, radial basis function), the regularization parameter (C), and the kernel coefficient (gamma) for non-linear kernels. For NB, tuning involves selecting the type of NB algorithm (e.g., Gaussian, Multinomial, Bernoulli) and smoothing parameters (e.g., alpha for Laplace smoothing in Multinomial Naive Bayes). Adjusting these parameters can significantly impact the performance of each algorithm in detecting credit card fraud.

2 MATERIALS AND METHODS

2.1 Dataset

The dataset used in this study is sourced from Kaggle. It includes credit card transactions made by European cardholders in 2023 [28]. The data was collected from various financial institutions and payment processors across Europe. This ensures a diverse range of transactions reflecting different spending behaviors.

The dataset contains 568,630 rows and 30 columns. Each row represents a single transaction, while each column corresponds to a characteristic or label associated with that transaction. Key features include:

- V1 to V28: Anonymized attributes representing various transaction details, such as time, location, and transaction type.
- Amount: The transaction amount.
- Class: The target variable indicates whether a transaction is fraudulent (1) or not (0).

This dataset supports the development of fraud detection algorithms. It is ideal for training and evaluating machine learning models.

The class Balancing technique adjusts the distribution of classes in a dataset. It addresses issues when one class is much smaller than another. Such as, in fraud detection, there are often many true transactions compared to fake ones. Class balancing helps ensure machine learning models are trained effectively without bias toward the majority class. The feature scaling process transforms feature values into a common scale. It does not distort the differences in value ranges. Common methods include normalization, which scales values between 0 and 1, and standardization, which sets the mean to 0 and the standard deviation to 1. Feature scaling is crucial for algorithms like K-Nearest Neighbors and Support Vector Machines, which are sensitive to input data scale.

2.2 Descriptive Statistics

The statistics indicate that the feature columns (V1 to V28) have means that are about zero and standard deviations that are around 1, suggesting that they are most likely standardized. The Amount column displays an average of 12,041.96 with a significant standard deviation of 6,919.64, indicating a substantial range of transaction amounts. The Class column, which pertains to binary classification, has an average value of 0.5. This indicates that the class labels are evenly distributed, with an equal amount of 0 s and 1s. The data demonstrates a broad spectrum of values, with the feature columns covering a

large range of values and the Amount column ranging from 50.01 to 24,039.93. The percentiles for the Amount column indicate a distribution that is skewed to the right, with the majority of values concentrated at the lower end and a few bigger outliers. In general, the data seems to be well-prepared for analysis, since it contains standardized attributes and a balanced target variable.

2.3 Data Visualization

The dataset is visually represented using a number of histograms, each of which depicts the distribution of values inside distinct columns (Fig. 2).

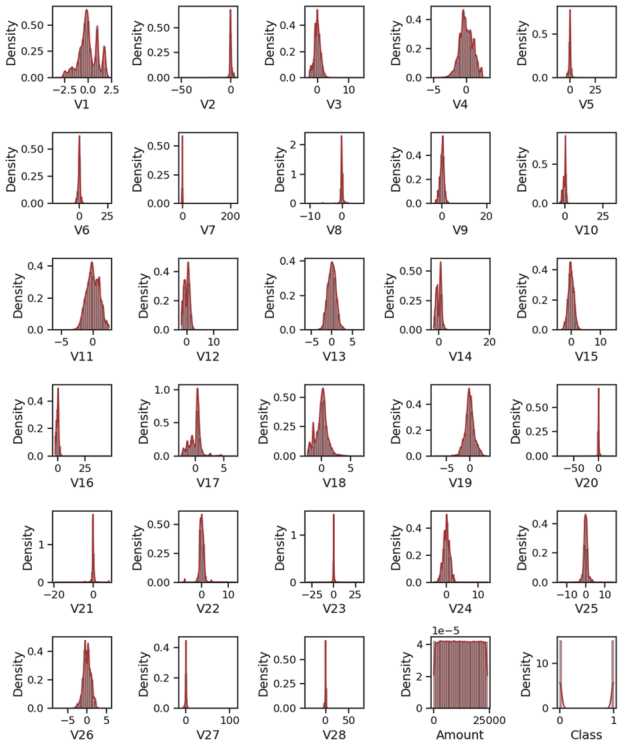

Figure 2 Histogram plot for distribution of values inside distinct columns

To account for the vast range of data values and frequencies, the *y*-axis of each histogram is scaled logarithmically.

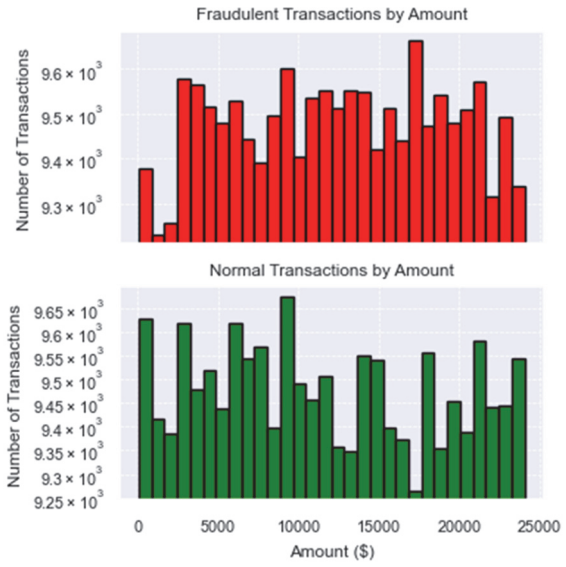

Figure 3 Transaction amounts for fraudulent and normal transactions

This method improves the visibility of both frequent and rare data points by highlighting patterns that would be concealed on a linear scale. The histogram in Fig. 3 shows transaction amounts for fraudulent (red) and normal (green) transactions. Both histograms use a logarithmic y-axis.

A heatmap was used to visualize correlations between features in the dataset (Fig. 4), with each cell annotated by the Pearson correlation coefficient. Notable findings include a positive correlation between features V1 and V10 (0.60), and a negative correlation between V2 and V3 (−0.69). The target variable, Class, shows a strong positive correlation with V4 (0.74) and a negative correlation with V1 (−0.51). The Amount feature exhibits weak correlations with most other features. These correlations are useful for feature selection and identifying potential multicollinearity issues in modeling.

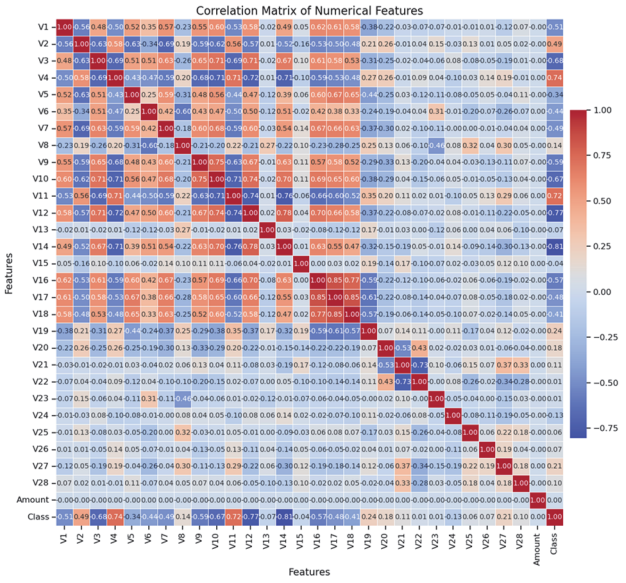

Figure 4 Correlation map

Principal Component Analysis (PCA)

Further, a line plot is presented in Fig. 5 to visualize the first 28 principal components of a dataset, which have been reduced using Principal Component Analysis (PCA). In this analysis, Principal Component Analysis (PCA) was applied to reduce the dimensionality of the credit card transaction dataset. The target variable (Class) and the Amount feature were removed, as they were not needed for PCA. The remaining numerical features (V1 to V28) were standardized using StandardScaler, which was essential because PCA is sensitive to the scale of the data. Without standardization, features with larger values could dominate the analysis.

After standardizing the data, PCA was performed, retaining all 28 components, corresponding to the 28 numerical features. PCA generates new variables called principal components (PCs), which are combinations of the original features that capture the variance in the data. The first few components explain most of the variance, while later components contribute less. To enhance clarity and manage the volume of data, the dataset is down sampled by selecting every 10-th row. This reduces the number of data points plotted and helps to avoid overcrowding.

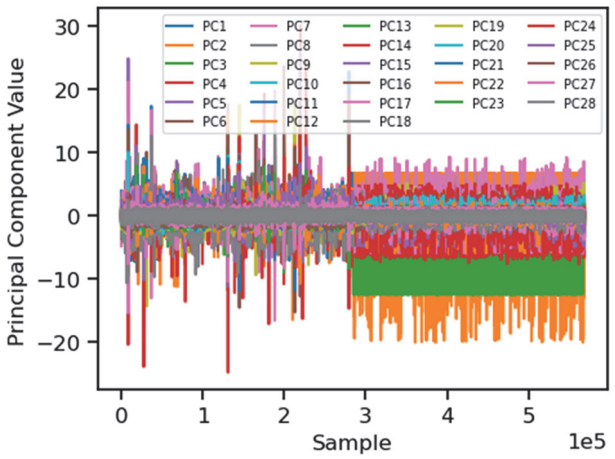

Figure 5 Line plot to visualize the first 28 principal components

2.4 Data Preparation

Feature and Target Variables: The 'id' column was dropped, and the 'Class' column was separated as the target variable y. The remaining columns were used as features X.

Train-Test Split: To evaluate model performance, the dataset was divided into two ratios of training and test sets. This was accomplished using 80-20 and 70-30 split ratios.

Data Normalization: The features were normalized using Standard Scaler to ensure that they had zero mean and unit variance, which is required by many machine learning techniques. Given a feature vector X with i samples, where each sample has a mean μ and standard deviation σ.

$$x_i^{\text{scaled}} = \frac{x_i - \mu}{\sigma}$$

The prepared dataset was then used for model training and evaluation.

2.5 Model Development

Model Evaluation and Optimization: Each model's performance was assessed using several key metrics, including precision, recall, accuracy, and F1 score. A confusion matrix was generated to visualize the counts of true positives, true negatives, false positives, and false negatives.

For optimization, hyperparameter tuning was performed using Grid Search and Random Search techniques. This involved systematically testing a range of hyperparameter values to identify the optimal combination for each model. Additionally, cross-validation was implemented to ensure the model's performance was robust and not overly fitted to the training data.

In this work, six ML algorithms were selected based on their effectiveness in binary classification and complementary strengths. Logistic Regression (LR) is chosen for its simplicity and interpretability, allowing understanding of predictor influence on outcomes. Random Forest (RF) is included for its ensemble method, which improves accuracy and reduces overfitting. K-Nearest Neighbors (KNN) is selected for its ability to capture local patterns in the data without strong assumptions. Decision Trees (DT) are useful for identifying key features and complex relationships due to their straightforward nature. Support Vector Machines (SVM) handle high-dimensional data well, especially when classes overlap. Lastly, Naive Bayes (NB) is effective and efficient for high-dimensional datasets, particularly in fraud detection.

Logistic Regression (LR): LR is a statistical approach that helps in understanding the influence of numerous predictors on the dependent variable under consideration. Using many predictor variables, LR attempts to estimate the probability of a binary event, such as fraud vs non-fraud. In this instance, the dependent variable (Y) reflects the binary result, which is commonly recorded as 0 or 1. Predictor variables (X) are independent variables used in predicting the result, and they may be categorical, continuous, or a mix of the two. Each predictor variable is seen as a possible contribution to the probability of the result. The mathematical expression is

$$\text{probability } P(Y=1 \mid X) = \frac{1}{1+\mathrm{e}^{\left(\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots+\beta_n X_n\right)}}$$

where β_0 is the intercept term. $\beta_1, \beta_2, \beta_3, \ldots, \beta_n$ are the coefficients for the predictor variables $x_1, x_2, x_3, \ldots, x_n$.

Random Forest (RF): RF is an ensemble learning approach that constructs many decision trees during training and returns the mode (classification) or mean prediction (regression) of the individual trees. Each tree is built using a random subset of the training data and characteristics, and the final prediction is combined from all of the trees to increase accuracy and prevent overfitting. The prediction for a sample x can be expressed as

$$\hat{y} = model\left(\left\{T_1(x), T_2(x), \ldots, T_K(x)\right\}\right)$$

where $T_i(x)$ $T_i(x)$ is the predicted from the i-th decision tree, and K is the total number of trees in the forest.

K-Nearest Neighbours (KNN): KNN is a non-parametric approach for classification and regression. The algorithm classifies a data point based on the majority class of its k closest neighbors in the feature space. The distance metric, commonly Euclidean distance, is used to determine the closest neighbors. For two points $x = (x_1, x_2, x_3, \ldots, x_n)$ and $x' = (x_1', x_2', \ldots, x_n')$, The Euclidean distance is given by:

$$d(x, x') = \sqrt{\sum_{i=1}^{n} (x_i - x_i')^2}$$

The prediction $\hat{y}$ for a sample x is

$$\hat{y} = \text{model}(\{y_i \mid x_i \text{ is among the } k \text{ nearest neighbours of } x\})$$

Decision Tree (DT): A decision tree is a model that divides data into subsets based on feature values, resulting in a tree structure with each internal node representing a

feature-based decision, each branch representing the decision's conclusion, and each leaf node representing a class label or regression value. Gini Impurity measures the purity of node.

$$G(t) = 1 - \sum_{i=1}^{K} p_i^2$$

Information Gain (IG) measures the reduction in entropy from a split:

$$IG(T, A) = H(T) - \sum_{v \in Values(A)} \frac{|T_v|}{|T|} H(T_v)$$

The entropy of a dataset (T) is represented by (T), whereas $H(T_v)$ represents the entropy of the subset (T) resulting from the split on attribute A.

Support Vector Machine (SVM): Support Vector Machines (SVM) are based on statistical learning theory and are used for classification and regression problems. The fundamental purpose of SVM is to create a hyperplane ($w_T x + b = 0$) that best separates data points of distinct classes. The margin $\frac{2}{w}$ represents the distance between the hyperplane and the closest data point in each class. The SVM optimization problem can be stated as follows:

$$\text{Minimize} \frac{1}{2} w^2 ;$$

Subject to: The inequality $y_i (w \cdot x_i + b) \geq 1$ holds for every i.

Naive Bayes (NB): The Naive Bayes Classifier assumes that the existence of a certain feature is independent of the presence of other characteristics inside the class label. The "naive" assumption simplifies the calculation of probabilities using Bayes' Theorem. The theorem outlines updating a hypothesis's probability estimate based on unseen data. The mathematical expression is as follows:

$$P(Q|R) = \frac{P(R|Q).P(Q)}{P(R)}$$

where $P(Q|R)$ is the posterior probability of Q given R. $(R|Q)$ is the probability, expressing the probability of R given Q, $P(Q)$ and $P(R)$ are the prior probabilities of Q, and R.

The Procedure used to evaluate fraud or non-fraud credit card transactions is displayed in Fig. 6.

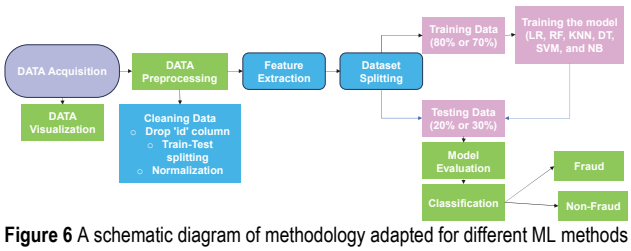

Figure 6 A schematic diagram of methodology adapted for different ML methods

2.6 Evaluation Metrics

When assessing a model's performance, several key metrics are used. Precision indicates the proportion of positive predictions that are correctly identified.

$$Precision, P = \frac{True\ positive(TP)}{True\ positive(TP) + False\ Positive(FP)}$$

Recall measures the proportion of actual positives correctly identified by the model.

$$Recall = \frac{True\ positive}{True\ Positive + False\ Negative}$$

Accuracy is the overall proportion of correct predictions (both true positives and true negatives).

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

A matrix showing the counts of true positives, true negatives, false positives, and false negatives is called a confusion matrix.

Table 1 Confusion matrix

	PREDICTIVE POSITIVE	PREDICTIVE NEGATIVE
ACTUAL POSITIVE	TP	FN
ACTUAL NEGATIVE	FP	TN

ROC Curve is a graphical representation of the model's performance across various threshold settings. It plots the True Positive Rate (Sensitivity) against the False Positive Rate (1 - Specificity). AUC (Area under the Curve) represents the overall ability of the model to discriminate between the positive and negative classes. AUC values range from 0 to 1, with 1 indicating perfect discrimination.

3 SIMULATION AND DISCUSSION

This section presents the optimization results for six machine learning algorithms used for fraud detection. The models were tested with two train-test ratios: 80:20 and 70:30. Key performance metrics included accuracy, precision, recall, and F1 score. The results highlight the strengths and weaknesses of each algorithm in identifying fraudulent transactions. The subsequent section discusses the optimization results for the six machine learning methods presented in Section 2.5. The experiments were conducted using two train-test ratios: 80:20 and 70:30. The analysis covers the performance results on the test dataset.

Figs. 7 and 8 display the acquired performance metrics (Precision, Recall, and F1 Score) for the LR, RF, KNN, DT, SVM, and NB models at both train-test ratios. Fig. 9a, b shows overall accuracy for all models. At the 80:20 train-test ratio, the RF model demonstrated the highest accuracy of 99.98%, indicating its exceptional ability to accurately identify both fraud and non-fraud transactions. This model achieved near-perfect precision, recall, and F1 scores, confirming its robustness in handling imbalanced datasets typical in fraud detection scenarios. LR and DT also

showed impressive results, with precision of 96.53% and 99.46%, respectively, reflecting their strong performance in fraud classification. In comparison, NB had the lowest precision (91.54%), indicating that it struggled more with recognising fraudulent transactions in this dataset. KNN and SVM demonstrated reasonable performance, with precision of 94.94% and 96.00%, respectively. These findings suggest that, although RF and DT excel at identifying fraud, NB may be less successful at capturing the complexity of fraud detection.

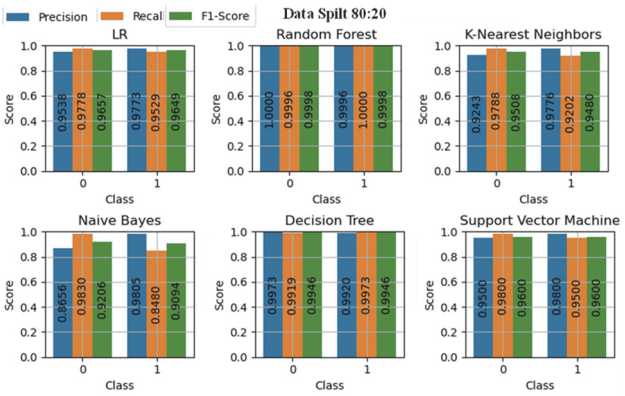

Figure 7 Evaluated classification metrics at 80:20 data spilt ratio

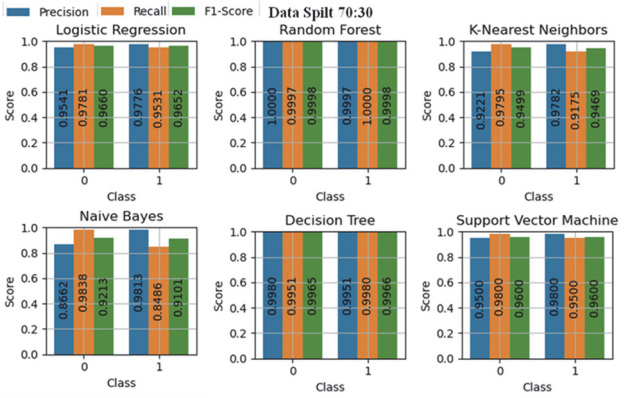

Figure 8 Evaluated classification metrics at 70:30 data spilt ratio

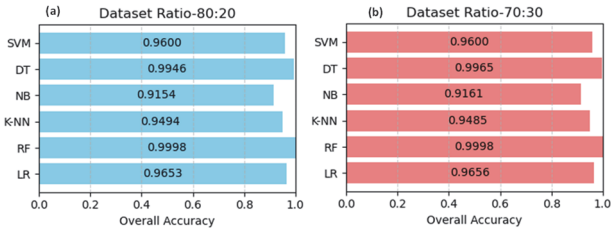

Figure 9 Accuracy (a) 80:20 (b) 70:30 ratios

When using a 70:30 train-test ratio, the RF algorithm demonstrated exceptional performance, with a precision of 99.98%. This outcome was similar to the results obtained using an 80:20 split. The LR model showed a slight rise in precision to 96.56%, whilst the DT model maintained a high precision of 99.65%. These results confirm the dependability of the models across various dataset settings.

The K-Nearest Neighbours and SVM algorithms exhibited consistent performance, with precision of 94.85% and 96.00% respectively. The NB model showed a slight improvement, reaching a precision of 91.61%, although it remained lower than other models. The accuracy range among the models at the 70:30 ratio ranged from 91.61% (NB) to 99.98% (RF), which was comparable to the range found in the 80:20 split. The consistency seen in this context underscores the adaptability of RF and DT algorithms, while also pointing out the enduring difficulties encountered by the NB algorithm.

When comparing the performance of ML models from the outputs of confusion matrices under different train-test ratios, notable differences emerge (Figs. 10 and 11). For the 80:20 split, LR and SVM both produce moderate false negatives, suggesting missed unauthorized transactions, with LR having 2,664 and SVM having 3,074. RF performs well with no misclassified transactions, demonstrating remarkable classification capabilities. KNN and NB had greater false negative rates, with KNN missing 5,905 fraud instances and NB missing 8,651, indicating their relative difficulties identifying fraud effectively. DT operates effectively with few mistakes.

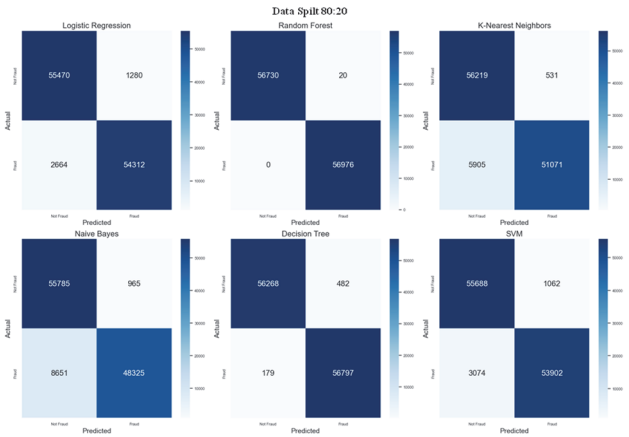

Figure 10 Confusion matrices for all models at 80:20 data division

In contrast, with a 70:30 split, the models often have more false negatives and fewer true positives owing to the larger dataset. RF retains its perfect classification performance, but LR and SVM have larger false negatives (3,949 and 4,586, respectively) than the 80:20 split. KNN and NB continue to struggle with significant false negative rates, although in larger absolute numbers. DT model demonstrates a reduction in false negatives while remaining effective. Overall, although the RF model consistently performs best across both ratios, other models exhibit varied degrees of performance depending on the train-test split, indicating trade-offs between model sensitivity and dataset size.

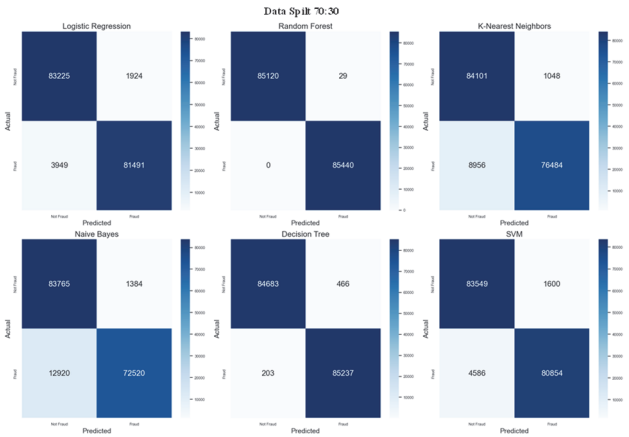

Figure 11 Confusion matrices for all models at 70:30 data division

The study evaluated ROC (AUC) curves for credit card fraud detection using two data split ratios: 80:20 and 70:30 (Fig. 12). RF attained a perfect AUC of 1.000 with an 80:20 split, indicating its outstanding ability to accurately categorise transactions. SVM was close behind, with an AUC of 0.9998, suggesting near-perfect performance. KNN again performed well, with an AUC of 0.9995. In comparison, LR achieved an AUC of 0.9934, indicating good performance, but somewhat lower than RF and SVM. DT obtained a high AUC of 0.9977, while NB had an acceptable 0.9745. When the data split was modified to 70:30, RF retained its ideal AUC of 1.000, while SVM and KNN maintained their excellent scores of 0.9998 and 0.9995, respectively. LR exhibited a slight rise with an AUC of 0.9935, while DT and NB showed small changes, with DT's AUC lowering to 0.9960 and NB's rising to 0.9750. Overall, RF was the most dependable algorithm across all data sets, demonstrating its better effectiveness in identifying fake transactions.

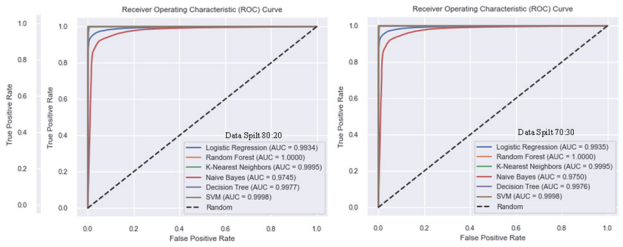

Figure 12 ROC (AUC) Curves at both data divisions

Moreover, the work focused on optimizing six ML models for fraud detection through hyperparameter tuning to enhance performance at both data division 80:20 and 70:30. Hyperparameter optimization is performed using GridSearchCV, a technique for systematically searching through a specified subset of hyperparameters to find the best combination for a given model. LR was optimized using a grid search over solvers, penalties, and regularization strengths. The tested solvers included "newton-cg", "lbfgs", and "liblinear", with "L2" as the penalty type. The regularization strength (C) values were [100, 10, 1.0, 0.1, 0.01].

In the analysis, the optimal model utilized the Logistic Regression algorithm with the "lbfgs" solver, "L2" penalty, and a regularization parameter (C) set to 10. This configuration yielded impressive results in both training and testing phases across different data split ratios. With an 80:20 data split, the model achieved a training accuracy of 96.49% and a test accuracy of 96.53%. When using a 70:30 split, the training accuracy slightly decreased to 96.48%, but the test accuracy improved marginally to 96.56%.

The RF model was optimised by fine-tuning numerous important hyperparameters to attain maximum efficiency. The tuning procedure included altering the number of trees (n_estimators), with values of 50, 100, and 200; the maximum number of features (max_features), set to either "sqrt" or "log2"; and the maximum depth of the trees (max_depth), with values of 3, 5, 10, 15, and 20 respectively. The most suitable layout was identified as 200 trees, "sqrt" for max_features, and a maximum depth of 5. The RF model, optimized for accuracy, achieved 99.98% in both training and testing with an 80:20 data split ratio. It also achieved 99.98% in both training and testing with a 70:30 data split, demonstrating its ability to accurately detect fraudulent transactions.

The K-Nearest Neighbors (KNN) model was optimized by adjusting the number of neighbors (n_neighbors), weight functions (weights), and distance metrics (p). Tested values included 3, 5, 7, and 9 neighbors; "uniform" and "distance" weights; and Manhattan distance (p = 1) and Euclidean distance (p = 2). The optimal configuration used 5 neighbors, "distance" weights, and Euclidean distance, achieving a training accuracy of 94.47% and a test accuracy of 94.94% with the 80:20 data split. With the 70:30 split, the model's performance was slightly lower, with a training accuracy of 94.29% and a test accuracy of 94.85%.

The DT model was optimized by tuning the maximum depth of the tree (max_depth) and the maximum number of features (max_features). The tested max_depth values included 3, 5, 10, 15, and 20, while max_features was set to "sqrt" or "log2". The best-performing configuration used a maximum depth of 5 and "sqrt" for max_features, achieving a training accuracy of 99.58% and a test accuracy of 99.46% with the 80:20 data split. With the 70:30 split, the model's performance improved slightly, reaching a training accuracy of 99.63% and a test accuracy of 99.65%.

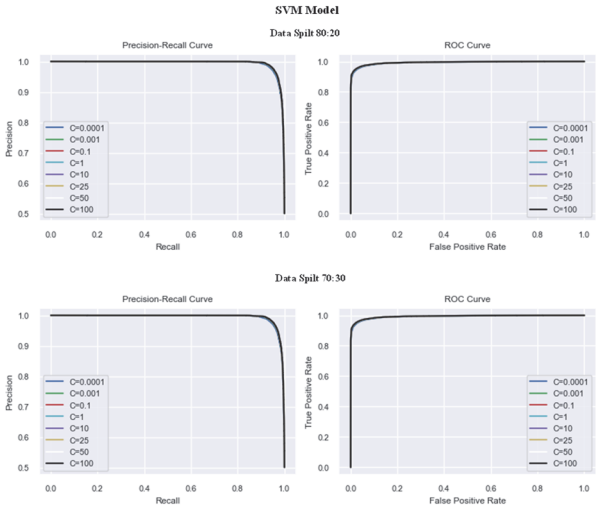

Figure 13 Optimisation of parameters for SVM model at both data spilt

To optimise the SVM model, hyperparameter tuning focused on three key parameters: C, gamma, and kernel type. The regularisation parameter (C) was assessed from 0.0001 to 100 to balance training error and model complexity (Fig. 13). Higher values of C decreased bias but increased variance, potentially leading to overfitting. Lower levels increased bias but enhanced generalisation. Gamma, the RBF kernel coefficient, was evaluated from 0.001 to 100 to determine the impact of each training sample on the decision boundary; lower gamma values allowed for larger impacts, while higher values focused on finer details. Different kernel types, such as "linear", "poly", and "rbf", were tested to change the input space and determine the best decision boundary. The best parameters were found to be C = 10, gamma = 10, and RBF kernel type. With these parameters, the model obtained a train and test accuracy of 96.00% across both data divisions.

For optimizing the NB model, hyperparameter tuning focused on two primary parameters: alpha and

var_smoothing. For the Multinomial Naive Bayes, the alpha parameter was tested with values ranging from 0.01 to 10. This smoothing parameter is crucial for handling zero counts and adjusting probabilities, with higher values providing more smoothing, which is beneficial for datasets with small sample sizes or rare events. In the Gaussian Naive Bayes model, var_smoothing was evaluated with values from 1×10^{-10} to 1×10^{-5}. This parameter adds a small value to the variance to stabilize calculations and prevent numerical issues, thus improving model performance on datasets with small variance. With an 80:20 split, the model attained a training accuracy of 91.54% and a test accuracy of 91.54%. When using a 70:30 split, the model showed a slightly improved training accuracy of 91.61% and a corresponding test accuracy of 91.61%. These results indicate consistent performance across different data partitions.

In comparison to the studies reviewed, this work aligns well with existing research on credit card fraud detection, while also demonstrating several key strengths. Xu (2024) evaluated various supervised learning algorithms for credit risk assessment and fraud detection. In fraud detection, the Neural Network led with an accuracy of 97.5% and a ROC-AUC of 0.88. The Gradient Boosting Tree and Random Forest performed similarly, with accuracies of 97.1% and 96.3%, and ROC-AUCs of 0.87 and 0.85, respectively. Kumar et al. (2024) focused on credit card fraud detection using machine learning. NB model achieved the highest accuracy at 99.30%, followed by RF and LR, both with 98.5% accuracy. Chang et al. (2024) investigated imbalanced datasets in fraud detection. Using SMOTE (oversampling), they achieved an accuracy of 86.75%, while Random Under-Sampling, which prioritized recall, was noted for improving recall, though its accuracy and precision were not explicitly reported.

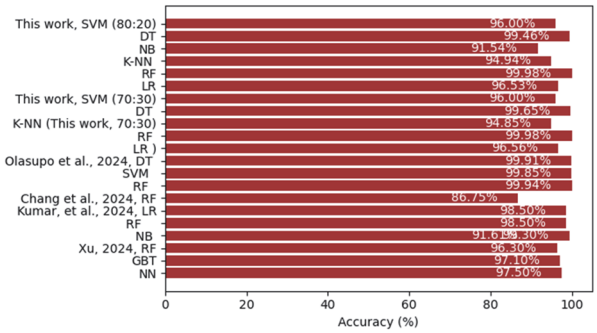

Figure 14 Comparison of results with existing literature for credit card fraud detection

Bhukya and Latha (2024) applied the RF algorithm for fraud detection using a highly imbalanced e-commerce dataset consisting of 85,275 genuine and 117 fraudulent transactions. Their evaluation metrics included accuracy, precision, sensitivity, specificity, and F1-score. The Random Forest model achieved an accuracy of 97.1%, precision of 95.7%, sensitivity of 95%, specificity of 95.9%, and an F1-score of 97.5%, showing its effectiveness in detecting fraud in large, imbalanced datasets. Olasupo et al. (2024) studied the use of three classifiers, SVM, RF, and DT, for credit card fraud detection. They applied the Modified Binary Bat Algorithm (MBBA) for feature selection, which improved the performance of the classifiers. RF achieved the highest accuracy at 99.945%, followed by Decision Tree at 99.909% and SVM at 99.847%.

In contrast, the proposed work demonstrates even higher performance, with RF achieving an accuracy of 99.98% when applied to the credit card fraud detection task. This result surpasses those found in previous studies, showcasing the effectiveness of the proposed models.

4 CONCLUSION

In summary, the work investigated the effectiveness of six ML algorithms in identifying fake credit card transactions. The results showed that Random Forest (RF) and Logistic Regression (LR) excelled in accuracy and area under the curve (AUC) metrics. RF achieved an accuracy of 99.98% and an AUC of 1.000, while LR reached an accuracy of 96.53% and an AUC of 0.9934. These models effectively distinguished between fraudulent and legitimate transactions. However, the study has limitations. It relied on a European cardholder dataset from 2023, which may restrict applicability to other regions and time periods. The performance of the models could also be influenced by the selection of only 30 features and the persistent class imbalance in the dataset. RF is complex and resource-intensive. LR may struggle with non-linear relationships. K-Nearest Neighbors (KNN) achieved an accuracy of 94.94% but can be sensitive to the choice of distance metric. Support Vector Machines (SVM) reached an accuracy of 96.00% but may underperform with noisy data. Decision Trees (DT) had an accuracy of 99.46% and are prone to overfitting. Naive Bayes (NB) had the lowest accuracy at 91.54% and assumes feature independence, limiting its effectiveness. Future research should incorporate diverse datasets, explore advanced feature engineering, and test sophisticated algorithms. This includes ensemble and deep learning approaches. Overall, this study highlights the potential of RF and LR to improve fraud detection systems and enhance financial security as credit card transactions continue to rise.

Acknowledgement

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ICAN (ICT Challenge and Advanced Network of HRD) support program (IITP-2025-2020-0-01825) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation).

5 REFERENCES

[1] TheGlobalEconomy.com. (n.d.). Percent people with credit cards by country, around the world. Retrieved from https://www.theglobaleconomy.com/rankings/people_with_credit_cards/

[2] White, A. (n.d.). Here's how credit card fraud happens and tips to protect yourself. CNBC Select. Retrieved from https://www.cnbc.com/select/credit-card-fraud/

[3] Jiang, S., Dong, R., Wang, J., & Xia, M. (2023). Credit card fraud detection based on unsupervised attentional anomaly detection network. *Systems*, *11*(6), 305. https://doi.org/10.3390/systems11060305

[4] Mohari, A., Dowerah, J., Das, K., Koucher, F., & Bora, D. J. (2021). Credit card fraud detection techniques: A review. *Soft Computing for Intelligent Systems. Algorithms for Intelligent Systems*, 157-166. https://doi.org/10.1007/978-981-16-1048-6_12

[5] Mekterović, I., Karan, M., Pintar, D., & Brkić, L. (2021). Credit card fraud detection in card-not-present transactions:

Where to invest? *Applied Sciences*, *11*(15), 6766. https://doi.org/10.3390/app11156766
[6] Mienye, I. D. & Sun, Y. (2023). A machine learning method with hybrid feature selection for improved credit card fraud detection. *Applied Sciences*, *13*(12), 7254. https://doi.org/10.3390/app13127254
[7] Rb, A. & Kr, S. K. (2021). Credit card fraud detection using artificial neural network. *Global Transitions Proceedings*, *2*, 35-41. https://doi.org/10.1016/j.gltp.2021.01.006
[8] Chandradeva, L. S., Amarasinghe, T. M., De Silva, M., Aponso, A. C., & Krishnarajah, N. (2019). Monetary transaction fraud detection system based on machine learning strategies. Fourth International Congress on Information and Communication Technology. *Advances in Intelligent Systems and Computing*, *1041*, 385-396. https://doi.org/10.1007/978-981-15-0637-6_33
[9] Baumann, M. (2021). Improving a rule-based fraud detection system with classification based on association rule mining. Gesellschaft für Informatik, 1121-1134.
[10] Stojanović, B., Božić, J., Hofer-Schmitz, K., Nahrgang, K., Weber, A., Badii, A., Sundaram, M., Jordan, E., & Runevic, J. (2021). Follow the trail: Machine learning for fraud detection in fintech applications. *Sensors*, *21*(5), 1594. https://doi.org/10.3390/s21051594
[11] Padhi, B. K., Chakravarty, S., Naik, B., Pattanayak, R. M., & Das, H. (2022). RHSOFS: Feature selection using the RoCkHyRax swarm optimization algorithm for credit card fraud detection system. *Sensors*, *22*(23), 9321. https://doi.org/10.3390/s22239321
[12] Feng, X. & Kim, S.-K. (2024). Novel machine learning-based credit card fraud detection systems. *Mathematics*, *12*(18), 1869. https://doi.org/10.3390/math12121869
[13] Chung, J. & Lee, K. (2023). Credit card fraud detection: An improved strategy for high recall using KNN, LDA, and linear regression. *Sensors*, *23*(18), 7788. https://doi.org/10.3390/s23187788
[14] Raval, J., Bhattacharya, P., Jadav, N. K., Tanwar, S., Sharma, G., Bokoro, P. N., Elmorsy, M., Tolba, A., & Raboaca, M. S. (2023). RAKSHA: A trusted explainable LSTM model to classify fraud patterns on credit card transactions. *Mathematics*, *11*(8), 1901. https://doi.org/10.3390/math11081901
[15] Carneiro, E. M., Forster, C. H. Q., Mialaret, L. F. S., Dias, L. A. V., & Da Cunha, A. M. (2022). High-cardinality categorical attributes and credit card fraud detection. *Mathematics*, *10*(20), 3808. https://doi.org/10.3390/math10203808
[16] Chang, V., Ali, B., Golightly, L., Ganatra, M. A., & Mohamed, M. (2024). Investigating credit card payment fraud with detection methods using advanced machine learning. *Information*, *15*(8), 478. https://doi.org/10.3390/info15080478
[17] Alfaiz, N. S. & Fati, S. M. (2022). Enhanced credit card fraud detection model using machine learning. *Electronics*, *11*(4), 662. https://doi.org/10.3390/electronics11040662
[18] Islam, S., Haque, Md. M., & Karim, A. N. M. R. (2024). A rule-based machine learning model for financial fraud detection. *International Journal of Electrical and Computer Engineering*, *14*(1), 759-771. https://doi.org/10.11591/ijece.v14i1.pp759-771
[19] Otoo, G., Appati, J. K., Yaokumah, W., Soli, M. A. T., Nwolley, S. J., & Ludu, J. Y. (2021). Evaluation of data imbalance algorithms on the prediction of credit card fraud. *International Journal of Intelligent Information Technologies*, *17*, 1-26. https://doi.org/10.4018/ijiit.289967
[20] Ileberi, E., Sun, Y., & Wang, Z. (2022). A machine learning-based credit card fraud detection using the GA algorithm for feature selection. *Journal of Big Data*, *9*(1). https://doi.org/10.1186/s40537-022-00573-8
[21] Alarfaj, F. K., Malik, I., Khan, H. U., Almusallam, N., Ramzan, M., & Ahmed, M. (2022). Credit card fraud detection using state-of-the-art machine learning and deep learning algorithms. *IEEE Access*, *10*, 39700-39715. https://doi.org/10.1109/access.2022.3166891
[22] Afriyie, J. K., Tawiah, K., Pels, W. A., Addai-Henne, S., Dwamena, H. A., Owiredu, E. O., Ayeh, S. A., & Eshun, J. (2023). A supervised machine learning algorithm for detecting and predicting fraud in credit card transactions. *Decision Analytics Journal*, *6*, 100163. https://doi.org/10.1016/j.dajour.2023.100163
[23] Esenogho, E., Mienye, I. D., Swart, T. G., Aruleba, K., & Obaido, G. (2022). A neural network ensemble with feature engineering for improved credit card fraud detection. *IEEE Access*, *10*, 16400-16407. https://doi.org/10.1109/access.2022.3148298
[24] Jebaseeli, T. J., Venkatesan, R., & Ramalakshmi, K. (2020). Fraud detection for credit card transactions using random forest algorithm. Intelligence in Big Data Technologies - Beyond the Hype. *Advances in Intelligent Systems and Computing*, *1167*, 189-197. https://doi.org/10.1007/978-981-15-5285-4_18
[25] Yan, C., Wang, J., Zou, Y., Weng, Y., Zhao, Y., & Li, Z. (2024). Enhancing credit card fraud detection through adaptive model optimization. *2024 IEEE 7th International Conference on Big Data and Artificial Intelligence (BDAI)*, 49-54. https://doi.org/10.1109/BDAI62182.2024.10692581
[26] Mienye, I. D. & Swart, T. G. (2024). A hybrid deep learning approach with generative adversarial network for credit card fraud detection. *Technologies*, *12*(10), 186. https://doi.org/10.3390/technologies12100186
[27] Alashwali, E., Chandrashekar, R. M., Lanyon, M., & Cranor, L. F. (2024). Detection and impact of debit/credit card fraud: Victims' experiences. https://arxiv.org/abs/2408.08131
[28] Dataset Link. (2024). Credit card fraud detection dataset 2023. Retrieved from https://www.kaggle.com/datasets/nelgiriyewithana/credit-card-fraud-detection-dataset-2023
[29] Xu, T. (2024). Fraud detection in credit risk assessment using supervised learning algorithms. *Computer Life*, *12*(2), 30-36. https://doi.org/10.54097/qw9j1892
[30] Kumar, A., Poojitha, M. V., Anuhya, T., Srinivas, K., & Bhargavi, M. (2024). Credit card fraud detection. *8th International Conference on Inventive Systems and Control (ICISC)*, 79-82. https://doi.org/10.1109/ICISC62624.2024.00020
[31] Chang, V., Ali, B., Golightly, L., Ganatra, M. A., & Mohamed, M. (2024). Investigating credit card payment fraud with detection methods using advanced machine learning. *Information*, *15*(8), 478. https://doi.org/10.3390/info15080478
[32] Bhukya, D. & Latha, D. (2024). An intelligent approach to credit card fraud detection using random forest. *Journal of Theoretical and Applied Information Technology*, *102*(20), 7347-7355.
[33] Olasupo, Y. A., Malgwi, M. Y., & Hambali, M. A. (2024). Enhancing credit card fraud detection with modified binary bat algorithm: A comparative study with SVM, RF, and DT. *Journal of Computer Science and Engineering (JCSE)*, *5*(2), 80-98.

Contact information:

Hye Jin KIM
Dept. of Smart Convergence Security, Busan University of Foreign Studies, 65, Geumsaem-ro 485beon-gil, Geumjeong-gu, Busan, Republic of Korea
E-mail: 20225432@office.bufs.ac.kr

Jung Soo RHEE
(Corresponding author)
Dept. of Smart Convergence Security, Busan University of Foreign Studies, 65, Geumsaem-ro 485beon-gil, Geumjeong-gu, Busan, Republic of Korea
E-mail: 20236457@bufs.ac.kr