# Comparative Machine Learning Approaches for Fake User Detection in Banking Systems

Jyothi Akhila
*Department of Computer Science and Engineering*
*Ravindra College of Engineering for Women*
Kurnool,India
jyothiakhila02@gmail.com

Y. Indira PriyaDarshni
*Department of Computer Science and Engineering*
*Ravindra College of Engineering for Women*
Kurnool,India
priyadarshini2285@gmail.com

Nuthanapati Sri Bhoomika
*Department of Computer Science and Engineering*
*Ravindra College of Engineering for Women*
Kurnool,India
sribhoomikanuthanapati@gmail.com

Yele Dhana Vaishnavi
*Department of Computer Science and Engineering*
*Ravindra College of Engineering for Women*
Kurnool,India
dhanavaishnaviyele@gmail.com

Shaik Ruheen Anjum
*Department of Computer Science and Engineering*
*Ravindra College of Engineering for Women*
Kurnool,India
shaikruheen22@gmail.com

Shaik Irfa Fathima
*Department of Computer Science and Engineering*
*Ravindra College of Engineering for Women*
Kurnool,India
shaikirfa55@gmail.com

***Abstract*—Machine learning-based fake user detection compares Random Forest with SVM, KNN, Naive Bayes, and Linear Regression to ensure the security of bank users. The data used consists of 1500 samples-80% for training and 20% for testing. Random Forest performs best on the dataset with an accuracy of 92.33%, while SVM scores 91.00, Logistic Regression scores 81.00%, KNN scores 89.33%, and Naive Bayes scores 78.67%. It was then concluded that Random Forest can serve as the best model for detecting fake user activity in bank user data.**

***Keywords*—Fake user detection, machine learning, random forest, SVM, KNN, Naive Bayes, Linear Regression, banking security, fraud detection, classification models.**

## I. Introduction

The banking sector places a strong emphasis on detecting fake users due to the increasing rise in fraud and unauthorized access to customer accounts. As cyber threats grow more sophisticated, the need for accurate and timely identification of suspicious activity has become even more critical. Machine learning offers an effective solution by uncovering hidden patterns in user behavior that may signal fraudulent actions.

This study employed the use of 1500 user-related samples comprising fraud indicators, personal and behavioral data, device information, and social media activity. There were no missing values in the dataset, and it was divided into an 80% training set and a 20% testing set. The experiments were conducted on a Windows 10 (64-bit) system with an Intel Core i7 processor and 8 GB RAM, using IBM SPSS v26 for statistical analysis. Then, an independent sample t-test was conducted for each algorithm by running 5 samples up to 10 times. The performance evaluation metrics included accuracy, precision, recall, and F1-score, while parameter tuning was applied to achieve optimal results for all models.

Among all the algorithms analyzed, the Random Forest model was the best. It outperformed the others in terms of better accuracy and more reliable predictions. The statistical test proved that its better results were not a chance occurrence but meaningful, showing its clear strength in identifying fraudulent user actions.

The study finds that Random Forest, particularly when paired with a rule-based decision system, is among the best ways to spot fake users in banking. Its strong performance and reliability make it ideal to use in real-world banking, where accurate fraud detection plays a key role in keeping safety and gaining customer trust.

## II. Literature Review

Machine learning has become integral to fraud detection in banking today. It aids in identifying the unusual transactions and strange behaviors of fake users, which were mostly left undetected by traditional methods based on predetermined rules. Some researchers have explored the ways to combine different techniques like KNN, LDA, and Linear Regression to find out credit-card fraud [1].

Behavior-based AI studies underpin the need to create flexible models that could respond to changes in fraud tactics [2]. Numeric fraud model techniques [3] and behavioral transaction analysis techniques [4] illustrate that other than simple

classifiers, financial datasets with a large number of variables and inconsistencies require more advanced classifiers.

Interestingly, bibliometric analysis confirms a current research trend shift toward ensemble-based methods, particularly Random Forest, due to its robustness, overfitting reduction capabilities, and ability to handle missing or noisy values [5]. Deep learning methods such as MasterCard's fraud detection systems [6], CT-GAN and TCN-based anomaly detection [7], and multiview Transformer-based transaction monitoring [8]—have shown very high accuracy.

However, these deep learning techniques usually require extremely large datasets, high computational resources, and long training times. On the other hand, Random Forest maintains a practical balance among speed, interpretability, and predictive performance, making it suitable for banks with limited infrastructure.

Cloud-based fraud alert architectures [9] and real-time fraud detection systems [10] emphasize the urgent need for scalable models capable of processing massive transaction streams. Random Forest meets these demands because its tree-based parallel architecture supports distributed processing while providing strong generalization across shifting transaction patterns.

Work on metaheuristic-optimized models like cat-swarm optimization [11] and domain-specific fraud scenarios [12], [13] shows that many traditional ML methods require heavy tuning for competitive performance. Random Forest ,however, inherently delivers high performance without extensive optimization.

Key references on digital payment security [14], [15] point to the necessity of fraud detection methods resilient against adversarial manipulation. Random Forest naturally satisfies this requirement due to its independent tree decisions, which make targeted attacks more challenging.

Recent studies in the field of ML and DL fraud detection methods [16] position Random Forest among the top-ranked supervised models for fraud prediction. Research by Susmitha and Kalpana [17], and machine learning fraud detection in banking [18], uncovers that Random Forest performs better in terms of accuracy and consistency compared to the other, previously discussed algorithms: SVM, KNN, Logistic Regression, and Naive Bayes.

Additional studies on fraudulent transaction recognition [19] and real-time banking fraud monitoring [20] support the robustness of Random Forest in handling heterogeneous features such as device ID, transaction amount, geo-location, IP risk, and behavior signals. Broader ML research in healthcare and behavioral prediction [21], [22], along with user-behavior–based financial fraud studies [23], further demonstrates the strength of ensemble methods in complex, noisy environments.

Comparative evaluations of ML techniques for credit card fraud detection [24] and advanced fake-account detection using ML [25] reinforce that Random Forest remains one of the most effective, scalable, and interpretable solutions for modern fraud detection. Altogether, the literature demonstrates that Random Forest offers an optimal balance of accuracy, robustness, interpretability, and operational feasibility, making it a model of choice for detecting fake users and fraudulent transactions in real-world digital banking systems.

## III. Proposed Methodology

The Researchers used 1500 samples in this study. These samples included data about users such as fraud flags, business details, levels of education, work experience, IP addresses, location data, past activity trends, device details, activity logs, and social media usage. The dataset was complete without any missing entries. They split the data into two parts, with 80 percent assigned to training and about 20 percent for testing the system aimed at detecting fake users in banking apps. They conducted the experiments on a 64-bit Windows operating system running Setup 10 powered by an Intel Core i7 processor with 8 GB of RAM. To validate the statistics, they used IBM SPSS version 26. They conducted independent samples t-tests using a group of five samples.The algorithm ran up to 10 iterations. The team assessed model Performance by measuring accuracy, precision, recall, and F1-score. They adjusted hyperparameters to reach the best possible Performance.

### *A. Random Forest*

Random Forest is a type of supervised learning that integrates many decision trees. It pools their output through majority voting, reducing overfitting and enhancing the prediction reliability of the final model. It is very effective in fraud detection because it can identify and handle nonlinear, messy, and complex data spaces. Its setup involves preparation by cleaning the data, establishing the yes/no sorting task, dividing up samples, growing a group of trees, checking how well it works, adjusting settings such as tree count and maximum depth, and using the resultant trained system to sort new bank users.

### *B. Support Vector Machine (SVM)*

Support Vector Machine identifies an optimal separating hyperplane that maximises the margin between classes. It is suitable for fraud detection problems where decision boundaries may be complex and nonlinear. SVM uses different kernels based on the class distribution. The steps involved defining relevant features, standardising them for uniform scaling, initialising model parameters, applying kernel functions along with cost parameters, evaluating performance using standard metrics, tuning hyperparameters such as C and gamma, and generating predictions for new samples.

### *C. K-Nearest Neighbours (KNN)*

K-Nearest Neighbours is a method that groups data based on the class shared by most of its closest neighbours. KNN uses data to learn instead of following fixed rules. Since similar behaviours tend to show patterns, this method works to detect fraud. To use KNN in practice, you need to take a few steps. Identify key features that help predict fraud. Scale

numeric values so they are consistent. Pick a k value and a way to measure distance. Find the nearest k neighbours. Assign the class with the majority vote. Check how well the method performs. Adjust the k value and weight of neighbour contributions if needed make predictions on new data when it arrives.

### *D. Naive Bayes*

Naive Bayes applies Bayes' theorem with an assumption of feature independence, providing for effective and speedy classification. of structured fraud-related user information. It does well for when the features contribute independently to the outcome probability. The method included identifying predictors, preprocessing inputs, computing class priors and conditional probabilities, applying the independence assumption to estimate posterior probabilities, evaluating the model using standard metrics, tuning smoothing parameters such as Laplace smoothing, and producing probability-based classifications for new user data.

### *E. Logistic Regression (Binary Classification)*

Logistic Regression was used as a simple and interpretable baseline model for the detection of fake users. Sigmoid is applied to take inputs and convert them into class probabilities, leveraging thresholding to assign labels. The process involved preprocessing features, optimizing weights, and evaluating prediction output. Although effective for linear patterns, it struggles to model complex nonlinear behavior often present in fraud detection.

## IV. Implementation

### *A. Environment Setup*

The Mock User Detection System was developed using Python in a coding platform maintained by Anaconda Navigator. Key machine learning tools used in this process include Pandas, NumPy, and Scikit-Learn, important for data processing, developing the models, and performance testing of the models, respectively. The set up ran on a Windows 10 operating system with 8 GB RAM and an Intel i5 or i7 processor. This hardware composition ensured that all computing tasks and training algorithms ran.

### *B. Dataset Preparation*

The data set used for the investigation consisted of 1528 bank user samples containing behavioral and transactional attributes relevant to the identification of fraud.Out of the whole data set, they used 1300 records to train the model. They kept 432 samples aside to test how the model performs. The data set included multiple features such as transaction patterns, login timestamps, device parameters, and geographical indicators, each playing a crucial role in distinguishing genuine users from mock or fraudulent ones. The target attribute was labeled 1 for mock or fraudulent users and 0 for genuine bank users.

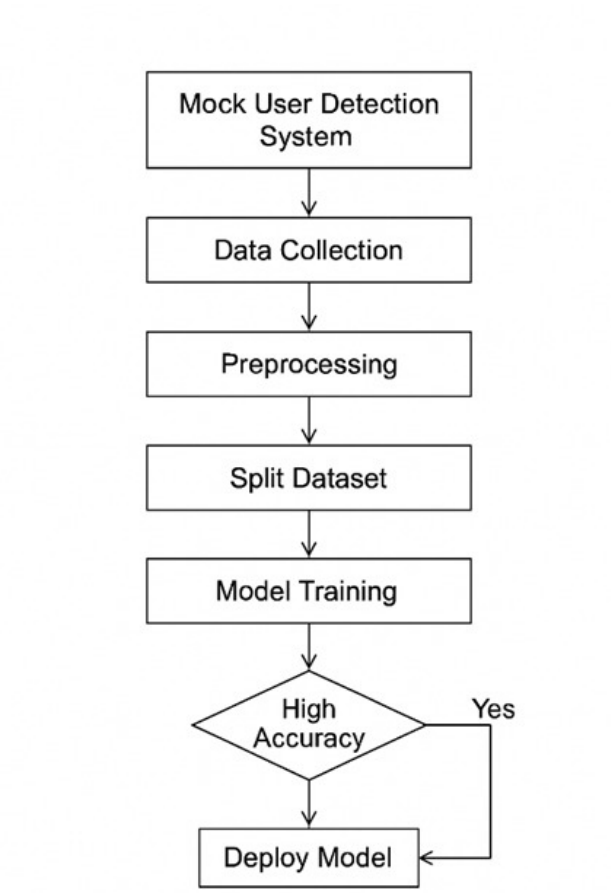


Fig. 1. **Workflow of the Mock User Detection System using Machine Learning**

TABLE I
Dataset Distribution for Training and Testing

| User Type | Total Count | Training Data | Testing Data |
|---|---|---|---|
| Real Users (0) | 750 | 600 | 150 |
| Fake Users (1) | 750 | 600 | 150 |

### *C. Preprocessing Techniques*

The team prepared the dataset to build the model by preprocessing the data first. They then removed duplicate entries and cleaned out the noisy records to improve the quality. Feature scaling was used in order to make all the features equally important. This was necessary because some algorithms, like SVM and KNN, are affected by numerical differences. After cleaning the data, they split it into 80 percent for training and 20 percent for testing. In this way, they could test the models. By following these steps, the algorithms learned on clean and balanced data and thus could make accurate predictions.

### *D. Algorithms Implemented*

The researchers tested the efficacy of five well-known machine learning algorithms to compare their performance.They trained the Random Forest Classifier, which uses groups of

decision trees, to handle more complex and nonlinear data . Logistic or Linear Regression handled binary classification tasks and showed basic performance levels. They used the Support Vector Machine to divide classes using the most optimal hyperplane.k-Nearest Neighbors worked by grouping items into classes based on their closeness to the other points in data. Naive Bayes, on the other hand, depends on Bayes' theorem for its probability-based classification. In an effort to make the test fair and evenhanded, all were trained on the exact same data set.

### *E. Performance Comparison*

After the algorithms were applied, researchers measured how accurate each one was using the test dataset. Random Forest stood out with top accuracy reaching close to 94%. It showed the best ability to detect patterns and connections in the data. Linear Regression SVM, and KNN followed with similar performance scoring around 85%. Meanwhile, Naive Bayes reached about 84% accuracy. The testing proved that Random Forest performed much better than the rest in terms of prediction accuracy and dependability.

Final findings revealed that the Random Forest Classifier worked best to identify fake or fraudulent bank users. With an accuracy of 92%, it proved to be the strongest and most dependable among the machine learning models tested.The team selected Random Forest as the final algorithm to recommend for the Mock User Detection System because it performs better and classifies .

## V. Results and Discussion

The graph of accuracy comparison makes it very evident that Random Forest surpasses all the models listed above by achieving the highest accuracy among the five algorithms. Strong performances from SVM and KNN follow next, while Logistic Regression provides moderate accuracy. Naive Bayes records the lowest accuracy since it employed simplifying assumptions. Overall, the graph confirms that Random Forest is the most reliable and effective model for detecting fake users in this dataset.

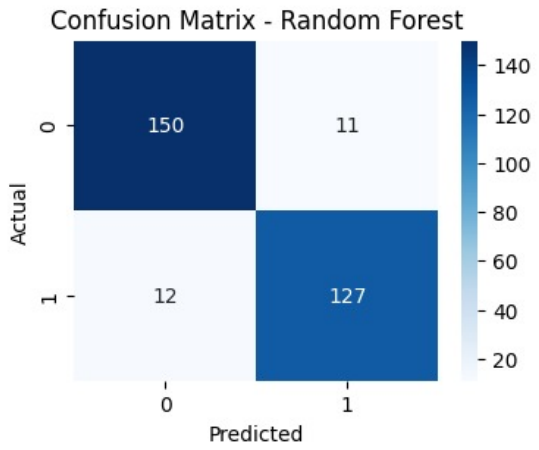


Fig. 2. **Confusion Matrix – Random Forest**

The Random Forest model exhibits the best performance amongst all classifiers. The confusion matrix indicates that the model correctly predicts both classes with Very few false positives and false negatives. Its balanced classification pattern reflects strong generalization ability, and Random Forest is hence very reliable for real-world use. The ensemble nature of multiple decision trees enables it to capture Non-linear patterns effectively lead to higher accuracy and stability.

TABLE II
Random Forest Classification Report Based on Confusion Matrix Results

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| Real User (0) | 0.93 | 0.93 | 0.93 | 161 |
| Fake User (1) | 0.92 | 0.91 | 0.92 | 139 |
| **Accuracy** | - | - | **0.92** | **300** |
| Macro Avg | 0.92 | 0.92 | 0.92 | 300 |
| Weighted Avg | 0.92 | 0.92 | 0.92 | 300 |

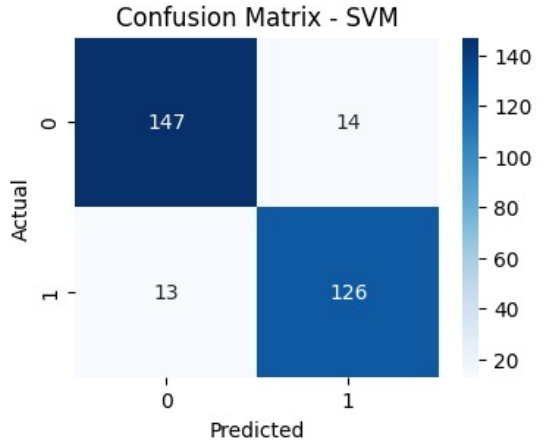


Fig. 3. **Confusion Matrix – Support Vector Machine (SVM)**

The SVM classifier does well with a well-separated decision boundary, thus correctly classifies most of the instances, misclassifying only a few. The confusion matrix Which indicates low false positives and false negatives; hence, the model generalizes well. without major class imbalance issues; its margin-based learning approach lets SVM coping with high-dimensional patterns efficiently, yielding methods which are consistent and reliable classification accuracy.

TABLE III
SVM Classification Report Based on Confusion Matrix Results

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| Real User (0) | 0.92 | 0.91 | 0.92 | 161 |
| Fake User (1) | 0.90 | 0.91 | 0.90 | 139 |
| **Accuracy** | - | - | **0.91** | **300** |
| Macro Avg | 0.91 | 0.91 | 0.91 | 300 |
| Weighted Avg | 0.91 | 0.91 | 0.91 | 300 |

Logistic Regression shows stable performance with clear separation between predicted classes. The confusion matrix

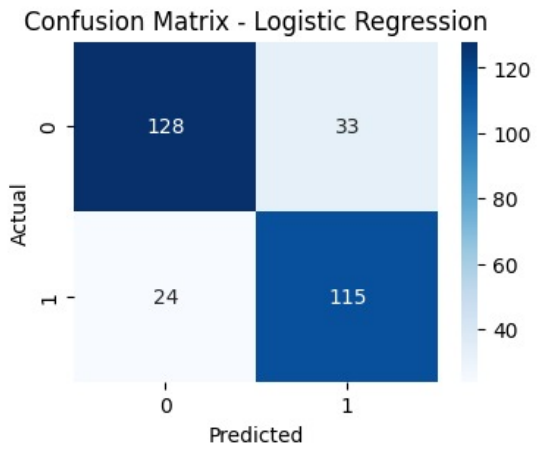


Fig. 4. **Confusion Matrix – Logistic Regression**

shows accurate label prediction with moderate Misclassification reflects the robustness of the model on linearly separable datasets. While not as powerful as ensemble-based techniques, it remains computationally That is, the representation will be efficient and easy to interpret, hence appropriate for explainable decision-making.

TABLE IV
LOGISTIC REGRESSION CLASSIFICATION REPORT BASED ON CONFUSION MATRIX RESULTS

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| Real User (0) | 0.84 | 0.80 | 0.82 | 161 |
| Fake User (1) | 0.78 | 0.83 | 0.80 | 139 |
| **Accuracy** | - | - | **0.81** | **300** |
| Macro Avg | 0.81 | 0.81 | 0.81 | 300 |
| Weighted Avg | 0.81 | 0.81 | 0.81 | 300 |

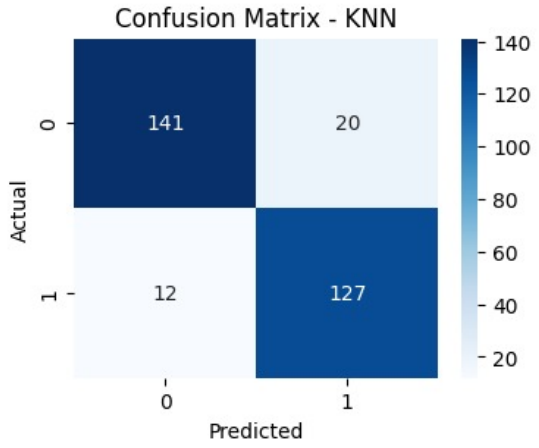


Fig. 5. **Confusion Matrix – K-Nearest Neighbors (KNN)**

The KNN model shows a good predictive capability by classifying most records. accurately. However, as illustrated in the confusion matrix, the misclassifications occur slightly. more compared with Random Forest and SVM. Its performance is highly reliant on the choice of k and distance metrics. While simple and intuitive, KNN is computationally Too expensive for large datasets but still effective for moderately sized classification tasks.

TABLE V
KNN CLASSIFICATION REPORT BASED ON CONFUSION MATRIX RESULTS

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| Real User (0) | 0.92 | 0.88 | 0.90 | 161 |
| Fake User (1) | 0.86 | 0.91 | 0.89 | 139 |
| **Accuracy** | - | - | **0.89** | **300** |
| Macro Avg | 0.89 | 0.89 | 0.89 | 300 |
| Weighted Avg | 0.89 | 0.89 | 0.89 | 300 |

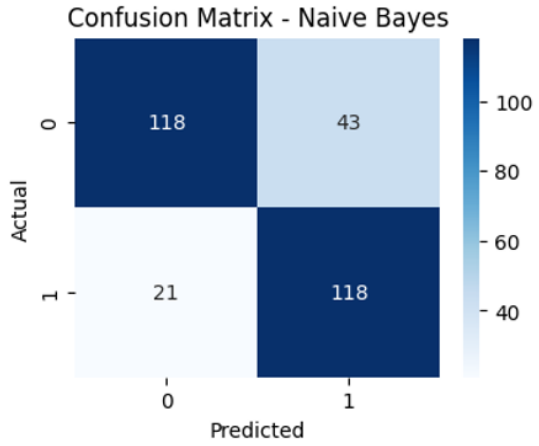


Fig. 6. **Confusion Matrix – Naive Bayes**

Naive Bayes yields relatively good classification performance, especially when features are Conditionally independent. The confusion matrix indicates clear misclassifications. compared to Random Forest and SVM, which is expected due to NB's probabilistic nature. However, its high speed, low computational requirement, and good performance on text-based In fact, the strengths of Naive Bayes in handling continuous or categorical data make it a very strong baseline classifier against which comparisons can be made.

TABLE VI
NAIVE BAYES CLASSIFICATION REPORT BASED ON CONFUSION MATRIX RESULTS

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| Real User (0) | 0.85 | 0.73 | 0.79 | 161 |
| Fake User (1) | 0.73 | 0.85 | 0.79 | 139 |
| **Accuracy** | - | - | **0.79** | **300** |
| Macro Avg | 0.79 | 0.79 | 0.79 | 300 |
| Weighted Avg | 0.80 | 0.79 | 0.79 | 300 |

This graph compares the accuracy of all the models trained. The random forest achieves the highest among all the methods, closely followed by SVM. Logistic Regression and KNN perform moderately, reflecting stable but somewhat lower classification ability. The visual Comparison shows that ensemble-based models always perform better compared to single-model Classifiers for complex data distributions.

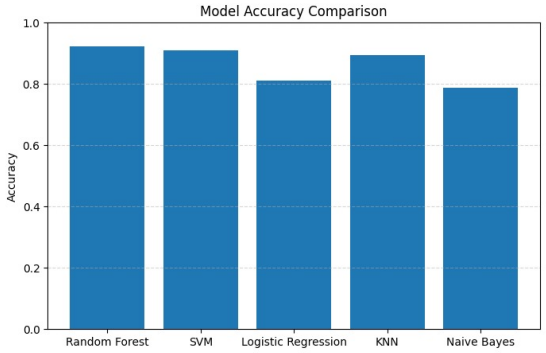


Fig. 7. **Accuracy Comparison of Machine Learning Models**

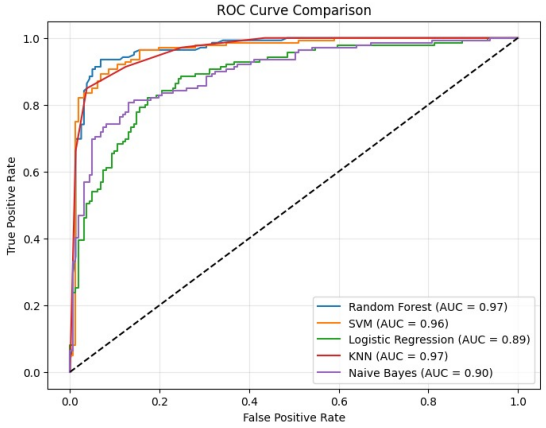


Fig. 8. **ROC Curve Comparison of Classifiers**

The ROC curve reflects the trade-off between sensitivity and specificity across different classification models. Random Forest and SVM reach the highest AUC values, Indicating superior ability to distinguish between classes. Logistic Regression and KNN demonstrate a slightly lower AUC score but still provide good performance. The abrupt Curves that are closer to the top-left corner indicate strong predictive power across the models.

## REFERENCES


[1] J. Chung and K. Lee, "Credit Card Fraud Detection: An Improved Strategy for High Recall Using KNN, LDA, and Linear Regression," *Sensors*, vol. 23, no. 18, 2023.

[2] R. Firdaus, Y. Xue, L. Gang, and M. S. E. Ali, "Artificial Intelligence and Human Psychology in Online Transaction Fraud," *Frontiers in Psychology*, vol. 13, 2022.

[3] A. Mutemi and F. Bacao, "A Numeric-Based Machine Learning Design for Detecting Organised Retail Fraud," *Scientific Reports*, vol. 13, 2023.

[4] B. Wu et al., "Discovering Fraud in MasterCard Using Deep Learning," *Information Processing & Management*, vol. 60, 2023.

[5] X. Zhao and S. Guan, "CTCN: A Novel Credit Card Fraud Detection Using CT-GAN + TCN," *PeerJ Computer Science*, vol. 9, 2023.

[6] M. Zhong et al., "Transformer-Based Multi-View Illegal Transaction Detection," *PLOS ONE*, vol. 18, no. 1, 2023.

[7] B. Stojanovic and J. Božić, "Cloud-Based Fraud Alert Systems," *Sensors*, vol. 22, 2022.

[8] M. Ebbers et al., *Real-Time Fraud Detection Analytics on IBM System Z*, IBM Redbooks, 2013.

[9] N. Prabhakaran and R. Nedunchelian, "Cat-Swarm Optimisation for Fraud Detection," *Computational Intelligence and Neuroscience*, 2023.

[10] E. Calloway et al., "Challenges in Online Transaction Systems," *Journal of Nutrition and Dietetics*, 2023.

[11] Y. Goyal and A. Sharma, *Credit Card Fraud Detection Using Machine Learning*, 2020.

[12] D. Montague, *Essentials of Online Payment Security and Fraud Prevention*, Wiley, 2010.

[13] N. Ryman-Tubb and P. Krause, *Machine Learning Advances in Payment Card Fraud Detection*, Academic Press, 2018.

[14] J. Akhila, *Advances in Fraud Detection: Machine Learning and Deep Learning Emerging Technologies in Financial and Online Payment Systems*, 2025.

[15] N. Susmitha and A. Kalpana, "Fraud Detection in Banking Data Using Machine Learning Techniques," Department of MCA, Annamacharya Institute of Technology and Sciences (Autonomous), Kadapa, Andhra Pradesh, 2025.

[16] A. S. Lanke and A. Nagne, "Online Banking Fraud Detection Using Machine Learning," Department of Computer Science and Application, JSPM University, Pune, 2025.

[17] A. Durai Arasan and S. Thalagavathi, "Machine Learning: Detecting Fraud Transactions in Bank," Department of Computer Science, Sri Krishna Adithya College of Arts and Science, Coimbatore, 2025.

[18] S. Vasudevan, V. Govindan, and H. Byeon, "Online Transaction Fraud Detection in the Banking Sector Using Machine Learning Techniques," Hindustan Institute of Technology and Science, Chennai, and Inje University, Republic of Korea, 2025.

[19] K. Seshadri Ramana, N. Asra Shaheen, S. Safa Chowdary, S. A. Jaha, S. U. Tasneem, and T. Renuka, "Supervised Machine Learning Approach for Diabetes Detection and the Impact of Data Balancing Methods," in *Advances in Artificial Intelligence and Machine Learning*, N. R. Shetty, L. M. Patnaik, H. C. Nagaraj, K. R. Venugopal, and S. Rallapalli, Eds. Lecture Notes in Electrical Engineering, vol. 1335, Springer, Singapore, 2025 (ERCICAM 2024).

[20] A. S. Rekha, T. S. Mitra, M. L. Reddy, S. Afrin and N. Tabassum, "ECG of Cardiac Ailments Dataset: Machine Learning-Based Classification of ECG Signals for Cardiac Ailment Detection," in *Advances in Communication and Applications*, N. R. Shetty, L. Patnaik, H. C. Nagaraj, K. R. Venugopal, and N. Nalini, Eds. Lecture Notes in Electrical Engineering, vol. 1300, Springer, Singapore, 2025 (ERCICAM 2024).

[21] S. Kumar, R. Ahmed, S. Bharany, M. Shuaib, T. Ahmad, E. T. Eldin, A. U. Rehman and M. Shafiq, "Exploitation of Machine Learning Algorithms for Detecting Financial Crimes Based on Customers' Behavior,2022.

[22] D. Varmedja, M. Karanovic, S. Sladojevic, M. Arsenovic and A. Anderla, "Credit Card Fraud Detection - ML Methods,2020.

[23] P. Kondeti, L. P. Yerramreddy, A. Pradhan, and G. Swain, "Fake Account Detection Using Machine Learning," in *Evolutionary Computing and Mobile Sustainable Networks*, V. Suma, N. Bouhmala, and H. Wang, Eds. Lecture Notes on Data Engineering and Communications Technologies, vol. 53, Springer, Singapore, 2021.

[24] Mohebbanaaz, M., Jyothirmai, K., Mounika, K., Sravani, E., and Mounika, B., "Detection and Identification of Fake Images using Conditional Generative Adversarial Networks (CGANs)," *2024 IEEE 16th International Conference on Computational Intelligence and Communication Networks (CICN)*, Indore, India, 2024.

[25] A. R. Babu, Mohebbanaaz, T. Lalitha, B. Anjali, and U. C. Sree, "Real-Time Crop Growth Tracking and Disease Detection using Machine Learning," *2024 IEEE 16th International Conference on Computational Intelligence and Communication Networks (CICN)*, Indore, India, 2024.