Available online at http://icsejournal.com/index.php/JCSE
Journal of Computer Science an Engineering (JCSE)
e-ISSN 2721-0251
Vol. 5, No. 2, August 2024, pp. 80-98

Enhancing Credit Card Fraud Detection with Modified Binary Bat Algorithm: A Comparative Study with SVM, RF, and DT

Y. A. Olasupo[1*], M. Y. Malgwi[2], M. A. Hambali[3]

[1,3]Department of Computer Science, Federal University Wukari, Taraba, Nigeria
[2]Department of Computer Science, Modibbo Adama University, Yola, Nigeria
Email: ademolay15@gmail.com; yumalgwi@mau.edu.ng; hamberlite@gmail.com
corresponding*

ARTICLE INFO

Article History:
Received December 22, 2023
Revised January 01, 2024
Accepted March 15, 2024

Keywords:
Credit card
Classification
Feature Selection
Fraud
Bat Algorithm

Correspondence:
E-mail: ademolay15@gmail.com

ABSTRACT

Numerous studies have revealed the problem of irrelevant features, noise, and dimensionality in a dataset, which can inhibit how the classification algorithm performs. In machine learning, feature selection approaches are critical, particularly in the context of credit card fraud detection, where relevant feature selection is critical. We use techniques such as machine learning algorithms, data mining techniques, and data science to stop and detect credit card fraud. These algorithms often classify genuine and fraudulent transactions in credit card datasets. However, the challenge of high dimensionality and irrelevant features persists, hindering improvements in classifier algorithms. This study centered on detecting credit card fraud (CCF) using a Modified Binary Bat Algorithm (MBBA) for feature selection. The MBBA selects the most informative features to improve the classifier algorithm's performance. The classifier algorithms used in this research are Support Vector Machine (SVM), Random Forest (RF), and Decision Tree (DT). We conducted the experiment using the Python programming language, and the results indicate that RF achieves 99.945% accuracy, SVM 99.847%, and DT 99.909%. As a result, RF has the best accuracy. In summary, the optimal performance of a classification algorithm depends on the selection of relevant features for credit card fraud detection. The paper suggests improving the effectiveness of classifier algorithms for credit card fraud detection by employing the Modified Binary Bat algorithm, which outperforms the Genetic Algorithm (GA) in feature selection.

1. Introduction

Credit cards are issued to customers to give them immediate access to goods and services and the ability to withdraw cash at a predetermined period for payback by Banks and other financial institutions. The bearer of a credit card can pay for products and services ahead of time. Credit card information is taken in various ways, including card details theft, phishing websites, credit card loss, intercepted cards, counterfeit credit cards, and many others [1]. The quantity of fraudulent operations is skyrocketing [2]. The prevalence of illicit transactions has increased, so obtaining goods and services and collecting money in unethical ways are commonplace [3]. Fraud is an unethical method of receiving money, products, or services, and it is a worldwide concern today [4].

Fraud involving credit cards poses a substantial threat to financial institutions and individuals. The key challenge is improving the effectiveness of fraud detection systems, especially when handling credit card transaction datasets. Challenges in this domain include irrelevant features, dynamic transaction patterns, and a higher frequency of genuine transactions than fraudulent ones. Credit card fraudsters exploit the speed at which they can withdraw money, often without the credit card owner's immediate awareness. The difficulty emerges from the fraudster's ability to disguise illicit transactions as legitimate, making fraud detection challenging. Within the domain of detecting credit card fraud, user activities are meticulously tracked to identify any abnormal patterns associated with defaulting, intrusion, or fraudulent behavior [5]. To counteract this, there is a

https://doi.org/10.36596/jcse.v5i2.771

crucial need for preventive measures to halt the progression of credit card fraud. Machine learning algorithms are utilized as an automated approach for detecting credit card fraud, playing a crucial role in recognizing and addressing illicit activities linked to credit card transactions. The emphasis is on developing preventive measures that can either identify or significantly minimize the occurrence of credit card theft in the future [5].

The tools employed in this preventive approach by many researchers encompass a range of sophisticated techniques, including machine learning algorithms, data mining approaches, and data science methodologies. Machine learning algorithms have been widely used to discern between genuine and fraudulent transactions within credit card datasets. Leveraging these advanced technologies aims to detect ongoing fraudulent activities and proactively prevent and reduce the likelihood of credit card fraud. Incorporating machine learning into preventive measures improves the system's capacity to adjust to developing fraudulent strategies, strengthening its defense against unauthorized transactions in the constantly changing landscape of credit card fraud. Conventional fraud detection methods, like rule-based systems, have demonstrated inadequacy in addressing this persistently evolving issue. The sheer volume of transactions and fraudsters' continuous adaptation makes it difficult to differentiate between legitimate and fraudulent transactions. Consequently, there is a need to create and implement advanced techniques and algorithms to achieve more precise and efficient fraud detection. A critical hurdle in credit card fraud detection revolves around feature selection, a crucial phase in crafting effective machine learning models. The careful selection of relevant features from the credit card transaction dataset is essential to significantly enhance the performance of the classification algorithm. Several approaches, including filter, wrapper, embedded, and meta-heuristic methods, have been explored to address this challenge. Each method has advantages and drawbacks, and selecting the right one depends on the specific dataset and the classification algorithm used.

One noteworthy meta-heuristic algorithm in the spotlight is the Modified Binary Bat Algorithm (MBBA), which has demonstrated its effectiveness in feature selection for high-dimensional datasets [6]. Despite these encouraging advancements, the challenge of credit card fraud detection persists. This is evident in ongoing research efforts to discover novel techniques and algorithms aiming to improve the precision and efficiency of fraud detection. Additionally, the issue of class imbalance in credit card fraud detection datasets remains a persistent challenge. Fraudulent transactions usually constitute a small fraction of the total transactions, posing a difficulty for machine learning models to learn effectively. Researchers like [7] have proposed innovative strategies to address this problem, including anomaly detection models and divide-and-conquer approaches. Despite these efforts, class imbalance remains a significant hurdle in achieving accurate fraud detection. Furthermore, the constantly changing nature of fraudulent activities demands adaptive and resilient solutions. Previous research [8] propose a technique like the ensemble model based on deep neural networks with long short-term memory (LSTM). They demonstrated the need for sophisticated and evolving methodologies to stay ahead of fraudsters.

The issue of detecting credit card fraud is a multifaceted and constantly changing problem that adapts to the ongoing advancements in fraudulent techniques. The introduction of cutting-edge algorithms and techniques, such as the Modified Binary Bat Algorithm, is a testament to the ongoing efforts to address this issue. Nevertheless, the enduring existence of class imbalance, the necessity for efficient feature selection, and the continually shifting landscape of fraud patterns underscore the formidable complexity of this challenge, demanding cutting-edge solutions to mitigate its consequences effectively. The rest of this study is divided into the following sections: Related work, Research methods, Result and Discussion, and Conclusion.

https://doi.org/10.36596/jcse.v5i2.771

2. Related Works

An effective fraud detection strategy must overcome challenges to attain optimal outcomes [9]. They noted that credit card fraud detection systems face various hurdles. Obstacles and challenges are Imbalanced data, Misclassification mistakes of varying importance, Overlapping data, Inability to adapt, and Lack of standard metrics. Research by [10] Demonstrated and examined the application of machine learning for credit card fraud detection, presenting both the algorithm and pseudocode. The algorithms employed include Isolation Forest and Local Outlier Factors.

In previous research, machine learning algorithms were proposed for developing normal and fraudulent behavior characteristics from historical transaction data to enhance real-time credit card fraud detection [11]. In their study, various algorithms, including Support Vector Classifier (SVC), Multinomial Naïve Bayes (MNB), K-Neighbors Classifier (KNC), Logistic Regression (LR), Random Forest (RF), Bernoulli Naïve Bayes (BNB), and Support Vector Machine (SVM), were evaluated. LR, RF, BNB, and SVM achieved 100% accuracy in the training dataset, with RF outperforming others by achieving 100% in both train and test data. The remaining algorithms showed satisfactory results ranging from 75.3% to 99.6%. A detailed comparison was conducted, including accuracy, ROC curve, train-test and validation scores, and the learning curve. RF emerged as the best-performing algorithm, scoring 100% in both train and test data, and was discussed based on its Area Under Curve (AUC) score and confusion matrix.

Research by [12] propose a model that uses the bat algorithm to improve neural network learning. To improve the accuracy of the suggested method, K-means clustering is utilized to exclude data from the dataset. The data was run using MATLAB software, and the information gathered is relevant to bank fraud. They compared the method to data mining and learning techniques like logistic regression, SVM, and backup machines. The study's accuracy, Sensitivity, and specificity indicators are 91.46 percent, 88.97 percent, and 90.32 percent, respectively, and they are more accurate and sensitive than the data mining and learning approaches utilized.

Removing irrelevant features and redundancy has helped to improve machine learning algorithms. Introduce a genetic algorithm for feature selection, identifying Random Forest as the most effective classifier among the selected priority features in terms of accuracy and precision [13]. Proposed a modified binary bat algorithm to identify optimal features in microarray datasets for cancer detection, demonstrating improved classification algorithm performance [6]. This study advocates for using the bio-inspired algorithm, a modified binary bat algorithm, for feature selection to enhance classifier algorithms in credit card fraud detection. The comparison of results with existing findings and each other reveals the superior performance of the bat algorithm, especially in unrestricted optimization tasks [14].

Employed an optimized light gradient boosting machine, OlightGBM, to create an intelligent credit card fraud detection solution. They utilized Bayesian-based hyperparameter optimization algorithms to fine-tune the light gradient boosting machine (Light GBM) settings, conducting a 5-fold cross-validation on two real-world public credit card transaction datasets. The researchers optimized the parameters using Bayesian-based hyperparameter optimization algorithms, resulting in impressive outcomes. The proposed approach achieved a 92.88 percent AUC, surpassing mild GBM with 90.62 percent AUC and catbost with 87.86 percent AUC. With a 98.40 percent accuracy rate, the recommended approach showed the best accuracy, outperforming other methods such as Local Outlier Factor (97 percent), Random Forest (95.50 percent), Isolation Forest (95 percent), ANN (92.86 percent), and Concept Drifts Adaptation (92.86 percent) (80 percent) [15].

https://doi.org/10.36596/jcse.v5i2.771

Propose a credit card fraud detection model, STAN, employing a spatio-temporal attention-based neural technique. The model incorporates features engineering, a spatio-temporal attention layer using a 3D convolutional network (3DConNet), and a detection layer, applying optimization strategies. Utilizing data from a large commercial bank covering January to December 2016, the dataset comprises 236,706 transaction records, 1021 users, and 1160 fraud-affected location codes. Their method emphasizes real-time fraud detection through an online learning mechanism, outperforming state-of-the-art methods on benchmark datasets. It notably introduces the first application of attention to 3DConNet to credit card fraud detection [16]. Argue that supervised learning faces challenges adapting to evolving customer behavior and emerging fraud patterns. They advocate for using unsupervised learning algorithms to identify anomalies in fraud detection. Their proposed strategy, tested on a dataset of 76 million transactions recorded by World line partners from February to December 2016, demonstrated effectiveness and improved fraud detection accuracy [17].

Table 1. Summary of Related Works

No	Authors/Date	Method/s used	Solution Proffered
1.	M. A. Hambali, T. O. Oladele, K. S. Adewole, A. K. Sangaiah, and W. Gao[6]	Combined the best of the series of filter algorithm with Modified Binary Bat Algorithm. The approaches used were implemented using R-Studio.	Small round blue cell tumors (SRBCT) dataset was used and the results show that the model is effective for feature selection(FS) techniques.
2.	S. T. Mohammad, F. Neda, , C. Shah, ,S. Wasei, A. K. Musaddiq, Y. A. Mst [11]	Utilizing machine learning algorithms to develop normal and fraudulent behavior characteristics from historical transaction data for effective real-time credit card fraud detection.	RF emerged as the best-performing algorithm, scoring 100% in both train and test data, and was discussed based on its Area Under Curve (AUC) score and confusion matrix.
3.	A. Harshita, G.Richa, and C. Raman[10]	Machine learning algorithm: Isolation Forest algorithm, Local Outlier Factor, and Support Vector Machine (SVM)	The result shows that the Isolation Forest algorithm reaches an accuracy of 99.75%, while that of the Local Outlier Factor was 99.66%. Accuracy increased to 33% when only a tenth of the dataset was used. This is due to the huge imbalance between the number of valid and authentic transactions.
4.	Y. K. Saheed, M. A. Hambali , M. O. Arowolo, Y. A.Olasupo [13]	Two priority phases: the first priority features phase and the second priority features. The Random Forest (RF), Naïve Bayes (NB), and Support Vector Machine (SVM) classifiers techniques. German credit card dataset was used.	The results show that the first priority features gave the most important features, and the RF performed better than other algorithms used in terms of accuracy(96.40%), precision (96.5%), and Sensitivity (96.4%), while NB has the highest specificity (96.7%).
5.	C. Dawei, X. Sheng, S.Chencheng, Z. Yiyi, Y. Fangzhou, and Z. Liqing [16]	The process of features engineering is thespatio–temporal attention layer in conjunction with a 3D convolution network.	The dataset comprises real-world credit card transaction records from a central commercial Bank. It includes 236,706 transaction records, 1021 users, and 1160 location codes affected by fraud.
6.	A. T. Altyeb , and J. M. Sharaf [15]	Dataset used was trained with parameters optimized by using based hyperparameter optimization Algorithm.	The proposed method has the highest accuracy (98.40%), Local Outlier Factor (97%), Random Forest (95.50%), Isolation forest (95%), ANN (92.86%), and Concept Drifts Adaptation (80%).
7.	C. Fabrizio , L. B. Yann-A¨el , C. Olivier , K.Yacine , O. Frederic, and B. Gianluca [17]	Unsupervised outlier scores are computed at different levels of granularity, compared, and tested on a real-world credit card fraud detection dataset.	The dataset used comprises 334 days of transactions recorded in 2016 by individual partners, worldline. It includes 76 million transactions and 0.36% of fraudulent transactions.

https://doi.org/10.36596/jcse.v5i2.771

Table 1. Summary of Related Works (Continued)

No	Authors/Date	Method/s used	Solution Proffered
8.	S.Nikfar , and B. Touraj [12]	Learning of neural networks and improving learning using the Bat algorithm. K-means clustering was used to remove data from the dataset to increase the accuracy of the proposed method.	Results show that the accuracy, Sensitivity, and specificity indicators are 91.46%, 88.97%, and 90.32%, respectively, and they are more accurate and sensitive than data mining and learning methods.
9.	Doreswamy, M., and UmmeSalma, [14]	It uses a bio-inspired modified binary bat algorithm for feature selection to enhance classifier algorithms in credit card fraud detection.	The comparison of results with existing findings and each other reveals the superior performance of the bat algorithm, especially in unrestricted optimization tasks.
10.	S. Samaneh, Z. Zahra, and E. Reza [9]	A survey on an effective fraud detection strategy	Effective fraud detection must overcome challenges to attain optimal outcomes.

3. Method

3.1. Conceptual Framework

The proposed approach's framework and methods are depicted in Figure 1. The procedure was divided into three (3) stages. In the first stage, data was partitioned into train and test datasets. The feature subset selection stages are the second stage, and they use an evolutionary strategy based on the Modified Binary Bat (MBB) algorithm. MBB Algorithm was used as the subset selection algorithm at this stage, and Random Forest (RF) was used as the fitness function. The classification stage was the third stage, where the performance of several classifier algorithms was compared with the existing research work and the dataset without feature selection. The Python programming language was used to accomplish the technique. Figure 1, depicts the steps of the conceptual framework of the proposed MBB classification model.

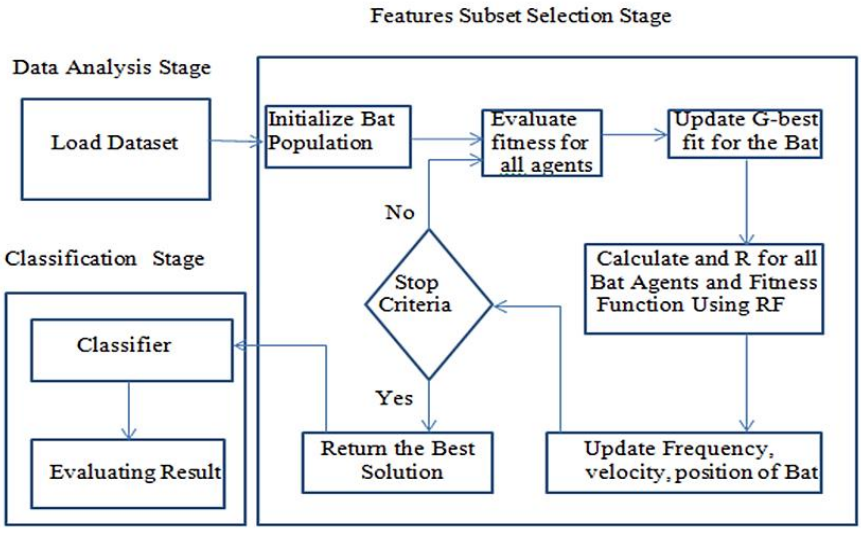

Figure 1. Conceptualized Framework of Modified Binary Bat Classifications Model

3.2. Dataset

A credible dataset is required to test the proposed method and compare its efficiency. Researchers utilized a genuine or synthetically generated dataset in certain credit card fraud detection studies. The credit card datasets employed in this research were acquired from Kaggle, a well-regarded platform for sharing datasets and hosting data science competitions. Kaggle datasets often undergo

https://doi.org/10.36596/jcse.v5i2.771

quality checks and are curated by the community, which adds to the credibility and trustworthiness of the data source. The choice of this specific dataset is justified because it offers a realistic representation of credit card transactions, has an imbalanced class distribution, is sourced from a reputable platform, and is publicly available. These characteristics make it relevant and suitable for evaluating the proposed modified binary bat algorithm and selecting the most informative features from the dataset.

The dataset employed for detecting credit card transaction fraud is a simulated dataset from kaggle.com. Available at https://www.kaggle.com/kartik2112/fraud-detection/, it covers genuine and fraudulent transactions conducted by a cardholder from January 1, 2019, to December 31, 2020. The dataset includes 284,807 transactions, with 492 (or 0.172 percent) identified as fraudulent. It features 1000 clients interacting with 800 merchants. Porandon Harris' Sparkor Data Generation GitHub program was utilized to generate this dataset. The rationale for choosing this specific dataset is supported by various key factors that establish its relevance to the research problem: (**i) Realistic Representation of Credit Card Transactions:** The dataset includes transactions carried out by a cardholder during a span of two years from January 1, 2019, to December 31, 2020. This time frame reflects recent and real-world transactions, which is crucial for evaluating the effectiveness of any fraud detection algorithm. Researchers often prefer real data as it closely resembles the challenges faced in practical scenarios. (**ii) Sufficient Data Volume:** Possessing a significant amount of data is crucial for efficiently training and evaluating machine learning models. In this case, the dataset's size provides enough samples to comprehensively assess the proposed modified binary bat algorithm. (**iii) Imbalanced Nature:** Credit card fraud detection often involves dealing with a highly imbalanced class distribution, where genuine transactions far outnumber fraudulent ones. This dataset reflects this scenario, with only 492 transactions (0.172 percent) being fraudulent. This makes it a suitable choice for assessing the algorithm's performance in handling imbalanced data, a common challenge in fraud detection research.

3.3. Feature Subset Selection Stage

Selecting features plays a crucial role in the methodology of credit card fraud detection using the modified binary bat algorithm, and this is attributed to several compelling reasons: **(i) Dimensionality Reduction:** Credit card transaction datasets may encompass numerous features, including transaction amount, merchant details, transaction time, and other parameters. A high number of features can result in heightened computational complexity and an elevated risk of overfitting. Feature selection diminishes irrelevant or redundant features, streamlining the dataset and bringing out the most informative features for analysis. **(ii) Enhanced Model Efficiency:** Removing irrelevant or redundant features through feature selection leads to a more efficient and faster model. With fewer features, machine learning algorithms require less computational resources and time to train. This efficiency is crucial in real-time systems where quick decisions are essential. **(iii) Improved Model Generalization**: Feature selection contributes to model generalization by preventing overfitting. By selecting only the most relevant features, the model is less likely to overfit and can better generalize to new, unseen credit card transactions. **(iv) Enhanced Fraud Detection Rates:** The choice of features plays a pivotal role in detecting fraudulent transactions accurately. Choosing the most discriminative features can greatly enhance the model's capacity to detect fraudulent patterns, resulting in increased fraud detection rates.

The feature selection (FS) approach enhances the learning performance, which reduces duplicate features in the dataset [6]. The Credit Card Transaction dataset's influence of irrelevant attributes is lessened using the FS approach, which also increases the fraud detection rate. When employing a feature selection measure like correlation, consistency, etc., the feature subset selection technique

https://doi.org/10.36596/jcse.v5i2.771

creates possible combinations of feature subsets. The feature selection system enables them to hunt even in full darkness by choosing between obstruction and prey. Thus, feature selection is important in the credit card fraud detection methodology because it reduces dimensionality, enhances model efficiency, and improves fraud detection rates. It is supported by relevant literature and studies emphasizing its importance in achieving accurate and efficient fraud detection.

3.4. Binary Bat Algorithm

Researchers from various fields have been captivated by the impressive echolocation abilities of bats. Echolocation, a type of sonar predominantly used by microbats, involves emitting a brief, powerful sound pulse, awaiting its reflection of an object, and then receiving the echo after a short delay. By utilizing this technique, bats can accurately determine the distance of objects from them, showcasing exceptional orienting capabilities.

In artificial intelligence, the artificial Bat can traverse the continuous real domain, moving between locations in search of a space to update its velocity and position vectors. A transfer technique is necessary to handle a binary search space, where the position vector is represented by binary bits (0's and 1's). This method replicates the change from "0" to "1" in position vector elements and velocity vector values, or vice versa. Introduced the Binary Bat Algorithm, as depicted in Equation (i), incorporating a sigmoid function to confine the new Bat's location exclusively to binary values of 0 and 1 [18].

$$S(\mathcal{V}_j^i) = \frac{1}{1+ e^{-\mathcal{V}i}} \quad (1)$$

$$(D_j^i) = \begin{cases} 1, & \text{if } (V_j^i) > \sigma, \\ 0, & \text{otherwise,} \end{cases} \quad (2)$$

With σ being drawn from a uniform distribution U(0,1), Equation (ii) is constrained to generate binary values for the coordinates of each Bat in the Boolean lattice. These binary values signify the presence or absence of features.

Explanation of the Modified Binary Bat Algorithm

Bat Algorithm (BA) based feature selection has a significant potential to become ensnared in local optima during the procedure. As a result, MBBA was created, which combines a global search strategy with certain extra processes in order to enhance the outcomes of a feature selection technique based on BA.

Encoding and Fitness Functions

The MBB Algorithm uses a binary string to encode each Bat in the population; the length of the string depends on the number of features in the dataset. Each bat bit so exhibits a unique quality. Equation iii illustrates how, for instance, each Bat in a dataset with n characteristics will be represented by an n-bit string.

$$D_i^j = (D_i^1, D_i^2, D_i^3, \ldots D_i^n),\ \text{i} = 1, 2, 3, \ldots, \text{k}\ , D_i^j = \{0, 1\} \quad (3)$$

Where n defines the size of the bats, and $D_i^j = \{0, 1\}$ specifies the jth feature within the selected feature subset was assessed by a fitness function defined in Equation iv. This function assesses each potential solution, where S(D) is the total number of selected features and Acc is the classification accuracy determined by the Random Forest classifier.

$$\text{Fit (D)} = \alpha \times \text{Acc} + (1 - \alpha) \times \frac{n - S(D)}{n} \quad (4)$$

MBB begins by initializing the bat population and parameters, including frequency, loudness, velocity, and pulse rate. If the maximum iteration criteria are met, the frequency is adjusted, other population characteristics are updated, and the fitness function is computed using Equation (iv).

https://doi.org/10.36596/jcse.v5i2.771

The Acc is calculated using the Random Forest (RF) classifier as the fitness function. The procedure continues until the best and most acceptable Gb solution is found, and then it stops. Table 2 illustrates the parameters of the MBB algorithm. This table outlines the parameters and actions involved in the Modified Binary Bat Algorithm.

Table 2. Parameters Description for MBB Algorithm

Parameters	Description
Population	Bat population (Di, i = 1, 2, 3,...,n)
Randomization	rand (0 or 1)
Velocity	$\mathcal{L}$i = 0
Pulse Frequency	$\mathcal{F}$i
Pulse Rates	$\mathcal{R}$i
Loudness	Li
Maximum Iterations	Maximum iterations (t)
Update Rules	Update velocities and adjust frequencies
Fitness Function	Fit (D) = α × Acc + (1 – α) × (n - S(D)) / n
Dimensions	D_i^j = (D_i^1, D_i^2, D_i^3, …, D_i^n) |
Solution Space	D_i^j = {0, 1}
Randomness Thresholds	rand, $\mathcal{R}$i, Li
Best Solution Selection	Choose the overall best solution ($\mathcal{G}$b) and alter Di
New Solution Generation	Generate new solution randomly
Acceptance Condition	if ((rand < Li) and (f(Di) < f($\mathcal{G}$b)) then
Update Rates	Raise $\mathcal{R}$i and decrease Li
Best Solution Assessment	Locate the present $\mathcal{G}$b and assess the ranking of the Bat

Justification for Selecting Modified Binary Bat Algorithm (MBBA)

Binary Encoding Suitability: The primary advantage of BBA is its binary encoding, which aligns well with the nature of feature subset selection in the dataset, either included (1) or excluded (0) from the subset. This binary representation allows for straightforward feature selection without the complexity of continuous values. **(i) Global Search Strategy:** BBA's global search strategy is particularly valuable in locating the most informative features from the dataset. Fraud patterns can be highly diverse and evolve over time. BBA's global exploration of the search space ensures that it can identify relevant features across various types of fraudulent activities. This is especially important for capturing both common and rare fraud scenarios. (ii) **Diversity Preservation:** BBA introduces diversity in the search process through frequency and loudness parameters. The diversity aids in preventing the algorithm from becoming trapped in local optima, increasing the likelihood of uncovering valuable feature subsets, even when confronted with noise or intricate data distributions. (iii) **Adaptive Frequency:** BBA's ability to adjust the pulse frequency ($\mathcal{F}$i) allows it to allocate more exploration or exploitation efforts based on the problem's characteristics. In the context of credit card fraud detection, where fraud patterns dynamically change, this adaptability proves advantageous.

Modified Binary Bat Algorithm (MBBA)

Enhanced Binary Search: MBBA builds upon BBA by introducing additional processes to improve the feature selection process. Its modification aims to address potential limitations of BBA, such as trapping in local optima. **Improved Global Search:** MBBA combines a global search strategy with extra processes, making it particularly suitable for feature selection. By enhancing the global search, MBBA increases the chances of finding optimal or near-optimal feature subsets. **Fitness Function Integration:** The use of Random Forest (RF) as a fitness function within MBBA is well-justified. RF is recognized for effectively managing high-dimensional data and intricate relationships, making it well-suited for assessing feature subsets in credit card fraud detection. The ensemble approach of RF mitigates overfitting risks and enhances the evaluation's robustness. **Binary Encoding Maintenance:** MBBA maintains the binary

https://doi.org/10.36596/jcse.v5i2.771

encoding of features, aligning with the binary nature of feature selection. This ensures that the selected subset consists of discrete features crucial for interpretability and practical application.

Advantages of MBBA Over Other Feature Selection Methods

Global Exploration: Unlike traditional feature selection methods, such as wrapper approaches, BBA and MBBA provide a more thorough feature space exploration. This proves crucial in credit card fraud detection, where fraudsters employ diverse tactics that may not align with specific feature subsets. **Reduced Dimensionality:** BBA and MBBA efficiently reduce the dataset's dimensionality by selecting the most relevant features. This improves the efficiency of subsequent classification algorithms and reduces the risk of overfitting, a common challenge in fraud detection. **Adaptability:** The adaptive nature of BBA and MBBA, including parameter adjustments, enables them to respond to changes in the fraud landscape. As fraud patterns evolve, the algorithms can adjust their search strategies to identify new patterns. **Complex Relationships:** BBA and MBBA's global search and RF-based fitness function empower them to capture intricate relationships and interactions among features. This is vital for detecting subtle fraud patterns that may not be apparent with simple feature selection methods.

In summary, the BBA and MBBA offer a potent combination of binary encoding, global search, and adaptability, making them well-suited for selecting feature subsets in credit card fraud detection. Their unique characteristics provide advantages over traditional feature selection methods and enhance the ability to identify relevant features in a dynamic and complex fraud detection environment.

3.5. Random forest

A random forest is a machine learning algorithm that operates as an ensemble. Unlike regular decision tree algorithms, it randomly generates root node splits. Random forests are frequently identified as the most accurate learning algorithms. Table 3. outlines the various parameters and actions involved in the GenerateDecisionTree'function of the Random Forest algorithm.

Table 3. Parameters Description for Random Forest Algorithm

Parameters	Description
Function	GenerateDecisionTree (Sample S, Features F)
Stopping Condition	stopping_condition(S, F) is true
Leaf Node Creation	leaf = createNode(), leafLabel = classify(S)
Root Node Creation	root = createNode()
Test Condition	root.test_condition= findBestSplit(S, F)
Possible Outcomes	V = {v | v is a possible outcome of root.test_condition}
Loop Over Outcomes	for each value v in V
Subset Creation	S_v = {s | root.test_condition(s) = v and s ∈ S}
Recursive Call	child = GenerateDecisionTree(S_v, F)
Add Child to Root	add child as a descendant of root
Edge Labeling	label the edge {root → child} as v
Return Root	return root

Justification for using Random Forest (RF)

Ensemble Learning: Random Forest is an ensemble method that combines multiple decision trees to improve accuracy and robustness. Ensemble methods are precious in credit card fraud detection, where accuracy and robustness are paramount. **Mitigating Overfitting:** Random Forest's ensemble strategy helps mitigate overfitting, making it more reliable when dealing with noisy or imbalanced data. **Complex Pattern Recognition:** Fraud patterns can be highly complex and dynamic. Random Forest excels at capturing complex patterns and interactions among features.

Strengths of RF: Ensemble learning reduces overfitting and enhances generalization, Can handle both classification and regression tasks, is Effective in high-dimensional spaces, Robust to outliers and noisy data, and Provides feature importance ranking.

https://doi.org/10.36596/jcse.v5i2.771

Weaknesses of RF: It may be computationally intensive for large datasets, Less interpretable compared to a single decision tree, and Parameter tuning is required for optimal performance.
Ensemble Methods (Random Forest): Ensemble techniques such as RF prove highly beneficial in detecting credit card fraud, given their capacity to manage intricate data distributions and noisy datasets. As credit card fraud patterns may change over time, these approaches demonstrate effective adaptability to evolving patterns. Additionally, ensemble methods are robust to outliers and can provide more accurate results by aggregating predictions from multiple trees.

3.6. Support Vector Machine (SVM)

The SVM is a widely employed machine learning method for regression and classification tasks. It leverages the kernel trick technique to transform data and establish an optimal boundary between various outcomes based on the transformation's findings. The border the algorithm calculates in a non-linear SVM does not have to be a straight line. This allows you to record far more intricate relationships between your data pieces without executing complex modifications yourself. The disadvantage is that it takes much longer to train because it is more computationally costly. Table 4 outlines the main steps and conditions in the given section of the Support Vector Machine (SVM) algorithm.

Table 4. Parameters Description for SVM Algorithm

Parameters	Description
Loop Initialization	candidateSV = {closest pair from opposite classes}
Loop Condition	while there are violating points
Find a Violator	Find a violator
Update candidateSV	candidateSV = U candidateSV S
Update Violator	Violator
Loop Initialization	candidateSV = {closest pair from opposite classes}
Check Condition	if any a < O due to addition of c to S then
Prune Points	candidateSV = candidateSV\p # Prune points until all such such
points are eliminated	points are eliminated
Repeat Pruning	repeat till all such points are pruned
End While	end while

Justification for using SVM

Effective Separation of Classes: SVM is recognized for its capability to discover an optimal hyperplane that maximizes the margin between classes. In the context of credit card fraud detection, where the objective is to distinguish legitimate transactions from fraudulent ones, SVM's ability to establish distinct class boundaries proves to be highly advantageous. **Non-Linearity Handling:** SVM can address both linear and non-linear classification challenges by incorporating kernel functions. This is essential as credit card fraud patterns may exhibit complexity and non-linearity. **Robustness to High-Dimensional Data:** credit card transaction datasets frequently encompass many features. SVM demonstrates efficacy in high-dimensional spaces, rendering it well-suited for datasets abundant in features. **Regularization:** SVM includes regularization parameters that help prevent overfitting, an essential factor in fraud detection that lies in the model's ability to generalize effectively to unfamiliar data. **Strengths of SVM:** Effective in finding clear class boundaries, versatile with handling linear and non-linear data, robust in high-dimensional spaces, and able to mitigate overfitting through regularization. **Weaknesses of SVM: Parameter Tuning:** SVM requires tuning of hyperparameters like the kernel function and regularization parameters, which can be time-consuming. **Resource-intensive in Computation:** Computational demands can be significant when working with SVM, particularly with extensive

https://doi.org/10.36596/jcse.v5i2.771

datasets. **Limited Interpretability:** SVM models are less interpretable than decision trees, which can be a drawback in some scenarios.

3.7. Decision Tree

A supervised learning approach that can address classification and regression issues is the decision tree (DT). It can provide a training model that learns decision rules derived from training data and can predict classes of variables or target variables. Table 5 shows the main steps and conditions in the given section of the Decision Tree algorithm.

Table 5. Parameters Description for DT Algorithm

Parameters	Description
Input	GenDecTree(Sample S, Features F)
Output	Root
If Condition	If stopping_condition(S, F) = true then
Leaf Node Creation	Leaf = createNode()
Leaf Label Assignment	leafLabel = classify(s)
Root Node Creation	root = createNode()
Condition Test	root.test_condition = findBestSplit(S, F)
Set of Possible Outcomes	V = {v | v is a possible outcome of root.test_condition}
For Each Value in V	For each value v Є V: |
Subset Creation	S_v = {s | root.test_condition(s) = v and s Є S}
Recursive Tree Growth	Child = TreeGrowth(S_v, F)
Add Child to Root	Add child as descendant of root and label the edge {root → child} as v
Return Root	return root

Justification for using DT

Interpretability: Decision trees are highly interpretable, rendering them apt for offering insights into the elements influencing decisions in fraud detection, which can be valuable for investigation. **Non-Linearity Handling:** Decision trees capture non-linear relationships between features, which is essential for detecting complex fraud patterns. **Feature Importance:** Decision trees naturally prioritize features based on their significance in the classification process, assisting in the selection of features. **Strengths of DT:** Interpretability, making it easier to understand decision-making; Can handle non-linear relationships and interactions between features; Feature importance ranking aids in feature selection; robust to outliers. **Weaknesses of DT:** (i) Susceptible to Overfitting: Decision trees may overfit the training data, diminishing generalization performance. (ii) Unstable: Minor fluctuations in the data can result in distinct tree structures. (iii) Limited to Binary Decision: Traditional decision trees are binary, which may not capture more complex decision boundaries.

Conclusively, the choice of SVM, DT, and RF as classifier algorithms for credit card fraud detection considers their unique strengths and weaknesses. SVM effectively separates classes and handles non-linearity, Decision Trees offer interpretability and non-linear handling, and Random Forest combines the benefits of ensemble learning, robustness, and feature importance ranking. Through employing this combination, the study seeks to harness the advantages of each algorithm to enhance the overall performance of fraud detection.

3.8. Data analysis and preprocessing

The file name is the credit card dataset, as shown in Figure 2. The read dataset file is a .csv, which can be opened and read with the help of the Pandas package in the Python code environment. Table 7. shows the credit dataset after reading the file.

https://doi.org/10.36596/jcse.v5i2.771

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	...	-0.018307	0.277838	-0.110474	0.066928	0.128539	-0.189115	0.133558	-0.021053	149.62	0
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.225775	-0.638672	0.101288	-0.339846	0.167170	0.125895	-0.008983	0.014724	2.69	0
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	...	0.247998	0.771679	0.909412	-0.689281	-0.327642	-0.139097	-0.055353	-0.059752	378.66	0
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	...	-0.108300	0.005274	-0.190321	-1.175575	0.647376	-0.221929	0.062723	0.061458	123.50	0
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	...	-0.009431	0.798278	-0.137458	0.141267	-0.206010	0.502292	0.219422	0.215153	69.99	0

5 rows × 31 columns

Figure 2. Credit card dataset after reading the file

Figure 3. illustrates the class distributions of the dataset. Class 0 is no fraud, while 1 is with fraud.

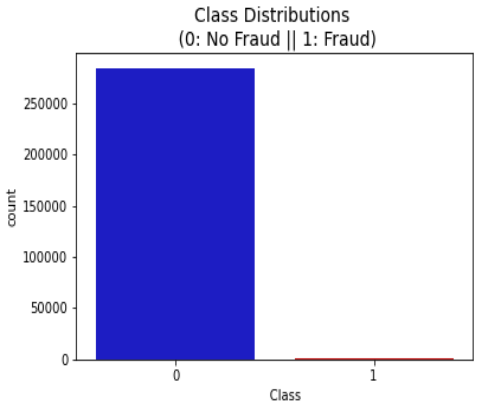

Figure 3. Class distributions of the dataset

Data preprocessing

There are 31 features and 284807 instances in the credit card dataset. There are no missing values, as shown in Figure 4. making it easy for the classifier to be trained.

Scaled_amount	0
Scaled_time	0
V1	0
V2	0
V3	0
V4	0
V5	0
V6	0
V7	0
V8	0
V9	0
V10	0
V11	0
V12	0
V13	0
V14	0
V15	0
V16	0
V17	0
V18	0
V19	0
V20	0
V21	0
V22	0
V23	0
V24	0
V25	0
V26	0
V27	0
V28	0
Class	0
dtype: int64	0

Figure 4. Dataset without missing values

https://doi.org/10.36596/jcse.v5i2.771

3.9. Feature selection model

The feature selection outputs are given below. The number of features selected by binary Bat is 10 of the best solutions out of 32 features in the datasets, and the accuracy is over 99.9%, looking at the solutions bat agent. The corresponding "1" is the feature selected. The selected features are used for classification. After 32 minutes, the first iteration was performed using the Saturn cloud to run the algorithm. Saturn Cloud is a cloud-based server that uses AWS, Azure, Databricks, and GCP, with 4 cores and 64 GB RAM. It took 1hr 10 minutes to complete the iteration. Table 2 and Table 3 show first and second iteration results, respectively.

3.10. Classification Stage

The approach of finding a function (model) that explains and separates concepts or classes is known as classification. Classification models are used to identify unknown class labels in new datasets. The ability of the selected subset characteristics to produce good performance accuracy with the classifier algorithms is used to assess their importance. SVM, DT, and RF algorithms are used in this study.

3.11. Evaluation Metrics

The classifiers' performance was assessed using several standard metrics, such as accuracy, precision, and AUC (ROC curve).

Confusion matrix

A confusion matrix assists in illustrating the outcomes of a classification task by presenting a tabular arrangement of the different results of predictions and findings. The matrix includes four values: true positive (TP), false positive (FP), true negative (TN), and false negative (FN).

Table 6. Confusion Matrix

	Predicted Positive	**Predicted Negative**
Actual Positive	TP	FN
Actual Negative	FP	TN

The predicted accuracy of a classifier is computed by dividing the total instances of correctly identified positives and negatives by the overall sample count. TP represents the correct positive predictions, FP denote the incorrect positive predictions, TN signify the accurate negative predictions, and FN indicate the inaccurate negative predictions.

The confusion matrix is employed to assess the performance of a classification model, particularly in scenarios with imbalanced classes. It facilitates the calculation of various evaluation metrics, including accuracy, precision, recall, and F1-score. The accuracy of a classifier is determined by dividing the total correctly classified positive and negative instances by the overall sample count.

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

$$\text{Specificity} = \frac{TN}{TP + FP} \quad (6)$$

$$Sensitivity = \frac{TP}{TP + FN} \quad (7)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (8)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (9)$$

"Recall" comprises the sum of true positives and false negatives, representing instances incorrectly identified as part of the positive class but shouldn't have been. It is computed as the total number of true positives divided by the overall count of components truly belonging to the positive class.

https://doi.org/10.36596/jcse.v5i2.771

Specificity and Sensitivity are both used to describe the genuine negative rate and the false negative rate, respectively.

4. Results and Discussion

This section provides an account of the experimental outcomes of the suggested approach, a crucial aspect in research settings. It shows the output of features selected by Bat at different iterations. Results of different classification algorithms used to implement the model were produced. We provide screenshots of the findings to illustrate the data further.

The column headers in Table 7 and Table 8 respectively denote individual bats within the algorithm, identified by distinct labels (Bat 1, Bat 2, ..., Bat 30). For this specific iteration, the algorithm employed 30 agents, symbolized as bats. The fitness column quantifies the performance of each Bat, reflecting how well its solution addresses the optimization problem. Notably, higher fitness values are indicative of superior solutions. The "Number of Features" column signifies the count of features (or variables) each Bat chooses in its solution. Each row corresponds to a unique bat, presenting its distinctive characteristics, including fitness value and the number of features selected.

Table 7. First iteration

Bat	Fitness	Number of Features
Bat 1	0.9995259997893332	8
Bat 2	0.9995259997893332	7
Bat 3	0.9995259997893332	7
Bat 4	0.9995259997893332	7
Bat 5	0.9995259997893332	9
Bat 6	0.9995259997893332	10
Bat 7	0.9995259997893332	8
Bat 8	0.9995084442259752	9
Bat 9	0.9995084442259752	8
Bat 10	0.9994908886626171	10
Bat 11	0.9994908886626171	10
Bat 12	0.9994908886626171	8
Bat 13	0.9994908886626171	8
Bat 14	0.9994733330992591	9
Bat 15	0.9994733330992591	9
Bat 16	0.9994733330992591	9
Bat 17	0.9994733330992591	7
Bat 18	0.9994733330992591	7
Bat 19	0.9994733330992591	11
Bat 20	0.9994733330992591	11
Bat 21	0.9994733330992591	14
Bat 22	0.9994557775359011	9
Bat 23	0.9994557775359011	10
Bat 24	0.9994382219725431	24
Bat 25	0.999385555282469	6
Bat 26	0.9993504441557529	10
Bat 27	0.9992802219023208	6
Bat 28	0.9992626663389628	7
Bat 29	0.9992626663389628	8
Bat 30	0.9990871107053826	9

Figure 4 represents the best agent's result in the bat algorithm's first iteration. The fitness value of the best agent is 0.9995259997893332. This indicates a high level of performance in the optimization task, as higher fitness values are generally desirable. The best agent selected 8 features in its solution.

https://doi.org/10.36596/jcse.v5i2.771

Number of agents: 30
---------- Best Agent ----------
Fitness: 0.9995259997893332

Agent	Fitness Value	Number of Features
0	1	0.999526	8
1	2	0.999526	7
2	3	0.999526	7

Figure 4. Result of best agent for the first iteration

Table 8. Second iteration

Bat	Fitness	Number of Features
Bat 1	0.9995435553526912	10
Bat 2	0.9995259997893332	8
Bat 3	0.9995259997893332	7
Bat 4	0.9995259997893332	8
Bat 5	0.9995259997893332	9
Bat 6	0.9995259997893332	9
Bat 7	0.9995259997893332	10
Bat 8	0.9995259997893332	9
Bat 9	0.9995259997893332	7
Bat 10	0.9995259997893332	7
Bat 11	0.9995259997893332	7
Bat 12	0.9995259997893332	11
Bat 13	0.9995084442259752	12
Bat 14	0.9994908886626171	10
Bat 15	0.9994908886626171	10
Bat 16	0.9994908886626171	10
Bat 17	0.9994908886626171	7
Bat 18	0.9994733330992591	10
Bat 19	0.9994733330992591	9
Bat 20	0.9994733330992591	11
Bat 21	0.9994733330992591	9
Bat 22	0.9994733330992591	7
Bat 23	0.9994733330992591	9
Bat 24	0.9994733330992591	9
Bat 25	0.9994733330992591	11
Bat 26	0.9994733330992591	9
Bat 27	0.9994557775359011	8
Bat 28	0.9994382219725431	11
Bat 29	0.999420666409185	10
Bat 30	0.9993504441557529	10

Figure 5. shows the result of the best agent for the second iteration run with Python codes by the bats. The fitness value of the best agent is 0.9995435553526912. This indicates a high level of performance in the optimization task. The best agent selected 10 features in its solution.

Number of agents: 30
---------- Best Agent ----------
Fitness: 0.9995435553526912

Agent	Fitness Value	Number of Features
0	1	0.999544	10
1	2	0.999473	9
2	3	0.999456	8

Figure 5. Result of best agent for the Second iteration

https://doi.org/10.36596/jcse.v5i2.771

The solution for the Best Agent after the first iteration and the second iteration is given in the array below.

array([0., 1., 0., 1., 0., 0., 0., 0., 1., 0., 0., 0., 0., 1., 0., 0., 1., 1., 0., 1., 0., 1., 1., 1., 0., 0., 0., 0., 0.])

The number of features selected by the best agent is indicated by one (1) in the array, and the features chosen by the model are shown in Figure 6.

	Time	V2/1	V4/1	V9/1	V14/1	V17/1	V18/1	V20/1	V22/1	V23/1	V24/1	Class
0	0	-0.072781	1.378155	0.363787	-0.311169	0.207971	0.025791	0.251412	0.277838	-0.110474	0.066928	0
1	0	0.266151	0.448154	-0.255425	-0.143772	-0.114805	-0.183361	-0.069083	-0.638672	0.101288	-0.339846	0
2	1	-1.340163	0.379780	-1.514654	-0.165946	1.109969	-0.121359	0.524980	0.771679	0.909412	-0.689281	0
3	1	-0.185226	-0.863291	-1.387024	-0.287924	-0.684093	1.965775	-0.208038	0.005274	-0.190321	-1.175575	0
4	2	0.877737	0.403034	0.817739	-1.119670	-0.237033	-0.038195	0.408542	0.798278	-0.137458	0.141267	0

Figure 6. Feature selected using binary bat algorithm

The dataset was trained, and Figure 7. The display 5_cross validation of the training dataset is below.

```
Train [     0      2      3 ... 284804 284805 284806] validation: [     1     17     21 ... 284785 284792 284799]
Train [     0      1      3 ... 284803 284804 284806] validation: [     2     12     13 ... 284791 284794 284805]
Train [     1      2      3 ... 284803 284804 284805] validation: [     0      6     10 ... 284793 284796 284806]
Train [     0      1      2 ... 284804 284805 284806] validation: [     3      4      9 ... 284797 284800 284803]
Train [     0      1      2 ... 284803 284805 284806] validation: [     5      7      8 ... 284801 284802 284804]
```

Figure 7. Training the Dataset

Model evaluation using the classification algorithms

The performance evaluation metric and confusion matrix of RF, DT, and SVM with feature selection are displayed in Table 9, Table 10, and Table 11 below. Table 9 shows the Confusion Matrix of RF with feature selection.

Table 9. Confusion Matrix of RF with Feature Selection

	Predicted_Fraudlent	Predicted_not Fraudulent
is_Fraudlent	78	30
is_ not Fraudulent	0	56853

The AUC (ROC Curve) shows the performance of the classifier algorithm. The closer to one (1) of the AUC test value, the better its accuracy. Figure 8 illustrates the ROC Curve RF.

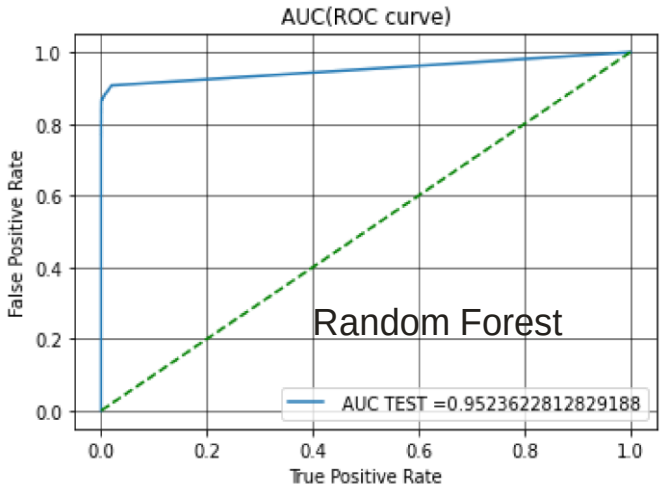

Figure 8. ROC curve for Random Forest

https://doi.org/10.36596/jcse.v5i2.771

Table 10. The Confusion Matrix of DT with feature selection.

	Predicted_Fraudlent	Predicted_not_Fraudulent
is_Fraudlent	76	32
is_ not Fraudlent	20	56833

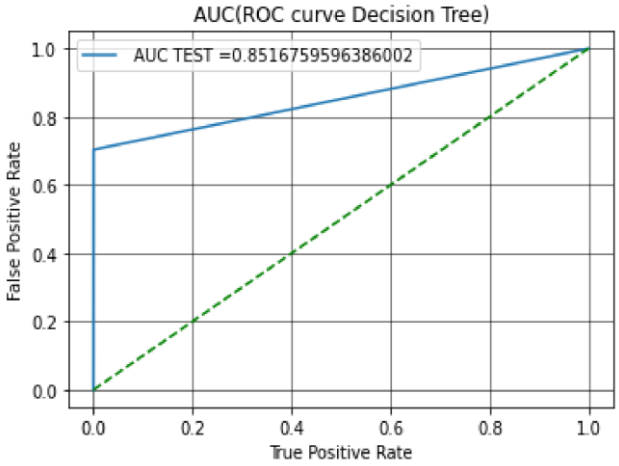

Figure 9. ROC Curve for Decision Tree

Table 11. The Confusion Matrix of SVM with Feature Selection.

	Predicted_Fraudlent	Predicted_not Fraudulent
is_Fraudlent	33	75
is_ not Fraudulent	12	56841

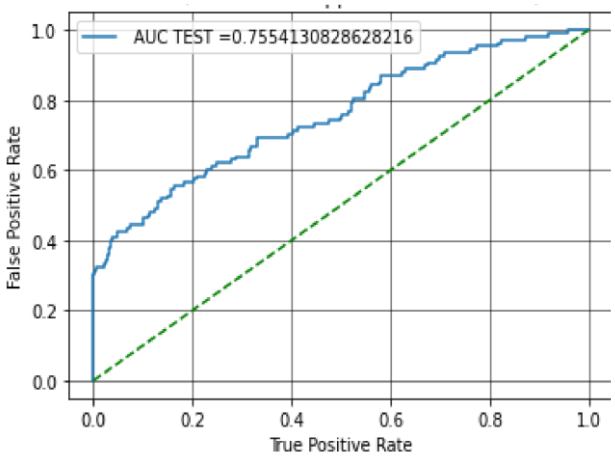

Figure 10. ROC Curve for Support Vector Machine

The summary of the performance evaluation of the Decision Tree, Random Forest, and Support Vector Machine using MBBA as feature selection are shown in Table 12.

Table 12. DT, RF, and SVM Performance Measures using MBBA as Feature Selection.

Performance Metrics	DT (%)	RF(%)	SVM (%)
Accuracy	99.91	99.95	99.85
Precision	90.00	100.00	87.00
F1-score	87.00	92.00	72.00
Recall	85.00	86.00	65.00

https://doi.org/10.36596/jcse.v5i2.771

Comparing the Study Model with the Existing Model

Table 4.9 shows the result that works best in the existing work by Saheed, Hambali, and Olasupo (2022) using a Genetic Algorithm for feature selection.

Table 13. Performance Measures of the First Priority Features using GA as Feature Selection from Existing Technique (Saheed, Hambali, and Olasupo, 2022)

Performance Metrics	Naïve Bayes	RF	SVM
Precision	94. 7	96.5	92.7
Accuracy	94.3	96.4	96.3

It was reported in their study that out of the two priority features evaluated with a classifier algorithm using the GA as feature selection, the priority gave the best result in terms of performance accuracy, precision, and Sensitivity. Also, the RF classifier performed better than other classifier algorithms used with accuracy (96.40%) and precision (96.5%).In the proposed study using MBBA as feature selection when evaluated with a classifier algorithm, RF outperformed the other classifiers with an accuracy of 99.947%.

Overall Discussion

The experiment's output indicates that the MBBA performs better when compared with GA using the evaluation metric, and of all the classifiers used, random forest works best. It is necessary to reflect that the research model has enhanced the performance of the classifier algorithm.

5. Conclusion

Detecting credit card fraud effectively necessitates the selection of pertinent features for detection, ensuring the efficient operation of the classification algorithm. This research introduces an MBBA for feature selection to improve the performance of the classifier algorithm in credit card fraud detection. Like any other bio-inspiring algorithm, the bat algorithm mimics a bat's behavior pattern, which uses eco-location to catch its prey. The classifier algorithms used are the RF, SVM, and Decision Tree. The RF works best when compared with others, with an accuracy of 99.947%. The result was further compared with the existing study where GA was used as feature selection, and the RF performed with an accuracy of 96.40% as the best classifier in their research. In future work, this study can be improved by using two or more datasets with two or more bio-inspired algorithms for feature selection and then comparing their performance.

References

[1] V. N. Dornadula and S. Geetha, “Credit Card Fraud Detection using Machine Learning Algorithms,” Procedia Computer Science, vol. 165. Elsevier BV, pp. 631–641, 2019. doi: 10.1016/j.procs.2020.01.057.

[2] R. J. Bolton and D. J. Hand, “Statistical Fraud Detection: A Review,” Statistical Science, vol. 17, no. 3. Institute of Mathematical Statistics, Aug. 01, 2002. doi: 10.1214/ss/1042727940.

[3] B. Andhavarapu, S. V. K. Ratna, P. Jyothi, S. G. Varun, and S. R.Rohith. (2020, February). "Credit card fraud detection using Machine learning algorithms". Quest Journals. vol. 8, issue 2, pp. 4-11.

[4] Y. Sahin, and E. Duman, "Detecting Credit Card Fraud by Decision Trees and Support Vector Machines", Proceedings of International Multi-Conference of Engineers and Computer Scientists (IMECS 2011), vol. 1, pp. 1-6, Mar. 16-18 2011, ISSN 2078-0966, ISBN 978-988-18210-3-4.

[5] S. P. Maniraj, S. Aditya , D. S. Swarna, and A. Shadab. (2019, September). "Credit Card Fraud Detection using Machine Learning and Data Science". vol. *8, issue* 9, pp. 1-5. doi: 10.17577/IJERTV8IS090031.

[6] M. A. Hambali, T. O. Oladele, K. S. Adewole, A. K. Sangaiah, and W. Gao, “Feature selection and computational optimization in high-dimensional microarray cancer datasets via InfoGain-modified bat algorithm,” Multimedia Tools and Applications, vol. 81, no. 25. Springer Science and Business Media LLC, pp. 36505–36549, Aug. 11, 2022. doi: 10.1007/s11042-022-13532-5.

https://doi.org/10.36596/jcse.v5i2.771

[7] Z. Li, M. Huang, G. Liu, and C. Jiang, “A hybrid method with dynamic weighted entropy for handling the problem of class imbalance with overlap in credit card fraud detection,” Expert Systems with Applications, vol. 175. Elsevier BV, p. 114750, Aug. 2021. doi: 10.1016/j.eswa.2021.114750.
[8] J. Forough and S. Momtazi, “Ensemble of deep sequential models for credit card fraud detection,” Applied Soft Computing, vol. 99. Elsevier BV, p. 106883, Feb. 2021. doi: 10.1016/j.asoc.2020.106883.
[9] SamanehSorournejad, Z. Zojaji, R. E. Atani, and A. H. Monadjemi, “A Survey of Credit Card Fraud Detection Techniques: Data and Technique Oriented Perspective.” arXiv, 2016. doi: 10.48550/ARXIV.1611.06439.
[10] A. Harshita, G.Richa, and C. Raman. "Credit Card Fraud Detection Using Machine Learning". EasyChair Preprint, pp.1-8. 2021.
[11] M. S. Tahsin, N. Firoz, S. A. H. Chowdhury, S. M. A. Wasei, M. A. Karim and M. Y. Ara, "Investigation of Credit Card Fraud detection under the lens of Comparative Machine Learning models," 2022 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), Sakheer, Bahrain, 2022, pp. 25-31, doi: 10.1109/3ICT56508.2022.9990897.
[12] S.Nikfar , and B. Touraj. (2018). Detection of the Suspicious Transactions by Integrating the Neural Network and Bat Algorithm. *Specialty Journal of Electronic and Computer Sciences, vol. 4, issue*1, pp. 9-19.
[13] Y. K. Saheed, M. A. Hambali, M. O. Arowolo and Y. A. Olasupo, "Application of GA Feature Selection on Naive Bayes, Random Forest and SVM for Credit Card Fraud Detection," 2020 International Conference on Decision Aid Sciences and Application (DASA), Sakheer, Bahrain, 2020, pp. 1091-1097, doi: 10.1109/DASA51403.2020.9317228.
[14] Doreswamy and U. S. M, “A Binary Bat Inspired Algorithm for the Classification of Breast Cancer Data,” International Journal on Soft Computing, Artificial Intelligence and Applications, vol. 5, no. 2/3. Academy and Industry Research Collaboration Center (AIRCC), pp. 01–21, Aug. 30, 2016. doi: 10.5121/ijscai.2016.5301.
[15] A. A. Taha and S. J. Malebary, "An Intelligent Approach to Credit Card Fraud Detection Using an Optimized Light Gradient Boosting Machine," in IEEE Access, vol. 8, pp. 25579-25587, 2020, doi: 10.1109/ACCESS.2020.2971354.
[16] D. Cheng, S. Xiang, C. Shang, Y. Zhang, F. Yang, and L. Zhang, “Spatio-Temporal Attention-Based Neural Network for Credit Card Fraud Detection,” Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 01. Association for the Advancement of Artificial Intelligence (AAAI), pp. 362–369, Apr. 03, 2020. doi: 10.1609/aaai.v34i01.5371.
[17] F. Carcillo, Y.-A. Le Borgne, O. Caelen, Y. Kessaci, F. Oblé, and G. Bontempi, “Combining unsupervised and supervised learning in credit card fraud detection,” Information Sciences, vol. 557. Elsevier BV, pp. 317–331, May 2021. doi: 10.1016/j.ins.2019.05.042.
[18] R. Y. M. Nakamura, L. A. M. Pereira, K. A. Costa, D. Rodrigues, J. P. Papa and X. . -S. Yang, "BBA: A Binary Bat Algorithm for Feature Selection," 2012 25th SIBGRAPI Conference on Graphics, Patterns and Images, Ouro Preto, Brazil, 2012, pp. 291-297, doi: 10.1109/SIBGRAPI.2012.47.

https://doi.org/10.36596/jcse.v5i2.771