International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

A Hybrid Grey Wolf - Meta Heuristic Optimization and Random Forest Classifier for Handling Imbalanced Credit Card Fraud Data

[1]V. Uma Rani, [2]V Saravanan, [3]J. Jebamalar Tamilselvi

Submitted: 28/04/2023 **Revised**: 26/06/2023 **Accepted**: 06/07/2023

Abstract: Because of COVID 19 and the development of information technology, individuals prefer to frequently purchase online for necessities and pay with credit cards. In these online digital transactions, credit card fraud is one of the main problems that result in financial loss for customers. The identification of such online credit card fraud has been the subject of numerous studies. To automate this process of detecting credit card fraud, a number of machine learning and data mining approaches have been developed. This study presents a Hybrid Grey Wolf optimization approach and Random Forest classifier (HGWRF) with three sequence levels for detecting credit card fraud. In the first level, a credit card data set is collected and balanced using a combined SMOTE ENN sampling technique. Grey wolf meta heuristic approach is used in the second level to optimize the subset of features. The Random Forest machine learning classifier is employed in the third level to learn the model for the credit card fraudulent detection system. It assesses performance using basic metrics and MCC, CV score, R2 score, MSE, kappa score. The suggested HGWRF improves accuracy by 0.87 to 0.946 and outperforms well when compared with other non-optimization machine learning algorithms.

Keywords: *Credit Card fraud, Grey Wolf meta heuristic, random Forest, SMOTE ENN*

1. Introduction

Because of technological advancements, Credit Cards (CC) have become a convenient means for people to purchase necessities both online and offline. CC fraud occurs as a result of the rapid increase of online transactions and enhanced cybercrime technology. According to the RBI annual report of India, card and internet-linked fraud grew to Rs 155 crore at the end of the fiscal year March 2022. According to insider information, overall credit card loss in the United States is $12 billion, with a projected increase to $12.6 billion by the end of 2022. This is primarily because of the pandemic-induced shift to eCommerce.

CC fraud affects the people in all around the world and financial organizations. The illicit withdrawal of monies from a cardholder's account without the cardholder's knowledge is referred to as CC fraud. To detect such misused transactions, data from the service provider's database such as transaction date, receiver, and transaction amount [2] are used. Several security solutions have been developed to prevent such kind of credit card theft at the application level. Machine Learning (ML), a low-cost automated technique for detecting credit card fraud in online purchases by analysing transaction records [3]. ML techniques are now being utilised to analyse real-time data in sectors such as flood prediction, software failure prediction, bio informatics, air quality prediction, medical diagnosis, and food quality prediction and others. This ML classifier's accuracy and its performance are hampered by class imbalance in acquired real-time data sets [4]. Class imbalance is one of the main problem in binary and multi class classification of datasets, caused by a large number of class negative samples (C1) and a small number of positive samples (C2).

Researchers have devised a number of techniques to address the problem of class imbalance. These approaches are applied at the data level, the algorithmic level, and the Ensemble or cost sensitive level of learning. Training examples are transformed into a balanced class distribution through data level sampling, allowing the classifier to predict. Classifiers are adjusted at the algorithmic level to address issues of class imbalance. To reduce misclassification rates at the cost sensitive level, combine both data and algorithmic levels. Here, data level sampling is simple and easy to use with an existing classifier. The data level sampling methods are further classified as under sampling, oversampling, and hybrid sampling. To construct a balanced data set, eliminate the C1 samples under sampling methods until all classes reach a specified level. In oversampling, generate some C2 samples until all classes reach a specified level. Hybrid sampling combines under and over sampling approaches. However, these strategies make learning more difficult,

[1]*Associate Professor, Saveetha Engineering College, Thandalam, Chennai*
Email: umaranibharathy@gmail.com
[2] *Professor, Department of Computer Science and Applications*
SRM Institute of Science and Technology, Ramapuram Campus, Chennai 60089, Tamilnadu, India
Email: saravanv10@srmist.edu.in
[3]*Associate Professor, Department of Computer Science and Applications*
SRM Institute of Science and Technology, Ramapuram Campus, Chennai 60089, Tamilnadu, India
Email: jebamalj@srmist.edu.in

resulting in poor performance and overfitting [5,6]. To address this issue, several oversampling and under sampling strategies have been developed. However, they are insufficient to improve the classifier's accuracy.

Meta heuristics are a prominent technique for tackling optimization and classification problems that are widely employed in business, engineering, commerce, and many other industries. They include the concepts of biological evolution, intelligent problem solving, mathematics and physical sciences, the nervous system, and statistical mechanics [7]. Several prominent meta heuristic algorithms have been created to determine the best solution. Among these, Grey wolf optimization (GWF) is a well-known swarm intelligence meta heuristic method based on the behaviour of grey wolves that was created in 2014 by Seyedali Mirjalili et al[8,9]. GWF is employed in this study to pick the best subset of features for detecting CC fraud.

This paper's key contribution is as follows:

i) It handles the class imbalance problem using a hybrid sampling technique known as SMOTE ENN. The SMOTE ENN combines the advantages of the SMOTE oversampling approach and the Edited Nearest Neighbour sampling technique (ENN).

ii) It improves the process of selecting a subset of features using the Grey Wolf optimization technique. The GWO aids in the optimization of the subset of features produced by data sampling methods.

iii) It boosts classifier accuracy and other performance measures. The comparison analysis demonstrates that the optimization strategy, when combined with sampling, outperforms the non-optimization technique.

2. Literature Survey

Initial research focused on machine learning algorithms[10,11] for credit card fraud detection, but did not address the issue of imbalanced classification. Several under sampling, oversampling, and hybrid sampling methods have been developed to address this issue, however they are insufficient to boost accuracy. So some work will be done utilizing a meta-heuristic approach to optimize feature selection and increase classification accuracy. This section describes some of the existing work toward detecting CC fraud.

Chun-Yang Peng et al [12] have created a hybrid sampling approach that combines the DBSCANN, BNF, and OBN methods. It first finds borderline noise samples using BNF, then utilizes OBN to discover outlier samples and DBSCANN to cluster the samples. They employ SVM for binary classification of 16 different datasets and evaluate classifier performance using Gmean and AUC scores. V. Cerqueira et al [13] have used a layered learning strategy for dealing with imbalances in a 100-dataset benchmark. It builds a layer using an agglomerative clustering approach and divides it into three groups: pure majority, pure minority, and mixed. Based on their findings, LL+SMOTE outperforms other sampling approaches such as CURE and Balanced RF, Random Under-sampling and Oversampling.

To create samples, Wei Wei et al [14] have devised a weighted complexity process (WCP) based on sampling approach. They filter the generated majority samples based on their weighted complexity and choose the best majority samples for balanced data set construction. They use CART and KNN for classification, and their performance is measured by AUC, accuracy, and F1-measure.

For prediction, Akira Tanimoto et al [15] have used cost-sensitive learning and stratified sampling. They tweak the baseline logistic regression and SVC by utilizing near miss positive instances to balance the dataset. They compare the GPU Kernal performance data set with 12 distinct datasets from the UCI repository. According to their findings, the proposed technique obtained greater than 90% balanced accuracy.

Jun Wang et al [16] have proposed LDL with Class shared and Class specific Knowledge for multi-class autism spectrum disorder (NYU dataset) classification. They employ SMOTE to deal with class imbalances and the Augmented Lagrange method to find the best solution. To test the performance of the classifier, they employ the assessment parameters ChebyShev, Cosine, Clark, Canberra, Intersection, Kullbeck-Leibler, and MAP.

Piyush Bhardwaj et al [17] created a machine learning system based on SMOTE for prediction. They forecast the performance of the classifier in terms of precision, recall, F1-score, and ROC curve. They utilise seven classifiers for classification: XGB, RFC, GNB, Adaboost, SGD, SVC, DTC, and KNN. Adaboost, according to their survey, provides good accuracy when compared to other models.

Venkata Krishnaveni Chennuru et al. [18] employed a SA-based under-sampling strategy to balance the dataset and obtain sensitivity measures ranging from 0.68 to 0.86. Bharat Kumar Padhi et al. [19] created a Rock Hyrax Swarm optimization technique for optimal relevant feature selection from imbalanced high dimensional Europian credit card fradulant data set in 2013. They use NBC, DTC, SVM, and KNN for classification.

Nilesh Kunhare et al. [20] developed a genetic method for optimal feature selection in an NSL-KDD dataset. For classification, they employ the Greywolf optimization algorithm as well as LR and DT. Ameer Tamoor Khan et al. [21] created a Beetle Antennae Search method to detect ambiguity in corporate financial fraud. They use data from the Securities and Exchange Commission's Accounting and Auditing Enforcement releases. Aleksandar Petrovic et al. [18] created an SMOTE to help balance the credit card fraud data set. For classification, they use the firefly meta heuristic and the adaboost classifier.

Jiaju Tang et al.[22] used artificial ecosystem-based optimization and a self-organizing RBF neural network to detect credit card fraud. Ehram Safari et al [23] employed fisher discriminant analysis for feature selection. They train with MLP centred on teaching and learning. AdaBoost, Random Forest, CNN, and RNN are some of the most recent algorithms accessible. Ajeet singh et al [24] use the firefly algorithm for optimization and SVC for classification. It provided 85.65% accuracy and classified 591 transactions successfully. For testing, they use an Australian data set. G.K Arun et al [25] suggested a binary emperor penguin optimization with GRU model for detecting credit card fraud in a German credit card dataset. This analysis provides an accuracy of up to 90.78%. Singh Yadav [26] et al have used a Manta Ray Foraging optimization algorithm for feature selection in financial statement fraud detection. They used convolution neural network for learning and adaptive density based clustering algorithm for labelling the selected features through clustering.

From this analysis, meta heuristic algorithm helps to optimize the feature selection and improve the accuracy of system.

3. Proposed Methodology

The Hybrid Grey Wolf Random Forest (HGWRF) approach is created to detect credit card fraud. The overall flow of HGWRF for credit card fraud prediction is depicted in Figure 1. It is divided into three stages. In the first stage, the data set is pre-processed and balanced using the hybrid sampling approach SMOTE ENN. In the second stage, use the Grey Wolf Meta heuristic technique to select an optimum subset of features. Finally, the Random Forest machine learning classifier is utilized to predict credit card fraud.

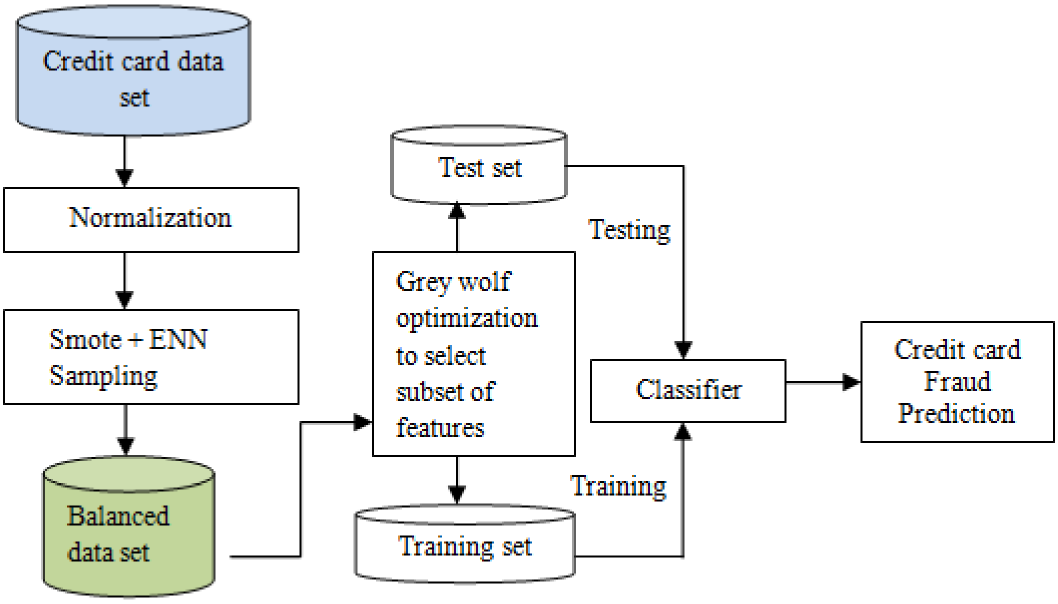


Fig 1. Flow of HGWRF for credit card fraud detection

3.1. Pre-processing: The initial stage in cleaning and organizing data is pre-processing. Credit card fraud data is sourced from University of California Irvine machine learning repository. The null and missing values are initially substituted with -1, and the features are normalized using a conventional scalar technique.

TABLE 1: Abbreviations used

Abbreviations	Expansion
SMOTE	Synthetic Minority Over-sampling Technique
ADASYN	Adaptive Synthetic oversampling technique

SMOTETomek	Synthetic Minority Over-sampling Technique
SMOTEENN	Synthetic Minority Over-sampling Technique and Edited Nearest Neighbours
GWO	Grey Wolf Optimization
DTC	Decision tree
RFC	Random Forest
LRC	Logistic Regression
KNN	K Nearest Neighbour
SVC	Support Vector Machine
C1	Majority class
C2	Minority Class
BNF	Border Line Noise Factor
LDL	Label Distribution learning
OBN	Outliers Based on Neighbours

3.2 SMOTE ENN

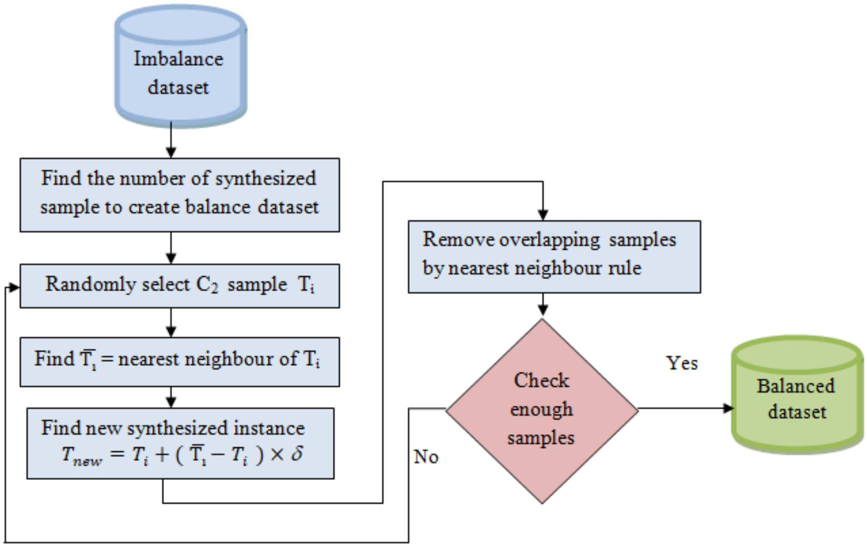
SMOTE ENN is a data level sampling approach that combines the advantages of oversampling and under-sampling. Synthetic Minority Oversampling Approach (SMOTE) is an oversampling technique that uses C2 samples to rebalance a given data set. To begin rebalancing, SMOTE computes the number of replicated samples (N) produced by the equation (1)

$$N = (T_{maj} - T_{min}) \times \beta \tag{1}$$

Where T_{maj} is number of C_1 samples and T_{min} is the number of C_2 samples in a dataset. Second, it randomly selects a positive instance (k) from the dataset. Then, using a distance metric, it finds the class's nearest neighbour (5 by default). Finally, N k instances are picked at random to generate new instances via interpolation. The new synthesized sample T_{new} is derived from random C_2 sample T_i. SMOTE generates a random sample C_2 every time by determining the k – nearest neighbour T_i and using the equation (2). Here δ is the random value is in the range [0,1].

$$T_{new} = T_i + (\overline{T}_i - T_i) \times \delta \tag{2}$$

Following SMOTE’s production of new set of samples, overlapping samples are deleted by using ENN nearest neighbour rule. This technique is performed until the dataset has reached an equilibrium state. This is depicted in figure 2.



Imbalance dataset
Find the number of synthesized sample to create balance dataset
Randomly select C_2 sample T_i
Find $\overline{T}_i$ = nearest neighbour of T_i
Find new synthesized instance $T_{new} = T_i + (\overline{T}_i - T_i) \times \delta$
Remove overlapping samples by nearest neighbour rule
Check enough samples
Yes
No
Balanced dataset

Fig 2. Flow chart for SMOTEENN algorithm

3.3 Grey wolf optimization technique

GWO is a Meta heuristic technique developed by Seyedali Mirjalili et al in 2014[27]. GWO imitates Grey Wolf's social leadership and hunting behaviour. From high to low level, the social hierarchy is constituted by four types of wolves: alpha, beta, omega, and delta. GWO optimization behaviour is formalised by wolf hunting behaviour to tracking, chasing, approaching, pursuing, harassing, encircling, and attacking prey. The GWO search began with a collection of candidate populations.

Based on the fitness value of candidate population, alpha, beta, omega are initialized. Let α is the fittest solution, β is the second best solution and δ is the third best solution and ω is the rest of candidate solutions. For each iteration, the new position of α, β, δ is calculated based on current location and prey location by the following equation (3) and (4).

$$\vec{D} = \left|\vec{C} \times \vec{X}_p - \vec{X}(t)\right| \qquad (3)$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \times \vec{D} \qquad (4)$$

Here t specifies the current iteration, $\vec{A}$ and $\vec{C}$ are coefficient vectors, $\vec{X}_p$ is the position vector of prey. $\vec{X}$ is the position vector of grey wolf. The coefficient vectors are calculated by equation (5) and (6).

$$\vec{A} = 2\vec{a} \times \vec{r_1} - \vec{a} \qquad (5)$$

$$\vec{C} = 2 \times \vec{r_2} \qquad (6)$$

Here $\vec{r_1}$ and $\vec{r_2}$ are random vectors in [0,1] and components of control parameter $\vec{a}$ are linearly decreased from 2 to low value 0. $\vec{A}$ is the random value between [-a,a]. $\vec{C}$ is the random value between [0,2]. The candidate population is diverged when $\vec{A} > 1$ and it is converged when $\vec{A} < 1$. The overall searching process of GWO for feature selection is shown in figure 3.

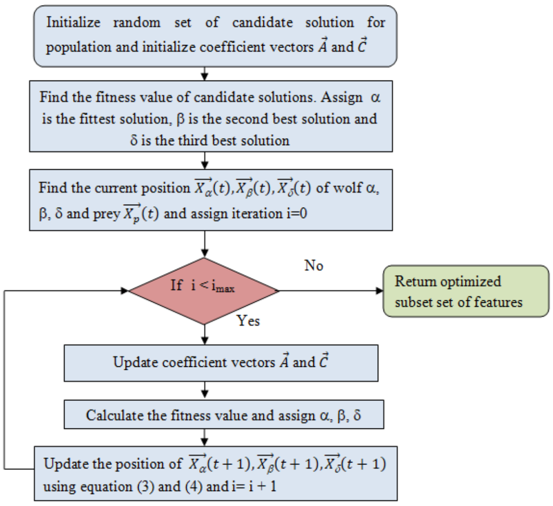

Initialize random set of candidate solution for population and initialize coefficient vectors $\vec{A}$ and $\vec{C}$
Find the fitness value of candidate solutions. Assign α is the fittest solution, β is the second best solution and δ is the third best solution
Find the current position $\vec{X_\alpha}(t), \vec{X_\beta}(t), \vec{X_\delta}(t)$ of wolf α, β, δ and prey $\vec{X_p}(t)$ and assign iteration i=0
If i < i_{max}
No
Yes
Return optimized subset set of features
Update coefficient vectors $\vec{A}$ and $\vec{C}$
Calculate the fitness value and assign α, β, δ
Update the position of $\vec{X_\alpha}(t+1), \vec{X_\beta}(t+1), \vec{X_\delta}(t+1)$ using equation (3) and (4) and i= i + 1

Fig 3. Flow chart for GWO algorithm

3.4 Random Forest Classifier

RFC is a collection of different decision trees that use different subsets and average the results. The average result helps to reduce model errors for better prediction and control over fitting. The bagging method is used to teach RFC to control the ensemble. It adds randomization and seeks the biggest characteristic among a subset of features, resulting in more diversity and a better model.

The proposed Hybrid Grey Wolf Random Forest algorithm is described in algorithm1.

Algorithm1: Hybrid Grey Wolf Random Forest (HGWRF)

Input : Unbalanced Credit Card fraud dataset (UCC)

Ouput : Prediction of CC transactions

Step 1: Apply pre processing techniques and standard scalar method to UCC.

Step 2: Find number of synthesized sample needed to create balanced CC data set.

```
Step 3: for each  randomly selected sample $T_i$  from $C_2$ class
            Create  new synthesized sample using  nearest neighbour of $T_i$
Step 4 :  Remove overlapping samples using nearest neighbour rule
Step 5 : Repeat step 3 and 4 until balanced CC data set obtained
Step 6: Split the data set into training data set and test data set
Step 7:  Apply GWO to optimize subset of features in training data set
Step 8: Apply random forest classifier to train the optimized data
Step 9: Apply  HGWRF to predict the test data
Step 10:  return  the predicted CC transaction
```

4. Experimental Results and Discussion

The experiment for credit card fraud detection is carried out on Windows 10 with Intel CORE i3 processor. The google Collaboratory, necessary packages and sklearn for classification. The imbalanced-learn package has been used for sampling and sklearn Niapy package has been used for meta heuristic algorithm grey wolf optimization. The result discussion is divided into four subsections. The first subsection describes the data set and evaluation metric used for experimental analysis. The second subsection choose the best classifier RFC based on evaluation. The third subsection shows the performance of RFC after sampling. The sampling process is carried out by oversampling method SMOTE, SMOTE Tomek, ADASYN and combined sampling approach SMOTEENN. The fourth subsection discuss about the proposed HGWRF and non-optimization techniques.

4.1 Data set and Evaluation metrics:

The data set is collected from UCI repositories and CC dataset contain 28382 samples with 21 features having majority classes 23122 and minority classes 5260. The outline description of credit card fraud data set is shown in Table1. From this 21 features, the statistical summary of 17 features are shown in Table 2.

Table 1: Outline of CC Dataset

Total Number of samples	Majority samples (Class C_1)	Minority samples (Class C_2)	No of missing values	Total no of Features	Features taken for classification
28382	23122	5260	525	21	17

Table 2. Statistical summary of numerical features in CC data set

Feature	Name of the feature	Mean	Std	Min	25%	50%	75%	Max
F1	customer_id	15143.5	8746.45	1.0	7557.25	15150.50	22706.7	30301.0
F2	Vintage	2091.14	272.6	73.0	1958.0	2154.0	2292.0	2476.0
F3	Age	48.20	17.807	1.0	36.0	46.0	60.0	90.0
F4	Dependents	0.347	0.997	0.0	0.0	0.0	0.0	52.0
F5	City	796.10	432.87	0.0	409	834	1096	1649
F6	customer_nw_category	2.225530	0.660443	1.0	2.0	2.0	3.0	3.0
F7	branch_code	925.9	937.7	1.0	176.0	572.0	1440.0	4782.0
F8	current_balance	7.380552e+03	4.259871e+04	-5.503960e+03	1.784470e+03	3.281255e+03	6.635820e+03	5.905904e+06

F9	previous_month_end_balance	7.495771e+03	4.252935e+04	-3.149570e+03	1.906000e+03	3.379915e+03	6.656535e+03	5.740439e+06
F10	average_monthly_balance_prevQ	7.496780e+03	4.172622e+04	1.428690e+03	2.180945e+03	3.542865e+03	6.666887e+03	5.700290e+06
F11	average_monthly_balance_prevQ2	7.124209e+03	4.457581e+04	-1.650610e+04	1.832507e+03	3.359600e+03	6.517960e+03	5.010170e+06
F12	current_month_credit	3.433252e+03	7.707145e+04	1.000000e-02	3.100000e-01	6.100000e-01	7.072725e+02	1.226985e+07
F13	previous_month_credit	3.261694e+03	2.968889e+04	1.000000e-02	3.300000e-01	6.300000e-01	7.492350e+02	2.361808e+06
F14	current_month_debit	3.658745e+03	5.198542e+04	1.000000e-02	4.100000e-01	9.193000e+01	1.360435e+03	7.637857e+06
F15	previous_month_debit	3.339761e+03	2.430111e+04	1.000000e-02	4.100000e-01	1.099600e+02	1.357553e+03	1.414168e+06
F16	current_month_balance	7.451133e+03	4.203394e+04	-3.374180e+03	1.996765e+03	3.447995e+03	6.667958e+03	5.778185e+06
F17	previous_month_balance	7.495177e+03	4.243198e+04	-5.171920e+03	2.074407e+03	3.465235e+03	6.654693e+03	5.720144e+06

The following ten different evaluation metrics are used to test the performance of classifiers.

1. Precision: Precision specifies the proportion of true positive CC fraud samples from the total C2 samples in a dataset.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (7)$$

2. Recall: Recall specifies the proportion of the true positive rate of CC fraud samples.

$$Recall = \frac{True\ Positive\ \ (TP)}{True\ Positive(TP)\ + False\ Negative(TN)} \quad (8)$$

3. F1-Score: The F1 score is evaluated from harmonic mean of recall and precision.

$$F1\ score = \frac{2 \times (Recall\ \ \times Precision)}{Recall + Precision} \quad (9)$$

4. Accuracy: The accuracy specifies the number of correctly classified CC samples from the total samples in a dataset.

$$Accuracy = \frac{(TP + TN)}{(TP + TN\ \ + FP\ \ + FN)} \quad (10)$$

5. ROC AUC score: It specifies the degree of separability between the prediction of positive and negative CC samples which varies from 0 to 1.
6. Matthews correlation coefficient (MCC) is the balanced measure to assess the quality of classification which varies from -1 to +1. MCC is calculated by

$$MCC = \frac{\mathrm{TP} \times \mathrm{TN} - \mathrm{FP} \times \mathrm{FN}}{\sqrt{(TP + FP) \times (TP + FN) \times (TN \times FP)(TN \times FN)}} \tag{11}$$

7. Cross Validation mean score (CV) is the validation cc fraud detection dataset score.
8. R2 score is the measure of specifying how well the model is fitted for detecting CC fraud.
9. MSE is the Mean Squared Error value between the predicted output and actual output for CC fraud data set.
10. Cohen's Kappa Score is the difference between observed accuracy of CC fraud detection model and the overall accuracy of CC fraud detection model.

4.3. Classification without sampling and Meta heuristic algorithm:

Initially the CC data set is pre-processed and removes null values and missing value by maximum occurrence value. The standard scalar mechanism has been used to normalize the data. The data set is split into training set (80% of data) and test data set (20% of data). The classification is carried out by three most widely used classifier LRC, DTC and RFC. The performance of classifier is shown in table 2.

Table 2. The performance analysis summary of classifier without sampling and meta heuristic algorithm

Classifier	Class	Precision	Recall	F1-score	Accuracy	AUC-ROC	MCC	CV	R2 score	MSE	Kappa
LRC	0	0.83	0.99	0.90	0.82	0.77	0.208	78.76	-0.16	0.18	0.12
	1	0.75	**0.08**	**0.15**							
DTC	0	0.88	0.95	0.91	0.86	0.80	0.46	80.3	0.04	0.14	0.45
	1	0.67	**0.44**	**0.5**							
RFC	0	0.88	0.96	0.92	0.87	0.83	0.49	83.59	0.12	0.13	0.48
	1	0.74	**0.44**	**0.55**							

From this summary, RFC provide accuracy 87% and AUC score 83% for C1 class prediction which is better than LRC and DTC. But it provide 74% precision, 44% recall, 55% F1 score and less MSE 13% and good Kappa score - 48% , MCC- 49% for C2 class. From this analysis, Classifier performance is low for C2 class prediction. This is depicted in Figure.4.

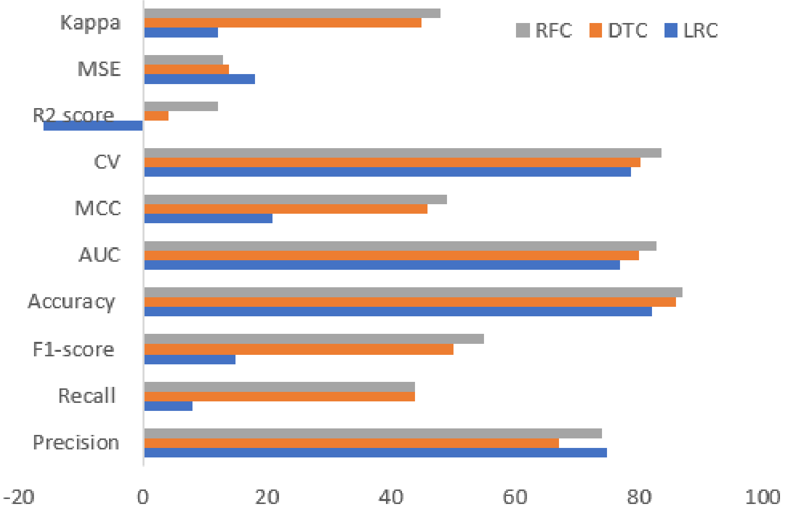


Figure 4. Performance analysis of RFC, DTC and LRC for CC dataset

4.4 Classification with sampling algorithm:

Initial CC data set is unbalanced one because it has 81.5% C_1 classes and 18.5 % C_2 classes. The data distribution of each class is shown in Figure 5.

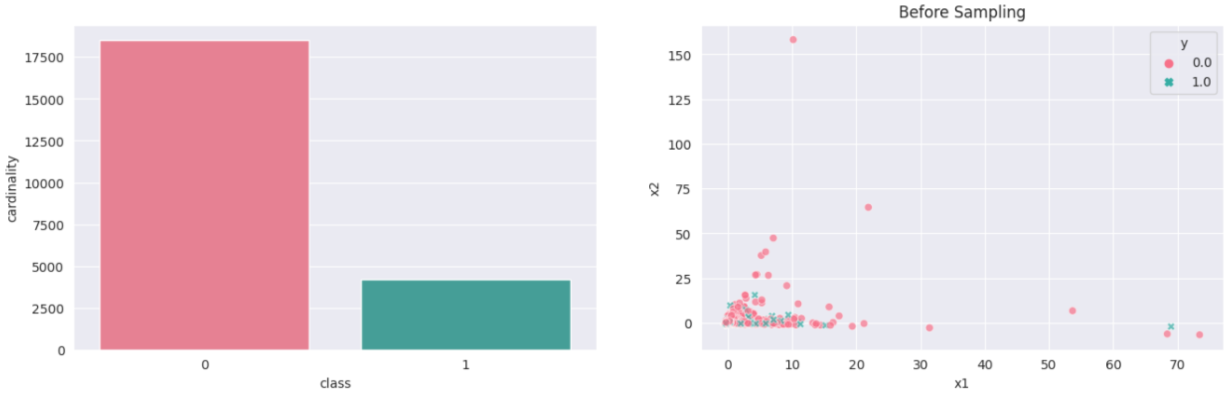


Fig 5. The unbalanced class distribution in CC data set.

The sampling algorithm has used to balance the data set before classification. The figure 6 shows the balanced class distribution after sampling. Now the CC data set have 17500 C1 and 17500 C2 classes.

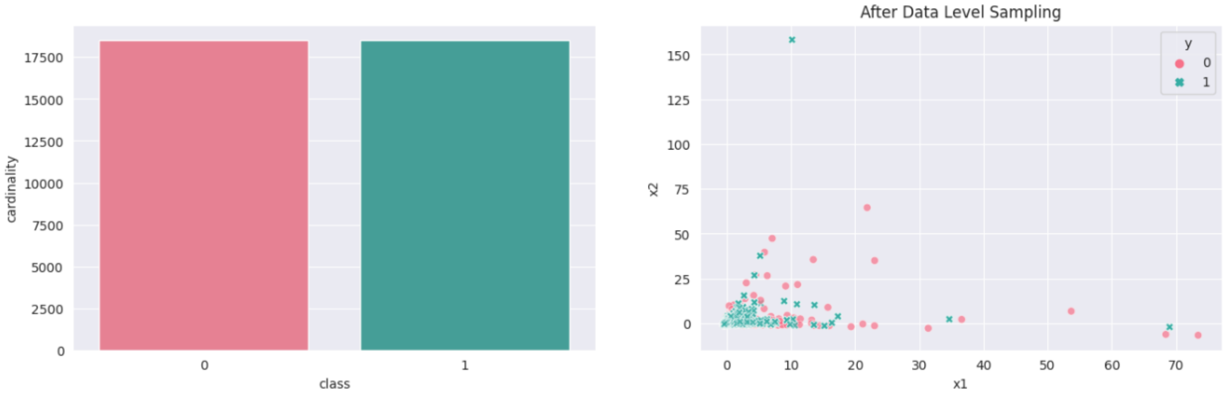


Fig 6. Balanced class distribution in CC data set after sampling algorithm.

The popular sampling technique SMOTE, SMOTE Tomek, ADASYN and combined sampling SMOTEENN are helps to balance the data set before training starts. The classifier LRC, DTC and RFC are tested and the summary of each classifier is specified by Table 3,4 and 5.

Table 3. Performance analysis summary of LRC with various sampling

LRC with Sampling	Class	Precision	Recall	F1-score	Accuracy	AUC-ROC	MCC	CV	R2 score	MSE	Kappa
SMOTE	0	0.92	0.81	0.86	0.78	0.78	0.42	78.7	-0.44	0.22	0.41
	1	0.45	**0.69**	**0.54**							
ADASYN	0	0.92	0.85	0.86	0.81	0.778	0.45	78.76	-0.25	0.19	0.45
	1	0.49	**0.66**	**0.55**							
SMOTE-TOMEK	0	0.92	0.81	0.86	0.78	0.778	0.42	78.76	-0.43	0.22	0.41
	1	0.45	**0.68**	**0.54**							
SMOTE_ENN	0	0.92	0.81	0.88	0.81	0.787	0.41	78.76	-0.16	0.18	0.46
	1	0.48	**0.69**	**0.56**							

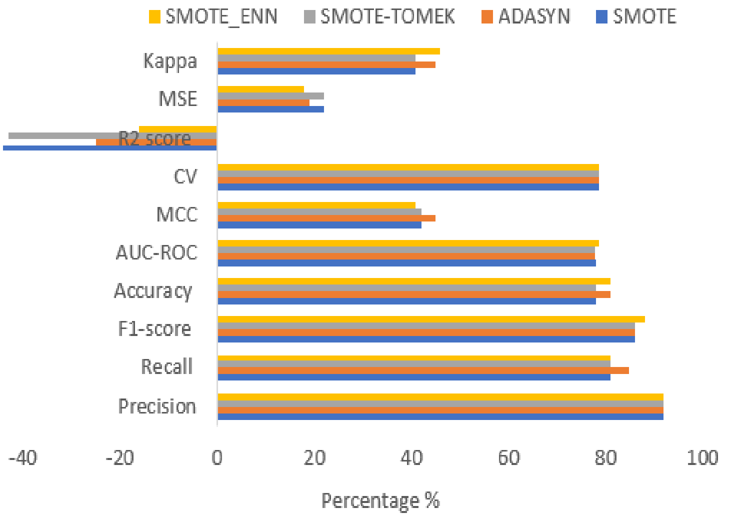


Fig 7a) Performance analysis of Sampling methods and LRC for C1 samples

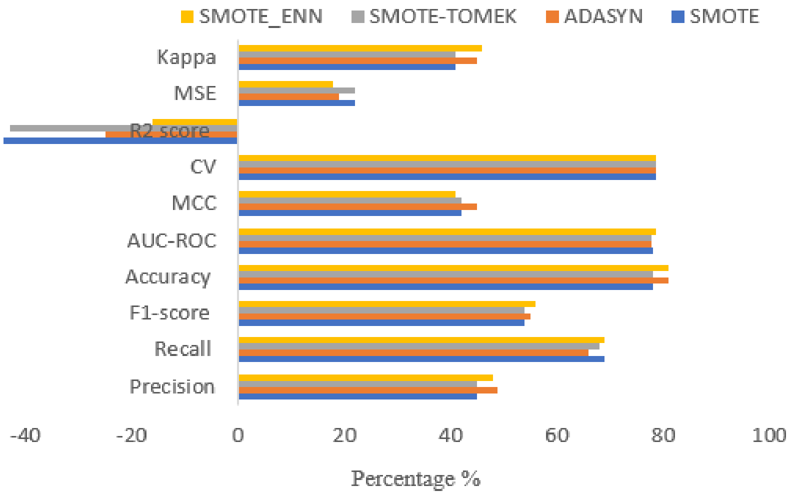


Fig 7b) Performance analysis of sampling methods and LRC for C2 samples

Table 4. Performance analysis summary of DTC with various sampling

DTC and Sampling	Class	Precision	Recall	F1-score	Accuracy	AUC-ROC	MCC	CV	R2 score	MSE	Kappa
SMOTE	0	0.91	**0.78**	**0.84**	0.76	0.78	0.37	80.3	-0.63	0.25	0.35
	1	0.43	0.52	0.49							
ADASYN	0	0.91	**0.81**	**0.86**	0.78	0.77	0.38	80.08	-0.49	0.22	0.37
	1	0.41	0.66	0.5							
SMOTE-TOMEK	0	0.91	**0.8**	**0.85**	0.78	0.78	0.36	80.3	-0.67	0.25	0.34
	1	0.43	0.63	0.51							
SMOTE_ENN	**0**	**0.92**	**0.82**	**0.87**	**0.8**	**0.79**	**0.39**	**80.9**	**0.67**	**0.14**	**0.45**
	1	0.48	0.67	0.53							

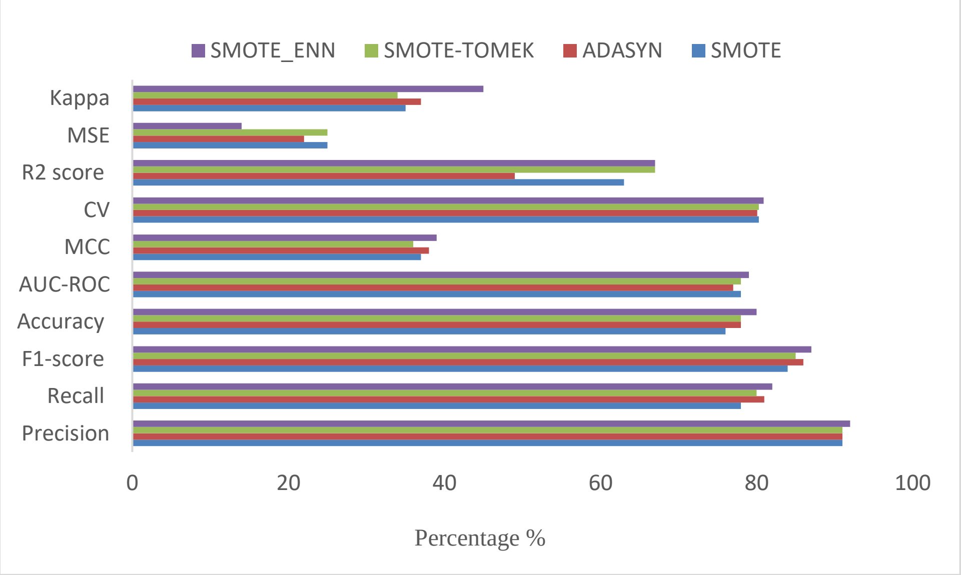


Fig 8a) Performance analysis of Sampling methods and DTC for C1 samples

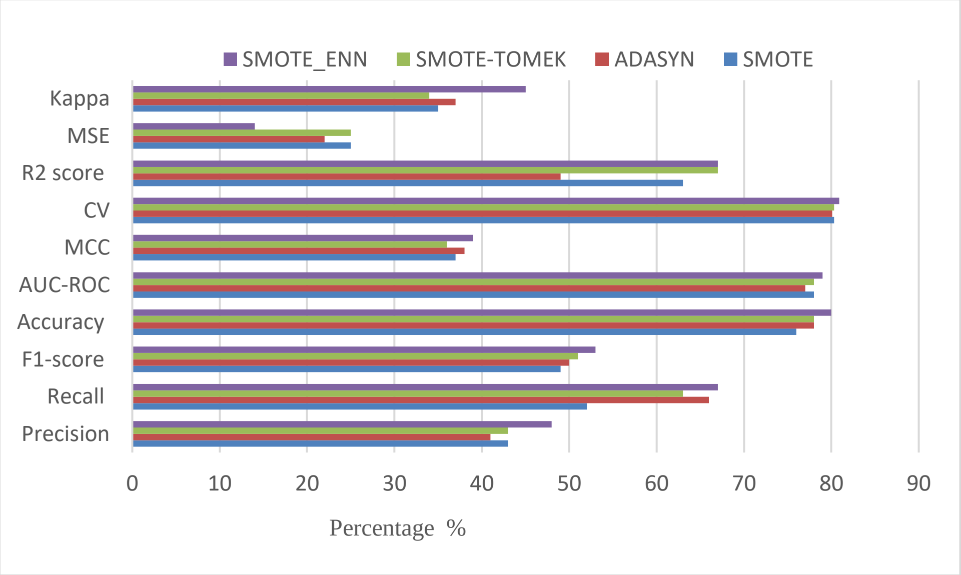


Fig 8b) Performance analysis of Sampling methods and DTC for C2 samples

Table 5. Performance analysis summary of RFC with various sampling

RFC and Sampling	Class	Precision	Recall	F1-score	Accuracy	AUC-ROC	MCC	CV	R2 score	MSE	Kappa
SMOTE	0	0.92	0.82	0.87	0.79	0.8	0.42	83.68	0.40	0.21	0.40
	1	0.46	0.67	0.55							
ADASYN	0	0.91	0.83	0.87	0.8	0.81	0.43	83.64	0.36	0.21	0.41
	1	0.46	0.66	0.54							
SMOTE-TOMEK	0	0.92	0.83	0.87	0.8	0.81	0.43	83.62	0.37	0.20	0.42
	1	0.47	0.67	0.55							
SMOTE_ENN	**0**	**0.93**	**0.69**	**0.79**	0.8	0.81	0.45	83.74	0.56	0.18	0.45
	1	**0.48**	**0.76**	**0.55**							

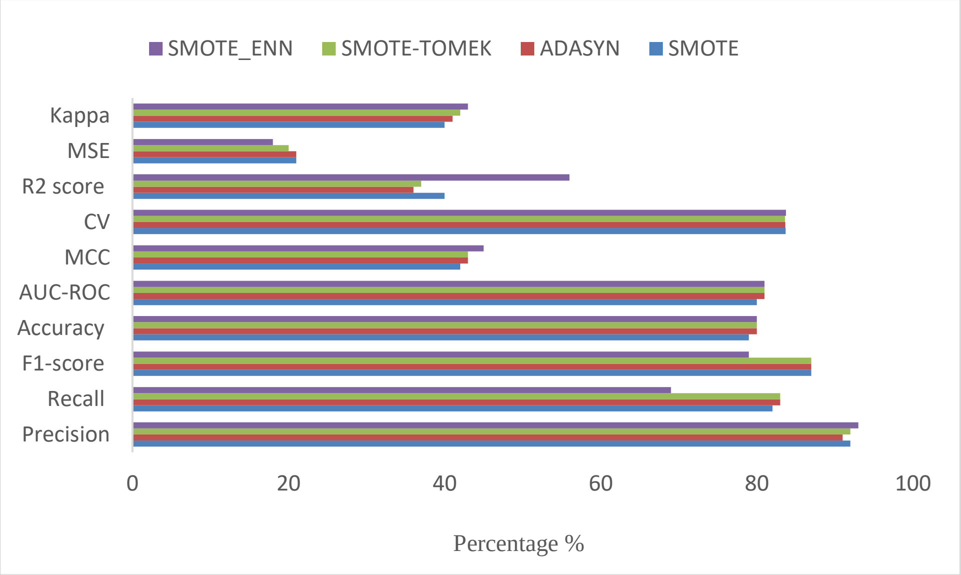


Fig 9a) Performance analysis of Sampling methods and RFC for C1 samples

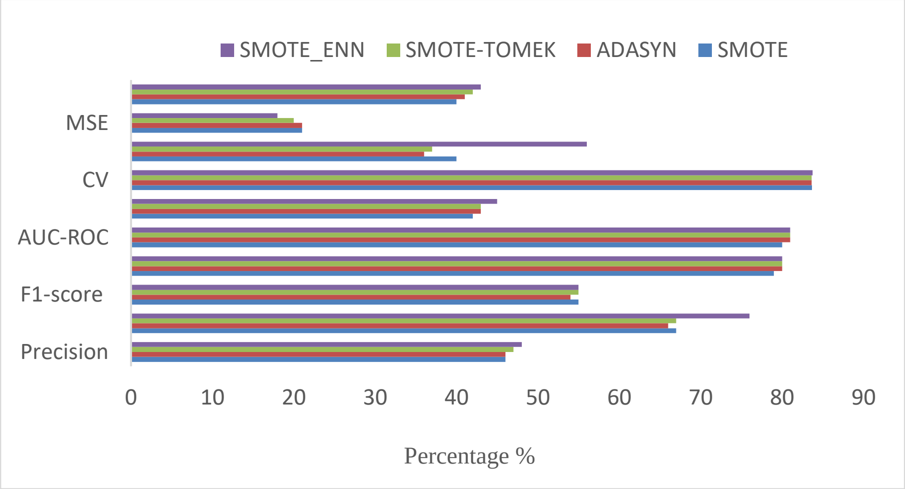


Fig 9b) Performance analysis of Sampling methods and RFC for C1 samples

From this summary, the recall and F1-Score is improved with the help of sampling algorithm SMOTE, ADASYN, SMOTE TOMEK and SMOTE ENN. From this analysis, the combined RFC+ SMOTE ENN provide better performance among three classifiers but it degrades the overall classifier accuracy, precision and AUC score. Among the three classifiers, RFC provide better performance for CC prediction.

4.5 Classification using Hybrid Grey Wolf Optimization and Random Forest Classifier

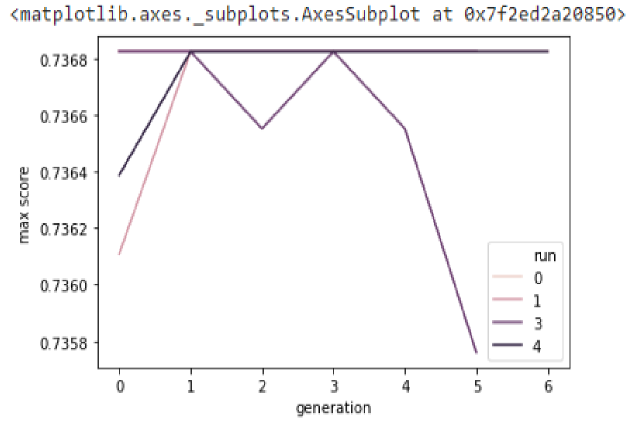
The meta heuristic algorithm GWO is used to optimize the feature after pre-processing. The parameters used in nia search is shown in Table 6.

Table 6. Outline of the parameters used in nia-search

Parameter	**Value**
clf	Random Forest
param grid	'n_estimators': range(20, 200, 20), 'max_depth': range(5, 100, 20), 'min_samples_split': range(2, 10, 5), 'max_features': ["auto", "sqrt", "log2"]

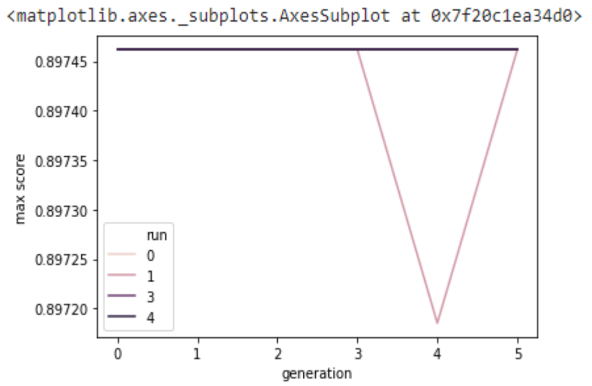
Cv	5
verbose	0
algorithm	Gwo
Population	50
Max_n_gen	100
max_stagnating_gen	5
runs	5
n_jobs	-1
scoring	'f1_macro',
random state	42

The proposed HGWRF is tested by without sampling and with sampling approaches. The summary is listed in Table 7. The maximum score generated by HGWRF is shown in figure a to e.



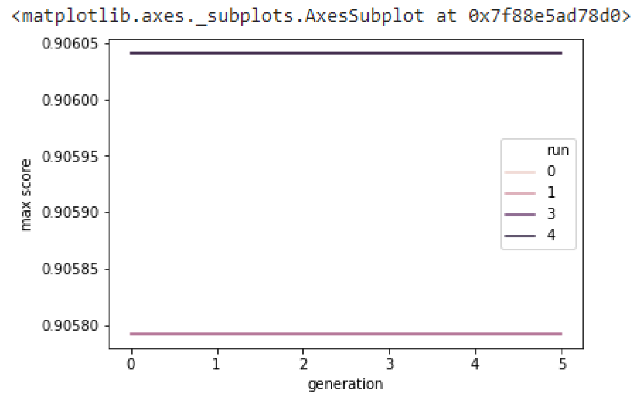
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2ed2a20850>
```

Fig a. HGWRF



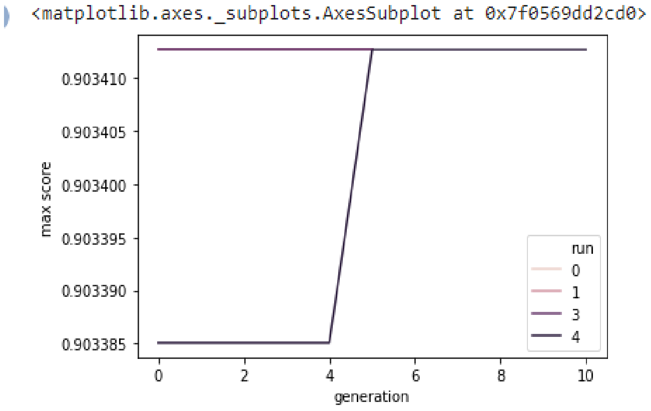
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f20c1ea34d0>
```

Fig b.SMOTE+ HGWRF



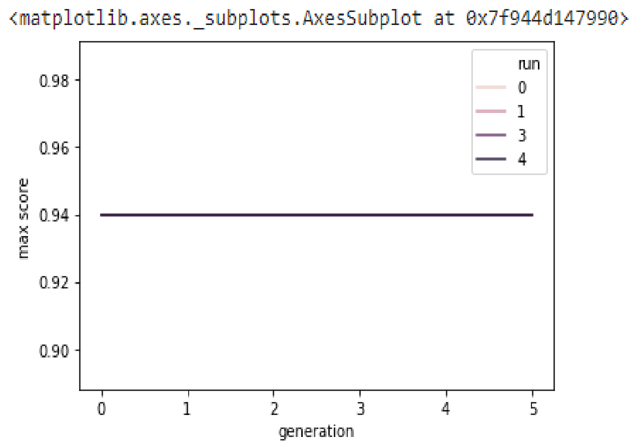
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f88e5ad78d0>
```

Fig c. SMOTETomek+HGWRF



```
<matplotlib.axes._subplots.AxesSubplot at 0x7f0569dd2cd0>
```

Fig d. ADASYN+HGWRF



```
<matplotlib.axes._subplots.AxesSubplot at 0x7f944d147990>
```

Fig e SMOTEENN+HGWRF

Table 7. Performance analysis summary of HGWRF with sampling and without sampling

Sampling and *Optimization*	*Class*	*Precision*	*Recall*	*F1-score*	*Accuracy*	*AUC-ROC*	*MCC*	*CV*	*R2 score*	*MSE*	*Kappa*
HGWRF	0	0.88	0.96	0.92	0.88	0.8	0.509	83.52	0.23	0.13	0.49
	1	0.73	0.45	0.56							
SMOTE + HGWRF	0	0.89	0.92	0.91	0.89	0.88	0.82	84.1	0.65	0.09	0.82
	1	0.61	0.52	0.56							
ADASYN + HGWRF	0	0.89	0.93	0.91	0.90	0.89	0.818	85.3	0.64	0.09	0.82
	1	0.92	0.88	0.90							
SMOTE-TOMEK + HGWRF	0	0.90	0.92	0.91	0.91	0.90	0.82	86.7	0.64	0.09	0.82
	1	0.92	0.89	0.90							
SMOTE ENN + HGWRF	0	0.96	0.94	0.93	0.94	0.93	0.825	87.2	0.65	0.09	0.82
	1	*0.92*	*0.92*	*0.95*							

From this summary, the HGWRF give better accuracy 88%, high CV (83.52%), MCC (50.9%) , less MSE (0.13) compared to RFC with sampling and without sampling. The HGWRF along with SMOTE ENN provide good accuracy 94% and AUC score 93% , higher cross validation score 87.2 and good kappa score 82%. Hence the proposed HGWRF and SMOTEENN is suited for credit card prediction.

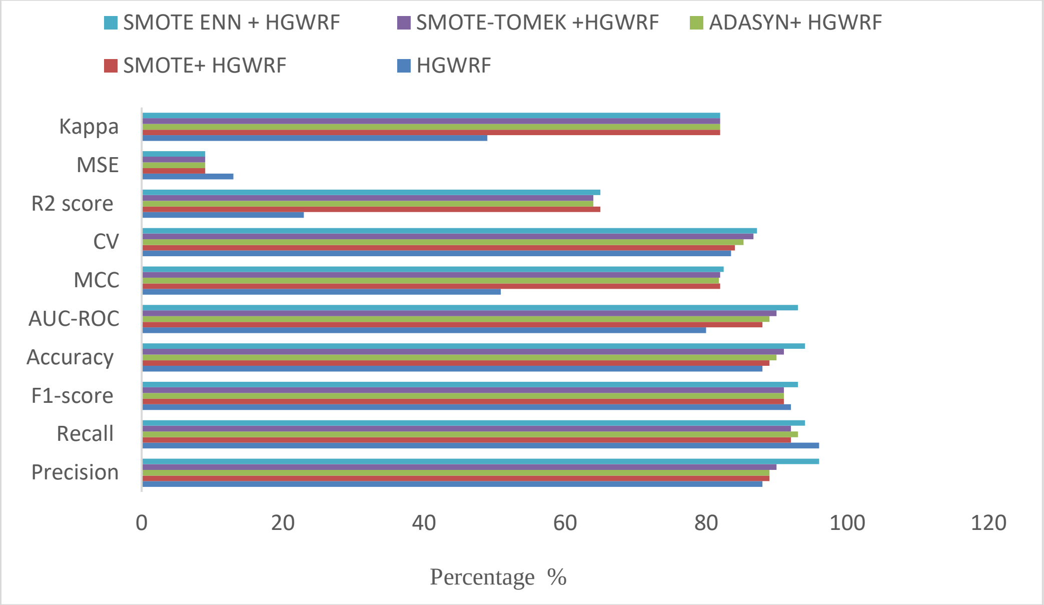


Fig 9a) Performance analysis of Proposed HGWRF and Sampling methods for C1 samples

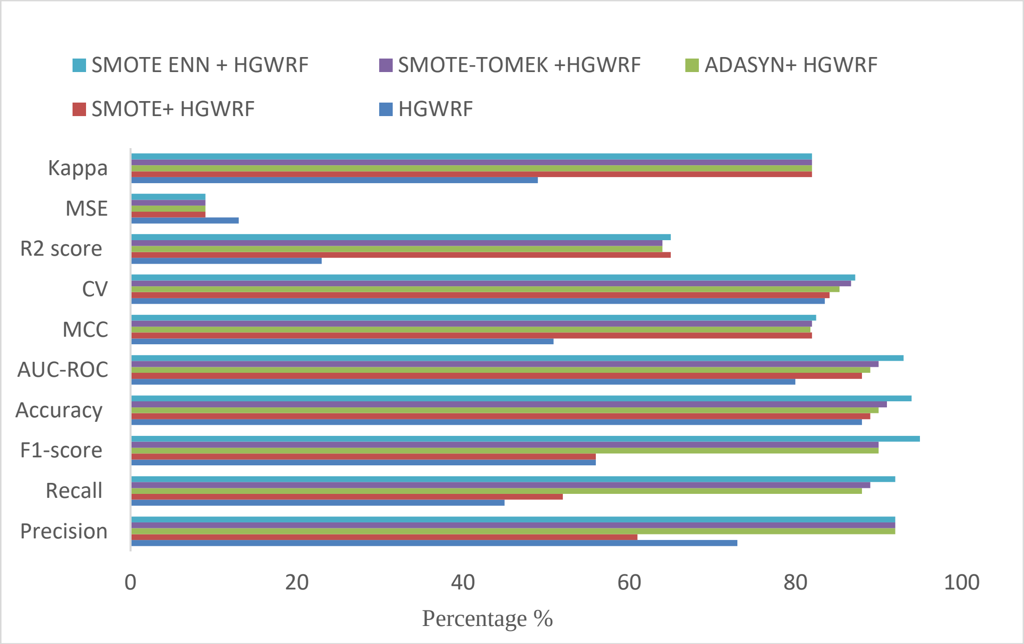


Fig 9b) Performance analysis of Proposed HGWRF and Sampling methods for C2 samples

5. Conclusion and Future work

In this paper, the financial fraud detection is predicted by HGWRF algorithm. It uses GWO meta heuristic algorithm for best feature selection. The optimized performance of classifier is compared with LRC and DTC classifier. The HGWRF provide 94% accuracy when it is incorporated with SMOTE ENN sampling algorithm. The proposed HGWRF improve the basic classification metrics and provide good AUC, CV, MCC and Kappa score. It reduces the mean square error value for CC fraud prediction. It solves the problem which occur in imbalanced data set and optimize the process of feature selection. The feature selection using GWO improve the model prediction and provide better perspective on significant features. In future, demonstrate the performance of HGWRF in handling other kinds of imbalanced data set and do analysis on usage of meta heuristics in various applications,

References

[1] Toor A and Usman M (2022). Adaptive telecom churn prediction for concept-sensitive imbalance data streams.Journal of Supercomputing,78,3,pp. 3746-3774.

[2] Shi, Meifeng and Liao, Xin and Chen, Yuan (2022). A dual-population search differential evolution algorithm for functional distributed constraint optimization problems,Annals of Mathematics and Artificial Intelligence,90,10,1055–1078.

[3] Sefati, S., Mousavinasab, M., Zareh Farkhady, R. (2022). Load balancing in cloud computing environment using the Grey wolf optimization algorithm based on the reliability: February 24, 2023 19:33 WSPC/INSTRUCTION FILE Manuscript Imbalanced creditcard fraud dataset handiling and prediction using SMOTEENN and HGWRF 17 performance evaluation. Journal of Supercomputing., 78,1, 18-42.

[4] Kurniawati Y. E and Prabowo Y. D. (2022). Model optimisation of class imbalanced learning using ensemble classifier on over-sampling data. Int J Artificial Intelligence, 2252,(8938), 8938.

[5] Cherif A, Badhib, A., Ammar, H., Alshehri, S., Kalkatawi, M., Imine, A. (2022). Credit card fraud detection in the era of disruptive technologies: A systematic review. Journal of King Saud University-Computer and Information Sciences Prusti D and Rout J. K. (2023).

[6] Detection of credit card fraud by applying genetic algorithm and particle swarm optimization.Machine Learning, Image Processing, Network Security and Data Sciences:pp. 357-369.

[7] Mittal S, Tyagi, S. (2020). Computational techniques for real-time credit card fraud detection.In Handbook of Computer Networks and Cyber Security: 653-68.

[8] Osman I. H and Kelly, J. P. (1997). Meta-heuristics theory and applications. Journal of the Operational Research Society, 48,6, 657-657. Liu J., Wei, X., Huang, H. (2021). An improved grey wolf optimization algorithm and its application in path planning. IEEE Access, 9, 121944-121956.

[9] Van Belle, R., Baesens, B., De Weerdt, J. (2023). CATCHM: A novel network-based credit card fraud detection method using node representation learning. Decision Support Systems, 164, 113866.

[10] Peng C. Y , Park, Y. J. (2021). A New Hybrid Under-sampling Approach to Imbalanced Classification Problems. Applied Artificial Intelligence, 1-18.

[11] Cerqueira, V., Torgo, L., Branco, P., Bellinger, C. (2022). Automated imbalanced classification via layered learning. Machine Learning, 1-22.

[12] Wei, W., Jiang, F., Yu, X., Du, J. (2022, January). An Under-sampling Algorithm Based on Weighted Complexity and Its Application in Software Defect Prediction. 5th International Conference on Software Engineering and Information Management (ICSIM): 38-44.

[13] Tanimoto, A., Yamada, S., Takenouchi, T., Sugiyama, M., Kashima, H. (2022). Improving imbalanced classification using near-miss instances.Expert Systems with Applications, 201, 117130.

[14] Wang, J., Zhang, F., Jia, X., Wang, X., Zhang, H., Ying, S., ... Shen, D. (2022). MultiClass ASD Classification via Label Distribution Learning with Class-Shared and ClassSpecific Decomposition.Medical Image Analysis, 75, 102294.

[15] Bhardwaj, P., Tiwari, P., Olejar Jr, K., Parr, W., Kulasiri, D. (2022). A machine learning application in wine quality prediction. Machine Learning with Applications, 8, 100261.

[16] Chennuru, V. K., Timmappareddy, S. R. (2022). Simulated annealing based undersampling (SAUS): A hybrid multi-objective optimization method to tackle class imbalance. Applied Intelligence, 52,2, 2092-2110.

[17] Padhi, B. K., Chakravarty, S., Naik, B., Pattanayak, R. M., Das, H. (2022). RHSOFS: Feature Selection Using the Rock Hyrax Swarm Optimization Algorithm for Credit Card Fraud Detection System. Sensors,,22,23, 9321.

[18] Kunhare, N., Tiwari, R., Dhar, J. (2022). Intrusion detection system using hybrid classifiers with meta-heuristic algorithms for the optimization and feature selection by genetic algorithm. Computers and Electrical Engineering, 103, 108383.

[19] Khan A. T, Cao, X., Li, S., Katsikis, V. N., Brajevic, I., Stanimirovic, P. S. (2022). Fraud detection in publicly traded US firms using Beetle Antennae Search: A machine learning approach. Expert Systems with Applications, 191, 116148.

[20] Petrovic, A., Bacanin, N., Zivkovic, M., Marjanovic, M., Antonijevic, M., Strumberger, I. (2022, June). The AdaBoost Approach Tuned by Firefly Metaheuristics for Fraud Detection. In 2022 IEEE World Conference on Applied Intelligence and Computing (AIC): 834-839.

[21] Tang, J., Luo, Q., Zhou, Y. (2022). Enhanced artificial ecosystem-based optimization selforganizing RBF neural network. Journal of Ambient Intelligence and Humanized Computing,: 1-13.

[22] Safari, E., Peykari, M. (2022). Improving the multilayer Perceptron neural network using teaching-learning optimization algorithm in detecting credit card fraud. Journal of Industrial and Systems Engineering,14,2, 159-171.

[23] Singh, A., Jain, A., Biable, S. E. (2022). Financial Fraud Detection Approach Based on Firefly Optimization Algorithm and Support Vector Machine. Applied Computational Intelligence and Soft Computing, 2022.

[24] Guha, D., Roy, P. K., Banerjee, S. (2016). Load frequency control of large scale power system using quasi-oppositional grey wolf optimization

algorithm. Engineering Science and Technology ,19,4, 1693-1713.

[25] Singh Yadav, A. K., Sora, M. (2022). Unsupervised learning for financial statement fraud detection usin.g manta ray foraging based convolutional neural network. Concurrency and Computation: Practice and Experience,34,27, e7340.

[26] Tayebi, M., El Kafhali, S. (2022). Performance analysis of metaheuristics based hyperparameters optimization for fraud transactions detection. Evolutionary Intelligence,:1-19

[27] Umarani V, Julian, A., Deepa, J. (2021). Sentiment Analysis using various Machine Learning and Deep Learning Techniques. Journal of the Nigerian Society of Physical Sciences, 385-394.

[28] Revathy, S. ., & Priya, S. S. . (2023). Enhancing the Efficiency of Attack Detection System Using Feature selection and Feature Discretization Methods. International Journal on Recent and Innovation Trends in Computing and Communication, 11(4s), 156–160. https://doi.org/10.17762/ijritcc.v11i4s.6322

[29] Mr. Kankan Sarkar. (2016). Design and analysis of Low Power High Speed Pulse Triggered Flip Flop. International Journal of New Practices in Management and Engineering, 5(03), 01 - 06. Retrieved from http://ijnpme.org/index.php/IJNPME/article/view/45