International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Ingénierie des Systèmes d'Information
Vol. 29, No. 4, August, 2024, pp. 1441-1452
Journal homepage: http://iieta.org/journals/isi

Evaluation of Financial Credit Risk Management Models Based on Gradient Descent and Meta-Heuristic Algorithms

Oluwasola S. Maitanmi, Olufunmilola A. Ogunyolu*, Afolashade O. Kuyoro

Department of Computer Science, Babcock University, Ilishan 121003, Ogun, Nigeria

Corresponding Author Email: Ogunyolu0363@pg.babcock.edu.ng

https://doi.org/10.18280/isi.290417

Received: 16 October 2023
Revised: 3 April 2024
Accepted: 9 May 2024
Available online: 21 August 2024

Keywords:
credit risk, deep learning, feature selection, gradient descent, multilayer perceptron, loan prediction, random forest

ABSTRACT

An efficient credit risk management model is a promising technique that provides Financial Institutions or Banks the ability to determine a creditworthy customer from a non-worthy customer. The fact remains that no country's economy can survive or improve without credit using historically available data. This paper presents an evaluation of several gradient descent techniques, and metaheuristic optimization algorithms implemented in Machine Learning and Multi-layer perceptron for better credit risk prediction. It also handles imbalanced dataset using smote Edited Nearest Neighbour. The study provided various architectures and advantages of the algorithms while addressing how the limitations can be improved to build a better credit risk model and improve model accuracy. The study showed MLP WOA achieved accuracy of 98.56% based on Adam gradient descent to achieve faster convergence and exploration compared to MLP PSO with 98.39%.

1. INTRODUCTION

Financial Institutions can avoid insolvency when customers are granted loans. To achieve this, a credit risk prediction system can classify a defaulter rather than pursue recovery due to financial loss. Over the years, researchers explored the use of machine learning techniques to build models that can identify credit-worthy customers; however, a financial credit risk dataset is usually imbalanced, and this can cause data classification to be biased towards the majority class, this implies that it can misrepresent a reliable borrower for an unreliable borrower [1]. As a result of limitations of machine learning in credit risks such as overfitting, underfitting, inability to handle large datasets, high use of computational resources, and the possibility of inability to handle bias and variance [2]. Deep Learning has been adopted to be able to handle limitations inherent in machine learning techniques such as handling missing data, the ability to make predictions, handling complex and large datasets, especially financial datasets, generalization as well as scalability [3].

A class of a deep Neural Network is a Multi-layer Perceptron, it comprises multiple layers having many neurons work together such that the inputs of neurons are from the output of some neurons and an input layer. Multilayer perceptron (MLP) is a feedforward Neural Network that connects several layers through nodes using a signal path. It is a supervised technique that uses backpropagation to handle and achieve minimal loss function after specified iterations. MLP has been applied in image recognition, machine translation, and neuroscience among others. In other to maximize the predictive power of MLP and handle optimization tasks like feature interdependencies, High dimensionality, non-convex optimization, prioritization, and multi-objective optimization. During training, the algorithm adjusts internal connections to map target output from input patterns making prediction possible by learning from data. Metaheuristic algorithms are used to handle optimization issues in deep learning Neural Network and improve the model unlike traditional techniques, this approach uses exploration and exploitation strategies to avoid getting stuck at local minima and achieve global minima through further exploration of other regions by the search agent. One of the optimization algorithms is the Whales Optimization Algorithm (WOA), this uses the hunting behaviour of humpback whales and has been used in active research like electrical power systems, robotics, and training artificial Neural Networks with an effective record of achieving convergence in terms of speed [4]. The Humpback Whale uses mathematical models such as searching, encircling, and hunting to determine the location of its prey using bubble nets. It does need gradient descent information to function.

This study makes use of Multilayer Perceptron optimized by whale's optimization algorithm known for its ability to handle large datasets, make fast predictions during training using Adam optimization. It has the ability to handle classification between defaulters and non-defaulters. In general, it has high processing power with the ability to handle both linear and non-linear patterns. There are three important Neural Network activation functions which include the Binary Step function, Linear activation Function, and Non-Linear activation function [5]. That which handles the limitations of the linear activation function is backpropagation. Backpropagation handles the reduction of cost function during training by adjusting biases and weights on the network after including another step at every layer during forward propagation. An optimization algorithm is used during training to minimize error while identifying the appropriate weight essential to the efficiency of the model. To minimize the cost function, there are optimization algorithms used, among which are gradient descent, and meta-heuristic algorithms among others. The objective of the study is to review several pieces of literature of multilayer perceptron based on gradient descent algorithms and meta-heuristic optimizations, elaborate on the findings of the analysis in credit risk management, make a comparison, evaluation and conclusion.

2. MULTILAYER PERCEPTRON

MLP is an integral aspect of deep learning and is a global approximator used for non-linearly mapped input-to-output networks, it is applicable in prediction, and classification among others, and is known as a feed-forward artificial Neural Network. It is made of an input x_i layer that accepts input signals for processing and flows forward with at least one hidden layer having an engine that handles the computation of the network. It comprises artificial neurons with mathematical functions used to compute the weighted total of several inputs and outputs with an activation function that regulates based on a threshold as shown in Figure 1.

The output layer y-predicted class label is expected to perform the prediction and classification of the credit risk model. As mentioned in Figure 2, at the output layer, an error is computed as the difference between the actual and predicted, if the error is high, the Backpropagation algorithm is used to train the neurons by adjusting parameters such as each weight w by calculating its derivative, and the network is updated. Other parameters adjusted are learning rates, and activation type suitable for the prediction model, to minimize the cost function.

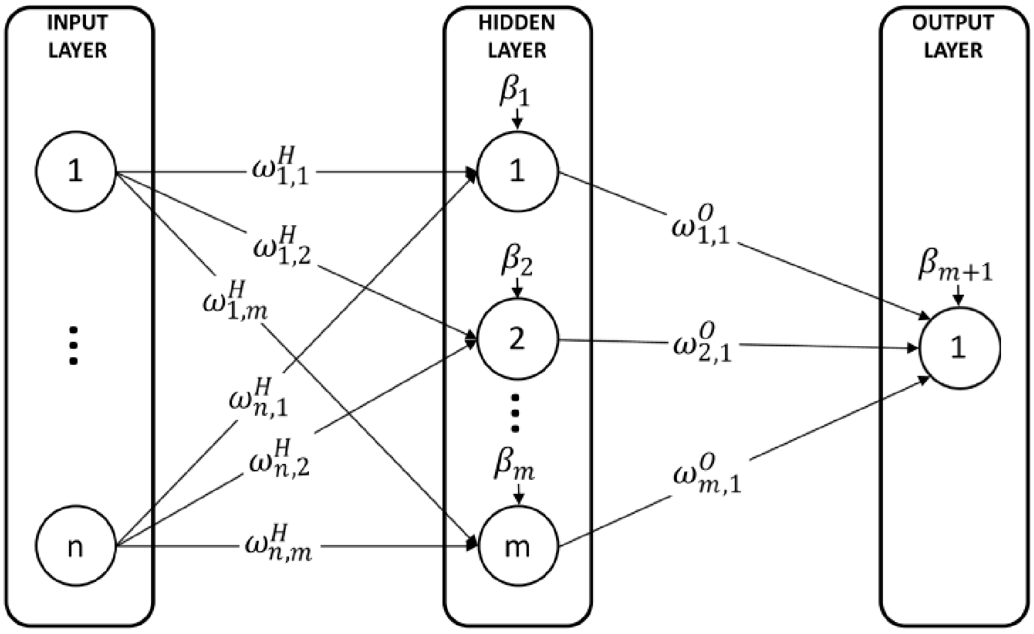


Figure 1. Architecture of multilayer perceptron [6]

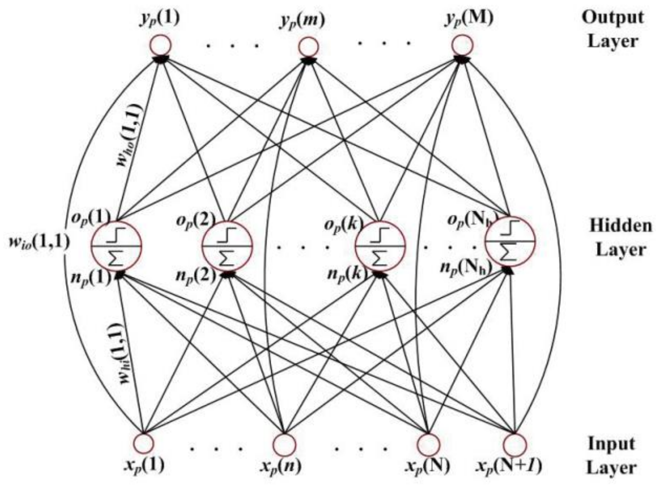


Figure 2. Unit function perceptron rule [7]

The importance of training a multilayer perceptron ANN is to avoid local minima, achieve global minima solution while minimizing execution time, and to achieve convergence at an appropriate and effective time. The minimal error to avoid local minima is achieved during backpropagation of the error into the network by adjusting weights per cycle called Epochs. The first technique was the use of gradient descent to achieve convergence but faced several issues like slow convergence, and vanishing gradients which are dependent on parameters like learning rate and momentum [8]. Gradient descent is expected to improve model accuracy after going through several cycles or epochs until convergence by fine-tuning the hyperparameters to minimize loss function using an optimization approach.

2.1 Gradient descent algorithm

This is an optimization algorithm that uses an iterative approach to get a minimal value of a convex function by identifying the parameters of a function. While developing a model, one limitation known is the ability to get stuck at the local minimum which makes it tedious to approach the global minimum. It is built on a function f(x; θ) where θ is a parameter vector within the neighborhood of a point while moving towards a negative gradient with minimal iterative steps as f(x,θ) tends to approach local minimum as an input. The most appropriate parameter is required while fitting the model to the data [9]. Different types of gradient descent algorithms have been applied in credit risk prediction to build a model that will classify defaulters from non-defaulters.

2.1.1 Stochastic gradient descent (SGD)

SGD is known as a Gradient algorithm for optimizing models of machine and deep learning. It majorly handles limitations of the gradient descent algorithm, and it is more computationally efficient when large datasets are involved as in the case of financial datasets. It takes more iterations to reach a minimum due to its randomness and also iterates per time an observation. One of its limitations is its slowness to convergence which is usually less accurate due to noise and makes it less useful in building a credit risk prediction model [10, 11]. As shown in Figure 3, the main objective of SGD is to minimize loss function that is the error that exists between the predicted and actual output.

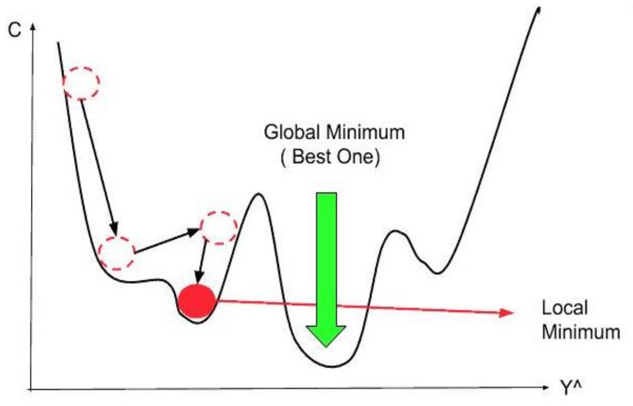


Figure 3. Global and local minimum [11]

2.1.2 Mini-batch gradient

One of the strengths of Mini-Batch Gradient is the ability to handle large datasets by dividing the financial datasets into smaller sizes when carrying out computation on each batch size as shown in Eq. (1). Where w is the parameter weight, *y* is the target variable, *b* is the batch size and *x* is the individual variable [12]. The loss function is also calculated per batch. It is computationally efficient with a higher frequency which makes convergence faster with a lesser possibility of falling into local minima [13].

$$w_{i+1} = w_i - a.\nabla_{wi}\, j(x^{i:i+b};\, w_i) \tag{1}$$

2.1.3 Adam optimization

This algorithm incorporates Momentum and Root mean square Propagation, it is known for its ability to enhance model speed and with its adaptive rate, makes a model learn faster while converging. Adam Optimizer, unlike SGD, which requires more epochs to converge [14], makes it less possible for the learning process to fall into a local minimum by updating the weights to provide a higher weights vector. Adam Optimization algorithm can handle classification tasks with a Multi-layer Perceptron which comprises an activation function, an input, an output layer, a loss function, and at least a hidden layer. Based on the algorithm adopted by deep learning, a subset of Machine Learning tries to make a prediction. During training, while mapping the inputs to output, the optimization algorithm tends to reduce the error by adjusting weights, and epoch weights are subsequently added while minimizing the loss function and enhancing accuracy.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{2}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \tag{3}$$

Adam Mathematical definition, where, m and v are moving averages used to maintain decay rates as mentioned in Eqs (2) and (3), this decay rate or learning rate is for both optimization and generalization, while β is the hyperparameter and g stands as the gradient [15]. Other enhanced gradient descent optimization techniques are Adagrad, Adadelta, RMSProp, and Nestrov as mentioned in Figure 4.

Financial dataset requires the balancing of dataset class distribution to improve model accuracy and performance. Smote is an oversampling technique necessary for generating new samples through sample minority interpolation along the nearest neighbour in data space as shown in Figure 5. Smote is known to prevent or minimise overfitting with several variations of Smote technique.

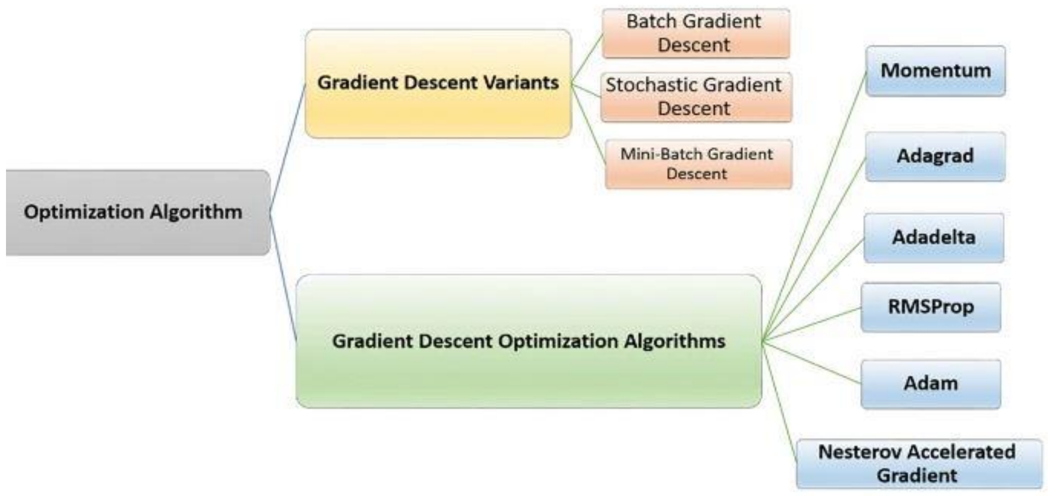

Optimization Algorithm
Gradient Descent Variants
Batch Gradient Descent
Stochastic Gradient Descent
Mini-Batch Gradient Descent
Gradient Descent Optimization Algorithms
Momentum
Adagrad
Adadelta
RMSProp
Adam
Nesterov Accelerated Gradient

Figure 4. Categories of gradient descent algorithms [16]

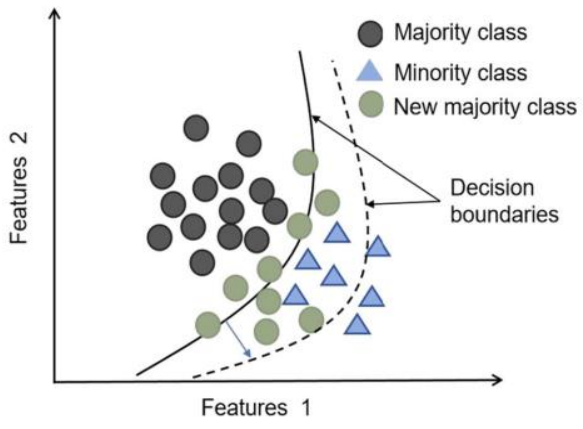


Figure 5. Smote by class distribution [17]

2.2 Synthetic minority oversampling technique-smote

Among others include Smote-NC popularly used for datasets that are categorical and continuous in nature. Another Oversampling Smote approach is borderline over samples close to the existing boarder line. A Smote approach that adopts the undersampling technique is Smote Tomek which removes majority classes that spread to minority classes through data cleaning. While Smote ENN combines oversampling with undersampling based on the Edited Nearest Neighbour approach for the removal of majority class samples present in both sample and original datasets.

Financial datasets are not only imbalanced in nature and large, which can lead to skewness such that the lower features might not be well represented in the process of building a model. Diverse techniques have been used in literature such as Resampling, Smote, Smote Edited Nearest Neighbour (ENN), and Smote Adasyn among others. Techniques like resampling have been used when the minority class is oversampled while the majority class requires the removal of some rows to make the dataset evenly distributed with minority class called undersample.

Another approach is Smote (synthetic minority oversampling technique) used in oversampling the observations in the minority class by synthesizing from the same distribution. Smote was proposed by Chawla et al. in 2002 for minority class observation [18]. Smote ENN is a hybrid technique where ENN is an undersampling using estimation of the individual majority class. The benefits of Smote in model building include minimizing overfitting, improving the model, and it is also simple to understand and implement [19].

2.3 Metaheuristic optimization

A heuristic optimization algorithm finds a solution to a difficult task using historical data gathered about the task, it however does not necessarily find the best solution but identifies a good or reliable solution. This same approach applies to metaheuristic optimization algorithms best known for solving optimization problems, examples are Genetic Algorithm, ant colony, and Particle Swarm Optimization among others. Metaheuristics uses high-level procedures to find solutions depending on task [20].

2.3.1 Particle Swarm Optimization (PSO)

PSO is a bird-like association introduced by de Almeida and Leite [21], applicable in fields like biology, chemistry, medicine, Fuel, and energy amongst others. Its advantages include minimal parameter tuning and from its particle association it gives the best solution. With slow speed to converge at a global optimal position using a velocity of observations and position observations of the financial dataset. It is also easy to implement as well as handle complex tasks, though one of its limitations is finding it difficult to reach global optimum when it comes to large datasets. However, despite the limitations of PSO, it can be easily hybridized with other well-known meta-heuristic optimization algorithms like Genetic Algorithms and swarm algorithms [22].

Other meta-heuristic algorithms are the Salp Swarm Algorithm (SSA), Whale-Optimization Algorithm, and Bayesian Optimization Algorithm among others.

2.3.2 Whale optimization algorithm (WOA)

WOA was developed to mimic the attitude of humpback Whales, it is a swarm intelligence-based algorithm for many optimization problems as well as the selection of parameters or features, it also uses assumption and then finds a suitable solution helpful for the problem identified. In recent years, WOA had its binary version which helps tackle issues of traditional WOA and is also known to handle the optimization of numerical problems. This algorithm is simple to understand and very efficient and can optimize weights and biases in MLP. It can achieve global optimum by avoiding local optimum, its effectiveness has also been upgraded using additional techniques in a hybridized way.

In the study [23], a new approach called mWOA-cee is introduced as a modified WOA which helped to improve the exploration ability of WOA, preventing local optimum issues as well as improving convergence, enhancement, and exploitability. A deep recursive Neural Network-based Whale social optimization algorithm was used to determine the creditworthiness of a customer by combining the social ski-driver algorithm and WOA using Box-Cox transformation for data processing with wrapper feature. This technique presented a higher performance with 94% accuracy [24].

2.4 Review of related works

This section summarises the impact of gradient descent algorithms and Meta-Heuristic algorithms in the search to achieve higher model accuracy by different authors. This is required while minimising the loss functions and Vanishing gradients among other limitations that cause model inefficiency.

In the study [25], it was identified stochastic gradient uses minimal computing resources when updating network parameters during the training of credit history data available. The result showed that the developed credit risk model gave a prediction accuracy of 97.11%. Another study adopted cut-off scores to establish loan defaulters from non-defaulters built with Multilayer perceptron, trained using Backpropagation and conjugate gradient. Gradient descent among others is used to train the network within the hidden layer where MLP outperformed logistic regression with higher output accuracy and lowered error loss function estimation [26].

When it comes to the gradient descent algorithm, it was first adopted before the advent of the metaheuristic optimization algorithm. One can say that Adam Optimization algorithms are mostly preferred over other algorithms in a bid to overcome the drawbacks associated with gradient descent, stochastic gradient as well as mini-batch. These limitations include high computational use of resources, and slow convergence while Adam is a hybrid of two gradient descent methods Momentum and RMSP- root means square propagation. Adam Optimization requires little memory and is easy to implement. It is also able to handle financial datasets noisy in nature, as well as its largeness in quantity and its computational time is highly sufficient. Like other gradient descent algorithms, it can be hybridized with another model to build a better credit risk prediction model.

A study identified that Mini batch gradient also shows better performance than binary logistic regression in credit risk management. This uses numerical optimization techniques to minimize the cost function. It also updates parameters during training while it requires fewer iterations to achieve convergence since updates of parameters are achieved often [27].

Another study used Back Propagation feed-forward ANN (BP-FFANN) credit risk dataset training using gradient descent based on a modified hyperbolic Tangent activation function where the error was lowered to 0.012. The result produced a sufficient accuracy and neither did it produce an Overfitting model as the derived error approached zero [28]. However, one of the limitations of activation functions like the Hyperbolic activation function is that it suffers a Vanishing Gradient such that weights are difficult to update by the network for previous layers to learn. In a multilayer network, studies used a metaheuristic algorithm approach called the Imperialist Competitive algorithm with modified min-max classification (ICA-MFMCN) to identify a reliable feature subset giving the best result at 81.0%. The study compared the result with the Genetic Algorithm Neural Network (GA-NN) with an accuracy of 78.5% using the same German dataset [29]. This performance is lower when compared with Genetic Algorithm based back propagation with accuracy of GA- BP 92.65% in another study [30].

In some studies, metaheuristic optimization algorithms were used as a hybrid approach with machine learning. Hybrid Grey Wolf optimization combined with random forest (HGWRF) improved credit card fraud detection using SMOTE-ENN as a sampling technique. This gave a better model accuracy of 94% and without SMOTE-ENN gave an accuracy of 84% as well as a lower mean square error of 0.13 compared with RFC with sampling and without sampling [31]. This model performance is similar to another hybrid technique WOA with Recurrent Neural Network (RNN) with accuracy of 94% [32]. As mentioned in Eq. (4), Studies hybridized the use of Metaheuristic and Neural Networks to improve accuracy and speed. This implies that metaheuristic optimization types like Particle Swarm Optimization helped converge faster during the initial search stage than gradient descent. PSO however began to reduce in speed at global optimum as GD improved in accuracy while converging and achieving excellent speed [33].

$$jw_{t+1} = w_t + GD \rightarrow w_{t+1} = w_t + Meta \tag{4}$$

Metaheuristic-optimization is based on artificial Neural Networks using Particle Swarm Optimization and cost-sensitive fitness functions in a majority voting ensemble [34]. This showed significant advancement and improvement with G-Mean from an initial 0.211 to 0.842, F1 score from 0.104 to 0.141. Data imbalance was handled such that prediction of minority class was improved, and accuracy improved significantly having adopted the ensemble cost-sensitive approach. PSO is usually preferred as it requires minimal iterations and only a few parameters.

In other studies, it was proposed that rather than adopting gradient descent techniques for training, using feed-forward Neural Networks for pattern classification to identify weights of the feed-forward Neural Network [35]. This study proposed the use of a metaheuristic optimization algorithm. This is because gradient descent can get stuck in local optima, better and more efficient metaheuristic optimization like ant colony is used to train the feed-forward Neural Network hybridized with gradient descent. The study showed that it gave a better model performance in peer-to-peer lending.

One peculiarity of gradient descent in credit risk prediction is that it points to the direction of the steepest ascent based on credit risk parameters being the gradient associated with the cost function. As such, with each parameter, computation of the gradient is done at every iteration. The learning rate handles the algorithm moving to the minimum by controlling the step size. One can say that what is common to all types of gradient descent algorithms is the choice of learning rate, number of iterations, and MLP architecture to ensure significant improvement in the model built [36]. During training, once the learning rate of the gradient descent algorithm is not changed, it might be too large to converge thereby causing fluctuation of loss function around local minima, hence the need for learning rate reduction is necessary. This approach is for a learning rate to reach local minima at the initial training process and then minimized as training continues, this is called learning rate annealing or learning rate adaptable [37].

3. METHODOLOGY

This section discussed the model built using data preparation, processing, removal of duplicate columns and model implementation. The credit risk dataset used in this study is supervised with labeled data. The number of minority and majority samples explained the number of default and non-defaults available in the 1,048,575 as mentioned in Table 1.

Figure 6 gives details on the data types in the financial credit risk dataset with integers, objects, and floats.

Table 2 describes information about the definition of attributes available in the financial dataset.

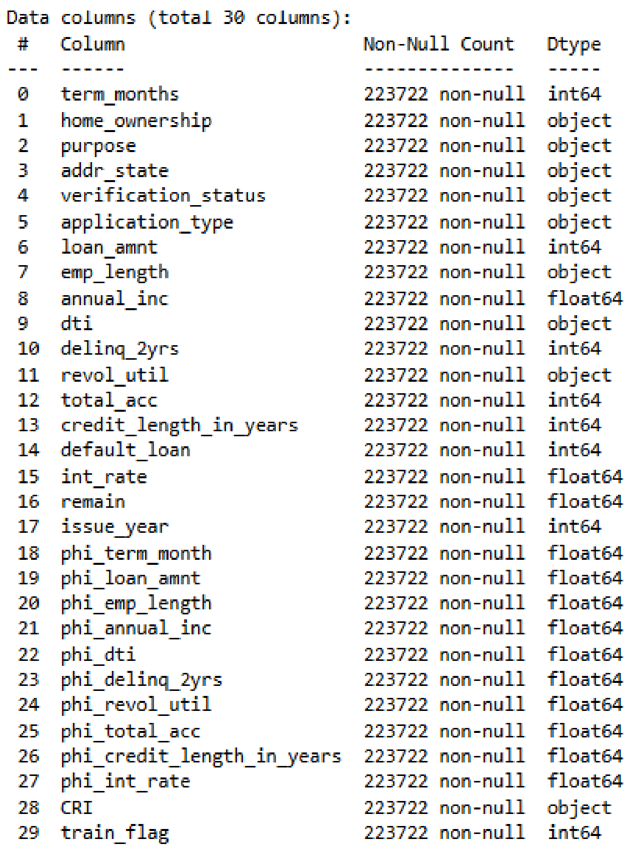
Outliers are harmful to the performance of the credit risk model, and it is necessary to identify and remove these outliers for better performance of the multi-layer Neural Network model as identified in Table 3.
The resulting information is shown in Table 4 for the target variable or class label which is default loan.

Table 1. Outline of dataset

Total Number of Samples	Total Number of Samples After Duplicate Checks	Majority Samples (Class 0)	Minority Samples (Class 1)	Number of Duplicated and Redundant Columns	Total Number of Features	Total Number of Features After Duplication Check	Class Label
1,048,575	998,922	788,544	210,378	11	30	18	1

Table 2. Definition of dataset features

Loan Information	
term_months:	Number of months over which the loan is to be repaid (indicating repayment period and risk).
home_ownership:	Borrower's homeownership status (indicating stability and potential assets).
purpose:	Reason for borrowing (different purposes may have varying risk profiles).
loan_amnt:	Amount of money borrowed (higher amounts generally carry greater risk).
int_rate:	Interest rate charged on the loan
remain:	Remaining balance on the loan (may impact repayment behavior).
issue_year:	Year the loan was issued (potentially relevant for economic trends).
Borrower Information	
addr_state:	Borrower's state of residence (could reflect economic conditions or regulations).
Verification_status:	Whether the borrower's income and employment were verified (impacts confidence in financial information).
application_type:	Whether the application was individual or joint (may influence risk assessment).
emp_length:	Length of borrower's employment in years (indicates stability and income security).
annual_inc:	Borrower's annual income (primary indicator of repayment capacity).
dti:	Debt-to-income ratio (measures proportion of income dedicated to existing debt).
delinq_2yrs:	Number of delinquent payments in the past 2 years (shows history of payment behavior).
revol_util:	Revolving credit utilization rate (percentage of available credit used).
total_acc:	Total number of credit accounts (reflects credit history and usage patterns).
credit_length_in_years:	Length of credit history in years (longer history generally indicates better credit management).
Credit Risk Indicators	
default_loan:	Whether the borrower defaulted on the loan (**the target variable for credit risk model)**
CRI:	A credit risk index or score, potentially calculated based on other features (summarizes creditworthiness).



```
Data columns (total 30 columns):
 #   Column                  Non-Null Count   Dtype
---  ------                  --------------   -----
 0   term_months             223722 non-null  int64
 1   home_ownership          223722 non-null  object
 2   purpose                 223722 non-null  object
 3   addr_state              223722 non-null  object
 4   verification_status     223722 non-null  object
 5   application_type        223722 non-null  object
 6   loan_amnt               223722 non-null  int64
 7   emp_length              223722 non-null  object
 8   annual_inc              223722 non-null  float64
 9   dti                     223722 non-null  object
 10  delinq_2yrs             223722 non-null  int64
 11  revol_util              223722 non-null  object
 12  total_acc               223722 non-null  int64
 13  credit_length_in_years  223722 non-null  int64
 14  default_loan            223722 non-null  int64
 15  int_rate                223722 non-null  float64
 16  remain                  223722 non-null  float64
 17  issue_year              223722 non-null  int64
 18  phi_term_month          223722 non-null  float64
 19  phi_loan_amnt           223722 non-null  float64
 20  phi_emp_length          223722 non-null  float64
 21  phi_annual_inc          223722 non-null  float64
 22  phi_dti                 223722 non-null  float64
 23  phi_delinq_2yrs         223722 non-null  float64
 24  phi_revol_util          223722 non-null  float64
 25  phi_total_acc           223722 non-null  float64
 26  phi_credit_length_in_years  223722 non-null  float64
 27  phi_int_rate            223722 non-null  float64
 28  CRI                     223722 non-null  object
 29  train_flag              223722 non-null  int64
```

Figure 6. Credit risk data types

Table 3. Dataset before and after the removal of outliers

No of Rows Before Removing Outliers	1,048,575
No of Rows After Removing Outliers	998,922

Table 4. Class distribution before and after resampling

Technique	0	1
Class distribution before SMOTE-ENN	788544	210378
Class distribution after SMOTE-ENN	709167	729731

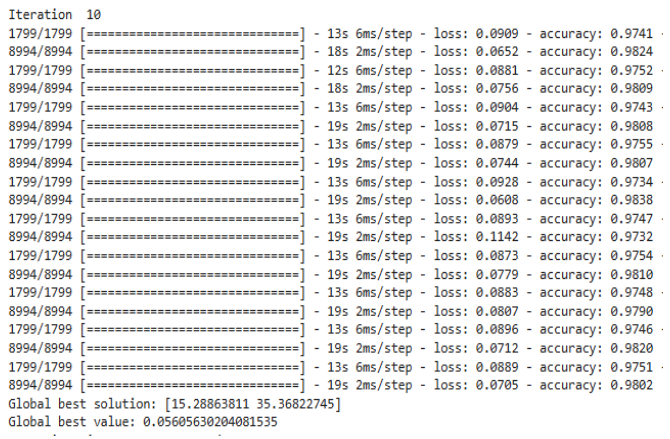
3.1 Implementation of multilayer Neural Network model

The multilayer perceptron was implemented and trained to an acceptable accuracy and later tested. The dense layer is connected as full layers, and each of the model neurons in the layer is connected directly to every other neuron in the initial layer. The Model comprises one input layer, three hidden layers with the addition of batch normalization to avoid overfitting, and a dropout of 0.5 to remove redundant features. Batch normalization was used to handle internal covariate shifts to ensure model stability. Dropout is used to remove neurons during training to help prevent overfitting and enhance generalization.

3.1.1 MLP Particle Swarm Optimization algorithm

The MLP was implemented also with the Particle Swarm Optimization (PSO) algorithm. PSO is a metaheuristic algorithm that can be used to solve optimization problems in credit risk models as shown in Figure 7.

The study also adopted an MLP with Adam optimizer and PSO after resampling with smote and ENN, as this approach is a promising approach for training classification models on imbalanced datasets. The approach shows a potential to achieve improved performance, and robustness., the best solution is represented by two values [15.28863811 35.36822745]. These values represent the weights or hyperparameters of the MLP model. Global best value indicates the fitness value associated with the global best solution. In this study, the value is 0.0561. It’s a measure of the model's performance, such as the training loss or validation accuracy. Before SMOTE-ENN, the global best solution had a value [-58.68538735 83.89741123], and the global best value was 0.06826207041740417. while after SMOTE-ENN, the global best solution had a value of [15.28863811 35.36822745], and the global best value was 0.05605630204081535. The global best value has decreased from 0.068262 to 0.056056 after resampling. This generally indicates better model performance, as lower loss or higher accuracy is often the goal of optimization. Table 5 Summary of MLP PSO Particle Search Before and After Resampling.



```
Iteration  10
1799/1799 [==============================] - 13s 6ms/step - loss: 0.0909 - accuracy: 0.9741 ·
8994/8994 [==============================] - 18s 2ms/step - loss: 0.0652 - accuracy: 0.9824
1799/1799 [==============================] - 12s 6ms/step - loss: 0.0881 - accuracy: 0.9752 ·
8994/8994 [==============================] - 18s 2ms/step - loss: 0.0756 - accuracy: 0.9809
1799/1799 [==============================] - 13s 6ms/step - loss: 0.0904 - accuracy: 0.9743 ·
8994/8994 [==============================] - 19s 2ms/step - loss: 0.0715 - accuracy: 0.9808
1799/1799 [==============================] - 13s 6ms/step - loss: 0.0879 - accuracy: 0.9755 ·
8994/8994 [==============================] - 19s 2ms/step - loss: 0.0744 - accuracy: 0.9807
1799/1799 [==============================] - 13s 6ms/step - loss: 0.0928 - accuracy: 0.9734 ·
8994/8994 [==============================] - 19s 2ms/step - loss: 0.0608 - accuracy: 0.9838
1799/1799 [==============================] - 13s 6ms/step - loss: 0.0893 - accuracy: 0.9747 ·
8994/8994 [==============================] - 19s 2ms/step - loss: 0.1142 - accuracy: 0.9732
1799/1799 [==============================] - 13s 6ms/step - loss: 0.0873 - accuracy: 0.9754 ·
8994/8994 [==============================] - 19s 2ms/step - loss: 0.0779 - accuracy: 0.9810
1799/1799 [==============================] - 13s 6ms/step - loss: 0.0883 - accuracy: 0.9748 ·
8994/8994 [==============================] - 19s 2ms/step - loss: 0.0807 - accuracy: 0.9790
1799/1799 [==============================] - 13s 6ms/step - loss: 0.0896 - accuracy: 0.9746 ·
8994/8994 [==============================] - 19s 2ms/step - loss: 0.0712 - accuracy: 0.9820
1799/1799 [==============================] - 13s 6ms/step - loss: 0.0889 - accuracy: 0.9751 ·
8994/8994 [==============================] - 19s 2ms/step - loss: 0.0705 - accuracy: 0.9802
Global best solution: [15.28863811 35.36822745]
Global best value: 0.05605630204081535
```

Figure 7. Particle Swarm Optimization iteration output

Table 5. Particle search before and after resampling

Metric	Before SMOTE-ENN	After SMOTE-ENN
Global best solution value	[-58.68538735 83.89741123]	[15.28863811 35.36822745]
Global best value	0.06826207041740417	0.05605630204081535

Table 6. MLP PSO comparison with other gradient descents

MLP PSO	Accuracy	F1 Score	Precision
Adam	98.41	98.42	98.00
SGD	98.44	98.44	97.00
Rmsprop	98.26	95.69	98.00

Before and after SMOTE-ENN, the global best solution value appears to be a two-dimensional vector, suggesting the model has multiple parameters being optimized by PSO. Interestingly, the values before SMOTE-ENN are significantly larger than after, suggesting that the resampling might have led to a smoother search space for the PSO algorithm, allowing it to find better solutions. The global best value, which reflects the model's performance metric, improves slightly after SMOTE-ENN. The original value of 0.0682 decreases to 0.0560, indicating potentially better performance or lower loss after resampling. details with other gradient descents as shown in Table 6.

3.1.2 MLP PSO with Adam, SGD and RMSProps

Adam: Achieves the highest accuracy (0.9841) and F1 score (0.982, suggesting it effectively learns complex relationships in the data and performs well at identifying both true positives and negatives. SGD: Delivers good performance with accuracy (0.9844) and F1 score (0.9844) close to Adam but slightly lower. RMSProp: Performs the weakest with lower model accuracy (0.9826) and F1 score (0.9569).

3.1.3 MLP Whales Optimization Algorithm

The model was further built using the Whales Optimization Algorithm using Adam, SGD and RMSprop to handle convergence, a metaheuristic algorithm that is used for binary tasks subjected to the same credit risk dataset, same 512 batch size, and loss function as binary cross entropy. This output provides information about the final epoch (50) of an MLP model optimized with WOA using the SGD optimizer after resampling the dataset with SMOTE-ENN.

Accuracy: 0.9847 indicates highly accurate predictions on the entire dataset. F1 Score: 0.9847 is also very high, describing good balance between precision (correctly identifying positive cases) and recall (correctly classifying all positive cases) as shown Figure 8. Classification Report also shows precision, recall, and F1 score for each class (0 and 1). Both classes have high scores, indicating strong performance for both.

Loss: 0.0479 is even lower than the training loss, suggesting good generalization to unseen data as shown in Figure 9.

A performance metric was done as shown in Figure 10 showing AUC-ROC value of 1.00.

RMSProp was also implemented with Accuracy: 0.9843074 as shown in Figure 11, F1 Score: 0.98431748, AUC-ROCScore: 0.995095

The study took a step further after resampling to implement WOA to achieve accuracy as shown in Figure 12.

The final accuracy of 0.9856 is excellent, indicating the model correctly classified almost 98.6% of the samples. The F1 score of 0.9857 further confirms strong performance, achieving a good balance between precision and recall. The AUC-ROC score of 0.9977 is exceptional, the model can effectively discriminate between classes. Accuracy however, validation accuracy improves in both epochs (0.9846 and 0.9855), reaching an even higher value (0.9856) in the final epoch. This suggests the model generalizes well to unseen data despite a slight increase in validation loss as shown in Figure 13.

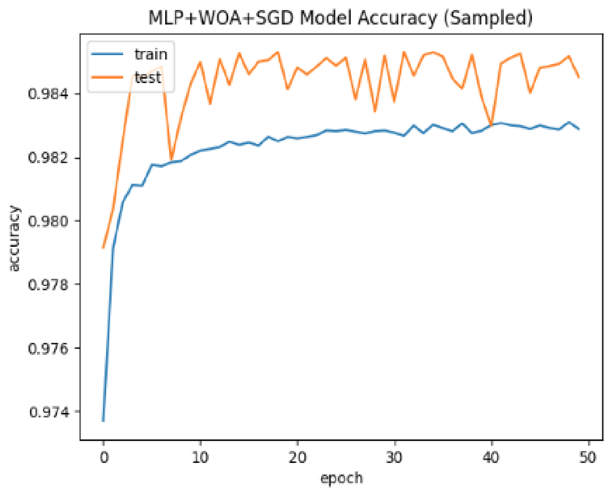


Figure 8. MLP WOA training with SGD

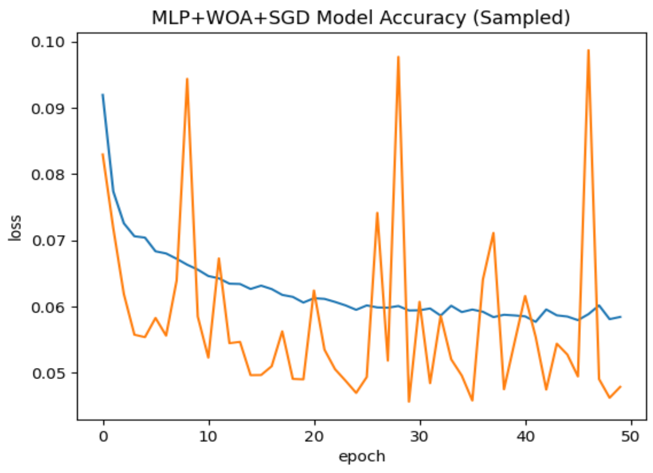


Figure 9. MLP WOA loss function with SGD

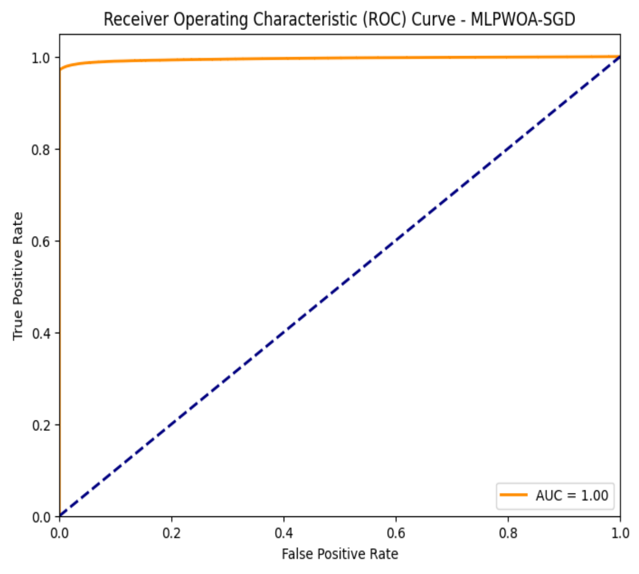


Figure 10. MLP WOA AUC-Roc with SGD

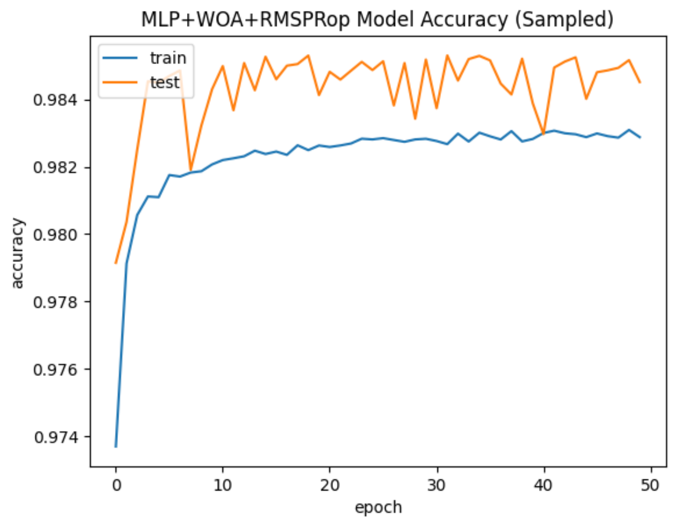
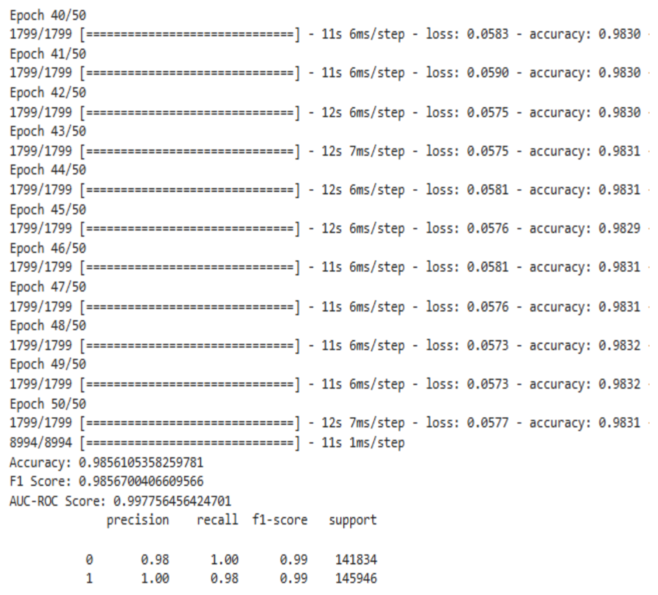


Figure 11. MLP WOA training with RMSProp



```
Epoch 40/50
1799/1799 [==============================] - 11s 6ms/step - loss: 0.0583 - accuracy: 0.9830
Epoch 41/50
1799/1799 [==============================] - 11s 6ms/step - loss: 0.0590 - accuracy: 0.9830
Epoch 42/50
1799/1799 [==============================] - 12s 6ms/step - loss: 0.0575 - accuracy: 0.9830
Epoch 43/50
1799/1799 [==============================] - 12s 7ms/step - loss: 0.0575 - accuracy: 0.9831
Epoch 44/50
1799/1799 [==============================] - 12s 6ms/step - loss: 0.0581 - accuracy: 0.9831
Epoch 45/50
1799/1799 [==============================] - 12s 6ms/step - loss: 0.0576 - accuracy: 0.9829
Epoch 46/50
1799/1799 [==============================] - 11s 6ms/step - loss: 0.0581 - accuracy: 0.9831
Epoch 47/50
1799/1799 [==============================] - 11s 6ms/step - loss: 0.0576 - accuracy: 0.9831
Epoch 48/50
1799/1799 [==============================] - 11s 6ms/step - loss: 0.0573 - accuracy: 0.9832
Epoch 49/50
1799/1799 [==============================] - 11s 6ms/step - loss: 0.0573 - accuracy: 0.9832
Epoch 50/50
1799/1799 [==============================] - 12s 7ms/step - loss: 0.0577 - accuracy: 0.9831
8994/8994 [==============================] - 11s 1ms/step
Accuracy: 0.9856105358259781
F1 Score: 0.9856700406609566
AUC-ROC Score: 0.997756456424701
              precision    recall  f1-score   support

           0       0.98      1.00      0.99    141834
           1       1.00      0.98      0.99    145946
```

Figure 12. Multilayer perceptron on whale's optimization algorithm

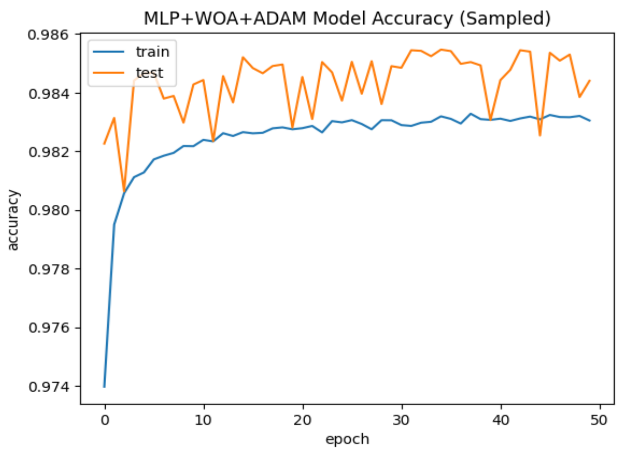


Figure 13. MLP WOA training with Adam

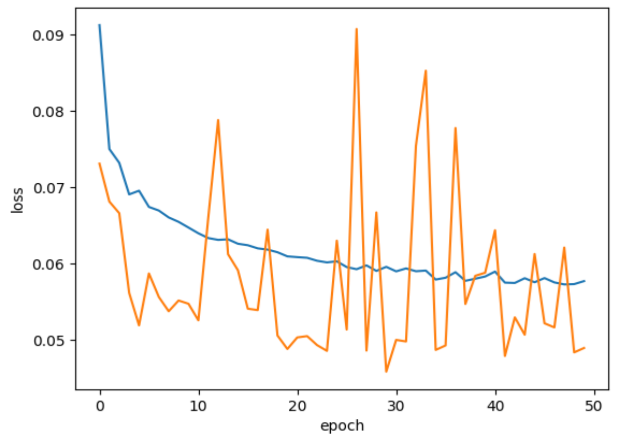


Figure 14. MLP WOA loss function with Adam

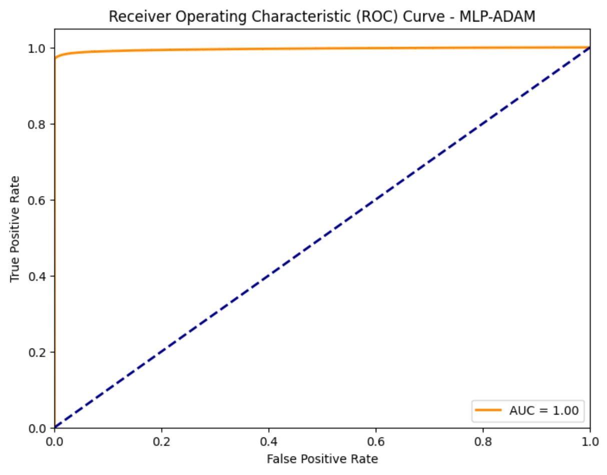


Figure 15. MLP WOA AUC-Roc with Adam

Table 7. Summary of MLP WOA result

Metric	Result
Accuracy	98.56
F1 Score	98.56
AUC-ROC Score	100
Population size	10
Batch Size	512
Iteration	10
Test Loss	0.049
Loss Function Type	Binary cross entropy

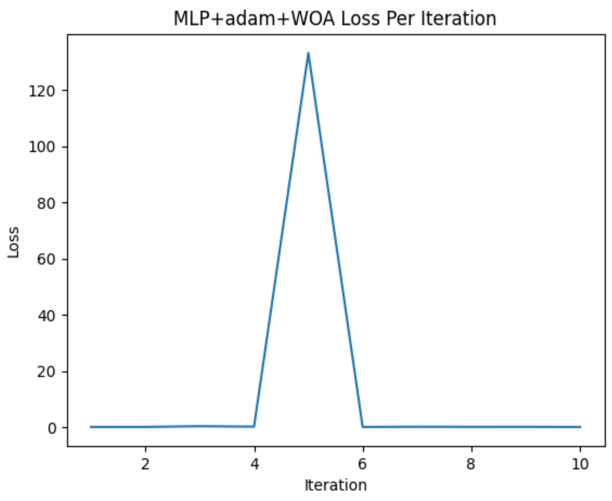


Figure 16. MLP WOA Adam loss per iteration

Table 8. Summary of MLP WOA model precision and recall

		Precision		Recall
After Resampling				
MLP+WOA+Adam	0	0.98	0	1.00
	1	1.00	1	0.98

Similar to training loss, validation loss increases slightly in both epochs 0.0484 as shown in Figure 14.

The AUC-ROC score of 0.9977 approximately 1.00 is exceptional as shown in Figure 15.

The summary of MLP-WOA result is shown in Table 7.

The lower the loss, the better the model is performing on the training data. Figure 16 described the WOA iteration and loss per iteration based on the model parameters with a minimum learning rate of 0.1.

The combination of WOA and Adam is of great advantage for training the model, as it can help to improve the model's performance, convergence and reduce the risk of overfitting. WOA with Adam works by first initializing a population of whale agents. Each whale agent represents a set of hyperparameters for the deep learning model. The whale agents then move through the search space in search of the global optimum. The precision for the default class is 1.00, which means that all of the samples that the model predicts as default are default as shown in Table 8.

Overall, the model has a very high precision score, which means that it is very good at correctly predicting both non-default and default examples. This is important for the credit risk model, as it is important to minimize both false positives (incorrectly predicting a default as non-default) and false negatives (incorrectly predicting a non-default as default). The recall for the non-default class is 1.00, which means that the model correctly predicts all of the non-default samples. The recall for the default class is 0.98, which means that the model correctly predicts 98% of the default.

Overall, the MLP WOA Adam with SMOTE-ENN model is a highly effective model for credit risk assessment. It can accurately identify both defaults and non-defaults, which is important for lenders in making informed decisions about who to lend money to. The study makes use of sigmoid on the outer layer being a binary classification and the combination of sigmoid at the outer layer and relu on hidden layers makes convergence faster, improved generalization while making the output interpretable. The learning rate is set to 0.1, especially with Adam also makes convergence faster, and ensures robustness to noise present in the gradients of the MLP model.

3.1.4 Comparison with other deep learning and machine learning works

The Results of several studies were compared based on accuracy and methodologies as shown in Table 9.

MLP+WOA (98.56%) is the best performing model, followed by MLP+PSO (98.39%). These results show that using a metaheuristic optimization algorithm like WOA to train the MLP can lead to significantly better performance compared to using Backpropagation alone (BP ANN-81.0%). AWOA-DL (MLP) (96.89%) also performs well, indicating the potential benefit of a specialized WOA variant for deep learning tasks. GA+BP (92.65%) and WOA+BP NN (92.75%) show that combining a metaheuristic algorithm with Backpropagation can still lead to improvements compared to Backpropagation alone. RNN WOA (94.0%) suggests that using WOA with a Recurrent Neural Network can achieve good performance, potentially suitable for tasks involving sequential data. GA ANN (78.5%) and BP ANN (81.0%) represent the baseline performance of using a Genetic Algorithm or Backpropagation alone, respectively.

Table 9. Accuracy comparison with other deep learning and machine learning works

	Methodology	Accuracy
Elhoseny et al. [2]	MLP+WOA	98.56%
	MLP+PSO	98.39%
	AWOA-DL (MLP)	96.89%
Wang et al. [30]	GA+BP	92.65%
	PSO+BP	91.10%
	WOA+BP NN	92.75%
Infant Cyril et al. [32]	RNN WOA	94.0%
Nourmohammadi-Khiarak et al. [29]	GA ANN	78.5%
Lucy et al. [28]	BP ANN	81.0%

Table 10. Precision and recall

Model	Precision		Recall		Accuracy	Loss @ 10th Iteration
MLP WOA	0	0.98	0	1.00	98.56%	0.049
	1	1.00	1	0.98		
MLP PSO	0	0.98	0	0.99	98.41%	0.051
	1	0.99	1	0.98		

Table 11. F1 score and accuracy

Model	F1 Score	Accuracy
MLP WOA	0.99	98.56%
MLP PSO	0.98	98.41%

This model is similar to the MLP model, but it has a different architecture as shown in Figure 17.

Accuracy measures all correctly identified cases as an obvious metric while F1 score further explains relevance of False negatives and False positives. F1 score uses harmonic mean since it allows similar values for both precision and recall as shown in Table 10 and Table 11.

$$\text{Precision} = {}^{TP}/_{TP + FP} \tag{5}$$

$$Recall = {}^{TP}/_{TP+FN} \tag{6}$$

A higher F1 score indicates that the model is better at both predicting positive examples and avoiding false positives. The MLP WOA model also has a slightly higher accuracy than the MLP PSO model, at 98.56% compared to 98.41%.

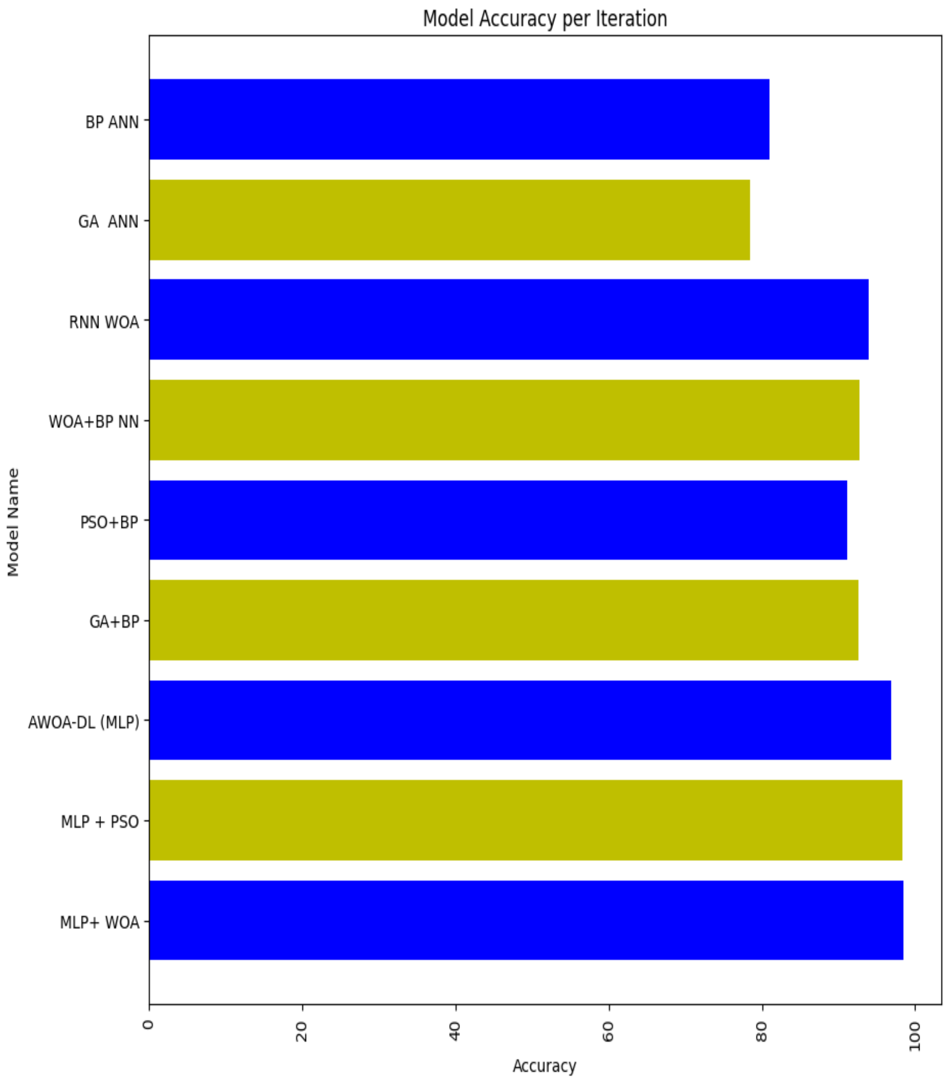


Figure 17. MLP WOA and other deep learning and machine learning works

4. RECOMMENDATION

The result of this study shows that MLP WOA model can be described as a suitable model for the reduction in financial losses and also helps Financial Institutions make better lending decisions. It can be used as a hybrid or robust technique with other machine learning algorithms to further strengthen the model. Implementing MLP WOA Adam could enhance exploration of weight and bias. It can help escape local minima and achieve objective function which is minimized loss or cost function. Further studies can be done through feature engineering techniques for better capturing of relevant information from data to achieve improved model performance.

5.CONCLUSION

This study explored the effectiveness of sampling techniques like smote ENN to handle the imbalanced nature of financial datasets by some level of balancing the minority and majority class for better model performance. The WOA algorithm was used to optimize the initial parameters of the MLP model. This involves iteratively updating the values of the parameters until the MLP model achieves the best and acceptable possible performance during training using Adam optimzer to achieve convergence. The MLP WOA Adam was able to minimise loss function, achieve higher model accuracy.

REFERENCES

[1] Khemakhem, S., Said, F.B., Boujelbene, Y. (2018). Credit risk assessment for unbalanced datasets based on data mining, artificial Neural Network and support vector machines. Journal of Modelling in Management, 13(4): 932-951. https://doi.org/10.1108/JM2-01-2017-0002

[2] Elhoseny, M., Metawa, N., Sztano, G., El-Hasnony, I.M. (2022). Deep learning-based model for financial distress prediction. Annals of Operations Research, 1-23.

https://doi.org/10.1007/s10479-022-04766-5

[3] Talaei Khoei, T., Ould Slimane, H., Kaabouch, N. (2023). Deep learning: Systematic review, models, challenges, and research directions. In Neural Computing and Applications, Springer Science and Business Media Deutschland GmbH, 35(31): 23103–23124. https://doi.org/10.1007/s00521-023-08957-4

[4] Rana, N., Latiff, M.S.A., Abdulhamid, S.I.M., Chiroma, H. (2020). Whale optimization algorithm: A systematic review of contemporary applications, modifications and developments. Neural Computing and Applications. Springer London, 32: 16245-16277. https://doi.org/10.1007/s00521-020-04849-z

[5] Lederer, J. (2021). Activation functions in artificial neural networks: A systematic overview. arXiv preprint arXiv:2101.09957. https://doi.org/10.48550/arXiv.2101.09957

[6] Rojas, M.G., Olivera, A.C., Vidal, P.J. (2022). Optimising Multilayer Perceptron weights and biases through a Cellular Genetic Algorithm for medical data classification. Array, 14: 100173. https://doi.org/10.1016/j.array.2022.100173

[7] Cai, X., Tyagi, K., Manry, M. T. (2011). Training multilayer perceptron by using optimal input normalization. In 2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011), Taipei, Taiwan, pp. 2771-2778. https://doi.org/10.1109/FUZZY.2011.600764

[8] Ravaut, M., Gorti, S. (2018). Gradient descent revisited via an adaptive online learning rate. arXiv preprint arXiv:1801.09136. https://doi.org/10.48550/arXiv.1801.09136

[9] Kornfeld, S. Predicting Default Probability in Credit Risk using Machine Learning Algorithms Degree Projects in Financial Mathematics (30 ECTS credits). Master's Programme in Applied and Computational Mathematics KTH Royal Institute of Technology year 2020 Supervisor at SEB: Eric Emtander Supervisor at KTH: Tatjana Pavlenko Examiner at KTH: Tatjana Pavlenko. https://www.kth.se/sci.

[10] Shi, S., Tse, R., Luo, W., D'Addona, S., Pau, G. (2022). Machine learning-driven credit risk: A systemic review. Neural Computing and Applications, 34(17): 14327-14339. https://doi.org/10.1007/s00521-022-07472-2

[11] Zafar, A. (2023). Stochastic gradient descent-A super easy complete guide! https://www.mltut.com/stochastic-gradient-descent-a-super-easy-complete-guide/.

[12] Li, M., Zhang, T., Chen, Y., Smola, A.J. (2014). Efficient mini-batch training for stochastic optimization. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, US, pp. 661-670. https://doi.org/10.1145/2623330.2623612

[13] Terven, J., Cordova-Esparza, D.M., Ramirez-Pedraza, A., Chavez-Urbiola, E. A., Romero-Gonzalez, J.A. (2023). Loss functions and metrics in deep learning. Machine Learning. https://doi.org/10.48550/arXiv.2307.02694

[14] Brenes, R.F., Johannssen, A., Chukhrova, N. (2022). An intelligent bankruptcy prediction model using a multilayer perceptron. Intelligent Systems with Applications, 16: 200136. https://doi.org/10.1016/j.iswa.2022.200136

[15] Bushaev, V. (2018). Adam-latest trends in deep learning optimization. Towards Data Science. https://towardsdatascience.com/adam-latest-trends-in-deep-learning-optimization-6be9a291375c.

[16] Hassan, E., Shams, M.Y., Hikal, N.A., Elmougy, S. (2023). The effect of choosing optimizer algorithms to improve computer vision tasks: A comparative study. Multimedia Tools and Applications, 82(11): 16591-16633. https://doi.org/10.1007/s11042-022-13820-0

[17] Chen, W., Yang, K., Yu, Z., Shi, Y., Chen, C.L. (2024). A survey on imbalanced learning: Latest research, applications and future directions. Artificial Intelligence Review, 57(6): 137. https://doi.org/10.1007/s10462-024-10759-6

[18] Alamsyah, A.R.B., Anisa, S.R., Belinda, N.S., Setiawan, A. (2022). SMOTE and Nearmiss methods for disease classification with unbalanced data. Proceedings of The International Conference on Data Science and Official Statistics, 2021(1): 305-314. https://doi.org/10.34123/icdsos.v2021i1.240

[19] Kotsiantis, S., Kanellopoulos, D., Pintelas, P. (2006). Handling imbalanced datasets: A review. GESTS International Transactions on Computer Science and Engineering, 30.

[20] Gunantara, N., Nurweda Putra, I.D.N. (2019). The characteristics of metaheuristic method in selection of path pairs on multicriteria ad hoc networks. Journal of Computer Networks and Communications, 2019(1): 7983583. https://doi.org/10.1155/2019/7983583

[21] de Almeida, B.S.G., Leite, V.C. (2019). Particle swarm optimization: A powerful technique for solving engineering problems. Swarm Intelligence-Recent Advances, New Perspectives and Applications, pp. 31-51.

[22] Shami, T.M., El-Saleh, A.A., Alswaitti, M., Al-Tashi, Q., Summakieh, M.A., Mirjalili, S. (2022). Particle Swarm Optimization: A comprehensive survey. IEEE Access, 10: 10031-10061. https://doi.org/10.1109/ACCESS.2022.3142859

[23] Liang, X., Zhang, Z. (2022). A whale optimization algorithm with convergence and exploitability enhancement and its application. Mathematical Problems in Engineering, 2022: 2904625. https://doi.org/10.1155/2022/2904625

[24] Infant Cyril, G.L.S., Ananth, J.P. (2023). Whale social optimization driven deep Recurrent Neural Network for loan eligibility prediction. Concurrency and Computation: Practice and Experience, 35(3): e7510. https://doi.org/10.1002/cpe.7510

[25] Gao, L., Xiao, J. (2021). Big data credit report in credit risk management of consumer finance. Wireless Communications and Mobile Computing, 2021: 1-7. https://doi.org/10.1155/2021/4811086

[26] Mohammadi, N., Zangeneh, M. (2016). Customer credit risk assessment using artificial neural networks. International Journal of Information Technology and Computer Science, 8(3): 58-66. https://doi.org/10.5815/ijitcs.2016.03.07

[27] Seitshiro, M.B., Mashele, H.P. (2020). Assessment of model risk due to the use of an inappropriate parameter estimator. Cogent Economics & Finance, 8(1): 1710970. https://doi.org/10.1080/23322039.2019.1710970

[28] Lucy, O.N., Anireh, V.I.E., Bennett, E.O. (2022). A Model for Credit Risk Analysis Using Artificial Neural Network. https://www.globalscientificjournal.com/.

[29] Nourmohammadi-Khiarak, J., Feizi-Derakhshi, M.R.,

Razeghi, F., Mazaheri, S., Zamani-Harghalani, Y., Moosavi-Tayebi, R. (2020). New hybrid method for feature selection and classification using meta-heuristic algorithm in credit risk assessment. Iran Journal of Computer Science, 3: 1-11. https://doi.org/10.1007/s42044-019-00038-x

[30] Wang, C., Wang, Y., Ye, Z., Yan, L., Cai, W., Pan, S. (2018). Credit card fraud detection based on whale algorithm optimized BP Neural Network. In 2018 13th International Conference on Computer Science & Education (ICCSE), Colombo, Sri Lanka, pp. 1-4. https://doi.org/10.1109/ICCSE.2018.8468855

[31] Rani, V.U., Saravanan, V., Tamilselvi, J.J. (2023). A hybrid grey wolf-meta heuristic optimization and random forest classifier for handling imbalanced credit card fraud data. International Journal of Intelligent Systems and Applications in Engineering, 11(9s): 718-734. https://www.ijisae.org/index.php/IJISAE/article/view/3220

[32] Infant Cyril, G.L.S., Ananth, J.P. (2023). Whale social optimization driven deep Recurrent Neural Network for loan eligibility prediction. Concurrency and Computation: Practice and Experience, 35(3): e7510. https://doi.org/10.1002/cpe.7510

[33] Tian, Z., Fong, S. (2016). Survey of meta-heuristic algorithms for deep learning training. Optimization Algorithms-Methods and Applications, 195-220. https://doi.org/10.5772/63785

[34] Safi, S.A.D., Castillo, P.A., Faris, H. (2022). Cost-sensitive metaheuristic optimization-based neural network with ensemble learning for financial distress prediction. Applied Sciences, 12(14): 6918. https://doi.org/10.3390/app12146918

[35] Mavrovouniotis, M., Yang, S. (2015). Training Neural Networks with ant colony optimization algorithms for pattern classification. Soft Computing, 19: 1511-1522. https://doi.org/10.1007/s00500-014-1334-5

[36] Crypto1. (2020). Gradient descent algorithm: How does the gradient descent algorithm work in machine learning? https://www.analyticsvidhya.com/blog/2020/10/how-does-the-gradient-descent-algorithm-work-in-machine-learning/.

[37] Alyafi, B., Tushar, F.I., Toshpulatov, Z. (2018). Cyclical learning rates for training neural networks with unbalanced datasets. Jmd in Medical Image Analysis and Applicationspattern Recognition Module, 1-4.

NOMENCLATURE

AWOA	Adaptive Whales Optimization Algorithm
ENN	Edited Nearest Neighbour
GA	Genetic Algorithm
NN	Neural Network
PSO	Particle Swarm Optimization
WOA	Whales Optimization Algorithm