Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Financial Fraud Detection Using AEO-DMOA Based 1D-FRCNN Model with Effective Feature Selection Technique

B Jyothi [1], V Rama Krishna [2], S. Venkatramulu [3], Urshottam J. Assudani [4], Kalyanapu Srinivas [5]
[1]*School of engineering, Anurag University, Hyderabad, India*
[2] *School of engineering, Anurag University, Hyderabad, India*
[3]*Department of CSE(Networks), Kakatiya Institute of Technology and Science, Warangal, India*
[4] *School of Computer Science and Engineering, Ramdeobaba University, Nagpur, India,*
[5] *Department of CSE, Vaagdevi Engineering College, Warangal, India*

ARTICLE INFO

Received: 26 Dec 2024

Revised: 14 Feb 2025

Accepted: 22 Feb 2025

ABSTRACT

As a global issue, financial fraud has severely hampered the steady expansion of financial markets. However, if the ratio of legitimate businesses to fraudulent ones is particularly large, it might be difficult to spot fraud in a dataset. Therefore, solutions have been created to help stakeholders make better decisions through the use of intelligent financial statement fraud detection. However, most existing methods primarily take into account the numerical data found in financial statement ratios, while textual data, notably Chinese-language remarks on the subject, has been underutilized for classification. Therefore, the purpose of this study is to create a model-based scheme for financial fraud detection. In this study, the best features from the preprocessed data are chosen using Hybrid Enhanced Glowworm Swarm Optimization (HEGSO). Hybrid optimization (AEO-DMOA) is then used to regulate the best values for the network's hyper-parameters, including its learning rate, epochs, momentum, and batch sizes, before a one-dimensional faster region-based convolutional neural network (1D-FRCNN) is recommended for classification. Dwarf Mongoose Optimization Algorithm (DMOA) and Artificial Ecosystem-based Optimization (AEO) are combined in this model to create an effective algorithm that strikes a better balance between exploration and exploitation. Accuracy, precision, recall, and f1-score were optimized to 99.10%, 98%, 96%, and 97%, respectively, as shown by the assessment of the study effort. For fraud detection challenges, the suggested model performs improved than state-of-the-art learning procedures.

Keywords: Financial fraud; Hybrid Enhanced Glowworm Swarm Optimization; Dwarf Mongoose Optimization Algorithm; one-dimensional faster region-based convolutional neural network; Feature Selection.

1. INTRODUCTION

Many enterprises, especially those in rapidly developing nations like China, have turned to the securities market's recent expansion as a means to increase their financial base and the scope of their operations through listing. Financial market growth has been accompanied by a rise in fake financial reports, which have had a significant negative influence on capital markets and resulted in massive losses for shareholders [1, 2]. Capital markets are vulnerable to fraud because it erodes confidence between corporations, gatekeepers, and market participants [3]. Capital markets would suffer greatly without detecting financial fraud. The tremendous effects of the financial fraud at Luck in Coffee have recently brought the topic of financial fraud back to the forefront. The financial statement is the backbone of the annual report since it shows the current and projected financial health of the firm [4]. However, it is time-consuming and laborious to manually go through a financial statement for signs of accounting fraud or other types of financial crime. Analytical techniques, ratio analysis, and score propagation via an auction network are just some of the methods that have been used by academics in recent years to increase the quality and efficiency of fraud detection using financial statements [5]. However, most existing studies provide too many fraud risk indicators and are unable to effectively and quickly identify such scams [6]. It has become of crucial necessity to determine how to identify certain major fraud indicators useful for finding frauds and to rate the relevance of those fraud aspects. Among them are the Z-score, AR, stock on hand, GM, and other similar metrics [7]. Furthermore, several other financial proportions [8] are also utilised for fraud finding.

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

As information technology evolves, so too do the many techniques used to identify financial fraud. As was said in the preceding subsection, [9] statistical approaches have been employed to categorise and identify frauds, with financial pointers serving as the central and crucial role in prediction. Meanwhile, research shows that using cutting-edge technology to enhance information management productivity and quality is feasible, especially given the prevalence of combining structured and unstructured data. It has been widely utilised to solve complicated issues in a broad range of arenas [10], and as a strong modelling tool, it can swiftly and efficiently disclose the realities that are hidden in enormous volumes of data. Recently, deep learning has made its way into the field of finance research, where it has already shown promising results in predicting the financial risk likelihood of businesses. Recent years have seen the rise in popularity of data mining techniques as a powerful method for uncovering previously unknown relationships and patterns in massive data sets. Predictive and classifying technologies have been investigated by several scholars in an effort to combat fraud [11, 12]. Logistic regression, and support vector networks are all examples of suitable models [13]. Still, most contemporary financial fraud detection studies focus solely on statements while disregarding the textual data present in the annual report of the listed firm, particularly the remarks written in Chinese that address the topic at hand. In addition, conventional machine learning approaches would have a hard time distinguishing fake financial data from genuine data in practice due to purposeful concealment [14]. However, machine learning methods are also employed to spot financial scams, and there are currently no very effective fraud detection systems on the market [15].

In this study, we examine the use of ML, and deep learning in particular, to detect fraudulent transactions involving stolen credit cards in the financial sector. The SVM is a supervised ML approach useful for problems involving the classification of data [16]. SVM is better than other classifiers since it can handle both linear vectors. Credit card fraud was initially detected using neural networks [17]. As a result, this area of study has become a focal point for the DL method. The 1D-FRCNN method is used to analyse the financial fraud detection dataset in this study. Furthermore, AEO-DMOA enhances classification precision by selecting parameters appropriately. In addition, the HEGSO model ensures that the classifier has only used the most pertinent characteristics for categorization. The remainder of the paper shadows this structure: The relevant literature is presented in Section 2, and the suggested model is summarised and deliberated in Section 3. In Section 4, we display the experimental examination of the suggested model using the currently available methods. The task is brought to a close in Section 5.

2. RELATED WORKS

Using a deep learning approach and supervised deep learning techniques, Fania and Abbasi Mehr [18] offer a two-stage scheme for detecting fraudulent transactions. It was found through experimental evaluations that the suggested method enhances classifiers. As a result of the deep Autoencoder's data transformation, the used deep learning classifiers greatly outperform their baseline classifiers across the board. Moreover, models developed using deep Autoencoder outperform models developed with the dataset collected by (PCA) and preexisting models.

In this study, a new strategy, called a Shepherd (SSPO-based DRN), is built for identifying fraud. This was introduced by Ganji et al. Algorithms for Political Optimisation (PO) and Shuffled Shepherd Optimisation (SSOA) are combined to create SSPO. In order to normalise the data for efficient detection, the quantile normalisation model is a useful preprocessing tool. Fisher's score and class information gain both do a good job of narrowing down the elements that are needed. To enhance detection performance, data augmentation is used to increase the amount of the data. Credit card fraud finding uses a Deep Residual Network (DRN) that is trained using an SSPO method. With a sensitivity of 0.9279, specificity of 0.9023, and accuracy of 0.9120, the SSPO-based DRN method showed substantial improvement over conventional testing methods.

Using Long Short-Term Memory (LSTM) DNNs for sequential data modelling, Fakiha [20] proposes a forensic model for fraud. In this research, we test whether an LSTM-attention algorithm can pick out the key events from a sequence of inputs to reliably forecast fraudulent ones. Selecting the most appropriate predictive features, uniform manifold enhances the LSTM-attention model's efficacy. The findings demonstrate the efficacy of using LSTM-attention algorithms for forensic credit card fraud detection. The paper's original contribution is its use of an LSTM-attention approach to effectively identify cases of credit card fraud, therefore validating the model's potential for preventing fraudulent transactions at financial institutions.

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Different deep learning models, such as those developed by Wei et al. [21], are compared and contrasted for their effectiveness in identifying fraudulent chargebacks in the gaming industry. Traditional models are also compared and contrasted, including decision trees, k-nearest neighbours, and support forests. According to the evaluations, the deep learning models had Matthew's correlation coefficients between 0.84 and 0.97. In addition, the experimental results from this study show that the models based on long memory networks perform better than the more conventional machine learning replicas. In total, this study considers the practical viability by determining if there is a considerable rise in time expenses by estimating the time overhead of a single transaction. While deep learning models aren't as effective as more classic machine learning approaches, they're still good enough to help online gaming businesses cut down on their losses.

Successful methods for identifying and forecasting credit card fraud have been suggested by Bakhtiari et al. [22]. Machines are two such techniques. Predicting issues with any degree of precision is crucial in this context. In this research, we compare and contrast several Ensemble Learning techniques, such as gradient boosting, and their combinations using averaging techniques (Simple and Weighted Averaging approaches). When used together, these approaches improve efficiency and accuracy while decreasing mistake rates. The best results were obtained by combining LightGBM and LiteMORT via weighted averaging, as measured by Area, with respective values of 95.20, 90.65, 91.67, 92.79, and 99.44.

The Detection scheme for CCF (CCFD) was designed using Machine Learning (ML) approaches by Karthika and Senthilselvi [23], though these ML didn't provide much efficiency. Therefore, modern-day (DL) is employed in the field of CCFD to address these problems with ML. The problems with CCFD are addressed in this study by training a one-dimensional Dilated (DCNN) using spatial and temporal information. In this case, we use the dilated convolutional layer (DCL) to enhance the standard CNN model. Under-sampling and over-sampling methods are used to correct the imbalance. Experiments are conducted on three datasets with respect to a sum of parameters, and the results are compared to a current CNN model. The simulation consequences showed that the suggested DCNN perfect with the sampling strategy outperformed CNN (94.44%) on the same tiny card database.

Financial fraud discovery using a deep learning approach based on convolution network technology was proposed by AR, S. [24]. The goal of this model is to enhance the effectiveness and efficiency of current detection techniques when applied to massive datasets. In addition, we implement a flower pollination optimisation (FPO) procedure for feature selection, which takes into account the unintended consequences of making an optimal decision. The proposed model is evaluated by likening its results to those of state-of-the-art DL techniques on an actual dataset consisting of frauds. According to the experimental findings, the projected FPO_QCNN improves accuracy for the credit card data sets.

2.1. Research Gap

There are two chief schools of supposed when it comes to the study of spotting financial fraud. One method is keeping an eye out for any red flags that might indicate fraud. The opportunity, motivation, and rationalisation are the "three legs" of the Fraud Triangle, which together raise the likelihood of fraud. Based on this idea, researchers investigate the factors that lead to fraud, such as executive or business traits, external pressures, and personal desires or needs [18–20]. Another method involves spotting fraudulent activity by analysing raw financial data, indicators, textual information, and other fraud outcomes [21, 22]. Data-driven methods typically exclude the contribution of human insight. In a nutshell, the current literature treats businesses as isolated entities, which overlooks their interdependencies. However, businesses frequently exist inside an intricate ecological web. As a result, contextual clues may have an unanticipated impact on uncovering financial fraud [23]. This suggests that financial fraud detection might benefit from a greater focus on spotting fraud based on the nature of the connection. Due to their relative safety and ease of manipulation, related-party transactions (RPTs) have become the preferred method of cashing out the illicit gains from financial crime.

2.2. Objective of the Research Work

However, deep learning's superior fraud-detection performance comes at the expense of high computational complexity and subpar classification accuracy. Selecting DL hyper-parameters optimally is necessary to avoid the high computing complexity. Misclassification of fraud is another risk that arises when a large number of

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

characteristics are used. Meta-heuristic algorithms do optimum feature selection, which is crucial for preventing this kind of misclassification.

PROPOSED SYSTEM

The research work uses the China bank statement for the prediction process, and a brief explanation of the work is mentioned in the subsequent sub-sections. Figure 1 presents the working flow of the perfect

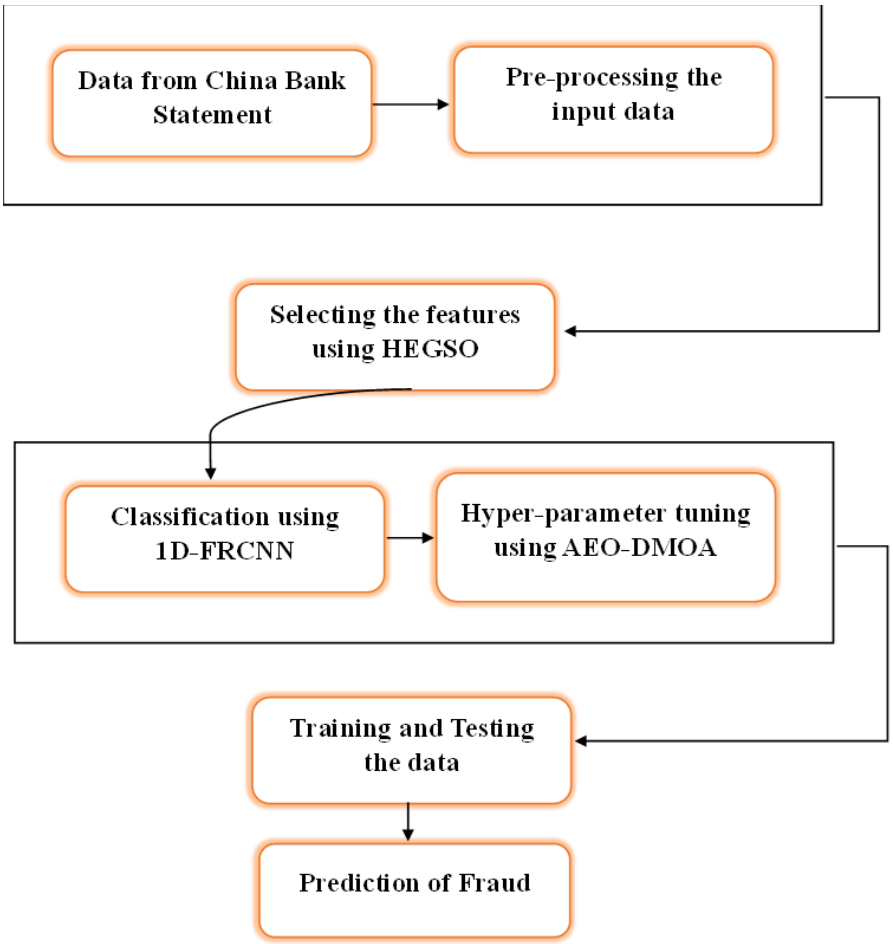

Figure 1: Working Flow of the Proposed Model

2.3. Dataset Description

A large Chinese online financial service company was the source of the initial experimental dataset. The cleaned-up dataset includes 192586 data samples and 4375 fraudulent data samples. There are nearly 60 different variables in the dataset, including things like the starting balance, financial health, currency, income, payment history, sales status, and more. Not all learning models are split into 8 subsets to protect the privacy of sensitive data. There is always around four times as much testing data as training data.

2.3.1. Data pre-processing

Data dividing should all be performed before smearing the perfect to the dataset.

2.3.2. Data Validation

This step is used to validate the data within the dataset, value, empty values, or quantity.

2.3.3. Normalization

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

To produce an accurate result, the model rescales the variables to be between -1 and 1. This step is required to convert the shared scale without misrepresenting the difference in value ranges or losing data. Equation 1 is used in the normalization procedure.

$$x(i) = \frac{x(i)-\overline{x}}{s(x)} \quad (1)$$

2.3.4. Dataset models divide

The division of data samples is critical for obtaining a realistic assessment of performance. The projected model uses 70% of the dataset for training and the residual 30% for testing.

2.3.5. Feature Scaling

The range, unit, and magnitude of most characteristics in a real-world dataset will be different. When the magnitude of one aspect is far larger than the others, it tends to overpower the others. Therefore, classification algorithms require raw data to be scaled to reduce the influence of different quantitative units [25]. This study rescaled the characteristics between 0 and 1 using the MinMaxScaler method. This method's strength lies in its resistance to outliers; it employs statistical methods that have no bearing on the variance of the data, as shown in Equation (2).

$$x' = \left(x - \frac{\min(x)}{\max(x)} - \min(x)\right) \quad (2)$$

Equation (2) above shows that the original value (x) is represented by the numerator, the scaled value (x'), the maximum possible feature value (max) is indicated by the denominator, and the minimum possible feature value (min) is represented by the denominator. Time is saved while using MinMaxScaler since it scales input data while retaining its sparsity [26].

2.3.6. Feature Selection Using Hegso Model

After the data has been cleaned and organised, the HEGSO model is used to extract meaningful insights [27]. The conventional GSO method is based on glow-worm activity. This originates from the nighttime behaviours of the glow worm. The luciferin in the glow worms seems to allow them to communicate with one another. A greater amount of luciferin will result in a brighter glow worm. As a result, the glow worm is heading in that direction. Updates to luciferin, movement, and the neighbourhood and update separate stages.

The first update step of the luciferin manufacturing algorithm is related to the generation of luciferin. The amount of luciferin is proportional to how well its current location in the space of objective functions meets its needs. The second stage involves action. At this point, the glowworm decides to head in the direction of a neighbour with a higher luciferin value than its own using probabilistic techniques. The final stage involves making adjustments to the range of the adaptive neighbourhood in order to detect the overlapping peaks. HEGSO is used as the optimisation algorithm. The HEGSO algorithm is used in conjunction with the crossover and mutation genetic algorithms in the proposed study, which is why it is called a hybrid algorithm. After a path is found between the nodes of the car and the road segment's beginning and finish, optimisation may commence. Here is what the HEGSO algorithm looks like in code:

Algorithm 1 Projected HEGSO Procedure
1: $\boldsymbol{Procedure\ HEGSO}$
2: $\boldsymbol{begin}$: $\boldsymbol{Input}$: $\boldsymbol{X_i, Q, z;\ N_{it};\ I_o;\ r_o}$. **3**: $\boldsymbol{Determine\ fitness\ function}$
4: $\boldsymbol{while\ t\ <\ N_{iy}\ do}$
5: $\boldsymbol{for\ each\ glowworm\ do}$
6: $\boldsymbol{l_j(n+1) = (1-\rho)l_j(n) + \beta J_j(n+1)}$
7: $\boldsymbol{x_j(n+1) = x_j(n) + Z\left(\frac{x_k(n)-x_j(n)}{\|x_k(n)-x_j(n)\|}\right)}$
8: $\boldsymbol{r_d(n+1) = min(r_s, max(0;\ r_d(n) + B(n_e - \|N_j(n)\|))}$
9: $\boldsymbol{end\ for}$

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

10: ***end while***
11: ***Perform the crossover and mutation*** **12**: ***Return*** X_{best}

13: ***end Procedure***

Step 1: Eq. 3 allows for the specification of a step size, a maximum sum of iterations N_it, an initial value for luciferin I_o, an initial value for the radial and a time instance n, all of which serve to initialise the 'Eearch Agents' (EA), each of which is represented by a single glowworm X.

$$X = \{x_1, x_2, \dots, x_Q\} \tag{3}$$

Step 2: Subtracts the fitness function applying the Eq. 4.

$$F = \sum_{t=1}^{n} \frac{L}{EV_t(V_i)} \tag{4}$$

The length of the road, represented by L, the regular speed of the cars on the road, signified by EV_t, and the individual car's speed, marked by V_i, are all shown here.

Step 3: The next stage, known as the luciferin phase, follows. Each of the new SAs is calculated in the luciferin phase with the help of Eq. 5.

$$I_j(n+1) = (1-\rho)l_j(n) + \beta J_j(n+1) \tag{5}$$

where $l_j(n)$ I_j (n+1) suggests the luciferin value of the SA at time j, and I_j (n) implies the luciferin value of the SA at time n. (n+1), J_j means the luciferin decay constant has a value between 0 and 1 and implies the fitness function.

Step 4: Each new SA's goal functions are evaluated with the fitness function discussed in step 2.

Step 5: The SA beams that travel to the nearby glowworm have their foundation here. The SA's motion may be calculated from the geometrical separation of the glowworms. At this point, the SA is making its way towards the area that was determined using a probabilistic approach. The glowworm's motion is described by the equation in Eq. 6.

$$x_j(n+1) = x_j(n) + Z\left(\frac{x_k(n) - x_j(n)}{\|x_k(n) - x_j(n)\|}\right) \tag{6}$$

Step 6: Here, we revise the decision interval. That is the SA's revised neighbourhood range. In this stage, we use the decision rule defined by Eq. 7.

$$r_d(n+1) = min\left(r_s, min\left(0, r_d(n) + B\left(n_e - |N_j(n)|\right)\right)\right) \tag{7}$$

where B is a constant and r_s is the greatest radius at which the glowworms can detect light. There are several exceptional SA. When luciferin concentrations are high inside the decision range, the neighbourhood range is updated to r_d (n+1), where n is an even number.

Step 7: Verify if the halting criterion has been met. The optimal solution is found if and only if the halting norm is satisfied. In the absence of a termination event, both genetic operators crossover explicitly and mutation is also applied. It uses a 2-point crossover. After this, certain genes undergo the mutation process. Crossover values are altered and then concatenated before being used.

2.4. Financial Fraud Detection using DL

Faster R-CNN [28] is presented based on the Fast RCNN that can act as a one-dimensional classifier in this work for the prediction process. The Fast R-CNN and the RPNs are combined to form the Faster R-CNN. The parameters for the two networks are similar. The Faster R-CNN architecture is portrayed in Figure 2.

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

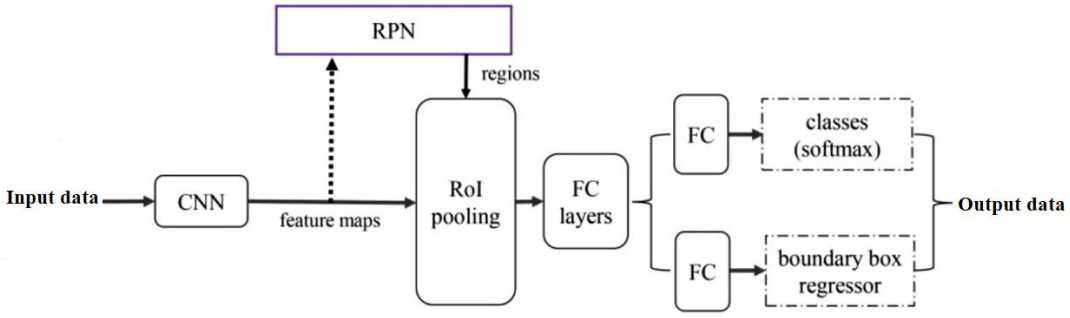

Figure 2: Construction of the projected model

In Faster R-CNN, the RPN is employed to speed up the application generation process for regions. Figure 8 is an illustration of the RPN. The last shared convolution layer's output convolution feature map is slid across a small network to create region proposals. A spatial window of the input feature map is linked to each of the nine detection boxes in this compact network. Each intermediate layer represents a mapping from a sliding window to a vector with less dimensions. Two FC levels then receive the vector as input. They are the regression (reg) layer and the classification (cls) layer. Since k region suggestions were made for each sliding window, the regression layer has 4k outputs to encode the locations of k boxes. The classification layer provides an estimate of the proposal's object/not-object likelihood as 2k scores. The FC layer is then utilised to determine the region categorization. Similarly, the layer is utilised to determine the deviation in location for each item proposal, leading to more precise suggestions for object recognition.

The loss function is used to quantify the discrepancy between the anticipated and observed values. The loss function was utilised to quantify the discrepancy between the RPN's foretold and ground-truth boxes during training. Definition of Loss Function

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i, p_i^*) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i, t_i^*) \quad (8)$$

The projected probability that anchor i is an item is denoted by p_i, and the ground-truth label is denoted by p_i*, where i is the anchor's index in the min-batch. Anchors are positive when p_i* is 1, and negative when p_i* is 0; the forecast bounding box is signified by the vector t_i; the ground-truth bounding box is represented by the vector t_i*; L_cls is the classification loss; and L_reg is the regression loss. Regression loss is zero when a positive anchor is used. Classification loss and regression loss are normalised by the constants N_cls and N_reg, respectively. The counterbalance is equal to l. You may figure out the four parameterized coordinates by using the formula

$$\begin{bmatrix} t_x, t_y \\ , t_h \\ t_x^*, t_y^* \\ t_w^*, t_h^* \end{bmatrix} = \begin{bmatrix} \frac{x-x_a}{w_a}, \frac{(y-y_a)}{h_a} \\ log\left(\frac{w}{w_a}\right), \quad log\left(\frac{h}{h_a}\right) \\ \frac{x^*-x_a}{w_a}, \frac{(y^*-y_a)}{h_a} \\ log\left(\frac{w^*}{w_a}\right), \quad log\left(\frac{h^*}{h_a}\right) \end{bmatrix} \quad (9)$$

where (x,y) are the coordinates of the centre of the box being predicted, and (x_a,y_a) are the coordinates of the box. The width of the forecast box is denoted by w, that of the anchor box by w_a, and that of the ground truth box by w*. The projected height, h, the anchor height, h_a, and the ground-truth height, h*, are all denoted by the letter h. The three boxes (ground truth, anchor, and forecast) are depicted in Figure 3. This may be thought of as a regression of bounding boxes from some "anchor" box to some "ground truth" box in the vicinity.

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

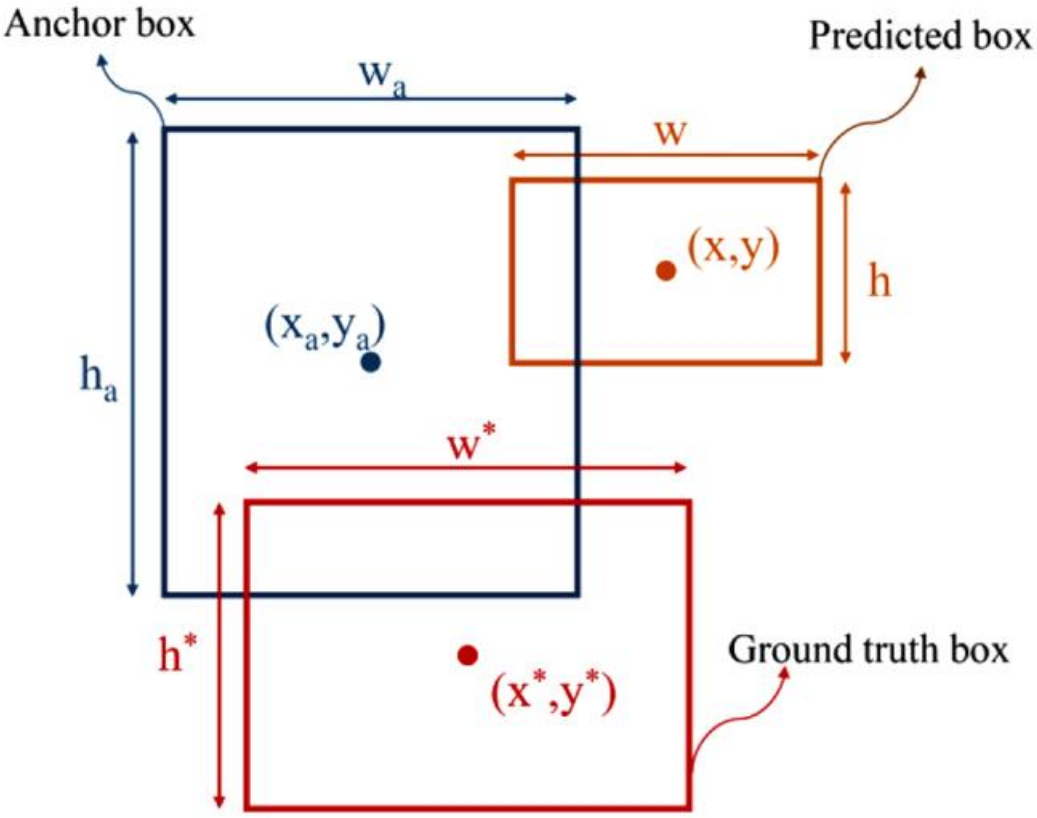

Figure 3: Geometry of Various boxes

The RPN is used to create region proposals in Faster R-CNN, whereas the Fast R-CNN is utilised to find and label objects. Both the RPN and the Fast R-CNN use the same convolutional features. Figure 10 depicts the Fast R-CNN's overall structure. From the feature map, a region of interest (RoI) pooling layer can generate a feature vector of a predetermined size. Next, FC layers are fed the fixed-size feature vector in order to determine where the anticipated boxes should go and how to categorise their contents.

The two last layers of a Fast R-CNN are the softmax and regression layers, as was previously explained. The probability, p, for all K+1 classes, is what is returned by the softmax layer. The regression layer provides the output of the box's bounds. When class is k, the bounding box's centre coordinates are (t_xk,t_yk), its width is (t_wk), and its height is (t_hk). A ground-truth class u and bounding-box regression goal v are assigned to each training region of interest (RoI). For the purpose of combined training for classification and bounding-box regression, we employ a loss L on each labelled RoI.

$$L(p,u,t^u,v) = L_{cls}(p,u) + \lambda[u \geq 1]L_{loc}(t^u,v) \quad (10)$$

in which $L_{cls}(p,u) = -\log p_u$ is log loss for true class u, l is -parameter, and L_{loc} is the loss for regression.

$$L_{loc}(t^u,v) = \sum_{i\in\{x,y,w,h\}} smooth_{L1}(t_i^u - v_i) \quad (11)$$

$$smooth_{L1}(x) = \begin{cases} 0.5x^2 & if|x| < 1 \\ |x| - 0.5 & otherwise \end{cases} \quad (12)$$

A practical four-step training method was devised to learn shared features between the two networks by alternating optimisation, allowing for the sharing of convolutional features. Before any region suggestions can be made, the RPN network must be educated. Second, training of Fast R-CNN commenced. Finally, we repeat the training of the RPN network. Fourth, a second round of training is performed using Fast R-CNN. There were two rounds of this training regimen. The reason for merely performing the drill twice was that increasing the sum of repetitions did not

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

significantly improve the training impact. Parameters for learning in a 1D-FRCNN model are shown in Table 1; the projected optimisation method does not lead to an optimal choice for the model's hyper-parameters.

Table 1: Details of Training parameters

Parameters	**Value**
Learning rate	0.01
Momentum	0.9
Weight decay	0.0005
The anchor scales for RPN	8, 16, 32
The anchor ratios for RPN	0.5, 1, 2

2.5. Hyper-parameter tuning using AEO-DMOA

In this subsection, the AEO-DMOA parameter-selection framework is presented [29, 30]. The main goal of the industrialised AEO-DMOA is to divide the total number of repetitions into two parts, with AEO being used in the first part to explore the entire search area for optimising the search area boundaries, and DMOA being used in the second part to exploit the AEO-optimized search area for solution. Both approaches benefit from this sequential implementation since it reduces the likelihood of becoming stuck in local optimums and keeps exploration and exploitation in reasonable proportions.

First, we set some default values for the AEO and DMOA hyper-parameters, such as the maximum number of iterations (T) and the number of possible solutions (N). Each feature dimension's upper (UB) and lower (LB) bounds in the specified search space are determined. According to the method outlined in [29], all N possible solutions are uniformly initialised in the interval [1, 1]. Each potential answer is assigned a Fitness Value (FV) by the 1D-FRCNN model. As a result, the applicant solution with the minimal FV is the one that is remembered as the best overall. Here's how to figure out the FV:

$$FV = \lambda \times (1 - AC) + (1 - \lambda) \times \frac{SF_i}{M} \quad (13)$$

Where AC is the accuracy of 1D-FRCNN, SF_i is the sum of features picked by the candidate solution, and M is the dimensionality of the data used to make the feature selections. In this study, it is set at 0.99, which is in the middle of the range from 0 (no relevance to classification accuracy) to 1. To determine the SF, we first determine a threshold for the solution candidate's present position:

$$SF_i = \begin{cases} 1 & if x_i > 0.5 \\ 0 & otherwise \end{cases} \quad (14)$$

The 0.5 threshold used to choose features is chosen empirically, so keep that in mind. Because the MH algorithm dynamically shifts the placements of crucial features above the threshold, feature selection is unaffected by the precise value of the threshold employed during training. When the threshold is set very high, the MH algorithm struggles to distinguish between useful and unnecessary characteristics. Therefore, a cutoff value of 0.5 is employed [31] as recommended by the literature. The AEO technique is used to begin the optimisation process. The Production phase of the AEO replaces the poorest candidate solution (the one with the biggest fitness value) with the best separate after taking UB and LB into account for the first time. Next, a random number (rand) between 0 and 1 is used to implement the remaining three steps of the AEO algorithm:

$$If\ rand \geq \frac{2}{3}, then\ Herbivore\ phase \quad (15)$$

$$elseif\ \frac{1}{3} \leq rand\ \frac{2}{3}, then\ Carnivore\ phase \quad (16)$$

$$else\ if\ rand \leq \frac{1}{3}, then\ Omnivore\ phase \quad (17)$$

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

To exhaust all possible options, the procedure is repeated indefinitely. Finally, the AEO Decomposition Phase [29] is implemented, which disassembles the failed candidate solutions into usable nutrients for producers to thrive on. At the end of each cycle, FV is determined for each potential answer. The latter is revised if a potential answer has a lower FV than the optimal answer globally. The procedure is then repeated until T2, where T is the maximum sum of repetitions.

Once the algorithm has run for (T2) times, it will transition from AEO to DMOA. To begin the (T2)th iteration, the counter C (which is initially set to 0 upon execution of DMOA) is incremented, and the AEO optimised candidate solution is updated using the Alpha is smaller than the babysitter exchange parameter L. If AvgSum_(t-1) is smaller than AvgSum_(t), then the current iteration's sleeping mound is larger than the previous iterations. Then, either the exploitation stage of DMOA is carried out in order to make use of the newly discovered solutions or the exploration stage is carried out. When the maximum number of DMOA iterations has been achieved, the fitness FV of the optimal solutions is figured, and the world's best solution is then updated. Babysitter groups are updated according to [30], and counter C is reset to 0 once it reaches the babysitter exchange parameter L. The procedure proceeds with an upgrade to the current best answer worldwide.

When a certain threshold, T, is reached, optimisation halts. After a sufficient number of repetitions, the best possible global solution is selected. Features with locations greater than 0.5 are added to the 1D-FRCNN, as shown in Eq. 14. Table 2 explains how to set the suggested classifier's parameters optimally.

Table 2. Training parameters after AEO-DMOA.

Parameters	**Value**
Learning rate	0.001
Momentum	0.7
Weight decay	0.0003

.

3. RESULTS

A cluster of 30 identical machines is used to conduct experiments in groups, with one machine serving as the master worker node. There are 8 physical processor cores and 64 GB of RAM in each system. CentOS 7 is the OS, and it has the Java SE Development Kit 10 and Scala 2.12 installed.

3.1. Performance Metrics

Some of the metrics used to evaluate performance included F1 and Cohen's kappa. We also used the k-fold cross-validation test in our analysis. Our performance measurements are computed with Equations (18-22), whereas training and prediction times were the measures.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (18)$$

$$Precision = \frac{TP}{TP+FP} \quad (19)$$

$$Recall(True\ positive\ Rate) = \frac{TP}{TP+FN} \quad (20)$$

$$F1Score = \frac{2TP}{2TP+FP+FN} = \frac{2\times Precision\times Recall}{Precision+Recall} \quad (21)$$

$$Cohen\ Kappa\ Score = \frac{Accuracy-P_{random}}{1-P_{random}} \quad (22)$$

In a TP (True Positive), both the prediction and the intended outcome are accurate. The objective may be off, but the forecast may be spot on. What we're talking about here is TN. An FP (False Positive) occurs when a forecast is true when the target is false, and a FN (False Negative) occurs when a forecast is false while the target is true. For the purpose of fraud detection, the current models are taken into account from [18-24]. Tables 3, 4, and 5 show the average results obtained by applying the current models to our data.

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Table 3: Validation Analysis of Projected feature selection model

Model	Accuracy	Precision	Recall	F1-score	Cohen Kappa score
Firefly	88.26	0.901	0.893	0.90	0.8112
FPO	90.71	0.916	0.912	0.913	0.8396
SSOA	91.58	0.923	0.920	0.915	0.8533
GSO	93.03	0.935	0.929	0.932	0.8797
HEGSO	95.16	0.952	0.943	0.942	0.9009

The validation analysis of the projected feature selection model is shown in Table 3 above. There are various models to evaluate the performances in the comparison analysis. The Firefly model initially achieved an accuracy of 88.26, a precision rate of 0.901, a recall of 0.893, an F1-score of 0.90, and finally Cohen Kappa score of 0.8112, all of which were respectively. After that, the FPO model achieved the following results: accuracy of 90.71, precision rate of 0.916, recall of 0.912, F1-score of 0.913, and finally, Cohen Kappa score of 0.8396. The SSOA model then achieved accuracy of 91.58, precision proportion of 0.923, recall of 0.920, F1-score of 0.915, and finally Cohen Kappa score of 0.8533, all of which are respective values. The GSO model then achieved an accuracy of 93.03, a precision rate of 0.935, a recall of 0.929, an F1-score of 0.932, and finally, Cohen Kappa score of 0.8797, all of which are respective values. The HEGSO model then achieved an accuracy of 95.16, precision degree of 0.952, recall of 0.943, F1-score of 0.942, and finally, Cohen Kappa score of 0.9009, all of which are respective values.

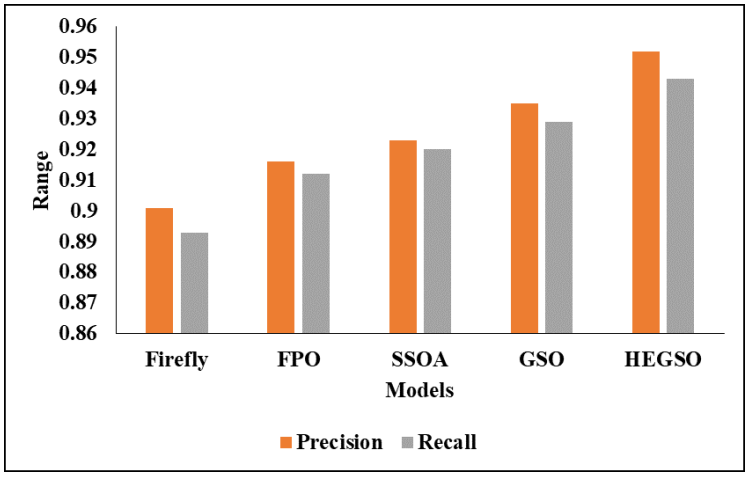

Figure 4: Analysis of Feature Selection Model

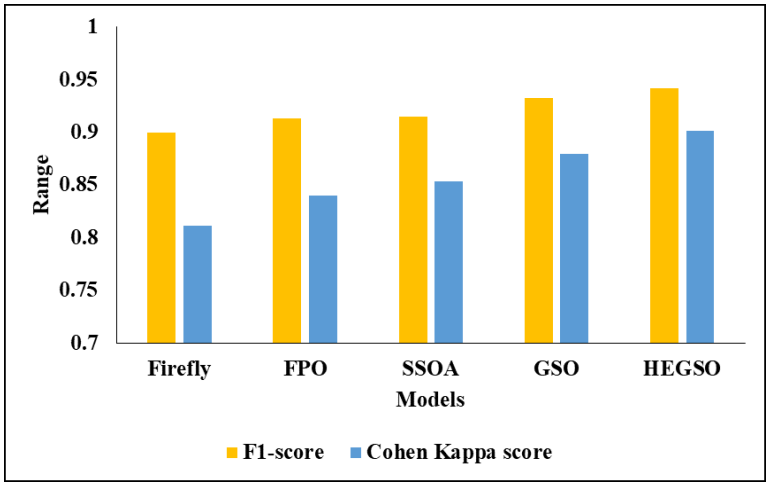

Figure 5: Graphical Representation of proposed feature selection

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

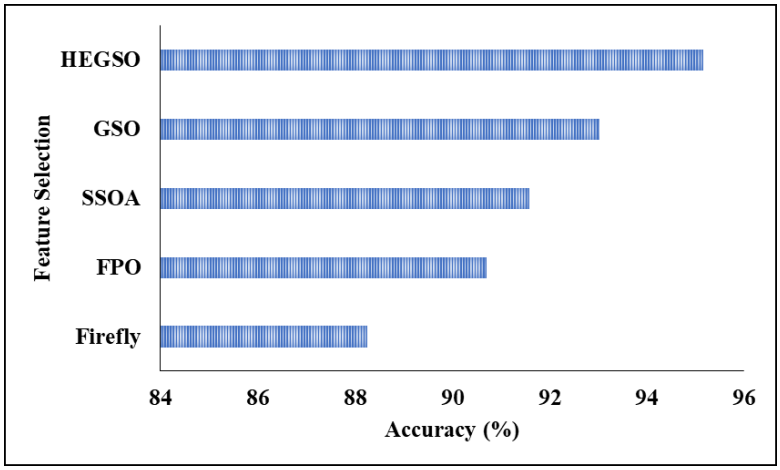

Figure 6: Accuracy analysis

Table 4: Analysis of Proposed Classifier without Hybrid Optimization

Methods	**Accuracy**	**Precision**	**Recall**	**F-score**	**Cohen Kappa score**
DBN	91.85	0.89	0.89	0.90	0.880
AE	92.69	0.90	0.90	0.91	0.893
CNN	94.85	0.92	0.91	0.93	0.904
RCNN	96.99	0.93	0.92	0.94	0.913
1D-FRCNN	**97.14**	**0.97**	**0.94**	**0.95**	**0.938**

The analysis of the proposed classifier without hybrid optimisation is shown in Table 4 above. In this analysis, the DBN model achieved an accuracy of 91.85, a precision rate of 0.89, a recall of 0.89, an F-score of 0.90, and finally, Cohen Kappa score of 0.880, all of which were achieved in that order. The accuracy of the AE model was 92.69 per cent, as were the precision rate of 0.90, recall of 0.90, F-score of 0.91, and Cohen Kappa score of 0.893. The Cohen Kappa score for the CNN model was 0.904, with accuracy of 94.85, precision rate of 0.92, recall of 0.91, F-score of 0.93, and recall of 0.91, correspondingly. The accuracy of the RCNN model was 96.99, the precision rate was 0.93, the recall was 0.92, the F-score was 0.94, and the Cohen Kappa score was 0.913. 1D-FThe RCNN model achieved an accuracy of 97.14 per cent, precision degree of 0.97, recall of 0.94 per cent, F-score of 0.95 per cent, and Cohen Kappa score of 0.938 per cent, all of which were achieved in that order.

Table 5: Analysis of Proposed Cla66ssifier with Hybrid Optimization

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**	**Cohen Kappa score**
DBN	93.07	0.91	0.90	0.91	0.891
AE	95.14	0.93	0.91	0.92	0.906
CNN	97.67	0.94	0.92	0.94	0.910
RCNN	98.18	0.96	0.94	0.95	0.935
1D-FRCNN	**99.10**	**0.98**	**0.96**	**0.97**	**0.948**

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

In the above table, 5 represent the Analysis of the Proposed Cla66ssifier with Hybrid optimization. In the analysis, the DBN model reached an accuracy of 93.07 and a precision rate as 0.91 and the recall rate as 0.90 and the F-score value as 0.91 and the Cohen Kappa score of 0.891, respectively. Another AE model reached an accuracy of 95.14 and a precision rate of 0.93 and a recall value of 0.91, an F-score value of0.92, and a Cohen Kappa score of 0.906, respectively. After the CNN model reached an accuracy of 97.67 and a precision rate of 0.94 and a recall value of 0.92, an F-score value of 0.94, and a Cohen Kappa score of 0.910, respectively. In next RCNN model reached an accuracy of 98.18 and a precision rate of 0.96 and a recall value of 0.94, an F-score value of 0.95, and a Cohen Kappa score of 0.935, respectively. Finally, the 1D-FRCNN model reached an accuracy of 99.10 and a precision rate of 0.98 and a recall value of 0.96 and the F-score value of 0.97 and the Cohen Kappa score of 0.948, respectively.

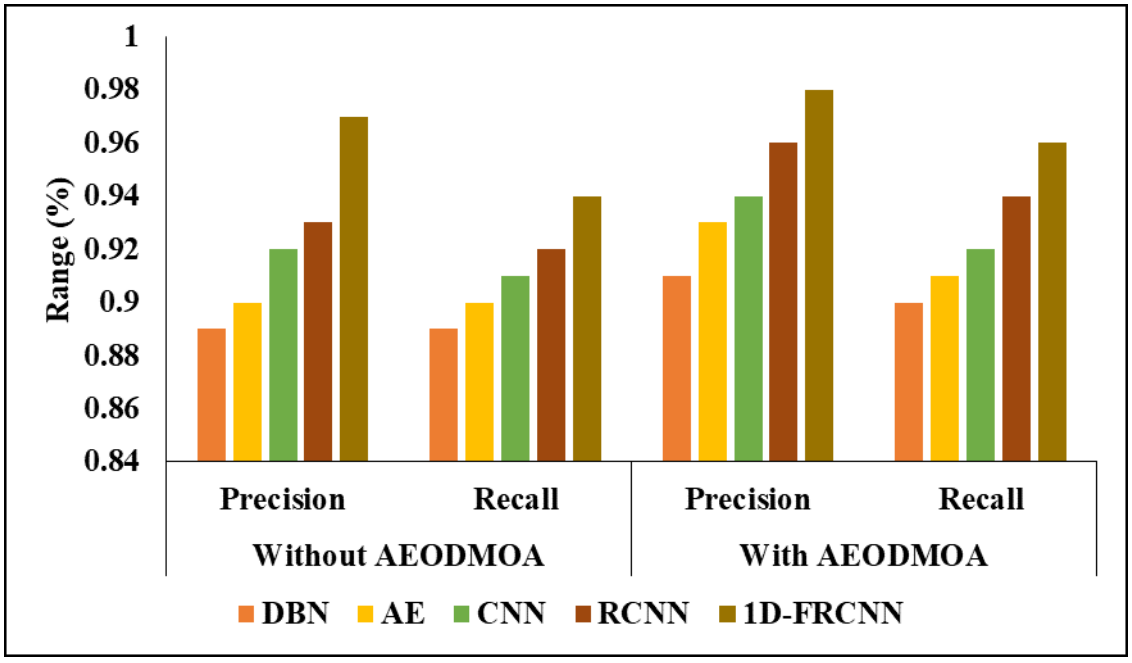

Figure 7: Comparative analysis of Hybrid Optimization with various classifiers

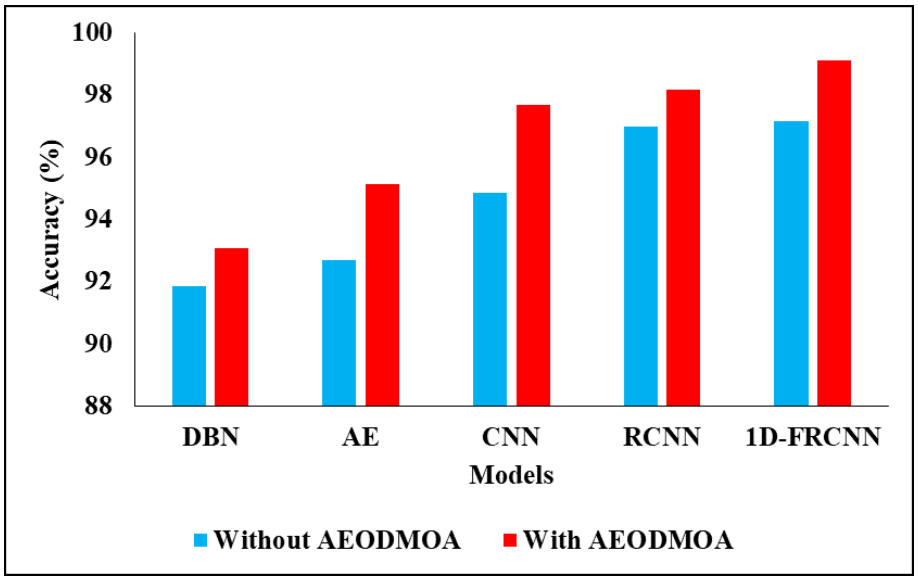

Figure 8: Accuracy Analysis of Various DL classifiers

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

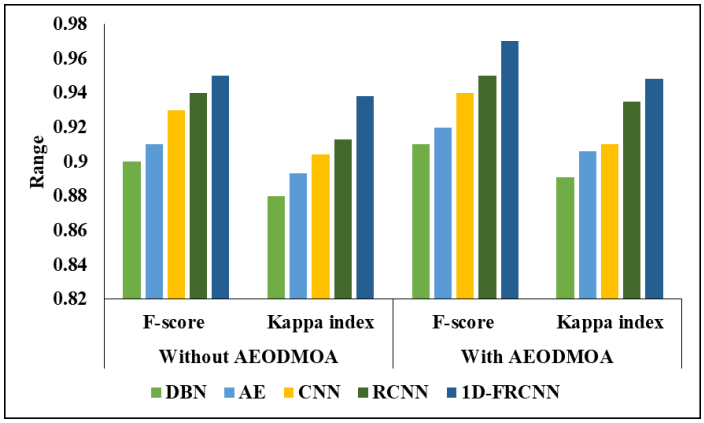

Figure 9: Graphical Representation of proposed DL classifier with Hybrid optimization

4. CONCLUSION AND FUTURE WORK

Criminals that commit fraud frequently employ novel schemes. The ever-evolving character of fraud may be handled with a powerful classifier. Fraud detection accurately forecasts fraud situations while simultaneously minimising false-positive cases. Companies employ strategies that make it difficult, if not impossible, to identify financial statement fraud using conventional methods. The 1D-FRNN model is utilised for classification in this study, with input from the HEGSO feature selection model. By combining AEO and DMOA, AEODMOA properly chooses the classifier's parameters to boost performance during exploration and exploitation. To begin the optimisation process, we split the total sum of iterations in half and used AEO for the first set before switching to DMOA for the second set. Tests showed that the suggested model had an accuracy of 97% without the AEODMOA model and 99% with the hybrid optimisation model. There are a plethora of sampling methods that can improve the efficiency of already-existing instances, but they all fall short when applied to the hidden data. As the gender gap widened, so did the performance of blind data. To counteract this shortcoming, future research may concentrate on a variety of topics, maybe beginning with the suggestion of data preparation techniques. The effect of various feature extraction techniques on prediction accuracy in the credit card industry should also be studied.

REFRENCES

[1] Xiuguo, W. and Shengyong, D., 2022. An analysis of financial statement fraud detection for Chinese listed companies using deep learning. IEEE Access, 10, pp.22516-22532.

[2] Ashtiani, M.N. and Raahemi, B., 2021. Intelligent fraud detection in financial statements using machine learning and data mining: a systematic literature review. IEEE Access, 10, pp.72504-72525.

[3] Bao, Y., Hilary, G. and Ke, B., 2022. Artificial intelligence and fraud detection. Innovative Technology at the Interface of Finance and Operations: Volume I, pp.223-247.

[4] Mehbodniya, A., Alam, I., Pande, S., Netware, R., Rane, K.P., Shabaz, M. and Madhavan, M.V., 2021. Financial fraud detection in healthcare using machine learning and deep learning techniques. Security and Communication Networks, 2021, pp.1-8.

[5] Błaszczyński, J., de Almeida Filho, A.T., Matuszyk, A., Szeląg, M. and Słowiński, R., 2021. Auto loan fraud detection using dominance-based rough set approach versus machine learning methods. Expert Systems with Applications, 163, p.113740.

[6] Hilal, W., Gadsden, S.A. and Yawney, J., 2022. Financial fraud: a review of anomaly detection techniques and recent advances. Expert systems With applications, 193, p.116429.

[7] Sanober, S., Alam, I., Pande, S., Arslan, F., Rane, K.P., Singh, B.K., Khamparia, A. and Shabaz, M., 2021. An enhanced secure deep learning algorithm for fraud detection in wireless communication. Wireless Communications and Mobile Computing, 2021, pp.1-14.

[8] Aslam, F., Hunjra, A.I., Ftiti, Z., Louhichi, W. and Shams, T., 2022. Insurance fraud detection: Evidence from artificial intelligence and machine learning. Research in International Business and Finance, 62, p.101744.

[9] Zhou, H., Sun, G., Fu, S., Wang, L., Hu, J. and Gao, Y., 2021. Internet financial fraud detection based on a distributed big data approach with node2vec. IEEE Access, 9, pp.43378-43386.

[10] Gupta, A., Lohani, M.C. and Manchanda, M., 2021. Financial fraud detection using naive Bayes algorithm in highly imbalanced data set. Journal of Discrete Mathematical Sciences and Cryptography, 24(5), pp.1559-1572.
[11] Sánchez-Aguayo, M., Urquiza-Aguiar, L. and Estrada-Jiménez, J., 2021. Fraud detection using the fraud triangle theory and data mining techniques: A literature review. Computers, 10(10), p.121.
[12] Izotova, A. and Valiullin, A., 2021. Comparison of Poisson process and machine learning algorithms approach for credit card fraud detection. Procedia Computer Science, 186, pp.721-726.
[13] Ashfaq, T., Khalid, R., Yahaya, A.S., Aslam, S., Azar, A.T., Alsafari, S. and Hameed, I.A., 2022. A machine learning and blockchain-based efficient fraud detection mechanism. Sensors, 22(19), p.7162.
[14] Zhang, X., Han, Y., Xu, W. and Wang, Q., 2021. HOBA: A novel feature engineering methodology for credit card fraud detection with a deep learning architecture. Information Sciences, 557, pp.302-316.
[15] Eltweri, A., Faccia, A. and KHASSAWNEH, O., 2021, December. Applications of Big Data within Finance: Fraud Detection and Risk Management within the Real Estate Industry. In 2021 3rd International Conference on E-Business and E-commerce Engineering (pp. 67-73).
[16] Stojanović, B., Božić, J., Hofer-Schmitz, K., Nahrgang, K., Weber, A., Badii, A., Sundaram, M., Jordan, E. and Runevic, J., 2021. Follow the trail: Machine learning for fraud detection in Fintech applications. Sensors, 21(5), p.1594.
[17] Khan, A.T., Cao, X., Li, S., Katsikis, V.N., Brajevic, I. and Stanimirovic, P.S., 2022. Fraud detection in publicly traded US firms using Beetle Antennae Search: A machine learning approach. Expert Systems with Applications, 191, p.116148.
[18] Fanai, H., & Abbasimehr, H. (2023). A novel combined approach based on deep Autoencoder and deep classifiers for credit card fraud detection. Expert Systems with Applications, 217, 119562.
[19] Ganji, V. R., Chaparala, A., & Sajja, R. (2023). Shuffled shepherd political optimization-based deep learning method for credit card fraud detection. Concurrency and Computation: Practice and Experience, 35(10), e7666.
[20] Fakiha, B. (2023). Forensic Credit Card Fraud Detection Using Deep Neural Network. Journal of Southwest Jiaotong University, 58(1).
[21] Wei, Y. C., Lai, Y. X., & Wu, M. E. (2023). An evaluation of deep learning models for chargeback Fraud detection in online games. Cluster Computing, 26(2), 927-943.
[22] Bakhtiari, S., Nasiri, Z., & Vahidi, J. (2023). Credit card fraud detection using ensemble data mining methods. Multimedia Tools and Applications, 1-19.
[23] Karthika, J., & Senthilselvi, A. (2023). Smart credit card fraud detection system based on a dilated convolutional neural network with sampling technique. Multimedia Tools and Applications, 1-18.
[24] AR, S., 2023. Flower Pollination Optimization Algorithm with Stacked Temporal Convolution Network-based Classification for financial anomaly Fraud Detection.
[25] Qi, Z.; Zhang, Z. A hybrid cost-sensitive ensemble for heart disease prediction. BMC Med. Inform. Decis. Mak. 2020, 21, 73.
[26] [26]. Chang, C.-C.; Lin, C.-J. LIBSVM: A library for support vector machines. ACM Trans. Intell. Syst. Technol. 2011, 2, 27.
[27] Upadhyay, P., Marriboina, V., Kumar, S., Kumar, S. and Shah, M.A., 2022. An enhanced hybrid glowworm swarm optimization algorithm for traffic-aware vehicular networks. IEEE Access, 10, pp.110136-110148.
[28] Ren S, He K, Girshick R, et al. Faster R-CNN: towards real-time object detection with region proposal networks. In: Cortes C, Lawrence ND, Lee DD, et al. (eds) Advances in neural information processing systems, vol.28. Red Hook, NY: Curran Associates, Inc., 2015, pp. 91–95.
[29] Zhao, W., Wang, L., Zhang, Z.: Artificial ecosystem-based optimization: a novel nature-inspired meta-heuristic algorithm. Neural Comput. Appl. 32(13), 9383–9425 (2020)
[30] Agushaka, J.O., Ezugwu, A.E., Abualigah, L.: Dwarf mongoose optimization algorithm. Comput. Methods Appl. Mech. Eng. 391, 114570 (2022).
[31] Al-Shourbaji, I., Helian, N., Sun, Y., Alshathri, S., Abd Elaziz, M.: Boosting ant colony optimization with reptile search algorithm for churn prediction. Mathematics 10(7), 1031 (2022).