Journal of Positive School Psychology http://journalppw.com
2022, Vol. 6, No. 7, 5328-5345

A Hybrid Deep Learning And Modified Butterfly Optimization Based Feature Selection For Transaction Credit Card Fraud Detection

N.Geetha[1] , Dr. G.Dheepa[2]

[1]Research Scholar, Department of Computer Science, P.K.R College of Arts & Science For Women, TamilNadu
[2]Assistant Professor, Department of Computer Science, P.K.R College of Arts & Science For Women, TamilNadu

Abstract: Credit cards are playing an extremely significant part in the modern economy. However, as the number of people who use credit cards continues to climb, the number of fraudulent credit cards transactions has also increased. There are numerous different ideas put up in order to combat the rising incidence of credit card theft. In the existing research work, developed an ENNs (Enhanced Neural Networks) for enhanced accuracy of results using feature selection techniques based on ABCs (Artificial Bee Colonies) which select relevant features from transaction level credit card datasets. Therefore, the quality of the classification will vary based on the input data dimension only and the expense of making accurate decisions grows to be a serious issue. So this research work, introduced a hybrid deep learning and adaptive feature selection methodology for efficiently detecting credit card frauds. First, the characteristics of the transactional documents need to be completely ordered, and then the contents of each feature need to be categorized. Construct a Logical Graph of Behavior Profile (LGBP) using them as a foundation. This graph should abstract and include all distinct transaction data. Then the Modified Butterfly Optimization Algorithm (MBOA) based feature selection has been utilized in order important to identify characteristics from transaction level credit card dataset. And finally the hybrid deep learning model is used in order to depict the rational connection between the characteristics of transactional details. Here the Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) is hybridized for improving the detection performance of the transactions. According to the findings of the simulations, the suggested hybrid deep learning model has a higher reliability and recognition rate than the other models that are currently available.

Keywords: Credit card fraud, Logical Graph of Behavior Profile (LGBP), Modified Butterfly Optimization Algorithm (MBOA), Fraud Detection, Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN).

1. Introduction

Credit cards, Automated Teller Machines (ATMs), online banking and mobile banking services are some examples of the new services that financial institutions are using in the present day and age to broaden the range of economic options that are available to customers [1]. Additionally, along with the fast advancements of e-commerce, the usage of credit cards has developed into both a convenient and essential component of modern days financial life. Credit cards are a kind of payment card that may be issued to consumers as a method of payments. Using a credit card may provide one with several benefits, including the following:

- Buy-ability Credit cards are helpful. They let clients buy on credit at any time, place and quantity [2]. Offer a simple payment solution for online, phone

and ATM transactions. Keep credit records.

- Good credit helps identify loyal clients. This background is useful for credit cards, loans, rentals and certain employment. Creditors and providers of credit mortgage firms, credit card businesses, retail outlets and utility organizations may analyse consumer credit score and history [3]. Buy protections.
- Customers may also be offered extra protection by credit cards in the event that the bought goods are missing, damaged or stolen. If the actual receipts are missing or stolen, the purchase may still be verified by the merchant's credit card statement and the consumer. Additionally, some credit card providers provide protection for significant transactions.

Credit card scam involves unlawful usage of a credit card or its contents. Registrant and behavioural fraud are two types of fraudulent activity. Applicant information is stolen when thieves apply for new cards using fake or hacked data [4]. One person may submit many applications using the same information, while another user may submit various applications using different data. On the other hand, behavioural fraud includes stolen/lost card, mail theft, fake card and "card holder not present" fraud. When fraudsters use a stolen or lost card, they conduct stolen/lost card fraud. When a credit card or private data is obtained via mail theft fraud, the scammer never really contacts the cardholder. Without the cardholders' awareness, credit card information is collected in both counterfeit and "card holder not present" scams [5]. In the former, card information may be used for remote transactions that are carried out through the phone, Internet or the mail. In the latter, fake cards are created using card data.

Scam analysis and customer behavioural assessment are two credit card fraud detection strategies. First, transaction-level supervised categorization [6]. Based on past data, these approaches identify transactions as fraudulent or regular. Using this database, classification algorithms may predict the condition of fresh records [7]. Rule induction, decision trees and neural networks are used to build two-class models. This technology called abuse identification has correctly detected most fraud schemes in the past [8]. The second option uses unstructured techniques based on account behavior. If a transaction is compared to a user's usual behavior, its authenticity may be questioned. This is because don't expect unauthorized users to behave like the bank user, nor do we expect them to know the bank customer's behavior pattern. In order to achieve this aim, first discover the authentic user behavioural pattern for each account and then detect deceptive behaviours [9]. By comparing new behaviours to this model, fraudsters are found. Account activity data, such as merchants, amounts, locations and times may be included in profiles. A different name for this approach is "anomaly detection".

The major distinctions among fraudulent detection methodologies and user's behavioural assessment should be made clear. The fraud analysis method has a low false positive rate and may spot common scams. These algorithms can quickly detect frauds by identifying the signature and model of fraud methods from the Oracle database. If test data has no fraud symptoms, no alarm is triggered [10]. Mistakes may be reduced. Since a fraud analysis system (classifier) learns from a limited number of fraud records, it can't discover new frauds. As a consequence, based on how cunning the scammers are, the false negatives rate might be quite high [11]. An ANN is a collection of linked nodes created to mimic how the human brain works. Each node is connected to a number of additional nodes in subsequent levels through a weighted connection. Each node applies the weights to

the input it receives from other nodes linked to it. However, since the classifier's effectiveness is solely dependent on the data dimensions, the selection cost for precision represents a serious issue. In order to effectively identify credit card fraud, this research effort proposed a hybrid deep learning and intelligence feature selection approach.

Section 2 discusses current credit card fraud detection technologies. Section 3 suggests a strategy. Section 4 presents results and comments. Section 5 discusses results and next steps.

2. Literature Review

In this part, an overview of some of the more current strategies detecting credit card theft with machine learning and feature selection techniques is presented.

Panigrahi et al [12] suggested an innovative method that incorporates information from both recent and historical activity in order to identify credit card abuse. The rule-based filter, the Dempster-Shafer adder, the transaction history database and the Bayesian learner are the four parts that make up the Fraud Detection System (FDS). The effectiveness of a method for detecting credit card frauds is greatly enhanced by the fusion of several pieces of evidence as matched to other techniques, according to thorough simulations using stochastic models.

Manlangit et al [13] introduced a rule-based system, a machine learning algorithm that is clever and flexible should be the solution to stop such sophisticated data theft. Principal Component Analysis (PCA) is used in the provided framework to modify raw data and perform classification using K-NN. A distance-based feature selection approach was used to construct neighbors or data anomalies, using the Synthetic Minority Oversampling Technique (SMOTE). When applied to the incorrectly classed cases, the suggested method worked well, with accuracy and F-Score values of 100% and 98.24% for the time subset and K-NN, respectively. This investigation also shows a wider and more distinct categorization breakdown, which contributes to a greater accuracy rate and an enhanced recall rate. Distance-based feature selection, Principal Component Analysis (PCA) and Synthetic Minority Oversampling Technique were utilized to obtain such high accuracy.

Kavitha et al [14] designed a decision tree strategy using an evolutionary algorithm to enhance node discovery. The recommended approach is evaluated using a PCA-based ANN classifier. The proposed approach is superior. Mqadi et al [15] created data-point machine learning. The study employed an uneven credit card database and the data-point methodology overfitting using SMOTE. Support Vector Machines, Logistic Regression, Decision Tree and Random Forest classifiers were used for classifications. Accuracy was tested using precision, recall, F1-score and the average precision metric. According to the findings, the model has difficulty identifying fraudulent transactions when the data are severely unbalanced. After implementing the SMOTE-based oversampling strategy, there was a discernible rise in the level of accuracy of the capacity to forecast positive classifications.

Murli et al [16] created a system to identify credit card fraud using neural networks with the help of Neuroph IDE. The many variables that were taken into account during training and testing the neural network are presented in this study. Sahu et al [17] employing five classifiers to determine which classifier is most appropriate for the circumstance, built models were developed to identify fraudulent credit card transactions and take two distinct approaches to the underlying issue of information asymmetry. The second method employs a cost-based approach and includes weights from each class into the error function, while the first strategy uses data interpolation to raise the amount of examples in the minority class. The samples of fraudulent

transactions might be given greater weight via the weights than the regular samples.

Asha et al. [18] created a number of machine learning techniques, including the Support Vector Machine (SVM), K-Nearest Neighbor (KNN) and Artificial Neural Network (ANN). They employ supervised machine learning and deep learning techniques to differentiate between fraudulent and lawful transactions. Ghobadi et al. [19] constructed a Credit Card Fraud Detection (CCFD) model using ANN and Meta Cost. ANN can prevent and identify credit card fraud. Inconsistent data makes it difficult to spot fraudulent transactions. Meta Cost addresses imbalanced data. Cost Sensitive Neural Network (CSNN) is a model for detecting abuse. This model demonstrated cost savings and improved detection rate when comparing to the Artificial Immune System (AIS) model. The study's data came from actual transactions information that was given by a major Brazilian credit card company.

Geetha et al. [20] proposed a new feature selection technique for enhancing classifications of credit card fraud. This research work identifies fraudulent accounts using ENNs (Enhanced Neural Networks) for enhanced accuracy of results using feature selection techniques based on ABCs (Artificial Bee Colonies) which select relevant features from transaction level credit card datasets. Various elements of utilized reduced dataset have been investigated in this study resulting in descriptions of logical relationships between transaction record attributes by ENNs which computes CCFs between attributes based on LGBPs (Logical Graph of Behavior Profiles) and user’s transaction data.

Soltani et al [21] introduced AI-based credit card fraud detection. This AIRS-based model considers user activity. Tracking account activity and generic thresholding, the two fraud detection approaches are combined in this model. While fraud memory cells are formed utilizing overall illegal data, normal memory cells are created by the system utilizing each user's transaction records. They analysed training data to regulate memory cells for high accuracy. Throughout the test phase, each user's transaction is displayed to normal and fraud memory cells. Using a neural network model, Georgieva et al [22] identified scam cases. When properly taught, an ANN may mimic human brain activity. Like humans, they're good classifiers and learn via observation. Credit card traffic has a large gap among legal and illegal transactions. They use resampling approaches to address the unbalanced dataset. A pattern recognition network was created and trained in Matlab's Neural Network Toolbox using a scaling conjugated gradients backpropagation technique.

3. Proposed Methodology

For effective credit card scam identification, this research effort proposed a hybrid deep learning and intelligent feature selection approach. Prior to categorizing the numbers of each characteristic, completely sort the attributes of transaction records. Create a Logical Graph of BP (LGBP) based on them that abstracts and encompasses all various transaction data. Then, using the transactional credit card dataset, relevant features have been chosen using the Modified Butterfly Optimization Algorithm (MBOA) based feature selection. And lastly, the logical relationship between the properties of transaction data is represented using a hybrid deep learning model. Convolutional and recurrent neural networks are used in this instance to enhance the efficacy of transaction identification. Figure 1 depicts the methodology's planned approach.

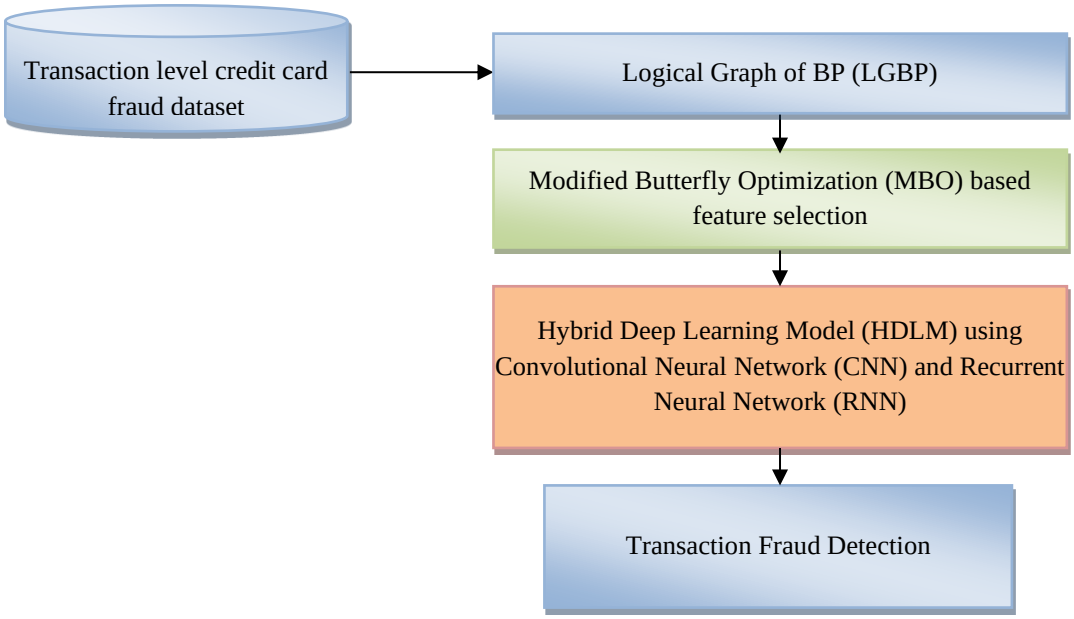

Figure .1. Proposed Transaction Fraud Detection Technique

3.1. Behavior Profile

The principles of transactional records and session log that are employed in the creation of BP are initially introduced in this part.

Definition 1 (Transaction Record):

A transaction record r consists of m attribute values, i.e., $r = \{a_1, a_2, \ldots\ldots, a_m | a_1 \in A_1, a_2 \in A_2, \ldots\ldots, a_m \in A_m\}$ where $A_i = \{a_1^i, a_2^i, \ldots\ldots a_n^i\}$ is the set of values of the i th attribute and $n_i = |A_i|$.

Given a user u, her/his transaction log is a compilation of all of her/his data for a certain duration of time and is identified as $L_u = \{r_1^u, r_2^u, \ldots\ldots r_{n^u}^u\}$ in which $n^u = |L_u|$.

Certain information needs to be pre-processed in the original records. All of these identical data are maintained in L_u to describe the user's activity. Signify Ru as the collection of all unique entries in L_u in order to easily express various equations. In actuality, L_u is a multiset, whereas R_u is a set.

Each transaction document includes the required characteristics, which are shown in Table 1. They are listed in the following order: Merchant_id, Average Amount/transaction/day, Daily_chargeback_avg_amt, 6_month_avg _chbk_amt, 6-month_chbk_freq, Transaction_amount, Total Number of declines/day.

Construct a Logic Graph of BP (LGBP) for a user depending on the user's Merchant _id and the user's transactions log. This logic graph should encompass all transaction records and describe the dependency relations of all attributes of this user's records. To begin, all of the characteristic variables that are present in the user's transactional data should be abstracted u as follows:

$$A_1^u = \{a \in A_1 | \exists r \in R_u : a \in r\} \quad (1)$$

$$A_2^u = \{a \in A_2 | \exists r \in R_u : a \in r\} \quad (2)$$

... ...

$$A_m^u = \{a \in A_m | \exists r \in R_u : a \in r\} \quad (3)$$

Obviously, $A_1^u \subseteq A_1, A_2^u \subseteq A_2, \ldots\ldots$ and $A_m^u \subseteq A_m$. Without limiting the scope of the statement, indicate $A_i^u = \{a_1^i, a_2^i, \ldots\ldots a_{n_i^u}^i\}$ in which $n_i^u = |A_i^u|$ for each $i \epsilon \{1, 2, \ldots, m\}$.

Definition 2 (LGBP):

Let $L_u = \{r_1^u, r_2^u, \ldots\ldots r_{n^u}^u\}$ user u's transaction log is being considered here. A directed non - cyclic graph describes the LGBP of u, $G_u = (V_u, E_u)$, where:

1) $V_u = \{a_s, a_e\} \cup A_1^u \cup A_2^u \cup \ldots\ldots \cup A_m^u$ in which a_s and a_e are the two unique vertices that signify a transaction's beginning and finish;
2) $\forall a \in A_1^u, (v_s, a) \in E_u$;
3) $\forall a \in A_1^u, (a, v_e) \in E_u$;
4) $\forall i \in \{1,2, \ldots, m-1\}, \forall a \in A_i^u, \forall a' \in A_{i+1}^u : (a, a') \in E_u$ if and only if $\exists r \in R_u : a \in r \Lambda a' \in r$.

Table 1. Example of Transaction Log

Transaction Records	Transaction Attributes						
	Merchant_id	Average Amount/transaction/day	Daily_chargeback_avg_amt	6_month_avg_chbk_amt	6-month_chbk_freq	Transaction__Amount	Total Number of declines/day
r_1^u	M_{id}^1	SM	LO	YE	YE	(0-200)	(0-5)
r_2^u	M_{id}^2	AV	HI	NO	NO	(0-200)	(6-10)
r_3^u	M_{id}^3	SM	LO	YE	NO	(1000-200)	(11-15)
r_4^u	M_{id}^4	AV	HI	NO	YE	(1000-2000)	(16-20)

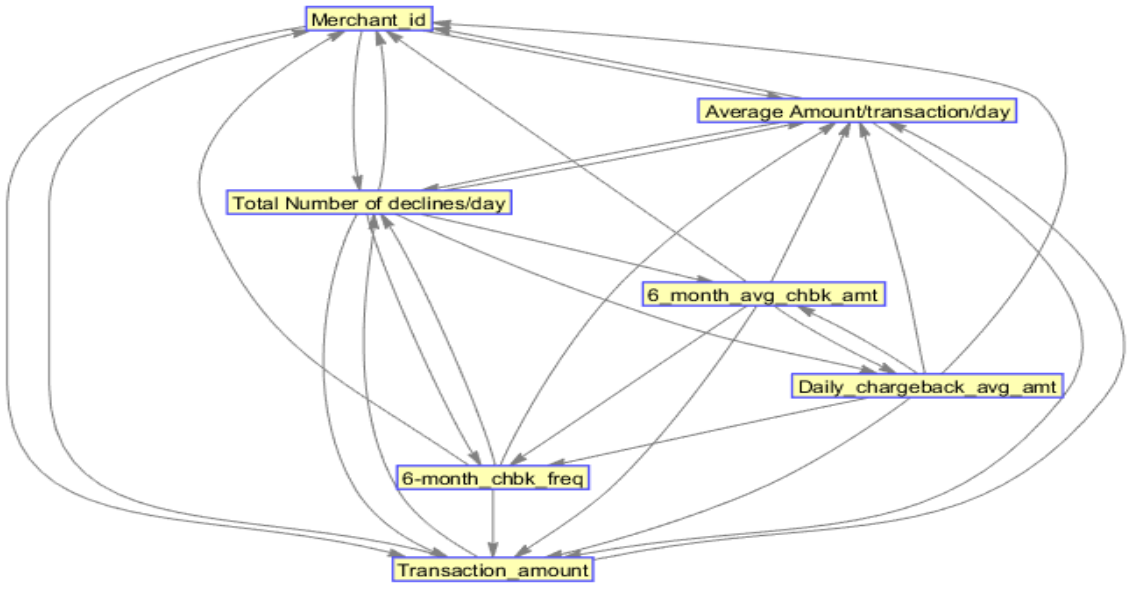

Figure. 2. LGBP of an users whose transactions history is shown in Table 1.

Definition 3 (Prepaths):

Let Gu= (Vu, Eu) be the LGBP of user u. ∀v ∈Vu, prepaths(v) is the collection of all routes that are routed from node a_s to node v in Gu.

Definition 4 (Post nodes):

Let Gu= (Vu, Eu) be the LGBP of u. ∀v ∈Vu, postnodes(v) is the collection of nodes that may be visited straight from v in Gu. There are four straight pathways which is illustrated in Figure. 2 and it is derived as,

$$\sigma_1 = a_s.M_{id}^1.SM.LO.YE.YE.(0,200).(0,5) \quad (4)$$

$$\sigma_2 = a_s.M_{id}^2.AV.HI.NO.NO.(0,200).(6,10) \quad (5)$$

$$\sigma_3 = a_s.M_{id}^3.SM.LO.YE.NO.(1000,200).(11,15) \quad (6)$$

$$\sigma_4 = a_s.M_{id}^4.AV.HI.NO.YE.(1000,2000).(16,20) \quad (7)$$

From the above equation the prepath and the post nodes are defined and its state transition, diversity co-efficient are calculated using the HMM model.

Definition 5 (Behavior Profile):

Let $L_u = \{r_1^u, r_2^u, \dots \dots r_{n^u}^u\}$ represent the users transactions log u. BPu= (Vu, Eu, Mu, ω_u) is the BP of u, where:

1) Gu= (Vu, Eu) is the LGBP of u;

2) Mu = {Mv|v∈Vu} is the collection of probability-based transitional paths [23] of all nodes in Gu;

3) ω_u is the diversity coefficient of u.

Based on each user's transaction history, a BP may be created for them. Proposed technique for determining acceptable transaction records is suitable to a BP in detailed in the next section.

3.2. Feature Selection using Modified Butterfly Optimization Algorithm (MBOA)

By carefully choosing a small group of elements, the characteristic selecting procedure minimizes the impact of noisy and unimportant factors on prediction outcomes. To create the subset of effective characteristics, filtration, wrapping and embedding approaches may be used to the whole dataset [24]. The system performs better when the correct feature set is chosen. The Modified Butterfly Optimization Algorithm (MBOA) was introduced in this research as a means of improving the component selecting process efficiency. The Butterfly Optimization Algorithm (BOA) and its problems are identified initially, the reason for introducing the MBOA is explained briefly in the below subsection.

3.2.1. Butterfly Optimization Algorithm (BOA)

To demonstrate the ideas mentioned in perspective of a search algorithm, the characteristics of butterflies are idealized as follows: 1. Every butterfly is expected to provide some kind of scent that attracts other butterflies. 2. Every butterfly will either migrate at random or in the direction of the butterfly with the strongest smell. 3. The geography of the goal function influences or determines a butterfly's sensory intensity. The BOA process is divided into three phases: the initiation stage, the iterative step and the final stage.

Each time BOA is run, the activation phase comes first, followed by iterative searching and in the final phase, the method is stopped when the optimal solution has been identified. Start-up algorithm defines target function and solution space [25] are BOA

parameter numbers. After defining variables, the algorithm constructs a beginning population of butterflies. Since the number of butterflies doesn't change throughout the BOA experiment, a fixed-size memory is used to store their data. Fitness and smell values are calculated and recorded for randomly placed butterflies. Now that initialization is complete, the algorithm continues on to iteration, where it searches utilizing the fake butterflies.

The method goes through many iterations in the second phase, this is referred to as the iteration process. In each iteration, all butterflies in the solution space move to new locations and their optimum levels are then computed. The procedure begins by determining the fitness values for each butterfly at various locations in the solution space. Then these butterflies will produce aroma where they are situated using Eq. (8). The method contains two essential phases: the phases of both local and global searches. The best butterfly/solution is getting closer to the butterfly g* which may be described by the equation, during the global search phase using Eq. (9).

$$f = cI^a \quad (8)$$

Where c-modality coefficient, I-stimulus intensity.

$$x_i^{t+1} = x_i^t + (r^2 \times g^* - x_i^t) \times f_i \quad (9)$$

where x_i^t is the ith butterfly's iteration-t iterative for solution vector xi. Here, g^* stands for the current top solution discovered in the most recent iteration of the problem. The ith butterfly's fragrance is represented by f, while r is a random integer in the range [0, 1]. Local search phase is characterized by

$$x_i^{t+1} = x_i^t + (r^2 \times x_j^t - x_k^t) \times f_i \quad (10)$$

where x_j^t and x_k^t are jth and kth butterflies from the solution space. If $x_j^t$and x_k^t belongs to the same swarm and Eq. (10) becomes a local random walk if r is a random number in [0, 1]. Locally and globally, butterflies search for food and a partner. Considering physical proximity and other factors like rain, wind, etc., a butterfly's mating or food-finding activities may account for a large fraction of its activity. BOA utilizes switch probability p to switch from global to local search.

The issues can be successfully solved using the traditional BOA method. However, it has significant drawbacks, including early converging, a propensity to enter local optima and poor efficiency. The collaboration strategy is integrated with the BOA algorithm and referred to as the MBOA algorithm in order to address BOA's drawbacks.

- **Collaboration Strategy**

The collaborative technique is used to update each new solution in turn. In this proposed method, the collaboration strategy is used which is discussed below.

The so-called cooperation process is a symbiotic interaction between two different species that results in personal gains from the synergy. Let X_i and X_j stands for the i^{th} and j^{th} creatures in the ecosystem with X_j being a randomly selected organism. The following two Eqs. (11) and (12) describe how X_i and X_j interact to produce new candidate solutions:

$$X_{inew} = X_i + rand[0,1] \times (X_{best} - Mutual_{vector} \times BF1) \quad (11)$$

$$X_{jnew} = X_j + rand[0,1] \times (X_{best} - Mutual_{vector} \times BF2) \quad (12)$$

where $Mutual_{vector} = (X_i + X_j)/2$

where rand [0, 1] is a random integer with a uniform distribution that falls between [0,1]. $X_{best}$is the top living thing in the environment. The benefit factors, BF1 and BF2 are produced at random as either 1 or 2. These elements represent the degree of each organism's advantage and Common Vectors

indicates the nature of the connection among two species X_i and X_j. Subsequently, X_{inew} and X_{jnew} are compared with X_i and X_j to choose the most suited organism from each pair. New organisms are created at this phase using the best organism, X_{best}. This helps to improve the capacity of exploits or local search. The algorithm 1. displays the suggested MBOA's workflow.

Algorithm 1. Pseudocode of MBOA

Input: Objective function f (X), $X_i = (X_1, X_2, X_3, \dots \dots, X_{dim})$, dim= no. of dimensions
Maxiter - maximum number of iteration
Bf: The number of butterfly in the ecosystem

Initialization:
Set the initial generation/iteration number G=0;
Generate initial population of n butterflies X_i = (i=1,2,..,n)
Stimulus Intensity I_i at X_i is determined by f (X_i)
Define sensor modality c, power exponent a and switch probability p
While stopping criteria do not met do
For each butterfly Bf in the population do
Calculate fragrance $f = cI^a$ for Bf

End for

Find the best Bf
For each butterfly Xi in the population
do
Generate a random number r from [0,1]
If r<p then
Move towards best butterfly solution using Eq. (9)
Randomly select one butterfly (i≠j);
Determine mutual relationship vector *($Mutual_{Vector}$)* by Eq. (10)
Update butterfly based on their mutual relationship according to the collaboration strategy using Eqs. (11) and (12);
Calculate fitness value of the new Butterfly;
Else
Update the new position
End If
End for
Update the best value
End while

Output: The best Butterfly with the minimum fitness function value in the ecosystem;

3.3. Hybrid Deep Learning Model (HDLM)

Multiple processing layer computer models may learn representations of data at several levels of abstraction to deep learning. Then, based on these depictions, guesses are produced. The hybrid learning technique is presented in this work to increase credit card accurateness forecast efficiency. Combining the recurrent neural network with the convolutional neural network for increasing the detection process than the single classifier.

The advancement of biotechnology has led to the creation of convolutional neural networks as models. Neurons are like well-organized local filters that can be applied to the

whole input space. Convolutional Neural Networks (CNN) are capable of extracting both local and deep features from input data [26]. Recurrent neural networks examine sequence data. Standard neural network architecture links input, hidden, output layers and nodes. A network cannot handle sequence data. The CNN layer learns low-level translation-invariant features to produce higher order features [27]. RNNs integrate convolution and pooling into a hierarchical process. Both models outperform a-priori approaches. These efforts encouraged combining CNN and RNN to classify fraud transactions. The proposed model is CNN-RNN. The recommended technique includes training and testing phases. Before training, the CNN model was pre-trained. Then, transmission learning is used to create a new CNN from a previously trained network. All CNN layers are frozen before training RNN. After thawing the CNN model, the CNN-RNN model is trained. Attention procedures combine CNN and RNN properties. During testing, pre-processed test data are put into the tuned CNN-RNN model and the Softmax layer produces classification results.

3.3.1. Proposed Model

The following components make up the proposed model: Layers providing Softmax output for pre-trained convolutional neural networks, RNN layers, Merge layers and fully connected layers.

1) Pre-Trained Convolutional Neural Network Layer

As the starting weights for CNN model, utilize the weight parameters that were acquired from pre-training on the dataset. Convolutional layer and pooling layer are components of convolutional neural networks.

2) Convolutional Layer

The fundamental method for calculating this layer, which is the most crucial component of the convolutional neural network, is to employ convolution windows of various widths to execute convolution working with the feature maps from the previous layer. Convolution windows of progressively larger sizes are applied to the feature map of the layer before. The convolutional layer's weight parameters alter in accordance with the window size, which is typically 33 or 55. The output is produced by convolution the values of the neurons on each feature map in the convolutional layer via the proper windows, in accordance with the activation function employed in the layer.

3) Pool Layer

Comparable to how the convolutional layer works, this layer's calculating procedure is also similar. The distinction is that, typically, the sliding step is 2 and the sliding window of the bottom sample layer is 2 2. Due to the fact that the size of the preceding layer's feature map will typically have been cut in half as a result of this operation, the convolution weights of neural network parameters may be significantly reduced, which is helpful for accelerating the network training process's overall speed. The network may also grow better adapted to the size of the picture changes as a result, which is another benefit of doing this. In this work, the activation function is the ReLU (Linear Rectification Function).

4) RNN Layer

Input, hidden and output layers are all present in both RNN and CNN. The most significant aspect of RNN is the connectivity between these hidden levels. Relationships among input and hidden nodes send the hidden layer to the output layer. Once again, the hidden layer node receives information from the node, which may also comprise hidden layer nodes that are close to one another. The network here is dynamic. RNNs are closer to the biological nervous system since they are cyclic and can understand serial data. This method can gain data on long-term reliance. The way LSTM (Long Short-Term Memory) differs from RNN is by adding a "processor" to judge whether the data is

helpful or not. The cell is the name of this processor's internal structure. Cells have an input gate, a forgotten gate and an output gate. A message enters LSTM and is evaluated. The incompatible data will be erased via the Oblivion Gate, leaving only the data that has been certified by the algorithm. LSTMs consist of a memory cell with input, output and forget gates. The internal structure of the LSTM is more sophisticated than the typical RNN repeat module.

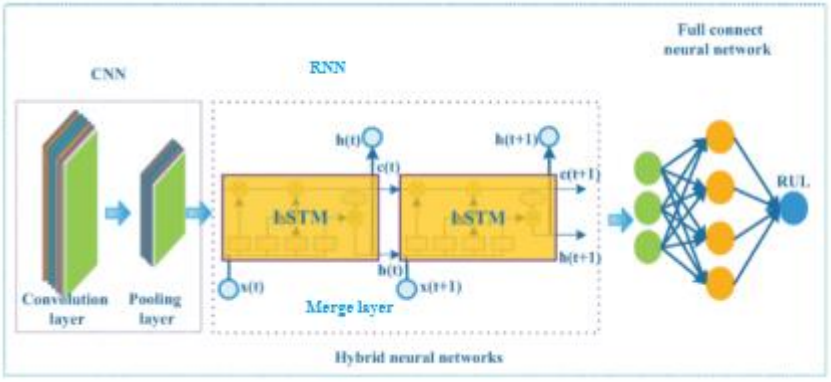

Figure 3. Proposed Hybrid Deep Learning Model

5) Merge Layer

The merge layer combines RNN and CNN features using a feature fusion process. The merge layer function will combine the features received from the RNN with the features obtained from the CNN using a feature fusion. Introduce neural network attentional processes into the sequential model. A neural network with the ability to pick certain input and concentrate on it (or its properties) is said to have a neural network attentiveness mechanism. Additionally, for feature merging, employ the appropriate element-wise multiplying procedures.

6) Fully Connected Layer with Softmax Output

The probability distribution of all classes is the output of the fully coupled Softmax layer, which receives the characteristics produced by the RNN and CNN after they have been combined. Additionally, quantify the discrepancy among the real output and the desired output using the cross-entropy loss function.

7) Network training

The two branches of this model are distinct from one another. The parameters used by the weights in the CNN branch were pre-trained on the dataset, while the parameters used by the RNN branch were initialized at random. Using the cross-entropy loss function gradient, these weights are repeatedly adjusted throughout the training phase. First, the CNN layer is frozen. The RMSProp optimizer then calculates the training samples, which take 100 iterations to complete. The CNN layer is then defrosted, the Adam optimizer is used across the network to compute training samples, the learning rate is set to 0.0001 and 70 epochs are needed for training.

These coupled models do not learn compact representations by using time-dependent features. Able to train deep architectures even with less samples available because to these underlying temporal requirements. Credit card payment information may be used to better predict time sequences and take advantage of temporal dependencies that often show up. In order to train the model quickly, this work added a Weight Activation Factor to reduce error propagating during training.

3.3.2. Weight Activation Factor

The j[th] month's history of credit card payments made by the i[th] customer is shown as $c_i^{(j)}, i = 1,2,\ldots..,K, j = 1,2,\ldots..,N$.The model was trained using data from 6 months worth of transaction record, as shown by the fact that K specifies the total amount of clients and N = 6. Let $z_i^{(j)}$indicate the values for the hidden state activating for the j[th] month for the i[th] client. Labels were converted into one-hot vectors and the value yi is used to indicate the actual label associated with the i[th] client. All activity levels are taken into account by a neural network., $Z_i = [z_i^{(j)}, \ldots., z_i^{(6)}]$, to anticipate credit card defaults label, $\hat{y}_{pre,i}$, for pre-training:

$$\hat{y}_{pre,i} = \sigma(WZ_i + b) \quad (13)$$

where σ (·) is a sigmoid function, W and b represents the neural network's weight activation factor and biases, respectively. The following is thus a possible formulation for the loss function for the pre-training model.:

$$\mathcal{L}_{pre} = -\sum_i logP_\theta(y_i|\{c_i^{(1)}, \ldots\ldots.. c_i^{(6)}\}) \quad (14)$$

where θ represents all of the network′s parameters and $P_\theta\left(y_i\middle|\left\{c_i^{(1)}, \ldots\ldots.. c_i^{(6)}\right\}\right)$easily provided by extracting the element from the output vector $\hat{y}_i$. In order to produce the final prediction, an RNN predictors was subsequently fed a new characteristic set that included demographic (static) characteristics and the activating values of hidden states. Indicate the ith client's static characteristics as x_i , then the following final forecast may be made:

$$\hat{y}_i = RNN\left(x_i, z_i^{(1)}, \ldots\ldots, z_i^{(6)}\right) \quad (15)$$

Algorithm 2. Process of hybrid CNN and RNN

Input: F $(A\ set\ of\ n\ features, F = F_1, F_2, \ldots\ldots F_N)$
Output: Y (fraud labels: 0 or 1)
For each features F_i in F do
    V_i =encodes the input function in hidden representation(F_i)
End For
For each V_i do
    $C_i = CNN(V_i)$
End For
For each C_i do
    $O_i = RNN(V_i)$
End For
For each O_i do
    update weight activation factor using Eqs (13-15).
    $Y_i = ReLU(O_i)$ // merge layer with Linear Rectification Function
End For

models, the suggested model performs best in terms of data predicting for credit card fraud.

4. Results and Discussion

The effectiveness of the technique suggested in this work is shown in this part. Described the parameters and data set first. Then, illustrated the comparison's outcomes. The MATLAB is used for evaluated.

Credit card information is often inaccessible to us in real life. Despite the fact that there are several publicly accessible data sets regarding credit card fraud detection, such as the one at https://www.kaggle.com/shubhamjoshi2130of/abstract-data-set-for-credit-card-fraud-detection#credit cardcsvpresent.csv.

In this work used seven attributes of transaction record's namely

Merchant_id, Average Amount/transaction/day, Daily_chargeback_avg_amt, 6_month_avg _chbk_amt, 6-month_chbk_freq, Transaction amount, Total Number of declines/day.

The selected attributes were found to be effective for identifying transactional fraud. Here, transaction amount/day was divided into two segments small (SM) and Average (AV). For experiment, four merchant id were mentioned and amount frequencies were denoted as Low (LO) and High (HI), amount was divided into four segments: (0-200), (0-200), (1000-200) and (1000- 2000). In actuality, the pre-processed data were utilized in experiment based on the prior modifications. Accuracy, Precision, Recall and F-measure are the external quality measures used in this suggested work.

Following is a formula for calculating accurateness of positives and negatives:

$$\text{Accuracy} = (TP+TN)/(TP+TN+FP+FN) \quad (16)$$

Precision is defined as the ratio of correctly found positive observations to all of the expected positive observations [23].

$$\text{Precision} = TP/(TP+FP) \quad (17)$$

According to [23], recall is calculated when expressed as the ratio of correctly recognized positive information to all data.

$$\text{Recall} = TP/(TP+FN) \quad (18)$$

The weighted average of Precision and Recall is referred to as the F-measure [23]. It thus considers both false positives and false negatives.

$$\text{F-measure} = 2*(\text{Recall} * \text{Precision}) / (\text{Recall} + \text{Precision}) \quad (19)$$

Table 2 tabulates performances of the suggested technique with existing methods and it can be clearly identified from table values that the suggested technique outperforms other methods.

Table 2. Performance results of the proposed and existing methods

Metrics	TAS	LGBPs	LGBP-ENN	HDLM
Accuracy	85.1500	91	93.100	94.300
Precision	82.0190	90.8352	93.9557	95.2272
Recall	84.230	91.0908	91.1612	95.8272
F-measure	83.0060	90.9629	92.537	95.5262

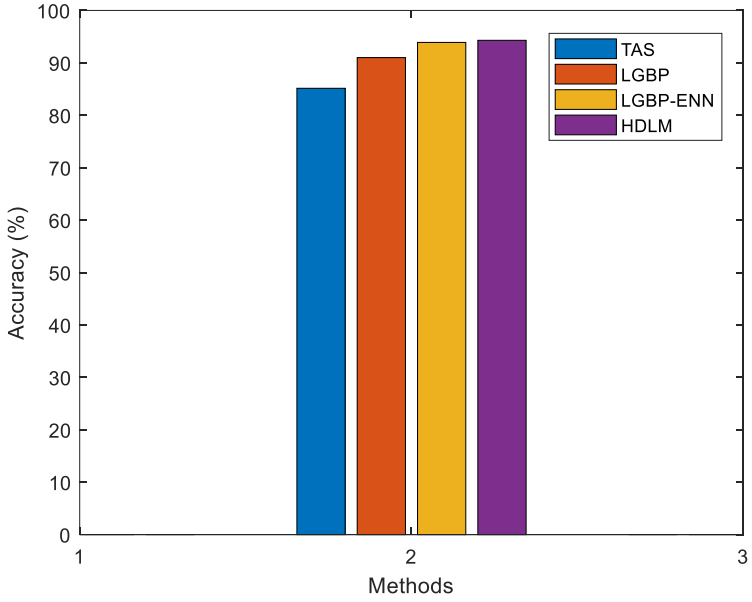

Figure 4. Accuracy comparison between the suggested and existing fraud detection technique

The accuracy comparison between the suggested and existing fraud detection techniques is shown in Figure 4. The proportion of all transactions both legitimate and fraudulent that have been successfully identified is known as accuracy. Compared to the existing LGBP-ENN, LGBP and TAS fraud detection techniques, the simulated findings show that the proposed HDLM delivers greater efficiency.

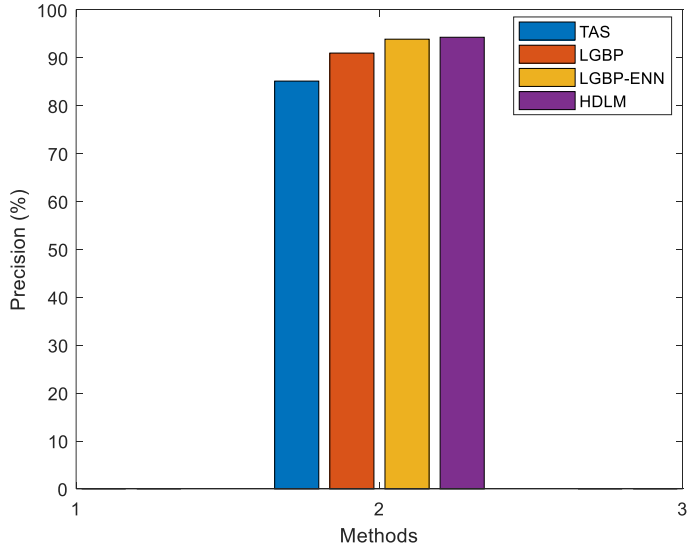

Figure 5. Precision comparison between the suggested and existing fraud detection technique

The Precision comparisons among the suggested and existing fraudulent detecting techniques is shown in Figure 5. It is concluded that when compared to the existing LGBP-ENN, LGBP and TAS, the proposed HDLM offers the best credit card fraudulent identification. It can be seen from the graph that

precision accurately predicts fraudulent transactions.

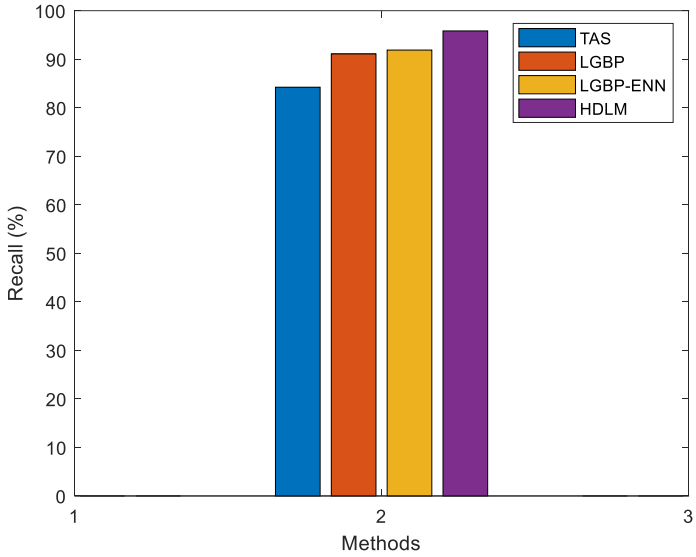

Figure 6. Recall comparison between the suggested and existing fraud detection technique

The Figure 6 illustrates recall comparisons among the suggested and existing fraudulent detecting techniques. According to the simulated findings, the suggested HDLM has a higher recall rate than the existing fraudulent detecting method. In this work, the Modified Butterfly Optimization Algorithm based feature selection technique is proposed for reducing the complexity of the classifier. Finally compared to the existing LGBP-ENN, LGBP and TAS, this proposed HDLM based model considers the both advanced deep learning model CNN and RNN provides the best detection rate.

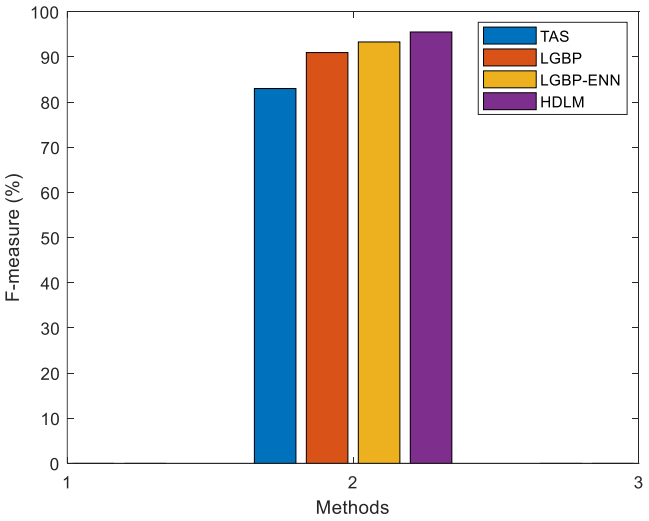

Figure 7. F-measure comparison between the suggested and existing fraud detection technique

The F-measure comparison between the proposed and existing fraud detection techniques is shown in Figure 7. Recall provides the fraud and non-fraud accuracies,

whereas F-measure provides the harmonic mean of precision. It finds that when compared to the existing LGBP-ENN, LGBP and TAS, the proposed HDLM offers the greatest credit card fraud detection.

5. Conclusion

The most frequent issue that causes customers to lose money as well as loss for credit card fraud committed by banks and credit card firms. In order to prevent people from losing their wealth, as well as for banked companies, this research aims to create a model that can more effectively distinguish between transactions that are fraudulent and those that are not fraudulent by utilizing the proposed Hybrid Deep Learning model in the provided data set. Prior to categorizing the values of each characteristic, completely sort the attributes of transaction records. Create a Logical Graph of BP (LGBP) based on them that encompasses all types of transaction data. Then, using the transaction level credit card dataset, relevant features have been chosen using the Modified Butterfly Optimization Algorithm (MBOA) based feature selection. And lastly, the logical relationship between the properties of transaction data is represented making use of a mixed deep learning model. Here, Recurrent Neural Network (RNN) and Convolutional Neural Networks (CNN) are combined to improve the efficiency of transaction detection. Based on their experimental findings and comparison to other approaches, it was discovered that, for the detection of fraud data, compared to CNN models and traditional machine learning methods, the combined structure of CNN and RNN models performed better. To achieve improved categorization reliability comparing to the previous models, the ReLU-based activation function is used in this work. When creating a successful hybrid model, it's crucial to constantly combine a costly method that takes a lot of time to operate but produces effective comes from using an optimizing technique to reduce the expense of the whole process.

REFERENCES

1. Raj, S. B. E., & Portia, A. A. (2011, March). Analysis on credit card fraud detection methods. In 2011 International Conference on Computer, Communication and Electrical Technology (ICCCET) (pp. 152-156). IEEE.
2. Shen, A., Tong, R., & Deng, Y. (2007, June). Application of classification models on credit card fraud detection. In 2007 International conference on service systems and service management (pp. 1-4). IEEE.
3. Dal Pozzolo, A., Caelen, O., Le Borgne, Y. A., Waterschoot, S., &Bontempi, G. (2014). Learned lessons in credit card fraud detection from a practitioner perspective. Expert systems with applications, 41(10), 4915-4928.
4. Quah, J. T., &Sriganesh, M. (2008). Real-time credit card fraud detection using computational intelligence. Expert systems with applications, 35(4), 1721-1732.
5. Makolo, A., &Adeboye, T. (2021). Credit Card Fraud Detection System Using Machine Learning.
6. Chaudhary, K., Yadav, J., &Mallick, B. (2012). A review of fraud detection techniques: Credit card. International Journal of Computer Applications, 45(1), 39-44.
7. Awoyemi, J. O., Adetunmbi, A. O., &Oluwadare, S. A. (2017, October). Credit card fraud detection using machine learning techniques: A comparative analysis. In 2017 international conference on computing networking and informatics (ICCNI) (pp. 1-9). IEEE.
8. Bahnsen, A. C., Stojanovic, A., Aouada, D., &Ottersten, B. (2013, December). Cost sensitive credit card fraud detection using Bayes minimum risk. In 2013 12th international

conference on machine learning and applications (Vol. 1, pp. 333-338). IEEE.
9. Halvaiee, N. S., & Akbari, M. K. (2014). A novel model for credit card fraud detection using Artificial Immune Systems. Applied soft computing, 24, 40-49.
10. Tripathi, K. K., &Pavaskar, M. A. (2012). Survey on credit card fraud detection methods. International Journal of Emerging Technology and Advanced Engineering, 2(11), 721-726.
11. Adewumi, A. O., &Akinyelu, A. A. (2017). A survey of machine-learning and nature-inspired based credit card fraud detection techniques. International Journal of System Assurance Engineering and Management, 8(2), 937-953.
12. Panigrahi, S., Kundu, A., Sural, S., &Majumdar, A. K. (2009). Credit card fraud detection: A fusion approach using Dempster–Shafer theory and Bayesian learning. Information Fusion, 10(4), 354-363.
13. Manlangit, S., Azam, S., &Shanmugam, B. (2019). Novel machine learning approach for analyzing anonymous credit card fraud patterns. International Journal of Electronic Commerce Studies, 10(2), 175-202.
14. Kavitha, C., &Iyakutti, K. (2014). Optimized Anomaly based Risk Reduction using PCA based Genetic Classifier. Global Journal of Computer Science and Technology.
15. Mqadi, N., Naicker, N., &Adeliyi, T. (2021). A SMOTE based oversampling data-point approach to solving the credit card data imbalance problem in financial fraud detection. International Journal of Computing and Digital Systems, 10(1), 277-286.
16. Murli, D., Jami, S., Jog, D., &Nath, S. (2015). Credit card fraud detection using neural networks. International Journal of Students' Research in Technology & Management, 2(2), 84-88.
17. Sahu, A., Harshvardhan, G. M., &Gourisaria, M. K. (2020, December). A dual approach for credit card fraud detection using neural network and data mining techniques. In 2020 IEEE 17th India council international conference (INDICON) (pp. 1-7). IEEE.
18. Asha, R. B., & KR, S. K. (2021). Credit card fraud detection using artificial neural network. Global Transitions Proceedings, 2(1), 35-41.
19. Ghobadi, F., &Rohani, M. (2016, December). Cost sensitive modeling of credit card fraud using neural network strategy. In 2016 2nd international conference of signal processing and intelligent systems (ICSPIS) (pp. 1-5). IEEE.
20. Geetha,N., & Dheepa,G. (2022, March).Transaction fraud detection using Artificial Bee Colony (ABC) based feature selection and Enhanced Neural Network (ENN) classifier. International Journal of Mechanical Engineering (Vol. 7, No. 3, ISSN 0974-5823).
21. Soltani, N., Akbari, M. K., &Javan, M. S. (2012, May). A new user-based model for credit card fraud detection based on artificial immune system. In The 16th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP 2012) (pp. 029-033). IEEE.
22. Georgieva, S., Markova, M., & Pavlov, V. (2019, October). Using neural network for credit card fraud detection. In AIP Conference Proceedings (Vol. 2159, No. 1, p. 030013). AIP Publishing LLC.
23. Zheng, L., Liu, G., Yan, C., & Jiang, C. (2018). Transaction fraud detection based on total order relation and

behavior diversity. IEEE Transactions on Computational Social Systems, 5(3), 796-806.
24. Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J., & Liu, H. (2017). Feature selection: A data perspective. ACM computing surveys (CSUR), 50(6), 1-45.
25. Arora, S., & Singh, S. (2019). Butterfly optimization algorithm: a novel approach for global optimization. Soft Computing, 23(3), 715-734.
26. Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017, August). Understanding of a convolutional neural network. In 2017 international conference on engineering and technology (ICET) (pp. 1-6). Ieee.
27. Zaremba, W., Sutskever, I., &Vinyals, O. (2014). Recurrent neural network regularization. arXiv preprint arXiv:1409.2329.