Credit card fraud detection using moth-flame earthworm optimisation algorithm-based deep belief neural network

S. Deepika*
Reva University,
Rukmini Knowledge Park, Kattigenahalli,
Yelahanka, Bengaluru, Karnataka 560064, India
and
Department of CSE,
Anurag University,
Hyderabad, India
Email: asdeepik46@gmail.com
*Corresponding author

S. Senthil
Reva University,
Rukmini Knowledge Park, Kattigenahalli,
Yelahanka, Bengaluru, Karnataka 560064, India
Email: dir.csa@reva.edu.in

Abstract: Nowadays, credit card fraud actions happen commonly, which results in vast financial losses. Fraudulent transactions can take place in a variety of ways and can be put into various categories. Hence, financial institutions and banks put forward credit card fraud detection applications. To detect fraudulent activities, this paper proposes a credit card fraud detection system. The proposed system uses the database with the credit card transaction information and sends it to the pre-processing. The log transformation is applied over the database for data regulation in the pre-processing step. After, the appropriate features are selected by the information gain criterion, and the selected features are utilised to train the classifier. Here, a novel classifier, namely moth-flame earthworm optimisation-based deep belief network (MF-EWA-based DBN) is proposed for the fraud detection. The weights for the classifier are selected by the newly developed moth-flame earthworm optimisation algorithm (MF-EWA). The proposed classifier carries out the training and detects the fraud transactions in the database. The proposed MF-EWA-based DBN classifier has improved detection performance and outclassed other existing models with 85.89% accuracy.

Keywords: credit card transactions; fraud detection; information gain; earthworm optimisation algorithm; deep belief network.

1 Introduction

In recent years, data mining-based techniques are commonly adopted for exploration and analysis. The data mining techniques help in discovering useful patterns and rules through automatic or semi-automatic means. The emerging trend in internet applications has improved online money transactions and credit card systems. However, fraudsters may steal the details of the card transactions, as the online transactions have large security loops. Credit card frauds can be associated with the stealing of the physical card, or data, like account number, user key, etc., associated with the bank account. Hacking sensitive details from the users can result in loss of money to the users. Cardholders report the loss of stolen card quickly, but the hacker can hoard the compromised account for several weeks, thus, making the identification process to be difficult. In most cases, the cardholders can sense the data fraud only after receiving the billing statement, and thus, makes money recovery to be impossible. Credit fraud can be carried out in two ways, like online fraud and offline fraud. As gaining the real time transaction details for the data mining is a crucial challenge, an online transaction processing (OLTP) domain has been developed in recent years for managing the transaction details. The details of the transactions are available in the online database. The online database helps in other fields, like financial transactions, customer relationship management (CRM), and retail sales (Kavipriya and Geetha, 2018).

A card transaction can be declared as the fraud transaction when the credit card is used without the consent or approval of the original user. The fraudster does not have any intention to payback the user, and thus, severely affects the original user (Bhatla et al., 2003). In contrast to the common belief, the merchants are responsible for payment, while the fraudster makes the payment, which is common during the online transaction. The original user can file the charge sheet with the company when the transaction is done without his/her consent, and then, the necessary action has to be taken by the company.
Credit card fraud detection

and return the money to the original user. Merchants can also file a chargeback sheet and collect fine from card associations in case of product loss (Montague, 2010). Identifying fraud detection is a challenging problem as it is difficult to differentiate the actual transaction from the fraud transaction. The fraudsters steal the information from customers in various ways and make the transaction look as legitimate one (Whitrow et al., 2009). A million online transactions occur in the real world, the ratio of fraud cases to the legitimate case is very feasible, which creates an unbalanced dataset for data mining. Processing the unbalanced dataset with the data mining algorithms yields inefficient results. Also, processing the unbalanced dataset requires additional precautions. It is necessary to develop dynamic systems for adapting the new fraud patterns (Quah and Sriganesh, 2008). Fraudsters find an innovative way to steal the information, and thus, it is necessary for the fraud detection system to evolve rapidly than fraudsters (Carneiro et al., 2017). Credit card frauds are rare when compared with normal transactions. Hence, the resultant data is unbalanced as it has details regarding fraudulent events and regular transactions (Kulkarni and Ade, 2016).

Other than stealing the credit card information, the intruders may also steal the information by hacking the confidential web sites of banks, the plunder credit card itself (Ditzler and Polikar, 2013). Thus, it is necessary to develop the algorithmic framework for tackling the credit card fraud transaction. The credit card fraud detection system (Fiore et al., 2019; Carcillo et al., 2019) should deal with the unbalanced nature of the data available in the non-stationary environment. Further, the data has incremental fashion, and thus, maintaining the efficiency of the algorithm is necessary (Kulkarni and Ade, 2016). Some of the innovative ways for preventing fraud cases are the address verification system (AVS), chip and pin verification, and card verification code (CVV). Even though these techniques have advanced security schemes, they may fail in certain scenarios. Developing the fraud detection model for the credit card transaction faces a crucial challenge (Fawcett and Provost, 1997). Also, the database with the credit card transaction information has a large volume of information leading to impractical solutions in manual detection (Carneiro et al., 2017). In general, credit card fraud detection is viewed as the data mining classification problem, as it tends to classify the user transaction as legitimate or fraudulent. Developing data mining techniques for credit card fraud detection has its own challenges, as the collection of real-world data is difficult (Kavipriya and Geetha, 2018). Literature has developed many real time solutions for accurate analysis (Srivastava et al., 2008) and scientific solutions to credit card frauds (Kulkarni and Ade, 2016). A lot of supervised techniques, like artificial neural networks (ANN), support vector machines (SVM), and logistic regression (Carneiro et al., 2017), have been developed in the literature for focusing on data from banking and credit-card operations (Ghosh and Reilly, 1994).

The main aim of this paper is to design and develop a data mining technique to perform credit card fraud detection in an efficient manner. Initially, the credit card data is given as the input of the credit card fault detection system, which is then, pre-processed using log transformation in order to make the patterns in the data more interpretable and to reduce the complexities associated with the further processing. Then, the filtered data is fed to the feature selection process using the information gain so as to extract the highly significant features from the input. The extracted features are subjected to the classification process to classify the data samples using the deep belief network (DBN) (Vojić, 2016), which is trained with the proposed algorithm, moth-flame earth worm optimisation (MF-EWA) algorithm. The proposed MF-EWA algorithm is the integration
of the moth-flame optimisation algorithm (MFO) (Mirjalili, 2015) and the earth worm optimisation algorithm (EWA) (Wang et al., 2015). The proposed algorithm inherits the advantages of both the algorithms and it tunes the optimal weights of the DBN, for the detection of fraudulent behaviour.

The major contributions of this work for credit card fraud detection are depicted below:

- **MF-EWA-based DBN classifier**: this classifier is newly developed for detecting the fraud card transaction, in which the DBN is trained using the MF-EWA algorithm.
- **MF-EWA algorithm**: it is the novel optimisation algorithm developed by integrating MFO and EWA to train the DBN for fraud detection such that the credit card fraud detection system becomes effective, yielding a better solution.

Further, the paper organisation is done as follows: Section 1 deals with the introductory part for the credit card fraud detection, and Section 2 surveys some latest literature works related to the fraud detection algorithms. Section 3 presents the proposed credit card fraud detection model and explains the proposed MF-EWA-based DBN classifier. The simulation results of the proposed classifier are depicted and explained in Section 4 and Section 5 concludes the work.

## 2 Literature survey

Here, eight literary works related to credit card fraud detection are presented and the challenges in the schemes are discussed.

Carneiro et al. (2017) proposed a data mining approach involving both manual and automatic classification for fraud detection. The scheme developed a complete development process for detecting the frauds and thus, addresses the practical implementation issues. Further, the technique provides an exploratory analysis for improving the manual revision process, but the scheme lacks to provide the complete application for fraud detection. Fiore et al. (2017) proposed a detection framework with the Generative Adversarial Networks. The scheme provided an effective detection scenario as it is trained with the original data. The scheme was depended on the labelled instances, and thus, had effective fraud detection. However, it was not suitable for the unsupervised setting. Patil et al. (2018) developed the big data analytical framework for credit card fraud detection, and the model dealt with the large volume of information. The model had effective performance while dealing with real time data, and thus, provided low risk and high customer satisfaction. As the technique was based on logistic regression, handling the outlier points is not done effectively. Kulkarni and Ade (2016) had developed the universal framework using a logistic regression model for fraud detection. The technique adopted the logistic regression for tackling the challenges posed by the incremental learning. Further, the scheme adopts deep learning for knowledge transformation and preservation. The scheme failed to perform well in non-stationary environments, like Gaussian distribution.

John et al. (2017) developed the bank fraud detection model with the data mining technique. As the model used the data mining-based technique for the detection, the detection process had low risk and high customer satisfaction. The scheme does not allow new transactions without the branch authentication. Saia and Carta (2017) developed a
detection strategy with the discrete Fourier transform conversion so as to deal with the unbalanced dataset. It reduced the data heterogeneity problem by using the past legitimate transactions, and also, tackled the cold start issues. The model faced difficulties while dealing with small classes. Kavipriya and Geetha (2018) developed an efficient clustering and classification method for fraud detection in the credit card data. They adopted the algorithms, like Apriori and SVM for the improved clustering and classification. The model faced problems for handling different attacks, as it used only known patterns for training. Zareapoor and Yang (2017) proposed a supervised learning scenario for developing the fraud detection model. The scheme took less time for fraud detection, as the model was developed by crossing the smaller pattern databases. However, it had lacked to provide a strong pattern, and thus, had less supervised samples for training.

2.1 Challenges

Credit card fraud detection faces a lot of challenges and some of the challenges are listed below:

- One of the prime challenges in the design of the credit card fraud detection model is dealing with the imbalanced data environment. The occurrence of the fraudulent case to the normal case is very rare. Thus, collecting the balanced database for the research is a challenging task. There is more chance for the instance of one class to outstrip other resulting in inefficient classifiers (Zareapoor and Yang, 2017).

- Some of the researchers have adopted the statistical classification model for defining the credit score model in the detection system. The models use the statistical methods for classifying the transaction as good and bad. Even though the model provides good classification performance, most of the researches are not revealed in public due to the confidentiality reason, and a similar case can be applied to the data collection (Yeh and Lien, 2009).

- Logistic regression-based models have been used for defining the credit scoring applications, and are found to be as efficient as linear discriminant analysis (LDA). The logistic regression assumes strong models for identifying the detection, and it is limited to handle credit scoring problems (Lee et al., 2006).

- Using the neural network for fraud detection has significant advantages, as it provides different learning methods. But, the long training process of the NN can impact the real time performance of fraud detection (Lee et al., 2006).

- The LDA-based model while used for fraud detection assumes the data to be categorical nature, but in the real case, the covariance matrices of different classes in the database are dissimilar (Lee et al., 2006).

Since credit card becomes the most accepted mode of payment for both regular purchase and online, frauds related with credit card are also increasing. All the techniques discussed in this section have their own merits and demerits. This survey motivates us to develop a hybrid technique for effectively identifying fraudulent credit card transactions.
3 Proposed credit card fraud detection scheme using MF-EWA-based DBN classifier

This section presents the proposed credit card fraud detection model by developing a deep learner model. The basic block diagram of the proposed credit card fraud detection model using the proposed MF-EWA-based DBN classifier is given in Figure 1.

As given in the figure, the entire process is done in three steps:
1. pre-processing
2. feature selection
3. classification.

The data for the proposed model is the card transaction information, which has details about the normal transaction and the fraud transactions. The information in the database cannot directly be used for the analysis, and hence, pre-processed by applying the log transformation. The pre-processed data is subjected to the feature selection, as the data has millions of information, which may increase the classifier training time. The selection criterion is based on the information gain for selecting the highly appropriate features from the pre-processed data. Then, the selected features are provided to the DBN for the classification. Here, the MF-EWA algorithm is newly developed using EWA and MFO algorithms for tuning the DBN. The proposed classifier declares the transaction as an attack or normal based on the training.
3.1 Pre-processing

The input data for the credit card fraud detection is the large set of card transaction details acquired from various customers. The information obtained from the database may contain several misleading details leading to an increase in classification training time. So, as the initial step, the input data in the database is pre-processed and given a suitable structure. As the pre-processing step, the credit card transaction database is subjected to the log transformation, by applying the log function over the data. Consider the credit card transaction database $C$ is subjected to the pre-processing, and the resultant database after the log transformation is expressed as:

$$C_s = \log(C)$$  \hspace{1cm} (1)

where $C$ signifies the input database provided to the credit card fraud detection system, and $C_s$ refers to the pre-processed database and the database has the size of $Y \times X$, where $Y$ is the data records and $X$ is the number of features.

3.2 Feature selection through information gain

After the pre-processing, it is necessary to collect the useful features from the database $C_s$. The card transaction details may have unnecessary information for the training, and these irrelevant features are eliminated in the feature selection step. The features in the pre-processed database $C_s$ are subjected to the feature selection based on the information gain. The features satisfying above a threshold greater than the calculated information gain are selected for the next step.

The condition for selecting the $x^{th}$ feature for the training is given based on the information gain as follows:

$$F_x = \begin{cases} f_x = 1; & IG(f_x) > T \\ f_x = 0; & IG(f_x) \leq T \end{cases}$$  \hspace{1cm} (2)

where $f_x$ signifies the $x^{th}$ feature in the pre-processed database $C_s$. The term $IG(f_x)$ refers to the information gain of the $x^{th}$ feature and $T$ specifies the selection threshold. Then, the selected features are collected in the database specified as $F_x$ and it has the size as $Y \times N$. The size of the features selected in the $F_x$ is less than the actual size of the database $X$. The information gain defines the change in the entropy. As the database has a lot of features, selecting the features based on the information helps in finding the features with the more appropriate attributes. The information gain (Kent, 1983) of the $x^{th}$ feature in the database is expressed as,

$$IG(f_x) = \Delta E$$  \hspace{1cm} (3)

where $E$ refers to the entropy of the feature and it is specified as,

$$E = -\sum_{x=1}^{X} p_x \log p_x$$  \hspace{1cm} (4)

where $p_x$ indicates the conditional probability for the occurrence of the $x^{th}$ feature.
3.3 Classification using the proposed MF-EWA-based DBN for the detection of fraudulent transactions

The features selected from the database $F_i$ are directly provided to the classifier for detecting the credit card fraud cases. For the detection, this work has developed a novel optimisation-based DBN classifier, namely MF-EWA-based DBN, which is elaborated as follows,

3.3.1 Architecture of the proposed MF-EWA-based DBN

The basic architecture of the proposed MF-EWA-based DBN classifier is explained in this subsection and is presented in Figure 2.

As shown in Figure 2, the classifier has three different layers, like restricted Boltzmann machine (RBM) 1, RBM 2, and Multilayer perceptron (MLP). The features are directly fed to the RBM 1 layer, which is interconnected with the RBM 2 and consequently with the MLP with the weighted neurons. The RBM layers have the input and the hidden neurons, and the MLP layer has the additional output layer.

The features selected through the information gain are directly fed to the input of the RBM1, and it is expressed in the following way:

$$c^1 = \{c^1_1, c^1_2, \ldots, c^1_p, \ldots, c^1_N\}; \quad 1 \leq p \leq N$$

where $c^1_p$ refers to the $p^{th}$ feature fed as the training input, and it is given to the $p^{th}$ input neuron of the RBM 1 layer. As $N$ features are selected in the feature selection process, the RBM 1 layer is designed with $N$ input neurons. Then, the input is processed through the hidden neurons, expressed as,
where \( y_u^1 \) refers to the \( u \)th neuron in the hidden unit of the RBM 1 layer. For the training, the RBM 1 considers weighted hidden units higher in count than the input neurons. Let us consider the RBM 1 layer has \( K \) number of hidden units and the range of \( u \) extends in the interval \([1, K]\). The RBM 1 layer has the bias for modulating the input features and the hidden units. The biases constituted in the input and the hidden units of RBM 1 can be given as,

\[
g^1 = \{g_1^1, g_2^1, \ldots, g_p^1, \ldots, g_N^1\} \tag{7}
\]

\[
h^1 = \{h_1^1, h_2^1, \ldots, h_u^1, \ldots, h_K^1\} \tag{8}
\]

where \( g_p^1 \) indicates the bias for the \( p \)th input unit and \( h_u^1 \) refers to the bias for the \( u \)th hidden unit. The hidden units of the RBM layer are provided with the weighted neurons so as to improve the classification scenario, and thus, the RBM 1 layer has \( K \) weighted neurons in the RBM 1 layer expressed as,

\[
D^1 = \{D_{pu}^1\}; \quad 1 \leq p \leq N; 1 \leq u \leq K \tag{9}
\]

where \( D_{pu}^1 \) indicates the weight provided between the \( p \)th input unit and the \( u \)th hidden unit. The hidden layer output depends on the weights, bias, and the input feature. The activation function \( \sigma \) also has an influence on the hidden layer output and the output provided by the hidden unit of the RBM 1 is expressed as,

\[
y_u^1 = \sigma \left[ h_u^1 + \sum_p c_{pu}^1 D_{pu}^1 \right] \tag{10}
\]

where \( \sigma \) indicates the activation function for the hidden unit of RBM 1, and the output of the hidden unit can be expressed as,

\[
y^1 = \{y_u^1\}; \quad 1 \leq u \leq K \tag{11}
\]

where \( y_u^1 \) indicates the \( u \)th hidden layer of RBM layer 1. As the DBN considered for the classification has two RBM layers, and they are interconnected with each other, the output of the RBM 1 is fed as the training input to the RBM 2. Hence, the input units of the RBM 2 have a similar number of neurons as the hidden units of RBM 1, and the input layer of RBM 2 is specified as,

\[
c^2 = \{c_1^2, c_2^2, \ldots, c_K^2\} = \{y_u^1\}; \quad 1 \leq u \leq K \tag{12}
\]

And similarly, the hidden units of RBM 2 are expressed as,

\[
y^2 = \{y_1^2, y_2^2, \ldots, y_u^2, \ldots, y_K^2\}; \quad 1 \leq u \leq K \tag{13}
\]

The RBM 2 has the bias in its input and hidden units, expressed as \( g^2 \) and \( h^2 \), respectively. The weighted neurons provided amidst the input and hidden units of RBM 2 can be represented as,
where $D_{uu}$ indicates the weight provided in the $u^{th}$ input neuron and $u^{th}$ hidden neuron of RBM 2. The expression of the RBM layer 2 output, provided by the hidden unit is expressed as,

$$y^2_u = \sigma\left(h^2_u + \sum_p c^2_p D_{uu}\right) \quad \forall c^2_p = y^1_u \quad (15)$$

Now, the output of each hidden unit of RBM 2 can be expressed via the following expression,

$$y^2 = \{y^2_u\} \quad 1 \leq u \leq K \quad (16)$$

The MLP layer has significant importance in the DBN as it further refines the classification process. The RBM 2 provides the hidden layer output to the input layer of the MLP, and the input unit in MLP is expressed as,

$$a = \{a_1, a_2, \ldots, a_u, \ldots, a_K\} = \{y^2_u\} \quad 1 \leq u \leq K \quad (17)$$

where $a_u$ indicates the $u^{th}$ input of the MLP, and the hidden unit of MLP is given as,

$$b = \{b_1, b_2, \ldots, b_r, \ldots, b_s\} \quad 1 \leq r \leq s \quad (18)$$

The hidden units of the MLP have $s$ number of layers. The output of the MLP is specified as,

$$O = \begin{cases} 
1; & \text{fraudulent} \\
0; & \text{legitimate} 
\end{cases} \quad (19)$$

For finding the output of the MLP layer, it is necessary to define the weights of the MLP. The MLP layer has weights in both the input and hidden units as specified in below expressions,

$$D^I = \{D^I_{uw}\} \quad 1 \leq u \leq K; 1 \leq r \leq s \quad (20)$$

$$D^H = \{D^H_{wr}\} \quad 1 \leq r \leq s \quad (21)$$

where $D^I_{uw}$ indicates the weight of the input MLP unit, and $D^H_{wr}$ specifies the weight of the hidden unit in the MLP. Based on the input weights and input features, the hidden unit outputs are computed as,

$$b_r = \left[ \sum_{u=1}^K D^I_{uw} \cdot a_u \right] B_r \quad \forall a_u = y^2_u \quad (22)$$

where $B_r$ indicates the bias provided to the MLP hidden unit. The final expression for the MLP layer output is expressed as,

$$O = \sum_{r=1}^s D^H_{wr} \cdot b_r \quad (23)$$
3.3.2 Proposed MF-EWA algorithm

For training the DBN, a new optimisation algorithm, namely MF-EWA, has been developed in this work. MF-EWA algorithm has the characteristics of both MFO (Mirjalili, 2015) and EWA (Wang et al., 2015) in defining the search space and the optimal solution. The MFO algorithm behaves based on the properties of the moths, where the moths try to change the direction based on the light intensity. The moths follow a traverse direction by attracting towards the direction of light. Here, the position update of the moths is modified based on the EWA algorithm. In the EWA algorithm, the solution space is modified based on the earthworm population. The solution space of the EWA gets influenced based on the properties of crossover and mutation. Here, the proposed MF-EWA algorithm considers the position update of EWA based on the mutation operator to influence the position of moths in MFO. The algorithmic steps of the proposed MF-EWA are explained as follows:

- **Moth initialisation**: as the initialisation step, the position of moths is randomly selected. In the MFO algorithm, the solution space considers a 2-dimensional space with the position of the moth and the flame. The randomly initialised population $M_i$ has random positions, and the terms $i$ and $j$ refer to the position of the moth and flame respectively.

- **Fitness evaluation**: the fitness for the proposed MF-EWA is based on the error estimate provided by the DBN classifier. The DBN classifier tries to identify the credit card fraud by minimising the classification error and this algorithm uses the classification error expressed in equation (34) as the fitness criteria for the optimal weight selection.

- **Solution update**: in the MFO, the position of the moths is changed according to the position of the flame. The update of the MFO as specified in the standard algorithm is expressed as,

$$M_i(t + 1) = d_i e^{bt} \cos(2\pi t) + S_j$$

where $d_i$ indicates the distance between the moth and the flame, and $b$ signifies a constant for defining the search space. The term $S_j$ refers to the position of the $j^{th}$ flame in the solution and $M_i(t + 1)$ indicates the updated position of the moth for the iteration $(t + 1)$. In the above expression, the distance measure $d_i$ is calculated based on the position of the $i^{th}$ moth and the $j^{th}$ flame in the solution space, and it is represented as $d_i = |S_j - M_i(t)|$. The position of the flame is considered to be fixed, and the $i^{th}$ moth $M_i(t)$ moves around the flame. Now, the equation (24) is expressed as,

$$M_i(t + 1) = |S_j - M_i(t)| e^{bt} \cos(2\pi t) + S_j$$

Here, the proposed algorithm considers the condition $S_j > M_i(t)$ for solving the above expression, and the above equation is modified as follows:
Solving the above expression,

\[ M_\phi(t+1) = S_j \left[ 1 + e^{i\phi} \cos(2\pi t) \right] - M_\phi(t) e^{i\phi} \cos(2\pi t) \]  \hspace{1cm} (27)

Equation (27) indicates the actual position update of the moth, and in the proposed MF-EWA, the update process of the MFO is modified based on the EWA. EWA depends on the characteristics of the earthworm population. The position update depends on various factors, like selection, mutation, crossover, etc. The position of the EWA is greatly influenced by the Cauchy mutation operator, and the update specified by the EWA is given as follows:

\[ M_\phi(t+1) = M_\phi(t) + W_j \ast A \]  \hspace{1cm} (28)

where \( W_j \) indicates the position operator and it depends on the population size and \( A \) indicates the Cauchy distribution constant, having the values as \([0, 1]\). From equation (28), the current iteration position of the solution \( M_\phi(t) \) is identified, and it is expressed as,

\[ M_\phi(t) = M_\phi(t+1) - W_j \ast A \]  \hspace{1cm} (29)

As the update has to be dependent on the previous iteration position after the substitution of equation (29) in (27), it is modified by multiplying R.H.S and L.H.S by 2,

\[ 2M_\phi(t+1) = 2S_j \left[ 1 + e^{i\phi} \cos(2\pi t) \right] - 2M_\phi(t) e^{i\phi} \cos(2\pi t) \]  \hspace{1cm} (30)

\[ 2M_\phi(t+1) = 2S_j \left[ 1 + e^{i\phi} \cos(2\pi t) \right] - M_\phi(t) e^{i\phi} \cos(2\pi t) - M_\phi(t) e^{i\phi} \cos(2\pi t) \]  \hspace{1cm} (31)

Now, substituting equation (29) in equation (31),

\[ 2M_\phi(t+1) = 2S_j \left[ 1 + e^{i\phi} \cos(2\pi t) \right] - M_\phi(t) e^{i\phi} \cos(2\pi t) - (M_\phi(t+1) - W_j \ast A) e^{i\phi} \cos(2\pi t) \]  \hspace{1cm} (32)

\[ M_\phi(t+1) = \frac{1}{2 + e^{i\phi} \cos(2\pi t)} \left[ 2S_j \left[ 1 + e^{i\phi} \cos(2\pi t) \right] - M_\phi(t) e^{i\phi} \cos(2\pi t) \right] + W_j \ast A e^{i\phi} \cos(2\pi t) \]  \hspace{1cm} (33)

The above expression indicates the modified position update for the proposed MF-EWA algorithm that combines the update rule of both EWA and MFO, improving the compatibility of the solution, and thereby, making the update process more effective. In this step, the position of the randomly initialised solutions is updated based on equation (33), and the new position of the moths is identified.
• **Finding the best solution:** the solution gets updated in the previous step, and the new position for the moth is identified. In this step, the fitness of the newly updated solution is identified based on the error minimisation. As the algorithm considers the solution with minimal error as the optimal solution, the solution with low error is identified and is considered as the best solution. Once the best solution is identified, the previous solution is replaced with the best solution.

• **Termination:** the algorithm gets terminated at the maximum iteration limit, and the final solution in the optimisation space is considered as the optimal weight for the MLP training.

### 3.3.3 Training phase of MF-EWA-based DBN classifier

The DBN classifier training procedure is explained in this section. The RBM layer and the MLP layer of the DBN are trained through the training algorithm separately for finding the suitable weights and bias for the classifier. The weights and the bias for the RBM layer are identified based on the backpropagation algorithm. The MLP layer considers both the gradient descent algorithm and the newly developed MF-EWA algorithm for the position update. The training procedure of the MLP is expressed as follows:

1. As the initial step, the weights and bias depicted in the input and hidden units of MLP are randomly chosen. The randomly chosen input and the hidden layer weights, \( D_i \) and \( D_h \), are initialised based on expressions (20) and (21).

2. Provide the input to the MLP layer from the RBM layer 2 and the training sample to the MLP layer is expressed as \( \{ y_{ui} \} \).

3. The randomly initialised weights and the training sample are used to compute the hidden layer and the output layer values, and the calculation is carried on based on equations (22) and (23), respectively.

4. Here, the error-based fitness function is derived for the classification. The fitness is considered to be a minimal average error, which is calculated from the computed output \( O \) and the ground truth response \( R \). The expression for the average error of the MLP training is expressed in the following equation,

\[
Q_{avg} = \frac{1}{n} \sum_{i=1}^{n} (O^{i} - R^{i})^2
\]

where \( O \) refer to the computed output, and \( R \) is the ground truth response.

5. Then, the randomly initialised weights are updated based on the gradient descent algorithm. The gradient descent algorithm calculates the update based on the partial derivative, calculated as,

\[
\Delta D_{iw} = -\eta \frac{\partial Q_{avg}}{\partial D_{iw}} \quad \text{(35)}
\]
\[ \Delta D_i^H = -\eta \frac{\partial Q_{avg}}{\partial D_i^H} \]  

(36)

where \( \eta \) indicates the learning rate for the DBN training.

6 Both the input and the hidden layer weights are modified based on the gradient descent algorithm, and they are computed as,

\[ D_{ur}^I(t + 1) = D_{ur}^I(t) + \Delta D_{ur}^I \]  

(37)

\[ D_{r(G)}^H(t + 1) = D_{r(G)}^H(t) + \Delta D_r^H \]  

(38)

where \( \Delta D_{ur}^I \) and \( \Delta D_r^H \) indicate the change in the input and hidden layer weight through the gradient descent algorithm.

7 The weights of the input and the hidden layer units updated through the proposed MF-EWA algorithm are specified as follows:

\[ D_{ur(MF-EWA)}^I(t + 1) = \frac{1}{[1 + e^{j\beta} \cos(2t)]} \left[ S_j \left[ 1 + e^{j\beta} \cos(2\pi t) \right] + W_j * C e^{j\beta} \cos(2\pi t) \right] \]  

(39)

\[ D_{r(MF-EWA)}^H(t + 1) = \frac{1}{[1 + e^{j\beta} \cos(2\pi)]} \left[ S_j \left[ 1 + e^{j\beta} \cos(2\pi t) \right] + W_j * C e^{j\beta} \cos(2\pi t) \right] \]  

(40)

Equation (39) specifies the weight update for the input layer weights, and expression (40) indicates the weight update for the hidden layer units.

8 Finally, the weights are identified through the minimal error as the fitness, and the weights providing the minimal error are identified as the optimal weights for the MLP layer.

9 The steps in the MLP training are continued for the iteration limit, and the final solution provides the optimal weight and based on the optimal weight, the classification is carried out.

After the training, the MLP layer computes the classified response based on the optimal weights through the MF-EWA algorithm. While the test data arrives into the classifier, the DBN classifier provides the response as 1 for the fraud detection and 0 for the normal transaction.

4 Results and discussion

This section presents the simulation results of the MF-EWA-based DBN for the credit card fraud detection analysis, and the results are compared with the several states of art techniques.
4.1 Experimental setup

The implementation of the proposed MF-EWA-based DBN is done in the JAVA tool and the PC configuration used for the implementation is 4 GB RAM, Windows 10 OS, and Intel i3 processor.

4.1.1 Database description

In this work, the Credit Card Fraud Detection database (https://www.kaggle.com/mlg-ulb/creditcardfraud) is utilised for the processing of the proposed MF-EWA-based DBN classifier. The database has credit card transaction information done by the European cardholders. It has the details of 284,807 card transactions and 492 fraud cases. The database is highly imbalanced, such that the fraud positive class is 0.172 % of the entire transaction. The database has in total of 28 feature classes.

4.1.2 Performance metrics

For evaluating the performance of the MF-EWA-based DBN classifier, three metrics, like accuracy, sensitivity, and specificity, which are explained below, are employed.

- **Accuracy**: the accuracy defines the closeness of the credit detection approach to clearly identify the fraud cases from the entire data, and it is expressed as,

  \[
  \text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}
  \]

  where TP, TN, FP, and FN signify true positive, true negative, false positive, and false negative.

- **Sensitivity**: the sensitivity measure defines the closeness to identify the fraud positive class, and it clearly differentiates the correctly detected fraud cases.

  \[
  \text{Sensitivity} = \frac{TP}{TP + FN}
  \]

- **Specificity**: the specificity measure defines the total false cases eliminated by algorithm, i.e. elimination of correct transactions.

  \[
  \text{Specificity} = \frac{TN}{TN + FP}
  \]

4.1.3 Comparative techniques

The performance of the proposed MF-EWA-based DBN classifier is compared with the three other existing models, such as development and deployment technique (DDT) (Carneiro et al., 2017), k-NN (Zhang and Zhou, 2015), and deep learning model (Hinton, 2009). The techniques are explained as follows:
• **DDT**: in Carneiro et al. (2017), DDT is developed for fraud detection, and the model was suitable for the detection in the e-tail services. Here, the detection process is carried out by employing both the manual and the automatic classifiers.

• **k-NN**: the k-NN classifier performs the classification of fraud detection by defining \( k \) classes based on the nearest neighbour data.

• **Deep learner**: here, the DBN is used for the classification, and the training of DBN is carried out by standard procedure.

### 4.2 Comparative analysis

This section presents the comparative analysis of the proposed MF-EWA-based DBN classifier for credit card fraud detection. The comparative analysis is done by modifying the threshold of training features selected through the information gain parameter. Here, the feature size is chosen to be 7, 8 and 9.

#### 4.2.1 Comparative analysis for the feature size as 7

Figure 3 presents the comparative analysis of the proposed MF-EWA-based DBN classifier while using the features of size 7. According to the sensitivity analysis depicted in Figure 3(a), the existing models, like DDT, k-NN, and the deep learner has the sensitivity as 0.375, 0.5, and 0.7003 for 50 percentage data training. For the same data training, the proposed MF-EWA-based DBN classifier has the sensitivity as 0.7689. For higher training percentage of 90, the existing models, DDT, k-NN, and deep learner have the sensitivity as 0.42, 0.7003, and 0.75, but the proposed MF-EWA-based DBN has high sensitivity as 0.7985. Figure 3(b) presents the comparative performance of the proposed MF-EWA-based DBN based on the specificity metric with the feature size as 7. At the data training percentage of 50, DDT, k-NN, and deep learner have the specificity as 0.3245, 0.333, and 0.5. Meanwhile, the proposed MF-EWA-based DBN has the specificity as 0.7297. For the data percentage as 90, the specificity of DDT, k-NN, and deep learner remains as 0.5, 0.5, and 0.7674, while the proposed MF-EWA-based DBN has attained a better value of 0.833. Figure 3(c) shows the comparative analysis of the proposed MF-EWA-based DBN for the feature size as 7 based on accuracy measure. At the data training percentage of 50, DDT, k-NN, and deep learner have the accuracy as 0.4285, 0.5161, and 0.7003. Meanwhile, the proposed MF-EWA-based DBN has the accuracy as 0.7589. For the data percentage as 90, the accuracy of the existing works, DDT, k-NN, and deep learner is 0.5806, 0.7003, and 0.7778, while the proposed MF-EWA-based DBN has attained better accuracy of 0.8.
Figure 3  Comparative analysis of the proposed MF-EWA-based DBN for the feature size as 7, (a) sensitivity (b) specificity (c) accuracy (see online version for colours)
4.2.2 Comparative analysis for the feature size as 8

Figure 4 depicts the comparative analysis of the proposed MF-EWA-based DBN classifier while using the size of the features as 8 for training. According to the sensitivity analysis depicted in Figure 4(a) for feature size 8, the existing models, like DDT, k-NN, and deep learner have the sensitivity as 0.4, 0.4772, and 0.7003, for 50 percentage data training. For the same data training, the proposed MF-EWA based DBN classifier has the sensitivity as 0.8753. For higher training percentage of 90, DDT, k-NN, and deep learner have the sensitivity as 0.3684, 0.5, and 0.7003, but the proposed MF-EWA-based DBN has high sensitivity as 0.8757. Figure 4(b) presents the comparative performance of the proposed MF-EWA-based DBN based on the specificity metric for the feature size as 8. At the data training percentage of 50, DDT, k-NN, and deep learner have the specificity as 0.1111, 0.333, and 0.5, while the proposed MF-EWA-based DBN has the specificity as 0.7959. For the data percentage as 90, the specificity of the existing DDT, k-NN, and deep learner have the specificity of 0.3846, 0.5, and 0.571, while the proposed MF-EWA-based DBN has attained better value of 0.7222. Figure 4(c) shows the comparative analysis of the proposed MF-EWA-based DBN for the feature size as 8 based on accuracy measure. At the data training percentage of 50, the existing methods, DDT, k-NN, and deep learner have the accuracy as 0.375, 0.6451, and 0.7003. Meanwhile, the proposed MF-EWA-based DBN has the accuracy as 0.8156. For the data percentage as 90, the accuracy of DDT, k-NN, and deep learner is 0.5053, 0.5454, and 0.7003, while the proposed MF-EWA-based DBN has attained better value of 0.8465.

4.2.3 Comparative analysis for the feature size as 9

Figure 5 presents the comparative analysis of the proposed MF-EWA-based DBN classifier while selecting nine features for the training. According to the sensitivity analysis depicted in Figure 5(a) for feature size 9, the existing models, like DDT, k-NN, and deep learner have the sensitivity as 0.4, 0.4117, and 0.7003 for 50 percentage data training. For the same data training, the proposed MF-EWA-based DBN classifier has the sensitivity as 0.8839. For higher training percentage of 90, DDT, k-NN, and deep learner have the sensitivity as 0.3333, 0.4565, and 0.7003, but the proposed MF-EWA-based DBN has high sensitivity as 0.8885. Figure 5(b) presents the comparative performance of the proposed MF-EWA-based DBN based on the specificity metric with the feature size as 9. At the data training percentage of 50, DDT, k-NN, and deep learner have the specificity as 0.5, 0.6, and 0.7272. Meanwhile, the proposed MF-EWA-based DBN has the specificity as 0.7619. For the data percentage as 90, the specificity of the existing techniques, DDT, k-NN, and deep learner is 0.2857, 0.5, and 0.5, while the proposed MF-EWA-based DBN has attained better value of 0.7872. Figure 5(c) shows the comparative analysis of the proposed MF-EWA-based DBN for the feature size as 9 based on accuracy measure. At the data training percentage of 50, DDT, k-NN, and deep learner have the accuracy as 0.5, 0.5698, and 0.70037, while the proposed MF-EWA-based DBN has the accuracy as 0.8156. For the data percentage as 90, the accuracy values of DDT, k-NN, and deep learner are 0.3, 0.6236, and 0.7003, while the proposed MF-EWA-based DBN has attained better value of 0.8589.
Figure 4  Comparative analysis of the proposed MF-EWA-based DBN for the feature size as 8, (a) sensitivity (b) specificity (c) accuracy (see online version for colours)
Figure 5  Comparative analysis of the proposed MF-EWA-based DBN for the feature size as 9, (a) sensitivity (b) specificity (c) accuracy (see online version for colours)
4.3 Comparative discussion

Here, the comparative discussion of the proposed credit card fraud detection scheme is presented and the values are presented in Table 1. The best performance attained by each comparative model is discussed below and the results are presented in terms of accuracy, sensitivity, and specificity.

Table 1 Comparative discussion

<table>
<thead>
<tr>
<th>Comparative techniques</th>
<th>Performance of algorithms</th>
<th>Sensitivity</th>
<th>Specificity</th>
<th>Accuracy</th>
<th>Computational time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDT</td>
<td></td>
<td>0.3333</td>
<td>0.5</td>
<td>0.3</td>
<td>12.5</td>
</tr>
<tr>
<td>k-NN</td>
<td></td>
<td>0.4565</td>
<td>0.5</td>
<td>0.6236</td>
<td>11</td>
</tr>
<tr>
<td>Deep learner</td>
<td></td>
<td>0.7003</td>
<td>0.7674</td>
<td>0.7003</td>
<td>11.3</td>
</tr>
<tr>
<td>Proposed MF-EWA-based DBN</td>
<td></td>
<td>0.8885</td>
<td>0.8333</td>
<td>0.8589</td>
<td>5.6</td>
</tr>
</tbody>
</table>

The credit card fraud detection scheme developed in this work has attained improved performance in comparison with other existing models, for each evaluation metric. Using deep learner for fraud detection has shown the best performance of 0.7003, 0.7674, and 0.7003 as sensitivity, specificity, and accuracy, respectively. The proposed MF-EWA-based DBN classifier has improved fraud detection performance with sensitivity as 0.8885, specificity as 0.8333, and accuracy as 0.8589. The computational time of the proposed method is 5.6 sec, which is minimum than the computational time of the other comparative methods. From the comparative analysis, it can be concluded adopting the MF-EWA-based DBN classifier effectively improves credit card fraud detection.

5 Conclusions

This work develops a credit card fraud detection model by employing a deep learner with the optimisation algorithm. The database having the card transaction information is considered as the input data, and the log transformation is applied as the pre-processing step. The model uses the information gain as the criterion to select the appropriate features for the training. Then, the selected features are provided to the proposed MF-EWA-based DBN classifier for fraud detection. Here, the MF-EWA algorithm is newly developed through the integration of the MFO and EWA so as to tune the weights of DBN during the training phase. The proposed MF-EWA-based DBN classifier classifies the information according to the optimal weights and effectively identifies the credit card related frauds. For the experimentation, this work uses the credit card fraud detection database, and the results are evaluated based on the three metrics, accuracy, sensitivity, and specificity. The proposed MF-EWA-based DBN classifier has improved fraud detection performance with sensitivity as 0.8885, specificity as 0.8333, and accuracy as 0.8589. The drawback of the proposed method is that the overlapping of data in the preparation of credit card transaction data. In the future, we will use the combination of advanced deep learning techniques and the data mining techniques for fraud detection to further improve the performance.
References


Vojt, J. (2016) Deep Neural Networks and Their Implementation, Department of Theoretical Computer Science and Mathematical Logic, Prague.


