Fraud detection in bank transaction with wrapper model and Harris water optimization-based deep recurrent neural network

Chandra Sekhar Kolli and Uma Devi Tatavarthi
Department of Computer Science, Gandhi Institute of Technology and Management, Deemed University, Visakhapatnam, India

Received 25 April 2020
Revised 19 June 2020
Accepted 3 July 2020

Abstract

Purpose – Fraud transaction detection has become a significant factor in the communication technologies and electronic commerce systems, as it affects the usage of electronic payment. Even though, various fraud detection methods are developed, enhancing the performance of electronic payment by detecting the fraudsters results in a great challenge in the bank transaction.

Design/methodology/approach – This paper aims to design the fraud detection mechanism using the proposed Harris water optimization-based deep recurrent neural network (HWO-based deep RNN). The proposed fraud detection strategy includes three different phases, namely, pre-processing, feature selection and fraud detection. Initially, the input transactional data is subjected to the pre-processing phase, where the data is pre-processed using the Box-Cox transformation to remove the redundant and noise values from data. The pre-processed data is passed to the feature selection phase, where the essential and the suitable features are selected using the wrapper model. The selected feature makes the classifier to perform better detection performance. Finally, the selected features are fed to the detection phase, where the deep recurrent neural network classifier is used to achieve the fraud detection process such that the training process of the classifier is done by the proposed Harris water optimization algorithm, which is the integration of water wave optimization and Harris hawks optimization.

Findings – Moreover, the proposed HWO-based deep RNN obtained better performance in terms of the metrics, such as accuracy, sensitivity and specificity with the values of 0.9192, 0.7642 and 0.9943.

Originality/value – An effective fraud detection method named HWO-based deep RNN is designed to detect the frauds in the bank transaction. The optimal features selected using the wrapper model enable the classifier to find fraudulent activities more efficiently. However, the accurate detection result is evaluated through the optimization model based on the fitness measure such that the function with the minimal error value is declared as the best solution, as it yields better detection results.

Keywords Water wave optimization (WWO), Harris hawks optimization (HHO), Wrapper model, Credit card fraud, Deep recurrent neural network

Paper type Research paper

1. Introduction

Owing to the rapid growth of e-commerce, the usage of credit cards is dramatically increased for purchasing the products. However, the fraudulent usage of credit cards becomes major attention to the criminals. Because of the weakness of security in processing the existing credit card, the occurrence of card fraud is dramatically increasing, and such that it results in the loss of billions of dollars every year. Accordingly, the fraudsters are using some sophisticated methods for penetrating the credit card fraud. However, the fraudulent actions present major challenges to the financial institutions and banks that issue the credit cards (Darwish, 2020; Eweoya *et al.*, 2019; de Sá *et al.*, 2018). In recent

Kybernetes
© Emerald Publishing Limited
0368-492X
DOI 10.1108/K-04-2020-0239

decades, the amount of credit card transactions is increasingly growing owing to the rapid growth of e-service, such as e-commerce, mobile payments and e-finance and the popularization of the credit card. However, different transaction scenarios and the large-scale usage of credit cards without supervision and rigid verification result in the loss of billions of dollars caused by the fraud (Zhang *et al.*, 2019; Carneiro *et al.*, 2017). However, an estimation of accurate loss is very difficult, as the card issuers hesitant to liberate the statistics (Zhang *et al.*, 2019; Bolton and Hand, 2002), and some publicly available information shows the severity of fraud (Zhang *et al.*, 2019). However, credit card fraud detection poses various challenges, one of the challenges is that the feature set that describes the transaction using the credit card specifically ignore some sequential data. For instance, only the typical model uses raw features, such as merchant category, amount and time (Lucas *et al.*, 2020; Nami and Shajari, 2018; Zhang *et al.*, 2019).

Credit card fraud detection has become a major issue in financial organizations and to credit card users. To detect the less number of fraudulent transactions may protect the large volume of money and hence detecting the fraud has become a major issue for the researchers (Taha and Malebary, 2020). However, the usage of large-scale cards and the deficiency in providing the security system result in the loss of billions of dollars to the credit card fraud (Taha and Malebary, 2020). The credit card firms are not willing to declare the facts, so it is very crucial to find the approximation of card loss. Accordingly, certain information that corresponds to the financial loss that is occurred by the credit card fraud is freely accessible. Moreover, to use the credit card without providing security results in financial loss (Taha and Malebary, 2020; Carneiro *et al.*, 2017). Because of the credit card fraud, the global financial loss results in $22.8bn in the year 2017 and is expected to increase to $31bn by 2020 (Lebichot *et al.*, 2019; Taha and Malebary, 2020; John and Naaz, 2019). The credit card fraud is classified into two different types, namely, behavioral fraud (Bolton and Hand, 2002) and application fraud (Zhang *et al.*, 2019). The application fraud represents the situation where the application for the credit card is fraudulent, and it occurs while applying for the new credit card by the fraudster based on the card issuer and the identity information. The behavior fraud occurs only after approving and issuing the credit card such that the behavior fraud reflects the fraudulent behavior. Accordingly, fraud prevention and fraud detection become a major concern for the card issuers. It is a significant research topic for the researchers, as preventing and detecting the few portions of fraudulent action may save a large volume of money. Accordingly, fraud detection is classified into two types, as misuse detection and anomaly detection (Pumsirirat and Yan, 2018; Seyedhossein and Hashemi, 2010). However, the anomaly detection (More Umesh Katkar *et al.*, 2018) uses normal transaction for detecting the frauds, whereas the misuse detection model uses the labeled transaction to detect the frauds (Pumsirirat and Yan, 2018; Singh and Jain, 2019; Fiore *et al.*, 2019).

Because of the enormous amount of credit card transactions, verifying each transaction manually to detect the fraud is crucial for the credit card issuers. Hence, statistical or machine learning techniques are widely used for detecting the fraudulent transaction. However, there exist various good reviews in detecting the frauds using machine learning techniques (Zhang *et al.*, 2019; Abdallah *et al.*, 2016; Bolton and Hand, 2002). Various existing research works use statistical methods based on shallow architecture, such as neural network (NN), hidden Markov model (HMM), logistic regression and support vector machine (Marsaline *et al.*, 2014) for fraud detection. However, the model with the shallow architecture contains a single layer of nonlinear transformation, which maps the raw data into the feature space. With the extension to shallow architecture, the deep architecture is modeled called as deep learning (Menaga and Revathi, 2020) that uses multiple layers and stacks the layers in the hierarchical structure (Zhang *et al.*, 2019; Lucas *et al.*, 2019). In the past decades, NN (Chithra and Jagatheeswari, 2019) is introduced for detecting the credit card frauds. Recently, the deep learning that is the subfield of machine learning is used for fraud detection in-band transaction (Pumsirirat and Yan, 2018). However, the deep learning methods reported outperformed results than the existing machine learning techniques in the field of pattern recognition, speech coding and information retrieval (Zhang *et al.*, 2019; Abdel-Hamid *et al.*, 2014; Hayat *et al.*, 2014). Therefore, the deep learning is successfully applied in credit card fraud detection. Accordingly, the fraud detection model is regarded as the predictive model in the transactional behavior, where the behavior of past transactions is used for predicting the legitimacy of recent transaction behavior. However, feature engineering is expected to model feature variables that are used to represent the behavior of transactions with the transactional records (Zhang *et al.*, 2019; Carcillo *et al.*, 2018; Singh *et al.*, 2019).

This research is focused to design the fraud detection mechanism using the proposed HWO-based deep RNN. The proposed fraud detection strategy includes three different phases, namely, pre-processing, feature selection and fraud detection. Initially, the input transactional data is subjected to the pre-processing phase, where the data is pre-processed using the Box-Cox transformation to remove the redundant and noise values from data. The pre-processed data is passed to the feature selection phase, where the essential and the suitable features are selected using the wrapper model. The selected feature makes the classifier to perform better detection performance. Finally, the selected features are fed to the detection phase, where the deep recurrent neural network (deep RNN) classifier is used to achieve the fraud detection process such that the training process of the classifier is done by the proposed Harris water optimization (HWO) algorithm.

The major contribution of this research is explained as follows:

> *Proposed HWO-based deep RNN*: An effective fraud detection method named HWO-based deep RNN is designed to detect the frauds in the bank transaction. The optimal features selected using the wrapper model enable the classifier to find fraudulent activities more efficiently. However, the accurate detection result is evaluated through the optimization model based on the fitness measure such that the function with the minimal error value is declared as the best solution, as it yields better detection results.

The paper is organized as follows: Section 2 describes the review of various fraud detection methods, and Section 3 explains the proposed HWO-based deep RNN classifier to detect fraudulent activities. Section 4 elaborates the results and discussion of the proposed fraud detection mechanism and finally, Section 5 concludes the paper.

2. Motivation

In this section, some of the traditional fraud detection techniques along with their merits and demerits are explained, which motivate the researcher to design HWO-based deep RNN for detecting the fraud transactions.

2.1 Literature survey

Various existing fraud detection techniques are reviewed in this section. Zhang *et al.* (2019) developed a homogeneity-oriented behavior analysis model for detecting the frauds in the credit card system. This method effectively identified the fraudulent transaction under a false-positive rate. This method enabled to protect the interest of users and to minimize the regulatory costs and fraud losses. However, it failed to integrate the deep learning with the data mining techniques. Moreover, the computational cost of this mechanism was too high. Lucas *et al.* (2020) introduced a multi-perspective HMM for modeling the temporal correlation in transactional data. It increased the performance and classification task and effectively detected the fraudulent transactions. This method was created in a supervised manner, so it required less expert knowledge for detecting the frauds. However, it failed to integrate the prediction of long short-term memory networks with HMM to detect the frauds. Akila and Reddy (2018) introduced a risk-induced Bayesian inference bagging (RIBIB) model for detecting the frauds in credit cards. This method uses the bag creation model to handle the imbalanced data. Here, the prediction cost was reduced using the probability-based learner model. Because of the sequential workflow and probabilistic predictor, this method required less computational requirements. However, it failed to handle the concept of drift. Darwish (2020) modeled a two-level fraud tracking model based on the fusion of artificial bee colony and K-means. This method accelerated better convergence toward fraud detection and effectively increased the identification precision. It effectively detected the forged credit cards and increased the accuracy in detection. However, it failed to include extract rules in the rule engine. Moreover, it failed to validate the effectiveness of this scenario using a real data set.

Taha and Malebary (2020) developed an intelligent model named optimized light gradient boosting machine (OLightGBM) to detect the frauds in the transactions of the credit card. However, the computational complexity of this method was very low. The experimental results showed that this method outperformed the existing machine learning methods and achieved better performance in terms of precision, *F*-measure and accuracy. Yang *et al.* (2019) introduced federated learning for fraud detection based on the behavioral features using federated learning. It enabled the bank to find the frauds using the training data that is distributed in the local database. A shared fraud detection system was constructed through the aggregation of local computing updates of fraud models. However, the performance was evaluated with a large-scale data set. However, this method was failed to protect data privacy. Pumsirirat and Yan (2018) introduced a deep learning-based autoencoder model for detecting the credit card frauds. It is the unsupervised learning approach that used backpropagation and set the input equal to output. This method was focused to detect the frauds that were not detected by supervised learning. This method accurately detected the frauds using a large data set, namely, the European data set. Jiang *et al.* (2018) introduced a window-sliding strategy and feedback mechanism to detect the frauds. The behavioral patterns were extracted for individual cardholders with respect to the historical transaction of cardholders and aggregated transactions. It increased the accuracy and recall performance. However, it failed to model the periodic time window to maximize the performance of detection.

2.2 Challenges

Some of the challenges associated with the existing detection methods are discussed as follows:

- The intrinsic imbalance data present in the data transaction result in a major challenge in fraud detection. Because of the large volume of credit card transactional data, the legitimate transactions are large in number than the fraudulent transaction such that it creates bias at the training process and leads unreliable prediction (Akila and Reddy, 2018).
- The substantial amount of money and time spend by the bankers to investigate the huge number of the legitimate transaction causes potential dissatisfaction and customer inconvenience (Darwish, 2020).

- Fraud detection is a challenging issue because of the distribution of information that continually changes over time and so only very few percentages of the transactions are fraudulent (Taha and Malebary, 2020).
- It is very complex to learn the patterns of frauds, which formulate the fraud detection process a great challenge (Yang *et al.*, 2019).
- Fraud detection poses a challenging problem, as the fraudsters make some effort to show their behavior as legitimate. The legitimate records are greater than the fraudulent cases in bank transactions and such unbalanced data requires more precautions from the data analyst, which poses a great challenge in credit card fraud detection.

3. Proposed Harris water optimization-based deep recurrent neural network for fraud detection in bank transaction

Because of the unbalanced data of legitimate transaction with that of fraudulent transaction, detecting the frauds in bank transaction is very essential in communication technology. Various fraud detection methods are developed in the existing research works, but revealing the effective solution remains a great challenge. Therefore, an effective fraud detection mechanism named HWO-based deep RNN is proposed in this research to detect the frauds in the bank transaction. Initially, the input data is obtained from the database and is fed to the pre-processing module. The input data is pre-processed using the Box-Cox transformation model (Maciejewski *et al.*, 2012; Yang and Zhu, 2018). Once the data is pre-processed, the resultant pre-processed data is passed to the feature selection phase, where the wrapper model (Kohavi and John, 1997) is used to select the essential features. Finally, the fraud detection process is carried out for the selected features using the deep RNN classifier, which is trained by the proposed HWO algorithm. However, the proposed HWO algorithm is developed by integrating the Harris hawks optimization (HHO) (Heidari *et al.*, 2019) with the water wave optimization (WWO) (Zheng, 2015), respectively. Figure 1 depicts the schematic diagram of the proposed HWO-based deep RNN.

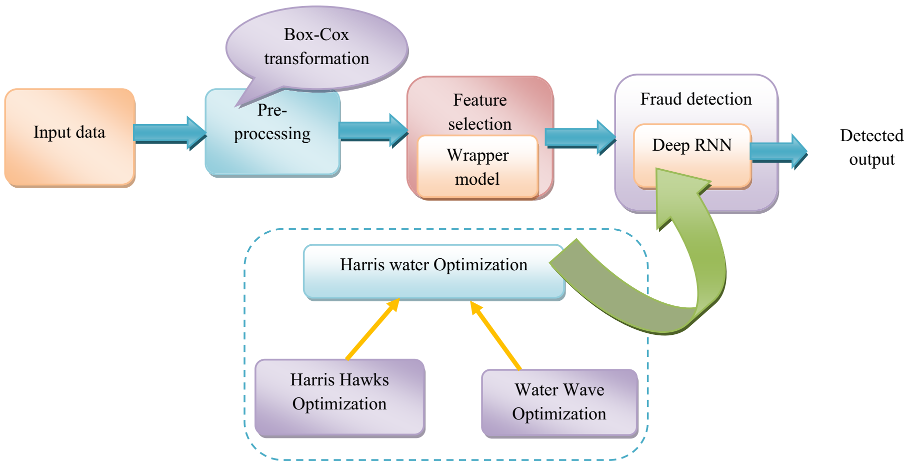


Figure 1. Schematic of the proposed HWO-based deep RNN

3.1 Get the input data

The input data used to perform the fraud detection mechanism is the band transactional data. The input transactional data is collected from the transaction database to perform the fraud detection process. Let us consider the database as K with n number of transactional data, H, is represented as follows:

$$K = \{H_1, H_2, \ldots H_l, \ldots H_n\}; 1 \leq l \leq n \tag{1}$$

where K denotes the database, H represents the transaction data and n specifies the total number of transactional records. The input data H_l is selected from the database and is passed to the pre-processing stage for further processing the detection mechanism.

3.2 Data pre-processing using Box-Cox transformation

The input data H_l selected from the database is passed to the pre-processing module, where the data is pre-processed using the Box-Cox transformation (Maciejewski *et al.*, 2012; Yang and Zhu, 2018). However, it is required to pre-process the data to transform the data into some other understandable format. The intention of the pre-processing module is to remove the duplicate values for increasing the quality of data. It belongs to the category of power transformation. However, the Box-Cox transformation model pre-processes the transactional data by converting the non-normal data into the transactional data that follow normal distribution using the optimal parameters. Owing to the conversion property, it gains more benefits in data pre-processing. Accordingly, the Box-Cox transformation is represented as follows:

$$B_l = \begin{cases} \dfrac{H_l^{(i)} - 1}{i}; & i \neq 0 \\ \ln(H_l); & i = 0 \end{cases} \tag{2}$$

where B_l represents pre-processed data, H_l is the input data and i specifies the undetermined parameter. However, this transformation assumes the value of H_l to be higher than zero $H_l > 0$, if this condition satisfies, the proper translation will occur. Moreover, the value of i is computed using the likelihood estimation strategy. The input data is transformed into normal distribution based on the value of i. The result of pre-processed data is represented as B_l, which forms the input to the feature selection phase.

3.3 Feature selection using wrapper model

The pre-processed data B_l is passed as the input to the feature selection stage, where the essential and the appropriate features are effectively selected from the pre-processed transactional data. Here, the feature selection process is carried out using the wrapper model (Kohavi and John, 1997). However, the wrapper model uses some possible parameters and performs the search operations. Accordingly, the search in the space needs initial state, termination condition, state space and a search engine. However, each state in the space specifies the feature subset. Consider u number of features and v number of bits in state space, such that each bit in the state represents the values as “1” for the indication of features and the value as “0” for the absence of features, respectively. Accordingly, the operators in space define the connection between the states such that the operators effectively add or remove the features from state space. It uses the heuristic function for finding the state space that has maximal evaluation function, and the accuracy is estimated in both the heuristic and the evaluation function.

However, the accuracy estimation performs the cross-validation with a small data set because the small data set uses less time for learning and estimating the function. Moreover, the wrapper model uses a feed-forward and backward elimination strategies. The searching process is started using an empty feature set in a forward strategy, whereas the backward strategy begins the search with a fully occupied feature set. However, the empty feature set is widely used by the state, as the empty set enable the classifier to perform quickly. Accordingly, the evaluation function uses the complexity penalty for penalizing the feature subset. Here, the penalty value is set as 0.1%, which is very small than the accuracy estimation such that the estimated accuracy is used to select the feature subset. In the feature selection phase, the dimension of the pre-processed transactional data may change with respect to the data set. However, the features that are selected using the wrapper model is represented as A with the dimension of $[U \times V]$ in such a way that the selected feature A is passed as the input to the deep RNN classifier to perform fraud detection in the bank transaction.

3.4 Fraud detection using proposed Harris water optimization-based deep recurrent neural network

The fraud detection mechanism is carried out using the deep RNN classifier, which takes the selected feature A as input and processes the features with the recurrent hidden layers. To enhance the performance of detection, it is required to train the classifier using the proposed HWO algorithm.

Solution encoding: It is the representation of the solution to be determined using the HWO algorithm, which decides the effective internal model parameters of deep RNN.

Fitness function: It is the function used to compute the solution for fraud detection using the hidden layers of the deep RNN classifier. Accordingly, the fitness with the minimal error value is declared as the optimal solution for fraud detection. However, the fitness function is expressed as follows:

$$F = \frac{1}{\eta} \sum_{\rho=1}^{\eta} D_{\rho}^{(s,z)} - \vartheta_{\rho} \tag{3}$$

where F indicates the fitness measure, $D_{\rho}^{(s,z)}$ shows the output of classifier and ϑ_{ρ} denotes the estimated output.

3.4.1 Architecture of deep recurrent neural network classifier. The feature A that is selected using the wrapper model is given as the input to the deep RNN classifier. Deep RNN (Inoue *et al.*, 2018) is the temporal network structure that consists of a number of hidden recurrent layers in the network hierarchy. However, it is exponentially more effective and efficient for representing some function than other classifiers. Here, the recurrent connection only exists between the hidden layers. It effectively performs the detection mechanism under varying feature length using the sequence of data. It uses the previous state result as the input to the next state and continues the iteration with the hidden information. Accordingly, the recurrent feature makes the classifier to generate better fraud detection result. Figure 2 portrays the architecture of the deep RNN classifier.

The structure of deep RNN is designed by considering the input vector of sth layer at zth time as $A^{(s,z)} = \{A_1^{(s,z)}, A_2^{(s,z)}, \ldots A_l^{(s,z)}, \ldots A_a^{(s,z)}\}$ and the output vector of sth layer at zth time as $D^{(s,z)} = \{D_1^{(s,z)}, D_2^{(s,z)}, \ldots D_l^{(s,z)}, \ldots D_a^{(s,z)}\}$, respectively. In this structure, the pair of

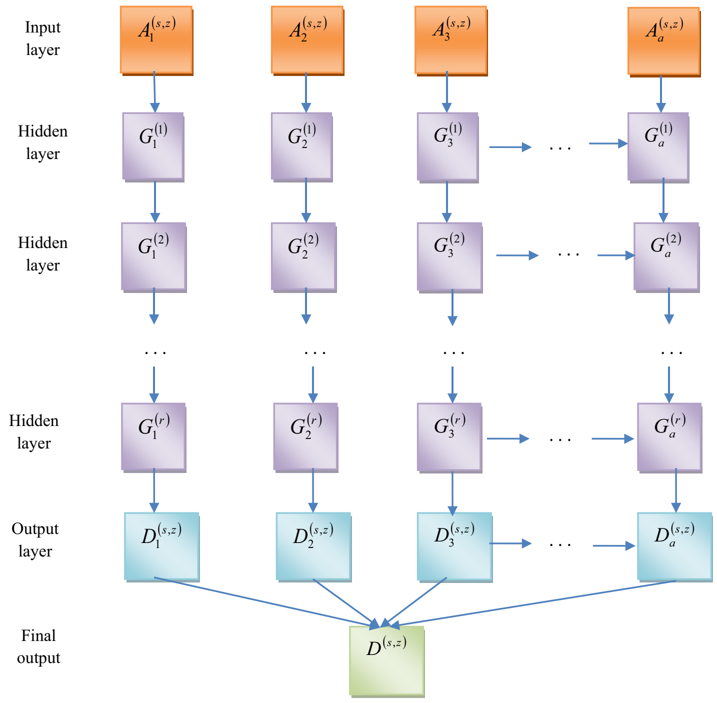


Figure 2.
Architecture of deep RNN classifier

each element in the input and output vectors is specified as a unit. Here, l specifies the arbitrary unit number of sth layer, and a indicates the total number of units of sth layer. In addition to the input and the output parameters, the arbitrary unit number of $(s-1)$th layer is specified as g, and the total number of units of $(s-1)$th layer is represented T, respectively. At this instance, input propagation weight from $(s-1)$th layer to sth layer is specified as, $\omega^{(s)} \in \lambda^{a \times T}$, and the recurrent weight of sth layer is indicated as $w^{(s)} \in \lambda^{a \times a}$. Here, λ represents the set of weights. Moreover, the components of the input vector are mathematically represented using the following equation:

$$A_l^{(s,z)} = \sum_{b=1}^{T} \alpha_{lb}^{(s)} D_b^{(s-1,z)} + \sum_{l'}^{a} \beta_{ll'}^{(s)} D_{l'}^{(s,z-1)} \tag{4}$$

where $\alpha_{lb}^{(s)}$ and $\beta_{ll'}^{(s)}$ are the elements of $\omega^{(s)}$ and $w^{(s)}$. l' indicates the arbitrary unit number of sth layer. However, the elements of the output vector of sth layer are specified as follows:

$$D_l^{(s,z)} = \delta^{(s)}\left(A_l^{(s,z)}\right) \tag{5}$$

where $\delta^{(s)}$ denotes the activation function. Moreover, the activation functions, namely, rectified linear unit function (ReLU) as $\delta\ (A) = \max(A,\tau)$, logistic sigmoid function as $\delta(A) = \frac{1}{(1+e^{-A})}$ and the sigmoid function as $\delta\ (A) = \tanh(A)$ are the commonly used activation functions. To make the detection process simpler, let us assume τth weight as $\alpha_{l\tau}^{(s)}$ and τth unit as $D_\tau^{(s-1,z)}$ and hence, the bias is denoted as follows:

$$D^{(s,z)} = \delta^{(s)}\left(\omega^{(s)} D^{(s-1,z)} + w^{(s)} A_l^{(s,z-1)}\right) \tag{6}$$

Here, $D^{(s,z)}$ denotes the output of the classifier.

3.4.2 Proposed Harris water optimization algorithm. The training procedure of the deep RNN classifier is achieved by the proposed HWO algorithm, which is designed by integrating the HHO (Heidari *et al.*, 2019) and the WWO (Zheng, 2015). The WWO is the meta-heuristic approach that shows optimal phenomena in water waves, such as refraction, breaking and propagation, which are effectively used to derive better detection performance. WWO takes the inspiration of the shallow water model and solves the optimization problem more effectively. Along with the operations behavior of waves, the exploitation and the exploration strategy of HHO are used to boost the performance of the detection mechanism. The algorithmic steps involved in the proposed HWO algorithm are explained as follows:

- *Population initialization*: Let us initialize the random population of hawks as R with J number of hawks, which is expressed as follows:

$$R = \{R_1, R_2, \ldots R_j, \ldots R_J\};\, 1 \le j \le J \tag{7}$$

where R denotes the population of hawks. Moreover, the population size of the hawks is represented as J:

- *Compute the fitness value of hawks*: The fitness value of the hawks is computed for determining the best solution such that the equation used to calculate the fitness measure is specified in equation (3).
- *Update initial energy and jump strength*: The initial energy P_0 is used to determine the state of rabbit such that initial energy is expressed as follows:

$$P_0 = 2rand_{()} - 1 \tag{8}$$

where P_0 denotes the initial energy such that the initial energy lies in the range of [−1,1]. However, the random jump strength of rabbit throughput the escaping process is represented as follows:

$$E = 2\left(1 - rand_{()}\right) \tag{9}$$

where E represents the random jump strength of the rabbit, and $rand(\,)$ denotes the random number that lies between 0 and 1, respectively.

- *Update the energy*: The energy of the rabbit is calculated using the following equation:

$$P = 2P_0\left(1 - \frac{x}{X}\right) \tag{10}$$

where P denotes the external energy from the rabbit, x denotes the present iteration and X denotes the maximum number of iterations.

- *Update location vector at exploration phase*: The Harris hawks track and detect their prey with their powerful eyes. It monitors and observes the desert site for detecting the prey. However, the position vector of the hawks is expressed as follows:

$$R(x+1) = \begin{cases} R_{rand}(x) - m_1|R_{rand}(x) - 2m_2R(x)| & ;d \geq 0.5 \\ R_{rabbit}(x) - R_c(x) - m_3(M + m_4(Q - M)) & ;\text{d} < 0.5 \end{cases} \tag{11}$$

where $R(x+1)$ denotes the position vector of hawks at next iteration x; $R_{rabbit}(x)$ denotes the location of rabbit; $R(x)$ represents the current position vector of the hawks; M denotes the lower bound; Q represents the upper bound; and m_1, m_2, m_3, m_4 and d denotes the random number that lies in the interval of [0,] such that these random numbers are updated at each iteration x, respectively. $R_{rand}(x)$ denotes the randomly selected hawk from the population, and R_c represents the average location of the hawks population, which is represented as follows:

$$R_c(x) = \frac{1}{J}\sum_{n=1}^{J} R_n(x) \tag{12}$$

Here, R_n (x) denotes the location of an individual hawk at the iteration x, and N indicates the total number of hawks, respectively.

- *Update location vector at exploitation phase of soft besiege*: The rabbit uses enough energy and try to escape from misleading jumps. In this phase, the Harris hawks encircles the prey softly and performs the surprise pounce and this behavior is modeled as follows:

$$R(x+1) = \Delta R(x) - P|E\,R_{rabbit}(x) - R(x)| \tag{13}$$

$$\Delta R(x) = R_{rabbit}(x) - R(x) \tag{14}$$

Let us assume R_{rabbit} $(x) < R(x)$

$$R(x+1) = R_{rabbit}(x) - R(x) - PE\,R_{rabbit}(x) + PR(x) \tag{15}$$

$$R(x+1) = R(x)[P-1] + R_{rabbit}(x)[1 - PE] \tag{16}$$

where $\Delta R(x)$ represents the difference between the present location and the position vector of rabbit iteration x, respectively.

However, the new waves created by the propagation operator of the WWO algorithm is expressed as follows:

$$R(x+1) = R(x) + rand(-1,1).W.Y \tag{17}$$

$$R(x) = R(x+1) - rand(-1,1).W.Y \tag{18}$$

By substituting equation (18) in equation (16), the resultant equation is expressed as follows:

$$R(x+1) = \big(R(x+1) - rand(-1,1).W.Y\big)[P-1] + R_{rabbit}(x)[1 - PE] \tag{19}$$

$$R(x+1) = \big(R(x+1)P - R(x+1) - rand(-1,1).W.Y(P-1) + R_{rabbit}(x)[1-PE] \tag{20}$$

$$R(x+1) - \big(R(x+1)P + R(x+1) = -rand(-1,1).W.Y(P-1) + R_{rabbit}(x)[1-PE] \tag{21}$$

$$R(x+1)(2-P) = -rand(-1,1).W.Y(P-1) + R_{rabbit}(x)[1-PE] \tag{22}$$

$$R(x+1) = \frac{R_{rabbit}(x)(1-PE) - rand(-1,1).W.Y(P-1)}{(2-P)} \tag{23}$$

where $R(x+1)$ denotes the position vector of hawks at next iteration x, $R_{rabbit}(x)$ denotes the location of the rabbit, P denotes the external energy from rabbit, E represents the random jump strength of rabbit, Y is the length of search space and W represents the wavelength of the wave, respectively:

- *Update location vector at exploitation phase of hard besiege*: In this phase, the prey is exhausted such that it has less escaping energy. However, the position vector equation of this phase is expressed as follows:

$$R(x+1) = R_{rabbit}(x) - P|\Delta R(x)| \tag{24}$$

- *Update location vector using progressive rapid lives*: The final strategy of updating the location of hawks at the soft besiege phase is represented as follows:

$$R(x+1) = \begin{cases} \chi; F(\chi) < F(R(x)) \\ \xi; F(\xi) < F(R(x)) \end{cases} \tag{25}$$

Here the term χ and ξ are expressed as follows:

$$\chi = R_{rabbit}(x) - P|E\,R_{rabbit}(x) - R(x)| \tag{26}$$

$$\xi = \chi + N \times \lambda(I) \tag{27}$$

Accordingly, the final strategy of updating the location of hawks at the hard besiege phase is expressed as follows:

$$R(x+1) = \begin{cases} S; F(S) < F(R(x)) \\ C; F(C) < F(R(x)) \end{cases} \tag{28}$$

However, the term S and C are expressed as follows:

$$S = R_{rabbit}(x) - P|E\,R_{rabbit}(x) - R_c(x)| \tag{29}$$

$$C = S + N \times \lambda(I) \tag{30}$$

where N denotes the random vector with the dimension of $[1 \times I]$, I denotes the dimension of the problem and λ is the levy flight function. Algorithm 1 represents the pseudo-code of the proposed HWO algorithm.

Algorithm 1. Pseudo-code of the proposed HWO algorithm

Sl. NoPseudo-code of the proposed HWO Algorithm

```
1 Input: Population size $J$, and maximum iterations $X$
2 Output: $R(x+1)$
3 Population initialization
4 while (stopping criteria is not met)
5 do
6   Compute $F$
7   Set $R_{rabbit}$ as the best location of rabbit
8 for (each hawk ($R_j$))
9 do
10    Update $P_0$ and $E$
11    Update $P$ using equation (10)
12 if ($|P| \geq 1$) then
13          Update $R(x+1)$ at exploration phase using equation (11)
14 if ($|P| < 1$) then
15       if ($m \geq 0.5 \& |P| \geq 0.5$) then
16            Update $R(x+1)$ at exploitation phase of soft besiege
              using equation (23)
17       else if ($m \geq 0.5 \& |P| < 0.5$) then
18      Update $R(x+1)$ at exploitation phase of hard besiege using
        equation (24)
19       else if ($m < 0.5 \& |P| \geq 0.5$) then
20 Update $R(x+1)$ using progressive rapid dives of soft besiege
   using equation (25)
21       else if ($m < 0.5 \& |P| < 0.5$) then
22   Update $R(x+1)$ using progressive rapid dives of hard besiege
     using equation (28)
23Return $R_{rabbit}$
```

By integrating the parametric features from both the optimization algorithm, the performance of the proposed method revealed more effectiveness in detecting the frauds in bank transactions. However, the dynamic behaviors and the patterns are integrated together mathematically to solve the optimization problem more efficiently.

4. Results and discussion

This section elaborates the results and discussion made using the proposed HWO-based deep RNN for fraud detection.

4.1 Experimental setup

The experimentation of the proposed approach is carried out in the PYTHON tool using the synthetic data set (Synthetic data from a financial payment system, 2020) that is formed using the payment simulator of BankSim. This data set contains transactional data that is offered by the bank from Spain. This data set is mainly used for the purpose of fraud detection. It contains the synthetic data that is mainly intended for the purpose of fraud detection in the research area. Here, the relationship between customers and merchants are used for social and statistical network analysis. The BankSim is developed with 180 steps and takes several times to calibrate the parameters to get the distribution that is reliable for testing. Several log files are collected but selected only the more accurate files. The thieves are entered to steal the average of three cards to achieve the fraudulent transaction. Here, 594,643 records are totally produced, where 587,443 are the normal payments and the remaining 7,200 are the fraudulent transaction. As it is the randomized simulation, the values obtained by this simulator are the course data and is not relevant to the original data.

4.2 Evaluation metrics

The performance of the proposed HWO-based deep RNN is evaluated using the metrics, namely, accuracy, specificity and sensitivity, respectively.

Accuracy: It is the measure that estimates the proportion of true positive and true negative values in all the cases, which is represented as follows:

$$A = \frac{L_p + L_q}{L_p + L_q + O_p + O_q} \tag{31}$$

where L_p denotes the true positive, L_q represents the true negative, O_p indicates the false positive, Q_q specifies the false negative and A is the accuracy, respectively.

Sensitivity: It is the measure used to detect the fraudulent activities correctly, which is represented as follows:

$$B = \frac{L_p}{L_p + O_q} \tag{31}$$

Here, B denotes sensitivity.

Specificity: It is the measure that detects the legitimate activities correctly, which is expressed as follows:

$$X = \frac{L_q}{L_q + O_p} \tag{32}$$

where X represents specificity.

4.3 Performance analysis

This section elaborates the performance analysis of the proposed HWO-based deep RNN by varying the number of layers and the number of neurons.

4.3.1 Performance analysis by varying number of neurons. Figure 3 represents the performance analysis of the proposed HWO-based deep RNN by varying the number of neurons. Figure 3(a) depicts the performance analysis of accuracy by varying the number of neurons. For 70% training data, the accuracy of the proposed HWO-based deep RNN with 50 neurons, HWO-based deep RNN with 100 neurons, HWO-based deep RNN with 150 neurons and HWO-based deep RNN with 200 neurons is 0.8827, 0.8881, 0.8991 and 0.9130, respectively. For 90% training data, the accuracy of the proposed HWO-based deep RNN with 50 neurons, HWO-based deep RNN with 100 neurons, HWO-based deep RNN with 150 neurons and HWO-based deep RNN with 200 neurons is 0.9176, 0.9375, 0.9382 and 0.9396, respectively.

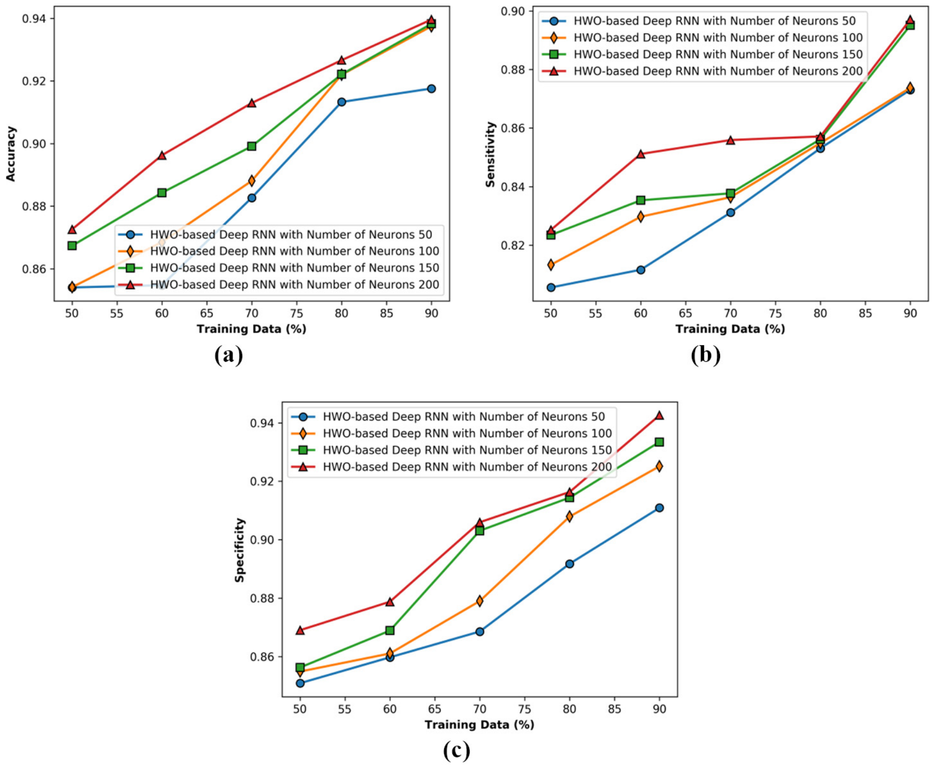


Figure 3.
Performance analysis of HWO-based deep RNN by a varying number of neurons

Notes: (a) Accuracy; (b) sensitivity; (c) specificity

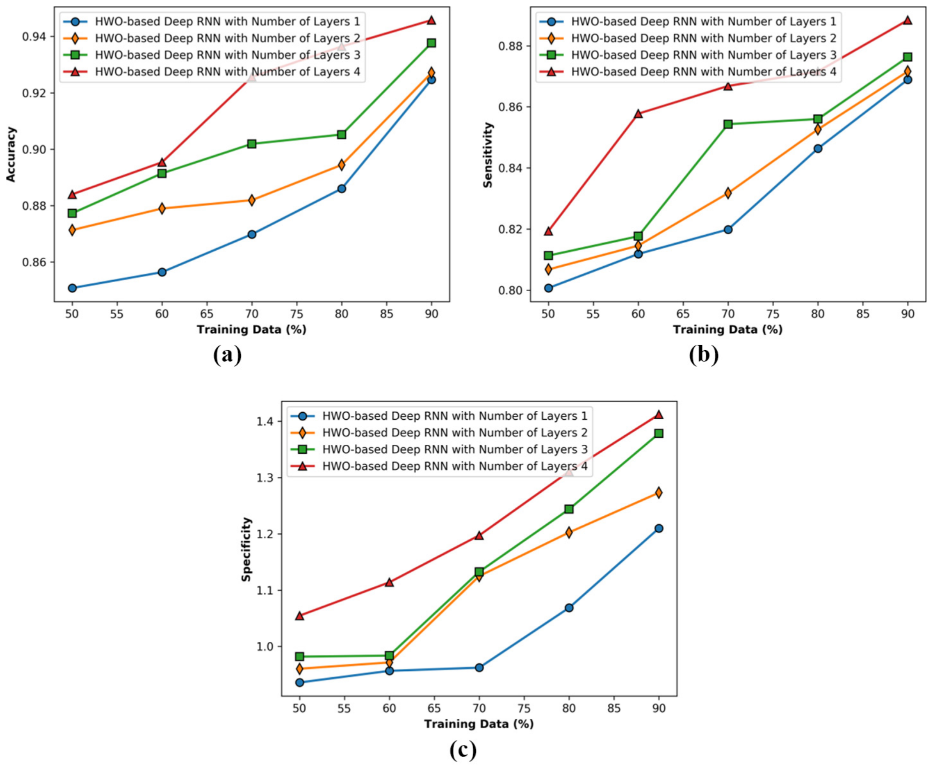
Figure 3(b) depicts the performance analysis of sensitivity by varying the number of neurons. For 70% training data, the sensitivity of the proposed HWO-based deep RNN with 50 neurons, HWO-based deep RNN with 100 neurons, HWO-based deep RNN with 150 neurons and HWO-based deep RNN with 200 neurons is 0.8311, 0.8364, 0.8377 and 0.8559, respectively. For 90% training data, the sensitivity of the proposed HWO-based deep RNN with 50 neurons, HWO-based deep RNN with 100 neurons, HWO-based deep RNN with 150 neurons and HWO-based deep RNN with 200 neurons is 0.8730, 0.8737, 0.8951 and 0.8971, respectively.

Figure 3(c) depicts the performance analysis of specificity by varying the number of neurons. For 70% training data, the specificity of the proposed HWO-based deep RNN with 50 neurons, HWO-based deep RNN with 100 neurons, HWO-based deep RNN with 150 neurons and HWO-based deep RNN with 200 neurons is 0.8686, 0.8790, 0.9031 and 0.9060, respectively. For 90% training data, the specificity of the proposed HWO-based deep RNN with 50 neurons, HWO-based deep RNN with 100 neurons, HWO-based deep RNN with 150 neurons and HWO-based deep RNN with 200 neurons is 0.9109, 0.9251, 0.9335 and 0.9426, respectively.

4.3.2 Performance analysis by varying number of layers. Figure 4 represents the performance analysis of the proposed HWO-based deep RNN by varying the number of layers. Figure 4(a) depicts the performance analysis of accuracy by varying the number of layers. For 70% training data, the accuracy of the proposed HWO-based deep RNN with one layer, HWO-based deep RNN with two layers, HWO-based deep RNN with three layers and HWO-based deep RNN with four layers is 0.8698, 0.8819, 0.9019 and 0.9256. For 90% training data, the accuracy of the proposed HWO-based deep RNN with one layer, HWO-based deep RNN with two layers, HWO-based deep RNN with three layers and HWO-based deep RNN with four layers is 0.9247, 0.9271, 0.9377 and 0.9458, respectively.

Figure 4(b) depicts the performance analysis of sensitivity by varying the number of layers. For 70% training data, the sensitivity of the proposed HWO-based deep RNN with one layer, HWO-based deep RNN with two layers, HWO-based deep RNN with three layers and HWO-based deep RNN with four layers is 0.8199, 0.8318, 0.8543 and 0.8668. For 90% training data, the sensitivity of the proposed HWO-based deep RNN with one layer, HWO-based deep RNN with two layers, HWO-based deep RNN with three layers and HWO-based deep RNN with four layers is 0.8689, 0.8716, 0.8763 and 0.8884, respectively.

Figure 4(c) depicts the performance analysis of specificity by varying the number of layers. For 70% training data, the specificity of the proposed HWO-based deep RNN with one layer, HWO-based deep RNN with two layers, HWO-based deep RNN with three layers and HWO-based deep RNN with four layers is 0.9623, 1.1248, 1.1325 and 1.1971. For 90% training data, the specificity of the proposed HWO-based deep RNN with one layer, HWO-



Notes: (a) Accuracy; (b) sensitivity; (c) specificity

Figure 4. Performance analysis of HWO-based deep RNN by varying the number of layers

based deep RNN with two layers, HWO-based deep RNN with three layers and HWO-based deep RNN with four layers is 1.2097, 1.2728, 1.3781 and 1.4115, respectively.

4.4 Comparative methods

The performance revealed by the proposed approach is evaluated by comparing the proposed with the existing approaches, such as gradient boosting decision tree (GBDT) (Zhou *et al.*, 2018), Majority voting+ Adaboost (Randhawa *et al.*, 2018), Random forest (Carneiro *et al.*, 2017), RIBIB (Akila and Reddy, 2018) and OLightGBM (Taha and Malebary, 2020), respectively.

4.5 Comparative analysis

Figure 5 depicts the comparative of the proposed HWO-based deep RNN in terms of sensitivity, specificity and accuracy with respect to training data. Figure 5(a) portrays the analysis of the proposed HWO-based deep RNN in terms of accuracy by varying training data. For 60% training data, the accuracy of the existing GBDT is 0.7865, majority voting + Adaboost is 0.8065, random forest is 0.8299, Harris Grey Wolf (HGW)-based deep stacked autoencoder is 0.8575, RIBIB is 0.8135 and OLightGBM is 0.8212, whereas the proposed HWO-based deep RNN obtained better accuracy of 0.8792, respectively. For 70% training data, the accuracy of the existing GBDT is 0.8474, majority voting + Adaboost is 0.8346, random forest is 0.8651, HGW-based deep stacked autoencoder is 0.8826, RIBIB is 0.8476 and OLightGBM is 0.8565, whereas the proposed HWO-based deep RNN obtained better accuracy of 0.8922, respectively. For 90% training data, the accuracy of the existing GBDT is 0.8872, majority voting + Adaboost is 0.8772, random forest is 0.8828, HGW-based deep stacked autoencoder is 0.8962, RIBIB is 0.8867 and OLightGBM is 0.9009, whereas the proposed HWO-based deep RNN obtained better accuracy of 0.9192, respectively.

Figure 5(b) portrays the analysis of the proposed HWO-based deep RNN in terms of sensitivity by varying training data. For 60% training data, the sensitivity of the existing GBDT is 0.6944, majority voting + Adaboost is 0.6545, random forest is 0.6513, HGW-based deep stacked autoencoder is 0.7102, RIBIB is 0.6518 and OLightGBM is 0.6939, whereas the proposed HWO-based deep RNN obtained better sensitivity of 0.7320. For 70% training data, the sensitivity of the existing GBDT is 0.6989, majority voting + Adaboost is 0.6827, random forest is 0.6908, HGW-based deep stacked autoencoder is 0.7253, RIBIB is 0.6869 and OLightGBM is 0.7091, whereas the proposed HWO-based deep RNN obtained better sensitivity of 0.7402. For 80% training data, the sensitivity of the existing GBDT is 0.7096, majority voting + Adaboost is 0.6966, random forest is 0.7122, HGW-based deep stacked autoencoder is 0.7294, RIBIB is 0.7038 and OLightGBM is 0.7268, while the proposed HWO-based deep RNN obtained better sensitivity of 0.7536. For 90% training data, the sensitivity of the existing GBDT is 0.7160, majority voting + Adaboost is 0.7349, random forest is 0.7375, HGW-based deep stacked autoencoder is 0.7524, RIBIB is 0.7183 and OLightGBM is 0.7277, whereas the proposed HWO-based deep RNN obtained better sensitivity of 0.7642.

Figure 5(c) portrays the analysis of the proposed HWO-based deep RNN in terms of specificity by varying training data. For 60% training data, the specificity of the existing GBDT is 0.9101, majority voting + Adaboost is 0.9032, random forest is 0.9557, HGW-based deep stacked autoencoder is 0.9505, RIBIB is 0.9480 and OLightGBM is 0.9521, whereas the proposed HWO-based deep RNN obtained better specificity of 0.9631. For 70% training data, the specificity of the existing GBDT is 0.9359, majority voting + Adaboost is 0.9390, random forest is 0.9567, HGW-based deep stacked autoencoder is 0.9521, RIBIB is 0.9418 and OLightGBM is 0.9435, whereas the proposed HWO-based deep RNN obtained better specificity of 0.9691, respectively. For 90% training data, the specificity of the existing

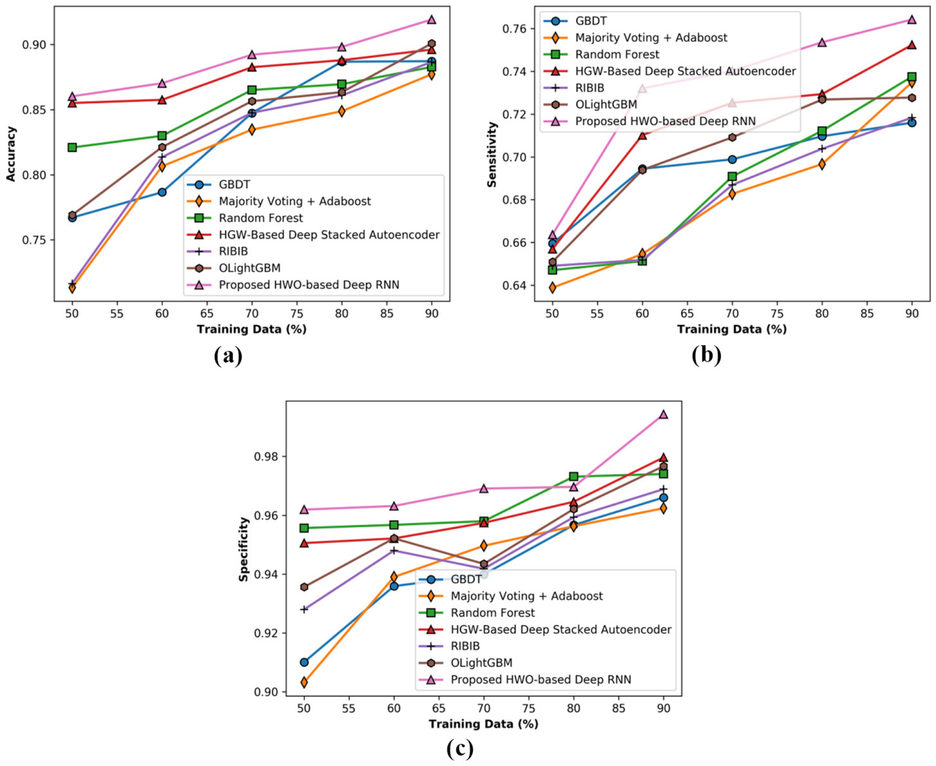


Figure 5. Comparative analysis of HWO-based deep RNN

Notes: (a) Accuracy; (b) sensitivity; (c) specificity

GBDT is 0.9660, majority voting + Adaboost is 0.9624, random forest is 0.9740, HGW-based deep stacked autoencoder is 0.9796, RIBIB is 0.9689 and OLightGBM is 0.9767, whereas the proposed HWO-based deep RNN obtained better specificity of 0.9943, respectively.

4.6 Comparative discussion

Table 1 represents the comparative discussion. The best performance is attained at 90% of the training data. The accuracy of the existing GBDT is 0.8872, majority voting + Adaboost is 0.8772, random forest is 0.8828, HGW-based deep stacked autoencoder is 0.8962, RIBIB is 0.8867 and OLightGBM is 0.9009, whereas the proposed HWO-based deep RNN obtained better accuracy of 0.9192. For 90% training data, the specificity of the existing GBDT is

Metrics/ methods	GBDT	Majority voting+Adaboost	Random forest	HGW-based Deep stacked autoencoder	RIBIB	OLightGBM	Proposed HWO-based deep RNN
Accuracy	0.8872	0.8772	0.8828	0.8962	0.8867	0.9009	*0.9192*
Sensitivity	0.7160	0.7349	0.7375	0.7524	0.7183	0.7277	*0.7642*
Specificity	0.9660	0.9624	0.9740	0.9796	0.9689	0.9767	*0.9943*

Table 1. Comparative discussion

0.9660, majority voting + Adaboost is 0.9624, random forest is 0.9740, HGW-based deep stacked autoencoder is 0.9796, RIBIB is 0.9689 and OLightGBM is 0.9767, whereas the proposed HWO-based deep RNN obtained better specificity of 0.9943. For 90% training data, the sensitivity of the existing GBDT is 0.7160, majority voting + Adaboost is 0.7349, random forest is 0.7375, HGW-based deep stacked autoencoder is 0.7524, RIBIB is 0.7183 and OLightGBM is 0.7277, whereas the proposed HWO-based deep RNN obtained better sensitivity of 0.7642. It is clearly depicted that the proposed HWO-based deep RNN obtained better performance for the metrics, such as accuracy, sensitivity and specificity is 0.9192, 0.7642 and 0.9943, respectively.

5. Conclusion

In this research, an efficient and effective fraud detection approach named HWO-based deep RNN is proposed for detecting the frauds in the sequence of transactions. The proposed approach performs the detection process by involving three phases, namely, pre-processing, feature selection and a fraud-detection phase. The input data used to perform the detection process is collected from the transactional data set. Initially, the input data is passed to the pre-processing phase, where the redundant values are removed from the data using Box-Cox transformation. The pre-processed data is fed to the feature selection phase to select suitable and appropriate features using the wrapper model. Finally, the selected features are allowed to the detection phase, where the deep RNN classifier is used to detect the fraudulent transaction. For accurate fraudulent transaction detection results, the deep RNN is trained by using the proposed HWO algorithm, which is the integration of WWO and HHO. The performance obtained by the proposed HWO-based deep RNN is evaluated using the metrics, such as accuracy, sensitivity and specificity with the values of 0.9192, 0.7642 and 0.9943. The experimental results showed that the proposed approach is superior to the existing classifiers. The limitation of the proposed method is the less labelled samples. In the future, this problem will be overcome by adding a semi-supervised strategy in the proposed system. Also, the hybrid classifier will be used for increasing the performance of fraud detection .

References

Abdallah, A., Maarof, M.A. and Zainal, A. (2016), "Fraud detection system: a survey", *Journal of Network and Computer Applications*, Vol. 68, pp. 90-113.

Abdel-Hamid, O., Mohamed, A.R., Jiang, H., Deng, L., Penn, G. and Yu, D. (2014), "Convolutional neural networks for speech recognition", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Vol. 22 No. 10, pp. 1533-1545.

Akila, S. and Reddy, U.S. (2018), "Cost-sensitive risk induced Bayesian inference bagging (RIBIB) for credit card fraud detection", *Journal of Computational Science*, Vol. 27, pp. 247-254.

Bolton, R.J. and Hand, D.J. (2002), "Statistical fraud detection: a review", *Statistical Science*, Vol. 17 No. 3, pp. 235-249.

Carcillo, F., Dal Pozzolo, A., Le Borgne, Y.A., Caelen, O., Mazzer, Y. and Bontempi, G. (2018), "Scarff: a scalable framework for streaming credit card fraud detection with spark", *Information Fusion*, Vol. 41, pp. 182-194.

Carneiro, N., Figueira, G. and Costa, M. (2017), "A data mining based system for credit-card fraud detection in e-tail", *Decision Support Systems*, Vol. 95, pp. 91-101.

Chithra, R.S. and Jagatheeswari, P. (2019), "Enhanced WOA and modular neural network for severity analysis of tuberculosis", Vol. 2 No. 3.

Darwish, S.M. (2020), "A bio-inspired credit card fraud detection model based on user behavior analysis suitable for business management in electronic banking", *Journal of Ambient Intelligence and Humanized Computing*, pp. 1-15.

de Sá, A.G., Pereira, A.C. and Pappa, G.L. (2018), "A customized classification algorithm for credit card fraud detection", *Engineering Applications of Artificial Intelligence*, Vol. 72, pp. 21-29.

Eweoya, I.O., Adebiyi, A.A., Azeta, A.A. and Azeta, A.E. (2019), "Fraud prediction in bank loan administration using decision tree", in *Journal of Physics: Conference Series*, Vol. 1299 No. 1, p. 012037.

Fiore, U., De Santis, A., Perla, F., Zanetti, P. and Palmieri, F. (2019), "Using generative adversarial networks for improving classification effectiveness in credit card fraud detection", *Information Sciences*, Vol. 479, pp. 448-455.

Hayat, M., Bennamoun, M. and An, S. (2014), "Learning non-linear reconstruction models for image set classification", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1907-1914.

Heidari, A.A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M. and Chen, H. (2019), "Harris hawks optimization: algorithm and applications", *Future Generation Computer Systems*, Vol. 97, pp. 849-872.

Inoue, M., Inoue, S. and Nishida, T. (2018), "Deep recurrent neural network for mobile human activity recognition with high throughput", *Artificial Life and Robotics*, Vol. 23 No. 2, pp. 173-185.

Jiang, C., Song, J., Liu, G., Zheng, L. and Luan, W. (2018), "Credit card fraud detection: a novel approach using aggregation strategy and feedback mechanism", *IEEE Internet of Things Journal*, Vol. 5 No. 5, pp. 3637-3647.

John, H. and Naaz, S. (2019), "Credit card fraud detection using local outlier factor and isolation Forest".

Kohavi, R. and John, G.H. (1997), "Wrappers for feature subset selection", *Artificial Intelligence*, Vol. 97 Nos 1/2, pp. 273-324.

Lebichot, B., Le Borgne, Y.A., He-Guelton, L., Oblé, F. and Bontempi, G. (2019), "Deep-learning domain adaptation techniques for credit cards fraud detection", In *INNS Big Data and Deep Learning Conference*, Springer, Cham, pp. 78-88.

Lucas, Y., Portier, P.E., Laporte, L., Calabretto, S., Caelen, O., He-Guelton, L. and Granitzer, M. (2019), "Multiple perspectives HMM-based feature engineering for credit card fraud detection", *In Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pp. 1359-1361, April.

Lucas, Y., Portier, P.E., Laporte, L., He-Guelton, L., Caelen, O., Granitzer, M. and Calabretto, S. (2020), "Towards automated feature engineering for credit card fraud detection using multi-perspective HMMs", *Future Generation Computer Systems*, Vol. 102, pp. 393-402.

Maciejewski, R., Pattath, A., Ko, S., Hafen, R., Cleveland, W.S. and Ebert, D.S. (2012), "Automated box-cox transformations for improved visual encoding", *IEEE Transactions on Visualization and Computer Graphics*, Vol. 19 No. 1, pp. 130-140.

Marsaline Beno, M., Valarmathi, I.R., Swamy, S.M. and Rajakumar, B.R. (2014), "Threshold prediction for segmenting tumour from brain MRI scans", *International Journal of Imaging Systems and Technology*, Vol. 24 No. 2, pp. 129-137.

Menaga, D. and Revathi, S. (2020), "Deep learning: a recent computing platform for multimedia information retrieval", *Deep Learning Techniques and Optimization Strategies in Big Data Analytics*, pp. 124-141.

More Umesh Katkar, N.S., Agrawal, G., Gund, R. and Lad, M. (2018), "Online secure payment system using stegnography and cryptography", *International Engineering Research Journal*.

Nami, S. and Shajari, M. (2018), "Cost-sensitive payment card fraud detection based on dynamic random Forest and k-nearest neighbors", *Expert Systems with Applications*, Vol. 110, pp. 381-392.

Pumsirirat, A. and Yan, L. (2018), "Credit card fraud detection using deep learning based on auto-encoder and restricted Boltzmann machine", *International Journal of Advanced Computer Science and Applications*, Vol. 9 No. 1, pp. 18-25.

Randhawa, K., Loo, C.K., Seera, M., Lim, C.P. and Nandi, A.K. (2018), "Credit card fraud detection using AdaBoost and majority voting", *IEEE Access*, Vol. 6, pp. 14277-14284.

Seyedhossein, L. and Hashemi, M.R. (2010), "Mining information from credit card time series for timelier fraud detection", *IEEE 5th International Symposium on Telecommunications*, pp. 619-624.

Singh, A. and Jain, A. (2019), "Adaptive credit card fraud detection techniques based on feature selection method", *in Advances in Computer Communication and Computational Sciences*, Springer, Singapore, pp. 167-178.

Singh, M., Kumar, S. and Garg, T. (2019), "Credit card fraud detection using hidden markov model", *International Journal of Engineering and Computer Science*, Vol. 8 No. 11, pp. 24878-24882.

Synthetic data from a financial payment system (2020), www.kaggle.com/ntnu-testimon/banksim1/, accessed on April 2020.

Taha, A.A. and Malebary, S.J. (2020), "An intelligent approach to credit card fraud detection using an optimized light gradient boosting machine", *IEEE Access*, Vol. 8, pp. 25579-25587.

Yang, Y. and Zhu, H. (2018), "A study of Non-Normal process capability analysis based on Box-Cox transformation", *IEEE 3rd International Conference on Computational Intelligence and Applications (ICCIA)*, pp. 240-243.

Yang, W., Zhang, Y., Ye, K., Li, L. and Xu, C.Z. (2019), "FFD: a federated learning based method for credit card fraud detection", *in International Conference on Big Data*, Springer, Cham, pp. 18-32.

Zhang, X., Han, Y., Xu, W. and Wang, Q. (2019), "HOBA: a novel feature engineering methodology for credit card fraud detection with a deep learning architecture", *Information Sciences*.

Zhang, F., Liu, G., Li, Z., Yan, C. and Jiang, C. (2019), "GMM-based under sampling and its application for credit card fraud detection", *IEEE International Joint Conference on Neural Networks (IJCNN)*, pp. 1-8.

Zheng, Y.J. (2015), "Water wave optimization: a new nature-inspired metaheuristic", *Computers and Operations Research*, Vol. 55, pp. 1-11.

Zhou, H., Chai, H. and Qiu, M. (2018), "Fraud detection within bankcard enrollment on mobile device based payment using machine learning", *Frontiers of Information Technology and Electronic Engineering*, Vol. 19 No. 12, pp. 1537-1545.

Corresponding author
Chandra Sekhar Kolli can be contacted at: usercsk@gmail.com