IDENTIFYING A DEFAULT OF CREDIT CARD CLIENTS BY USING A LSTM METHOD: A CASE STUDY

Jui-Yu Wu and Pei-Ci, Liu

Department of Business Administration,
Lunghwa University of Science and Technology, Taiwan

ABSTRACT

Detecting fraudulent transactions is critical and challenging for financial banks and institutes. This study used a deep learning technique, which is a long short-term memory (LSTM) method, for identifying a default of credit card clients (an imbalanced dataset). To evaluate the performance of optimizers for the LSTM approach, this study employed three optimizers based on gradient methods, such as adaptive moment estimation (Adam), stochastic gradient descent with momentum (Sgdm) and root mean square propagation (Rmsprop). This study used 10-fold cross-validation. Moreover, this study compared the best numerical results of the LSTM method with those of supervised machine learning classifiers, which are back-propagation neural network (BPNN) with a gradient descent algorithm (GDA) and a scaled conjugate gradient algorithm (SCGA). Numerical results indicate that the LSTM-Adam and the BPNN-SCGA classifiers have identical performance, and that selecting an appropriate classification threshold value is important for an imbalanced dataset. Based on the numerical results, the LSTM-Adam classifier can be considered for dealing with credit scoring problems, which are binary classification problems.

KEYWORDS

Deep Learning, Machine Learning, Long Short-Term Memory, Back-Propagation Neural Network, Credit Scoring .

1. INTRODUCTION

For financial banks and institutes, credit card default prediction, credit approval and bankruptcy prediction are significant tasks and challenges. The frauds of credit card can be divided into many activities, such as lost card, card holder not present and counterfeit card [1]. Hence, these tasks of monitoring credit card data and transactions are essential. These issues relate to credit risk management. In credit risk management, many methods have been used, such as judgmental methods, expert systems (e.g. lending committees), statistical models (e.g. credit scoring) and behavioural models [2]. These tasks can be considered as binary classification problems. For solving these classification problems, supervised machine learning (ML) approaches can be considered, such as back-propagation neural networks (BPNN), support vector machines (SVMs), K Nearest Neighbor (KNN) algorithms and random forests. These supervised ML methods have been applied in many fields. For instance, Sehgal [3] used a BPNN classifier for human activity recognition. Lawi and Aziz [4] employed a Least Square SVM (LS-SVM) ensemble classifier for classification of credit card default clients, and indicating that the performance of LS-SVM ensemble classifier is superior to that of a SVM classifier. Vaishnnave et. al., [5] applied a KNN classifier for detection and classification of groundnut leaf diseases.

David C. Wyld et al. (Eds): CONEDU, CSITA, MLCL, ISPR, NATAP, ARIN - 2022
pp. 157-166, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.121012

Deep learning (DL) is a subfield of ML and is the multi-layer NNs (more than three layers) that can perform ML algorithms. DL algorithms can learn and extract features from data representation. Many supervised and unsupervised deep learning networks have been developed, such as deep multi-layer perceptrons, convolutional NNs (CNNs), recurrent NNs (RNNs), autoencoder and restricted Boltzmann machine [6]. The CNNs have the capabilities that deal with signals of multi-dimensional arrays and efficiently process imaging problems. The RNNs that contain the information at the previous timesteps are specifically presented to handle sequential signals to capture temporal features. Moreover, the RNNs update the weights of network topology by using an error back-propagation (EBP) algorithm, which causes the limitations of gradient vanishing and exploding. To overcome these drawbacks, a long short-term memory (LSTM) approach with advanced RNN cells has been developed. The LSTM methods have been applied to prediction and classification problems [7, 8].

BPNNs are well-known NNs and have been widely applied to various fields. A conventional BPNN updates the weights of a network topology by using an EBP method, which is a gradient descent algorithm (GDA). The GDA has the limitation that is easily to trap into local optima. To overcome this drawback, a scaled conjugate gradient algorithm (SCGA) has been developed by Moller [9]. Therefore, the SCGA is employed in this study.

To evaluate the performance the LSTM classifier that is a deep learning scheme for binary classification problems, this study used the LSTM approach with three three optimizers (training algorithms), such as adaptive moment estimation (Adam), stochastic gradient descent with momentum (Sgdm) and root mean square propagation (Rmsprop) algorithms to identify a default of credit card clients. The dataset was taken from UCI machine lerning repository [10, 11] and is an imbalanced dataset. For the imbalanced dataset, this study employed different *CT* (classification threshold) values, such as 0.5 and 0.3. Furthermore, this study compared the best numerical results obtained by using the LSTM method and with those yielded from BPNN-GDA and BPNN-SCGA classifier.

The rest of this study is organized as follows. Section 2 describes the concept of ML and deep DL, a LSTM method, BPNN scheme and performance evaluation factors of a classifier. Section 3 then introduces the implementation of the LSTM and the BPNN classifier. Next, Section 4 compares the numerical results. Conclusions are finally drawn in Section 5.

2. RELATED WORK

2.1. Machine Learning and Deep Learning

ML algorithms, which consist of supervised, unsupervised and reinforcement learning, are that computers can simulate a human learning, identify and acquire knowledge from the real world and can enhance its performance based on the obtained knowledge on some tasks [12].

1. ML approaches with a supervised learning algorithm can be used to solve forecasting (times series and regression) and classification tasks.
2. ML methods with an unsupervised learning algorithm can be applied to deal with clustering problems.
3. ML systems with a reinforcement learning algorithm that learns the optimal behavior in an environment to yield a maximum reward. The conception of reinforcement learning algorithm is composed of agent, action, discount factor, environment, state, reward, penalty and policy [13].

The difference between the ML and the DL methods are that the DL approaches can extract high-level features from a huge amount of raw data by using a general-purpose learning procedure [6, 14].

2.2. LSTM Method

To improve the limitations of gradient vanishing and exploding of an RNN, a specific cell is introduced into a network topology of a LSTM method. The cell executes a mission of decision making by considering the values of previous memory cell, current input and previous output. The information of the memory cell is then updated by creating a new output value. The network topology of a LSTM approach is shown in Figure 1.

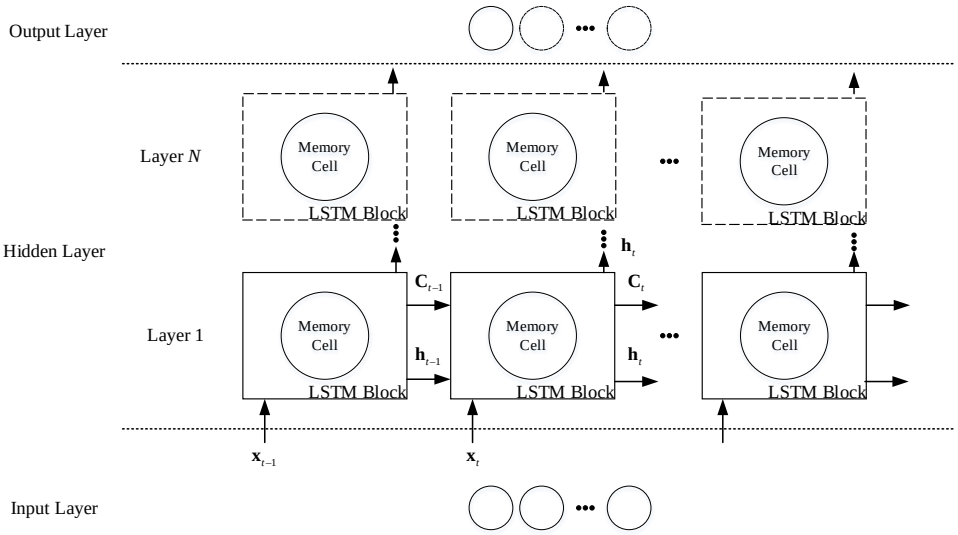

Figure 1. A network topology of a LSTM method

The implementation of a forget gate, an input gate, an output gate and update of a memory cell can be found in the literature [8, 15]. Many optimizers have been used, such as the Adam, the Sgdm and the Rmsprop methods. Chang et. al., [16] presented that a LSTM-Adam works well and is superior to existed optimizers for electricity price forecasting. Hence, this study compared the performance of the LSTM-Adam, the LSTM-Sgdm and the LSTM-Rmsprop classifiers for solving a credit scoring problem.

2.3. BPNN Scheme

A BPNN is a multi-layer NN, which consists of an input layer, some hidden layers and an output layer. Activation functions in input-hidden and hidden-output layers are responsible for biasing the neurons. Moreover, parameter settings for a BPNN include the number of hidden layers, number of hidden neurons, learning rate and momentum term. To evaluate the performance of optimizers based on gradient methods, this study used the BPNN classifiers with the GDA and the SCGA to identify a default of credit card clients and compared the numerical results with those obtained using the LSTM-Adam, the LSTM-Sgdm and the LSTM-Rmsprop classifiers.

2.4. Performance Evaluation Factors of a Classifier

For a binary classification problem, a confusion matrix for visualizing the classification performance can be defined, as shown in Table 1. The confusion matrix is composed of *TP* (true positives), *FN* (false negatives), *FP* (false negatives) and *TN* (true negatives). To evaluate the performance of classification models, this study used four factors, such as *Acc* (accuracy), *Pre* (precision), *Rec* (recall) and *F1-Score*.

Table 1. Confusion matrix

Ture condition				factor
Predicted condition		1 (fraudulent)	0 (non-fraudulent)	
	1 (fraudulent)	TP	FP	*Pre*
	0 (non-fraudulent)	FN	TN	
factor		*Rec*		

The factor *Acc* represents a percentage of the number of correct predictions and total predictions, as defined by using Eq. (1). The factor can be used to evaluate the effectiveness of a classifier. For balanced dataset, the factor *Acc* can be considered as a main factor.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

The factor *Pre* is a proportion of the number of *TP* with the number of *TP* and *FP*, as represented by using Eq. (2). The factor *Pre* can be applied to estimate the correctness of positive outcome of a classifier.

$$Pre = \frac{TP}{TP + FP} \tag{2}$$

The factor *Rec* represents a percentage of the number of *TP* with the number of *TP* and *FN*, as defined by using Eq. (3). The factor *Rec* can be employed to evaluate the rate that a classifier correctly identifies actual positives.

$$Rec = \frac{TP}{TP + FN} \tag{3}$$

Increased factor *pre* associates with the decreased factor *Rec*. The factor *F1-Score* balances the factors *Pre* and *Rec*, as expressed by using Eq. (4). This study employed the factor *F1-Score* to evaluate mainly the performance of a classifier.

$$F1\text{-}Score = \frac{2 \times Rec \times Pre}{Rec \times Pre} \tag{4}$$

3. METHODS

This study used the LSTM-Adam, the LSTM-Sgdm, the LSTM- Rmsprop, the BPNN-GDA and the BPNN-SCGA classifiers to identify a default of credit card clients taken from the literature [10, 11]. This study selected the best numerical results from the LSTM methods and compared the results with those of the BPNN-GDA and the BPNN-SCGA classifiers. A one-way ANOVA is performed. The Fisher test is then executed when the one-way ANOVA is statistically significant.

3.1. Dataset Description

The dataset consists of 23 features and 30,000 instances, which include default instances (fraudulent transactions) of 6,636. The description of features is listed in Table 2. This study employed 10-fold cross-validation. This study used 90% of the dataset to train the LSTM and the BPNN models and employed other 10% to test models.

Table 2. The description of each feature in the dataset

Variables	Features	Description	Type
x_1	LIMIT_BAL	amount of the given credit	numerical
x_2	SEX	gender (1 = male; 2 = female)	categorical
x_3	EDUCATION	education (1 = graduate school; 2 = university; 3 = high school; 4 = others)	categorical
x_4	MARRIAGE	marital status (1 = married; 2 = single; 3 = others)	categorical
x_5	AGE	age (year)	categorical
x_6	PAY_0	history of past payment, the repayment status in September, 2005	numerical
x_7	PAY_2	history of past payment, the repayment status in August, 2005	numerical
x_8	PAY_3	history of past payment, the repayment status in July, 2005	numerical
x_9	PAY_4	history of past payment, the repayment status in June, 2005	numerical
x_{10}	PAY_5	history of past payment, the repayment status in May, 2005	numerical
x_{11}	PAY_6	history of past payment, the repayment status in April, 2005	numerical
x_{12}	BILL_AMT1	amount of bill statement in September, 2005	numerical
x_{13}	BILL_AMT2	amount of bill statement in August, 2005	numerical
x_{14}	BILL_AMT3	amount of bill statement in July, 2005	numerical
x_{15}	BILL_AMT4	amount of bill statement in June, 2005	numerical
x_{16}	BILL_AMT5	amount of bill statement in May, 2005	numerical
x_{17}	BILL_AMT6	amount of bill statement in April, 2005	numerical
x_{18}	PAY_AMT1	amount paid in September, 2005	numerical
x_{19}	PAY_AMT2	amount paid in August, 2005	numerical
x_{20}	PAY_AMT3	amount paid in July, 2005	numerical

Table 2. The description of each feature in the dataset (count.)

Variables	Features	Description	Type
x_{21}	PAY_AMT4	amount paid in June, 2005	numerical
x_{22}	PAY_AMT5	amount paid in May, 2005	numerical
x_{23}	PAY_AMT6	amount paid in April, 2005	numerical
y	Default	1 for fraudulent transactions, 0 otherwise	categorical

3.2. Data Pre-Processing

A data normalization is used to rescale the feature values to create the expected inputs, as defined by using Eq. (5).

$$x_{ij}^{'} = \frac{x_{ij} - \mathbf{x}_i^{\min}}{\mathbf{x}_i^{\max} - \mathbf{x}_i^{\min}}\left(E_{\max} - E_{\min}\right) + E_{\min}, \quad (5)$$
$$i = 1, 2, ..., i_{\max} \;\; j = 1, 2, ..., n_{\text{total}}$$

Where

$x_{ij}^{'}$ = normalized desired output j of input i
$\mathbf{x}_i^{\min}$ = minimum value of input vector i
$\mathbf{x}_i^{\max}$ = maximum value of input vector i
$E_{\min}$ = minimum value of expected output
$E_{\max}$ = maximum value of expected output
n_{total} = total number of a dataset

These values [$E_{\min}$, $E_{\max}$] are generally set to [0.2, 0.8].

3.3. Parameter Settings

The parameter settings of the LSTM and the BPNN classifiers are listed in Tables 3. and 4. For each parameter settings, this study used 10-fold cross-validation.

Table 3. The parameter settings of the proposed LSTM classifier

Parameter Settings	Values
Training function	Adam, Sgdm, Rmsprop
Gradient threshold	1
Maximum epoch	1000
Number of hidden units	10, 20, 30 ,40, 50
Initial learning rate	0.1
Drop period of a learning rate	500
Drop factor of a learning rate	0.2

Table 4. The parameter settings of the BPNN classifier

Parameter Settings	Values
Training function	trainscg
Learning function	learngdm
Number of hidden neurons	10, 20, 30, 40, 50
Learning rate	0.1
Transfer function in a hidden layer	tansig
Transfer function in an output layer	purelin
Terminal conditions	maximum epoch = 1000 or reaching the error goal = 0.000001

3.4. Classification Threshold Value

For an imbalanced dataset, a *CT* (classification threshold) value must be carefully defined. This study employed the *CT* values 0.3 and 0.5, as expressed by using Eq. (6).

$$\begin{cases} \text{class 1, if network output} \geq CT \\ \text{class 0, if network output} < CT \end{cases} \tag{6}$$

4. NUMERICAL RESULTS

The DL and parallel computing toolboxes in the MATLAB 2020b software were executed on a notebook that has an Intel Core (TM) i9-1190H 2.50 GHz and 64 GB RAM. The LSTM (GPU mode) and BPNN classifiers were performed based parameter settings described in subsection 3.3. Several numerical results were summarized, such as mean training *Acc* (%), mean testing *Acc* (%), mean training *Pre* (%), mean training *Rec* (%), mean training *F1-score* (%), mean testing *Pre* (%), mean testing *Rec* (%), mean testing *F1-score* (%) and mean computation time (MCT).

4.1. Numerical results obtained from the LSTM classifier

This study used the LSTM with the Adam, the Sgdm, and the Rmsprop optimizers for the number of neurons {10, 20, 30, 40, 50} by using the *CT* value = 0.3. The best numerical results are shown in Table 5., indicating that the LSTM-Adam classifier can obtain the best mean testing *F1-score* (%).

Table 6. shows the best numerical results obtained from the LSTM-Adam, the LSTM-Sgdm and the LSTM-Rmsprop methods for the number of neurons {10, 20, 30, 40, 50} by using the *CT* = 0.5, and showing that the LSTM-Adam classifier can find the best mean testing *F1-score* (%).

According to Tables 5. – 6., the LSTM-Adam classifier can obtain the best mean testing *F1-score* (%) by using the *CT* = 0.3 for the imbalanced dataset.

Table 5. The best numerical results obtained from the LSTM-Adam, the LSTM-Sgdm and the LSTM-Rmsprop methods with the *CT* = 0.3

Methods	the number of neurons	mean training *Acc* (%)	mean testing *Acc* (%)	mean training *Pre* (%)	mean training *Rec* (%)	mean training *F1-score* (%)	mean testing *Pre* (%)	mean testing *Rec* (%)	mean testing *F1-score* (%)	MCT (sec.)
LSTM-Adam	**10**	**80.67**	**80.63**	**57.30**	**49.41**	**53.05**	**57.41**	**49.36**	**53.03**	**561.27**
LSTM-Sgdm	20	80.36	80.58	56.95	47.04	51.45	57.19	46.68	51.26	541.89
LSTM-Rmsprop	10	80.55	80.26	56.91	49.90	53.15	56.28	48.95	52.31	558.08

Table 6. The best numerical results yielded from the LSTM-Adam, the LSTM-Sgdm and the LSTM-Rmsprop methods with the *CT* = 0.5

Methods	the number of neurons	mean training *Acc* (%)	mean testing *Acc* (%)	mean training *Pre* (%)	mean training *Rec* (%)	mean training *F1-score* (%)	mean testing *Pre* (%)	mean testing *Rec* (%)	mean testing *F1-score* (%)	MCT (sec.)
LSTM-Adam	**10**	**81.39**	**81.30**	**67.68**	**30.35**	**41.90**	**66.87**	**30.10**	**41.43**	**562.27**
LSTM-Sgdm	10	81.12	81.04	69.63	26.03	37.83	69.13	25.42	37.11	561.27
LSTM-Rmsprop	10	81.38	81.17	68.31	29.50	41.19	66.84	28.94	40.33	561.02

4.2. Numerical results obtained from the BPNN classifier

This study employed the BPNN classifiers with the SCGA and GDA for the number of neurons {10, 20, 30, 40, 50} by using the *CT* value = 0.3. The best numerical results are shown in Table 7., indicating that the BPNN-SCGA classifier can yield the best mean testing *F1-score* (%).

Table 7. The best numerical results obtained from the BPNN-SCGA and the BPNN-GDA classifiers with *CT* = 0.3

Methods	the number of neurons	mean training *Acc* (%)	mean testing *Acc* (%)	mean training *Pre* (%)	mean training *Rec* (%)	mean training *F1-score* (%)	mean testing *Pre* (%)	mean testing *Rec* (%)	mean testing *F1-score* (%)	MCT (sec.)
BPNN-SCGA	**10**	**80.69**	**80.57**	**57.62**	**48.38**	**52.56**	**57.20**	**48.04**	**52.14**	**8.75**
BPNN-GDA	50	80.03	80.11	56.34	43.58	49.10	56.37	43.39	48.98	12.42

Table 8. shows the best numerical results obtained from the BPNN-SCGA and the BPNN-GDA classifiers for the number of neurons {10, 20, 30, 40, 50} by using the *CT* = 0.5, showing that the best mean testing *F1-score* (%) obtained from the BPNN-SCGA classifier is strongly superior to that of the BPNN-GDA classifier.

Referring to Tables 7. – 8., the numerical results obtained from the BPNN-SCGA and the BPNN-GDA classifiers by using the *CT* = 0.3 are superior to those yielded from the BPNN-SCGA and the BPNN-GDA classifiers by using the *CT* = 0.5.

Table 8. The best numerical results yielded from the BPNN-SCGA and the BPNN-GDA classifiers with the *CT* = 0.5

Methods	the number of neurons	mean training *Acc* (%)	mean testing *Acc* (%)	mean training *Pre* (%)	mean training *Rec* (%)	mean training *F1-score* (%)	mean testing *Pre* (%)	mean testing *Rec* (%)	mean testing *F1-score* (%)	MCT (sec.)
BPNN-SCGA	**10**	**82.02**	**81.97**	**67.42**	**36.19**	**47.09**	**66.76**	**35.89**	**46.64**	**8.65**
BPNN-GDA	50	78.45	78.47	70.73	4.32	8.08	71.12	4.51	8.40	12.22

4.3. Comparison

Table 9. Lists the comparison of the best numerical results obtained from the LSTM-Adam, the BPNN-SCGA and the BPNN-GDA classifiers. For fraud issues, a classifier can correctly detect the factor *Rec* is important, miss fraud may cause a critical risk. Therefore, this study focuses on the mean testing *Recs*. A one-way ANOVA was performed, and indicating that the *P* value (0.026) is smaller than or equals to a significant level 0.05, and that at least two mean testing *Recs* are statistically different. The Fisher test was then executed, showing that the mean testing *Rec* obtained from the LSTM-Adam and the BPNN-SCGA classifiers are identical and are superior to that of the BPNN-GDA classifier. As shown in Table 9., the performance of the LSTM-Adam and the BPNN-SCGA classifiers is similar and spent MCT of the LSTM-Adam classifier is more than that of the BPNN-SCGA classifier.

Learning algorithms of the LSTM-Adam, the BPNN-SCGA and the BPNN-GDA classifiers are EBP algorithms based on gradient methods. The LSTM-Adam and the BPNN-SCGA methods can overcome the drawback that traps into local optima of the standard BPNN-GDA approach.

Table 9. Comparison of the best numerical results obtained from the LSTM-Adam, the BPNN-SCGA and the BPNN-GDA classifiers

Methods	mean testing *Pre* (%)	mean testing *Rec* (%)	mean testing *F1-score* (%)	MCT (sec.)
LSTM-Adam	57.41	49.36	53.03	561.27
BPNN-SCGA	57.20	48.04	52.14	8.75
BPNN-GDA	56.37	43.39	48.98	12.42

5. CONCLUSION

This study used the LSTM-Adam, the LSTM-Sgdm, the LSTM-Rmsprop, the BPNN-SCGA and the BPNN-GDA classifiers for identifying a default of credit card clients, which is an imbalanced dataset. This study employed 10-fold cross-validation. Many remarks are given. For the imbalanced dataset, a *CT* value must be carefully selected. For the LSTM method, the performance of the Adam optimizer is superior to the Sgdm and the Rmsprop algorithms. The LSTM-Adam and the BPNNS-SCGA classifier can achieve identical performance. Therefore, the LSTM-Adam classifier has the potential to deal with credit scoring problems, which are binary classification problems.

6. FUTURE WORK

Future work will compare the performance of the LSTM-Adam classifier with those of supervised ML algorithms, such an SVM classifier (statistical based algorithm), a KNN classifier (instance-based learners) and a Logistical regression method. Moreover, this study will use the resampling methods (such as the random undersampling and the random oversampling methods) for the imbalanced dataset.

REFERENCES

[1] Khan, F. N., Khan, A. H. & Israt, L., (2020) "Credit card fraud prediction and classification using deep neural network and ensemble learning," in 2020 IEEE Region 10 Symposium (TENSYMP), pp 114-119.

[2] Brown, K. & Moles, P., (2014) Credit Risk Management. Edinburgh, United Kingdom: Edinburgh Business School.

[3] Sehgal, S., (2018) "Human activity recognition using BPNN classifier on HOG features," in 2018 International Conference on Intelligent Circuits and Systems (ICICS), pp 286-289.

[4] Lawi, A. & Aziz, F., (2018) "Classification of credit card default clients using LS-SVM ensemble," in 2018 Third International Conference on Informatics and Computing (ICIC), pp. 1-4.

[5] Vaishnnave, M. P., Devi, K. S., Srinivasan, P. & Jothi, G. A. P., (2019) "Detection and classification of groundnut leaf diseases using KNN classifier," in 2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN), pp 1-5.

[6] Zhu, T., Li, K., Herrero, P. & Georgiou, P., (2021) "Deep learning for diabetes: A Systematic review," IEEE Journal of Biomedical and Health Informatics, vol. 25, no. 7, pp 2744-2757.

[7] Chaabane, F., Réjichi, S. & Tupin, F., (2020) "Comparison between multitemporal graph based classical learning and LSTM model classifications for sits analysis," in 2020 IEEE International Geoscience and Remote Sensing Symposium, pp 144-147.

[8] Chen, D., Zhang, J. & Jiang, S., (2020) "Forecasting the short-term metro ridership with seasonal and trend decomposition using loess and LSTM neural networks" IEEE Access, vol. 8, pp 91181-91187.

[9] Moller, A. F., (1993) "A scaled conjugate gradient algorithm for fast supervised learning," Neural Networks, vol. 6, no. 4, pp 525-533.

[10] Dua, D. & Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[11] Yeh, I. C. & Lien, C.H., (2009) "The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients," Expert Systems with Applications, vol. 36, no. 2, Part 1, pp 2473-2480.

[12] Portugal, I., Alencar, P. & Cowan, D., (2018) "The use of machine learning algorithms in recommender systems: A systematic review," Expert Systems with Applications, vol. 97, pp 205-227.

[13] Gupta, G. & Katarya, R., (2021) "A study of deep reinforcement learning based recommender systems," in 2021 2nd International Conference on Secure Cyber Computing and Communications (ICSCCC), pp. 218-220.

[14] Hsu, T. C., Liou, S. T., Wang, Y. P., Huang, Y. S. & Che, L., (2019) "Enhanced recurrent neural network for combining static and dynamic features for credit card default prediction," in 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp 1572-1576.

[15] Olah, C. (2021). Understanding LSTM networks. Available: http://colah.github.io/posts/2015-08-understanding-lstms/

[16] Chang, Z., Zhang, Y. & Chen, W., (2018) "Effective Adam-optimized LSTM neural network for electricity price forecasting," in IEEE 9th International Conference on Software Engineering and Service Science (ICSESS), pp. 245-248.