SN Computer Science (2023) 4:557
https://doi.org/10.1007/s42979-023-01977-w

ORIGINAL RESEARCH

Auto-Encoder and LSTM-Based Credit Card Fraud Detection

Deepthi Sehrawat[1] · **Yudhvir Singh**[2]

Received: 21 February 2023 / Accepted: 30 May 2023
© The Author(s), under exclusive licence to Springer Nature Singapore Pte Ltd 2023

Abstract
The increased fraud risk due to the most recent methods of paying with a credit card, such as real-time payments and cards with near-field communication (NFC) capabilities, makes detecting credit card fraud an essential topic of study. Deep learning has shown encouraging results in recent years, whenever it is used to detect credit card fraud. There have been several deep learning-based models presented in this field of research. Still, not all of them have proven to be the most effective because each technique is best used with a particular dataset, and fraudsters constantly refine their methods to evade detection by the systems in place today. This work's primary focus is detecting credit card fraud using an auto-encoder with GRU and LSTM models. This is an efficient mechanism; first, data are passed to auto-encoder without the labels, and after that, the output generated by the auto-encoder is passed to the LSTM model as input with labels to detect fraud.

Keywords Credit card · Auto-encoder · Gated recurrent unit (GRU) · Long short-term memory (LSTM) · Performance metrics

Introduction

Credit cards are essential payment instruments because they give users quick and straightforward access to a short-term credit line while completing a transaction. This enhances spending power while delivering benefits such as convenience, gift cards, and rewards. Paying payments on time and a regular basis might help the user create a better credit score, making it simpler to get long-term loans. Credit cards not only provide numerous benefits but can also be used to commit fraud.

Most e-commerce platforms struggle greatly with credit card theft because it can have direct and indirect financial repercussions for the victim. Genuine credit card customers suffer enormous losses as a result of fraud. Credit card fraud is rising as credit card transactions become the most popular payment method for all online and offline transactions. Credit card fraud is classified as either internal or external. In contrast to external card fraud, which involves using a stolen credit card to get funds through dubious means, internal card fraud involves cooperation between cardholders and banks to commit fraud while using false identities. Kinds of credit card scams:

1. Applications fraud: When a fake account is formed, the imposter takes over the installation process by gaining private user information such as a secret code and username. Identity theft generally happens to a family member. When a fraudster applies for a loan or a new credit card entirely on behalf of the cardholder, they steal supporting paperwork to back up or substantiate their false claim.
2. POS (point-of-sale) fraud: Simple reading devices are installed to hijack user data on popular PoS devices, such as cash registers. These gadgets scan and save the customer's card data when swiping them.
3. Phishing and vishing refer to impersonating official bank communication to trick users into clicking on bogus links. These scams frequently link users to websites that look legitimate. While utilizing such websites, the user is sent to websites that look accurate. Scammers can use

This article is part of the topical collection "Machine Intelligence and Smart Systems" guest edited by Manish Gupta and Shikha Agrawal.

✉ Deepthi Sehrawat
sehrawatdeepthi@gmail.com

Yudhvir Singh
Yudhvir.singh@mdurohtak.ac.in

[1] UIET, MDU, Rohtak, India

[2] Computer Science and Engineering, UIET, MDU, Rohtak, India

Published online: 29 July 2023

a consumer who clicks on one of these bogus URLs and submits their credit card details.

4. Counterfeit card fraud: Bogus card fraud attempts typically use the scaling process. Constructing a fake swipe card with identical qualities to the real one is possible. The phoney card is used to complete further purchases and is fully functioning.
5. Keystroke logging: As more financial transactions are done online, fraudsters increasingly utilize malicious software to log keystrokes. This usually occurs after accidentally downloading malware into the system's hard drive by clicking an untrusted link. The malware records every keystroke on the machine and takes credit card numbers, PINs, and other sensitive data.
6. Personal identification information (PII) theft: PII scams are identical to application scams. Identity fraud occurs when the first card is used fraudulently to use the card or establish a new account. It is not easy to detect.

Much research has been conducted to identify outside card fraud, which accounts for most credit card thefts. Fraud detection is a complicated mathematical process. The success of any fraud detection procedure is determined by the change of the parameter selection, which is one of the many parameters that must be made. Detecting fraudulent transactions via human methods is expensive and inefficient; as a result, the emergence of machine learning has rendered traditional tactics outdated. A typical fraud detection system contains both an automated tool and a human method. The computerized method is based on fraud detection criteria. It checks all new transactions and assigns each one a fraud score. The focus of fraud investigators is on transactions that are highly likely to be fraudulent, and they provide a binary answer (fraud or not fraud) for every transaction they look at.

This paper proposes a new methodology by combining auto-encoder developed using GRU and LSTM models to improve the process of credit card fraud detection. Precision, recall, F1 score and AUC score are used for evaluation.

Many researchers published papers presenting some new techniques to discover fraud. Javad et al. [1] proposed a unique voting system based on ANN and an ensemble model for identifying fraudulent behavior based on sequential dataset processing deep recurrent neural networks. They demonstrated a unique algorithm for training the previously discussed voting method. Between April 14, 2004, and September 12, 2004, they exploited two datasets: one from European Bank Customers and the other from a major Brazilian bank. They displayed the AUC-ROC and AUC-PR and generated the recall, precision, and F1 score for performance verification. For the experiments, they employed LSTM and GRU, as well as an ensemble approach using GRU/LSTM. Voting classifier training, middle vector generation, and base classifier training were the three processes for ensemble approaches. Mohamad et al. [2] developed a novel hybrid model for identifying credit card fraud. The proposed approach is based on employing deep auto-encoders in conjunction with OSVM. After training, the model is an unsupervised machine learning model trained on a single class (a regular or genuine transaction) and can distinguish between legal and fraudulent transactions. Compared to auto-encoders and OSVM, the suggested model achieved similar results across all relevant metrics. The suggested model is created by combining deep auto-encoders with OSVM, where the auto-encoder is trained using only authentic transactions and a gradient descent approach to minimize the model's cost function, which is the distance between the input and output assessed by mean square error (MSE) (reconstructed input). Following auto-encoder training, the MSE of the entire training set is utilized for training an OSVM with a one-dimensional feature space. The paper by Awoyemi et al. [3] investigates the effect of hybrid sampling on the fraudulent identification performance of Naïve Bayes, K-nearest neighbor, and logistic regression classifiers on a highly diverse credit card fraud dataset. This research uses innocent, K-nearest neighbor, and logistic regression approaches to compare credit card fraudulent identification on a highly diverse dataset. The accuracy, sensitivity, specificity, Mathew's correlation coefficient (MCC), and equilibrium classification rate are used to compare the three approaches' performance. This study makes use of a dataset of 284,807 transactions from European cardholders. The dataset is extremely biased. Abhimanyu et al. [4] use a dataset of around 80 million credit card transactions to examine the efficiency of a subset of deep learning topologies, ranging from the standard artificial neural network to topologies with built-in temporal and memory components, such as long short-term memory, and various parameters in identifying fraud. They avoid common fraud detection difficulties such as class imbalance and scalability by employing a high-speed, distributed cloud computing architecture. Their research provides a comprehensive reference to model parameter sensitivity analysis regarding fraud detection performance. In addition, they provide a framework for adjusting the parameters of deep learning topologies for detecting credit card fraud, which would assist financial institutions in reducing losses by preventing fraudulent activity. Adriano et al. [5] introduced the customized fraud BNC (Bayesian Network Classifier) technique to identify genuine credit cards. A hyper-heuristic evolutionary algorithm (HHEA) was used to develop customized fraud BNC, which incorporates information on BNC approaches into a taxonomy and finds the ideal combination of these components for a given dataset. By applying N combinations of algorithm components and choosing the best one, HHEA takes two inputs—a dataset and algorithm components—and generates the best and fittest algorithm for a given dataset. HHEA finds and

explores the universe of BNC algorithms using a real-coded evolutionary algorithm. In Johannes et al. [6] study, the detection of fraud problems is defined as a sequence classification task, and transaction sequences are incorporated utilizing LSTM networks. Using two different datasets—offline (face-to-face) transactions and internet transactions. To improve the comparability of the Dataset, they deleted those transactions which did not precede by at least $w = 9$ earlier transactions. On matching sets of transactions, the random forest classifier (RFC) and LSTM may be trained, verified, and tested. They trained LSTM using 5 (SHORT) and 10 (LONG) sequences (LONG). GridSearchCV was used to discover the most optimally configured Model. Combating and detecting fraud is a time-consuming, costly, and labor-intensive undertaking. Zainab et al. [7] did thorough experimental research to address the imbalance classification issue and developed several novel solutions. They examined the solutions and machine learning techniques to detect fraud. Using a credit card fraud labeled dataset, they uncovered their flaws and summarized their findings. This study shows that current methods produce a lot of false alarms that cost financial organizations money. If this occurs, it may result in identification mistakes and a rise in fraud. In this study, they focused on counterfeit fraud since it is more difficult to detect, and the damage it does is irreparable. This study used eight machine learning classification algorithms, including C5.0, SVM, ANN, NB, Bayesian belief network, LR, and AI. Recall, precision, and accuracy were calculated for the model's performance using a confusion matrix. They employed the receiver operating characteristics (ROC) curve, which is well recognized for performance measurement and is commonly used in classifiers. This may be calculated using the area within the precision–recall curve (AUPRC). The study by Fiorea et al. [8] aids in dealing with unbalanced datasets and allows the classifier to function optimally. Because the dataset is skewed towards the majority class, imbalanced datasets can greatly limit the performance of binary classifiers, making it difficult for the classifier to identify fraud. To alter the dataset, use generative adversarial networks. Reproductive deep learning models based on GAN are dynamic, adaptable, and capable of creating unique figures. By training a GAN, these smaller class instances were produced. The training dataset was mixed with the upgraded training set to produce the final product. GAN has two components: generating and discriminative, which compete. When constructing the final class, a 3-layer network comprised of 30 ReLu modules, 30 Sigmoid modules, and 2 Softmax modules was chosen as the best classifier. To make its final choice, the discriminator D employs a three-layer perceptron with hidden layers comprised of 36 Sigmoid units each. Generator G also has three levels, the first two of which include ReLU units and the third of which contains Sigmoid units. The study by Xinwei et al. [9] focuses mostly on feature selection and optimizing the model, allowing the model to give the best results using the modified dataset. One of their most significant achievements has been developing a fraud detection system based on a deep learning architecture and an enhanced feature engineering technique based on homogeneity-oriented behavior analysis (HOBA). The dataset was also compared with RFM and HOBA. They combined SVM, random forest, DBN, CNN, and RNN deep learning models. By assembling many Bernoulli–Bernoulli restricted Boltzmann machines (RBM) with binary input characteristics, DBMs (deep belief networks) are created. Using the RFM and HOBA frameworks independently, they calculated F1 score, accuracy, recall, precision, and false-positive rate for all models. The paper by Fabrizio et al. [10] offers a hybrid strategy for improving fraud detection accuracy by combining supervised and unsupervised algorithms. Unsupervised outlier scores are compared and evaluated on a genuine, labeled detection of credit card fraud dataset. The results of the trials show that the combination is advantageous and improves accuracy. They trained the model using the best-of-both-worlds approach, which is a sequential strategy. They modified an original dataset using multiple unsupervised outlier algorithms utilizing a collection of outlier scores. To supplement DS, the unsupervised model's outlier score vector s0 over the original dataset DS is used: DS' = (DS, s0). The team then used a logistic regression model to analyze the results regarding AUC-ROC in three different scenarios: initial dataset alone, outlier scores alone, and original dataset plus outlier score. Before the augmentation of the dataset DS, they used global, local, and cluster granularity techniques. Global strategy: the complete dataset is considered while determining the outlier. To detect, they considered the amount. Using the average amount, any transaction amount that differs significantly from the average will be considered an outlier. Local approach: for some time, examined each dataset one by one and detected the abnormality. Cluster approach: select two cluster types, one with high and one with low spending, then average each separately while looking for outliers within their cluster. Jinliang et al. [11] used denoising auto-encoders to train the model effectively. With the help of this model, they added noise to the data so that the auto-encoder neural network could learn how to remove it and recover as much of the original information as possible. After the dataset was balanced, it was passed to the denoised auto-encoder, which consists of seven layers of auto-encoders for thedataset's denoising procedure. An evenly distributed training dataset is produced by oversampling, which is subsequently contaminated with Gaussian noise and supplied to the denoised auto-encoder. After training, this denoised auto-encoder model could denoise the testing dataset throughout the prediction process.

Methodology

Credit card fraud detection has recently emerged as a complex and trending research topic. Credit card transaction dataset is not widely accessible; banks do not offer dataset directly for security concerns. Credit card frauds are complicated to detect because most datasets we used had relatively few fraud instances. Fraudsters improve themselves regularly by applying advanced techniques, making algorithms harder to detect. For this research, we have downloaded the publicly available dataset. The information includes card transactions performed in September 2013 by European cardholders. The dataset has a total of 30 features. It has 284,807 transactions, of which 492 are fraud incidents, representing 0.172% of all trades. This suggests that the dataset they utilized is highly skewed. Another distinguishing feature is that the dataset solely contains numerical input variables and has undergone a PCA transformation. Only the time and amount features have not altered. This dataset still needs a few pre-processing procedures because the 'Time' element was unnecessary for our method, so it was removed. To scale down the dataset and decrease computing costs, the MinMaxScaler technique was used. Because the dataset is highly skewed, the SMOTE technique with a random state of 10 was used. After using SMOTE, the number of fraud and genuine transactions became equal, i.e., in 199,014 transactions of genuine and fraud. After that, dataset was transformed into the 3D format since GRU and LSTM models take data in a 3D design (samples, time steps, features).

Auto-Encoder

An artificial neural network called an auto-encoder [12] duplicates input to output while attempting to approximate the identity function. As a result, auto-encoders do not need to be trained on any label or production to learn how to reconstruct the input. A rudimentary auto-encoder may be built by combining one information, hidden, and output layer, as shown in Fig. 1. The hidden layer typically has a smaller dimension than the input layer to learn the input's latent space representation.

The output layer has the exact dimensions as the input layer since it attempts to anticipate it. Figure 1 shows a schematic representation of an auto-encoder.

Any deep learning model may be utilized to build an auto-encoder; however, in our situation, we made the auto-encoder using the recurrent gated unit (GRU) model.

Gated Recurrent Unit (GRU)

GRU [13] is a model based on LSTM that Cho et al. introduced in 2014. For tackling vanishing gradients, GRU keeps the LSTM properties. Additionally, it is faster than LSTM and has fewer parameters and a more straightforward underlying complexity. Unlike GRU, which struggles with unbounded counting, the LSTM can perform it quickly, making it "strictly stronger" than GRU. As seen in Fig. 2, GRU contains two gates: an update gate, z and a reset gate, r.

For GRU, the hidden state h_t is calculated as follows [14]:

$$z_t = \sigma\left(W_z x_t + U_z h_{t-1} + b_z\right),$$
$$r_t = \sigma\left(W_r x_t + U_r h_{t-1} + b_r\right),$$
$$\tilde{h}_t = \tanh(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h),$$
$$h_t = \left(1 - z_t\right) \odot h_{t-1} + z_t \odot_t .$$

A recurrent gated unit allows each recurrent unit to record relationships on multiple time scales adaptively. GRUs, like LSTMs, include gating units that control input flow into the team; however, GRUs do not have different memory cells.

Short-Term Long Memory (LSTM)

Extended short-term memory networks (LSTM) [15, 16] are a subset of recurrent neural networks (RNN). In the 1980s, recurrent neural networks were developed to represent time series data. The construction of an RNN is

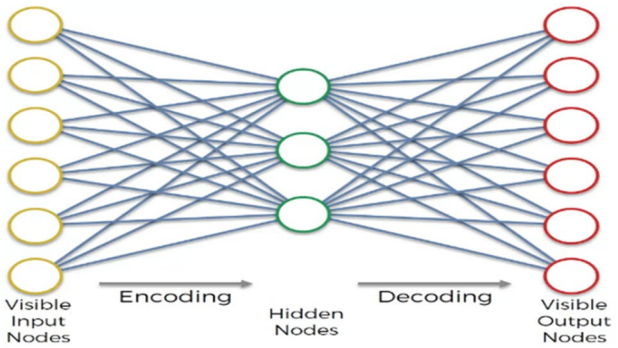

Fig. 1 Auto-encoder

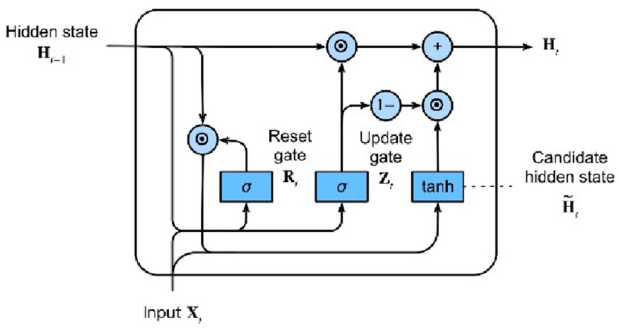

Fig. 2 Gated recurrent unit internal structure

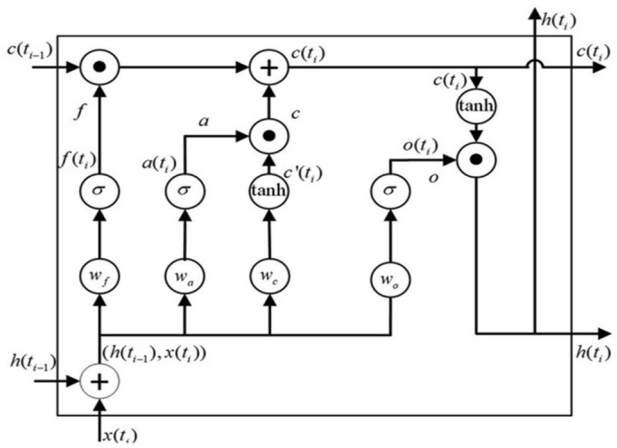

Fig. 3 LSTM internal structure [17]

the same as that of a standard multilayer perceptron, with the addition of connections between hidden units linked by discrete time steps. Long-term dependencies can be learned through LSTM.

A typical LSTM block, as shown in Fig. 3, consists of a cell with an input gate, an output gate, and a forget gate. Mathematically the forward learning of an LSTM is as follows [17]:

$$
\begin{aligned}
a(t_i) &= \sigma\left(w_a x(t_i) + w_{ha} h(t_{i-1}) + b_a\right),\\
f(t_i) &= \sigma\left(w_f x(t_i) + w_{hf} h(t_{i-1}) + b_f\right),\\
c(t_i) &= f_t \times c(t_{i-1}) + a_t \times \tanh(w_c x(t_i) + w_{hc}\left(h(t_{i-1}) + b_c\right),\\
o(t_i) &= \sigma\left(w_o x(t_i) + w_{ho} h(t_{i-1}) + b_o\right),\\
h(t_i) &= o(t_i) \times \tanh(c(t_i)),
\end{aligned}
$$

where $x(t_i)$: the input value, $h(t_{i-1})$ and $h(t_i)$: the output value at time point t_{i-1} and t_i, $c(t_{i-1})$ and $c(t_i)$: cell state at t_{i-1} and t_i, $b = \{b_a, b_f, b_c, b_o\}$ are biases of input, forget, internal state and output gate. $W_1 = \{w_a, w_f, w_c, w_o\}$ are weight matrixes of input, forget, internal state and output gate. $W_2 = \{w_{ha}, w_{hf}, w_{hc}, w_{ho}\}$ are the recurrent weights. $a = \{a(t_i), f(t_i), c(t_i), o(t_i)\}$ are the output results for input, forget, internal state and output gate. σ and tan$h$ are activation functions, and $\times$ indicates point-wise multiplication.

Additionally, it is essential to note that there is a more potent and extensive variation of LSTM called Bidirectional LSTM, which includes data from present and forthcoming fresh samples to anticipate the current point of data. We do not have access to subsequent new transactions after the current transaction, which is a problem when dealing with the issue of identifying credit card fraud in a real-world scenario. In our proposed Model, we must continue to use a basic LSTM layer because it is impossible to use bidirectional LSTM as a classifier in this task.

Results

As mentioned, data flow passes data without a label to the auto-encoder. Its output is sent to the LSTM model as input, making our proposed Model more efficient when dealing with fraud cases where the number of fraud rows is deficient.

The auto-encoder outperformed with a minimum loss of 0.0054. It consists of two parts, the encoder and the decoder–encoders comprise four layers. The 1st layer consists of 24 nodes with an input size of 29, the 2nd layer consists of 19 nodes, the 3rd layer consists of 10 nodes, and the last layer, i.e., the 4th layer, consists of 4 nodes. The decoder also consists of 4 layers. The 1st layer consists of 10 nodes, the 2nd layer consists of 19 nodes, the 3rd layer consists of 24 nodes and the last layer consists of 29 nodes, an output layer. Each hidden layer and input layer's output is sent to the ReLU activation function. The production of the output layer was transmitted to the sigmoid function, which produced the final 29 values. Auto-encoder is constructed using a GRU model with several stacked layers 1. The loss function used for this model is Mean Square Error Loss (MSELoss), and Adam optimizer is used with a learning rate of 0.001. Early stopping has been done for this auto-encoder after the 8th epoch; loss did not change, as shown in Fig. 4.

The data of the auto-encoder have been passed to the LSTM model, which consists of an input size of 29, a hidden size of 35 and a no. of a stacked layer of 5. The output of the LSTM model has been sent to the linear model with a discreet size of 35 and an output size of 1. After that output of the linear model is sent to the sigmoid activation function. The loss function used for this model is binary cross entropy loss (BCELoss), and the Adam optimizer was used with a learning rate of 0.001. The proposed model was trained on GPU to reduce the computational cost and time. After training loss of each epoch is shown in Fig. 5.

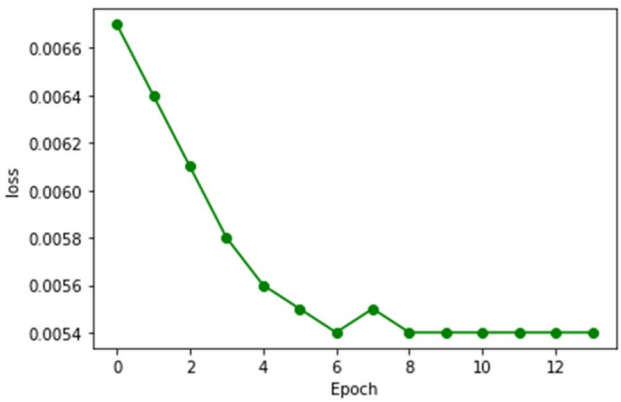

Fig. 4 Loss of auto-encoder constructed using gated recurrent unit (GRU)

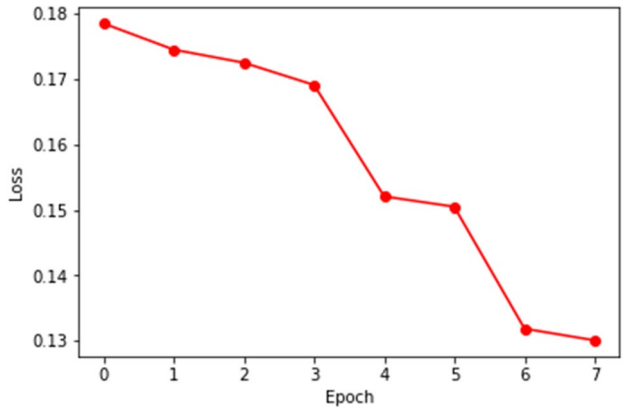

Fig. 5 Loss of LSTM model

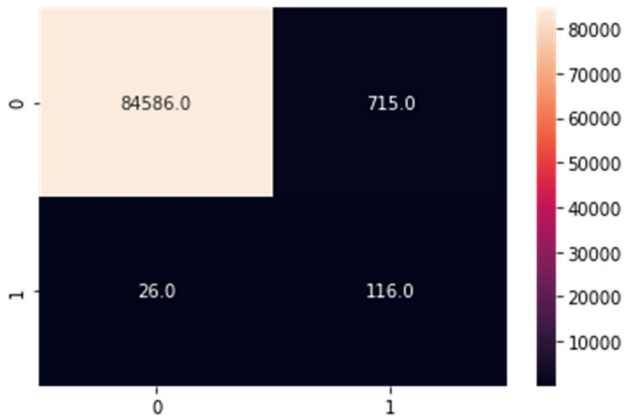

Fig. 7 Confusion matrix of whole proposed model

Accuracy of the LSTM model after each epoch is shown in Fig. 6.

For performance evaluation [18], accuracy, F1 score, precision, and recall have been calculated for test data which comes out to be 99.13%, 0.62, 0.57 and 0.90, respectively. The confusion matrix of predicted values of the LSTM model is shown in Fig. 7.

As shown in Fig. 7, true-negative values are 84,586, false-positive values are 715, false-negative values are 26, and true-positive values are 116. The test data's legitimate value consists of 85,301 and fraud cases 142. Even though precision is low, it is not that relevant in the case of imbalanced fraud data, whereas recall is high. It is appropriate as it states that it measures the proportion of actual positive patients that got predicted as positive (or true positive). In other words, a high recall value means there were very few false negatives and that the classifier is more permissive in the criteria for classifying something as positive. Table 1 shows comparison between the proposed model (in bold) and other similar models.

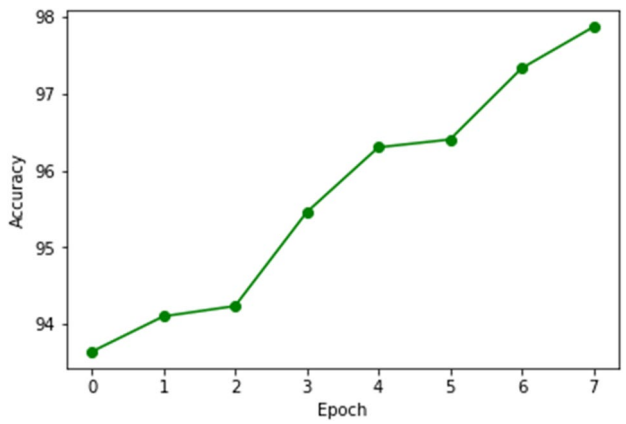

Fig. 6 Accuracy of LSTM model

Conclusion and Future Work

Many fraud detection techniques are available today, but none can detect all frauds as they occur; instead, they are seen after the crime has been committed. A relatively small fraction of all transactions are fraudulent, which explains why. To eliminate fraudulent transactions quickly and cheaply, we need technology to spot them as they occur.

As a result, today's fundamental goal is to create a credit card fraud detection system that is exact, fast to detect, and precise. It must recognize online frauds like phishing and credit card frauds when a compromised credit card is used. Although each strategy has benefits, none can ensure the same outcomes in every circumstance. For specific datasets, they yield excellent findings; while for others, they produce results that are subpar or unsatisfactory. Artificial neural networks (ANN) and Naive Bayesian networks, for example, have reasonable detection rates and precision but require significant training. While KNN, Naïve Bayes and SVM [19] perform well when dealing with small datasets, they are not very scalable when working with massive datasets.

In this paper, we employed the LSTM model with auto-encoder. Auto-encoder is constructed using GRU model with no. of stacked layers 1. The output of the auto-encoder has been sent to the LSTM model whose no. of stacked layers is 5. Auto-encoder outperforms with a loss of 0.0054, and the LSTM model also performs well with an accuracy of 99.13% in test data and a recall score of 0.90. Precision is too high because the test data he highly skewed, and the Model predicted most of the genuine transactions as genuine and almost all fraud transactions as fraud. We can consider accuracy and recall to check whether the model is working correctly.

Many proposed models are restricted to a particular dataset as credit card fraud detection is a problematic field whose dataset varies with a specific bank. Similarly, our proposed model works well with a particular dataset, and

Table 1 Comparison with similar models:

Model	Dataset	Accuracy	Recall	Precision
LSTM GRU [1]	European dataset	–	GRU—0.72 LSTM—0.74	GRU—0.862 LSTM—0.857
	Brazil dataset		GRU—0.68 LSTM—0.714	GRU—0.804 LSTM—0.877
Auto-encoder (AE)	Kaggle credit card fraud detection	–	1	0.422
OSVM		–	1	0.889
AE + OSVM [2]		–	1	0.938
Auto-encoder [11]		83.56%	0.906	–
Naïve Bayes		97.37%	0.807	0.968
KNN		96.91%	0.883	0.968
Proposed model	**European dataset**	**99.13%**	**0.9**	**0.57**

our model is inefficient in finding all fraud cases without missing any. We will do more research in this field in future to purely predict all fraud cases.

Data availability Information regarding the dataset which is used for this research is mentioned in the Methodology section. The dataset is publicly available.

Declarations

Conflict of Interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Forough J, Momtazi S. Ensemble of deep sequential models for credit card fraud detection; 2020. Tehran: Computer Engineering Department, Amirkabir University of Technology.
2. Jeragh M. Combining auto encoders and one class support vectors machine for fraudulant credit card transactions detection; 2018. Kuwait: Computer Engineering Department, Kuwait University, Mousa AlSulaimi, Information Technology Department, Boubyan Bank.
3. Awoyemi JO, Adetunmbi AO, Oluwadare SA. Credit card fraud detection using machine learning techniques: a comparative analysis; 2017. Akure: Department of Computer Science Federal University of Technology Akure.
4. Roy A, Sun J, Mahoney R, Alonzi L, Adams S, Beling P. Deep learning detecting fraud in credit card transactions; 2018. IEEE, University of Virginia.
5. de Sá AGC, Pereira ACM, Pappa GL. A customized classification algorithm for credit card fraud detection; 2018. Computer Science Department, Universidade Federal de Minas Gerais (UFMG), 31270–010, Belo Horizonte, Minas Gerais: Elsevier.
6. Jurgovskya J, Granitzer M, Ziegler K, Calabretto S, Portier P-E, He-Guelton L, Caelen O. Sequence classification for credit-card fraud detection. New York: Elsevier; 2018.
7. Makki S, Assaghir Z, Taher Y, Haque R, Hacid M, Zeineddine H. An experimental study with imbalanced classification approaches for credit card fraud detection. IEEE Access 2019;7:93010-22.
8. Fiorea U, De Santis A, Perla F, Zanetti P, Palmieri F. Using generative adversarial networks for improving classification effectiveness in credit card fraud detection. New York: Elsevier; 2017.
9. Zhang X, Hana Y, Xua W, Wang Q. HOBA: a novel feature engineering methodology for credit card fraud detection with a deep learning architecture. New York: Elsevier; 2019.
10. Carcillo F, Le Borgne Y-A, Caelen O, Kessac Y, Oblé F, Bontempi G. Combining unsupervised and supervised learning in credit card fraud detection. New York: Elsevier; 2019.
11. Jiang P, Zhang J, Zou J. Credit card fraud detection using autoencoder neural network; 2019. Department of Electrical & Computer Engineer, University of Western Ontario.
12. Zhai J, Zhang S, Chen J, He Q. Autoencoder and its various variants. 2018 IEEE international conference on systems, man, and cybernetics (SMC); 2018. p. 415–19. https://doi.org/10.1109/SMC.2018.00080.
13. Dey R, Salem FM. Gate-variants of Gated Recurrent Unit (GRU) neural networks, In: 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), Boston, MA, USA; 2017. p. 1597–1600. https://doi.org/10.1109/MWSCAS.2017.8053243.
14. Dey R, Salem FM. Gate-variants of gated recurrent unit (GRU) neural networks. 2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS); 2017. Boston. p. 1597–600. https://doi.org/10.1109/MWSCAS.2017.8053243.
15. Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput. 1997;9:1735–80. https://doi.org/10.1162/neco.1997.9.8.1735.
16. Sherstinsky A. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. Phys D Nonlinear Phenom. 2020;404: 132306. https://doi.org/10.1016/j.physd.2019.132306.
17. Hossein A, Mostafa S, Mohsen Y. An optimized model using LSTM network for demand forecasting. Comput Ind Eng. 2020;143: 106435. https://doi.org/10.1016/j.cie.2020.106435. (**ISSN 0360-8352**).
18. Liu Y, Zhou Y, Wen S, Tang C. A strategy on selecting performance metrics for classifier evaluation. Int J Mob Comput Multimed Commun. 2014;6:20–35. https://doi.org/10.4018/IJMCMC.2014100102.
19. Awoyemi JO, Adetunmbi AO, Oluwadare SA. Credit card fraud detection using machine learning techniques: a comparative analysis. 2017 international conference on computing networking and informatics (ICCNI); 2017. Lagos, Nigeria. p. 1–9. https://doi.org/10.1109/ICCNI.2017.8123782.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.