Original Research Article

Credit Card Fraud Detection Using Autoencoder Model in Unbalanced Datasets

Abstract

Fraudulent credit card transaction is still one of problems that face the companies and banks sectors; it causes them to lose billions of dollars every year. The design of efficient algorithm is one of the most important challenges in this area. This paper aims to suggest an efficient approach that detects fraud credit card related to insurance companies using a new method based on neural networks of type Autoencoder. The effectiveness of the proposed method has been proved in identifying fraud in actual data from a credit card-related dataset. In addition, a solution for data unbalancing is provided in this article, which affects most current algorithms. The suggested solution relies on training for the autoencoder for the reconstruction normal data. Anomalies are detected by defining a reconstruction error threshold and considering the cases with a superior threshold as anomalies. The algorithm's performance was evaluated using several metrics as well as comparing it with the logistic regression algorithm.

Keywords: Autoencoder; Fraudulent Credit Card; Machine Learning; Logistic Regression.

1 Introduction

The Association for Payment Clearing Services (APACS) has estimated that total losses through credit card fraud in the United Kingdom have been growing rapidly from £122 million in 1997 to £440.3 million in 2010 [1]. According to the Nelson report [2], the losses on global credit and prepaid cards reached $ 24.71 billion in 2016, up 11.2 percent from 2015. Gross fraud losses are absorbed by card issuers and merchants as well as by acquirers of transaction from ATMs and Merchant. A central feature of the report, the LexisNexis Fraud Multiplier[[3], estimates the total amount of loss a merchant incurs, based on the actual dollar value of a fraudulent transaction. , According to the Fraud Multiplier tool, In 2016, every dollar of fraud cost merchants $2.40, up from $2.23 a year ago. Also, the report finds that the volume of fraud raised sharply in the last year, from a monthly average of 156 to 206 successful fraudulent transactions, and from 177 to 236 prevented fraudulent transactions, while the level of fraud as a percentage of revenues also inched upward from 1.32 percent to 1.47 percent. Cases of financial and banking fraud in the Kingdom Saudi Arabia have halved in 2017 to 2,046 compared to 4,275 cases a year earlier. Financial fraud has amounted to SAR 214 million last year versus SAR 520 million worth of fraudulent activities in 2016[4]. Financial institutions in the present situation are exposed to many risks; the most important of which is the problem of fraud, especially with the advancement of modern technologies such as the Internet and computers [5] where fraudsters are developing their methods of obtaining illegal economic benefits, which needs fraud detection techniques capable of getting improved as rapidly as possible. Financial fraud is an issue that has wide reaching consequences in both the finance industry and daily life. Fraud can reduce confidence in industry, destabilize economies, and affect people's cost of living. Jarrod West et al. [6] defined the financial fraud as the intentional use of illegal methods or practices for the purpose of obtaining financial gain. Financial losses resulting from fraud on traders and financial institutions, such as unpaid amounts or non-financial losses, can affect the loss of institutions to customers. Although it is difficult to identify them in the short term, they become clear in the long term. The electronic disclosure of financial fraud can be said to be the use of computer systems to determine whether a new licensed transaction belongs to the category of fraudulent or

legitimate transactions. Fraud Detection System (FDS) should not only be effective, but should also be cost-effective. FDS receives the card details and the value of purchase to verify whether the transaction is genuine or not. Bhatla [7] maintained that examining 2% of the transaction may result in reducing fraud losses by 1 % of the actual transaction value, but fraud detection costs will increase. To minimize costs, expert rules and models based on machine learning are used to conduct the firstly examination between fraudulent and legitimate transactions and to require investigators to review high-risk cases only.
Transactions are first filtered by checking certain basic conditions (secure code, card number, expiration date etc.) and then recorded by a predictive model, urging that a predictive model can be formed based on expert rules only. These rules require manual control and human supervision. With techniques machine learning (ML) we can detect fraudulent patterns efficiently and impact transactions that are likely to be fraudulent. The machine learning (ML) techniques are the conclusion of a prediction model based on a set of pre-defined examples. In most cases, this model is a parametric function which allows predicting the probability that the transaction will be fraudulent.

This research aims to identify modern technologies detection of fraudulent credit card transactions and identify techniques that rely on machine learning and continuously update the models based on new data and the detection of new cases efficiently.
Objectives the research

1- Briefly introduce previous algorithms, used to detect fraudulent credit card transactions depends in machine learning.
2- Adopt a new model for detecting financial fraud using deep Learning Algorithm called Autoencoders
3- Present detailed experimental results to show the effectiveness of our approach and compare it with other techniques available in the literature.

The first section provides a brief review about the various studies conducted on the detection of credit card fraud, which enables us to determine the most effective technique. The next section deals with the need to detect credit card fraud and also provides an overview of the autoencoder model as a solution. Finally draws conclusion on the basis of different analyses and the results obtained.

2 Literature Review

Neural networks [8] [9] [10] and logistic regression [11] [12] are often chosen for their well-established popularity, giving them the ability to be used as a control method by which other techniques are tested. Comparatively, more advanced methods such as support vector machines and genetic programming have received substantially less attention [6]. Most studies have focused on the use of machine learning methods for supervised learning and unsupervised. However, recent studies indicate a trend towards using hybrid methods of the former two types to combine their advantages.
This section discusses a number of methods used in fraud detection, the most important traditional methods such as algorithms for optimization and machine learning. This section will focus on machine learning methods, as they are considered the most widely used methods so far.

- ***Machine learning Methods***

The machine learning methods are divided into groups: supervised learning methods and unsupervised learning methods**.** In this paper, the researcher attempts to examine the use of machine learning methods in the classification between fraudulent transactions and legitimate transactions. However, classification is located algorithms within the field of machine learning supervision, so the study will focus only on research that fall within this area.

- **Supervised Learning Methods**

Some techniques of machine learning treat transaction fraud as a problem of supervised classification. In this manner, together with annotations, we can train a classifier based on training data, then classify test transaction data into normal and abnormal classifications. A systematic review of 49 papers in the same field showed that decision trees, neural networks, logistic regression and SVM were the preferred methods among many other methods [13]. Bhattacharyya [11] compared between the accuracy of logistic regression and random forest and SVM on real data, which contain varying percentage of fraud in training groups, random forest showed high precision versus low rates of recall. Some studies have dealt with the problem of

unbalanced data, which is one of the most important problems facing algorithms classification, using several methods: Over-sampling, Under-Sampling and Synthetic Minority Over-sampling Technique.
Over-sampling refers to the process of increasing the number of records in the minority class, but increasing the number of records leads to minority class bias and increase the size of the training set and also, increase training time and the amount of memory required to hold the training set, it is not efficient In the case of big data. Under-Sampling refers to the process of decreasing the number of records in the majority class.
As a result, the overall number of records in the training set is greatly reduced. This means that during classification, training time is also greatly reduced. It is possible that we will lose a lot of valuable information if we eliminate documents that could be useful to our classifier in building an accurate model [14]. Synthetic Minority Oversampling, oversamples the minority class by generating synthetic examples in the neighborhood of observed ones. The idea is to form new minority examples by interpolating between samples of the same class [14]. Studies have proven that the combination of these two technologies has great effectiveness in achieving the balance of data. Successfully applied these techniques to the problem of detecting fraud in the credit card, however, Fraud detection algorithms need to know that the conditional balance of a class may change over time [15]. The methods of detecting anomalies take a different perspective, the model is constructed for legitimate instances and the transaction is then evaluated as anomalies not in accordance with this model. These methods require that the Score be determined for anomalies that determine the extent to which a situation is abnormal [16].
Fraud Credit card is not restricted to transactions only, but to transactions and the features in which they occur. In recent years, interest in features has been observed by studying several factors such as the date of purchase, the record of the customer activity, that enable the classifier to better identify fraudulent transactions. So this paper will review two strategies of study literature that allow context description.

- **Feature engineering for Temporal Sequences**

Choosing features when creating fraud credit card is critical to accurate classification. It is not surprising that great research efforts are devoted to the development of expressive features. However, as noted in [17] a single transaction information is not sufficient to detect a fraudulent transaction, since using only the raw features leaves behind important information such as the consumer spending behavior, which is usually used by commercial fraud detection systems. Detect traditional fraud system features as inputs for the training of binary systems works adopted as it deals with the treatment level of total disregard of the fact that the frequency and size of transactions at certain time intervals can carry valuable information for each individual account. Credit card data is represented as a graph. The node is the cardholder or the merchant while the edges are transactions between the nodes. The weight of the edges is determined by the size of transactions between these entities and decreases over time.
The graph extracts network features that measure the extent to which each entity is exposed to a fraud. These features include a score for the cardholder, the merchant and the transaction grouped at short, medium and long intervals [18].

- ***Sequence Classification***

Sequential learning is the study of learning algorithms for sequential data. These methods include sliding window methods, recurrent sliding windows or conditional random fields. While sliding window based approaches tend to ignore the sequential relationship between data points inside the windows, a better solution is to resort to model-based approaches that assume explicitly a sequential dependency between consecutive data points. In its simplest form such model could be a Markov chain defined on the data points [19]. However, the sequential dependence is presumably more evident or useful in many practical applications as a sequence of latent, so-called hidden, states that control the sequence of observed data points [19]. Recurrent neural network is represents a hidden identity of the family of non-probability models. The Recurrent neural network is trained to periodically identify fraudulent transactions given the sequence of transactions in the past. Long Short-Term Memory network (LSTM) has recently raised a lot of attention because of its ability to learn long-term dependency. It constitutes the state of the art on many real world tasks such as speech recognition, hand writing recognition and statistical machine translation [19]. Fraud detection ways recently started the trend towards the use of a hybrid approach by integrating more than algorithm to take advantage of the features of each, such us: Latent Dirichlet Allocation(LDA) ,to analyze text data within a set of documents that represent traffic accident reports and extract text features using natural language processing techniques, such as (the color of the car - the type of car - the description of the incident) and then used the advantages extracted to train the deep neural network to detect fraud within a

range of textual claims submitted to insurance companies. The accuracy of neural networks in classifying claims has increased significantly as a result of the use of natural languages [20]. Neural networks were integrated with genetic algorithm to detect credit card fraud. A neural network of back propagation and its components was used from several layers and the genetic algorithm was introduced to decide the structure of the network, the network topology, the number of hidden layers and the number of nodes in each layer [21]. Used algorithms decision trees and Supper Vector Machines (SVM), respectively, to build a classification model for fraud detection within the real data of credit cards class, results showed that decision tree approaches outperform SVM approaches in solving the problem [22].

3 Experimental Setup

In this section, first the dataset used for the experiments is d scribed. Afterwards, the partitioning of the dataset is presented. Lastly, the algorithms used to detect fraud are shown.

- ***Dataset***

The datasets contain transactions made by credit cards in September 2013 by European cardholders. This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions [23]. The dataset does not need any data cleaning process; it does not contain duplicate entries, Huge Outliers, and Null values.

Autoencoder Algorithm

Was developed algorithm depend on neural networks type autoencoder and evaluate their performance and ensure its ability to detect fraud cases as appropriate. Several measures were used:

- **Reconstruction Error**

The Mean squared error is used to calculate the value of the reconstruction error

$$\text{MSE}= \frac{1}{n} \sum_{i=1}^{n} (x_{i-\widehat{x_i}})^2 \qquad (1)$$

Where, n: size of the input and output , x: input data $\hat{x}$: output data of the reconstruction.

The high error value indicates the discovery of fraudulent transactions while the low value reveals legitimate transactions.

- **Precision & Recall**

Precision and Recall are one of the most widely used standards in unbalanced data, which reflects the precision of the suitability of the result scale and proximity to the expected solution, while recall measure of the number of relevant results returned, the goal in each of them to approach the one. High score recall indicates a low False Negative (FN) rate, while high precision indicates a low False Positive (FP) rate. High Scores for both show that the classifier restores accurate results in addition to the recovery of the majority of the positive results [24].

- **Confusion Matrix**

The confusion matrix is used to describe the performance of the proposed classification model for selecting a data set and is the form of 4 different sets of expected real values, where the confusion matrix provides the number of transactions per set.

In the following table I and table II [25] we provide an overview of performance measures based on the confusion matrix:

Table I. Confusion Matrix in Credit card fraud

	Predicted genuine(0)	**Predicted fraud(1)**
Actual genuine (0)	TN – true negative	FP – false positive
Actual fraud (1)	FN – false negative	TP – true positive

Table II. Classification performance measures

Measure	**Definition**
Sensitivity	TP/(TP + FN)

(Recall)	
Precision	TP/(TP + FP)
F-measure	2 * Precision * Recall / (Precision + Recall)
Accuracy	(TP+TN)/(TP+TN+FP+FN)
F1 Score	2*(Precision-Recall)/(Precision +Recall)

However, these measures may not be the most appropriate evaluation criteria when evaluating fraud detection models, because they tacitly assume that misclassification errors carry the same cost, similarly with the correct classified transactions.

4 Autoencoder Classifier

Autoencoder learn is a unsupervised learning seeking to be output corresponding to their income and therefore can be considered the network as a supervised learning , the output $\hat{x}$ is the result of reconstruction the original income x. An autoencoder learns to map from input to output through a pair of encoding and decoding phases. The encoder maps from the input to hidden layer, the decoder maps from the hidden layers to the output layer to reconstruct the inputs. Hidden layers of the autoencoder are low dimensional and nonlinear representation of the input data [26].

There is a bottleneck issue in the autoencoder. A bottleneck constrains the amount of information that can traverse the full network, forcing a learned compression of the input data [27]. Figure 1 shows the autoencoder with the hidden layer.

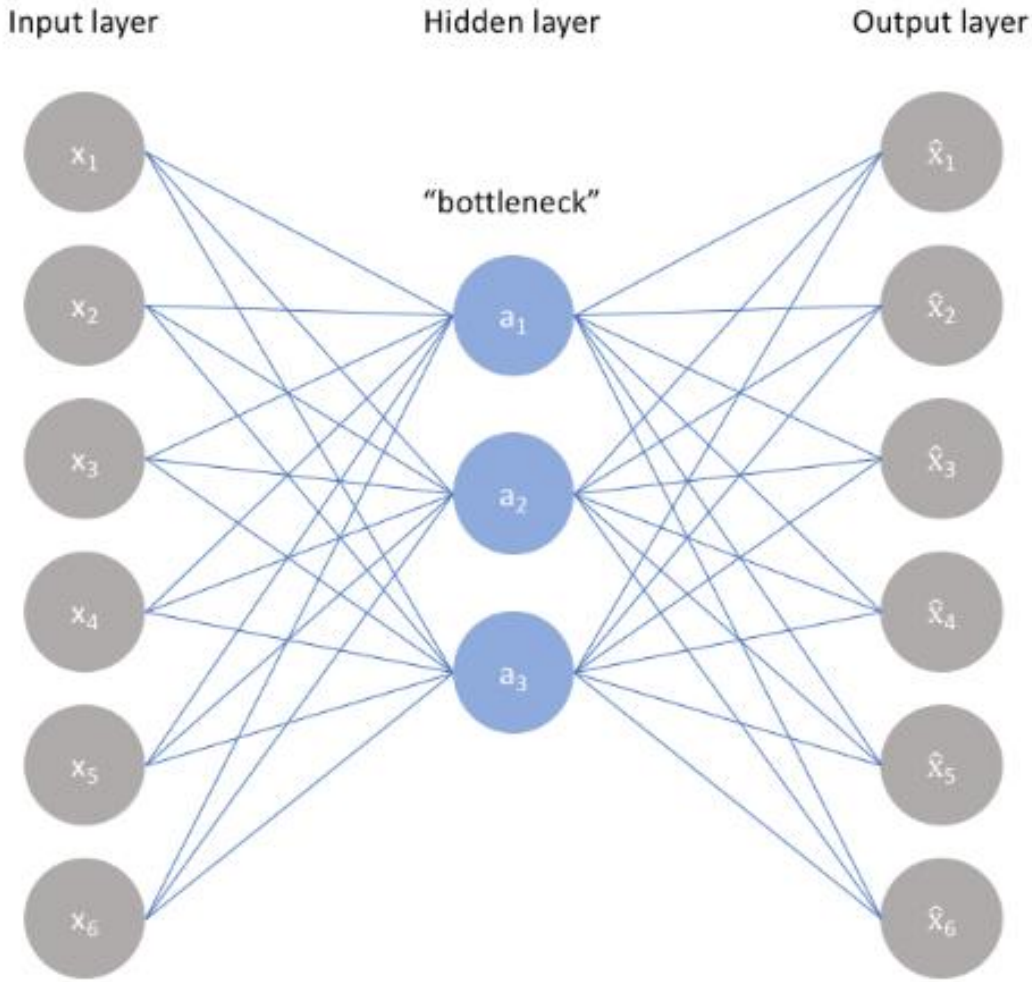

Fig. 1. Autoencoder with Hidden Layers

▪ Architecture Neural Network

The network architecture for autoencoders can vary between a simple Feedforward network, LSTM network or Convolutional Neural Network depending on the use case. In this case the Feedforward network will be used.

Autoencoders architecture consists of four main parts:

- Encoder: it is the part in which the model learns how to reduce the input dimensions and compress the input data into an encoded representation.
- Bottleneck: it is the layer that contains the compressed representation of the input data. This is the lowest possible dimensions of the input data.
- Decoder: it is the model that learns how to reconstruct the data from the encoded representation to be as close to the original input as possible.
- Reconstruction Loss: this is the method that measures measure how well the decoder is performing and how close the output is to the original input.

The training then involves using back propagation in order to minimize the network's reconstruction loss.

There are four hyperparameters that are required before setting out training an autoencoder:

1. code size: when the number of nodes in middle layer is small then the great pressure.
2. Number of layers: flexible number of layers (depth of layers).
3. Number of nodes per layer: the number of nodes in each layer decreases after the encoder, and is increased again in the decoder, and the number of nodes can be selected in each layer according to need.
4. Loss Function : the error resulting from the reconstruction of the input data in the output layer, and the Mean square error is used to calculate the error value, such as equation 2 below:

$$l(x,\hat{x}) = \frac{1}{n}\sum_{i=1}^{n}(x_{i-\hat{x}_i})^2 \qquad (2)$$

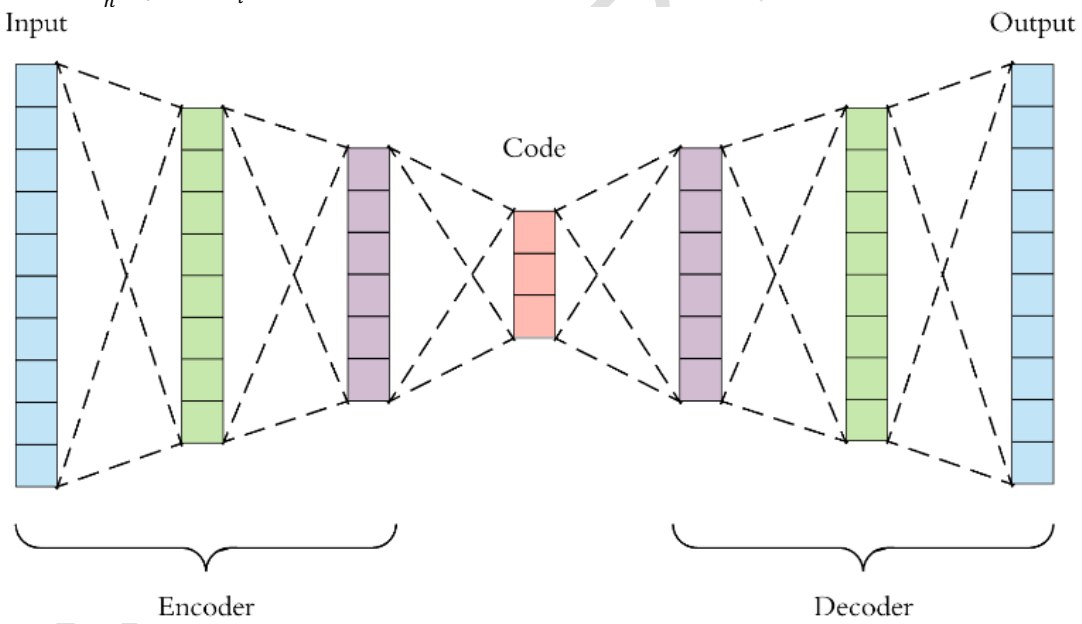

Fig. 2. Autoencoder Architecture

▪ Autoencoder Pseudo-coding

The following steps in table III, show the Pseudo code for the autoencoder algorithm.

Table III. Autoencoder Pseudo code

Step 1: Prepare	Input Matrix X // input dataset Parameter of the matrix // parameter (w,bx , bh)

the input date	where: w : Weight between layers, bx Encoder's parameters , bh Decoder's Parameters
Step 2: initial Variables	h ← null // vector for hidden layer X ← null // Reconstructed x L ← null // vector for Loss Function l ← batch number i ← 0

Step 3: loop statement	While i < l do
	// Encoder function maps an input X to hidden representation h:
	$h = f(p[i].w + p[i].b_x)$
	/* Decoder function maps hidden representation h back to a
	Reconstruction X :*/
	$X = g(p[i].w^T + p[i].b_x)$
	/*For nonlinear reconstruction, the reconstruction loss is generally
	from cross-entropy :*/
	$L = -sum(x * \log(X) + (1 - x) * \log(1 - X))$
	/* For linear reconstruction, the reconstruction loss is generally
	from the squared error:*/
	$L = sum(X - X)^2$
	$\theta[i] = \min_p L(X - \hat{X})$
	End while
	Return θ

Step 4: output	θ ← <null matrix>//objective function /*Training an autoencoder involves finding parameters = (W,bx , bh) that minimize the reconstruction loss on the given dataset X and the objective function*/

5 Results and Discussion

- **Build the Proposal Model**

The application was built using the Python language based on a set of software libraries and the most important:

Karas: library provides simple and consistent software interfaces for communication with the end user, not the machine, and contains a set of models such as neural networks, decision trees and activation subsystems, as well as their scalability. The main task of the library is to make the application more responsive and give the user more power on the interface control.

TensorFlow: library is applied in many fields such as derivatives and large matrices as well as in the distribution of computer operations on central processing units(CPU) as well as on a distributed network consisting of a collection of remote devices including this library. Mainly used in machine learning at present.

- *Apply autoencoder algorithm*

The algorithm used for autoencoder has been applied in several stages:

1. The data were divided into 80% training data and 20% testing data based on the experiment. The training data contains only legitimate transactions, so that the network can form a compressed representation and distinguish it from fraudulent transactions.
2. Selecting the number and size of layers experimentally comes next. The following network was chosen experimentally from Left to Right 32-14-7-7-32 five Layers. The first layer represents the network input, while the second and third layers encode the data. In the fourth and fifth layer the data is reconstructed, and the Loss Function is calculated. the most important that the input layer is equal to the output layer in terms of the number of neurons.

Network training stops when it becomes reconstruction error as less possible. The network input is approximately equal to the output. So after experimenting number of epochs, in epoch No. 51, we note that the reconstruction error to less as possible, and that the network reached enough to reduce redundancy. The following figure illustrates the decrease in the value of loss from reconstruction as the number of repeat increases.

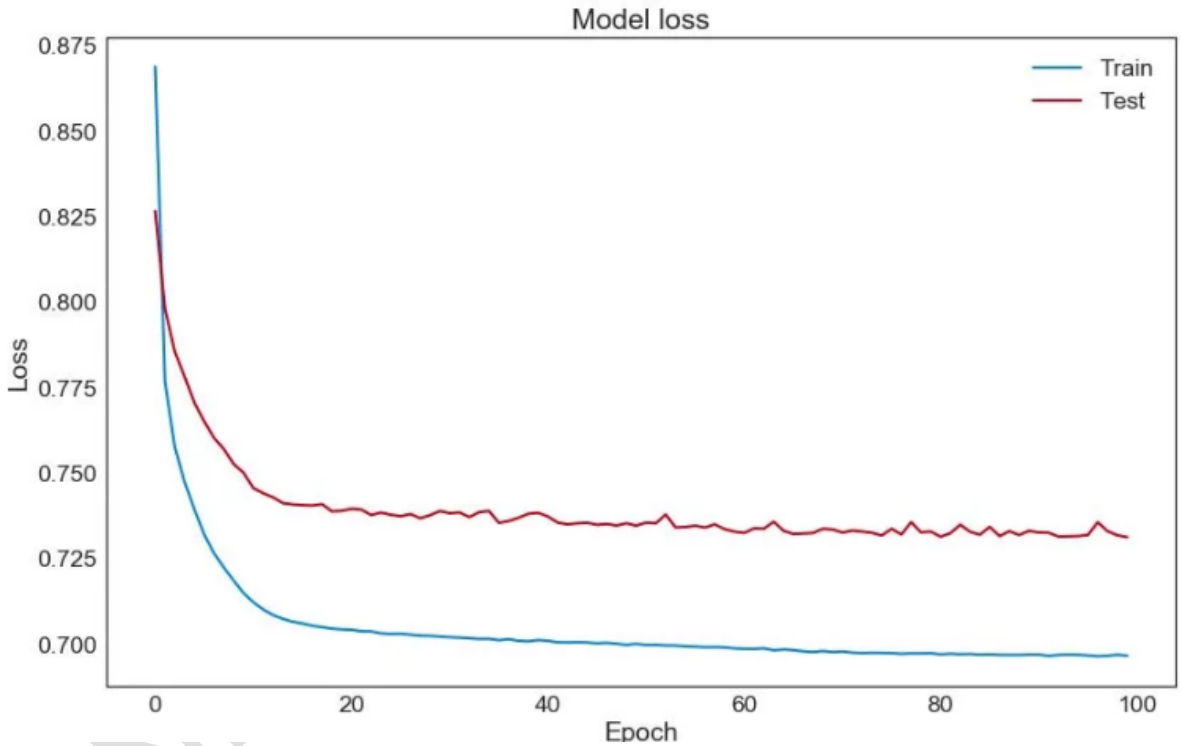

Fig.3. Reconstruction error Vs. Epochs

- ***Evaluate algorithm performance***

- Reconstruction Error

In the measuring of Reconstruction Error, figures 4 and 5 illustrate the value of the reconstruction error for both legitimate and fraudulent transactions, where the value of the error for legitimate transactions is very small, while its value is significant in the case of fraudulent transactions.

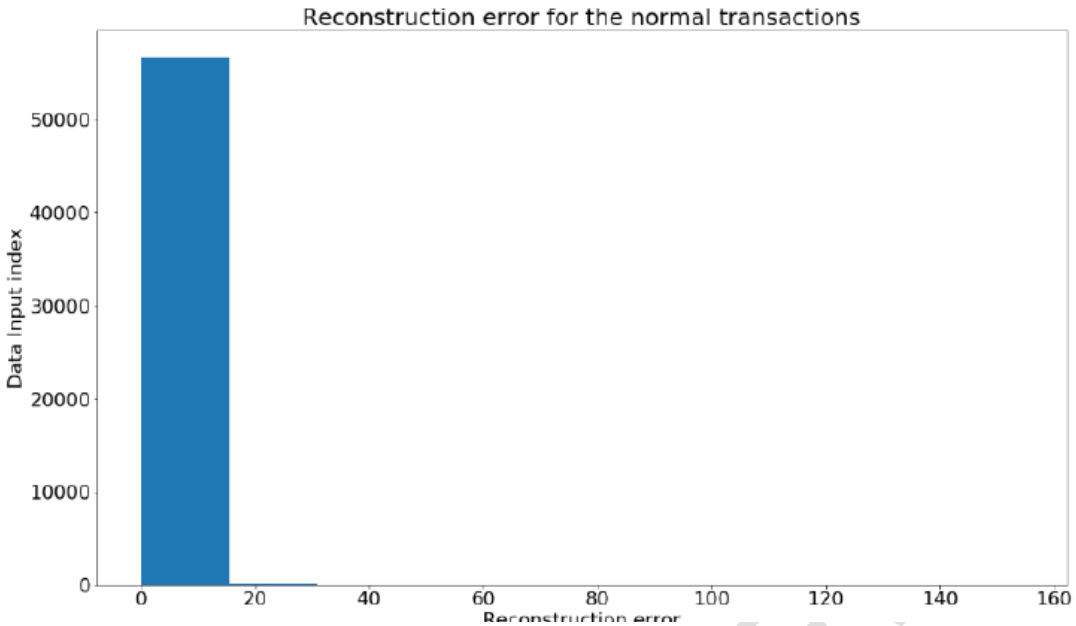

Fig.4. Reconstruction values for legitimate transactions

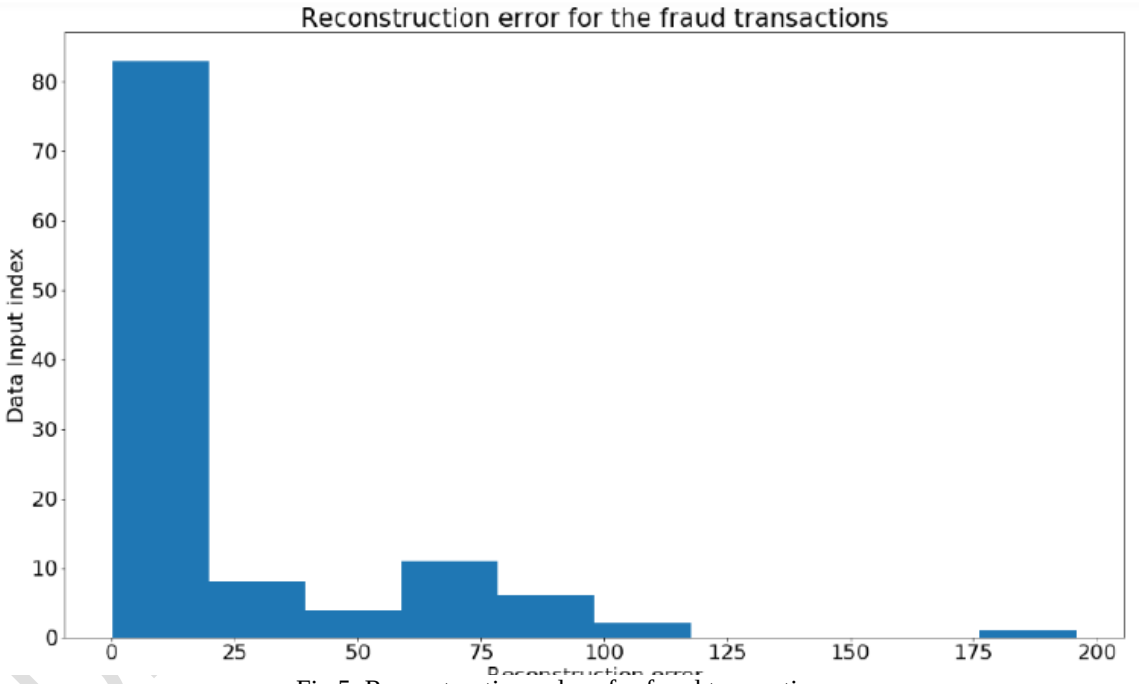

Fig.5. Reconstruction values for fraud transactions

- Precision & Recall

In the measuring of Precision and Recall, The following figure 6 shows the precision and recall values for different threshold values that represent the error of reconstruction and are used as a boundary between legitimate and fraudulent transactions within the credit card fraud detection model using the Autoencoder network.

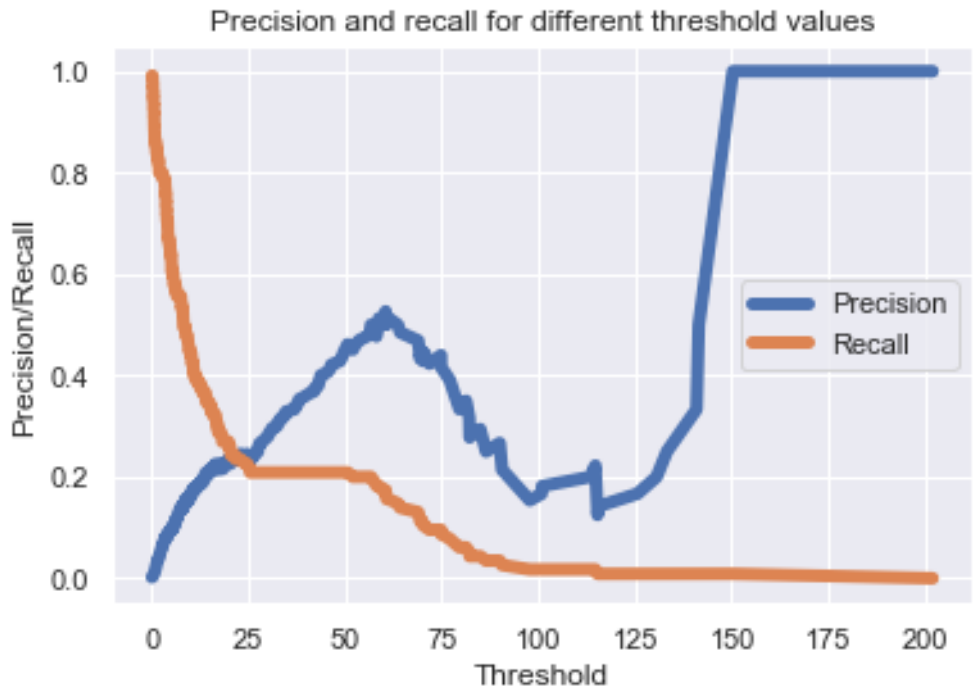

Fig.6. Precision and recall values with various values of the threshold

Figure 6, it shows that the higher the value of the threshold is, the higher the precision is, while the value of the recall decreases. For example, for the threshold value = 50, the precision value is = 0.4 while the value of the recall is = 0.2. Based on the differential shown in figure 6, the value of threshold 5 is chosen for the proposed model, and thus the data set is divided into two sub-groups. The first group contains a large majority of the data and the error of reconstruction is very small. Therefore, all transactions are considered legitimate. The second group contains a small percentage of the data with large values of the reconstruction error, and all transactions are considered fraudulent.

Consequently, all data point above the threshold represent fraudulent transactions, since this model must contain a low reconstruct error in legitimate transactions. The following figure 7 illustrates the classification of transactions using the threshold value = 5.

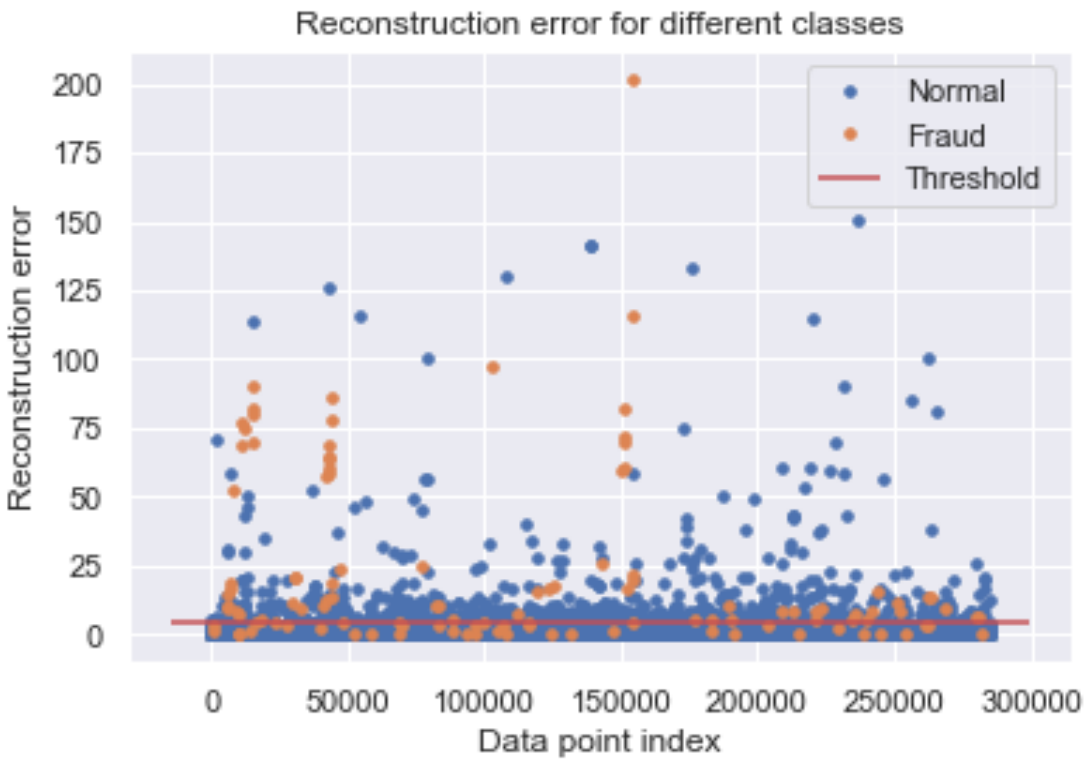

Fig.7. Data distribution in threshold 5

Figure 7 shows that most fraudulent transactions are properly classified with relatively few legitimate transactions classified as fraudulent. Confusion matrix confirms that.

Different threshold values can be selected based on the situation that appears. For example, if the problem of many false alarms can be ignored in exchange for more fraudulent transactions, then a low threshold value 3 may be chosen. However, the proportion of legitimate transactions would be classified as fraudulent. Figure 8 confirms this through the confusion matrix

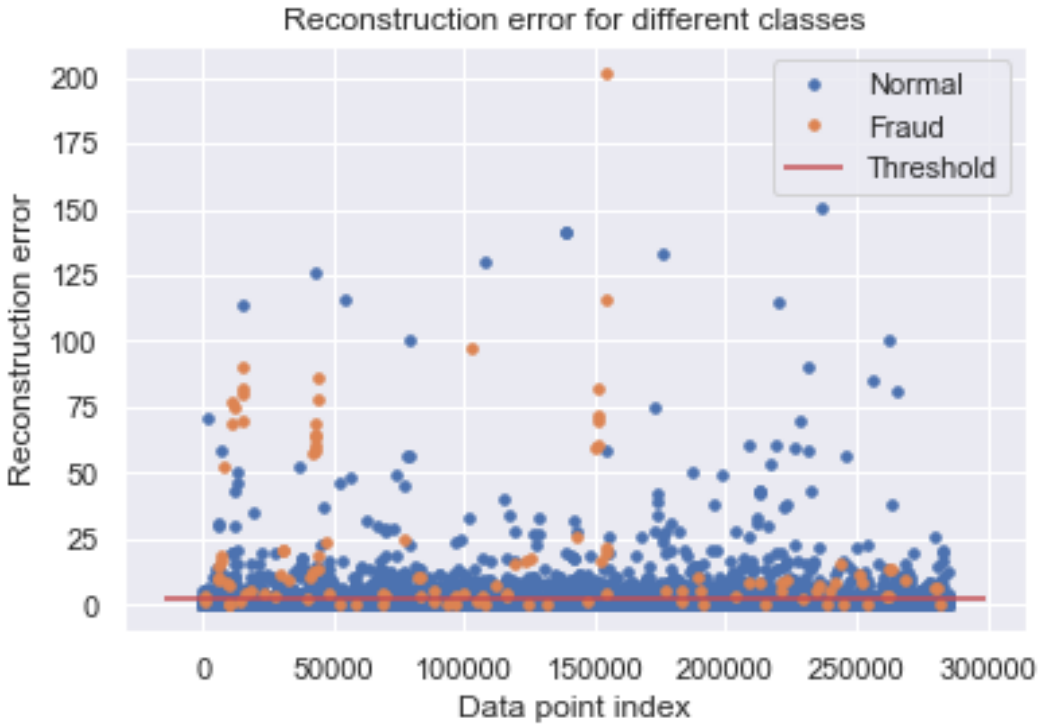

Fig.8. Data distribution in threshold 3

- **Confusion Matrix**

Finally, looking at the traditional confusion matrix for the 20% of the data with randomly held back in the testing set.

- When Threshold = 5

Table IV. Threshold=5

		Predicted Values	
		0	**1**
Actual Values	0	56150	697
	1	41	74

It shows a matrix of confusion Table IV, that the model with threshold =5 is able to control about 60% of cases of fraud.

$$hit\ rate = \frac{TP}{TP+FN} * 100 = \frac{74}{74+41} * 100 = 64\% \qquad (3)$$

While the proportion of legitimate transactions classified as fraudulent

$$False\ Positive\ rate = \frac{FP}{FP+TN} * 100 = \frac{697}{697+56150} * 100 = 1.2\%$$

- When Threshold = 3

Table V. Threshold=3

		Predicted Values	
		0	**1**
Actual Values	0	56552	1295
	1	24	91

It shows a matrix of confusion Table V, that the model with threshold =3 is able to control about 79% of cases of fraud.

$$hit\ rate = \frac{TP}{TP+FN} * 100 = \frac{91}{91+24} * 100 = 79\%$$

While the proportion of legitimate transactions classified as fraudulent

$$False\ Positive\ rate = \frac{FP}{FP+TN} * 100 = \frac{1295}{1295+56552} * 100 = 2.2\%$$

The confusion matrix shows that it has a significant role in determining what is required of the model. The lower values of the threshold reflect more fraudulent cases, but more false classifications of legitimate transactions as fraudulent. By choosing a high value threshold, there is a significant reduction in false notification for legitimate transactions and the discovery of fewer frauds. The discernment process is often subject to the decision of companies and financial institutions.

6 **Results Comparison**

The performance of the algorithm must be more closely compared with other algorithms used to classify data between fraudulent and non-fraudulent. Comparison with logistic regression has been made, because of its uses in classification.

The following table VL compares the Logistic Regression(LR) algorithm in the case of balanced data and unbalanced data with the Autoencoder network at several threshold values(Thr=5, 3,1 and 0.7)

Table VL. Comparison of LR and Autoencoder

	Accuracy	Recall	Precision	F1 Score
LR(balance Data)	97.23	0.90	0.06	0.12
LR(unbalance Data)	99.91	0.57	0.93	0.71
Autoencoder(Thr=5)	98.70	0.64	0.011	0.19
Autoencoder(Thr=3)	97.70	0.79	0.067	0.12
Autoencoder(Thr=1)	90.02	0.86	0.073	0.13
Autoencoder(Thr=0.7)	80.00	0.91	0.09	0.04

Table VL shows the superiority of the logistic regression in the case of balanced data on the state of the unbalanced data. Where the number of fraudulent transactions discovered is more important than the precision of the model if the fraud is discovered to reach its value in the case of balance of data 90% according to the value of the recall. There is also a slight superiority of the autoencoder network at the threshold of 0.7 on the logistic regression where the percentage of fraudulent transactions detected is 91%, on the other hand, the model suffers from more. False notification. The table VL also, shows the convergence of both algorithms at threshold 3. In the autoencoder network, the value of the threshold can be changed and reduced to show a high precision result in the fraudulent transaction detection, but the classifying legal transactions as fraudulent will increase.

For example, detect many fraudulent transactions? or reduce false warnings? And so on during the variation between the values of recall and accuracy, for example, note that the value of the accuracy exceeds the value of the recall at threshold 5, in contrast to the threshold at 0.7. The previous variation is not possible if the

logistic regression is used to build a fraud detection model. Autoencoder network does not need to use the data balance methods to achieve the model unlike logistic regression that needs to balance data before the construction of the model.

6 Conclusions and Recommendations

With the large and ongoing financial loss currently being experienced by financial companies, It was necessary to develop more efficient methods on which the electronic systems to detect fraudulent transactions, fraud detection is a very difficult and complex task. Fraudulent activities are rare events that are difficult to model, and the large volume of day-to-day transactions requires automated tools to support the science of fraud verification.

In this paper, some advanced techniques have been introduced to detect the financial fraud of the insurance company. This study reviewed how machine learning can be used to address some of the issues of financial fraud detection in credit cards. The focus, on the design model is capable of reporting the most fraud transactions for investigators using autoencoder algorithm way that can deal with unbalanced datasets. The algorithm was able to detect between 64% at the threshold = 5 and 79% at the threshold = 3 and 91% at threshold= 0.7.

The algorithm also provided a solution to avoid the problem of data balancing experienced by many of the algorithms currently used, which can be applied directly to data without the use of data balance methods such as the method of Under-Sampling.

The recommendation of the paper lies in the following suggestions for improvements to the current algorithm: Appling fraudulent work to different classification algorithms and compare them with this model; inserting a random value in an attempt to confuse the fraudsters and disrupt their previously acquired knowledge; and applying this algorithm to the data of Saudi companies and financial institutions.

Competing Interests

Authors have declared that no competing interests exist.

References

[1] L Delamaire, HAH Abdou and J Pointon"Credit card fraud and detection techniques: a review. Banks and Bank systems." Banks and Bank Systems 4, no. 2 (2009): 57-68.

[2] The Nilson report." URL http://www.nilsonreport.com, August,2016.

[3] LexisNexis. "True cost of fraud 2016 study." URL https://risk.lexisnexis.com/insights-resources/research/lexisnexis-2016-true-cost-of-fraud.

[4] Argaam report. September 2018. URL https://www.argaam.com/en/article/articledetail/id/570536.

[5] I.C Yeh and C. Lien, "The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients." Expert Systems with Applications 36, no. 2 (2009): 2473-2480.

[6] J. West, M. Bhattacharya, & R. Islam, "Intelligent financial fraud detection practices: an investigation." In International Conference on Security and Privacy in Communication Systems. Cham: Springer, 2014. 186-203.

[7] P. T. Bhatla, P. Vikram, and D. Amit, "Understanding credit card frauds." Cards business review 1, no. 6 (2003).

[8] I. Bose, and J. Wang. "Data mining for detection of financial statement fraud in Chinese Companies." In International joint Conference on e-Commerce, e-Administration, e-Society, and e-Education. International Business Academics Consortium (IBAC) and Knowledge Association of Taiwan (KAT). Taiwan, 2007.

[9] E Kirkos, C Spathis, Y Manolopoulos, "Data mining techniques for the detection of fraudulent financial statements." Expert systems with applications 32, no. 4 (2007): 995-1003.

[10] P Ravisankar, V Ravi, GR Rao, I Bose, "Detection of financial statement fraud and feature selection using data mining techniques." Decision Support Systems 50, no. 2 (2011): 491-500.

[11] S Bhattacharyya, S Jha, K Tharakunnel, "Data mining for credit card fraud: A comparative study." Decision Support Systems 50, no. 3 (2011): 602-613.
[12] J. Pinquet, M. Ayuso, and M. Guillen,"Selection bias and auditing policies for insurance claims." Journal of Risk and Insurance 74 (2007): 425-40.
[13] EWT Ngai, Y Hu, YH Wong, Y Chen, X Sun,"The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature." Decision support systems 50, no. 3 (2011): 559-569.
[14] N. V. Chawla, K. W. Bowyer, L. O Hall, and K. P. W. Philip, "SMOTE: synthetic minority over-sampling technique." Journal of artificial intelligence research 16 (2002): 321-357.
[15] A Dal Pozzolo, "Adaptive machine learning for credit card fraud detection." 2015.
[16] X Zhao, J Zhang, X Qin, "LOMA: A local outlier mining algorithm based on attribute relevance analysis." Expert Systems with Applications, no. 84 (2017): 272-280.
[17] A.C Bahnsen, D. Aouada, A. Stojanovic and B,Ottersten, "Feature engineering strategies for credit card fraud detection." Expert Systems with Applications 51 (2016): 134-142.
[18] V Van Vlasselaer, C Bravo, O Caelen,T. Eliassi-Rad,L. Akoglu,M. Snoeck,and B. Baesens, "APATE: A novel approach for automated credit card transaction fraud detection using network-based extensions." Decision Support Systems 75 (2015)): 38-48.
[19] A. Graves, "Supervised sequence labelling." In Supervised sequence labelling with recurrent neural network. Berlin, Heidelberg: Springer, 2012. 5-13.
[20] Y Wang and W Xu. "Leveraging deep learning with LDA-based text analytics to detect automobile insurance fraud." Decision Support Systems 105 (2018): 87-95.
[21] R. Patidar and L. Sharma, "Credit card fraud detection using neural network." International Journal of Soft Computing and Engineering (IJSCE) 10 (2011): 32-38.
[22] YG Şahin and E Duman. "Detecting credit card fraud by decision trees and support vector machines." proceeding of international multi-conference of engineering and computer statistics. 2011.
[23] "Download the credit card fraud dataset." URL https://www.kaggle.com/mlg-ulb/creditcardfraud/data.
[24] D. Tasche, "A plug-in approach to maximising precision at the top and recall at the top." arXiv preprint arXiv:1804.03077, 2018.
[25] I Mekterović, L Brkić and M Baranović, "A Systematic Review of Data Mining Approaches to Credit Card Fraud Detection." WSEAS Transactions on Business and Economics 15 (2018): 437.
[26] X. Niu,l. Wang, and X. Yang, "A Comparison Study of Credit Card Fraud Detection: Supervised versus Unsupervised." arXiv preprint arXiv:1904.10604, 2019.
[27] S Yu, R Jenssen, JC Principe and J.C Principe, "Understanding convolutional neural network training with information theory." arXiv preprint arXiv:1804.06537, 2018.