Deep Learning Enabled Fraud Detection in Credit Card Transactions

Reshma R S#

#*Research Scholar, Department of Mechanical Engineering*
APJ Abdul Kalam Technological University, College of Engineering Trivandrum, Kerala, India

***Abstract*-Fraudulent activities exist in all fields of financial domain; credit card is not an exception. The increased fraud in credit card transaction is obviously due to its wide spread popularity since the introduction of online banking and e-commerce platforms. The purpose of the fraud may be to obtain goods without paying, or to obtain unauthorized funds from an account. We cannot rely on pattern based fraud detection since the fraudster rarely follows any pattern. As technology advances the fraudulent activities will also increase. The project aims to introduce deep learning technique to detect fraud in credit card transactions. Deep learning consists of neural networks with many hidden layers. The project uses deep learning techniques such as Autoencoder, Restricted Boltzmann Machine, Variational Autoencoder and Deep Belief Network to detect fraud in the credit card transactions. The aim is to find the best technique among them. Unsupervised deep learning method is used to evaluate fraud since the model learn from data and not from labels.**

***Keywords:* Unsupervised Learning, Deep Learning, Autoencoder, Variational Autoencoder, Restricted Boltzmann Machine, Deep belief network**

I. INTRODUCTION

Technology is advancing day by day and this improvement in technology has significant influence on various areas including financial transactions. Nowadays credit card is the most widely used transaction method, this increased popularity of credit card is mainly due to the ease of transaction, the increasing popularity of network banking and mobile banking. Since technology is a two sided coin, there is also a great improvement in the type of fraud that can occur. Fraudster keeps on changing his fraud pattern and rarely uses same pattern for fraudulent activities. This shows us that simple fraud detection method based on labelled data cannot work well with all type of fraud. Its high time we should think about unsupervised learning method where we are not bounded to the limitations of labelled data. The paper aims to develop unsupervised deep learning models to detect fraud in credit card transactions. Credit card fraud increases day by day. The way by which fraud occur rarely follows pattern. Unsupervised deep learning method can be used to detect fraud in financial transactions. The paper aims to develop four unsupervised deep learning methods to detect anomaly in credit cards. The data belongs to a European credit card data which consist of about 284,807 transactions out of which only few are fraud. The aim is to detect the fraudulent transactions and find out the area under receiver operation characteristic (AUC). One with higher AUC value is considered to be a better model.

A. Literature Review

[1] discussed credit card fraud detection using autoencoder and restricted boltzmann machine. In this work three credit card fraud data, German , European and Australian data were used . The models were built and roc were plotted. It is shown that the European credit card data shows greater roc curve compared to other, both on autoencoder and restricted Boltzmann machine model. This is mainly due to the huge number of data points available for this model. Hence for this work European credit card data is used. The work also shows that the tanh function work well with autoencoder based model so tanh based function is used as the activation function both in hidden layer, encoder and decoder section to improve the accuracy of the function. The data is divided in the ratio 80: 20 since the amount of fraud is much lesser so this method is adapted to this work too. The work shows that deep learning is the most efficient method for fraud detection. H2O framework was used by the model to understand the pattern in the underlying dataset. H2O is an open source available for analysing big data. Multi-layered, feed forward neural network is used by the author to find patterns in credit card fraud. [2] Discussed the comparison of available techniques for fraud detection and their demerits. The paper points out that simple logistic regression can be using in case of linear data and it will fail if the data is non linear. In case of decision tree even small changes in the data can affect the structure of the tree. Choosing splitting criteria is another demerit since its selection is too complex. It cannot handle real time data too. From [1] the project adopts the following features that are mentioned below. The project aims to improve the fraud detection model mentioned in the work by adding variational autoencoder and deep belief network to it. The tanh activation function is used in both hidden and visible layers for getting better AUC value. The method of finding AUC for determining efficiency of the model is also adopted from the work. Above all from the three dataset analysed in the work the project choose European credit card fraud data since the work proves that this is the best data set available for fraud detection with more data points. The only disadvantage of the data is that the data is PCA transformed with unknown parameters so the attributes that makes the transaction fraud is still unknown and we cannot validate the model using other data since we have to mask the data with same parameter

which is unknown. The advantage of the dataset is that since the data set is already PCA transformed we don’t need to mask the data. The dataset consist of about 2 lakh transactions out of which only few are fraud, this is equivalent to the real case problems where the fraud cases are fewer than the actual cases.

B. Data and Methodology

The datasets is credit card fraud data from a European credit card company, it is obtained from kaggle. The data contains transactions made by credit cards holder in 2013 September. Transactions that had happened in two days is shown in the dataset, It is given that the data contains only 492 frauds out of 284,807 transactions which accounts only 0.172 % of all transactions. All input variables are converted to numerical value by PCA transformation due to confidentiality issues, the original features are not provided and Features V1, V2, ... V28 are the principal components obtained after PCA transformation , the features that are not PCA transformed are ’Time’ and ’Amount’. Time represents the difference in seconds between the particular transaction and first transaction. ’Amount’ represents the money transaction that had occurred, Feature ’Class’ is a variable that shows whether a transaction is fraudulent or not value 1 shows that the transaction is fraud else it is non fraudulent. The fig 3.8 shows the distribution of data on the data set. From this we could understand that the fraud data is much lesser than the non fraudulent data. When time is plotted against transaction it doesn’t show any abnormality so it is proven that we can do nothing with the time and only the data matters. So we should develop a model that should learn from the data. The screenshot of a part of the data is shown below.

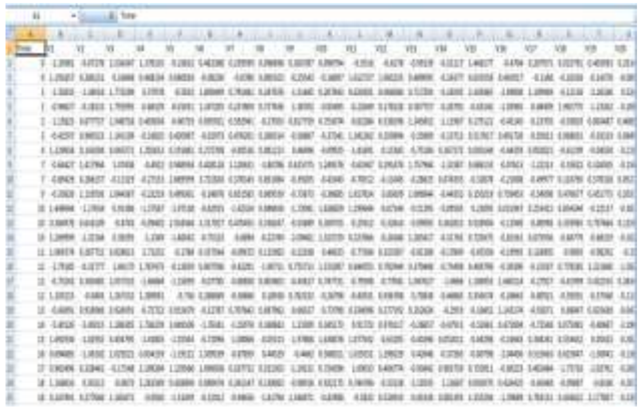

Figure 1. Dataset

C. Models Developed

1. Autoencoder Model (AE)

Autoencoder (AE) tries to approximate the following identity function:

$$F(x) \approx X$$

AE is that type of neural network which is teached to produce the input at the output layer. Like any other neural network it also contains an input, hidden and output layer. The aim of each AE model is to reproduce the input at its output with minimal or zero error. During each step, reconstruction error is measured and a threshold for the reconstruction error is set so that the model learns to reproduce with minimal error.

An autoencoder consist of encoders, lots of hidden layer and decoders. The input is compressed when it traverse through the hidden layer, where the code which helps in the regeneration of input at the output layer lies. The autoencoder model in this project is created using tanh fuction. Let x be the input of the function, h the hidden layer and x’ be the output function. The model created includes 4 layers, two encoders and two decoders. The input to the input layer is the input dimension of the data set. The activation function is represented by a(x). In the model developed, the activation function used in tanh. The equation for the encoder layer is given below.

$$i(x)\ (\text{Encoder}) = f(a(x)) = \tanh(Wx + b)$$

The input is given to the encoder model where it is multiplied by the bias function It is then compressed in the hidden layer. The input is reconstructed from its reduced form by using the transpose of the activation function in the hidden layer and decoder layers. By reducing the mean square error the output model is optimised ie, The model contains 4 layers which contains neuron numbers 29, 7,7, 29. One of the peculiarities of the model is that autoencoder wants the neurons in the input and output layers to be equal. The first two layers are the encoders and the last two layers are the decoders. The model is trained for 100 epochs and batch size of 32 and save the best performing model to a file. The Model Checkpoint in Keras is used for this task .

The accuracy of the model is determined by measuring the AUC score of the model. ROC curves are very useful tool for understanding the performance of unbalanced data. The true positive rate Vs false positive rate over different threshold values is plotted using ROC. The line should be as closer as possible to the upper left corner which defines high accuracy. The ROC curve with high area denotes high precision which is a sign of low false positive rate, and high recall is related to a low false negative rate. High score in AUC can be obtained if the model generates best output ie, detect all fraud cases without any failure.

2. Restricted Boltzmann Machine (RBM)

Restricted Boltzmann Machine Restricted Boltzmann machine can be used in regression, dimensionality reduction, classification, etc. RBMs have two layers of neural networks that is used as layers of DBMs. RBM have an input layer which is also called visible layer. Hidden layer represents the second layer. There is no output layer for the model since the transformed input at the hidden layer is feeded as input

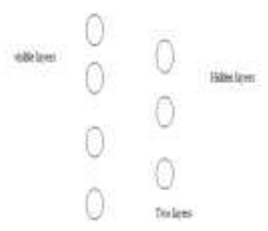

Figure 2 : Layers of RBM

in the second cycle. Data travels from visible layer to hidden layer and vice versa to obtain minimal reconstruction error.

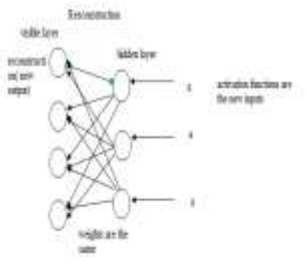
Reconstruction
visible layer
reconstruction/ new output
hidden layer
activation functions are the new inputs
weights are the same

Figure 3: Working of RBM

Each circles in the figure 2 denotes nodes, and calculations take place there. Nodes are connected across layers, but two Same nodes are not connected, The communications between each layers are restricted in RBM. The input is processed to generate outputs and the output is given to the hidden layer which re-transmits the data to the input layer to obtain the exact representation of the input with minimum reconstruction error on the input side of RBM. Here bias function for each layers are initialized randomly. In majority of cases the hidden and visible layers are binary valued but for fraud detection it is better to use the Gaussian function for fraud detection. As shown in the figure 3 whenever the input signal traverse from input to the ouput layer the data will be multiplied by the weight w, and will get added by the bias function b, of the hidden layer. At last the output is squezed between zero and one using the sigmoid fuction which shows the probability of each hidden layer being used. The data is splitted in the ratio 80:20 , and will train the model on the training set. Figure 3 shows Reconstruction of RBM, Restricted boltzmann machine is an energy based model. The Energy function E(x,h) is given below.

Energy function :

$E(x,h) = -h^T W x - c^T x - b^T h$

$= X_j X_k W_{j,k} h_j x_k - X_k ckxk - X_j b_j h_j$

Distribution : $p(x,h) = \exp(-E(x,h)) / Z$

Where Z is the partition function and p(x,h) is the joint probability distribution. x represents visible layer and hidden layer is represented using 'h'. From the energy function we can derive the value of free energy of x. Free Energy; F(x) is what we need for test data from which distribution to detect anomalies is obtained. The greater the free energy, the greater the chance for x to be a fraud. First of all the data is splitted into training and validation sets, and train the model on training set, while evaluate the performance on validation. Starting with the hidden layer which have dimensions smaller than the input layer, the learning rate is set to be a small value (such as 0.001), the validation data set is monitored for reconstruction error (not the actual error) The reconstruction error is basically the mean of square of the difference between predicted â and the actual data x, averaged over the entire batch.

3 Variational Autoencoder

A variational autoencoder (VAE) is obtained by combining a series of autoencoders. Here the input variable is converted to a latent representation and that variable is further converted to the output. The advantage of the model is that the underlying pattern of the data can be obtained. While using generative models a random new output which looks similar to the training data is required and this can be achieved with VAE. It is better to work with VAE than Standard autoencoders when the data is to be changed in a desired way or random way. The model learns to build compact representations and reconstructs the inputs well.

The basic problem for generation in a basic autoencoder is that the latent space converts the inputs to where the encoded vectors lies, It is not necessary that the data required is continuous. On the autoencoder, for MINST data each model required distinct encodings type which makes it easier for decoding them. This is ok if the replication of input is just to be done. Since a generative model is required, random sampling of latent

Space using a continuous data is required. If the space is discontinuous the decoder will generate an output which is unrealistic since the decoder has no idea to deal with the latent space. One of the unique properties that separate autoencoders from VAE is that their latent spaces are continuous which allows random sampling and interpolation.

4 Deep Belief Network

A Deep Belief Network (DBN) aims to learn the structure of the input dataset and features detector layers that can be one or more than one. A DBN is a combination of graphical model that can either be directed or undirected. An undirected RBM constitutes the top layer of the network where as lower layers are downwardly directed. Deep Belief Networks (DBN) are formed by stacking RBMs in greedy manner. The structure of the training data is extracted deeply using DBMs.With RBMs as the building blocks; layer-wise training in unsupervised manner can be applied to DBNs the steps are as follows

1. The input is given to the first RBM layer which act as the input layer.

2. The data in the first layer is processed and the underlying pattern is analyzed.

3. The second layer is trained as another RBM and the output obtained is given to the input of the other layer

4. Steps second and third can be repeated in required number of times. Here the number of hidden layer chosen is 3

5. The data is then pre-processed using logistic regression to obtain the output

6. The output obtained is the result required

Fine-tuning can be done to improve the result using logistic regression unsupervised training strategy used to determine the weight and bias functions.

II. RESULTS AND DISCUSSION

Deep learning based unsupervised learning models were developed using autoencoder, variational autoencoder, restricted boltzmann machine and deep belief network. The result is plotted by using area under ROC curve (AUC). ROC is the receiver operation characteristics. This method is used to determine the precision of the model since the data is highly imbalanced.

1. Autoencoder Model

Autoencoder model was developed in python using keras and tensor flow and the following results were obtained. The confusion matrix showing the normal and fraudulent behaviour is given below.

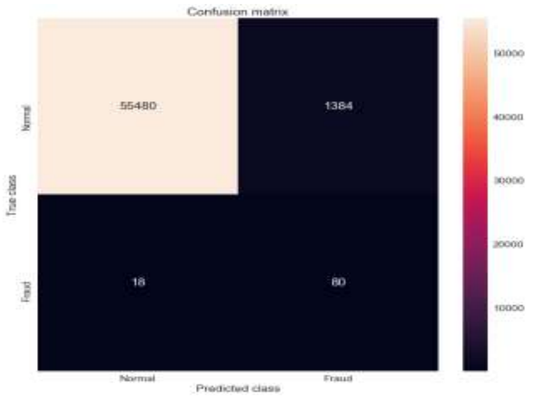

Fig:4 : Confusion Matrix of AE model

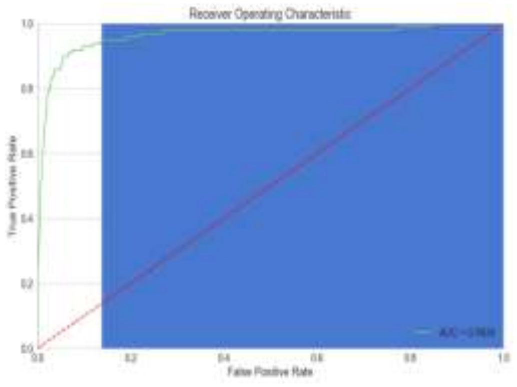

Fig: 5 : ROC curve of Autoencoder Model

The ROC curve plotted gave an Area of 0.9626. This is given in the figure 4. The reconstruction error is also plotted it is shown on figure 5. The AUC value is much closer to one which states that the model can work well with the data and will generate minimum reconstruction error. The threshold for reconstruction error is set close to zero so the model will generate greater output.

2 Restricted Boltzmann Machine

The result obtained when the model is developed using restricted botzmann machine is given below. Here we obtained the area under ROC curve to be 9.600.

Fig: 6 : Confusion Matrix of RBM model

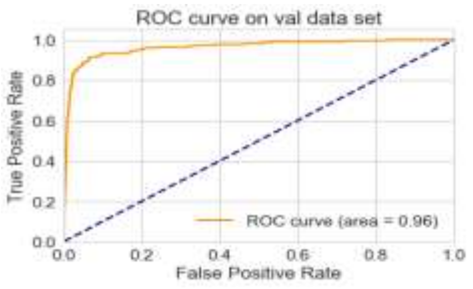

Fig: 7 : ROC curve of RBM model

3. Variational Autoencoder Model

Variational autoencoder is the new model introduced in the project. As explained earlier the result is expected to be more than that of the other two model including RBM and autoencoder. Variational autoencoder is a variation of the Autoencoder model which can deal with more unbalanced data than the former. The result obtained and the ROC curve obtained are given below which shows that the result obtained is the greatest. So this model can be adapted to other fraud detection models.

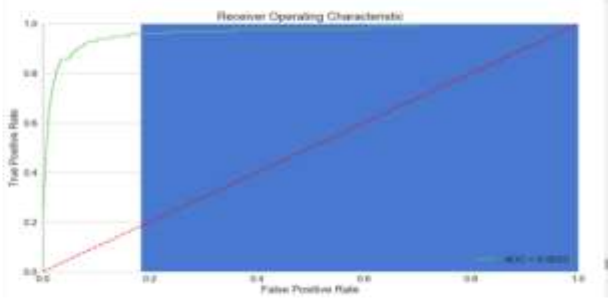

Fig: 8 : ROC curve of Variational Autoencoder Model

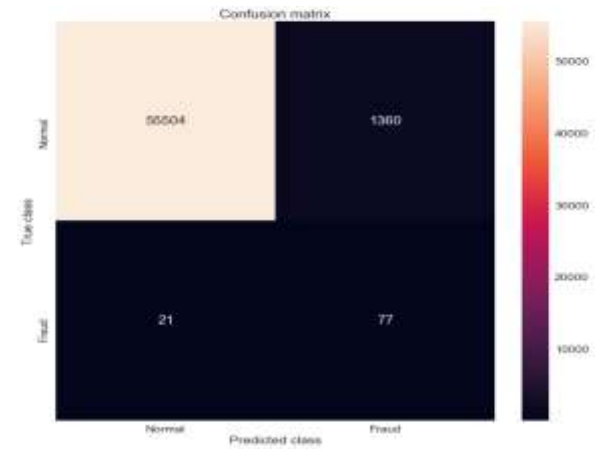

Fig: 9 : Confusion Matrix of Variational Autoencoder Model

4. Deep Belief Network

Deep belief network (DBN) is many layered restricted boltzmann machine. The output obtained is shown in the

fig:10. The model produced an AUC value of 0.9631.

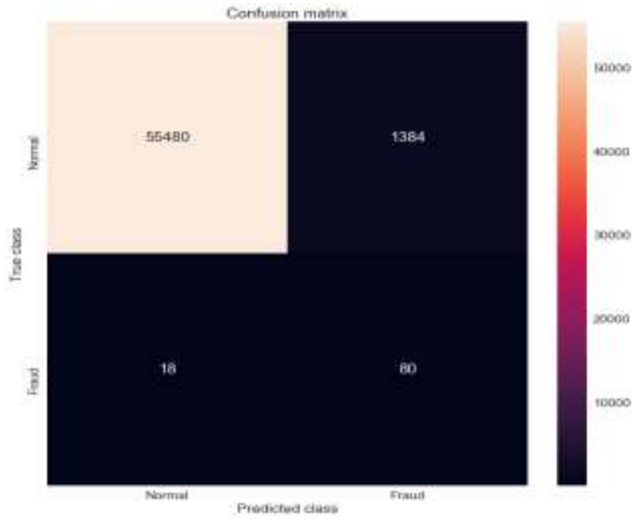

Fig: 10 : Confusion Matrix of Deep Belief Network

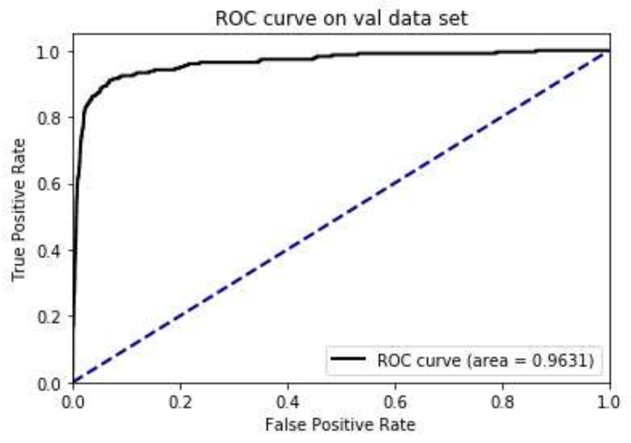

Fig: 11 : ROC curve of Deep Belief Network

For the fraud detection model developed variational autoencoder generated a larger AUC value of 0.9655 followed by deep belief network, autoencoder and restricted Boltzmann machine. If the data set is one in a real scenario the AUC score obtained would be higher than this data. For the same dataset different model generated different AUC curve . Cascading auto encoder to form variational autoencoder proves a better alternative.

AUC score	
Models	AUC Value
Autoencoder	0.9626
Variational Autoencoder	0.9655
Restricted Boltzmann Machine	0.9600
Deep Belief Network	0.9631

III. CONCLUSIONS

The credit card fraud detection model using Autoencoder, Variational Autoencoder, Restricted Boltzmann machine and Deep Belief Network were succesfully implemented and ROC curve were plotted. The result obtained in case of the variational autoencoder is higher than the others. This fraud detection model is trained using the European credit card fraud data. The data availability in case of credit card fraud detection is scarce. Since model learn from data and not from labels it can be transfered to other data set too. ROC curves were plotted and AUC was taken as accuracy criterion since the data is highly imbalanced. The variational autoencoder model created a greater AUC of 0.9655 which is greater than the other model used. If the model is used with real time data the AUC value could be improved.

ACKNOWLEDGMENT

Author is thankful to the faculty at College of Engineering, Trivandrum for guidance and technical support.

REFERENCES

[1]. Pumsirirat,A.andL.Yan(2018). Credit card fraud detection using deep learning based on auto-encoder and restricted boltzmann machine. (IJACSA) *International Journal of Advanced Computer Science and Applications* 9, 18–25.

[2]. Malini, N. and M. Pushpa (2017). Analysis on credit card fraud identification techniques based on knn and outlier detection. *3rd International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics(AEEEICB17)* 978 1(4), 1–4.

[3]. *https://www.kaggle.com/mlg-ulb/creditcardfraud*

[4]. *https://weiminwang.blog/2017/08/05/credit-card-fraud-detection-2-using-restricted-boltzmann-machine-in-tensorflow*