*CREDIT CARD FRAUD DETECTION*

Icon

Description automatically generated

***SUBMITTED BY***

***AJAY KUMAR***

***SUPERVISOR***

***DR. SHAHRAM AZIZI SAZI***

This dissertation is submitted for the degree of “M.Sc., Data Analytics” , (January 2020-21)

# *DECLARATION*

*I hereby Declare that apart from the explicit references made by others to the craft, the substance to the thesis is unique and has not been submitted in entire or to some extent in other colleges and some other degree. This project is my very own and contains nothing which is the result of work done in a group or a joint effort with others.*

*AJAY KUMAR*

*11/01/2021*

# *ACKNOWLEDGEMENT*

*I would firstly like to thank my thesis guide Dr. Shahram Azizi at Dublin Business School. The door to Sir Shahram’ office was always opened for me whenever I faced any issue on my exploration and execution. He guided me with his knowledge and encouraged me in my approach. Without his motivation, it would have been difficult for me to finish this project. Similarly.*

*The achievement would not have been Accomplished without Him.*

*Thank you.*

*Author: Ajay kumar*

# Abstract

In this present project the aim is to establish and understand how various websites, e-commerce sites use various filtering techniques so that credit card fraud can be mitigated. As in the present world, banks providing easy credit cards to one and all the usage and dependence over the credit card has visibly increased. With the essence of liquid cash losing its essence and people sifting more to digital money, the nature of malpractice has shifted to theft to online fraud. Collaborative filtering which has been prevalent technique used in data sensing as well as financial institutions to understand the nature of investment. As they focus on the similarities and differences between users therefore for them by the recommendations system it is possible to detect the credit card fraud that has taken place. The other techniques like the user-based filtering, digital learning, user algorithm have also been proven to be effective in detecting credit card frauds. Through the application of different methods, it will be seen in this project how these system help detecting credit card frauds.

**Table of Contents**

Contents

[*DECLARATION* 1](#_Toc61269854)

[*ACKNOWLEDGEMENT* 1](#_Toc61269855)

[Abstract 2](#_Toc61269856)

[Chapter 1: Introduction 4](#_Toc61269857)

[Chapter 2: Literature Review 7](#_Toc61269858)

[Chapter 3: Project Concept 13](#_Toc61269859)

[Chapter 4: Methodology 17](#_Toc61269860)

[Chapter 5: Experiments 28](#_Toc61269861)

[Conclusion 30](#_Toc61269862)

[Appendices 31](#_Toc61269863)

[References 40](#_Toc61269864)

# Chapter 1: Introduction

With the rapid advances of the online retailing and e-commerce, the uses of credit cards have increased to an extent (Randhawa *et al.,* 2018). Thereby, the transaction between any parts of the globe has become much easier with the credit cards and other transaction mediums. From the previous research, it has been explored that in the present context, the exploitation of search transaction mediums has increased, which has breached consumer protection as well as the validity and reliability of any transaction. The modern world intends to avoid the tangible money transactions and is much inclined towards the electric money. The misuse of technology is being executed to breach the security. The prodigies and malicious attackers put stress on the credit card transactions due to the ease of prodding using the ATM.

There are several kinds of credit card frauds such as clone transactions, False application fraud, Skimming, Account Takeover, suspicious transactions. Let us consider major fraud techniques from above.

A clone transaction is done by making duplicate transactions or similar transactions to the original transaction. This kind of credit card fraud cannot be detected by conventional fraud detection techniques. Conventional fraud detections are not good enough to distinguish a fraudulent transaction, but Artificial Intelligence-based techniques work. Because traditional programming approaches are not good enough to tackle this kind of data imbalanced, pattern recognition problems.

False application fraud is done by applying for a new credit card account using someone else's name. The basic solution is to deal with anomaly detection techniques. Anomaly detection can detect unusual patterns of transactions and transaction time. In Skimming fraudsters fix a machine called “skimmers” to extract card numbers and other cadential of original credit cards. This is based on hardware frauds and they use small pinhole cameras to detect pin user cadential. Based on classification algorithms using features such as geometric location and transaction patterns it can detect skimmers.

Any kind of above credit card fraud can be happening when it reveals the credit card number accidentally or when you lost the card or do transactions with counterfeit cards. So here the goal is to detect credit card frauds and reduce the amount of risk when customers are facing the above situations. However, some of those frauds cannot handle with only algorithmic perspective and need some hardware techniques to track skimmers, because frauds can be done by many methods due to the development of IoT related things and the most basic approach done by most countries is to identify possible fraud tactics done by fraudsters with their experience. This is not a very efficient way, because there may be a unique new method to do the same fraud. So, it needs a more advanced methodology to handle fraud detection. Fraud detection can be done in 2 major ways. Those two ways are Conventional Fraud Detection and Artificial Intelligence-based Fraud Detection. Other than Conventional Fraud Detection it is more appropriate to move with Artificial Intelligence-based Fraud Detection due to the following reasons. Appendix A indicates the difference between conventional methods and Artificial Intelligence based methods. From conventional methods fraud detection can be handled using Collaborative Filtering, User based and item-based Filtering. From Artificial Intelligence based methods it can be handled using both Machine Learning and Deep learning, because anomaly detection can be implemented using many algorithms which fall under both machine learning and deep learning.

The credit card frauds take place when the thief for broad starts illegally uses the stolen credit card on the information from the credit card for making an unauthorized purchase in the credit card holders name or check out the cash balance using the holders account (Dal Pozzolo *et al.,* 2017). The credit card frauds are the fastest growing form of The Identity theft as per the reports of federal trade commission. According to the report the incidences of credit card fraud land has increased by 104 % from 2019 to 2020, and still it is on the way to jump (Mahnken, 2020). Between 2017 and 2019 the credit card fraud land activities proved by 27%. Thereby it can be stated that technology is being exploited for illegal identity theft. In 2018, 246000 cases of credit card frauds have been reported (Mahnken, 2020). The study conducted by the federal trade commission depicts that the people of 60 to 69 years old have been reported the maximum of credit card fraud land cases. The identity theft including credit card fraud land activities are not only costly to the victim cardholders, but also the payment networks as well as banks are suffering massively (Mahnken, 2020). The banks of the financial institutions must refund the victims if the credit card problems have been influenced by their fault. Therefore, in the present content, safeguarding the credit card frauds are extensively crucial. To safeguard such identity theft, it is important to detect the vulnerability of cards and ensure secured transactions. The fraud detection can be regarded as the set of activities executed to ensure the prevention of property or money from being illegally obtained by another party (Randhawa *et al.,* 2018). The research will contribute to offer the knowledge of various credit card fraud detection methods and strategies, which will allow to reduce such frauds in the present context.

***Aim and Objectives***

The research aims to examine, explore, and analyze the potential strategies of detecting credit card frauds to develop the model which can ensure safety of transactions and minimize the vulnerability of the credit cards. The study will endeavor to explore the existing literatures and their opinions and recommendations regarding detection of credit card frauds. The study will be mainly a secondary study, as the information will be collected from literatures.

# Chapter 2: Literature Review

**Collaborative Filtering**

The term collaborative filtering primarily refers to a technique that is used by most recommender systems for data sensing. Most companies and online forums are using the collaborative filtering as a technique that helps in making predictions automatically which helps in forming an understanding about the interest of any user by collaborating and matching it with other users who share similar preferences (Li et al., 2016). The whole method of collaborative filtering works on the assumption that if one user for instance User X has or shares the same experiences as that of User y, then it can be said that on a similar occasion, an opinion given by Y on a matter will be matching with that if users X. For instance when a company is designing a good product and to know it is acceptance in the market, they will run a collaborative filtering to seek recommendations who understands the taste of food or similar category of foods in the market amongst the consumers (Li et al., 2016). The recommendation system in collaborative filtering provides the company with a primary list of tastes and preferences of the other users to understand the taste of any user. But on another's sense collaborative filtering hints towards the process of collecting information through patterns and filtering which involves the usage of different agents, various views from sources and enormous data sets. Collaborative filtering has been effective in cases of detecting and sensing large data sets where monitoring is needed from a large body of data and sensing needs to be enabled. Therefore collaborative filtering has been found effective in detecting the credit cards' frauds (Dong et al., 2017). Thus in almost all financial institutions across the world, for financial data sensing, financial data analysis and to understand the trait of the clients collaborative filtering has been used. Besides financial institutions, many e-commerce sites have used the recommendation application which is one of the most crucial elements of collaborative filtering that provides to users with suggestions of things or items that they are willingly searching. As the name suggests the collaborative filtering through the recommendations systems tries to fulfill the needs and preferences of their users and at the same time keep a tap on the likes and dislikes of the customers (Dong et al., 2017). Keeping a tap on the browsing pattern and the likes of the employees, they personalize the payment mode as well as payment options. Companies like Flipkart, Grofers have seen considerable growth in their business and the sales has been escalates by 40%. Now as per the topic of the project how does collaborative filtering help in detecting credit card fraud? Well, since collaborative filtering uses the recommendation system as well as filtering of data through user history and suggestions thus in a similar pattern they keep a tap on unusual transactions done from a particular card at a particular time, as they already know their user's thus it is easier for the system to detect the fraud (Yao & Huang, 2017).Moreover, as it is said that collaborative filtering has been largely used in financial institutions for financial data using through this technique they detect the credit card fraud that is taking place.

**User based and item based Filtering**

User based filtering as the very name suggests helps in providing to users the very products that they are willing to see or willing to buy. The search is done basing on the preferences and buying patterns of other users as well as search or browsing history of a particular user. Now, for instance User X looks for a microwave oven on Amazon, on a different day and different occasion User Y also looks for microwave along with other items, the user-based filtering which is an auto detector would result in providing suggestions of things to User x based on the preferences of the User Y just because they both looked for a similar item (Thakkar et al., 2019). Now the user-based filtering does not provide suggestions so easily, but they do it by collecting explicit points. They then analyze the explicit points and basing on that they also find similarities as to how many users have rated a similar product in a similar pattern. Through finding similarity between explicit points and by nearest algorithm neighbor they form an understanding about the like of products and the preferences that are specific to certain users. Thus the suggestions that is being provided follows a complex process of searching the nearest neighbour based on the algorithm and then averaging the similarities (Kluver, Ekstrand & Konstan, 2018). Since such a detailed process is being followed by the user-based system agents therefore it enables them to understand if there is any fraud. As they provide user-based suggestions therefore suggestions they know the pattern of the users and any move away from that pattern or budget helps them in understanding that there is potentiality of fraud in respect to expenditure through the credit card (Thakkar et al., 2019).

On the other hand, the item-based filtering is even more specific in terms that it gives suggestions and recommendations to users based on the similarities of items searched by a host of users. In the item-based filtering, the assumption is much higher which shows that if user X likes a certain product then it is given that User Y will also like it as they have similar likes and buying patterns (Tan & He, 2017). However, like user-based filtering, item-based filtering is also carried out in two ways one is obviously through the explicit rates given by the users on their purchasing and the other one is through keeping record of the purchase history and then analyze it. So now we know since they keep such detailed information about the users and analyze their purchasing records so they know how much they can spend at a time. Through this understanding they are being able to detect the frauds that occur in the credit cards. Moreover, the fraudsters have a very similar pattern of carrying out and since these systems have a strong data sensoring capacity they easily detect a fraud and often blocks the user before they can fulfill their deed (Tan & He, 2017). In the present times, where fraudsters have been increasing at an alarming rate the techniques have been proved to be very effective and often seen mitigating the risks in many e-commerce sites.

**Anomaly Detection**

Nowadays the use of credit cards has become a common thing, so the case of fraud has also increased. To detect the fraud in credit cards it is important to detect with a method of detection so that the fraud in the credit card can be detected easily (Kalid et al., 2020). Anomaly detection is a technique that helps to detect fraud in credit cards by recognizing irregular data in the credit card system. Anomaly detection has various techniques with the help of those techniques the unethical terms can be identified easily. Various techniques are as follows:

**Statistical Method-**

This is one of the easiest methods to recognize the unethical terms in the data. The statistical method helps to identify the irregular data in the credit card system by the simple deviation method of mean, median, mode, and quantities (Kalid et al., 2020). These are the division of statistical methods that help to find fraud in credit cards.

**Machine Learning method-**

In the machine learning method, the use of Artificial Intelligence (AI), helps to detect the abnormalities in the data automatically. The main goal of this method is to identify fraud in the credit card system because of its two main reasons- Speed and Adaptability. With the help of this method, the irregularities can be detected rapidly and faster than the human (Xuan et al., 2018). Nowadays, fraudsters change their method but with the help of this technique, the new method of fraud can be easily identified.

**Decision Tree-based Algorithms**

In the method of decision tree-based algorithms, the frauds can be prevented easily. The credit card system nowadays has become more safe and secure (Xuan et al., 2018). Through this decision tree method, credit cards have become more secured with the usage of chips and pins. Nowadays, all the cards are pin protected and even the chip protected so the fraudsters cannot interfere in the credit card system. Through the decision tree-based algorithms, the fraud can be solved easily by detecting the root cause of the issue (Awoyemi, Adetunmbi & Oluwadare, 2017). With the help of the algorithm method, the root cause can be detected, and credit cards can be easily protected. Nowadays, credit cards contain chips, pins, and passwords and to avoid fraud there are various software detections through which the fraud can be quickly identified. The credit card needs to be more safe and secure because it contains all the important credentials of the bank details, so fraud has increased a lot these days. Fraud is a criminal offense that is committed by the fraudsters so that they can transfer the fund unethically (Awoyemi, Adetunmbi & Oluwadare, 2017). Through this method, different techniques have been imposed on the credit card system to protect it from fraudsters.

**Hidden Markov Models (HMM)**

As the latest technologies have emerged the use of credit cards has increased extensively. To detect fraud in credit cards there are various techniques (Bhusari & Patil, 2016). But among all the other techniques, the Hidden Markov Model (HMM) is the simplest and easiest method of fraud detection in credit cards. This technique has two processes one is hidden and another one is open to everybody. This technique involved various steps like States, Transition Probabilities, and Observation (Bhusari & Patil, 2016). With these steps, the fraud can be easily detected by the spending nature of the holder of the card. This method decreases fraud and protects the cards from the fraudsters. Malicious fraud can be easily detected with this technique (Khandare, 2016). This also helps to protect the card by identifying the nature of spending. Nowadays, online transactions have become popular so protecting the credit card with various techniques is essential (Khandare, 2016). The HMM model also helps to mitigate the issues of fraud and impose more techniques to protect the credit cards. This is one of the most effective methods to protect credit cards from fraud and remove the problems from the cards. The detection of fraud has become much faster and easier so that fraudsters cannot commit the crime of transferring the fund (Khandare, 2016). The cardholder should also have essential knowledge like not to share the pin and card details with anybody. This will help to protect the card.

**K-Nearest Neighbour (KNN)**

K-Nearest Neighbor (KNN) technique is broadly used to detect fraud in credit cards. This technique helps to mine the data and detect fraud by using the process of regression and classification. With the help of this technique, the numbers of fraud can be easily identified by the data description method (Khodabakhshi & Fartash, 2016). This method is time-consuming as it includes various steps to detect fraud in credit cards. This method involves various steps so fraud can be very well-detected. KNN technique is based on the two software that helps to detect fraud. Artificial Neural Networks and Support Vector Machines. These are the tools widely used to detect fraud and help the cardholders to securely make online payments (Khodabakhshi & Fartash, 2016). As online transactions have become more popular this software is the latest and quickly determines fraud. This method enables other latest software to detect the fraud, this is very essential to use the latest technologies because everyday fraudsters are imposing new techniques to commit a crime. So, the latest software will help to detect the new techniques of the fraudsters (Khodabakhshi & Fartash, 2016). To remove the issues of fraud, it is very important to use various technologies so that the fraud can be easily detected and prevent the credit cards from becoming safe and secure.

***Literature Gap***

Here in this paper, various methods of detecting fraud have been identified. Those techniques are very crucial to detect fraud and help them to solve the issues (Goud & Premchand 2019). But in this paper, the latest technologies which can essentially remove fraud in the credit card have not been evaluated (Goud & Premchand, 2019). The analysis of various technologies that will help to detect fraud and help to resolve it is necessary to discuss. When fraud is being detected, the card gets blocked instantly, there must be other ways to resolve the issue rather than blocking the card. Further research needs to be done to analyze the essential steps to be done if the card is blocked. There is various software that can be imposed to detect fraud and stop the fraudsters from committing the crime that shall be discussed with proper analysis in further study (Goud & Premchand, 2019). Nowadays, online transactions have increased so it is important for apps that are made to do the online transactions, to protect the cardholders’ useful credentials and protect the credit cards from fraudsters. Further research will analyze the important steps that are required for the credit card holders to be more cautious so that the crime can be prevented from the root.

# 

# Chapter 3: Project Concept

The artificial neural network model is one of the most popular methods to detect credit card fraud. The model consists of an intertwined group of artificial neurons. The model is based upon the functions of the brain which helps in recognizing patterns and associative memories. The network identifies familiar patterns, anticipates future events, and values based on associative memory. The neural network model is largely implied in clustering and classification (Mishra, Gupta, & Singh, 2017). The key benefit of the model over other techniques is that these models adapt from the past and therefore, enhance outcomes with the passage of time. Based on the present scenario, the model extracts rules and anticipates future operations. Effective application of neural networks will enable banks to detect fraudulent activities in credit cards. In the context of the practicum setting, the artificial neural model is a nonlinear statistical information modeling tool. This tool can be used to model the complicated relationship between input and output. These relationships help the recognition of patterns in data.

The first step in the artificial neural network model deals with the transaction database. It symbolizes the activities performed within the non-fraud database. The information required for the model was available with banks where the database is not revealed. The information is nearly available due to confidentiality issues. The information regarding fraudulent and non-fraudulent credit card information is not easily accessible due to customer protection policies of financial places (Dal Pozzolo, et al., 2017). The extracted transaction statements of various account holders will be helpful since the information will be reliable. The second step of preprocessing data includes the transformation of raw data into a comprehensive format. The transformation is necessary since the real-world data is inconsistent and incomplete containing several errors. The data imbalance can be managed by using resampling techniques. PCA is used to diminish the dimensionality of large data sets. It transforms a large set into smaller units. The third step was the use of an artificial neural network as a fraud detection model. The model needs to generate various derivative features to enhance fraud detection performance (Ghobadi, & Rohani, 2016, December). The neural network can function like a human brain if properly trained.

The fourth step of the model deals with predictions that anticipate the probability of fraud and non-fraud activities. The production will help in recognizing possible credit card fraudulent patterns. The predictions are based on in depth learning models that classify these activities into fraud and non-fraud. The fifth step is model deployment. The deployment concept in machine learning enables the application of the model for the prediction of new information. The prediction and deployment aspects in the fraud detection model are interrelated. The information acquired while establishing a predictive model should be arranged and presented in an organized manner that allows efficient usage (Carcillo, et al., 2019). The model used to explain the concept in this project is a binary classifier. The model anticipates fraudulent and non-fraudulent activities of credit cards. It produces the competence to learn by themselves and establish the output that is not restricted to the inputs furnished.

***Why Different from others?***

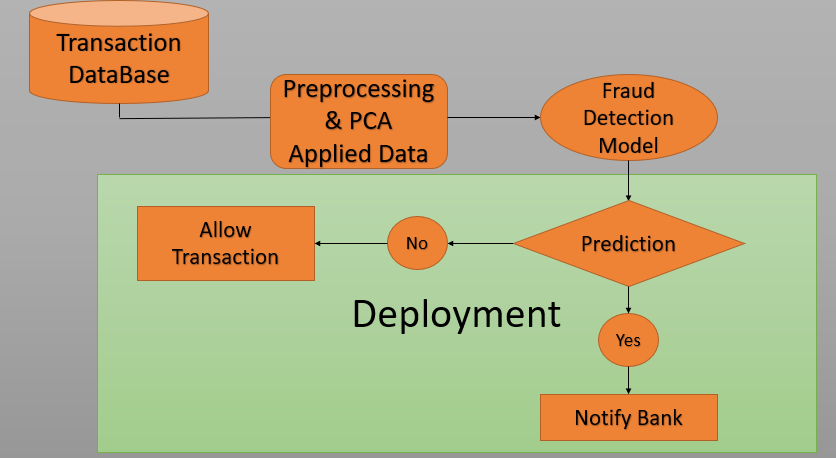
In a fundamental statistics-based machine learning approach, an uncomplicated model is proposed, and information is used to predict parameters. However, an artificial neural network model is different. The uniqueness arises since the neural networks do not include explicit statistical models. The neural network functions a series of transformations on input that can be regarded as computation. The major uniqueness of a neural network is that it allows machines to make reasons and undertake decisions like humans. The construction of a neural network is based upon human brains which itself is an aspect different from all other available models. The brain of a human is constructed through connected networks of neurons (Fu, et al., 2016, October). The neural network stimulates and comprehends these networks simultaneously using computers to play the role of an interconnected brain cell. This computer brain cell adapts to the surroundings and undertakes decisions like humans. The functioning of an artificial network like a human brain cell makes the neural network different from others.

Machine learning is an application that allows automatic learning and progress without specific programming. Whereas the neural network is deep learning technology that emphasizes solving Complex processes. Machine learning uses programming skills whereas neural networks use data modeling skills. Machine learning requires knowledge of big data, algorithms, and data structures. On the contrary, a neutral network requires algorithm, mathematics, Linear Algebra, and graphic theory skills. Machine learning can be applied in e-commerce, tracking price change, online recommendation, enhanced delivery system, and customer services. However, a neural network can be applied in Finance, stock exchange prediction, Artificial Intelligence, and machine learning as well. These differences make neural networks unique. A neural network or deep learning can be considered the answer to the next revolution for machine learning (Randhawa, et al., 2018).

Deep learning is a subcategory of machine learning under artificial intelligence. This system adopts data representation that is unstructured with unlabeled information. Deep learning includes neural networks within its architecture. However, there lies a Stark difference between them. The structure of a neural network consists of neurons, propagation function, learning rate, and connection, and weights (Carcillo, et al., 2019). Neurons and limitation of the biological neuron that estimates the weighted average of data input. Propagation functions provide the predicted value and the error value. Learning rate identifies the rate of updating the weight values of the model. Connection and weights target to diminish the weight value while reducing possibilities of error. On the contrary the components of a deep learning model consist of a motherboard, RAM, PSU, and processors. The motherboard of the deep learning model is based on PCI-e lanes. RAM is the physical storage and memory which should be huge due to larger CPU usage and Storage Area. PSU is required to manage innovative complex deep learning functions. Processors are determined based on cores and cost.

# Chapter 4: Methodology

The proposed method has five stages for building, training, and deploying of the model as in figure below. In the first stage it will collect data from financial authorities and create the initial transaction database. In the second phase using data preprocessing techniques filter the database and apply principal component analysis to transform data. In the third step the model will be built and trained using existing data. In the fourth step model will perform the inference (predictions on new data). Using that as the fifth and final step it will notify the bank immediately if the transaction is fraudulent and allow the transaction if not. As a special case using newly added test data to the database model will be restrained by a scheduler.

****

**Figure (1):** **Project Model**

**Data Collection**

Data Collecting is the tough part of this kind of problem because any financial place including banks never reveals their customer details due to customer protection and overcome fraud. However, after collecting the required amount of non-fraud transaction data, it needs to create fraud transactions as well. This is going to be key because the feature extraction procedure of these networks is mostly depending on the behavior of data distribution. So, it is necessary to analyze more accurately about fraud credit card detections and collect data related to that. Due to these kinds of obstacles it is hard to collect a proper dataset. So the available option is to use an available public dataset. The dataset taken to perform the model is the most usable dataset for credit card fraud detection tasks.

To build a more generalized model it needs to collect an equal amount of data from all credit card fraud techniques. Also, this cannot be done easily because some fraud transactions can only be done for perfectly experienced fraudsters, so it needs to get help from former fraudsters with their experience. Also feature engineering generates some issues because these qualities of the data need to be good. Again, it needs to get help from former fraudsters to implement a quality, reliable dataset. So this part probably needs to be done by experts.

**Data Preprocessing**

After setting up the initial transaction database it should need to perform some basic preprocessing steps to form the data into usable format. In the initial dataset which may have numerous features. But this data is highly confidential because it contains private details of users. So it needs some encoding mechanism to transform this data into some format which cannot be able to decode and backtrack to obtain original data. Principal component analysis is such a great concept which allows to transform data and reduce dimensions. So using principal component analysis dimension of data converted into 28 components as Appendix B.

According to appendix B there are only 492 fraud transactions from a total of 284, 807 samples. Which means there are 284, 315 non-fraudulent transactions. So the ratio between non-fraudulent vs fraudulent transactions is 577.87. So the dataset is highly imbalanced. To overcome the data imbalance problem there are 2 approaches. First method is to remove samples randomly from the majority class until the number of samples in the majority class is equivalent with the minority class. (Sometimes it does not need to be equal to the sizes of 2 sets. This random sample removing procedure does until the ratio between majority and minority class becomes maximum 4.0). The second method is to samples randomly from the minority class until the number of samples in the minority class is equivalent with the majority class. (As in the above case it does not need to be equal to the sizes of 2 sets. This random up sampling procedure does until the ratio between minority and majority class becomes minimum 0.25). So it needs to determine what is the best approach for the credit card fraud detection dataset. Since ratio between non-fraudulent vs fraudulent transactions (majority and minority) is 577.87. So if it up sample minority class until ratio becomes 1, it is probable that each minority sample may up sample for 578 times. Even if it chooses the minimum ratio 0.25, it is probable that minority sample up sample 145 times. So this is not a good approach because this may lead to a huge overfit of the model because of sample repetition. So if it uses a random sample removal approach it needs to convert 284, 315 non-fraudulent transactions into 493 transactions by random removal. This seems a better approach than random up sampling. Appendix D contains the data preprocessing of the model.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Code command: with these commands different python packages are installed

**import numpy as np**

**import pandas as pd**

**from collections import Counter**

**from sklearn.utils import shuffle**

**from sklearn.model\_selection import train\_test\_split**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

With this commands data is uploaded in the model and name the file df and dataset is then in features and output by using pandas’ data frames

**def get\_data():**

**df = pd.read\_csv(data\_path)**

**df\_cols = df.columns.values[1:]**

**Y = df[df\_cols[-1]].astype(np.int32).values**

**X = df[df\_cols[:-2]].values**

**return X, Y**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Code command: performing random sampling to balance the sample data and splitting data into train set and test data

**def load\_data():**

**X, Y = get\_data()**

**positive\_idxs = (Y == 1) # Fraud transactions**

**negative\_idxs = (Y == 0)**

**Xpositive = X[positive\_idxs]**

**Ypositive = Y[positive\_idxs]**

**Xnegative = X[negative\_idxs]**

**Ynegative = Y[negative\_idxs]**

**Nnegative = int(len(Xpositive) \* negative\_to\_positive\_ratio)**

**random\_idxs = np.random.choice(len(Xnegative), Nnegative, replace=False)**

**Xnegative = Xnegative[random\_idxs]**

**Ynegative = Ynegative[random\_idxs]**

**X = np.concatenate((Xnegative, Xpositive), axis=0)**

**Y = np.concatenate((Ynegative, Ypositive))**

**X, Y = shuffle(X, Y)**

**X, Xtest, Y, Ytest = train\_test\_split(**

**X, Y,**

**test\_size = test\_split,**

**random\_state = seed**

**)**

**return X, Xtest, Y, Ytest**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Fraud Detection Model**

For the fraud detection model, it uses an artificial neural network which contains several Dense layers, Batch Normalization layers, and dropout layers. Below figure 2 contains the architecture of the model.

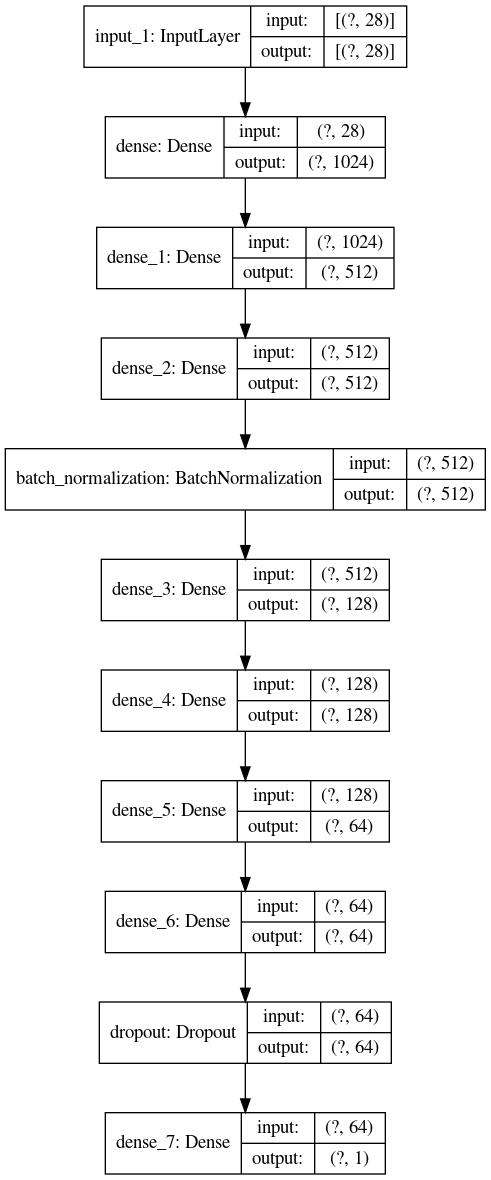
****

Figure 2 : Architecture of ANN

Usually dense layers perform linear operations. The **classifier method** in the below code command used for building the model. So, the purpose of using dense layers is to extract hidden features of the networks. So the first set of hidden dense layers is used to extract features and the last set of dense layers perform feature classification. Deep neural network training is hard when the model is not converging. Which means if the distribution of each layer’s input changes during training from the previous layer stats lead to model divergence and this is known as internal covariance shift. So, it uses batch normalization layer as a solution to overcome this problem. So using batch normalization it will calculate statistical parameters of the previous layer and use that to normalize the layer (by assuming normal distribution).

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Code command: Model fitting, Model compiling**

**import numpy as np**

**from matplotlib import pyplot as plt**

**from matplotlib import cm**

**import os**

**os.environ['TF\_CPP\_MIN\_LOG\_LEVEL'] = '3'**

**import tensorflow as tf**

**from sklearn.metrics import auc**

**from sklearn.metrics import roc\_curve**

**from sklearn.metrics import confusion\_matrix**

**from tensorflow.keras.layers import Dense, Input, Dropout, BatchNormalization, Conv1D, GRU**

**from tensorflow.keras.models import Model, load\_model**

**from tensorflow.keras.optimizers import Adam**

**import logging**

**logging.getLogger('tensorflow').disabled = True**

**from variables import\***

**from util import\***

**physical\_devices = tf.config.experimental.list\_physical\_devices('GPU')**

**if physical\_devices:**

**tf.config.experimental.set\_memory\_growth(physical\_devices[0], True)**

**class myCallback(tf.keras.callbacks.Callback):**

**def on\_epoch\_end(self, epoch, logs={}):**

**if (logs.get('accuracy') > 0.9975):**

**print("\nReached 99.75% train accuracy.So stop training!")**

**self.model.stop\_training = True**

**class FraudDetection(object):**

**def \_\_init\_\_(self):**

**X, Xtest, Y, Ytest = load\_data()**

**self.X = X**

**self.Y = Y**

**self.Xtest = Xtest**

**self.Ytest = Ytest**

**print("Train Input Shape : {}".format(self.X.shape))**

**print("Train Label Shape : {}".format(self.Y.shape))**

**print("Test Input Shape : {}".format(self.Xtest.shape))**

**print("Test Label Shape : {}".format(self.Ytest.shape))**

**def classifier(self):**

**n\_features = self.X.shape[1] #28**

**inputs = Input(shape=(n\_features,))**

**x = Dense(dense1, activation='relu')(inputs)**

**x = Dense(dense2, activation='relu')(x)**

**x = Dense(dense2, activation='relu')(x)**

**x = BatchNormalization()(x)**

**x = Dense(dense3, activation='relu')(x)**

**x = Dense(dense3, activation='relu')(x)**

**x = Dense(dense4, activation='relu')(x)**

**x = Dense(dense4, activation='relu')(x)**

**x = Dropout(keep\_prob)(x)**

**outputs = Dense(num\_classes, activation='sigmoid')(x)**

**self.model = Model(inputs, outputs)**

**self.model.summary()**

**tf.keras.utils.plot\_model(**

**self.model,**

**to\_file=dot\_img\_file,**

**show\_shapes=True**

**)**

**def train(self):**

**callbacks = myCallback()**

**self.model.compile(**

**loss='binary\_crossentropy',**

**optimizer=Adam(learning\_rate),**

**metrics=['accuracy'],**

**)**

**self.history = self.model.fit(**

**self.X,**

**self.Y,**

**batch\_size=batch\_size,**

**epochs=num\_epoches,**

**validation\_split=validation\_split,**

**# callbacks= [callbacks]**

**)**

**self.plot\_metrics()**

**self.save\_model()**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

So, because of this behavior batch normalization allows bigger learning rates of the model. And note that batch normalization is a trainable layer. According to appendix E it can realize that which has trainable parameters (2048 parameters).

Dropout layer used to reduce model overfitting. In forward pass of the network, it will set weights of some neurons to zero using probability for dropping parameters (keep\_prob). This method will force the neural net for more redundant representation. Also helps to reduce co-adaptation of features. So, the dropout layer is known as the most effective regularization technique in deep learning. Dropout layer does not have any trainable parameters. So according to Appendix H it can realize that there are no trainable parameters in the dropout layer. After Compiling the model using the **train method** in appendix C, the model will compile with **binary\_crossentropy** loss, **Adam** optimizer and **accuracy** metric. Finally, the model fits with train data and produces train validation results as in Appendix H. Also, to reuse the model for future tasks model will **save and load** using appendix D.

When Batch size increases it is better to use batch normalization layers as well. Based on the number of components used in PCA it needs to regulate the hidden layer nodes and other parameters. The task has the form of binary classification model and uses output sigmoid activation with intermediate rely activations as in Appendix E. While handling the number of layers and the number of nodes it needs to consider both model accuracy and model size because this kind of machine learning model needs to produce results in real-time with low latency. The trade-off between model accuracy, model size, and model speed is hard to manipulate because it must manage 3 components.

**Predictions**

This model is a supervised deep learning model so that it has output labels. Output labels categorize into fraud vs non-fraud as follows.

0 – non-fraud

1 – fraud

So before doing any transactions it needs to predict the probability of being fraud and non-fraud. That is the reason why those predictions came under the deployment process. At the output layer of the model, it used 1 neuron and sigmoid activation as in Appendix E. So, model prediction is a scalar. If the prediction value is greater than or equal 0.5 consider the output as 1 and if the prediction value is less than 0.5 consider the output as 0.

**Deployment**

Deployment is the most important and dominant process of this methodology. Because it is not worth the model performance when it cannot deploy successfully in a real-world environment. In brief, deployment happened as follows. After applying user cadential for the transaction, before confirming the transaction it needs to predict the transaction label: whether fraudulent transaction or not. If the transaction is a fraud it needs to notify the relevant financial company and if the prediction is non-fraud it needs to proceed with the transaction.

As the first step, it needs to find the right tool for deploying the model. TensorFlow introduces 3 deployment tools as TensorFlow.js, TensorFlow Lite, and TensorFlow Serving API (TFX) for deployment. Using TensorFlow Lite it can handle the edge deployment for embedded devices and mobile devices and TensorFlow.js gives the ability to deploy in web applications. For any kind of deployment scenario, TensorFlow Serving API will be able to serve scheduled learning procedures for the entire model with relevant system configurations. Here mostly it focuses on mobile devices and embedded devices (linked with automated teller machine) based deployment and scheduled learning TensorFlow Lite (*TensorFlow Lite guide*) would be the most appropriate tool. In advanced deployment scenarios it can be used as TensorFlow serving (*Tensorflow for Production*).

TensorFlow Lite is the mobile and embedded deployment tool developed by TensorFlow. It enables edge machine learning inference with low latency and reduced model size. TensorFlow Lite consists of two main components (*TensorFlow Lite guide*). First component is the TensorFlow Lite Interpreter (*TensorFlow Lite guide*) (Appendix G **TFconverter method**). Which can optimize the model to run on different hardware’s such as microcontrollers, mobile devices and more. The second component is a TensorFlow lite converter (*TensorFlow Lite guide*) (Appendix G **TFinterpreter method**). Which basically is used to reduce the size of the model while maintaining accuracy. (Note that model accuracy may reduce from a small portion). Typical koras h5 weights have the form of float32 and which means it needs 4 bytes to represent (*TensorFlow Lite guide*). TensorFlow lite converter convert float32 model into int8 model. So, it needs only one byte to store the same weights. That concludes it will reduce the keras model at least by factor 4. It can experience more reduction than factor 4. Here the size of the keras model is 11 MB and the size of the tflite model is 927 KB. So there is a reduction of factor 11.85. So this is some major improvement of the edge deployment.

Since TensorFlow 2.0 TFX provides native support for TensorFlow lite(*Tensorflow for Production*). TFX enables models for highly efficient inference. So it can apply scheduled learning for the model. In scheduled learning using newly added data into the database model will be retrained with transfer learning. In transfer learning model will freeze first set of hidden layers (feature extraction layers) and train only last few layers (classification layers) this will allow to finetune the model efficiency, in lesser time.

# Chapter 5: Experiments

The fraud detection model developed using TensorFlow. So the high level keras API was used to train the model. Appendix F contains the **hyperparameters** of the model. The model trained using 20 epochs and set the batch size for 64. Also used the learning rate as 0.001 and which is the efficient learning rate for the model. Appendix H contains the training progress of the model.

Basically in model evaluation it uses accuracy as the metric. But due to imbalance criteria of the original dataset and more satisfaction it needs to use more evaluation metrics as well. There are several evaluation metrics used for machine learning/ deep learning models except accuracy metric. Those are **Precision, Recall, F1 score** and in more it uses a special metric called **AUC** (Area under Curve). When finding these metrics is used a concept called **Confusion Matrix**.

Each model prediction can be categorized into 4 categories as in Appendix J, and those are **True Positive (Tp)**, **True Negative (Tn), False Positive (Fp)** and **False Negative (Fn).** True positive means positive ground truth samples predicted as positive. True Negative means negative ground truth samples predicted as negative. False Positive means Ground truth negative samples predicted as positive and False negative means ground truth positive samples predicted as negative. Appendix I, figure 2 indicates the values for each element for test data in the fraud detection task. The purpose of calculating evaluation metrics using these values. Appendix J indicates the equations for calculating these metrics.

There is some tradeoff between False positive and False negative. The error produced by the model is equal to False positive count + False Negative count. So there is a tradeoff between these 2 things. The **main objective is to maximize both precision and recall** (The maximum value for both precision and recall is 1). Due to this tradeoff it cannot increase both these items simultaneously. Due to this reason it introduces F1-score, and which uses both precision and recall getting the idea about model performance. So it is good to use F1-score for evaluation instead of considering precision or recall individually. Appendix N indicates the values for train, validation, test accuracies and precision, recall, F1-score metrics. Also there is another technique called AUC as discussed above which is also used for getting good evaluation to precision vs recall trade off. **ROC** is a graphical visualization of the diagnostic ability of the [binary classifier](https://en.wikipedia.org/wiki/Binary_classifier) system as its discrimination threshold is varied. Also this is known as precision vs recall graph. **AUC is the area under the ROC curve** and this value has the range of 0,1 and which is like evaluation metrics discussed above. Appendix L shows the ROC curve of the model for test data and Appendix K contains the AUC value of the model for test data. As the final evaluation step of the model using model history of fitting the model as in appendix C, it can plot the model variations in each epoch for validation and train losses and accuracies. Using this it can justify whether the model is overfit or underfit or just fit with data. Appendix M indicates the accuracy variation for both train and validation data and Appendix N indicates the loss variation for both train and validation data.

When considering overall evaluation of the model w.r.t Appendices G, K, L, M, N. It can conclude that model performance is pretty good because accuracy, precision, recall, F1-score, and AUC metrics all have values above 0.97. The ROC curve also has the best shape for the true positive and false positive trade off. The loss and accuracy plot for train and validation data also has smooth slopes in each graph. So due to these reasons and factors it can be concluded that the designed artificial neural network well fits with data and has the better model performance that previous work in the credit card fraud detection problem.

# Conclusion

Majority of people nowadays work with E-money instead of physical money. So Credit cards are popular among people and usability also increased. Due to this high usability of credit cards there is a risk of increment of credit card frauds. Nowadays people mostly focused on online platforms for anything. So millions of people use online transactions to make payments. So that is the reason the fraud has also increased these days. To prevent fraudulent transactions in this paper, various models and techniques have been used to detect fraudulent transactions and protect the credit card users from the fraudsters. Both conventional and Artificial intelligence-based techniques can be used to implement this task. Each of these techniques has their own pros and cons such as latency, requirement processing power, and other kinds of factors. After numerous comparisons it can conclude that Artificial Intelligence based techniques are much more efficient than conventional methods. The technique discussed in this paper is Artificial Neural Network based supervised learning binary classifier and which must distinguish a credit card transaction between fraudulent or not. Due to confidentiality of the data, it is hard to collect a regular dataset for this task. Also, Financial credentials must be protected as these days’ frauds have increased and the fraudsters are always trying to bring new techniques and ways to create issues in the credit cards. So proposed method can detect fraudulent transactions successfully and can be deploy in the real-world environment efficiently.

# Appendices

**Appendix A**

**Different Techniques of Fraud Detection**

|  |  |
| --- | --- |
| **Artificial Intelligence Methods** | **Conventional Methods** |
| Ultra-low latency | High latency |
| Ability to extract hidden features of predictions | Able to detect only obvious fraud |
| More improved accuracy | Less Accuracy |
| Efficient Edge-processing capabilities | Too heavy for edge processing. |

**Appendix B**

Credit Card Fraud Detection Dataset, Kaggle.

Anonymized credit card transactions labeled as fraudulent or genuine, Machine Learning Group

DOI: <https://www.kaggle.com/mlg-ulb/creditcardfraud>

**Details about the 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. It contains only numeric input variables which are the result of a PCA transformation. Unfortunately, due to confidentiality issues, we cannot provide the original features and more background information about the data. Features V1, V2, … V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are 'Time' and 'Amount'. Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature 'Amount' is the transaction Amount, this feature can be used for example-dependent cost-sensitive learning. Feature 'Class' is the response variable, and it takes value 1 in case of fraud and 0 otherwise.

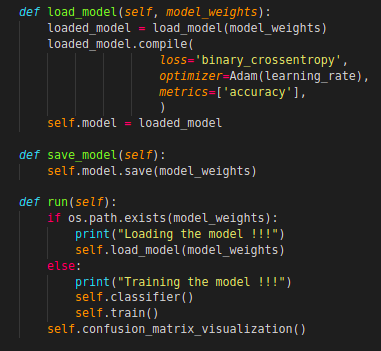
**Appendix C**

**Model building, Model compiling and Model Fitting**

****

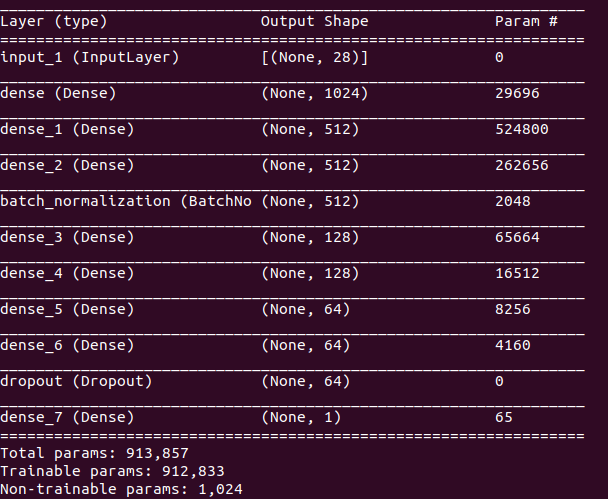
**Appendix D**

**Save and Load the Model for Reuse**

****

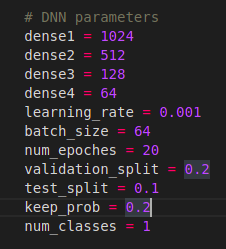
**Appendix E**

**Fraud Detection Model Summary**

****

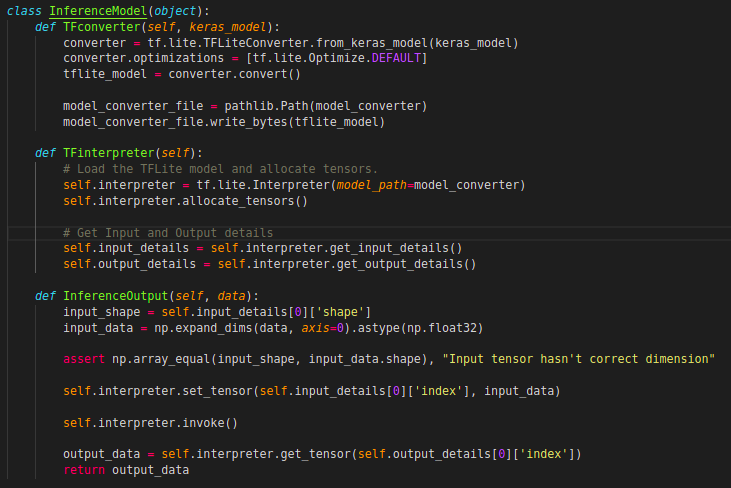
**Appendix F**

**Model Hyperparameters**

****

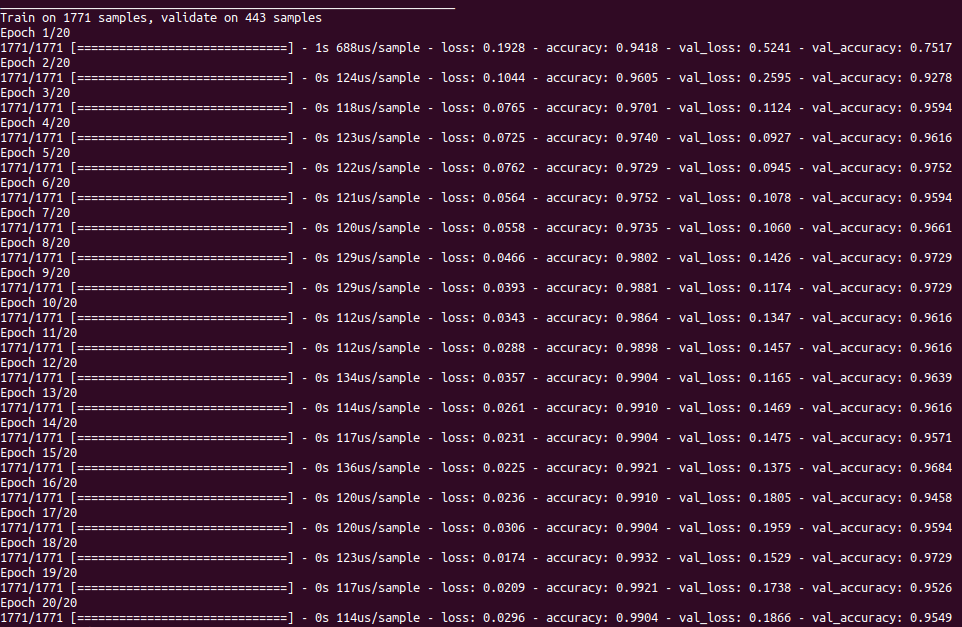
**Appendix G**

**TensorFlow Lite Inference**

****

**Appendix H**

**Model Training for 20 epochs**

****

**Appendix I**

|  |  |
| --- | --- |
| **200** | **4** |
| **3** | **39** |

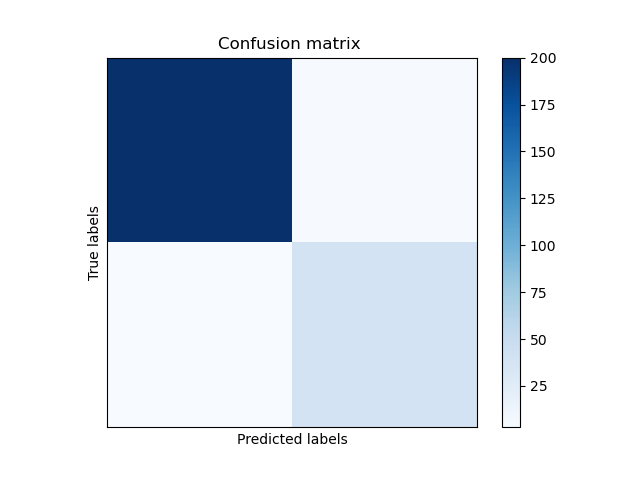
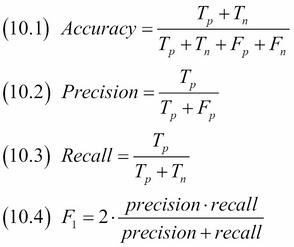
****

Figure: I (1)

Figure: I (2)

**Appendix J**

**Evaluation Metrics**

****

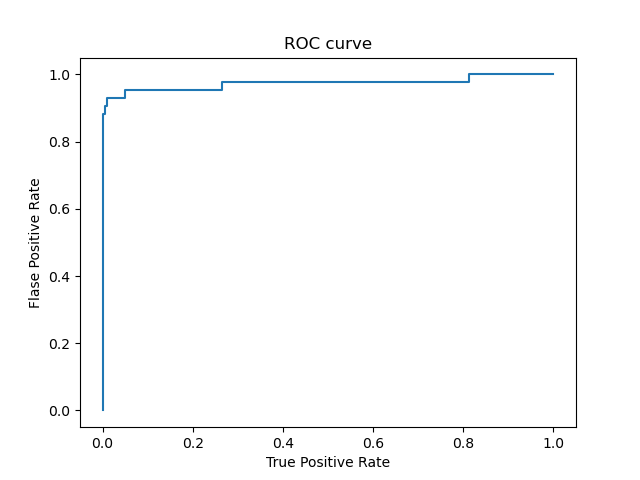
**Appendix K**

**Evaluation of the Model**

|  |  |
| --- | --- |
| Evaluation Metric | Value of Evaluation |
| Train Accuracy | 0.9904 |
| Validation Accuracy | 0.9549 |
| Test Accuracy | 0.9715 |
| Precision (For Test data) | 0.9852 |
| Recall (For Test data) | 0.9803 |
| F1-score (For Test data) | 0.9827 |
| AUC (For Test data) | 0.97280 |

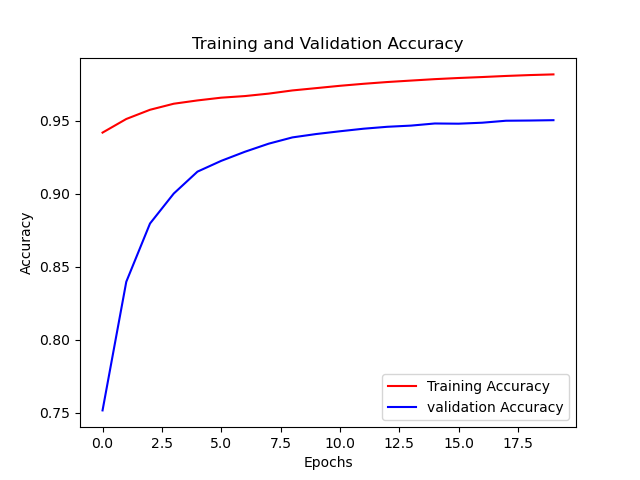
**Appendix L**

**ROC curve**

****

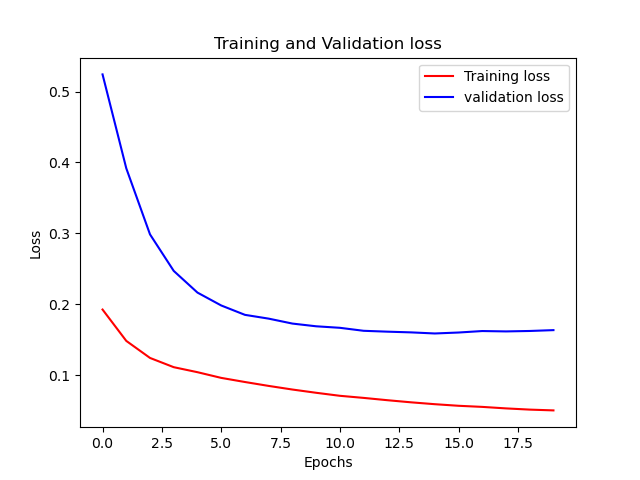
**Appendix M**

**Accuracy Plot of the Model**

****

**Appendix N**

**Loss Plot of the Model**

****

# References

Awoyemi, J. O., Adetunmbi, A. O., & Oluwadare, S. A. (2017, October). Credit card fraud detection using machine learning techniques: A comparative analysis. In *2017 International Conference on Computing Networking and Informatics (ICCNI)* (pp. 1-9). IEEE.

Bhusari, V., & Patil, S. (2016). Study of hidden markov model in credit card fraudulent detection. In *2016 World Conference on Futuristic Trends in Research and Innovation for Social Welfare (Startup Conclave)* (pp. 1-4). IEEE.

Carcillo, F., Le Borgne, Y. A., Caelen, O., Kessaci, Y., Oblé, F., & Bontempi, G. (2019). Combining unsupervised and supervised learning in credit card fraud detection. *Information Sciences*.

Dal Pozzolo, A., Boracchi, G., Caelen, O., Alippi, C. & Bontempi, G., (2017). Credit card fraud detection: a realistic modeling and a novel learning strategy. IEEE transactions on neural networks and learning systems, 29(8), pp.3784-3797.

Dong, X., Yu, L., Wu, Z., Sun, Y., Yuan, L., & Zhang, F. (2017, February). A hybrid collaborative filtering model with deep structure for recommender systems. In Proceedings of the thirty-first AAAI conference on artificial intelligence (pp. 1309-1315).

Fu, K., Cheng, D., Tu, Y., & Zhang, L. (2016, October). Credit card fraud detection using convolutional neural networks. In *International Conference on Neural Information Processing* (pp. 483-490). Springer, Cham.

Ghobadi, F., & Rohani, M. (2016, December). Cost sensitive modeling of credit card fraud using neural network strategy. In *2016 2nd international conference of signal processing and intelligent systems (ICSPIS)* (pp. 1-5). IEEE.

Goud, V. S., & Premchand, P. (2019). ENHANCED HIDDEN MARKOV MODEL FOR CREDIT CARD FRAUD DETECTION. *Complexity International*, *23*(2).

Kalid, S. N., Ng, K. H., Tong, G. K., & Khor, K. C. (2020). A Multiple Classifiers System for Anomaly Detection in Credit Card Data with Unbalanced and Overlapped Classes. *IEEE Access*, *8*, 28210-28221.

Khandare, N. B. (2016). Credit card fraud detection using hidden Markov model. *International Journal*, *1*(4).

Khodabakhshi, M., & Fartash, M. (2016, November). Fraud detection in banking using knn (k-nearest neighbor) algorithm. In *Proceedings of the 5th international conference on research in science and technology* (pp. 27-34).

Kluver, D., Ekstrand, M. D., & Konstan, J. A. (2018). Rating-based collaborative filtering: algorithms and evaluation. In Social Information Access (pp. 344-390). Springer, Cham.

Li, B., Wang, Y., Singh, A., & Vorobeychik, Y. (2016). Data poisoning attacks on factorization-based collaborative filtering. In Advances in neural information processing systems (pp. 1885-1893).

Mahnken, G., (2020). Credit Card Fraud Has Exploded in Recent Years (2020). Available at: https://www.creditcardinsider.com/blog/2020-fraud-and-identity-theft-analysis/#:~:text=Reports%20of%20credit%20card%20fraud%20jumped%20by%20104%25%20from%20Q1,2020%2C%20and%20continue%20to%20grow.&text=The%20opening%20of%20new%20credit,2018%20and%20246%2C000%20cases%20reported. (Accessed: 18 November 2020).

Mishra, C., Gupta, D. L., & Singh, R. (2017). Credit Card Fraud Identification Using Artificial Neural Networks. *International Journal of Computer Systems*, *4*(07).

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

Tan, Z., & He, L. (2017). An efficient similarity measure for user-based collaborative filtering recommender systems inspired by the physical resonance principle. IEEE Access, 5, 27211-27228.

Thakkar, P., Varma, K., Ukani, V., Mankad, S., & Tanwar, S. (2019). Combining user-based and item-based collaborative filtering using machine learning. In Information and Communication Technology for Intelligent Systems (pp. 173-180). Springer, Singapore.

Xuan, S., Liu, G., Li, Z., Zheng, L., Wang, S., & Jiang, C. (2018, March). Random forest for credit card fraud detection. In *2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC)* (pp. 1-6). IEEE.

Yao, S., & Huang, B. (2017). Beyond parity: Fairness objectives for collaborative filtering. In Advances in Neural Information Processing Systems (pp. 2921-2930).

*Tensorflow for Production, TFX for Mobile*

*DOI:* [*https://www.tensorflow.org/tfx/tutorials/tfx/tfx\_for\_mobile*](https://www.tensorflow.org/tfx/tutorials/tfx/tfx_for_mobile)

*TensorFlow Lite guide, TensorFlow Lite for*  [*Mobile & IoT*](https://www.tensorflow.org/lite)

*DOI:* [*https://www.tensorflow.org/lite/guide*](https://www.tensorflow.org/lite/guide)