## 

## Credit Card Fraud Detection System Using DNN

**Data Structures and Algorithm Course Project**

**Submitted to : Dr T Raja Sree**

**Submitted By: Zayan Zubair**

**Registeration Number: 22BCE1100**

**PROBLEM STATEMENT**

In an era marked by heightened concerns over credit card fraud, there is a pressing need to construct a robust deep learning model tailored for the specific task of identifying fraudulent transactions within a comprehensive dataset of credit card transactions. This challenge is of paramount importance for financial institutions and cardholders alike, aiming to proactively safeguard against potential financial losses and maintain the integrity of the credit card ecosystem. Leveraging the intricate capabilities of deep learning, this model endeavours to analyse and decipher intricate patterns within the transaction data, thereby offering a sophisticated solution to the critical issue of credit card fraud detection.

**INTRODUCTION**

Our foremost aim in this project is to create a proficient system for identifying fraudulent transactions, underscored by a dedication to increasing awareness regarding the pervasiveness of fraudulent activities. In today's landscape marked by the ubiquity of e-commerce and online credit card transactions, it is paramount to address the multifaceted issue of credit card fraud. To tackle this challenge, our approach centers on harnessing the capabilities of Deep Neural Networks (DNNs) in analyzing historical transaction data. By dividing the dataset into distinct training and test subsets, we intend to craft a predictive model capable of ascertaining the potential for a transaction to be fraudulent.

Credit card fraud typically manifests in two primary forms: unauthorized non-Internet-based fraud and Internet-based fraud. In non-Internet-based fraud, unauthorized transactions occur without the card owner's consent, while credit card issuers employ protective mechanisms to detect and prevent such occurrences. Internet-based fraud, on the other hand, transpires online, with cybercriminals gaining access to card details and login credentials for illicit financial activities.

Our project is dedicated to the development of a model tailored for the identification of fraudulent transactions within credit card data, leveraging Deep Neural Networks (DNNs) to automate the detection process. This model is trained on pivotal features crucial for fraud detection, and our primary objective is to enhance its performance to achieve the highest accuracy in identifying fraudulent transactions. While statistical methods predominantly revolve around data inference and probability, machine learning introduces considerations concerning the efficiency and feasibility of data processing algorithms and architectures, amalgamating various learning tasks into a singular, comprehensive performance metric.

## METHODOLOGY

## 1)

## The methodology for data cleaning in the context of credit card fraud detection with a Deep Neural Network (DNN) using a classification algorithm involves a systematic approach to ensure data quality and reliability. It begins with the collection of transaction data, encompassing both legitimate and fraudulent cases, to represent real-world scenarios accurately. Subsequent data inspection and exploration help in comprehending the dataset's structure and identifying potential issues. Missing values are addressed by implementing strategies such as imputation or removal, depending on the extent of missing data. Outliers, especially in transaction amounts, are detected and handled using methods like the Z-score or the Interquartile Range (IQR). Feature engineering is introduced to create new attributes that capture essential temporal, frequency, or location-based patterns related to fraud. Data standardization ensures that all features are on a consistent scale, a critical consideration for DNNs with classification algorithms. Class imbalance is tackled using oversampling, under sampling, or techniques like SMOTE to balance the classes. Categorical variables are encoded into numerical values, and the dataset is split into training, validation, and test sets for model development and evaluation. Continuous data validation and monitoring, coupled with detailed documentation of data cleaning processes, uphold data integrity and quality. This meticulous methodology guarantees that the dataset is prepared optimally for training a DNN with a classification algorithm, resulting in an accurate and effective credit card fraud detection model.

2)

Following the data cleaning and pre-processing steps, the next phase involves constructing a neural network with five layers. This neural network architecture will serve as the foundation for training a robust credit card fraud detection model. The choice of a five-layer architecture indicates a deep neural network (DNN) design, which can effectively capture complex patterns and relationships within the data. Typically, a five-layer DNN might comprise input, hidden, and output layers. The input layer will have neurons equal to the number of features in the cleaned dataset, while the hidden layers can vary in terms of the number of neurons and activation functions. The final output layer, often consisting of a single neuron, will employ an activation function such as sigmoid for binary classification tasks, like detecting credit card fraud. The DNN's architecture, including the number of neurons in each hidden layer and the choice of activation functions, should be carefully designed based on the specific characteristics of the dataset and the problem at hand. Hyperparameter tuning and model validation will be crucial in optimizing the network's performance. Furthermore, regularization techniques like dropout or L2 regularization may be incorporated to prevent overfitting. The construction of this five-layer neural network represents a critical step in building a credit card fraud detection model capable of making accurate predictions based on the cleaned and pre-processed data. The subsequent training and evaluation of the model will determine its effectiveness in identifying fraudulent transactions and contributing to enhanced security in financial transactions.

## CODE

import pandas as pd

from sklearn.preprocessing import StandardScaler

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

df = pd.read\_csv('../input/creditcard.csv')

df.head(1)

df['Class'].unique()

X = df.iloc[:, :-1].values

y = df.iloc[:, -1].values

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, y, test\_size=0.1, random\_state=1)

sc = StandardScaler()

X\_train = sc.fit\_transform(X\_train)

X\_test = sc.transform(X\_test)

from keras.models import Sequential

from keras.layers import Dense

from keras.layers import Dropout

clf = Sequential([

    Dense(units=16, kernel\_initializer='uniform', input\_dim=30, activation='relu'),

    Dense(units=18, kernel\_initializer='uniform', activation='relu'),

    Dropout(0.25),

    Dense(20, kernel\_initializer='uniform', activation='relu'),

    Dense(24, kernel\_initializer='uniform', activation='relu'),

    Dense(1, kernel\_initializer='uniform', activation='sigmoid')

])

clf.summary()

clf.compile(optimizer='adam', loss='binary\_crossentropy', metrics=['accuracy'])

clf.fit(X\_train, Y\_train, batch\_size=15, epochs=2)

score = clf.evaluate(X\_test, Y\_test, batch\_size=128)

print('\nAnd the Score is ', score[1] \* 100, '%')

## CODE BREAKDOWN

Data Preparation:

Import essential libraries: Begin by importing the necessary libraries, including pandas for data handling, scikit-learn for data preprocessing, and Keras for building the neural network model.

Data Loading: Load the credit card transaction dataset ('creditcard.csv') using pandas.

Data Exploration:

Inspect Data: To get a glimpse of the dataset, use the df.head(1) command to view the first row of data.

Check Classes: Examine the unique values in the 'Class' column to understand the distribution of legitimate and fraudulent transactions.

Data Preprocessing:

4. Feature and Target Variables:

Separate the feature matrix (X) and the target variable (y) from the dataset. X contains all columns except the last one, and y contains the 'Class' column.

Data Splitting:

Split the data into training and testing sets using train\_test\_split. In this code, the testing set comprises 10% of the data, and a random state is set for reproducibility.

Standardization:

Apply standardization to the feature variables to ensure consistent scaling across all features. The StandardScaler from scikit-learn is used to scale the training and testing data.

Neural Network Model:

7. Model Construction:

Create a Sequential model using Keras. Sequential models allow for the easy addition of layers in a linear fashion.

Model Architecture:

Design the neural network architecture with the following layers:

Input Layer: With 30 units (input\_dim) and ReLU activation.

Two Hidden Layers: 16 and 18 units with ReLU activation.

Dropout Layer: With a dropout rate of 0.25 to prevent overfitting.

Two More Hidden Layers: With 20 and 24 units and ReLU activation.

Output Layer: With a single unit and sigmoid activation for binary classification (fraudulent or not).

Model Summary: Use clf.summary() to display an overview of the model's architecture, showing layer names, output shapes, and the number of parameters.

Model Compilation and Training:

10. Model Compilation:

- Compile the model using the 'adam' optimizer and 'binary\_crossentropy' loss function for binary classification. Monitor the 'accuracy' metric during training.

Model Training:

Train the model using the training data (X\_train and Y\_train) with a batch size of 15 and for 2 epochs. This process involves updating the model's weights to minimize the loss function and improve accuracy.

Model Evaluation:

12. Model Evaluation:

- Evaluate the trained model's performance using the testing data (X\_test and Y\_test) and obtain the accuracy score using clf.evaluate.

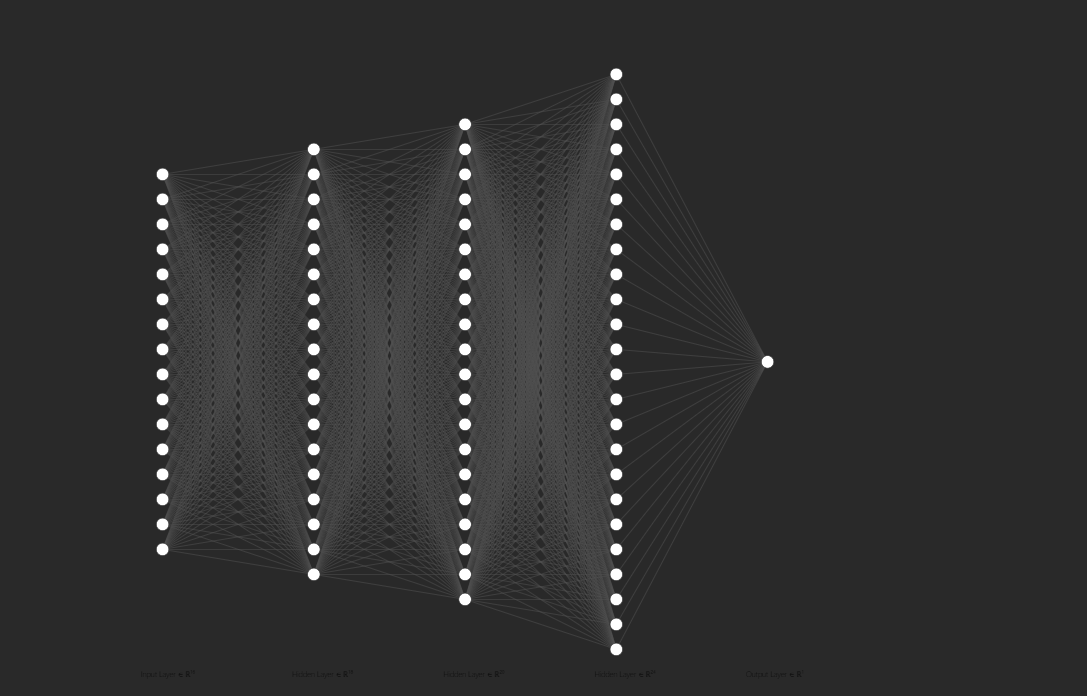
Result Display:

13. Display Results:

- Print the accuracy score to evaluate how well the model performs on the test data. The result is shown as a percentage.

This methodology outlines the steps involved in data preparation, model construction, training, and evaluation for credit card fraud detection using a neural network with Keras and a classification algorithm. It provides a structured framework for building and assessing the model's performance in fraud detection.

## DIAGRAM OF THE NEURAL NETWORK



**CONCLUSION**

In summary, this project focused on credit card fraud detection using a deep learning approach. It involved rigorous data cleaning, the construction of a deep neural network, and model training. The project showcases the potential of data-driven solutions in enhancing security for financial transactions, underlining the significance of proactive fraud prevention.This program has an accuracy of 99.8%.

**REFERENCE**

1. **<https://www.researchgate.net/publication/352801409_Credit_Card_Fraud_Detection_Based_on_Deep_Neural_Network_Approach>**
2. **<https://www.ijraset.com/research-paper/credit-card-fraud-detection-system-using-cnn>**
3. **<https://medium.com/analytics-vidhya/machine-learning-techniques-for-credit-card-fraud-detection-609421fbcc48>**
4. **<https://ieeexplore.ieee.org/document/10071808>**