Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 04, April-2020

Credit Card Fraud Detection using Classification, Unsupervised, Neural Networks Models

L. Bhavya[1], V. Sasidhar Reddy[2], U. Anjali Mohan[3], S. Karishma[4]
Assistant Professor (Adhoc), Dept of CSE,JNTUACEP, Pulivendula, AP, India
Student, Dept of CSE, JNTUACEP, Pulivendula, AP, India
Student, Dept of CSE, JNTUACEP, Pulivendula, AP, India
Student, Dept of CSE, JNTUACEP, Pulivendula, AP, India

***Abstract*: Nowadays online transactions have grown in large quantities. Among them, online credit card transactions hold a huge share. Therefore, there is much need for credit card fraud detection applications in bans and financial business. Credit card fraud purposes may be to obtain goods without paying or to obtain unauthorized funds from an account. With the demand for money credit card fraud events became common. This results in a huge loss in finances to the cardholder.**

Previously they used the most common methods like rule-induction techniques, fuzzy system, decision trees, Support Vector Machines (SVM), K-Nearest Neighbor algorithms to detect the fraud transaction using a credit card. From our perspective, neural networks will generate more accurate results.

To increase the accuracy and precision we use the algorithms Logistic Regression, K-Means, Convolution Neural Networks. Logistic Regression is a statistical model that tries to minimize the cost of how wrong a prediction is. CNN algorithm is used, to capture the intrinsic patterns of fraud behaviors learned from labeled data. So will make use of accuracy and precision to evaluate the performance of the proposed system.

Keywords: CNN, Credit Card, Fraud detection, Logistic Regression, K-Means.

I. INTRODUCTION

The rising of E-commerce business has resulted in a gentle growth within the usage of credit cards for online transactions and purchases. With the rise in the usage of credit cards, the number of fraud cases has also been doubled. Credit card frauds are those which are done with an intention to gain money in a deceptive manner without the knowledge of the cardholder.

A credit card fraud can be done in the subsequent ways:

1. Fake IDs: This type of fraud is done by using the personal information of other persons without any authorization.
2. Skimming Method: This type of fraud is done by installing a device called "skimmer" at ATM machines. The data present on the magnetic stripe of the card is collected when the card is swiped
3. Card not present: When the fraudster steals the data like the expiry date and account number of the card, they can use the card without being possessed physically.
4. False businessperson sites: this type of fraud is done by the method Phishing, where they create a website/webpage which looks similar to that of the original site.

Detecting fraud is extremely difficult. A large variety of parameters are used to choose and classify. Transactions cannot be classified into fraud/genuine strictly. However can be found by the intensive study of the defrayment habits, customer's behavior and by analyzing pre-fraud patterns. Several challenges are faced by fraud detection techniques as they need to detect the fraud in a very short time and large parameters ought to be processed while training.

In our paper, we compared the techniques to evaluate, which gives the most effective performance underneath what conditions.

Logistic Regression, K-Means, and Convolution Neural Networks algorithms are used to achieve high accuracy rates and to increase the detection process. Convolution Neural networks are used to identify the underlying patterns that are followed in the previous cases and thereby increasing efficiency.

II. LITERATURE SURVEY

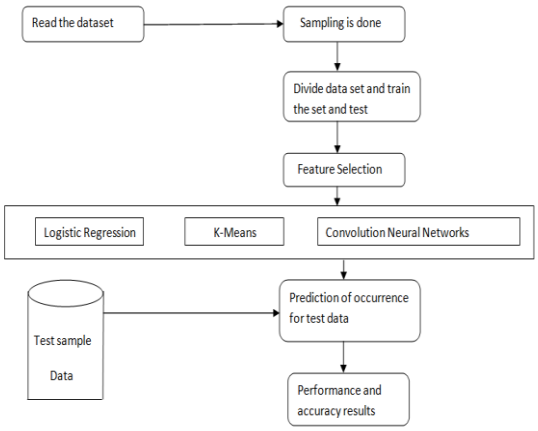
In 2019, Yashvin Jain, Namrata Tiwari proposed the comparative study of all the existing systems i.e. Support Vector machines, Bayesian network, Artificial neural networks, present to detect the credit card frauds.

In [2], they discussed the performance of Logistic Regression, Decision Tree and Random Forest to detect the fraud.

In [3], this paper discusses the probability of fraudulent transactions in prevalence and context of credit card usage.

III. METHODOLOGY

III.1 ARCHITECTURE



International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 04, April-2020

In the very step, we downloaded the dataset from the website Kaggle and the link is https://www.kaggle.com/mlg-ulb/creditcardfraud#__sid=js0.So, initially, the dataset is read carefully and understood. Next, the dataset is sampled i.e. apart of the dataset is taken for observation and preprocessing is done. Here, the unwanted data is removed if necessary. After data pre-processing the dataset is divided for training and testing. We divided 75% of the dataset for training and the remaining 25% of the dataset for testing. The next step is to create the models for the dataset. The architecture contains three types of models. Each model falls under a different category. The models are Logistic Regression which comes under Classification, K-Means which comes under Clustering, and Convolutional Neural Networks which comes under Neural Networks. Next, the testing data is taken and predicted with the help of trained data. Finally, the results are noted. Performance and the accuracy for each model are observed and the best model is to be found.

III.2 MODELS FOR DETECTING CREDIT CARD FRAUD

```
logistic = linear_model.LogisticRegression(C=1e5)

logistic.fit(X_train, y_train)

print("Score: ", logistic.score(X_test, y_test)p)
```

The models that we have used for detecting the credit card fraud are:

1. Logistic Regression
2. K-Means
3. Convolutional Neural Networks

III.2.1 LOGISTIC REGRESSION

The Logistic Regression is a Classification model used mainly for the binary classification datasets. Since our dataset is a classification dataset we used this Logistic Regression. It mainly classifies the dataset into two binary values finally which are 0's and 1's to detect the fraud in the credit card transaction. Initially, the dataset is loaded with the help of the panda's library. In the next step, the dataset is split into X and y values and sizes are printed. For training and testing, we will use the train_test_split() method and the test_size is given as 25% as shown below.

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

y = df['Class']

print("X and y sizes, respectively:", len(X),
len(y))

X_train, X_test, y_train, y_test =
train_test_split(X, y, test_size=0.25)
```

After the division of the dataset into training and testing size Logistic Regression model is implemented on divided dataset. In this model first we should train the dataset and next testing is done for the remaining data with the help of prediction method.

The logistic score that we got here is 99.88%.

Finally the binary confusion matrix is noted for this Logistic Regression with the usage of method ConfusionMatrix() method.

```
confusion_matrix = ConfusionMatrix(y_right,
y_predicted)

print("Confusion matrix:\n%s" %
confusion_matrix)

confusion_matrix.plot(normalized=True)

plt.show()

confusion_matrix.print_stats()
```

The status of the Confusion Matrix are:

Confusion matrix:

Predicted	False	True	__all__
Actual			
False	99477	30	99507
True	76	100	176
__all__	99553	130	99683

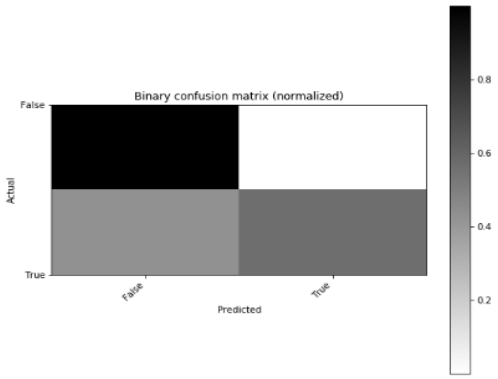


Fig. 2. Confusion Matrix for Logistic Regression

III.2.2. K-MEANS

K-Means clustering algorithm is one of the unsupervised mac hine learning algorithms. This K-Means algorithm group simi lar data points together and finds the underlying patterns. To obtain these clusters should be formed. Here the clusters are n othing but centroids. Here in this K-Means model, we used P CA to reduce the dataset attributes into 2 components and sca ling is also done for them. Then we used the train_test_split() method for training and testing..

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

y = df['Class']

X_scaled = scale(X)

pca = PCA(n_components=2)

X_reduced = pca.fit_transform(X_scaled)
```

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 04, April-2020

From here the implementation of the K-Means Algorithm starts. To create a K-Means model we use kmeans() method as shown below with two clusters..

```
kmeans = KMeans(init='k-means++', n_clusters=2, n_init=10)

kmeans.fit(X_train)
```

After fitting the model the output looks like,

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
    n_clusters=2, n_init=10, n_jobs=1, precompute_distances='auto',
    random_state=None, tol=0.0001, verbose=0)
```

OUTPUT:

With this, the K-means model is created and next, we should form clusters by creating centroids in the model and grouping the nearest data points to the nearest cluster. Since this algorithm works well on an unsupervised dataset this algorithm can't give good results for our dataset. After forming centroids for clusters if we plot the graph for K-Means it looks like as shown below

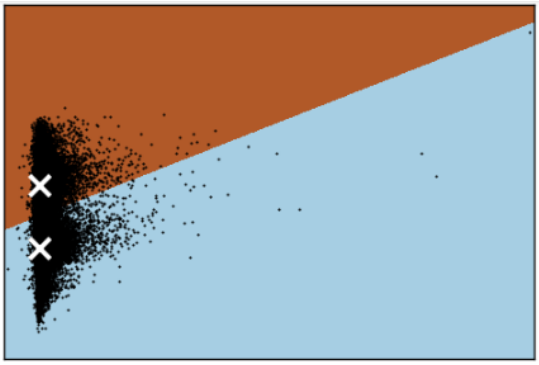


Figure: K-Means clustering on the credit card fraud dataset

In the above figure, the centroids are marked with white cross. Here we have two white crosses. So, we have two centroids for our dataset. This model gives a test accuracy of 54.27%. This is because this algorithm works only on an unsupervised dataset.

III.2.3. CONVOLUTIONAL NEURAL NETWORKS

The Convolutional Neural Networks is a neural network algorithm mainly used for classifying and detecting the image. But we can also use this in credit card fraud detection. Initially, the dataset is loaded and split for training and testing purposes with the ratio 75:25 as shown below.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25)
```

a) Simple Neural Networks

To create a neural network we Sequential() method. Initially, we created only two hidden layers in this model. Even though the accuracy is good but the loss value is high.

```
model = Sequential()

model.add(Dense(30, input_dim=30, activation='relu'))

model.add(Dense(1, activation='sigmoid'))

model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

model.summary()
```

The summary of the model that created is:

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 30)                930
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 31
=================================================================
Total params: 961
Trainable params: 961
Non-trainable params: 0
_________________________________________________________________
```

Fig.5. Neural Networks with PCA (10 components)

After fitting this model we got a 99.67% accuracy rate but with a high loss value rate which is 99.785%. It has used only one epoch for iteration.

Output:

Epoch 1/1
190820/190820 [==============================] -

```
model.fit(X_train.as_matrix(), y_train, epochs=1)
```

16s 83us/step - loss: 9.9785 - accuracy: 0.9961

b) Neural Networks with PCA (10 components)

In the simple neural network, it got high loss value. To decrease this loss value we create more hidden layers. We have created five hidden layers in this model. To create a neural network we use Sequential() method as shown.

```
model2 = Sequential()

model2.add(Dense(10, input_dim=10, activation='relu'))
model2.add(Dense(27, activation='relu'))
model2.add(Dense(20, activation='relu'))
model2.add(Dense(15, activation='relu'))
model2.add(Dense(1, activation='sigmoid'))
model2.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

model2.summary()
```

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 04, April-2020

The summary for this five hidden layer model is:

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_10 (Dense)             (None, 10)                110
_________________________________________________________________
dense_11 (Dense)             (None, 27)                297
_________________________________________________________________
dense_12 (Dense)             (None, 20)                560
_________________________________________________________________
dense_13 (Dense)             (None, 15)                315
_________________________________________________________________
dense_14 (Dense)             (None, 1)                 16
=================================================================
Total params: 1,298
Trainable params: 1,298
Non-trainable params: 0
_________________________________________________________________
```

Fig.5. Neural Networks with PCA (10 components)

After the creation of the model, the PCA reduced components are used for testing using five epochs. By using five epochs we will get finally an accuracy rate of 99.99% with less loss v alue 0.038%.

```
X2_test = pca.fit_transform(X_test)

h = model2.fit(X2, y2, epochs=5,
validation_data=(X2_test, y_test))
```

OUTPUT:

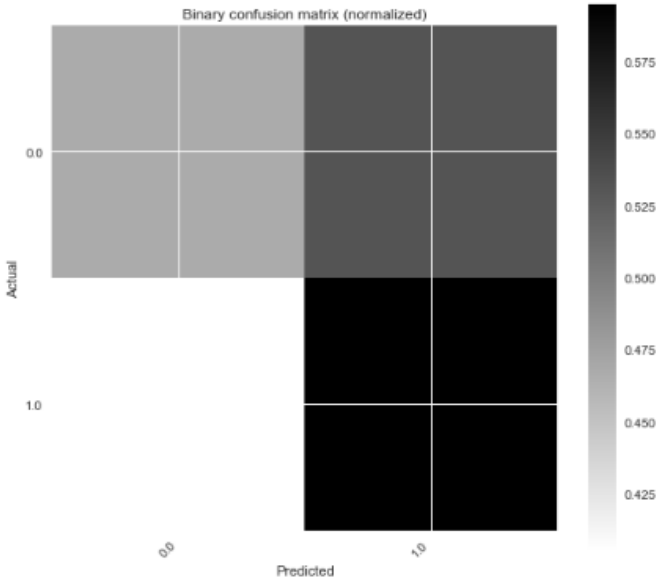
Train on 380962 samples, validate on 93987 samples
Epoch 1/5
380962/380962 [==============================] - 39s - loss: 0.0159 - acc: 0.9948 - val_loss: 8.4765 - val_acc: 0.4683
Epoch 2/5
380962/380962 [==============================] - 37s - loss: 0.0056 - acc: 0.9984 - val_loss: 8.4765 - val_acc: 0.4683
Epoch 3/5
380962/380962 [==============================] - 39s - loss: 0.0047 - acc: 0.9987 - val_loss: 8.4765 - val_acc: 0.4683
Epoch 4/5
380962/380962 [==============================] - 39s - loss: 0.0041 - acc: 0.9989 - val_loss: 8.4765 - val_acc: 0.4683
Epoch 5/5
380962/380962 [==============================] - 37s - loss: 0.0038 - acc: 0.9990 - val_loss: 8.4765 - val_acc: 0.4683

Finally, the performance of this model is evaluated by using t he Binary Confusion Matrix. The confusion matrix is created with usage of ConfusionMatrix() method and also plotted wit h matplot library.

```
confusion_matrix2 = ConfusionMatrix(y2_correct,
y2_predicted)

print("Confusion matrix:\n%s" % confusion_matrix2)

confusion_matrix2.plot(normalized=True)

plt.show()

confusion_matrix2.print_stats()
```

Confusion matrix:

Predicted	0.0	1.0	__all__
Actual			
0.0	43924	49910	93834
1.0	62	91	15
__all__	43986	50001	93987



IV.MODEL COMPARISON

Among the three models, Logistic Regression outperformed well. This is due to the changing of decision boundary with the class weights features. After Logistic Regression, the better one is the Convolutional Neural Networks and the K-Means has the poorest performance. K-Means has the poorest performance because it works with the clustering process and this clustering entirely depends on finding similarities and differences in the features from the dataset and grouping them into clusters. Here for this dataset grouping is a difficult task because fraud and genuine transactions look very similar. So, it is very difficult to put the fraud and genuine transactions into separate groups.

The Logistic Regression gave us the best result among the three models. The accuracy rate for the Logistic Regression model is 99.88% with 0.079% of the validation set being false. Finally, there were the best results for Logistic Regression with balanced weights. The accuracy rate is 97.5% for Logistic Regression with balanced weights.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 04, April-2020

The Convolutional Neural Network model stands next to Logistic Regression in showing better results. For the simple neural network, we got an accuracy rate of 99.61% and the loss rate is 99.87%. So, to decrease the loss we have used multiple inner layers. Here, we have also used Principal Component Analysis (PCA). For these multiple inner layers the accuracy rate is 99.9% and the loss is reduced to 0.037%. Along with this, the validation accuracy is 47.05%.

The K-Means clustering model has the poorest performance on the dataset. Because the dataset we have taken is a classification dataset. We should classify the dataset into 0's and 1's in the dependent attribute named class. K-Means has a low accuracy rate of 54.27%. Of the wrongly predicted transactions, 99.75% were false positives, giving only 0.24% false negatives with 0.11% of the validation set.

Model	**Accuracy**
Logistic Regression	99.88%
K-Means	54.27%
CNN	99.61%

Accuracy Table

V.CONCLUSION

It was clear from the model results that the Logistic Regression model performed well on our dataset. After the Logistic regression model, the next better model is the Neural Networks model because of more hidden layers. K-Means model has poor performance on our dataset. This K-Means model works efficiently on an unsupervised dataset. In addition to our model, there are many other submissions on our project with the Random Forest model is the most common which also works very efficiently. So, the Logistic Regression model works effectively on the credit card fraud dataset with the binary classification process.

VI.REFERENCES

[1] A Comparative Analysis of Various Credit Card Fraud Detection Techniques', Yashvin Jain, Namrata Tiwari, ShriPriyaDubey, SarikaJain, International journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-7 Issue-5S2, January 2019

[2] 'Detection and Prediction of Credit card Fraud Transactions Using Machine Learning', Kaithekuzhical Leena Kurien & Dr. Ajeet Chikkamannur VTU Research Scholar, VTU Research Centre, Bangalore, India 2Professor, Department of CSE & Engg, R. L. Jalappa Institute of Technology, Doddaballapur, Bangalore

[3] 'Machine Learning For Credit Card Fraud Detection System' International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 24 (2018) pp. 16819-16824 © Research India Publications. http://www.ripublication.com

[4] 'Credit Card Fraud Detection Using Machine Learning as Data Mining Technique' Ong Shu Yee, Saravanan Sagadevan and Nurul Hashimah Ahamed Hassain Malim *School of Computer Sciences, Universiti Sains Malaysia, Penang, Malaysia.*

[5] 'Credit card fraud detection using Machine Learning' K. Karthikeyan1, K. P. Sangeeth Raj1, S. Ramaganesh1, P. Parthasarathi2, Dr. N. Suguna3