**Credit Card Fraud Detection Using Random Forest Algorithm and Self-Organizing Maps**

*Sonali Subbu Rathinam,*

*Department of Computer Science,*

*BITS Pilani, Dubai Campus, Dubai International Academic City,*

*PO Box 345055, Dubai*

**Abstract**

The risks associated with credit card related fraud activities indicate that it is immensely significant to detect credit card fraudulent transactions. Early detection and prevention would prevent the loss of a huge amount of money. The aim of this study is to detect credit card frauds using two models. First, is a supervised learning model – the Random Forest Algorithm, and its evaluation metrics and performance are analyzed. Second, is an unsupervised learning model – the Self-Organizing Maps (SOM), is used to obtain a visual representation to detect frauds in credit card transactions, and the overall pattern and risks involved in the dataset are analyzed. The aforementioned methods when implemented, detect potential fraud transactions from the dataset used. These learning methods prove to be useful models for credit card fraud detection.

**Keywords**

Credit card fraud detection, transactions, Random Forest algorithm, Self – Organizing Maps, SOM

1. **Introduction**

The recent years have seen an enormous increase in the use of credit cards for transactions, and they are now an indispensable part of our lives. Additionally, we are in the process of transforming into cashless economy. This increased use of credit cards, due to the advantages it offers, has also witnessed an increase in credit card related frauds. Fraud can be classified as “Any activity with the intent of deception to obtain financial gain by any manner without the knowledge of the cardholder and the issuer bank”. Different types of credit card frauds are also present, such as application frauds, card ID theft, account takeover, phishing etc. Moreover, these types are dynamic and keep changing. Credit card related frauds not only result in losses for the individual card holder, they also result in huge financial losses for the economic institutions as well. Both online and offline credit card transactions are susceptible to credit card frauds. Hence, it is mandatory that fraud detection techniques are continuously implemented to prevent any malicious or illegal activity.

Fraud detection refers to the process of analyzing, observing and characterizing the transaction behavior of a credit card holder, in order to detect and prevent any different or illegal transaction. Fraud detection is particularly hard, as there is limited time to identify a fraud transaction from genuine one, and multiple parameters must be considered during detection of fraud transactions. As large volumes of data need to be analyzed for credit card fraud detection, various analytical techniques and models are used, and the objective of this paper is to analyze two such methods used for this purpose: Random Forest and Self-Organizing Maps.

**I.I Random Forest:** Random Forest, also known as Random decision forest, is an ensemble method based on supervised learning. The random forest algorithm finds its use in classification as well as regression. This algorithm is an aggregation of various different tree predictors, and the result of each respective tree is then combined to obtain the final result, which produces an accurate prediction.

**I.II Self – Organizing Maps:** Self – Organizing Maps (SOM), also known as self- organizing feature map (SOFM), belongs to the unsupervised artificial neural network architecture. It also called the Kohonen map, as it was introduced by Finnish professor Kohonen in 1980. In this method, a high and multi-dimensional input space is projected to produce a discrete and low dimension (usually 2 – dimension) output space. All this is done while maintaining original topology. It serves as an effective and useful visualization technique; its graphical representation makes it possible even for a non – expert to analyze and form conclusions about input dataset. When accompanied with appropriate classification model, SOMs can be used to detect hidden patterns from the input dataset, thus making it an effective model for credit card fraud detection

1. **Literature Review**

Over the past three years, various studies have been conducted and researched in order to analyze the credit card fraud detection techniques, especially the Random Forest and Self Organizing Maps techniques.

In [1], Random Forest Algorithm has been used on credit card transaction dataset. After the dataset is split into training and testing data, the random forest algorithm is applied. The dataset is categorized into four different classifications, which results in the confusion matrix. The study concludes that Random Forest algorithm gave a better accuracy than other existing models for the same dataset.

In [2], two different types of random forests are implemented to train the normal/illegal transactions. The first one is random tree-based random forest – in this, data is distributed by comparing distances between data points and centers. The second one is CART-based (Classification and Regression Trees) random forest – in this data is distributed by finding the attribute to which it has the least Gini impurity.

Random Forest Algorithm is also used in [3]. It provides the conclusion that the random forest algorithm is resistant to over-fitting and produced good estimate of the generalization error. In the experiment recorded in [3], training data set includes credit card data sets and testing data set includes user credit card queries. The RFA algorithm was then applied, and the accuracy of this model was obtained and observed.

In [4], various machine learning methods were used for classifying transactions as fraud or genuine. By comparing various evaluation metrics for the different algorithms, random forest gave the best result. Moreover, it highlights the importance of feature selection and balancing of dataset.

[5] provides insight about supervised and unsupervised methods that are in use for credit card fraud detection. It also highlights major challenges present in identifying and detecting credit card frauds. Various classifiers and the values of their evaluation metrics are observed, after splitting them into training and testing datasets.

In [6], extensive analysis of credit card fraud detections recorded by scientific research has been done. It aims at finding the most suitable technique among the various extant methods (including random forests and self-organizing maps) based on their accuracy, coverage and costs. It also provides insights about the most commonly observed characteristics of fraudulent transactions.

[7] provides useful information about the different types of credit card frauds encountered. After analyzing various detection models, it also highlights the gaps and limitations of these models, and future scope for the same

In [8], the pros and cons of fraud detection methods using supervised and unsupervised learning is observed. Comparison between the supervised and unsupervised learning methods is done by analyzing each one’s performance metrics.

In [9], apart from the performance metrics of various detection models, the impact they have and their scope for industry use is discussed. It uses additional performance metrics such as Matthews Correlation Coefficient (MCC) and Receiver Operating Characteristic (ROC) curve.

In [10], Self- organizing maps (SOM) model is used to detect credit card frauds. Large number of iterations and appropriate color schemes are used for the same. Data is categorized into Independent and dependent variables, and these properties are used to produce the desired outcome of designing SOM. SOMs use competitive learning instead of error-correction learning.

1. **Methodology**

**III.I Random Forest**

In order to detect potential credit cards frauds from the dataset, the random forest algorithm is used. Firstly, the dataset is loaded onto Google Colab and then the required packages for the algorithm is imported. In order to obtain a graphic and visual representation of the dataset, a co-relation matrix is used. It will show how the features are co-related to one another and can then identify the features that would be most relevant and appropriate for our fraud detection and prediction. The data values are then divided into Feature and Targets. Then, the dataset is split into training and testing data, where it’s split as 80% training and 20% testing. The random forest classifier is then built and used for detection of frauds in our dataset. The results of our detection are then printed for our view and use. In order to find the reliability of our model used, values of evaluation metrics (namely accuracy, precision, recall, F1-score and Mathews Correlation Coefficient) are found. To understand the working of our model better, a confusion matrix and a random single tree from the forest is viewed. A visual representation of our random single tree from the forest will help us better understand how the algorithm is taking decisions.

The random forest algorithm used in this study is imported from the sklearn library. For the splitting purpose, the default Gini index is used. The Gini index formula is as follows:

(1)

where Pi is the probability of element being classified to a particular class

Following are the equations of the performance metrics used:

Accuracy = (2)

Precision = (3)

Recall = (4)

F1-Score =

Mathews Co-relation Coefficient = (5)

**Block Diagram for Random Forest Algorithm**

**2. Import Necessary packages**

1. **Loading the Dataset**

**9. Visualize the confusion matrix and a random single tree of the random forest**

**8. Find the values of evaluation metrics of the Random Classifier Model**

**7. Print the results of the model used to detect frauds**

**6. Create and Train Random Classifier Model**

**5. Split the dataset into 80% Training and 20% Testing**

**4. Divide Data values into Feature and Target**

**3. Use of Co-relation Matrix**

*Figure 1: Block Diagram for Random Forest Algorithm*

**III.II Self – Organizing Maps**

In order to obtain a visual representation of credit card frauds, self-organizing maps are used. Firstly, the dataset is loaded onto Google Colab and then the necessary libraries are imported. The dataset is then split into X and Y, where X contains all but last column, and Y contains only the last column. The X dataset alone is used for SOM training. Then, feature scaling is done on the dataset for normalization. For implementing SOM, Minisom class is imported. Then, an object SOM of class Minisom is created. The values used in the parameters of object creation are as follows:

x = 10, y = 10 (this will produce a 10x10 grid) len = 15 sigma default value = 1 learning rate default value = 0.5

Once the object is created, weight vectors are initialized using the random\_weights\_init() method. It ensures that each weight w follows the condition .

The model is then trained using train\_random function: In this function is where the SOM is trained and implemented. A random observation is selected from the dataset. The Euclidean distance from this point to the different neurons in the network are computed. The formula for Euclidean distance is as follows:

Distance = (6) where V is the input vector and W is the node selected.

The winning node (also known as Best Matching Unit (BMU)), which is the node that has the minimum distance to the point, is selected. Weights of winning node is then updated. Then, the weights of winning node neighbors are updated using Gaussian neighborhood function. Sigma is the neighborhood radius value. This process is iteratively done and weights are updated accordingly, until the point where neighborhood stops decreasing. Following is the formula used for shrinking the sigma value over time:

, t = 1, 2, 3 (7)

width of lattice at time t0

λ time constant

t current time-step

Weights of nodes in the BMU neighborhood is updated according to the following equation:

(8)

t time-step

amount of influence a node’s distance from BMU has on its learning = (9) dist distance of node from BMU

L Learning rate

Decay of Learning rate after each iteration is as follows:

(10)

Then, the necessary functions to plot and visualize the SOM from pylab are imported. The different colors used correspond to different range values of the Mean Interneuron distance. With this, the SOM can be visualized, and outliers can be identified. In order to detect frauds, mapping function is used. New variable frauds is initialized to co-ordinate of outlying winning nodes. These co-ordinates are concatenated and then initialized in frauds. Inverse transform method is applied to the list of frauds and then potential frauds are detected.

**Block Diagram for Self-Organizing Maps**

1. **Import necessary libraries**
2. **Loading the dataset**

*Figure 2: Block Diagram for Self – Organizing Maps*

**9. After mapping function, inverse mapping function is applied to list of frauds for detecting potential frauds**

**8. Visualize SOM, where different colors correspond to different range of values of Mean Interneuron Distance**

**7. Train the model using train\_random method**

**6. Initialize weight vectors using random\_weight\_init() method**

**5. Import Minisom Class, and create object SOM with appropriate parameters.**

**4. Feature Scaling for normalization**

**3. Split the dataset into X and Y, X alone will be used for SOM training**

1. **Experimental Setup**

The experiment and its results were obtained from the following experimental setup. The dataset used for credit card fraud detection has 690 entries, from which the potential frauds were experimentally detected. The dataset has 14 attributes, of which 6 are numerical and 8 are categorical attributes. The last attribute (15th attribute) is the class attribute having either binary value, i.e., 0 or 1. The experiments were conducted on Google Colab using Python programming. In both the experiments conducted, the dataset was imported and read, and the requisite libraries were imported. A detailed description of the experimental setup for the individual experiments conducted are as follows:

**IV.I Random Forest Algorithm:** To implement this supervised learning model for detecting our fraudulent transactions and customers, a co-relation matrix was used, after importing the necessary libraries and the dataset. After the data values are divided into Target and Feature, the dataset is split into 80% training and 20% testing. For this, scikit-learn was used. The Random Classifier model was then created and trained. To ensure the reliability and utility of the model, the following evaluation metrics are used: Accuracy, precision, recall, F1 score, Matthews correlation coefficient. The confusion matrix for the model was also used. Appropriate packages from sklearn.metrics were imported for the evaluation metrices and confusion matrix. In addition, a random single tree from the forest was displayed to further develop understanding of the algorithm, and its result and working. For this, Image(from IPython.display), export\_graphviz(from sklearn.tree) and pydot were imported.

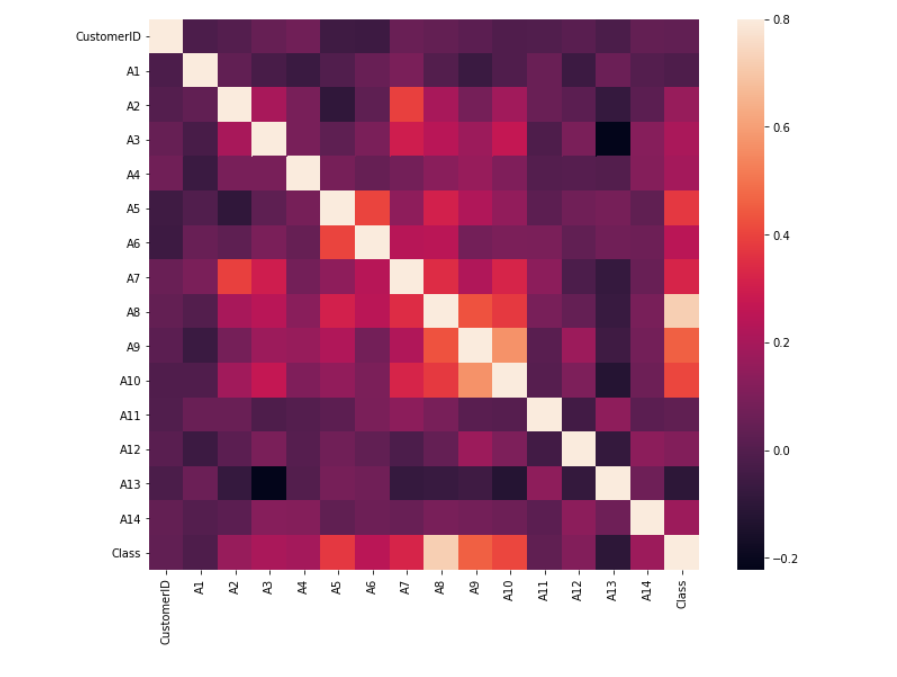
**IV.II Self – Organizing Maps:** For our unsupervised learning method, the MiniSom package is first installed, as it is a numpy-array based implementation of SOMs. In order to perform feature scaling for normalization, MinMaxScaler was imported from sklearn. This was followed by creating an object of MiniSom class, and the training of SOM was performed. For visualizing the SOM trained, functions were imported from pylab, and markers were added to analyze if the transactions were approved or not. Green square mark indication for approved transactions, and red circle mark indication for unapproved transactions have been used. After visualizing and analyzing the resultant SOM, the co-ordinates of the selected winning nodes are used in the inverse mapping function. From this, the fraud client customer IDs are printed.

1. **Experimental Results**

The analysis and results obtained from the dataset and the models used are as follows:

**V.I Random Forest Algorithm**

Figure 3 is the co-relation matrix of the dataset, which helps us understand our dataset better, and how associated and how positively or negatively co-related our data values are. As observed in the figure, our data values are mostly closely co-related to one another. For example, A3, A10, and A12 are highly negatively co-related to A13.



*Figure 3: Co – relation Matrix of the dataset used*

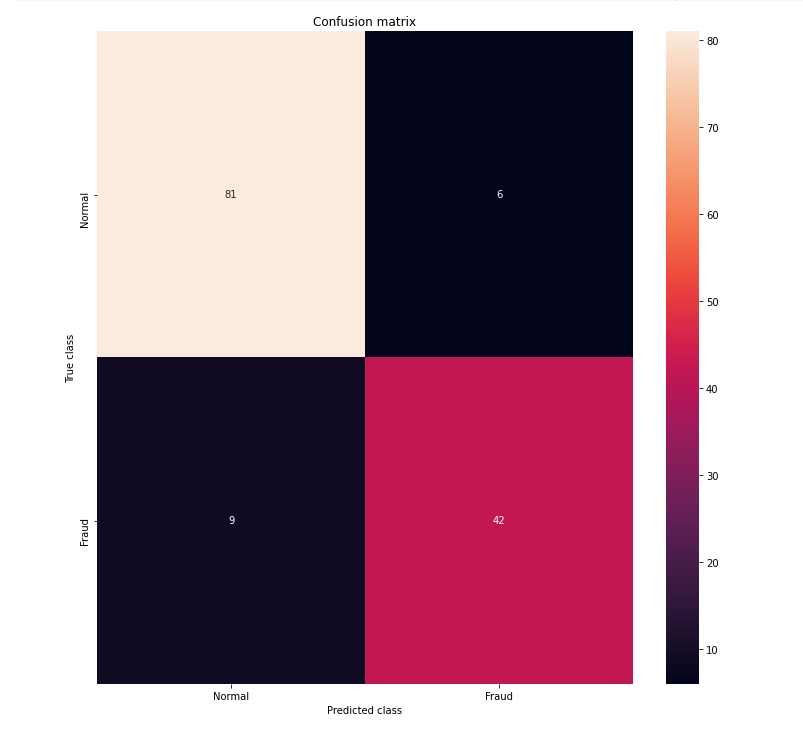
The model was created and trained, and the evaluation metrics were calculated and displayed to support the utility of this model. Table 1 provides the values for the evaluation metrics of the model.

*Table 1: Evaluation Metrics of the Random Forest Algorithm*

|  |  |  |
| --- | --- | --- |
| **S.no** | **EVALUATION METRIC** | **VALUE** |
| 1 | Accuracy | 0.891 |
| 2 | Precision | 0.875 |
| 3 | Recall | 0.823 |
| 4 | F1 Score | 0.848 |
| 5 | Matthews Co-relation Co-efficient | 0.764 |

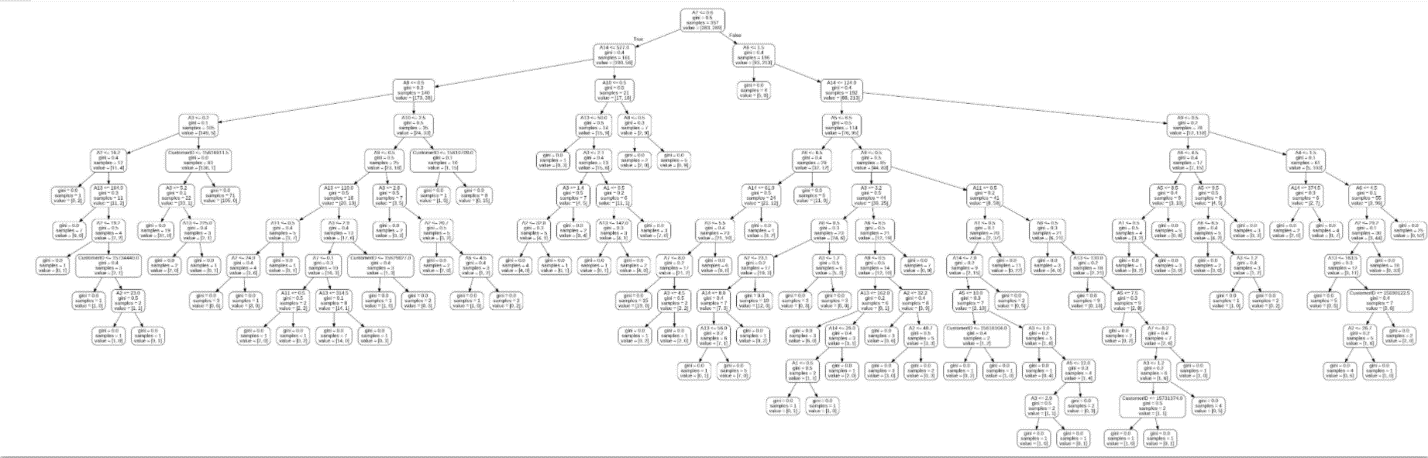
The accuracy of the algorithm used is 0.891, which means that the model is approximately 89.1% accurate. The precision rate recorded,0.875, is pretty good, as a high precision rate relates to low false positive rates. The recall value, which is the ratio of correctly predicted positive outcomes to all the correctly predicted values, is observed to be 0.823. The F1 score, which is the weighted average of precision and recall, is 0.848. As the values for accuracy, precision, recall and F1 score are fairly high, the Matthew’s Co-relation Co-efficient is also high enough, at a value of 0.764. As this value is closer to 1, it indicates that both the classes of the confusion matrix have been predicted well.

Figure 4 shows the confusion matrix of the algorithm used. From this, we observe that the number of true positives, which is the number of normal transactions correctly predicted as normal is 81. The number of false negatives, which is the number of normal transactions incorrectly predicted as fraud is 6. The number of false positives, which is the number of fraudulent transactions incorrectly predicted as normal is 9. The number of false positives, which is the number of fraudulent transactions correctly predicted as fraud is 42. These values indicate that our algorithm performs well, and incorrectly predicts only a few values from the whole set used.

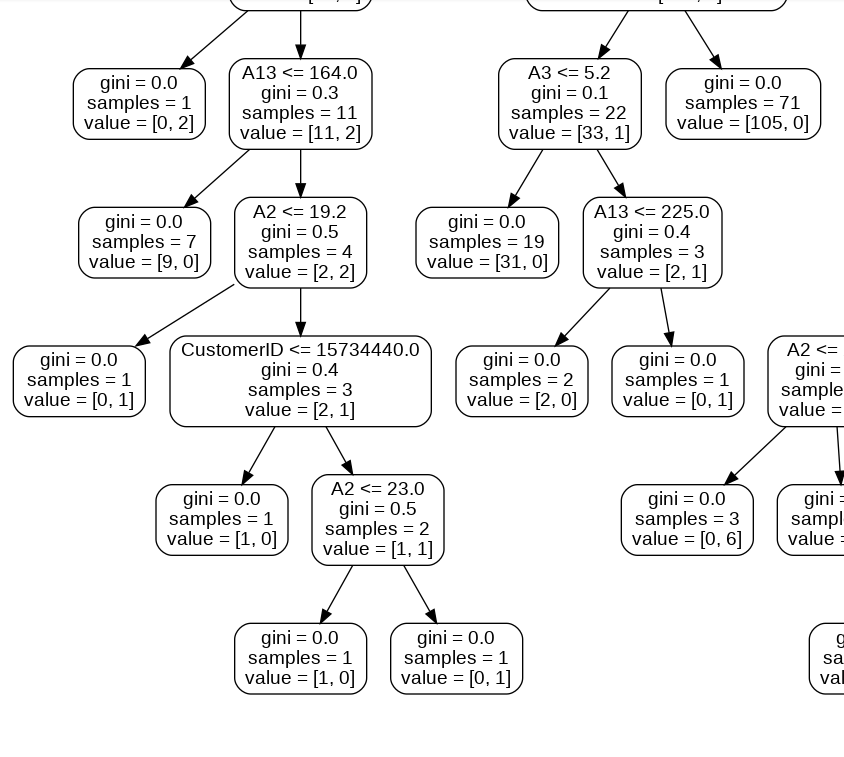


*Figure 4: Confusion Matrix of the Random Forest Algorithm*

In order to enhance our comprehension and understanding of the Random Forest Algorithm, a random tree from the forest was visualized, as in Figure 5. Figure 6 is a closer representation of the tree. From these figures, we observe that Gini impurity is used as splitting criterion, and that leaf nodes at certain instances may have more than one sample.



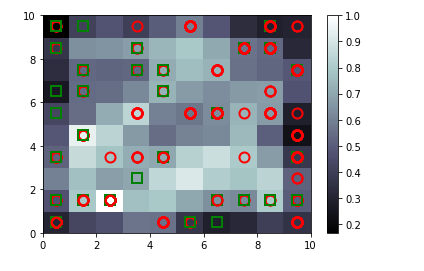
*Figure 5: Visual Representation of a random tree from the forest*



*Figure 6: Closer Visual Representation of the random tree in Figure 5*

**V.II Self – Organizing Maps:**

Figure 7 is the Self-Organizing Map obtained from the experiment conducted. From this map, we observe that the co-ordinate (3,2) is an outlier, having the lightest(whitest) value, indicating that it has very high potential to be a fraudulent transaction, as given in the color scheme of the SOM. For the purpose of the experiment, only one node/co-ordinate was chosen. However, more than one can also be chosen, and the threshold for choosing the number of nodes is subjective, and depends on the needs and interests of the user. The co-ordinates (3,2) were then used in inverse mapping function, and the potential fraud customer IDs are printed. Another important observation made from this SOM is that, a particular node can have both approved and unapproved transactions, as it can be seen that certain nodes have both green squares and red circles (e.g. (3,2), (5,8) etc.). Hence, an approved transaction also poses the risk of being a fraudulent transaction. Additionally, the Self-Organizing Map provides an overall pattern of how transactions occur, and observations and insights derived from these patterns can be later used for prevention of fraud transactions as well.



*Figure 7: Self – Organizing Map*

1. **Conclusion**

The objective of this study was to experiment and analyze the implementation of Random Forest Algorithm, a supervised learning model, and Self-Organizing Maps, an unsupervised learning model, to detect fraudulent credit card transactions. Detecting credit card frauds is vital and significant, as its negative effect and impact are tremendous. In the random forest algorithm, the dataset was split into 80% training and 20% testing, after which the random classifier model was trained and used. Analysis of the evaluation metrics of the model used indicate that it performs well, and detects transactions that possess a high risk of being a fraud transaction. Self-Organizing Maps provides an easy to comprehend visual analysis of the dataset, and using mean interneuron distance, detects credit card related frauds. SOMs help us identify the transactions that higher risks compared to others, due to the color scheme formed from the interneuron distance. This can help firms and banks act on those with higher risks immediately, and this possibility is non-existent in the supervised learning model used in this experiment. Various other learning models can also be used for credit card fraud detection, two such models have been implemented, and its results have been analyzed in this study.

**References**

1. M. Suresh Kumar, V. Soundarya, S. Kavitha, E.S. Keerthika and E. Aswini, “CREDIT CARD FRAUD DETECTION USING RANDOM FOREST ALGORITHM”, [2019 3rd International Conference on Computing and Communications Technologies (ICCCT)](https://ieeexplore.ieee.org/xpl/conhome/8811520/proceeding)
2. Shiyang Xuan, Guanjun Liu, Zhenchuan Li, Lutao Zheng, Shuo Wang and Changjun Jiang, “Random Forest for Credit Card Fraud Detection”, [2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC)](https://ieeexplore.ieee.org/xpl/conhome/8358983/proceeding)
3. S. Monika, K. Venkataramanamma, P. Pritto Paul, M. Usha, “Credit Card Fraud Detection using Random Forest Algorithm”, [2019 3rd International Conference on Computing and Communications Technologies (ICCCT)](https://ieeexplore.ieee.org/xpl/conhome/8811520/proceeding)
4. Dejan Varmedja, Mirjana Karanovic, Srdjan Sladojevic, Marko Arsenovic and Andras Anderla, “Credit Card Fraud Detection - Machine Learning methods”, [2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH)](https://ieeexplore.ieee.org/xpl/conhome/8713398/proceeding)
5. Naresh Kumar Trivedi, Sarita Simaiya, Umesh Kumar Lilhore and Sanjeev Kumar Sharma, “An Efficient Credit Card Fraud Detection Model Based on Machine Learning Methods”, International Journal of Advanced Science and Technology Vol. 29, No. 5, (2020), pp. 3414 – 3424
6. Elena-Adriana Minastireanu and Gabriela Mesnita, “An Analysis of the Most Used Machine Learning Algorithms for Online Fraud Detection”, Informatica Economică vol. 23, no. 1/2019
7. Yashvi Jain, Namrata Tiwari, Shripriya Dubey and Sarika Jain, “A Comparative Analysis of Various Credit Card Fraud Detection Techniques”, International Journal of Recent Technology and Engineering (IJRTE) Volume-7 Issue-5S2, January 2019
8. Xuetong Niu, Li Wang and Xulei Yang, “A Comparison Study of Credit Card Fraud Detection: Supervised versus Unsupervised”, Association for the Advancement of Artificial Intelligence 2019
9. Nick F. Ryman-Tubb, Paul Krause and Wolfgang Garn, “How Artificial Intelligence and machine learning research impacts payment card fraud detection: A survey and industry benchmark”, Engineering Applications of Artificial Intelligence 76 (2018)
10. E. Saraswathi, Prateek Kulkarni, Momin Nawaf Khalil and Shishir Chandra Nigam, “Credit Card Fraud Prediction And Detection using Artificial Neural Network And Self Organizing Maps”, [2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)](https://ieeexplore.ieee.org/xpl/conhome/8811524/proceeding)

[www.towardsdatascience.com](http://www.towardsdatascience.com)

[www.medium.com](http://www.medium.com)