International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-7, Issue-6, March 2019

Application of Big Data Analytics and Pattern Recognition Aggregated With Random Forest for Detecting Fraudulent Credit Card Transactions (CCFD-BPRRF)

B.J.Jaidhan, B. Divya Madhuri, K. Pushpa, B.V.S Lakshmi Devi, Shanmuk Srinivas A

Abstract: ***People today tend to make multiple transactions every day. It has been observed that around 150 million transactions are being carried out every 24 hours. There are several modes through which these transactions can be accomplished, but amongst them, credit-based transactions stand ahead. Using credit system for negotiations is worthwhile for both the users and the credit providers. But with the advent of newer methodologies, illicit usage of the credit system has been growing. This situation seems like a stumbling block for both the users and the credit providers. In this pursuit, Big Data provides better and utilitarian methods and algorithms to overcome this snag. Big Data in this context helps in building an analytical model that can be integrated with Hadoop for storage and is feasible to implement pattern recognition algorithms that are aided by few machine learning algorithms to predict fraudulent patterns. This paper reflects that our proposed model comes withhigher accuracy rates when compared to the other existing decision making models.***

Index Terms: Big Data, Credit Card Fraud,Classification, Machine Learning Algorithms, Pattern Recognition, Random Forests, Supervised Learning.

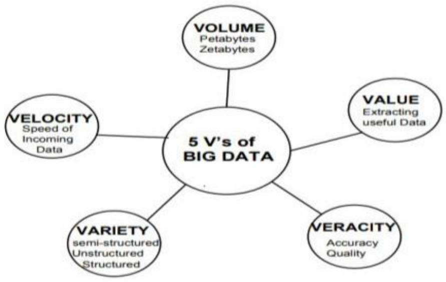
I. INTRODUCTION

These days credit cards have relatively trimmed down the effort involved in making transactions. Furthermore online credit card transactions have made the job much more easier, however fraudulence has subsequently been increasing.Consequently massive amounts of loss has been incurred in many economies in the recent years. Credit card frauds can be categorized into two sorts:

Application frauds: These kinds of frauds fall under the scenario wherein the fraudster intends to apply for the credit card portraying a false profile or by using some legitimate user's details.

Revised Manuscript Received on 30 March 2019.
***** Correspondence Author
Dr B J Jaidhan*, Associate Professor, Department of CSE, GITAM University, Visakhapatnam (Andhra Pradesh), India.
Shanmuk Srinivas Amiripalli, Assistant Professor, Department of CSE, GITAM University, Visakhapatnam (Andhra Pradesh), India
B. Divya Madhuri: Gandhi Institute of Technology and Management, University Visakhapatnam, (Andhra Pradesh), India.
B.V.S.Lakshmi Devi: Gandhi Institute of Technology and Management, University. Visakhapatnam, (Andhra Pradesh), India.
K. Pushpa, Gandhi Institute of Technology and Management, University. Visakhapatnam, (Andhra Pradesh), India.

Behavioral frauds: In such kinds of frauds the fraudster tends to plagiarize the credentials of some authorized credit card holder and loot him off his money.



Generally huge streams of data bits generated due to numerous transactions have been continually pressurizing the credit management systems, due to which the efficiency of the algorithms running on these systems aren't able to handle such large streams of incoming big data which can be characterized by the 5 V's – "Volume", "Velocity", "Veracity", "Value", "Variability" [2].

Figure 1: *5V's of Big Data*

As a result, filtering out the fraudulent patterns from such large piles of transactions becomes highly complicated. The fraudsters tend to transfigure the CCFD systems. Hence we need to equip these systems with greater reliability and highly sophisticated techniques to enhance the accuracy of the credit card fraud detection systems (CCFD). Categorically the detection mechanism can be labelled as two detection methods [11];

(i) Misuse detection
(ii) Anomaly detection

The potential problem of the credit card fraud can be resolved by using any of the several existing approaches that include Machine learning, AI, Data Mining techniques and hand-fully a few others. In this paper we have integrated Big data analytic framework with pattern recognition model. The Machine learning tools aid the pattern recognition schemes as pattern recognition algorithms alone cannot carry out the fraud detection mechanisms efficiently. Machine Learning algorithms have been widely known for their scalability and efficient working. The "MLib" tool that is a part of the Hadoop Ecosystem within the Big Data analytic framework consists of several Machine Learning algorithms that fit in to solve the fraud detection problem include Linear Regression, Logistic Regression, Decision Tree, Naive Bayes, KNN, K-means and Random Forest Decision Tree [5].

However not all of these algorithms provide the optimum results upon implementation. It is noteworthy to consider the cost of detection also. The prime rule is that the mechanism with least cost and authentic results is most likely to be chosen. This paper discusses about the working of the Random Forest Decision Tree algorithm that has outwritten the short-comes of the other Machine Learning Algorithms[22]. The Random Forest Algorithm turns out to be much more robust and efficacious due to some of it's featuressuch as:

- Handling larger datasets withhigher dimensionality.
- Handling the missing data values and maintains accuracy for missing data.
- Performs both classification and regression tasks.
- Doesn't overfit the model.
- We shall discuss in detail the architecture and working of the entire model in the forthcoming sections of this paper.

II. RELATED STUDY

In [1] the authors have made a study and showed that by the end of 2020, digital monetary losses would increase by double rates. Frauds are rampant these days and subsequently many methodologies have been proposed to resolve these issues. The authors in [2] have listed four big data techniques corresponding to the Hadoop framework: Apache Hadoop, Apache MapReduce, Apache Spark, Apache Flink. The results apparently show that based on some metrics that were used to measure the efficiency of these tools, Apache Spark gave the maximum efficiency. In [3] the authors have made an attempt to work on anomaly detection by implementing Graph Data Model based on Apache Lucene framework. The authors have tried highlighting the pros and cons in the current fraud detection systems. In [4] the authors have worked on the concept of One-Classification. This concept was developed based on a hybrid model (combination of Particle Swarm Optimization, Auto Associative Neural Network). This model works on Apache Spark framework. And this model has apparently shown 89% optimum results.
The authors in [5] have outlined the idea of coupling machine learning algorithms such as Decision Tree, KNN, Logistic Regression and Neural Networks. Based on three metrics that were taken into consideration, the authors landed onto a conclusion that not one single algorithm is potent enough to handle the scenario. Combinations of algorithms can provide expected output greater precision and accuracy. In [6] the authors have specifically made a comparison between three machine learning algorithms namely Logistic Regression, Decision Trees and Random Forest and evaluated them using confusion matrix. The experimental output shows that Random Forest algorithm has ranked first in terms of accuracy, precision and recall. Paper [7] has made a comparative model based on Artificial Neural Network (ANN) and a model based on Logistic regression. Both the models work based on suspicion score. Experimental results show that the model based on ANN generates much optimum results as compared to Logistic Regression. In [8] the authors have made an analysis comparing the Naïve Bayes algorithm and KNN algorithm. They have stated that running and implementation of single algorithms may put us into iteration and optimization problems. Hence implementing these algorithms would be the most feasible way to deal with fraud detection. In [9] the authors have worked on Support Vector Machine (SVM) along with Decision Tree algorithm. They have concluded that SVM is more likely to generate imprecise results. Hence SVM has to be coupled with Collective animal behavior to enhance the accuracy. The authors in [10] have implemented the Decision Tree algorithm single handedly. The proposed model works with Luhn's algorithm, Pattern matching, and Bayes theorem. The results speak of the authenticity in the correctness and effectiveness of the proposed system. The authors in [11] have shown the working of two types of Random Forest algorithms (Random-tree based and CART based). Here these algorithms work optimally but still there are a few liabilities within the Random Forest approach that need to be addressed. In [12] the author has mentioned several approaches for solving several machine learning approaches along with the performance metrics. They have proved that among all the algorithms random forest have got the highest percentage of performance by calculating the metrics. In [13] the authors have described that random forests work as classifiers. In paper [14] it has been said and proved that ensemble methods solve the decision problem as they combine individual decisions into one single and final decision. [15] shows that a base classifier of random forest is more or less an implementation of decision tree, which is termed as random tree. In [16] The authors have highlighted the simplicity of the decision tree algorithm. [17] paper shows the efficiency of the random forest algorithm due to it's randomly chosen attributes and dividing the nodes using best split approach.

III. PROPOSED METHODOLOGY

A. An Overview

In this experiment we have applied the concept of Pattern Recognition. Pattern recognition is basically a trained mechanism to recognize patterns and anomalies in the given data set using several approaches. [18] Based on the identified patterns, a classification is made and finally we arrive at a decision. As mentioned above there are several approaches towards decision making while holding on to the concept of pattern recognition all through. Machine learning is one among them. The Machine learning approach is nothing but mapping the input to predefined class labels so as arrive at a decision. When the pattern matching systems are trained based on the previously trained data, we coin the approach as Supervised Learning[19]. When there is no previously trained data available to train the system, we coin the approach as Unsupervised learning. As we have understood that the raw data pertaining the customer's credit card transactions are huge, surely in millions because transactions are carried out every single second all across the terrain of the earth, so we have applied the concept of Big Data Analytics to preprocess the data and tailoring it into a form that is decent enough for the pattern recognition algorithms to run on. For the purpose of storing and preprocessing the big data we employ Hadoop and Spark frameworks that run in a conjunction.

B. Apache Hadoop and Apache Spark

Within the Big Data analytic framework Hadoop is the basic tool that can be used for both storage as well as processing of data. In Hadoop large collections of datasets can be stored and processed in a parallel and distributed fashion. Hadoop within itself accommodates tools for both storage and data processing separately. HDFS [Hadoop Distributed File System] allows us to dump any format of data across the cluster[20]. HDFS is capable of storing relatively bulky amounts of data. The storage cluster follows a "Master-Slave" architecture. MapReduce allows parallel processing of data across the HDFS. MapReduce helps in processing the data at swift rates in a distributed fashion. MapReduce avoids congestion of data packets within the Hadoop cluster. It keeps the network clear and scalable. Hadoop does a great deal in Batch data processing. But Hadoop isn't good enough for real-time processing. Fortunately we have got better tools for performing the task of preprocessing, just like Apache Spark [2]. Within the analytical framework lies another tool called the Apache Spark. Spark is an open source cluster computing framework for real-time data processing. Working on Spark platform has trimmed down the edges of tedious tasks as Spark provides an interface for parallel programming in the clusters. But Spark hasn't got any proper storage systems. Hence the idea of integrating Spark with Hadoop is trustworthy. This coupling has proved to have provided the best forms of data output that is suitable enough to run our algorithms on. In the current scenario the incoming live transactions are dumped across the HDFS where they are stored in a cluster format. The Spark Shell that is aggregated to Hadoop gets access to the stored data through the "spark context". The "cluster manager" within the shell allocates resources responsible for processing those transactions[21]. Finally this data is sent to the "worker node" where the final execution is done and the spark framework is now ready with the pre-processed data that is fit for applying the fraud detection algorithms.

C. Pattern recognition

The pre-processed data is now sent to the pattern recognition block [8] where based on the previously available user's transaction pattern description, the newly obtained transaction pattern is attempted to map to the defined class labels. This mapping is usually done through Classification. We generally use Machine Learning algorithms to do the classification. In this paper we have used Random forest algorithm to do the classification.

D. Supervised learning

Supervised learning is a Machine learning task which is responsible for mapping the input to output based on the learning function. Supervised learning task always occurs in the form of input-output pairs. The input is mapped to the output based on class labels. Supervised learning performs classification and regression [12]. When we are provided with known number of possible outcomes to be made a decision from, in such cases we can apply classification. When we are unaware of the possible outcomes, then we apply regression. Both classification and regression help in decision making[23]. In this experiment we perform classification as the data set (Table 1) that we choose for fraud detection is trained based on the user's previous transaction record and that we are aware of the possible outcomes, i.e fraudulent or non-fraudulent.

E. Random Forest

Random Forest refers to construction of a mass of decision trees with randomly chosen features as the root nodes. Hence it is referred to as "Random Forest" [11]. By calculating the mode of the outputs derived from each individual tree, the final output is obtained, which is assumed to be the final decision. Here in this case we use random forest as a classifier. Random forest comes under the ensemble methods which generally overcomes the short-comes of other classification algorithms. Random forest has the potential to deal with noise and outliers within the dataset. And that it is resistant towards overfitting the model. Here each tree is trained individually based on the randomly chosen samples and decision tree is built. Furthermore each decision tree is extended based on randomly chosen features out of all the features present in the dataset. Since each tree is trained individually, that is the reason why random forest still remains so robust and efficient even if the features to choose are relatively more in number.

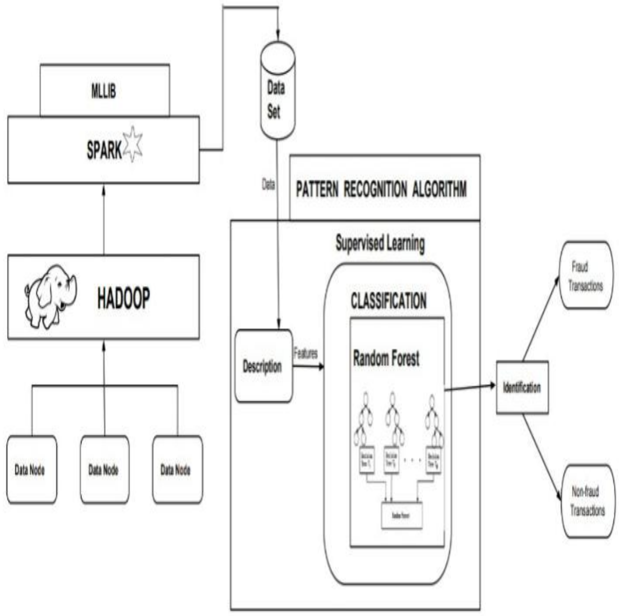


Figure 2 : *Architecture of CCFD-BPRRF*

F. Pattern Recognition in Fraud Detection:

Pattern recognition is branch of science that can work alongside any conventional method for fraud detection where the user himself has to train the system but in this paper we have collaborated Pattern recognition with machine learning. Using machine learning the system trains itself based on the predefined stipulations laid by the user. A learning function is generated within the system such that the input training data is mapped to the predefined class labels. Here the two class labels are – fraud and non-fraud. Hence based on the learning function the input mapping gives the final decision.

G. Algorithm: Input: Dataset (Table 1)Output: Labelled Transactions

H. Implementation details:

Random forest as already discussed is a collection of n number of decision trees with each tree trained individually by randomly picking the attributes. The algorithm for fraud detection can be designed in two steps.

(1) Random forest construction:

If we consider a dataset with "m" attributes in total, for each decision tree we adopt "k" attributes, where k<<m. Among the chosen "k" attributes, the best split point has to be calculated so as the put one of those attributes as the root node.

The best split point can be calculated using three parameters

- Information gain
- Entropy
- Gain

The values for these three parameters have to be calculated for every attribute in order to determine the root node for the decision tree.

Information gain:

Information gain can be typically defined as the information acquired about a particular random variable by observing another variable. Mathematically it can be represented as;

$I(Ti,Fi) = -T/T+F \log2(T/T+F) – F/T+F \log2(F/T+F)$

Entropy:

Entropy is used to calculate the homogeneity of the taken sample. If the sample is entirely homogeneous the entropy would be 0. If the sample is equally divided then the homogeneity would be 1.

$E = \Sigma (Ti+Fi)/T+F (I(TI,Fi)$

Gain:

Gain is nothing but the difference between the Information gain and the entropy. The attribute with the highest gain will be declared as the root node.

G = Information gain – Entropy.

Further splitting of the node into daughter nodes is done using the best split method. The formerly mentioned steps are repeated until we reach "i" number of nodes. The entire forest is built by iterating the same steps for "n" number of times to construct "n" number of trees.

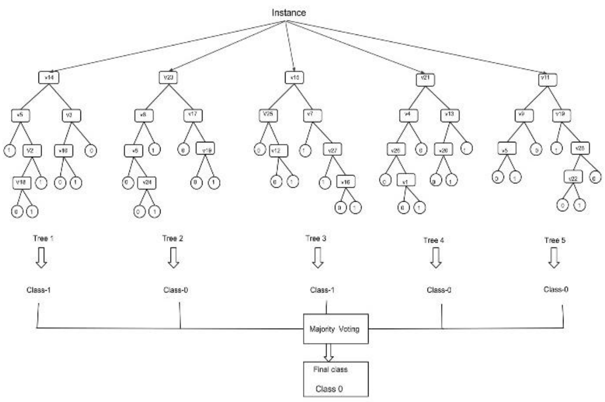


Figure 3: Illustration of Random forest for credit fraud detection

(2) Random forest prediction:

We consider the test features and by applying the rules of each randomly developed decision tree, the outcome is predicted and the value is stored. Eventually the votes for each predicted class label.

Algorithm for BPRRF

```
1. Begin
2. int NF, F
3.     RFUP ( )
4. Consider the previously generated patterns.
5. Map the incoming transaction with the corresponding
     previous pattern.
          for (q=1 to n) do
           for (j=1 to i) do
6.  Randomly select K features from m features (k<<m).
7.  Using "best split point" method find out the root node
     among the k features.Consider the test features and
     rules used to train each tree.
8.  The individual result generated by each decision tree is
     stored for final computation.
9.  From the stored values perform voting for each
     predicted target.
10.  for (t=0; t<p; t++)
                 if next stored value=0
                        F++
                  else
NF++
                  end if
            end for
11.  if (NF>F)  then
                  return (Non- Fraud)
           else
     return (Fraud)
           end if
12.        end for
                end for
13. end
```

IV. EXPERIMENT AND RESULT

Here we have considered a sample data set containing credit transactions. The table consists of 284808 rows and 30 columns. Our data set consists of 28 attributes and 2 class labels (0 and 1).

0 – fraud; 1- nonfraud

Sno	V1	V2	V3	V4	V5	......	V28	CLASS
1.	-2.2646	-0.70758	1.080243	-4.65755	-0.15718	......	0.615339	0
2.	2.65455	-0.23665	-1.12556	3.551564	-0.52219	......	-0.14328	1
3.	-1.26555	-3.65566	2.236585	0.145265	0.48937	......	-0.99379	0
4.	1.55866	-2.65555	1.266985	2.265874	1.359805	......	0.35764	1
5.	-2.35255	2.5556	2.265452	0.645879	0.328555	......	0.031249	0
6.	5.355555	0.555555	-236555	-0.58479	1.500273	......	-6.32854	0
7.	-2.36545	2.26665	0.36555	-2.6548	0.445444	.....	0.003211	0
.	.	.	.	.	.	.	.	.

Table 1: Sample Credit Card Transaction Data Set

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

A. Bias-Variance trade-off:

For any decision making model to work without any errors it is necessary to maintain a balance between Bias and Variance. Any ideal model would recommend low bias and low variance values but it is hard to achieve in real time because both bias and variance are inversely proportionate values. So the real task is to design the model in such a way that it satisfies the laid criterion (low bias and low variance).

To achieve low variance:

- training different samples of data.
- Using random subsets for training the data.
- Including several features within the dataset.

To achieve low bias:

- The model must be flexible with the taken dataset.

Our aim is to reduce the total error that is expected to occur due to presence of bias and variance in our model. However there is some amount of noise present in the dataset which cannot be reduced because no model is expected to provide 100 percentage accuracy. Hence such errors are termed as irreducible errors. Theoretically total error can be calculated as:

Total Error = (Bias error)2 + Variance error + Irreducible error

So the task is to free our model from bias and variance errors. Hence by fulfilling the above discussed criterion with the laid obligations error can be cut down to a much larger extent.

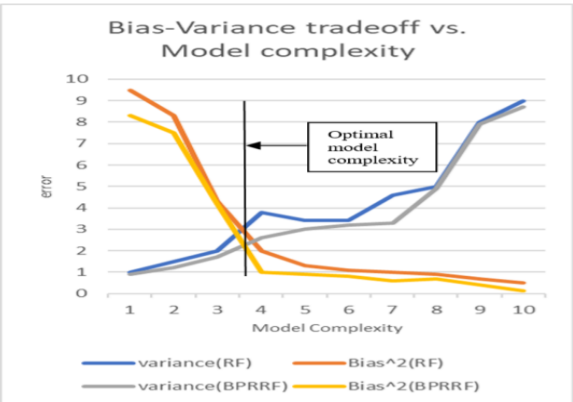


Figure 4: Graph comparing the bias-variance trade-off of random forest and BPRRF.

The above graph shows that variance and bias are inversely proportional and that at a particular point where both the values are low, gives the least amount of error and the model works at it's best at this particular point.

Random forest being a enemble technique is designed in such a way that it shows off low bias and low qvariance values due to it's randomness in selecting the attributes and training large number of samples in the data.

B.Metrics to Evaluate the Model:

METRICS	FORMULA
Accuracy	$1/n\ \Sigma^{n}_{n=1}(y_i - f^{n}(x_i))$
Precision	*TP/TP+FP*
Sensitivity	*TP/TP+FN*
Specificity	*TN/FP+TN*

Where TN= True Negative

TP= True Positive

FN= False Negative

FP= False Positive

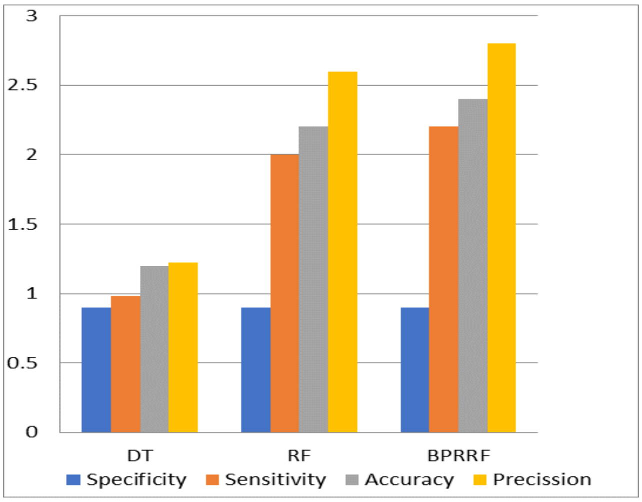


Figure 5: Graph comparing the metrics of several decision-making models

- In the above shown graph X-axis represents the various models (Decision Tree, Random Forests, proposed model-BPRRF) and the Y-axis represents the metric values.
- The final experimental results show that the proposed model (BPRRF) that uses Random forest for decision making has got the highest values (although the values are close to RF) in all the four mentioned metrics. Hence our proposed model (BPRRF) is justified to be the most efficient working model as compared to the other conventional models.

V. CONCLUSION

For performing pattern recognition on tremendously large data sets such as credit card transactions it would require strenuous human effort to train the system, if we had applied traditional fraud detection methods. Hence we have used machine learning algorithm specifically random forest because of it's "Randomness" in selecting attributes; it overwrites the short-comes of other decision making algorithm. The machine learning algorithms stand ahead because here the system trains itself based on the previous patterns (unlike traditional methods). The apache Hadoop and spark integration setup, preprocess the selected raw data and outputs data in a format feasible to run the pattern recognition and machine learning algorithms. Using this entire setup we have achieved an uplift in the efficiency rates by 0.267% as compared to the previously used models. Therefore the BPRRF would dynamically classify the transactions as fraud and non-fraud.

REFERENCES

1. Y. Gmbh And K. G. Co, "Global Online Payment Methods: Full Year 2016," Tech. Rep., 3 2016.M. Sathyapriya & Dr. V. Thiagarasu ,"Big Data Analytics Techniques For Credit Card Fraud Detection : A Review," International Journal Of Science And Research (Ijsr) Issn (Online): 2319-7064 Index Copernicus Value (2015): 78.96 | Impact Factor (2015): 6.391.
2. M Kavitha & Suriakala, "Fraud Detection In Current Scenario, Sophistications And Directions: A Comprehensive Survey," International Journal Of Computer Applications (0975 – 8887) Volume 111 – No 5, February 2015
3. Kamaruddin Sk & Ravi Vadlamani, "Credit Card Fraud Detection Using Big Data Analytics: Use Of Psoaann Based One-Class Classification," Conference Paper · August 2016 Doi: 10.1145/2980258.2980319
4. S. Venkata Suryanarayana , G. N. Balaji , G. Venkateswara Rao , "Machine Learning Approaches For Credit Card Fraud Detection," International Journal Of Engineering & Technology, 7 (2) (2018) 917-920
5. Suraj Patil, Varsha Nemade, Piyush Kumar Soni, "Predictive Modelling For Credit Card Fraud Detection Using Data Analytics," International Conference On Computational Intelligence And Data Science (Iccids 2018.
6. Yusuf Sahin And Ekrem Duman2, "Detecting Credit Card Fraud By Ann And Logistic Regression," Article · June 2011
7. Sai Kiran Usaikiran, Jyoti Guru, Rishabh Kumar, Naveen Kumar, Deepak Katariya, Maheshwar Sharma, "Credit Card Fraud Detection Using Naïve Bayes Model Based And Knn Classifier," International Journal Of Advance Research, Ideas And Innovations In Technology
8. Vijayshree B. Nipane, Poonam S. Kalinge, Dipalividhate, Kunal War, Bhagyashree P. Deshpande, "Fraudulent Detection In Credit Card System Using Svm & Decision Tree," May 2016 Ijsdr |Volume 1, Issue 5
9. Prajal Save, Pranalitiwarekar, Ketan N. Jain, Neha Mahyavanshi, "A Novel Idea For Credit Card Fraud Detection Using Decision Tree," International Journal Of Computer Applications (0975 – 8887) Volume 161 – No 13, March 2017
10. Shiyang Xuan, Guanjun Liu, Zhenchuan Li, Lutao Zheng, Shuo Wang, Changjun Jiang, "Random Forest For Credit Cardfrauddetection,"Conference Paper · March 2018
11. Navanshu Khare, Saad Yunussait "Credit Card Fraud Detection Using Machine Learning Models And Collating Machine Learning Models," International Journal Of Pure And Applied Mathematics Vol 118, No.20 2018, 825-83.
12. Breiman, L. (2001)." Random Forests. Machine Learning," 45(1), 5-32dietterich, T. G. (2000). "Ensemble Methods In Machine Learning." , 1857(1), 1-15.Beel, T., De Peer, Y. V. And Saeys, Y. Java-Ml: "A Machine Learning Library, Journal Of Machine Learning Research," 2009, 10, 931-934
13. Quinlan, J. R. (1986)." Induction On Decision Tree. Machine Learning," 1(1), 81-106.
14. Breiman, L., Friedman, J. H., Olshen, R., And Stone, C. J. (1984). "Classification And Regression Trees. Biometrics, "40(3), 358.
15. Shanmuk Srinivas Amiripalli, Veeramallubobba,' Research Onnetwork Design And Analysis Of Tgo Topology', International Journal Of Networking And Virtual Organizations.Vol. 19, No. 1, Pp 72-86,2018doi: 10.1504/Ijnvo.2018.1001503.
16. Shanmuk Srinivas Amiripalli, Kalyan Kumar A, B.Tulasi.(2016) 'Trimet Along With Its Properties And Scope', American Institute Of Physicsconference Proceedings 1705,Pp. 020032 1-9,2016.Doi/10.1063/1.4940280.
17. H. N. R. Rao And S. V. S. R. Raju, Shanmuk Srinivas A (2012) 'On Path Connector Sets,' Ijmsc, Vol. 2, No. 2, Pp. 55–65.
18. Amiripalli S.S., Bobba V., Potharaju S.P. (2019) A Novel Trimet Graph Optimization (Tgo) Topology For Wireless Networks. In: Mallick P., Balas V., Bhoi A., Zobaa A. (Eds) Cognitive Informatics And Soft Computing. Advances In Intelligent Systems And Computing, Vol 768. Springer, Singapore.Doihttps://Doi.Org/10.1007/978-981-13-0617-4_8
19. Potharaju S.P., Sreedevi M., Amiripalli S.S. (2019) An Ensemble Feature Selection Framework Of Sonar Targets Using Symmetrical Uncertainty And Multi-Layer Perceptron (Su-Mlp). In: Mallick P., Balas V., Bhoi A., Zobaa A. (Eds) Cognitive Informatics And Soft Computing. Advances In Intelligent Systems And Computing, Vol 768. Springer, Singaporedoi\Https://Doi.Org/10.1007/978-981-13-0617-4_24
20. Shanmuk Srinivas Amiripalli, Veeramallubobba,' Impact Of Trimet Graph Optimization Topology On Scalable Networks', Journal Of Intelligent And Fuzzy Systems,2018,Doi: 10.3233/Jifs-169954