



IJRASET

International Journal For Research in
Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** IV **Month of publication:** April 2025

DOI: <https://doi.org/10.22214/ijraset.2025.68242>

www.ijraset.com

Call:  08813907089

E-mail ID: ijraset@gmail.com

Fraud Detection in Credit Card Data Using Supervised Machine Learning Based Scheme

T Poornima¹, T Praveen Kumar², K Munibala subramanyam³, Sannapureddy Prathap⁴, Dasari Puneeth kumar⁵, Mr.V. Gopi,M.E.,(Ph.D)⁶

^{1, 2, 3, 4, 5}UG Scholar, ⁶Associate Professor, Dept. of CSE Siddartha Institute of Science & Technology, Puttur, india

Abstract: In fraud detection, Decision Trees, Random Forest, and XG Boost are utilized for their effectiveness in classifying transactions. Decision Trees create a model that splits data based on feature values, forming an intuitive tree structure that leads to final classifications. Random Forest improves upon this by using multiple Decision Trees with random data subsets, aggregating their predictions to enhance accuracy and reduce overfitting. XG Boost employs a gradient boosting approach, building trees sequentially and optimizing performance through techniques like regularization and parallel processing. Together, these algorithms form a robust system capable of adapting to complex transaction patterns while minimizing false positives and negatives. The algorithm creates many decision trees during training, each trained with a specific random noise. The algorithm then uses the results from all the trees to make a prediction

Keywords: Financial transactions, Fraud, Patterns, Decision tree, random Forest, XG Boost, etc.

I. INTRODUCTION

A. Motivation:

The motivation for Credit Card Fraud Detection using Machine Learning lies in the need to safeguard financial transactions. By leveraging advanced algorithms, this approach aims to detect and prevent fraudulent activities, ensuring the security of both businesses and consumers, thereby reducing potential financial losses and enhancing trust in online transactions.

B. Problem Statement:

Develop a machine learning model to detect fraudulent credit card transactions with high accuracy, minimizing false positives, and reducing financial losses for both customers and financial institutions. The model must handle large-scale data, recognize subtle patterns indicative of fraud, and maintain real-time processing efficiency.

C. Objective of the Project:

The objective of this project is to develop a robust machine learning model for accurate detection of credit card fraud. Through comprehensive analysis of transaction data, the aim is to identify fraudulent activities with high precision and efficiency, thereby minimizing financial losses and enhancing overall security for cardholders and financial institutions.

D. Scope:

Develop a machine learning model to detect credit card fraud by analysing transaction patterns, user behaviour, and historical data. Implement algorithms for anomaly detection, classification, and predictive modelling. Enhance the model's accuracy by integrating real-time monitoring, feature engineering, and advanced data preprocessing techniques.

E. Project introduction

In recent years, the proliferation of digital transactions has led to an alarming increase in credit card fraud, posing significant financial risks to both financial institutions and consumers. Consequently there is an urgent need for robust and efficient fraud detection systems to mitigate these risks. leveraging the power of machine learning has emerged as a promising approach to combat the pervasive issue.

This project aims to develop an advanced credit card fraud detection system using machine learning techniques. By harnessing the capabilities of ml algorithms, this system will analyse historical transactional data and identify patterns indicative of fraudulent activities. Though the utilization of various models, including but not limited to

Logistic regression, decision tree and neural networks, the project seeks to create a comprehensive and accurate fraud detection mechanism capable of detecting even the most intricate fraudulent patterns.

The implementation of this system will not only enhance the security of financial transactions but also minimize the potential financial losses incurred by fraudulent activities. By leveraging the power of machine learning, this project end to contribute to the ongoing efforts to establish a secure and trust worthy environment for digital financial transactions.

II. LITERATURE SURVEY

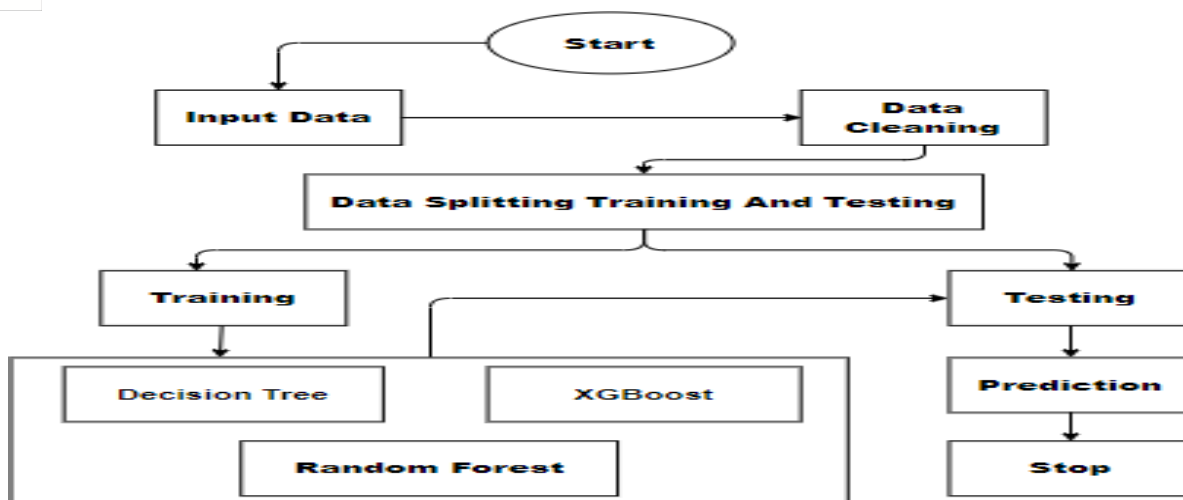
- 1) Bhattacharyya, S., Jha, D., Thara kunnel, K., & Westland, J. C. Year 2011 Credit cardfraud detection with a neural-network. This paper explores the use of neuralnetworks for credit card fraud detection and discusses how this machine learning approach can effectively identify fraudulent transactions.
- 2) Bhattacharyya, S., Jha, D., & Thara kunnel, K. Year 2011 Data mining for credit cardfraud: A comparative study. Thisstudyprovidesacomparativeanalysisofvariousdataminingtechniquesandtheir effectiveness in detecting credit card fraud, offering insights into the best methods.
- 3) Dal Pozzolo, A., Boracchi, G., Caelen, O., Alippi, C., & Bontempi, G. Year 2015 Credit card fraud detection: A realistic modeling and a novel learning strategy. Thispaperintroducesarealisticcreditcardfrauddatasetandanovellearningstrategythat improves the performance of machine learning models for fraud detection.
- 4) Shapoval, A., & Sokolov, V. Year 2017 Comparative analysis of credit card fraud detection using neural networks and logistic regression. This paper conducts a comparative analysis of neural networks and logistic regression for credit card fraud detection, offering insights into the strengths and weaknesses of each approach.
- 5) Islam, M. Z., Biswas, M., & Hyder, S. A. Year 2019 Fraud detection in credit card transactions using machine learning. This paper presents an overview of machine learning techniques used in credit card fraud detection, emphasizing the importance of feature selection and model evaluation.
- 6) Ahmed, M., Mahmood, A. N., & Hu, J. Year 2016 A survey of network anomaly detection techniques. While not exclusively focused on credit card fraud, this survey provides valuable insights into anomaly detection techniques, which are commonly employed in fraud detection systems.

III. PROPOSED SYSTEM

We propose this application that can be considered a useful system since it helps to reduce the limitations obtained from traditional and other existing methods. The objective of this study to develop fast and reliable method which detects fraudulent transactions accurately. To design this system is we used some powerful algorithms in a based Python environment Like Decision Tree, Random Forest, XGBoost.

A. Advantages

- 1) Enhanced Accuracy: The system's machine learning algorithms can accurately detect and predict fraudulent activities, minimizing false positives and negatives.
- 2) Real-time Monitoring: With its ability to process data swiftly, the system can monitor transactions in real time, enabling rapid fraud identification and prevention.
- 3) Adaptability: The system can adapt to evolving fraud patterns, ensuring that it remains effective in detecting new types of fraudulent activities.
- 4) Cost Efficiency: By automating the detection process, the system reduces the need for manual intervention, resulting in cost savings for the credit card companies.
- 5) Improved Customer Experience: Through swift detection and prevention of fraudulent transactions, the system helps protect customers from potential financial losses, enhancing overall satisfaction and trust.



Work Flow Of Proposed System

IV. IMPLEMENTATION

Modules

A. USER

- 1) Users can upload a dataset, which is a crucial initial step for the system to work with relevant data. This dataset likely contains historical information or examples that the system will use for its predictions.
- 2) Users have the capability to view the dataset they've uploaded. This feature helps users confirm the data they've provided and ensures transparency in the process.
- 3) Users need to input specific values or parameters into the system to request predictions or results. These input values likely correspond to the variables or features in the dataset.

B. SYSTEM

- 1) Take the Dataset: The system accepts and processes the dataset provided by the user. This dataset forms the foundation for building the predictive model.
- 2) Preprocessing: Before training a predictive model, the system preprocesses the dataset. This includes handling missing data, data cleaning, and feature extraction. Preprocessing ensures that the data is in a suitable format for modeling.
- 3) Training: The system uses machine learning techniques and Python modules to train a model based on the preprocessed dataset. The model learns patterns and relationships within the data, allowing it to make predictions.
- 4) Generate Results: Once the model is trained, the system can generate results based on user input values. These results typically indicate whether the input data corresponds to a specific condition, event, or prediction, such as Medical Insurance Cost.

V. ALGORITHMS

A. Decision Tree

A tree has many analogies in real life and turns out that it has influenced a wide area of machine learning covering both classifications and regression, in decision analysis a decision tree can be a tree like model of decision. Though a commonly used tool in data mining for deriving a strategy to reach a particular goal. A decision tree is drawn upside down with its root at the top. In the image on the left, the bold text in black represents a condition/internal node, based on which the tree splits into branches/ edges. The end of the branch that doesn't split anymore is the decision/leaf, in this case, whether the passenger died or survived, represented as red and green text respectively.

Although, a real dataset will have a lot more features and this will just be a branch in a much bigger tree, but you can't ignore the simplicity of this algorithm. The feature importance is clear and relations can be viewed easily. This methodology is more commonly known as learning decision tree from data and above tree is called Classification tree as the target is to classify passenger as survived or died. Regression trees are represented in the same manner, just they predict continuous values like price of a house. In general, Decision Tree algorithms are referred to as CART or Classification and Regression Trees.

So, what is actually going on in the background? Growing a tree involves deciding on which features to choose and what conditions to use for splitting, along with knowing when to stop. As a tree generally grows arbitrarily, you will need to trim it down for it to look beautiful. Let's start with a common technique used for splitting.

B. XG Boost

XG Boost stands for "Extreme Gradient Boosting". XG Boost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements Machine Learning algorithms under the Gradient Boosting framework. It provides a parallel tree boosting to solve many data science **problems in a fast and accurate way.**

Boosting

Boosting is an ensemble learning technique to build a strong classifier from several weak classifiers in series. Boosting algorithms play a crucial role in dealing with bias-variance trade-off. Unlike bagging algorithms, which only controls for high variance in a model, boosting controls both the aspects (bias & variance) and is considered to be more effective.

Below are the few types of boosting algorithms:

- AdaBoost (Adaptive Boosting)
- Gradient Boosting
- XG Boost
- Cat Boost

XG Boost-XG Boost stands for extreme Gradient Boosting. It became popular in the recent days and is dominating applied machine learning and Kaggle competitions for structured data because of its scalability. XG Boost is an extension to gradient boosted decision trees (GBM) and specially designed to improve speed and performance.

XG Boost Features

- Regularized Learning: Regularization term helps to smooth the final learnt weights to avoid over-fitting. The regularized objective will tend to select a model employing simple and predictive functions.
- Gradient Tree Boosting: The tree ensemble model cannot be optimized using traditional optimization methods in Euclidean space. Instead, the model is trained in an additive manner.
- Shrinkage and Column Subsampling: Besides the regularized objective, two additional techniques are used to further prevent over fitting. The first technique is shrinkage introduced by Friedman. Shrinkage scales newly added weights by a factor η after each step of tree boosting. Similar to a learning rate in stochastic optimization, shrinkage reduces the influence of each tree and leaves space for future trees to improve the model.

The second technique is the column (feature) subsampling. This technique is used in Random Forest. Column sub-sampling prevents over-fitting even more so than the traditional row sub-sampling. The usage of column sub-samples also speeds up computations of the parallel algorithm.

System Features

Parallelization of tree construction using all of your CPU cores during training. Collecting statistics for each column can be parallelized, giving us a parallel algorithm for split finding. Cache-aware Access: XG Boost has been designed to make optimal use of hardware. This is done by allocating internal buffers in each thread, where the gradient statistics can be stored. Blocks for Out-of-core Computation for very large datasets that don't fit into memory. Distributed Computing for training very large models using a cluster of machines. Column Block for Parallel Learning. The most time-consuming part of tree learning is to get the data into sorted order. In order to reduce the cost of sorting, the data is stored in the column blocks in sorted order in compressed format.

Goals of XG Boost

Execution Speed: XG Boost was almost always faster than the other benchmarked implementations from R, Python Spark and H2O and it is really faster when compared to the other algorithms. Model Performance: XG Boost dominates structured or tabular datasets on classification and regression predictive modelling problems.

C. Random Forest

A random forest is a machine learning technique that's used to solve regression and classification problems. It utilizes ensemble learning, which is a technique that combines many classifiers to provide solutions to complex problems.

A random forest algorithm consists of many decision trees. The ‘forest’ generated by the random forest algorithm is trained through bagging or bootstrap aggregating. Bagging is an ensemble meta-algorithm that improves the accuracy of machine learning algorithms.

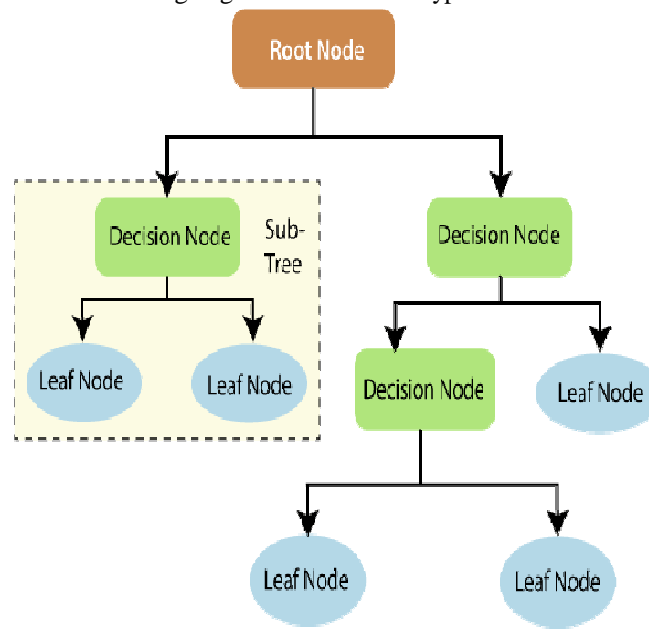
The (random forest) algorithm establishes the outcome based on the predictions of the decision trees. It predicts by taking the average or mean of the output from various trees. Increasing the number of trees increases the precision of the outcome.

Random forest eradicates the limitations of a decision tree algorithm. It reduces the overfitting of datasets and increases precision. It generates predictions without requiring many configurations in packages (like [Scikit-learn](#)).

Decision trees are the building blocks of a random forest algorithm. A decision tree is a decision support technique that forms a tree-like structure. An overview of decision trees will help us understand how random forest algorithms work.

A decision tree consists of three components: decision nodes, leaf nodes, and a root node. A decision tree algorithm divides a training dataset into branches, which further segregate into other branches. This sequence continues until a leaf node is attained. The leaf node cannot be segregated further.

The nodes in the decision tree represent attributes that are used for predicting the outcome. Decision nodes provide a link to the leaves. The following diagram shows the three types of nodes in a decision tree.



The information theory can provide more information on how decision trees work. Entropy and information gain are the building blocks of decision trees. An overview of these fundamental concepts will improve our understanding of how decision trees are built. Entropy is a metric for calculating uncertainty. Information gain is a measure of how uncertainty in the target variable is reduced, given a set of independent variables.

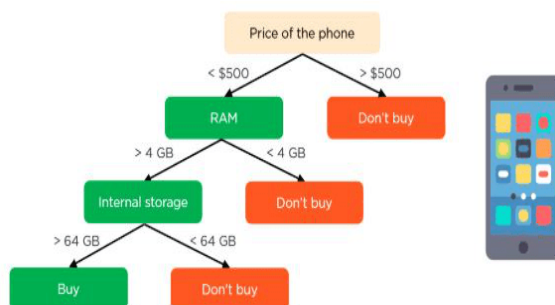
The information gain concept involves using independent variables (features) to gain information about a target variable (class). The entropy of the target variable (Y) and the conditional entropy of Y (given X) are used to estimate the information gain. In this case, the conditional entropy is subtracted from the entropy of Y.

Information gain is used in the training of decision trees. It helps in reducing uncertainty in these trees. A high information gain means that a high degree of uncertainty (information entropy) has been removed. Entropy and information gain are important in splitting branches, which is an important activity in the construction of decision trees.

Let's take a simple example of how a decision tree works. Suppose we want to predict if a customer will purchase a mobile phone or not. The features of the phone form the basis of his decision. This analysis can be presented in a decision tree diagram.

The root node and decision nodes of the decision represent the features of the phone mentioned above. The leaf node represents the final output, either *buying* or *not buying*. The main features that determine the choice include the price, internal storage, and Random Access Memory (RAM). The decision tree will appear as follows.

Applying decision trees in random forest



The main difference between the decision tree algorithm and the random forest algorithm is that establishing root nodes and segregating nodes is done randomly in the latter. The random forest employs the bagging method to generate the required prediction. Bagging involves using different samples of data (training data) rather than just one sample. A training dataset comprises observations and features that are used for making predictions. The decision trees produce different outputs, depending on the training data fed to the random forest algorithm. These outputs will be ranked, and the highest will be selected as the final output.

Our first example can still be used to explain how random forests work. Instead of having a single decision tree, the random forest will have many decision trees. Let's assume we have only four decision trees. In this case, the training data comprising the phone's observations and features will be divided into four root nodes.

The root nodes could represent four features that could influence the customer's choice (price, internal storage, camera, and RAM). The random forest will split the nodes by selecting features randomly. The final prediction will be selected based on the outcome of the four trees.

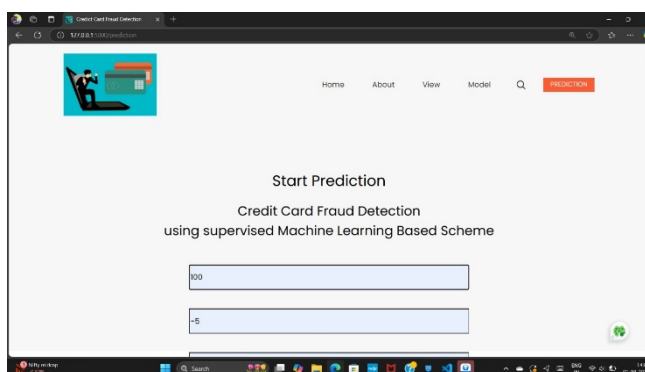
VI. FUTURE ENHANCEMENT

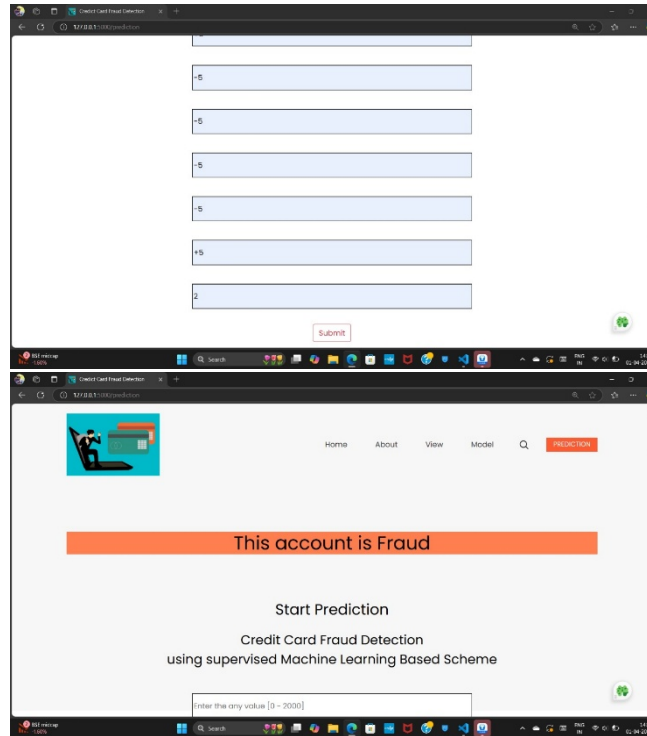
Future enhancements for Credit Card Fraud Detection using machine learning could include the integration of advanced deep learning techniques, such as recurrent neural networks or transformers, to capture complex temporal dependencies in transaction data. Additionally, implementing real-time anomaly detection algorithms and leveraging blockchain technology for secure transaction verification could bolster the system's fraud detection capabilities. Integrating explainable AI models to provide transparent insights into the decision-making process would enhance trust and understanding, while also facilitating regulatory compliance. Moreover, exploring the potential of federated learning to enable collaborative model training across multiple institutions without sharing sensitive data could significantly improve the overall fraud detection framework's robustness and accuracy.

VII. RESULT

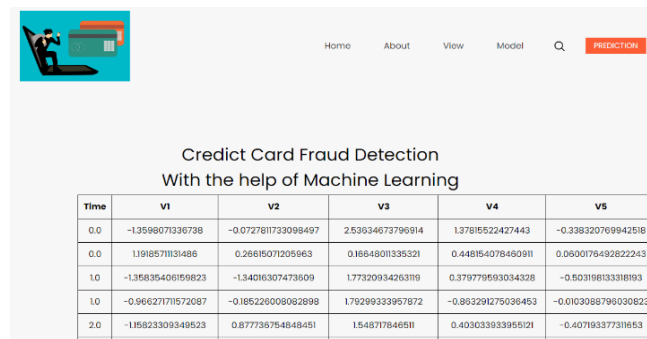
By using following algorithms we will predict the outputs

- Decision tree
- Random forest
- XG Boost





Data Set:



Time	V1	V2	V3	V4	V5
0.0	-1.3598071336738	-0.072781733098497	2.53934673796914	1.37815522427443	-0.338320769942518
0.0	1.1916571131486	0.26615071205963	0.16648011335321	0.448154078460911	0.06007549222243
1.0	-1.35835406159823	-1.34016307473509	1.7732093426319	0.379779593034328	-0.503198133381893
1.0	-0.96627171572087	-0.185226009062898	1.79299333957672	-0.893291275036453	-0.0103086790030623
2.0	-1.15823309349523	0.877736754848451	1.548177846811	0.403033933955121	-0.407193377311653

VIII. CONCLUSION

The application of machine learning in credit card fraud detection has proven to be an indispensable tool in safeguarding financial transactions. Through the utilization of sophisticated algorithms, the system can efficiently identify fraudulent activities, thereby minimizing the risks associated with unauthorized transactions. The implementation of such technology not only enhances the security of financial institutions and their customers but also contributes to the overall stability and trust within the financial ecosystem. As advancements continue to refine these detection systems, the future holds promising prospects for even more robust and reliable fraud prevention measures, ensuring a safer and more secure financial landscape for all stakeholders.

REFERENCES

- [1] Bhattacharyya S, Jha S, Tharakunnel K. Credit Card Fraud Detection using Machine Learning: A Survey. IEEE Access. 2019;7:37393-37420.
- [2] Dal Pozzolo A, Caelen O, Le Borgne Y-A, et al. Calibrating Probability with Undersampling for Unbalanced Classification. In: Intelligent Data Analysis. Springer; 2015:160-173.
- [3] Phua C, Lee V, Smith K, Gayler R. A Comprehensive Survey of Data Mining-based Fraud Detection Research. 2010.
- [4] Bhattacharyya S, Jha S, Tharakunnel K. Unsupervised Machine Learning in Credit Card Fraud Detection: A Comparison of Novel Approaches. Expert Systems with Applications. 2019;118:437-453.



- [5] Jiang Z, Cao J, Cao J, et al. Credit Card Fraud Detection: A Realistic Modeling and a Novel Learning Strategy. IEEE Transactions on Neural Networks and Learning Systems. 2016;27(10):2064-2077.
- [6] Lázaro-Gredilla M, Ranga A, Arapakis I, Vallet D. Sequential Anomaly Detection in Credit Card Transactions. Information Sciences. 2015;303:140-156.
- [7] ZhengY, LuX, ChenH, JajodiaS. VDDoS: Virtual Currency in the DDoS Service. IEEE Transactions on Dependable and Secure Computing. 2017;14(2):154-168.
- [8] Nasrabadi NM. Pattern Recognition and Machine Learning. Journal of Electronic Imaging. 2007;16(4):049901.
- [9] Bramer M. Principles of Data Mining. Springer; 2007.
- [10] Shmueli G, Patel NR, Bruce PC. Data Mining for Business Intelligence: Concepts, Techniques, and Applications in Microsoft Office Excel with XLMiner. John Wiley & Sons; 2007.



10.22214/IJRASET



45.98



IMPACT FACTOR:
7.129



IMPACT FACTOR:
7.429



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24*7 Support on Whatsapp)