International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

xx

COPY RIGHT

IJIEMR Transactions, online available on 04th May 2024. Link

https://www.ijiemr.org/downloads/Volume-13/ISSUE-5

10.48047/IJIEMR/V13/ISSUE 05/09

TITLE: AN IMPROVED COST-SENSITIVE PAYMENT CARD FRAUD DETECTION BASED ON DYNAMIC RANDOM FOREST AND K-NEAREST NEIGHBOURS

Volume 13, ISSUE 05, Pages: 68-94

Paper Authors **Dr Ali Mirza Mahmood, Borra Loknath**

USE THIS BARCODE TO ACCESS YOUR ONLINE PAPER

To Secure Your Paper As Per UGC Guidelines We Are Providing A Electronic Bar Code

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

AN IMPROVED COST-SENSITIVE PAYMENT CARD FRAUD DETECTION BASED ON DYNAMIC RANDOM FOREST AND K-NEAREST NEIGHBOURS

Dr Ali Mirza Mahmood[1], Borra Loknath[2]

[1]Assistant Professor, Department of Computer Science and Engineering, Krishna University, Machilipatnam, Email:alimirza.md@gmail.com
[2]Student, Department of Computer Science and Engineering, Krishna University, Machilipatnam, Email:loknathborra77@gmail.com

ABSTRACT:

With the widespread use of electronic payment systems, the risk of fraudulent activities in card transactions has become a significant concern for financial institutions and cardholders alike. Traditional fraud detection methods often focus on achieving high accuracy without considering the varying costs associated with false positives and false negatives. This paper proposes a cost-sensitive approach to card fraud detection, leveraging the dynamic random forest (DRF) and k-nearest neighbours (KNN) algorithms to enhance the accuracy of identifying fraudulent transactions while minimizing the financial impact of misclassifications. In the academe, as well, payment card fraud detection has become an important research topic in recent years. The extraction of suitable transactional features is one of the key issues in constructing an effective fraud detection model. A cardholder's spending behaviours vary over time so that new behaviour of a cardholder is closer to his/her recent behaviours. This measure assigns greater weight to recent transactions. The dynamic random forest algorithm is applied for the evolution of transactions as fraudulent or legitimate and KNN algorithm is used for capturing similarities between transactions and detecting local patterns. Finally, prevention of damage is achieved with the proposed cost-sensitive approach. capturing similarities between transactions and detecting local patterns.

KEYWORDS:

Payment card, Fraud Detection, Dynamic Random Forest , K-Nearest Neighbour Algorithm

1. INTRODUCTION:

The continuous evolution of technology in the financial sector has led to an increased reliance on electronic payment systems, making card transactions a vital aspect of modern commerce. However, this convenience comes with the downside of an escalating threat from fraudulent activities. Payment card fraud is a pervasive and evolving challenge in the financial sector, threatening the integrity of electronic transactions and the security of cardholders. As the sophistication of fraudulent activities continues to grow, there is an increasing need for robust

and adaptive fraud detection systems. Traditional fraud detection methods, though effective to some extent, often neglect the asymmetrical costs associated with misclassifying transactions. False positives (incorrectly identifying a legitimate transaction as fraudulent) and false negatives (failing to identify a fraudulent transaction) have distinct financial consequences for both financial institutions and cardholders. In this context, the integration of machine learning algorithms, specifically random forest and k-nearest neighbours (KNN), has proven to be a powerful approach to enhance the accuracy and efficiency of fraud detection.

To address these challenges, this project focuses on developing a cost-sensitive card fraud detection system that goes beyond conventional accuracy metrics. The approach integrates the dynamic random forest (DRF) algorithm and the k-nearest neighbours (KNN) algorithm, harnessing their strengths to create a robust and adaptive model capable of handling the dynamic nature of fraudulent activities. The DRF algorithm offers an ensemble learning technique that adapts to changing patterns in transaction data, while the KNN algorithm provides a locally sensitive approach to identifying anomalies in transaction behaviour.

Payment card fraud involves unauthorized transactions, identity theft, and other deceptive activities that exploit vulnerabilities in electronic payment systems. Detecting fraud in real-time is crucial for financial institutions to mitigate financial losses and protect the trust of their customers. The consequences of undetected fraud are severe, impacting both financial institutions and cardholders. Fraudulent transactions can lead to significant financial losses, damage to the reputation of the institution, and potential legal implications. Therefore, implementing effective fraud detection mechanisms is paramount.

One of the key challenges in credit card fraud detection is the imbalanced nature of the dataset, where the number of legitimate transactions far exceeds the instances of fraud. This imbalance can lead to biased models that prioritize accuracy but may fail to detect rare fraudulent cases. Traditional rule-based systems for fraud detection have limitations in adapting to dynamic fraud patterns. To address this, a cost-sensitive approach is incorporated, where the model is trained to minimize the cost associated with misclassifying fraud and non-fraud instances. Machine learning algorithms, such as random forest and KNN, offer a more sophisticated and adaptable solution. Random forest is an ensemble learning algorithm that constructs a multitude of decision trees and combines their outputs to improve overall accuracy. In the context of payment card fraud detection, random forest can analyze a diverse set of features associated with transactions, identifying intricate relationships that may indicate fraudulent behaviour. Random forest excels at capturing complex relationships in data.

KNN is a non-parametric algorithm that classifies data points based on the majority class of their k-nearest neighbours. KNN is effective in identifying local patterns. In fraud detection, KNN can be employed to identify local patterns within the feature space, making it particularly

effective for detecting anomalies in transaction data. This ensures that the system is not only accurate but also economically viable for financial institutions.

The primary goal of this research is to strike a balance between precision and the associated costs by assigning different misclassification costs to false positives and false negatives. The objectives of integrating dynamic random forest and KNN in payment card fraud detection include improving accuracy, reducing false positives and negatives, enhancing scalability, providing insights into feature importance, adapting to changing fraud patterns, ensuring model interpretability, and lowering operational costs. By doing so, the proposed model aims to minimize the financial impact of fraud detection errors, ultimately enhancing the overall efficiency of the card fraud detection system.

Combining the strengths of random forest and KNN through ensemble learning techniques further enhances the robustness and effectiveness of the fraud detection system. Ensemble models can leverage the complementary strengths of individual algorithms, resulting in improved overall performance. As payment card fraud continues to evolve, the integration of advanced machine learning algorithms will remain critical. The future of fraud detection lies in the continuous refinement of models, the incorporation of new features, and the exploration of emerging technologies to stay ahead of increasingly sophisticated fraudulent activities.

This project contributes to the advancement of card fraud detection methodologies by introducing a cost-sensitive perspective and leveraging the complementary strengths of DRF and KNN algorithms. The integration of dynamic random forest and KNN algorithms in payment card fraud detection represents a strategic and effective approach to safeguarding electronic transactions. The subsequent sections will delve into the methodology, experimental setup, results, and discussions, providing insights into the effectiveness and practical implications of the proposed approach in mitigating the impact of card fraud in electronic payment systems.

The objectives of payment card fraud detection using random forest and k-nearest neighbours (KNN) algorithms can include:

1. Enhance the accuracy of fraud detection systems by leveraging the strengths of both random forest and KNN algorithms. Random forests are known for their ability to handle complex relationships in data, while KNN can capture local patterns.

2. Minimize the occurrence of false positives (legitimate transactions classified as fraud) and false negatives (fraudulent transactions classified as legitimate) to improve the overall reliability of the fraud detection system.

3. Develop a scalable fraud detection system capable of handling large volumes of transaction data. Random forests and KNN can be efficient for processing data, and their parallelization capabilities can contribute to scalability.

4. Implement real-time fraud detection to quickly identify and prevent fraudulent transactions as they occur. Both random forest and KNN algorithms can be adapted for real-time processing, although considerations for computational efficiency are important.

5. Gain insights into the most influential features contributing to fraud detection. Random forests provide a feature importance score, allowing analysts to identify key variables in the dataset. KNN can also help identify patterns in feature space.

6. Build a fraud detection system that can adapt to changing fraud patterns over time. Random forest, with its ability to handle non-linear relationships, and KNN, which adapts well to local patterns, can collectively contribute to a system that stays effective against evolving fraud techniques.

7. Improve the interpretability of the fraud detection model, allowing stakeholders to understand the reasoning behind the classification decisions. Random forests can provide insights into feature importance, and KNN can offer transparency in terms of local decision boundaries.

2. BACKGROUND:

Before we take a look at the details of various machine learning methods, let's start by looking at what machine learning is, and what it isn't. Machine learning is often categorized as a subfield of artificial intelligence, but I find that categorization can often be misleading at first brush. The study of machine learning certainly arose from research in this context, but in the data science application of machine learning methods, it's more helpful to think of machine learning as a means of building models of data.

Fundamentally, machine learning involves building mathematical models to help understand data. "Learning" enters the fray when we give these models tunable parameters that can be adapted to observed data; in this way the program can be considered to be "learning" from the data. Once these models have been fit to previously seen data, they can be used to predict and understand aspects of newly observed data. I'll leave to the reader the more philosophical

digression regarding the extent to which this type of mathematical, model-based "learning" is similar to the "learning" exhibited by the human brain. Understanding the problem setting in machine learning is essential to using these tools effectively, and so we will start with some broad categorizations of the types of approaches we'll discuss here.

Categories Of Machine Learning:

At the most fundamental level, machine learning can be categorized into two main types: supervised learning and unsupervised learning.

Supervised learning involves somehow modelling the relationship between measured features of data and some label associated with the data; once this model is determined, it can be used to apply labels to new, unknown data. This is further subdivided into classification tasks and regression tasks: in classification, the labels are discrete categories, while in regression, the labels are continuous quantities. We will see examples of both types of supervised learning in the following section.

Unsupervised learning involves modelling the features of a dataset without reference to any label, and is often described as "letting the dataset speak for itself." These models include tasks such as clustering and dimensionality reduction. Clustering algorithms identify distinct groups of data, while dimensionality reduction algorithms search for more succinct representations of the data.

Need for Machine Learning

Human beings, at this moment, are the most intelligent and advanced species on earth because they can think, evaluate and solve complex problems. On the other side, AI is still in its initial stage and haven’t surpassed human intelligence in many aspects. Then the question is that what is the need to make machine learn? The most suitable reason for doing this is, “to make decisions, based on data, with efficiency and scale”.

Lately, organizations are investing heavily in newer technologies like Artificial Intelligence, Machine Learning and Deep Learning to get the key information from data to perform

several real-world tasks and solve problems. We can call it data-driven decisions taken by machines, particularly to automate the process. These data-driven decisions can be used, instead of using programing logic, in the problems that cannot be programmed inherently. The fact is that we can't do without human intelligence, but other aspect is that we all need to solve real-world problems with efficiency at a huge scale. That is why the need for machine learning arises.

Challenges in Machines Learning :-

While Machine Learning is rapidly evolving, making significant strides with cybersecurity and autonomous cars, this segment of AI as whole still has a long way to go. The reason behind is that ML has not been able to overcome number of challenges. The challenges that ML is facing currently are −

Quality of data − Having good-quality data for ML algorithms is one of the biggest challenges. Use of low-quality data leads to the problems related to data preprocessing and feature extraction.

Time-Consuming task − Another challenge faced by ML models is the consumption of time especially for data acquisition, feature extraction and retrieval.

Lack of specialist persons − As ML technology is still in its infancy stage, availability of expert resources is a tough job.

No clear objective for formulating business problems − Having no clear objective and well-defined goal for business problems is another key challenge for ML because this technology is not that mature yet.

Issue of overfitting & underfitting − If the model is overfitting or underfitting, it cannot be represented well for the problem.

Curse of dimensionality − Another challenge ML model faces is too many features of data points. This can be a real hindrance.

Difficulty in deployment − Complexity of the ML model makes it quite difficult to be deployed in real life.

How to start learning ML?

This is a rough roadmap you can follow on your way to becoming an insanely talented Machine Learning Engineer. Of course, you can always modify the steps according to your needs to reach your desired end-goal!

Step 1 – Understand the Prerequisites

In case you are a genius, you could start ML directly but normally, there are some prerequisites that you need to know which include Linear Algebra, Multivariate Calculus, Statistics, and Python. And if you don't know these, never fear! You don't need a Ph.D. degree in these topics to get started but you do need a basic understanding.

(a) Learn Linear Algebra and Multivariate Calculus

Both Linear Algebra and Multivariate Calculus are important in Machine Learning. However, the extent to which you need them depends on your role as a data scientist. If you are more focused on application heavy machine learning, then you will not be that heavily focused on math as there are many common libraries available. But if you want to focus on R&D in Machine Learning, then mastery of Linear Algebra and Multivariate Calculus is very important as you will have to implement many ML algorithms from scratch.

(b) Learn Statistics

Data plays a huge role in Machine Learning. In fact, around 80% of your time as an ML expert will be spent collecting and cleaning data. And statistics is a field that handles the collection, analysis, and presentation of data. So it is no surprise that you need to learn it!!! Some of the key concepts in statistics that are important are Statistical Significance,

Probability Distributions, Hypothesis Testing, Regression, etc. Also, Bayesian Thinking is also a very important part of ML which deals with various concepts like Conditional Probability, Priors, and Posteriors, Maximum Likelihood, etc.

(c) Learn Python

Some people prefer to skip Linear Algebra, Multivariate Calculus and Statistics and learn them as they go along with trial and error. But the one thing that you absolutely cannot skip is Python! While there are other languages you can use for Machine Learning like R, Scala, etc. Python is currently the most popular language for ML. In fact, there are many Python libraries that are specifically useful for Artificial Intelligence and Machine Learning such as Keras, TensorFlow, Scikit-learn, etc.

So if you want to learn ML, it's best if you learn Python! You can do that using various online resources and courses such as **Fork Python** available Free on Geeks for Geeks.

Step 2 – Learn Various ML Concepts

Now that you are done with the prerequisites, you can move on to actually learning ML (Which is the fun part!!!) It's best to start with the basics and then move on to the more complicated stuff. Some of the basic concepts in ML are:

(a) Terminologies of Machine Learning

- **Model –** A model is a specific representation learned from data by applying some machine learning algorithm. A model is also called a hypothesis.
- **Feature –** A feature is an individual measurable property of the data. A set of numeric features can be conveniently described by a feature vector. Feature vectors are fed as input to the model. For example, in order to predict a fruit, there may be features like color, smell, taste, etc.

- **Target (Label) –** A target variable or label is the value to be predicted by our model. For the fruit example discussed in the feature section, the label with each set of input would be the name of the fruit like apple, orange, banana, etc.
- **Training –** The idea is to give a set of inputs(features) and it's expected outputs(labels), so after training, we will have a model (hypothesis) that will then map new data to one of the categories trained on.
- **Prediction –** Once our model is ready, it can be fed a set of inputs to which it will provide a predicted output(label).

(b) Types of Machine Learning

- **Supervised Learning –** This involves learning from a training dataset with labelled data using classification and regression models. This learning process continues until the required level of performance is achieved.
- **Unsupervised Learning –** This involves using unlabelled data and then finding the underlying structure in the data in order to learn more and more about the data itself using factor and cluster analysis models.
- **Semi-supervised Learning –** This involves using unlabelled data like Unsupervised Learning with a small amount of labelled data. Using labelled data vastly increases the learning accuracy and is also more cost-effective than Supervised Learning.
- **Reinforcement Learning –** This involves learning optimal actions through trial and error. So the next action is decided by learning behaviours that are based on the current state and that will maximize the reward in the future.

Advantages of Machine learning:-

1. Easily identifies trends and patterns -

Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, for an e-commerce website like Amazon, it serves to understand the browsing behaviours and purchase histories of its users to help cater to

the right products, deals, and reminders relevant to them. It uses the results to reveal relevant advertisements to them.

2. No human intervention needed (automation)

With ML, you don't need to babysit your project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own. A common example of this is anti-virus software they learn to filter new threats as they are recognized. ML is also good at recognizing spam.

3. Continuous Improvement

As **ML algorithms** gain experience, they keep improving in accuracy and efficiency. This lets them make better decisions. Say you need to make a weather forecast model. As the amount of data you have keeps growing, your algorithms learn to make more accurate predictions faster.

4. Handling multi-dimensional and multi-variety data

Machine Learning algorithms are good at handling data that are multi-dimensional and multi-variety, and they can do this in dynamic or uncertain environments.

5. Wide Applications

You could be an e-tailer or a healthcare provider and make ML work for you. Where it does apply, it holds the capability to help deliver a much more personal experience to customers while also targeting the right customers.

Disadvantages of Machine Learning:

1. Data Acquisition

Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality. There can also be times where they must wait for new data to be generated.

2. Time and Resources

ML needs enough time to let the algorithms learn and develop enough to fulfill their purpose with a considerable amount of accuracy and relevancy. It also needs massive resources to function. This can mean additional requirements of computer power for you.

3. Interpretation of Results

Another major challenge is the ability to accurately interpret results generated by the algorithms. You must also carefully choose the algorithms for your purpose.

4. High error-susceptibility

Machine Learning is autonomous but highly susceptible to errors. Suppose you train an algorithm with data sets small enough to not be inclusive. You end up with biased predictions coming from a biased training set. This leads to irrelevant advertisements being displayed to customers. In the case of ML, such blunders can set off a chain of errors that can go undetected for long periods of time. And when they do get noticed, it takes quite some time to recognize the source of the issue, and even longer to correct it.

DYNAMIC RANDOM FOREST:

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. It takes less training time as compared to other algorithms. It predicts output with high accuracy, even for the large dataset it runs efficiently. It can also maintain accuracy when a large proportion of data is missing. Random Forest works in

two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase.

The Working process can be explained in the below steps and diagram:

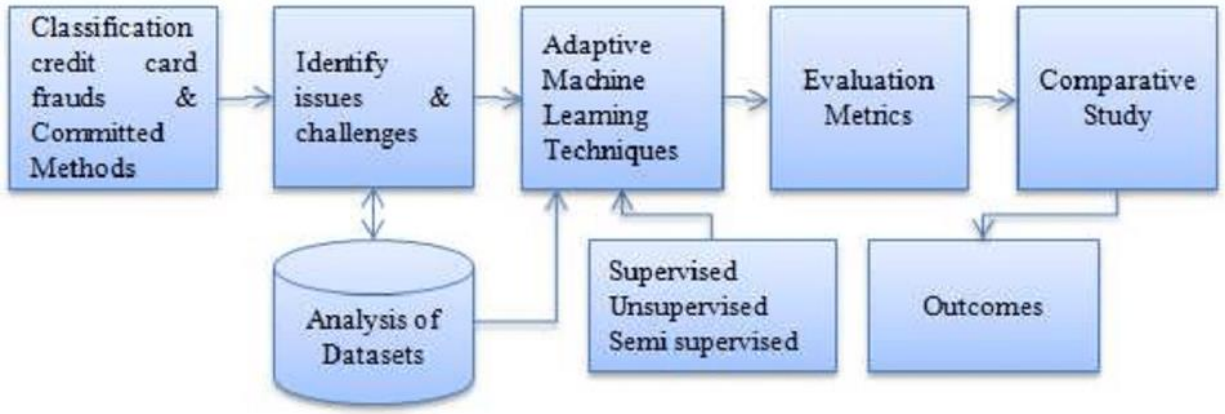
Step-1: Select random K data points from the training set.

Step-2: Build the decision trees associated with the selected data points (Subsets).

Step-3: Choose the number N for decision trees that you want to build.

Step-4: Repeat Step 1 & 2.

Step-5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.



Advantages:

- Random Forest is capable of performing both Classification and Regression tasks.
- It is capable of handling large datasets with high dimensionality.
- It enhances the accuracy of the model and prevents the overfitting issue.

K-NEAREST NEIGHBOUR:

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique. K-NN algorithm assumes the similarity between the new case/data and

available cases and put the new case into the category that is most similar to the available categories. K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm. K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems. K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data.

It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.

KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

The K-NN working can be explained on the basis of the below algorithm:

Step-1: Select the number K of the neighbours

Step-2: Calculate the Euclidean distance of K number of neighbours

Step-3: Take the K nearest neighbours as per the calculated Euclidean distance.

Step-4: Among these k neighbours, count the number of the data points in each category.

Step-5: Assign the new data points to that category for which the number of the neighbor is maximum.

Step-6: Our model is ready.

Advantages:

- It is simple to implement.
- It is robust to the noisy training data
- It can be more effective if the training data is large.

3. DATASET DESCRIPTION:

1. Class Distribution: Information about the distribution of classes in the dataset, including the number of samples belonging to each class. In an imbalanced dataset, one class (e.g., the minority class) is significantly smaller in size compared to the other class(es) (e.g., the majority class).

2. Class Labels: Description of the class labels used in the dataset, indicating the categories or outcomes being predicted. For example, in a binary classification problem, there may be two class labels such as "fraudulent" and "legitimate" for credit card fraud detection.

3. Data Size: Details about the total number of samples in the dataset and the proportion of samples belonging to each class. This helps to provide context on the imbalance between classes.

4. Feature Description: Description of the features included in the dataset, such as transaction amount, transaction type, time of transaction, and various other attributes. Understanding the features is crucial for feature selection and model training.

5. Data Source: Information about the source of the dataset, including any relevant citations or references to the original data provider. This helps researchers understand the context in which the data was collected.

6.Data Preprocessing: Details about any preprocessing steps applied to the data, such as feature scaling, normalization, handling missing values, or dealing with outliers. Preprocessing techniques may have implications for handling imbalance in the dataset.

7. Evaluation Metrics: Guidance on which evaluation metrics to use when evaluating models trained on the imbalanced dataset. Common metrics include accuracy, precision, recall, F1-score, area under the ROC curve (AUC-ROC), and area under the precision-recall curve (AUC-PR).

8. Previous Research: Any previous research or studies that have used the dataset, along with key findings or insights from those studies. Understanding previous work on the dataset can provide valuable context for new research projects.

4. RELATED WORK:

Anomaly detection is an important data analysis task. It is used to identify interesting and emerging patterns, trends and anomalies from data.

Anomaly detection is an important tool to detect abnormalities in many different domains including financial fraud detection, computer network intrusion, human behavioural analysis, gene expression analysis and many more.

Recently, in the financial sector, there has been renewed interest in research on detection of fraudulent activities.

There has been a lot of work in the area of clustering based unsupervised anomaly detection in the financial domain.

This paper presents an in-depth survey of various clustering based anomaly detection technique and compares them from different perspectives. In addition, we discuss the lack of real world data and how synthetic data has been used to validate current detection techniques.

YEAR	AUTHOR	TITLE	METHODS
2014	A. C. Bahnsen, A. Stojanovic, D. Aouada, and B. Ottersten	Credit card fraud detection with calibrated probabilities	Calibrating the probabilities and then using Bayes minimum Risk.
2015	M. Zareappor , P. Shamsolmoali	Application of Credit card Fraud Detection: Based on Bagging Ensemble Classifier	Naive Bayes (NB), Support Vector Machines (SVM), K-Nearest Neighbor algorithms (KNN)

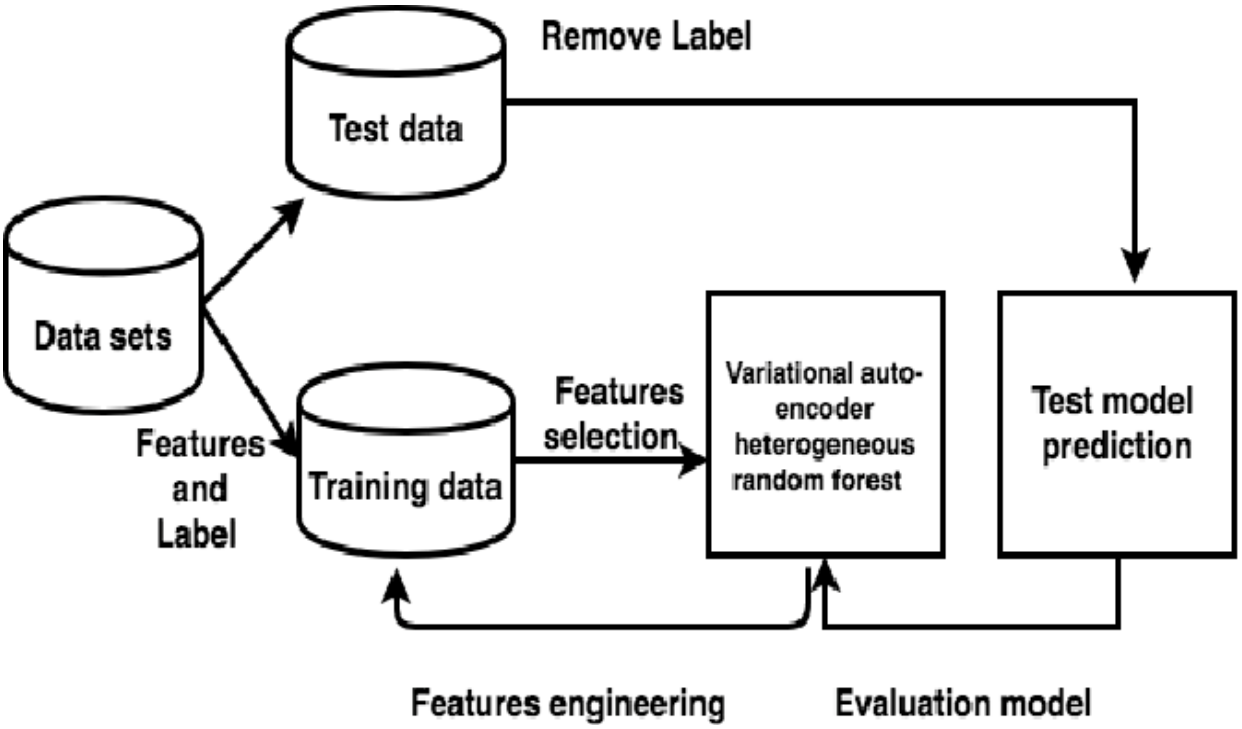
2017	J. O. Awoyemi, A. O. Adetunmbi, and S. A. Oluwadare	Credit card fraud detection using Machine Learning Techniques: A Comparative Analysis	Genetic algorithm (GA) for feature selection Decision Tree (DT), Random Forest (RF), Logistic Regression (LR)
2019	B. Meenakshi Devi, B. Janani, S. Gayathri, N.Indira	Credit Card Fraud Detection using Random Forest	Classification and regression algorithms, Random forest

5. PROPOSED SYSTEM

In this paper, the imbalanced nature of the credit card data has been taken into consideration, and a cost-sensitive weighted random forest technique has been proposed to overcome the issue. The Random Forest (RF) algorithm is basically an ensemble learning technique which works on the principle of Bagging, i.e. the dataset has been divided into n-bags or samples, with randomly selected instances, and each of which is termed as "Tree". The final output of the model is achieved based on majority voting (in case of classification model), or averaging (in case of regression model) of all the test outcomes, with respect to all the trained trees. The foremost advantage of using RF technique for credit card fraud detection is that it can readily deal with the enormous size the training data, as RF involves to divide the dataset into a number samples, and then learn

The Euclidean distance as the default distance is the most used distance metric in KNN. However, the Euclidean distance is an unfavorable metric for this purpose given that different types of features are used to construct a fraud detection model for card payments. We aimed to analyze user behaviour and not the distance between two points in space. In Table 5, A and B represent two transactions that have a low difference in the value of transaction amount and the high difference in transaction time. Euclidean distance of these two transactions is equal to 0.7, whereas the average distance between two features is equal to 0.38. These two transactions show

similar behaviour. Therefore, their distance should not be too high. With this simple example, it is clear that Euclidean distance is not a good metric for calculation of similarity between two transactions.



The implementation of the proposed algorithm is done on python with a system unit of Intel(R) Core(TM) i3 CPU working on 2.90 GHz and 4.0 G RAM on Windows 8 operating system.

Tensor flow

Tensor Flow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google.

Tensor Flow was developed by the Google Brain team for internal Google use. It was released under the Apache 2.0 open-source license on November 9, 2015.

Numpy:

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows Numpy to seamlessly and speedily integrate with a wide variety of databases.

Pandas:

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Matplotlib:

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar

charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

Scikit – learn:

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use.
Python:

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

- Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

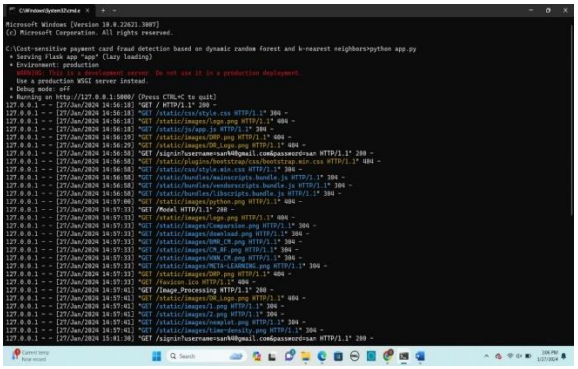
Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviours. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

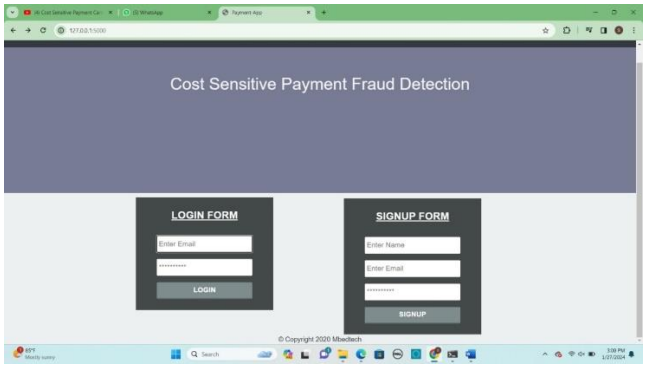
where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

To run the project, open project file and select app file and click cmd in workspace. Then the command prompt window will open and type the command : **python app.py**

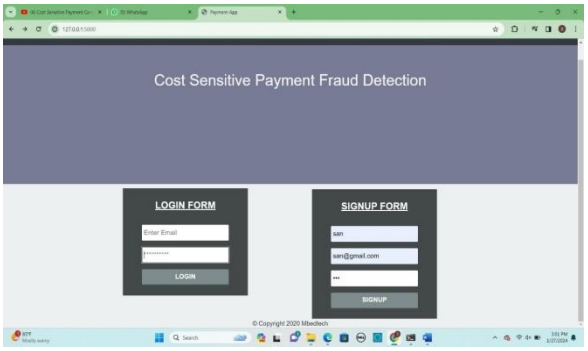
You will get an URL : http://127.0.0.1:5000/



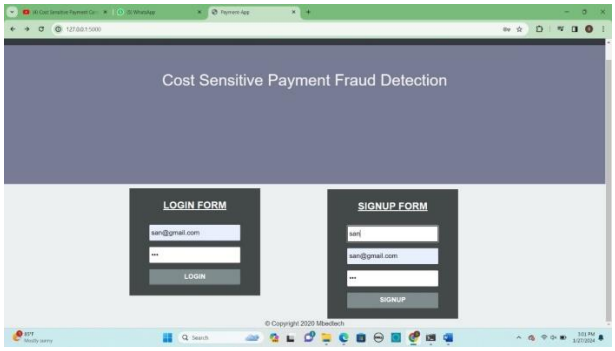
Paste the URL in web browser.



Enter the details to Sign Up.

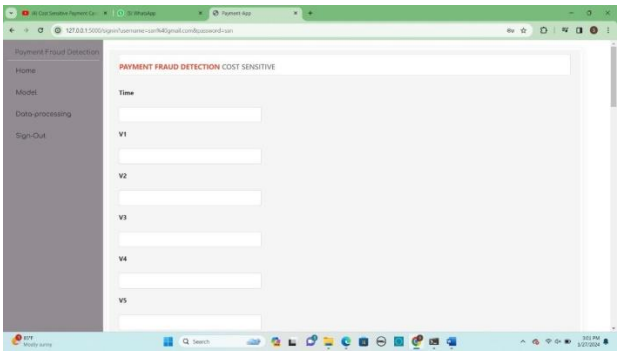


Use the same details to login the payment app.

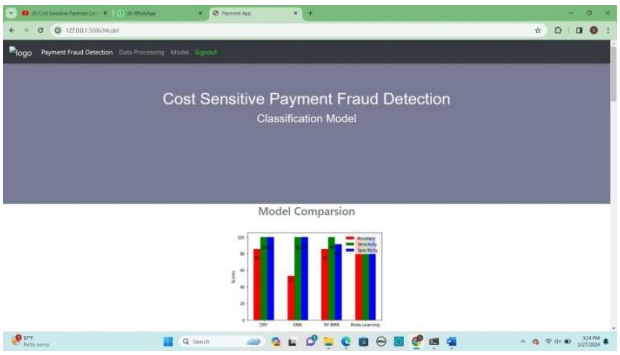


Now you will get the payment app.

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

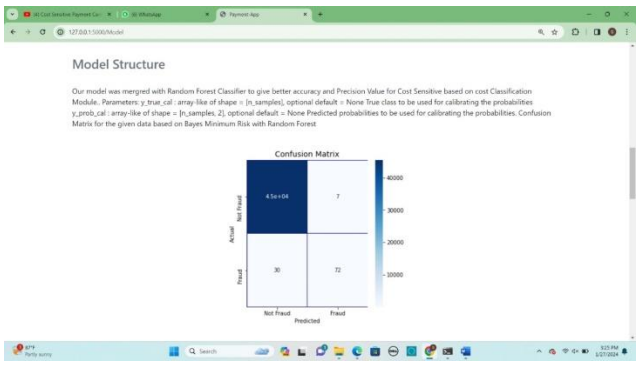


Model Comparison

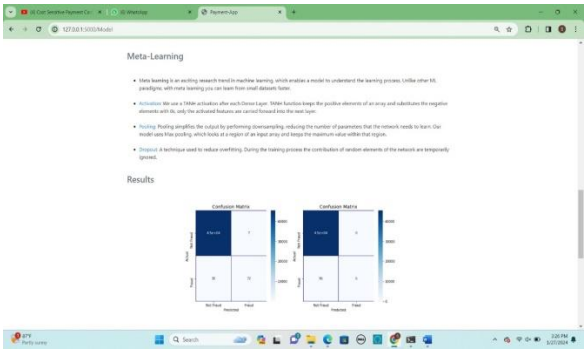


Accuracy

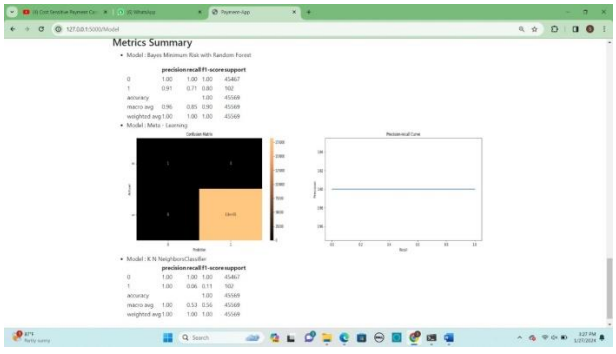
Model Structure



Meta-Learning

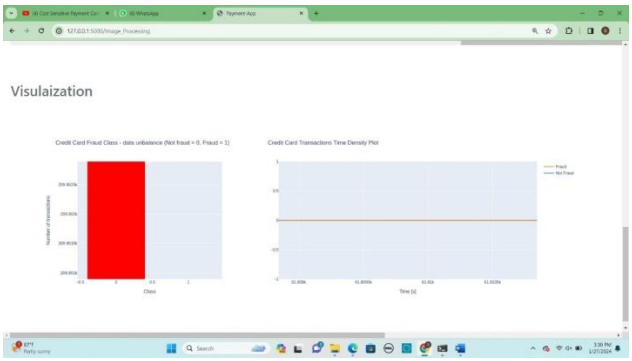


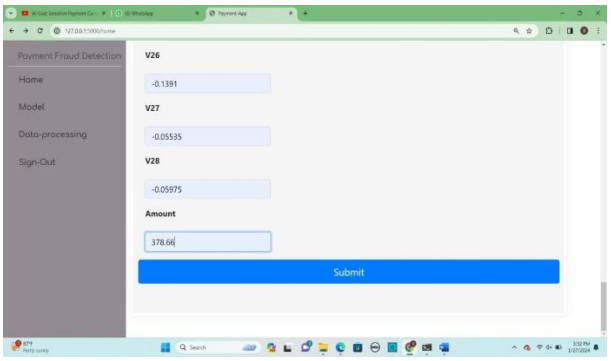
Metrics Summary



Visualization

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org





In the payment app, fill all the values from v1 to v28 , Time and Amount and then click submit.

6. EXPERMENTAL RESULTS

In this paper you are asking to use dataset which has low, high or medium risk but on internet we don't have any such dataset and there is only one European Credit Card dataset is there which has only two types of class labels such as NORMAL or FRAUD and we are using same dataset to implement this project.

You can read complete details from below URL about dataset used in this project

Dataset URL: https://www.kaggle.com/mlg-ulb/creditcardfraud

In this project we have designed following modules

1) Upload dataset: using this module we will upload dataset to application

2) Preprocess Dataset: In this module we will read all records and then preprocess them to remove missing values or to drop TIME column which is not require to build machine learning module. After preprocessing we will split dataset into train and test part where application use 80% dataset for training and 20% dataset for testing accuracy of trained prediction model.

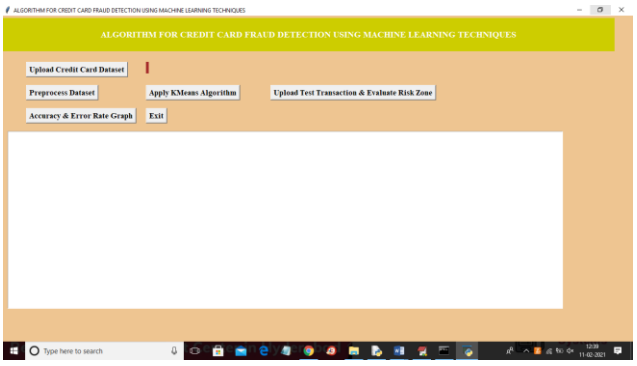
3) run KMEANS: Using this module we will divide train data into 2 clusters such as normal or fraud and then build prediction model.

4) Upload Test Transaction & Evaluate Risk Zone: In this module we will upload new test transactions data and then apply fuzzy logic and KMEANS prediction model to calculate accuracy of transaction as normal or fraud and if accuracy of transaction closer to zero then it will consider as NORMAL transaction and if its accuracy is closer to 1 then it will consider as fraud transaction.

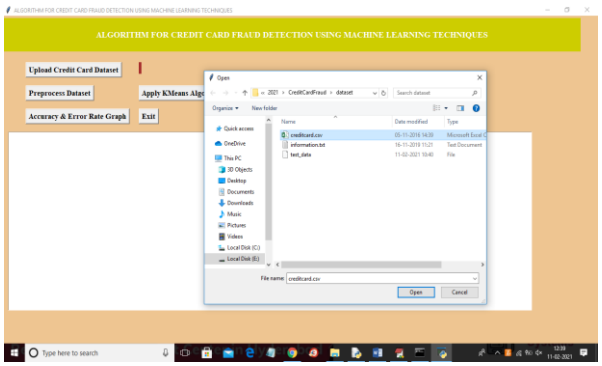
International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

Note: You are asking to get ACK and to perform deny transaction process but we don't have any real time application or data to perform this step just we build machine learning model using KMEANS and then we upload transaction test data and then predict test transaction signature is closer to NORMAL or FRAUD.

To run project double click on 'run.bat' file to get below screen



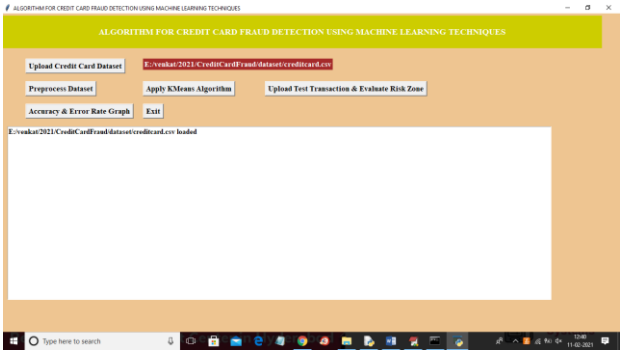
Upload Credit Card Dataset



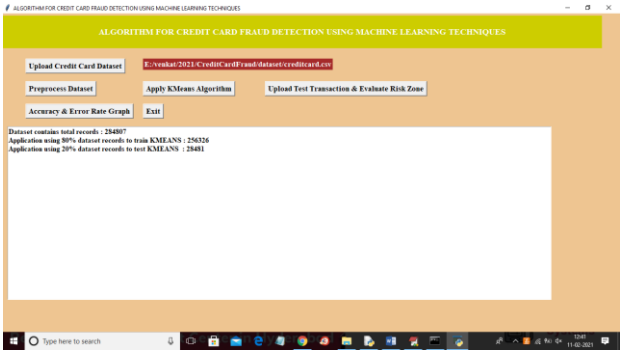
Uploading (creditcard.csv)

In above screen click on 'Upload Credit Card Dataset' button and then load dataset

In above screen selecting and uploading 'creditcard.csv' file and then click on 'Open' button to load dataset and to get below screen



Preprocess Dataset

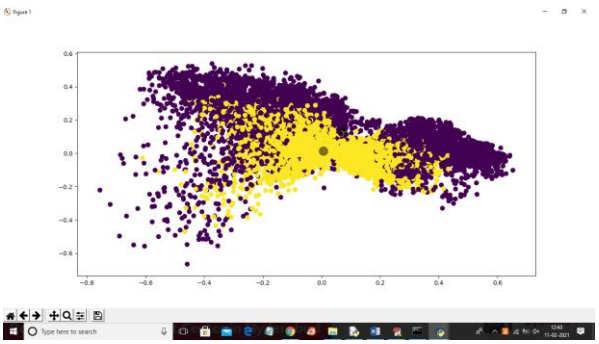


Apply K-Means Algorithm

In above screen dataset loaded and now click on 'Preprocess Dataset' button to remove missing values and to remove transaction TIME column and the split dataset into train and test part

In above screen we can see dataset contains 284807 records and then application using 90% (256326 records) for training and 28481 for testing and now both train and test data is ready and now click on 'Apply K-Means Algorithm' button to divide data into NORMAL and FRAUD and then build prediction model and to get below screen

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

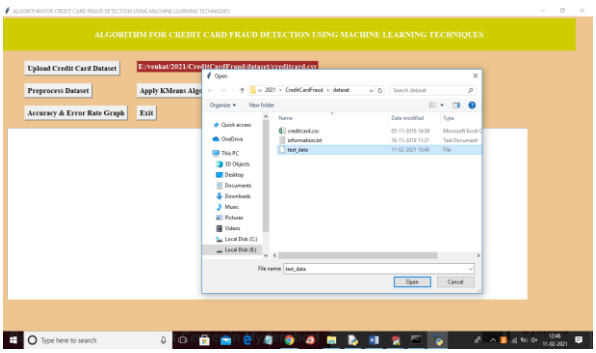


K-MEANS Accuracy

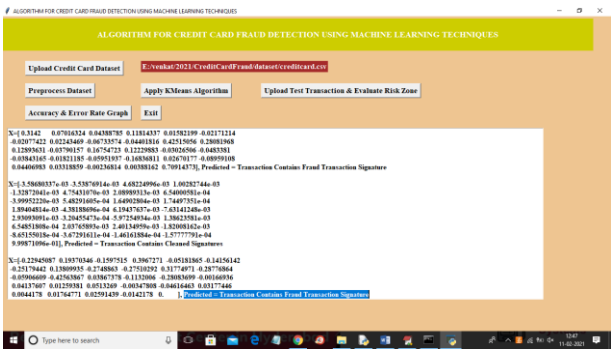
Upload Test Transaction & Evaluate Risk Zone

In above graph back colour dots are the normal transaction and yellow colour dots are the fraud transaction which is generated from KMEANS two clusters and in below screen we can see KMEANS accuracy

In above screen KMEANS prediction accuracy is 0.79% and now model is ready and now click on 'Upload Test Transaction & Evaluate Risk Zone' button to upload test data and then application apply fuzzy logic to evaluate test data signature into normal or fraud



Uploading (test_data.csv)

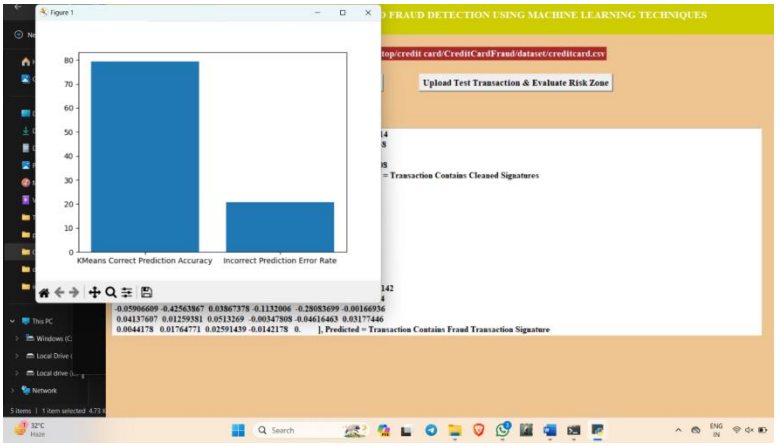


Transaction contains normal or fraud signature

In above screen selecting and uploading 'test_data.csv' file and then click on 'Open' button to get below result

In above screen data inside square brackets are the test data or transaction signature and after square bracket we can see prediction result as transaction contains normal or fraud signature.

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org



Accuracy & Error Rate graph

Similarly you can upload other transaction signature and perform prediction

7. CONCLUSION

Payment card fraud is a massive problem for the Banking sector. Hence, an effective fraud detection system for card payments is needed by any bank or financial institution to reduce the damages caused by fraudulent activities. In this research, we assumed that deviation from the normal behaviour of the cardholder could serve as the basis for fraud detection. Our experiments showed that the calculation of the similarity between existing transactions in a cardholder's profile and test transactions could be used for the efficient detection of payment card frauds. Moreover, our results showed that recent transactions exert considerable influence on evaluations of transactions as fraudulent or legal. We also realized that external causes such as a change in income and lifestyle of a cardholder might change cardholders' spending habits over time. Tree ensembles (such as dynamic random forests, random forests, gradient boosted trees, etc.) learn signals from both classes due to their hierarchical structure and have become very popular in solving problems with imbalanced data such as payment card fraud detection. Also, the use of DRF algorithm is appropriate when there are many input features, and it is known as an accurate and fast learning algorithm that runs efficiently on large datasets. We used DRF to re-examine

suspicious transactions and to prevent the occurrence of false positives. Our research results confirmed the effectiveness of DRF in payment card fraud detection

8.REFERENCES:

[1] Ahmed, M., Naser, A., & Islam, R. (2016). A survey of anomaly detection techniques in financial domain. Future Generation Computer Systems, 55, 278–288. http://doi.org/10.1016/j.future.2015.01.001

[2] Aleskerov, E., Freisleben, B., & Rao, B. (1997). Cardwatch: A neural network based database mining system for credit card fraud detection. In Computational Intelligence for Financial Engineering (CIFEr), Proceedings of the IEEE/IAFE (pp. 220–226).

[3] Bahnsen, A. C., & Aouada, D. (2014). Example-Dependent Cost-Sensitive Logistic Regression for Credit Scoring. In IEEE International Conference on Machine Learning and Applications (pp. 263–269). http://doi.org/10.1109/ICMLA.2014.48

[4] Bahnsen, A. C., Aouada, D., & Ottersten, B. (2015). Example-dependent cost-sensitive decision trees. Expert Systems With Applications, 42(19), 6609–6619. http://doi.org/10.1016/j.eswa.2015.04.042

[5] Bahnsen, A. C., Aouada, D., & Stojanovic, A. (2015). Detecting Credit Card Fraud using Periodic Features. In IEEE International Conference on Machine Learning and Applications (pp. 208–213

[6] Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32. This seminal paper introduces the concept of Random Forests and discusses their advantages in classification tasks, providing a solid theoretical foundation for your approach.

[7] Batista, G. E., & Monard, M. C. (2003). A study of K-nearest neighbour as an imputation method. In Intelligent Data Analysis (Vol. 6, No. 03, pp. 257-270). This study investigates the effectiveness of KNN in imputation tasks, which can provide insights into its applicability in fraud detection, especially in scenarios with missing or incomplete data.

[8] S. Makki, R. Haque, Y. Taher, Z. Assaghir, Mohand-Said Hacid, Hassan Zeineddine less Published in International Conference on Big Data and Cyber Security 13 December 2018. Presented a novel cost-sensitive KNN approach that we developed using Cosine Similarity (CoS). We compared our model with the other methods to verify its efficiency, and we proved using several performance measures that it's a better approach than other KNN algorithms.

[9] Olushola, A. , Mart, J. , (2022). Fraud Detection Using Machine Learning Techniques. 10.13140/RG.2.2.33044.88961/1. FRAUD DETECTION USING MACHINE LEARNING

[10] Dal Pozzolo, A., Boracchi, G., Caelen, O., Alippi, C., & Bontempi, G. (2015). Credit card fraud detection: a realistic modeling and a novel learning strategy. IEEE transactions on neural networks and learning systems, 29(8), 3784-3797.