A Survey on Credit Card Fraud Detection Using Holdout Cross Validation and Stratified K-fold Cross-Validation

A. Kavitha[1] *, P. Suganya[2], A. Gnana Prakash[3], Sarangam Jeevan[4], R.V. Vijay Kumar[5]

[1]Assistant Professor, Department of Computer Science(UG & PG), Dwaraka Doss Goverdhan Doss Vaishnav College, Ch-106. kavithaa@dgvaishnavcollege.edu.in
[2] Head, Assistant Professor Department of Computer Science(UG & PG), Dwaraka Doss Goverdhan Doss Vaishnav College, Ch-106. suganya@dgvaishnavcollege.edu.in
[3] 2nd M.Sc. Computer Science Department of Computer Science(UG & PG), Dwaraka Doss Goverdhan Doss Vaishnav College, Ch-106. Prakashgp029@gmail.com
[4] 2nd M.Sc. Computer Science Department of Computer Science(UG & PG), Dwaraka Doss Goverdhan Doss Vaishnav College, Ch-106. jeevanramesh768@gmail.com
[5] 2nd M.Sc. Computer Science Department of Computer Science(UG & PG), Dwaraka Doss Goverdhan Doss Vaishnav College, Ch-106. Vijayvidyarajesh6913@gmail.com

Article History
Received: **05.07.2023**
Revised and Accepted: **08.08.2023**
Published: **15.09.2023**

https://doi.org/10.56343/STET.116.017.001.001
www.stetjournals.com

ABSTRACT

In the area of credit card fraud detection, the implementation of machine learning algorithms is paramount to ensure the security of financial transactions. In this survey paper, we leverage the power of two robust cross-validation techniques, holdout and stratified K-fold cross-validation to explore decision trees, random forests, isolation forests, and k-means clustering in fraud detection scenarios. Considering the actual application, Decision Trees serve as an interpretable baseline, allowing transparent visualization of decision paths and helping identify fraudulent patterns. Random Forests, an ensemble of Decision Trees, reduce overfitting and improve prediction accuracy by aggregating various decision-making processes. An Isolation Forest, designed for anomaly detection excels at isolating anomalous credit card transactions. The ability to efficiently identify outliers provides effective protection against fraud. K-means clustering, on the other hand, divides transactions into clusters and highlights potential anomalies in the data. This paper evaluates the performance of these algorithms using Holdout and Stratified K-fold cross-validation. Holdout validation allows us to easily split the data training and test sets, while Stratified K-fold cross-validation ensures balanced class representation in each fold. This is important for fraud detection scenarios with imbalanced datasets. This survey report sheds light on the evolving landscape of credit card fraud detection by examining the use of these algorithms and cross-validation techniques to achieve high levels of accuracy and security in real-world finance.

Keywords: Credit Card Fraud Detection, Decision Tree, Holdout Cross Validation, Isolation Forest, K Means Clustering, Random Forest, Stratified K-Fold Cross Validation.

A. Kavitha

Assistant Professor, Department of Computer Science(UG & PG), Dwaraka Doss Goverdhan Doss Vaishnav College, Ch-106
email : kavithaa@dgvaishnavcollege.edu.in

P-ISSN 0973-9157
E-ISSN 2393-9249

INTRODUCTION

While the modern proliferation of digital transactions has brought unprecedented convenience, it also exposes the financial system to increased risks of fraud, especially credit card transactions. Detecting these fraudulent activities is critical for financial institutions and businesses to protect both assets and customer confidence. In the quest for effective fraud detection, machine learning algorithms have proven to be essential tools that promise to quickly and accurately identify anomalous or potentially fraudulent transactions. This paper examines his four major machine learnings: decision tree, random forest, isolation forest, and k-means clustering in the context of credit card fraud detection. These algorithms represent different approaches to address the multiple challenges of fraud. Decision trees provide interpretability and help you understand the logic behind fraud detection. Random forests harness the power of ensemble learning, while isolation forests specialize in efficiently isolating anomalies. K-means clustering is a versatile clustering technique that helps identify unusual patterns within transactions. To evaluate the effectiveness of these algorithms, we use two main cross-validation techniques:

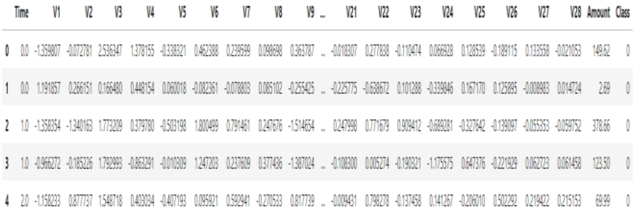
Holdout and Stratified K-fold cross-validation. The Holdout method makes it easy to split the data into a training set and a test set, allowing an initial assessment of the algorithm's performance. On the other hand, Stratified K-fold cross-validation ensures robust evaluation and addresses the problem of imbalanced datasets that often arise in fraud detection scenarios. This comprehensive study aims to shed light on the implementation of these algorithms and cross-validation strategies and provide insights into their strengths, weaknesses, and suitability for credit card fraud detection. As the financial landscape continues to evolve, understanding and leveraging the capabilities of these tools is critical to maintaining the integrity and security of digital transactions. The rest of the page is divided as follows: Section II describes relevant studies and addresses research gaps. Dataset acquisition, preprocessing, and feature selection are described in Section III. The methodology is described in Section IV. Section V presents the results and metrics used to improve the reliability of the proposed concepts. From the study, we illustrate the conclusion in Section VI.

LITERATURE REVIEW

In the field of credit card fraud detection, the classification challenge is essentially a dichotomy, distinguishing between legitimate and fraudulent transactions. This task becomes increasingly complex as fraud evolves (sameneh *et al*, 2016). Moreover, the imbalance between legitimate and fraudulent cases further complicates the detection process (Bolton, 2001; Duman *et al*, 2013). To overcome these challenges, cross-validation techniques that ensure reliable evaluation of the effectiveness of machine learning algorithms play a key role. Among these techniques, the holdout method and the stratified K-Fold cross-validation method are commonly used to evaluate model performance (Bolton, 2001). Machine learning algorithms such as decision trees, random forests, isolation forests, and k-means clustering offer different approaches to fraud detection (Kulatileke, 2022). Decision trees provide decision transparency, and the ensemble technique random forest improves accuracy and reduces overfitting. Isolation forest efficiently identifies anomalies, and K-means clustering segments transactions into meaningful clusters (15 Shocking Credit Card Fraud Statistics & Facts for 2022, 2022). Strategies to deal with imbalanced datasets include undersampling, oversampling, or a combination of them (Niveditha *et al* 2019; Duman *et al* 2013). Effective fraud detection requires a complex balance between supervised and unsupervised methods. Supervised methods train models on labeled data, while unsupervised methods detect outliers (Bolton, 2001). Numerous studies have implemented random forest algorithms and reported promising results in terms of accuracy, precision, recall, and F1 score (Bagga *et al.* 2019; Xuan *et al.* 2018*;* Kumar *et al.* 2019). However, the main challenge in detecting credit card fraud is not only accuracy but also achieving a balanced prediction of both fraudulent and non-fraud events (Bahnsen *et al*, 2016; Lever *et al*, 2016; Robinson and aria, 2018). This study addresses this issue by infusing the supervised and unsupervised machine learning algorithms with cross-validation techniques and finds which has better performance and also gives better prediction.

Dataset

The dataset contains 31 features and 2,84,807 online credit card transactions classified as fraudulent or legitimate. Once captured, the recordings were masked according to the client's privacy concerns and published on the Kaggle website. This includes only numeric input variables that are the result of PCA transformations. Unfortunately, for confidentiality reasons, we are unable to provide the original features or detailed background information of the data. Features V1,V2,V3,V4, …… V26, V27, V28 are the main components obtained in PCA. The only features that are not transformed by PCA are "time" and "amount". The Time function includes the number of seconds that have passed between each transaction and the first transaction in the record. The amount function is the transaction amount. This function can be used for example-dependent, cost-sensitive learning. Class is the response variable and takes a value of 1 if cheating and a value of 0 otherwise.



	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	...	-0.018307	0.277838	-0.110474	0.066928	0.128539	-0.189115	0.133558	-0.021053	149.62	0
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.225775	-0.638672	0.101288	-0.339846	0.167170	0.125895	-0.008983	0.014724	2.69	0
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	...	0.247998	0.771679	0.909412	-0.689281	-0.327642	-0.139097	-0.055353	-0.059752	378.66	0
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	...	-0.108300	0.005274	-0.190321	-1.175575	0.647376	-0.221929	0.062723	0.061458	123.50	0
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	...	-0.009431	0.798278	-0.137458	0.141267	-0.206010	0.502292	0.219422	0.215153	69.99	0

Figure 1: Dataset

Data preprocessing

The original dataset contains an extensive record of 2,84,807 transactions, among which a vast majority, approximately 2,84,315 transactions (99.82%), were deemed as non-fraudulent, while a much smaller fraction approximately 492 transactions (0.18%), were deemed fraudulent. This stark contrast in numbers between legitimate and fraudulent transactions underscores the inherent challenge posed by imbalanced datasets in credit card fraud detection. It's crucial to acknowledge that these transactions, whether genuine or fraudulent, often exhibit similar characteristics due to the evolving tactics of fraudsters, who continuously adapt to mimic the behavior of lawful cardholders. Consequently, the dynamics of both legitimate and fraudulent activities are in constant flux, rendering the direct use of this imbalanced dataset as input for our model impractical. The significant disparity in sample sizes between the positive (fraudulent) and negative (legitimate) classes depicts & highlights the risk of classifier bias toward predicting the negative class. This highlights the need to effectively address the problem of data imbalance (Banerjee *et al*, 2018).

METHODOLOGY

Supervised learning:

Supervised learning is a machine learning approach that uses datasets of labeled information to learn how algorithms make predictions and decisions. In this method, the algorithm is given a training data set consisting of input data and corresponding correct output labels. The goal of this algorithm is to understand the relationship between input data and output labels to make predictions and classifications based on unseen data. The main elements of learning include input data, output labels, training data, model representation, loss or cost function definition, running the training process, testing phases, and final evaluation.

Unsupervised learning:

Unsupervised learning is an approach in machine learning in which an algorithm is trained on a set of data with no output values or labeled targets. In other words, it involves learning patterns and structures in data without explicit instruction or supervision. The main goal of unsupervised learning is to discover patterns, relationships, or structures inherent in input data. Unsupervised learning is especially useful when dealing with large data sets or when there is no clear target variable to predict. It can help with exploratory data analysis, data preprocessing, and feature engineering, providing valuable insights into the underlying data structure. Common tasks include clustering and dimensionality reduction.

Decision Tree:

A decision tree is a diagram-like tree structure where each inner node represents the function, the branches represent the rules, and the leaf node represents the results of the algorithm. It is a general-purpose supervised machine learning algorithm used for both classification and regression problems.

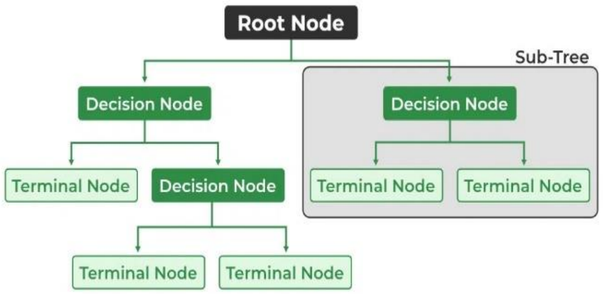


Figure 2: Representation of Decision Tree.

Random forest:

Random forest is an aggregation technique capable of performing both regression and classification tasks using multiple decision trees and a technique called Bootstrap and Aggregation, commonly known as bagging. The basic idea behind this is to combine multiple decision trees to determine the result instead of relying on individual decision trees. It has many decision trees as basic learning model.

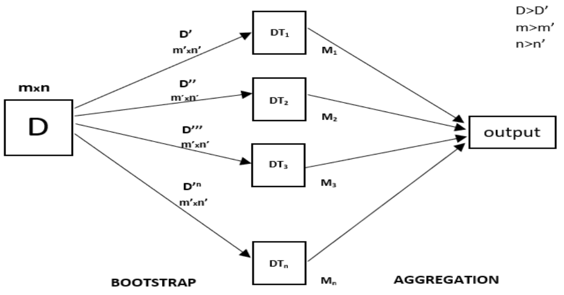


Figure 3: Working of Random Forest.

Isolation Forest:

Isolation forest is an algorithm for anomaly detection. This is an unsupervised learning algorithm that identifies outliers by isolating outliers in the data. Isolated forest based on decision tree algorithm. It isolates outliers byrandomly selecting a feature from a given set of features and then randomly selecting a value that is split between the maximum and minimum value of that feature. Randomly partitioning these objects creates shorter paths through the tree for the anomalous data points, thus distinguishing them from the rest of the data.

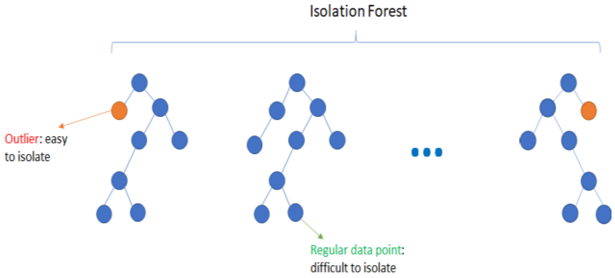


Figure 4: Representation of Isolation Forest.

K-Means Clustering:

K-Means is a popular clustering algorithm in machine learning and unsupervised learning. Its primary goal is to partition a dataset into a set of distinct, non-overlapping groups or clusters, where each data point belongs to the cluster with the nearest mean(center). K-Means aims to minimize the within-cluster variance, which is the sum of the squared distance between each other data points in a clustered group with its centroid. It does so by iteratively refining the cluster assignments and centroids until convergence. The result is a set of K clusters, each with its centroid, that groups similar data points together.

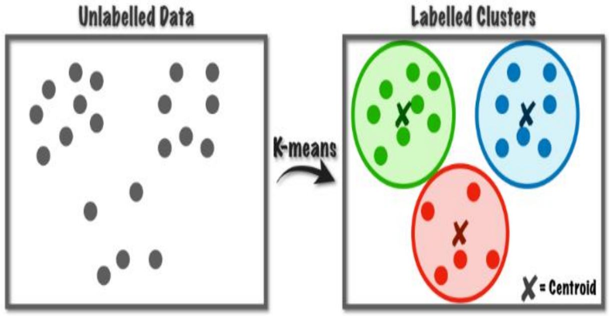


Figure 5: Representation of K-Means Clustering.

Cross Validation:

Cross-validation is a technique where we train our model using a subset of a dataset and then evaluate using an additional subset of the data set. The steps involved are:

- Reserve part of the data sample.
- Use the remaining dataset to train the model.
- Test the model using the reserve portion of the data set to cross-validate.

Holdout Cross Validation:

This is also known as training test split, the process of retaining cross-validation involves randomly partitioning the data set into training test split and the process of retaining cross-validation involves randomly partitioning the data set into training and validation sets. The general rule for data partitioning is that about 70% of the dataset will be used as the training set and the remaining 30% will be used as the validation set. Since the data set is only divided into two sets, the model is built once on the training set and is executed faster.

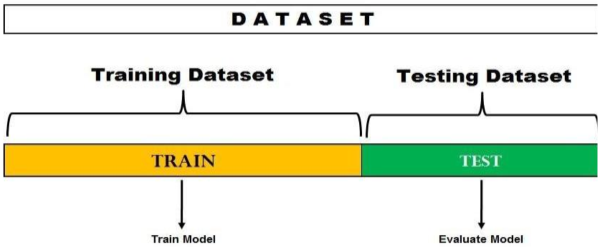


Figure 6: Representation of Holdout Cross Validation

Stratified K-Fold Cross Validation:

K-fold validation is not suitable for unbalanced datasets as it deals only with the data that is divided into k-folds with a uniform probability distribution. But k-fold stratification is an improved version of the k-fold cross-validation technique. Although it also divides the data set into k equal parts, each part has the same proportion of target variable instances as the full dataset. This allows it to work perfectly with unbalanced data sets, but not with time series data.

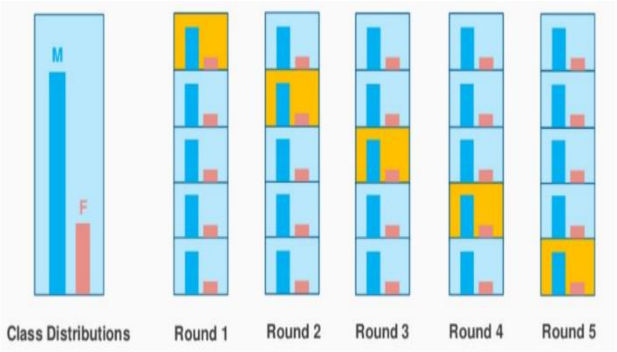


Figure 7: Representation of Stratified K-Fold Cross Validation

Evaluation Metrics:

a) Accuracy:

Accuracy is a common evaluation metric in machine learning (ML) that measures how well a classification model accurately predicts the class label of input data. It quantifies the ratio of correctly predicted cases to the total number of cases in the data set. Mathematically, the precision is calculated using the following formula:

Accuracy = No. of. Correct Predictions / Total No. of. Predictions

b) Precision:

Precision is an important evaluative metric in machine learning, especially in the context of classification problems. It measures the accuracy of the positive predictions made by the model, namely the proportion of correctly predicted positives (true positives) out of all the cases predicted to be positive. positive (true positive plus false positive). Precision is especially useful when you want to make sure that when your model predicts a positive class it is most likely correct. Mathematically, precision is defined as:

P = TP / TP + FP

Where,

- P – Precision
- TP – True Positive
- FP – False Positive

c) Recall:

Recall known as sensitivity or true positive rate, is an essential evaluative metric in machine learning, especially in classification tasks. It measures the model's ability to correctly identify all relevant instances of a particular class in the data set. Quantitative recall of the proportion of true positives (correctly predicted positivity) out of all actual positives. Mathematically, recall can be represented as:

R = TP / TP + FN

Where,
R – recall
TP – True Positive
FN – False Negative

d) F1 Score:

F1 score is a widely used evaluation metric in machine learning, especially for binary classification tasks. It combines both precision and recall into a single metric to provide a balanced measure of model performance. The F1 score is especially useful when you want to strike a balance between making accurate positive predictions (high precision) and capturing as many positives as possible (high recall). F1 score formula:

FS = 2 * (P * R) / P + R

Where,

- FS – F1 score
- P – Precision
- R – recall

e) ROC AUC:

ROC AUC, which stands for Receiver Operating Characteristic Area Under the Curve, is a metric commonly used in machine learning to evaluate the performance of binary classification models. It quantifies a model's ability to distinguish between positive and negative classes on different probability thresholds. The ROC curve is a graphical representation of the performance of a binary classifier as the discriminant threshold changes. It plots the true positive rate (sensitivity or recall) on the y-axis against the false-positive rate (1 - specificity) on the x-axis. Each point on the ROC curve represents a different threshold value, and the curve illustrates how the model's performance changes as you move the threshold. The AUC quantifies the overall performance of the model by calculating the area under the ROC curve.

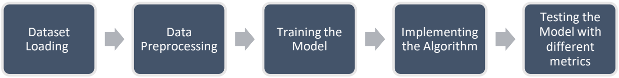
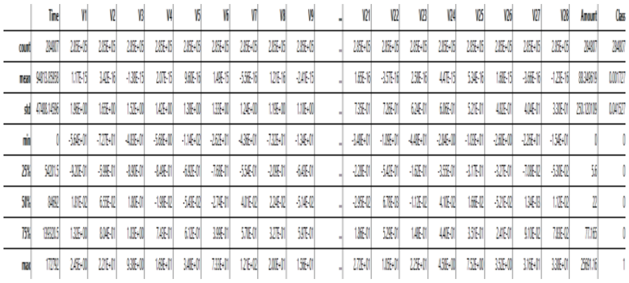

Dataset Loading
Data Preprocessing
Training the Model
Implementing the Algorithm
Testing the Model with different metrics

Figure 8: Framework for predicting fraud in credit card transactions.

RESULTS AND DISCUSSIONS:

Experiment:

The experimental work was done using Python language in Jupyter Notebook, Anaconda. The experiment aimed to offer a reliable fraud detection model that can successfully classify and detect fraudulent transactions. The dataset was split into training and testing data to avoid bias when running the models and to overcome the problem of the imbalanced dataset. Finally, we compare all the models with the usage of the cross-validation technique to find the best model with the best cross-validation technique to detect fraud in credit card transactions. The dataset contains 2,84,807 transactions with 31 features. The described method results in the count, mean, standard deviation, minimum, maximum, etc, Representation of the histogram for the Dataset is done using the Seaborn library in Python.



	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
count	284807	2.85E+05	2.85E+05	2.85E+05	2.85E+05	2.85E+05	2.85E+05	2.85E+05	2.85E+05	2.85E+05	...	2.85E+05	2.85E+05	2.85E+05	2.85E+05	2.85E+05	2.85E+05	2.85E+05	2.85E+05	284807	284807
mean	94813.85958	1.17E-15	3.42E-16	-1.38E-15	2.07E-15	9.60E-16	1.49E-15	-5.56E-16	1.21E-16	-2.41E-15	...	1.65E-16	-3.57E-16	2.58E-16	4.47E-15	5.34E-16	1.68E-15	-3.66E-16	-1.23E-16	88.349619	0.001727
std	47488.14596	1.96E+00	1.65E+00	1.52E+00	1.42E+00	1.38E+00	1.33E+00	1.24E+00	1.19E+00	1.10E+00	...	7.35E-01	7.26E-01	6.24E-01	6.06E-01	5.21E-01	4.82E-01	4.04E-01	3.30E-01	250.120109	0.041527
min	0	-5.64E+01	-7.27E+01	-4.83E+01	-5.68E+00	-1.14E+02	-2.62E+01	-4.36E+01	-7.32E+01	-1.34E+01	...	-3.48E+01	-1.09E+01	-4.48E+01	-2.84E+00	-1.03E+01	-2.60E+00	-2.26E+01	-1.54E+01	0	0
25%	54201.5	-9.20E-01	-5.99E-01	-8.90E-01	-8.49E-01	-6.92E-01	-7.68E-01	-5.54E-01	-2.09E-01	-6.43E-01	...	-2.28E-01	-5.42E-01	-1.62E-01	-3.55E-01	-3.17E-01	-3.27E-01	-7.08E-02	-5.30E-02	5.6	0
50%	84692	1.81E-02	6.55E-02	1.80E-01	-1.98E-02	-5.43E-02	-2.74E-01	4.01E-02	2.24E-02	-5.14E-02	...	-2.95E-02	6.78E-03	-1.12E-02	4.10E-02	1.66E-02	-5.21E-02	1.34E-03	1.12E-02	22	0
75%	139320.5	1.32E+00	8.04E-01	1.03E+00	7.43E-01	6.12E-01	3.99E-01	5.70E-01	3.27E-01	5.97E-01	...	1.86E-01	5.29E-01	1.48E-01	4.40E-01	3.51E-01	2.41E-01	9.10E-02	7.83E-02	77.165	0
max	172792	2.45E+00	2.21E+01	9.38E+00	1.69E+01	3.48E+01	7.33E+01	1.21E+02	2.00E+01	1.56E+01	...	2.72E+01	1.05E+01	2.25E+01	4.58E+00	7.52E+00	3.52E+00	3.16E+01	3.38E+01	25691.16	1

Figure 9: Statistical description of the dataset.

The correlation matrix provides the correlation coefficients between variables. Each cell in the table depicts the correlation between two variables. Correlation quantifies the strength between two variables. The most common correlation coefficient is the Pearson correlation coefficient (r), which ranges from -1 to 1:

- A positive correlation ($r > 0$) indicates that as one variable increases, the other tends to increase as well.
- A negative correlation ($r < 0$) indicates that as one variable increases, the other tends to decrease.
- A correlation coefficient of 0 ($r = 0$) suggests no linear relationship between the variables.

Correlation matrices are often used in statistics and data analysis for several purposes:

1. Data Exploration
2. Feature Selection
3. Multicollinearity Detection
4. Hypothesis Testing

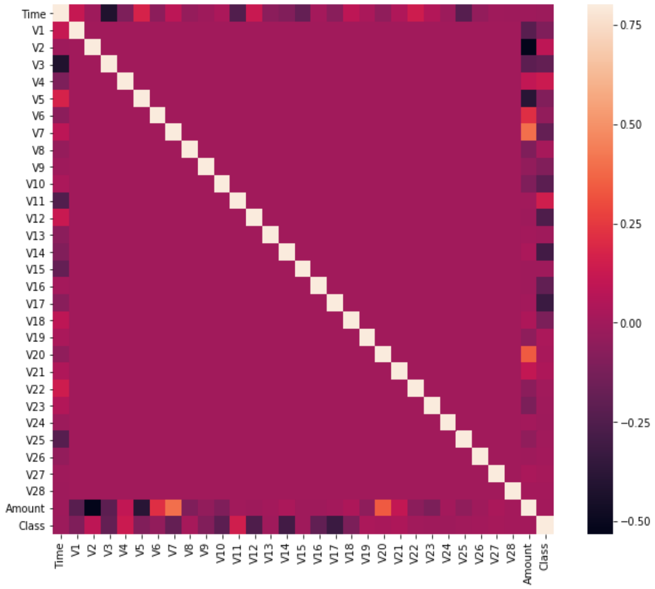


Figure 10: Correlation Matrix

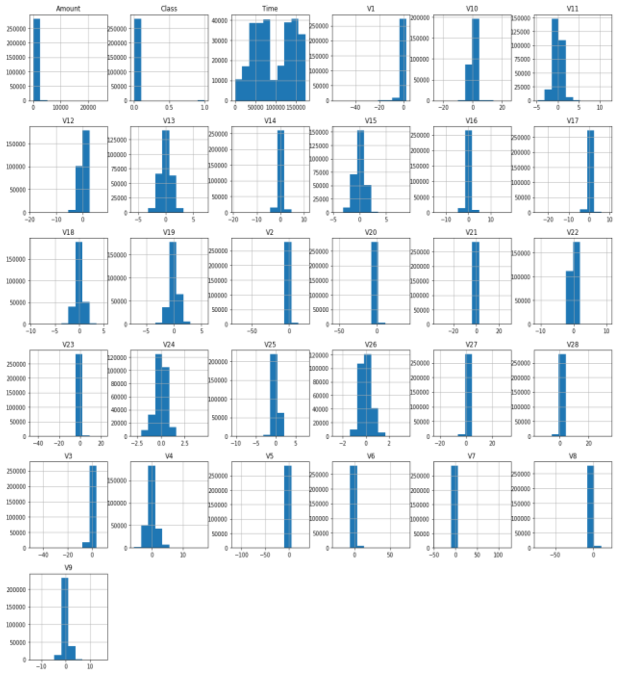


Figure 11: Histogram of dataset features.

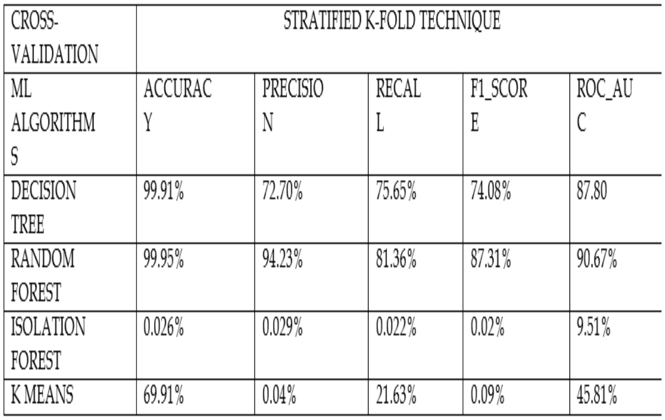
Metrics and results:

Although dealing with a highly imbalanced dataset, testing the algorithm showing only the accuracy is not sufficient to show the reliability of the algorithm. As a result, precision, recall, F1 score, and Receiver Operator Characteristic (ROC) curve measureswere applied. Accuracy, precision, recall, and F1 score measures. Accuracy is the proportion of correctly anticipated results. The overall accuracy of the classifier can be calculated by adding the number of true positives and true negatives and dividing by the total number of predictions. The classifier performs well in accuracy. However, accuracy is not always the best metric to evaluate a classifier's performance, especially when the classes are imbalanced. The classifier's performance should be evaluated in terms of othermetrics including precision, recall, and F1-score to get a better understanding of the classifier's performance. Precision is measured by the number of correctly identified outputs (Precision = [TP/(TP+FP)]). Recall is the percentage of True Positives that the model properly identified (Recall = [TP/(TP+FN)]). While the harmonic mean of precisionand recall is called F1-score (F1 score = [2*precision*recall / (precision + recall)]). In summary, the classifier has high accuracy, precision, recall, and F1-score, which are goodresults.

CROSS VALIDATION	HOLD OUT TECHNIQUE				
ML ALGORITHMS	ACCURACY	PRECISION	RECALL	F1_SCORE	ROC_AUC
DECISION TREE	99.93%	85.12%	73.04%	78.62%	91.90%
RANDOM FOREST	99.95%	97.46%	78.57%	87%	92.47%
ISOLATION FOREST	99.79%	38.82%	33.67%	36.06%	66.79%
K MEANS	45.33%	0.21%	65.06%	0.41%	55.21%

Table 1: Evaluation metrics values using Holdout Cross Validation Technique

As shown in Table 1, the supervised machine learning algorithms – decision tree & random forest models with the infusion of cross-validation result in very high performance & play an efficient role in detecting fraud in credit card transactions.



CROSS-VALIDATION	STRATIFIED K-FOLD TECHNIQUE				
ML ALGORITHMS	ACCURACY	PRECISION	RECALL	F1_SCORE	ROC_AUC
DECISION TREE	99.91%	72.70%	75.65%	74.08%	87.80
RANDOM FOREST	99.95%	94.23%	81.36%	87.31%	90.67%
ISOLATION FOREST	0.026%	0.029%	0.022%	0.02%	9.51%
K MEANS	69.91%	0.04%	21.63%	0.09%	45.81%

Table 2: Evaluation metrics values using Stratified K-fold Cross Validation Technique

As shown in Table 2, the supervised machine learning algorithms – decision tree & random forest models with the infusion of stratified K-fold cross-validation result in very high performance & play an efficient role in detecting fraud in credit card transactions.

Conclusion and Future Work:

In conclusion, the survey paper provides a comprehensive overview of the critical techniques and methodologies employed in the domain of credit card fraud detection. The study highlights the significance of cross-validation techniques, specifically Holdout Cross Validation and Stratified K-fold Cross-Validation, in assessing the performance of fraud detection models. The paper systematically examines various algorithms and evaluation metrics commonly used in credit card fraud detection. It discusses the advantages and limitations of each approach and offers valuable insights for researchers, practitioners, and organizations looking to enhance their fraud detection systems. For future enhancements, researchers may consider exploring novel machine learning algorithms, with other cross-validation techniques like LOOCV, etc, Furthermore, investigating the applicability of emerging technologies such as blockchain and explainable AI in fraud detection could be an exciting avenue for future research.

References:

Bagga S., A. Goyal, N. Gupta, and A. Goyal. 2020. Credit Card Fraud Detection using Pipeline and EnsembleLearning, *Procedia Comput Sci*, vol. 173, pp. 104–112, doi: 10.1016/J.PROCS.2020.06.014.

Bahnsen A. Correa, D. Aouada, A. Stojanovic, and B. Ottersten. 2016. Feature engineering strategies for credit card fraud detection, *Expert Syst Appl*, vol. 51, pp. 134–142, doi: 10.1016/J.ESWA.2015.12.030.

Banerjee R., G. Bourla, S. Chen, M. Kashyap, and S. Purohit. 2018. Comparative Analysis of Machine LearningAlgorithms through Credit Card Fraud Detection, *2018 IEEE MIT Undergraduate Research Technology Conference, URTC 2018*, Oct. 2018, doi: 10.1109/URTC45901.2018. 9244782.

Bolton R., D. H.-C. scoring and credit control VII, and undefined. 2001. Unsupervised profiling methods for fraud detection, *Citeseer*, Accessed: Jan. 26, 2023. [Online]. Available: https://citeseerx.ist.psu.edu/ document? repid=rep1&type=pdf&doi=5b640c367ae9cc4bd072006b05a3ed7c2d 5f496d

Duman E., A. Buyukkaya, and I. Elikucuk. 2013. A novel and successful credit card fraud detection system implemented in a Turkish bank, *Proceedings - IEEE 13th International Conference on Data Mining Workshops, ICDMW 2013*, pp. 162–171, doi: 10.1109/ICDMW.2013.168.

Fraud Detection Credit Card With Decision Tree | Kaggle.https://www.kaggle.com/code/ denidiana/fraud-detection-credit-card-with-decision-tree/notebook.

Kulatilleke, G. K., 2022. Challenges and Complexities in Machine Learning based Credit Card Fraud Detection, doi: 10.48550 /arxiv. 2208.10943.

Kumar M. S., V. Soundarya, S. Kavitha, E. S. Keerthika, and E. Aswini. 2019. Credit Card Fraud Detection Using Random Forest Algorithm, *2019 Proceedings of the 3rd International Conference on Computing and Communications Technologies, ICCCT 2019*, pp. 149–153, doi: 10.1109/ICCCT2.2019.8824930.

Lever J., M. Krzywinski, and N. Altman. 2016. Classification evaluation, *Nat. Methods, 13:8*, Jul. 2016, Accessed: Jan. 08, 2023. [Online]. Available:https://www.nature.com/articles /nmeth.3945

Niveditha G., K. Abarna, and G. v. Akshaya, “Credit Card Fraud Detection Using Random Forest Algorithm,” *Int. j. sci. res. comput. sci. eng. inf. technol*., 5(2). 301–306, doi: 10.32628/CSEIT195261.

Robinson W. N. and A. Aria. 2018. Sequential fraud detection for prepaid cards using hidden Markov model divergence, *Expert Syst Appl*, vol. 91, pp. 235–251, doi: 10.1016/J.ESWA.2017.08.043.

Samaneh, Sorournejad, Z. Zojaji, R. E. Atani, and A. H. Monadjemi, 2016. A Survey of Credit Card Fraud Detection Techniques: Data and Technique Oriented Perspective, doi: 10.48550/arxiv.1611.06439.

Xuan S., G. Liu, Z. Li, L. Zheng, S. Wang, and C. Jiang. 2018. Random forest for credit card fraud detection,*ICNSC 2018 - 15th IEEE International Conference on Networking, Sensing, and Control*, pp. 1–6, doi: 10.1109/ICNSC.2018.8361343.

15 Shocking Credit Card Fraud Statistics & Facts for 2022. https://moneytransfers.com /news /content /credit-card-fraud-statistics (accessed Dec. 25, 2022).