STM JOURNALS

Recent Trends in
Parallel Computing

https://journals.stmjournals.com/rtpc

ISSN: 2393-8749
Volume 11, Issue 1, 2024
January—April
DOI (Journal): 10.37591/RTPC

Review RTPC

# Cloud-driven Fraud Detection: Evaluating Decision Tree and Random Forest Classifiers for Credit Card Transaction Security

Anjana Verma[1,*], Nitya Khare[2]

***Abstract***

*With the alarming rise in global financial fraud, necessitating substantial annual losses, modern techniques for fraud detection are continuously evolving across various business domains. Fraud detection involves constant monitoring of user activities to estimate, perceive, or prevent undesirable behaviour. Cloud Computing emerges as a promising solution, accelerating application deployment, fostering creativity and innovation, reducing costs, and enhancing overall business acumen. This study introduces a cloud-driven approach to fraud detection, specifically tailored for credit card transactions. Leveraging machine learning applications deployed on Google Cloud Engine, the study employs classification techniques on a dataset comprising 284,807 credit card transactions. The chosen techniques adeptly process both numerical and categorical datasets. Decision tree and random forest classifiers are implemented for classification, and their performance is meticulously evaluated and compared through various metrics. The findings highlight the efficacy of the proposed cloud-driven strategy, shedding light on the comparative performance of decision tree and random forest classifiers in bolstering the security of credit card transactions. This research contributes valuable insights to the ongoing endeavours aimed at fortifying fraud detection mechanisms, providing a nuanced understanding for organizations grappling with the intricate challenges posed by fraudulent activities in the contemporary digital landscape.*

**Keywords:** Cloud computing, private cloud, machine learning application, security, SSH (Secure Shell)

## INTRODUCTION

Email communication has become a fundamental method for connecting vast numbers of individuals, necessitating adherence to communication standard protocols for efficient mail transmission. In this context, the need for robust text categorization algorithms is evident, especially for classifying emails into either ham (legitimate) or spam messages. While ham messages originate from genuine users and are desirable, spam messages, often generated by promotional companies, need to be identified and separated upon reaching the recipient's mobile station.

The repercussions of email spam extend beyond being merely annoying; they consume valuable time, resources, money, and network bandwidth. Although spam filtering software exists, its accessibility is limited. Furthermore, the misclassification problem arises when legitimate messages (ham) are mistakenly treated as spam, leading to potential service performance degradation and user dissatisfaction.

***Author for Correspondence**
Anjana Verma
E-mail: anjana9859@gmail.com

[1]Assistant Professor, Department of Computer Science & Engineering, Sagar Institute of Research & Technology (SIRT), Bhopal, Madhya Pradesh, India
[2]Assistant Professor, Department of Computer Science & Engineering, Sagar Institute of Research & Technology (SIRT), Bhopal, Madhya Pradesh, India

Received Date: January 25, 2024
Accepted Date: February 01, 2024
Published Date: April 04, 2024

**Citation:** Anjana Verma, Nitya Khare. Cloud-driven Fraud Detection: Evaluating Decision Tree and Random Forest Classifiers for Credit Card Transaction Security. Recent Trends in Parallel Computing. 2024; 11(1): 13–27p.

 

This study addresses the challenges posed by email spam, emphasizing its impact on user experience and the performance of email services. As email spam typically affects a broad audience and is disseminated through both mobile networks and the World Wide Web, there is a pressing need for effective spam detection solutions. Despite various methodologies and filtering techniques developed for email spam detection, researchers face challenges, including the scarcity of openly accessible datasets, which hinders the development and evaluation of robust detection models.

In light of these challenges, this study aims to contribute to the advancement of email spam detection methodologies, exploring innovative approaches to improve accuracy, minimize misclassifications, and enhance the overall efficiency of spam filtering systems.

**Brief of Dataset**

- *Context:* Credit card companies must effectively identify fraudulent transactions to prevent customers from being charged for unauthorized purchases.
- *Content:* The dataset consists of credit card transactions conducted by cardholders in Europe during the month of September 2013. It spans two days, with a total of 284,807 transactions, out of which, 492 are identified as fraud. Notably, the dataset is highly unbalanced, with frauds accounting for only 0.172% of all transactions.

**Features**

The dataset comprises solely numerical input variables derived from a Principal Component Analysis (PCA) transformation. Owing to confidentiality constraints, the original features and supplementary background information are withheld. Principal components V1 to V28 are obtained through PCA, while 'Time' and 'Amount' remain untransformed.

'Time' signifies the duration in seconds between each transaction and the initial transaction in the dataset, while 'Amount' indicates the transaction value and can be utilized for learning that takes into account cost sensitivity. The 'Class' variable functions as the dependent variable, with a value of 1 indicating fraud and 0 representing non-fraudulent instances.

## CLOUD COMPUTING OVERVIEW

Cloud Computing is a revolutionary computing paradigm characterized by the delivery of virtualized resources as services over the internet, with dynamic scalability [1]. It transforms the traditional approach to computing by providing remote management of computing resources, encompassing applications delivered as services and system software housed in data centres, collectively referred to as the "cloud" [2].

**Paradigm Shift**

Marking a significant paradigm shift, Cloud Computing has gained widespread adoption in enterprises due to its rapid implementation and deployment capabilities, improved customer experiences, scalability, and cost control. Notably, major enterprises are migrating their applications to the cloud to leverage these advantages [3].

**Key Considerations**

Reliability, availability, and security stand out as the three paramount concerns for organizations transitioning to the cloud. Businesses across various domains, such as customer relationship management (CRM), HR, accounting, and more, are seamlessly running applications in the cloud [3]. Prominent entities in the realm of cloud computing encompass Google, Amazon, Microsoft, and IBM.

**Virtualization as the Enabler**

Virtualization emerges as the key technology empowering Cloud Computing. It enables the provisioning of virtual servers, allowing for the efficient utilization of resources and adaptation to

 

fluctuations in demand. This transformation has evolved from simply renting infrastructure to providing and maintaining virtual servers, facilitating remote hosting [3].

**Leading Cloud Service Providers**

Among the frontrunners in cloud computing, Amazon stands out as an early adopter, introducing Amazon Web Services in 2005. Known for its maturity and longevity, Amazon's Web Services remains a cornerstone in the public cloud service domain. Microsoft Azure, with its comprehensive evolution of operating systems and strategic vision, has become a formidable player. Google, a pioneer in virtualization and cloud computing, continues to contribute significantly to the development and expansion of cloud-based services.

In summary, Cloud Computing represents a ground breaking approach to computing, offering unparalleled flexibility, scalability, and efficiency. Its impact extends across diverse sectors, influencing how businesses manage and leverage their computing resources in the digital era.

## DEPLOYMENT OF CLOUD

Cloud deployment refers to the manner in which cloud computing services are implemented and made available to users. Cloud computing typically employs three primary deployment models.

### Public Cloud

***Definition***

Third-party cloud service providers deliver public cloud services via the internet. These services are open for use by the general public.

***Characteristics***

- *Multi-tenancy:* Resources are shared among multiple users or organizations.
- *Accessibility:* Internet-accessible services are available on a pay-as-you-use model.
- *Scalability:* Easily scalable to accommodate varying workloads.

### Private Cloud

***Definition***

Exclusive to a single organization, private cloud services can be administered either internally or by a third party.

***Characteristics***

- *Dedicated:* Resources are solely allocated to a single organization, providing more control over data and security.
- *Customization:* Tailored to meet specific business requirements.
- *Security:* Enhanced security measures due to exclusive usage.

### Hybrid Cloud

***Definition***

Hybrid cloud integrates both public and private cloud environments, facilitating the sharing of data and applications between the two.

***Characteristics***

- *Flexibility:* Striking a balance between the advantages of public and private clouds is provided by this solution.
- *Workload Optimization:* Allows organizations to optimize workload placement based on specific requirements.
- *Data Portability:* Enables seamless movement of data between public and private environments.

 

## TYPES OF CLOUD

Clouds can be classified into various types based on the services they offer.

### Infrastructure as a Service (IaaS)

- *Definition:* IaaS offers users the ability to lease virtualized computing resources, including virtual machines, storage, and networks, through the internet.
- *Use Cases:* Ideal for businesses in need of a scalable computing infrastructure without the need for upfront investments in physical hardware.

### Platform as a Service (PaaS)

- *Definition:* Platform as a Service (PaaS) provides a platform that enables users to create, deploy, and oversee applications without having to contend with the intricacies of managing infrastructure.
- *Use Cases:* Designed for developers, it allows them to concentrate on developing applications without the need to handle the underlying infrastructure.

### Software as a Service (SaaS)

- *Definition:* SaaS distributes software applications via the internet, eliminating the necessity for users to locally install, maintain, and update the software.
- *Use Cases:* Frequently employed in tasks like managing emails, facilitating collaboration, and overseeing customer relationship management (CRM) functions.

These deployment and service models collectively contribute to the flexibility, scalability, and efficiency that characterize cloud computing, catering to diverse organizational needs and requirements.

## LITERATURE REVIEW

### "A Survey of Fraud Detection Techniques" [1]

***Key Points***

- Provides a comprehensive survey of various fraud detection techniques.
- Classifies fraud detection methods into rule-based, anomaly detection, and machine learning-based approaches.
- Examine the advantages and drawbacks of each approach, providing perspectives on the dynamic terrain of fraud detection as it continues to evolve.

### "Realistic Modelling and a Novel Learning Strategy for Detecting Credit Card Fraud" [2]

***Key Points***

- Proposes a realistic model for credit card fraud detection, considering imbalanced datasets.
- Introduces a novel learning strategy using a combination of under-sampling and over-sampling techniques to address class imbalance.
- Demonstrates improved performance in detecting credit card fraud compared to traditional methods.

### "Machine Learning Approaches to Fraud Detection and Prevention: A Survey" [3]

***Key Points***

- Examines the utilization of machine learning in the identification and mitigation of fraud.
- Explores various machine learning algorithms, including decision trees, neural networks, and ensemble methods, in the context of fraud detection.
- Examines the obstacles, prospects, and forthcoming trends within the realm of fraud detection utilizing machine learning.

 

**"Fraud Detection in Healthcare: A Review" [4]**
***Key Points***

- Focuses on fraud detection within the healthcare domain, acknowledging the unique challenges in this sector.
- Explores various forms of healthcare fraud, including fraudulent billing and prescription-related fraud.
- Reviews existing fraud detection methods and highlights the importance of advanced analytics and machine learning techniques in healthcare fraud prevention.

**"Deep Learning for Fraud Detection: A Comprehensive Review" [5]**
***Key Points***

- Delves into a comprehensive analysis of how deep learning techniques is employed in the realm of fraud detection.
- Explores the benefits of deep learning, highlighting its capacity to autonomously grasp complex patterns from data.
- Highlights various deep learning architectures, such as deep neural networks and recurrent neural networks, employed for fraud detection.

**"Fraud Detection in E-commerce: A Review" [6]**
***Key Points***

- Examines fraud detection challenges specific to the e-commerce sector.
- Examine both conventional and contemporary methods for detecting fraud, encompassing rule-based systems, anomaly detection, and machine learning algorithms.
- Discusses the importance of real-time processing and adaptive systems in addressing the dynamic nature of fraud in e-commerce.

**"Blockchain Technology for Fraud Detection: A Review" [7]**
***Key Points***

- Investigates the possibilities of leveraging blockchain technology to improve fraud detection.
- Discusses how the immutability and transparency of blockchain can contribute to creating secure and fraud-resistant systems.
- Reviews existing applications and research on integrating blockchain into fraud detection mechanisms.

This literature review provides insights into diverse aspects of fraud detection, including various techniques, applications in specific domains, and the role of emerging technologies like deep learning and blockchain. The surveyed studies collectively contribute to the evolving landscape of fraud detection methodologies [8].

## PROBLEM DEFINITION

*Problem Using Cloud:* The adoption of cloud computing, which relies on the Internet Protocol (IP) and often utilizes Transport Control Protocol (TCP), faces significant challenges, primarily centred around security concerns. The apprehension arises from the idea of entrusting data and software to external servers, raising issues of data loss, phishing, and the potential threat of botnets [9]. The dynamic allocation of different IP addresses to virtual machines in the cloud introduces complexities in maintaining security and data integrity.

*Problem in Credit Card Data Classification:* The implementation of an effective fraud detection technique encounters specific challenges that impact its performance:

**Imbalanced Data**

- *Challenge:* The dataset for credit card transactions exhibits an imbalanced nature, with fraudulent instances being a minority.

 

- *Impact:* Imbalanced data can lead to biased model training, affecting the ability to accurately identify fraud, as the model may be skewed toward the majority class.

**Different Misclassification Importance**

- *Challenge:* Various misclassification errors carry different levels of importance in credit card fraud detection.
- *Impact:* Assigning equal weight to different misclassifications may not reflect the real-world impact, where certain errors, such as false negatives, can be more detrimental than others.

**Overlapping Data**

- *Challenge:* The dataset contains transactions that overlap, where some fraudulent transactions may appear similar to normal transactions, and vice versa.
- *Impact:* Managing a low false positive and false negative rate becomes challenging due to the overlapping nature of data, complicating the task of distinguishing between genuine and fraudulent transactions.

Addressing these challenges in credit card data classification is essential for developing robust and accurate fraud detection techniques, ensuring the integrity and security of financial transactions in an ever-evolving digital landscape.

## PROPOSED WORK

In this proposed work, we advocate the utilization of a private cloud over other cloud deployment models for enhanced security, with a specific emphasis on using secure shell (SSH) for the connection between users and virtual machines [10]. The private cloud offers heightened security measures, crucial for safeguarding sensitive data in comparison to public and hybrid clouds. The ability to scale storage and processing power dynamically in response to application demands adds to the flexibility of the private cloud infrastructure as shown in Figure 1.

### Block Diagram

#### *Steps*

*User Login*

Users access the cloud service website through a browser.

Upon successful login, a secure SSH connection is established between the user's browser and the virtual machine on the private cloud through API calls [11].

### Machine Learning Application Deployment

A machine learning application is deployed on the virtual machine within the private cloud environment as shown in Figure 2.

## PROPOSED MODEL

### Decision Tree Algorithm

#### *Creating a Root Node*

*Initialize a Root Node:*

If every tuple in the dataset belongs to the same class, generate a leaf node marked with that class. In case the attribute list is devoid of elements, generate a leaf node labelled with the predominant class in the dataset [12].

#### *Attribute Selection*

- Utilize the attribute selection method to identify the most optimal attribute for the splitting criterion.
- Label the node with the chosen splitting criterion.
- Update the attribute list.

 

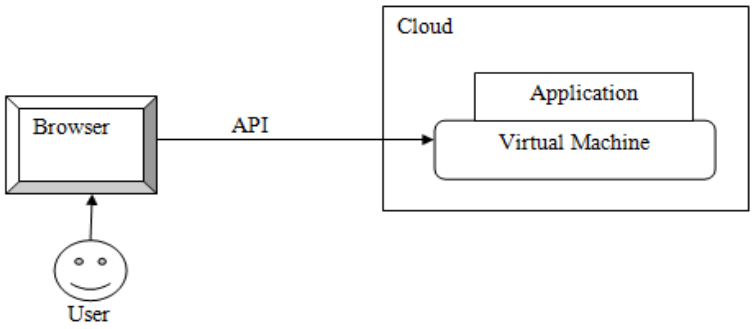


**Figure 1.** Proposed block diagram.

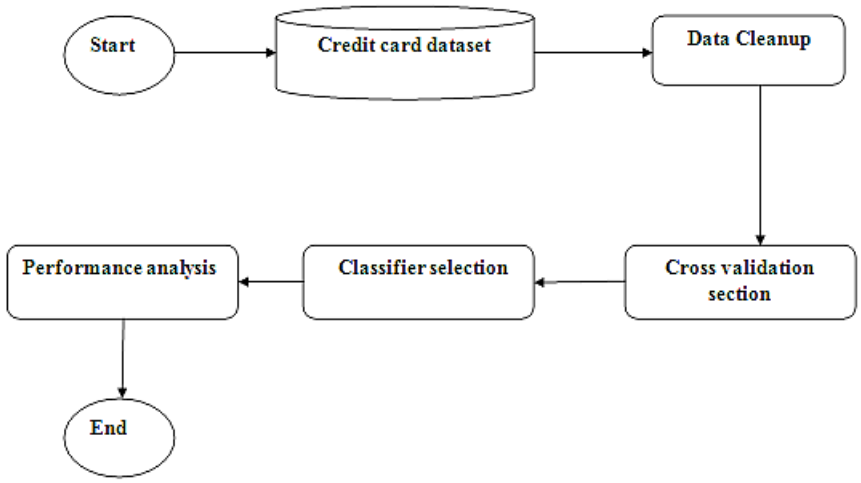


**Figure 2.** Proposed model.

***Split the Tree***

Divide the tree based on each outcome of the splitting criterion, creating two branches for each. Add a leaf labelled with the majority class to nodes that do not undergo further splitting. For other nodes, recursively apply the Decision Tree algorithm.

**Random Forest Algorithm**

Build decision trees for individual subsets of data, using a random subset of features for each tree. Aggregate the decision trees by determining the majority vote across all the trees [13]. The proposed network Intrusion Detection System (IDS) is based on Decision Tree and Random Forest classifiers, providing a robust framework for detecting and mitigating network-based threats. The private cloud deployment ensures security and scalability, making it suitable for handling sensitive applications and data [14].

## IMPLEMENTATION AND RESULT ANALYSIS

**Virtual Machine on Cloud Environment (Figure 3)**

A virtual machine on the cloud (preferably Google Cloud Platform-GCP) is chosen for the experiment (Figure 3). The use of a private cloud ensures better security, with the connection between users and the virtual machine secured by SSH. The ability to scale storage and processing power dynamically in response to application requirements enhances flexibility [15].

**Starting Python Over Cloud Computing Machine (Figure 4)**

After starting the virtual machine, a secure channel (SSH) is established between the user's browser and the virtual machine's terminal. Python environment is initiated on the cloud computing machine, providing a platform for deploying machine learning code for classification.

 

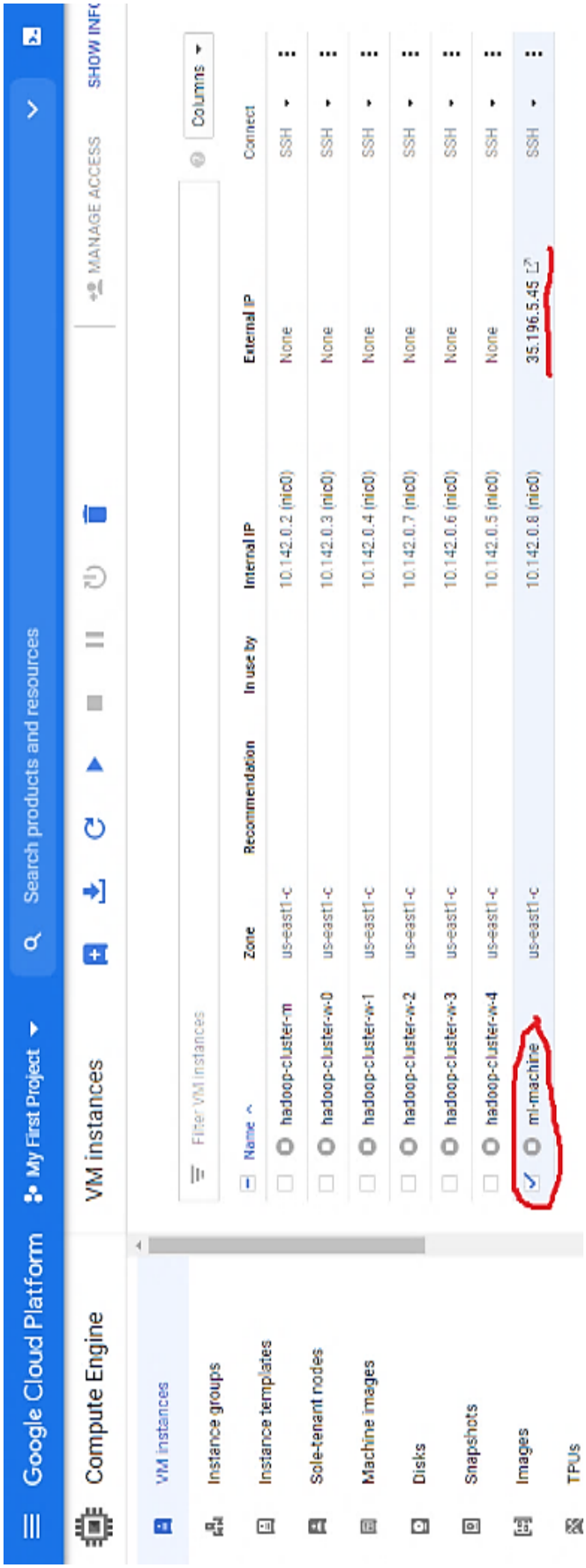


**Figure 3.** Virtual machine on cloud environment.

 

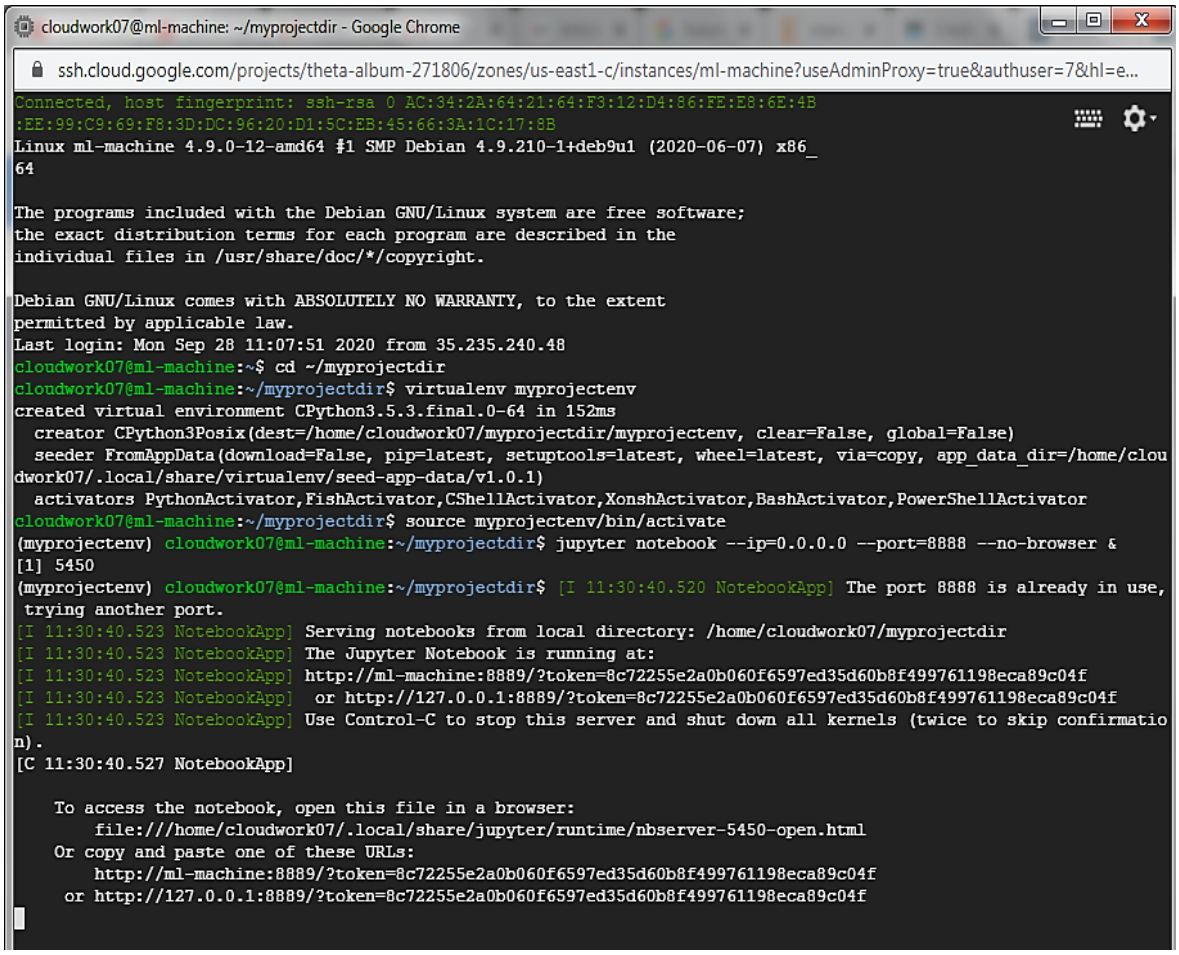


**Figure 4.** Starting Python over cloud computing machine.

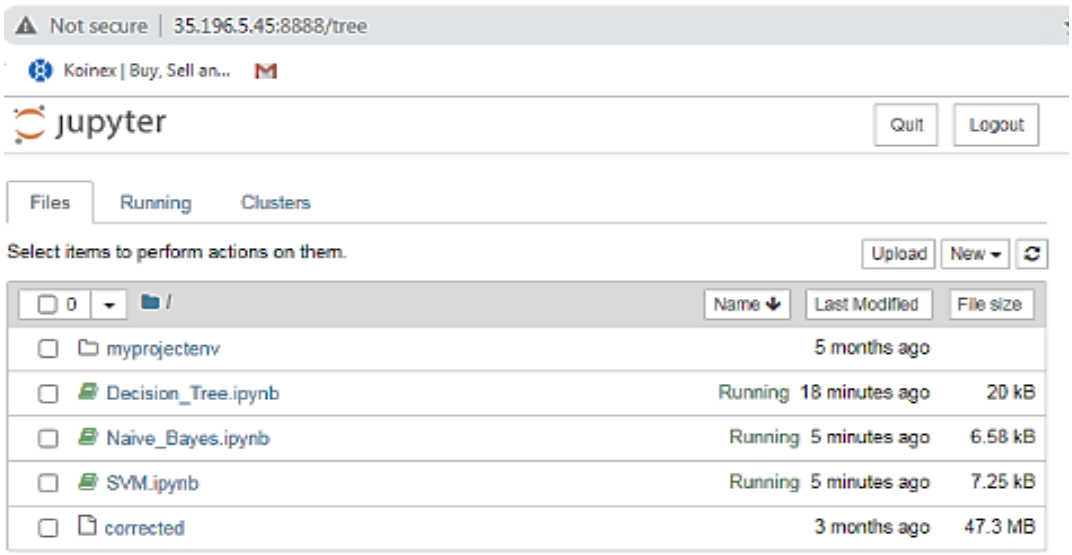


**Figure 5.** Jupyter notebook.

**Jupyter Notebook (Figure 5)**

Python and Jupyter Notebook are connected through a web browser using a secure IP and port number. Authentication, such as a secure token or password, ensures secure access to Jupyter Notebook on the cloud machine.

 

**Data Set Details (Figure 6)**

The credit card dataset, consisting of 284,807 observations and 31 variables, is loaded into Python for experimentation. Figure 7 displays the correlation of feature engineering with the class variable.

Correlation analysis is conducted to examine the associations between features and the target variable (Class) in order to gain insights into their relationships.

Correlation coefficients help identify the intensity and direction of connections, assisting in the process of selecting features.

**Feature Scaling (Figure 8)**

Feature scaling is applied to bring features onto the same scale.

Standardization is used to centre feature columns at mean 0 with standard deviation 1, facilitating learning weights and making the algorithm less sensitive to outliers.

**Model Training-Decision Tree (Figure 9)**

Decision Tree algorithm is employed for classification.

The model has been trained on a dataset that has undergone preprocessing. Performance measures, including accuracy, precision, recall, and F1 score, are computed to evaluate the effectiveness of the Decision Tree classifier.

## RESULT ANALYSIS

The performance measures obtained from the Decision Tree model provide insights into its accuracy, precision, recall, and F1 score.

These metrics aid in evaluating the model's capacity to accurately differentiate between instances of fraudulent and non-fraudulent activities within the credit card dataset.

Further analysis and comparison with other classification algorithms, such as Random Forest, may be conducted to determine the most suitable model for fraud detection in credit card transactions.

**Random Forest**

The Random Forest model constitutes an ensemble of classification or regression trees. Decision tree models are widely utilized in data mining owing to their user-friendly nature, adaptability in handling diverse data attribute types, and interpretive capabilities. Nevertheless, individual tree models may exhibit instability and excessive sensitivity to particular training data. Ensemble methods, such as Random Forest aims to tackle this problem by generating a collection of models and combining their predictions to ascertain the class label for a given data point.

***Key Characteristics of Random Forest***

Each tree within the ensemble is constructed using an independent bootstrap sample derived from the training data. This involves randomly sampling instances with replacement, creating a diverse set of training datasets for each tree.

The tree evaluates the best split at each node by considering only a randomly chosen subset of attributes. This introduces further diversity among individual trees in the ensemble.

Create an individual tree in the ensemble by first obtaining a bootstrap sample of N cases. At each node, choose a subset of b attributes randomly, and identify the optimal split at the node using this reduced set of attributes. Expand the entire tree without implementing pruning.

 

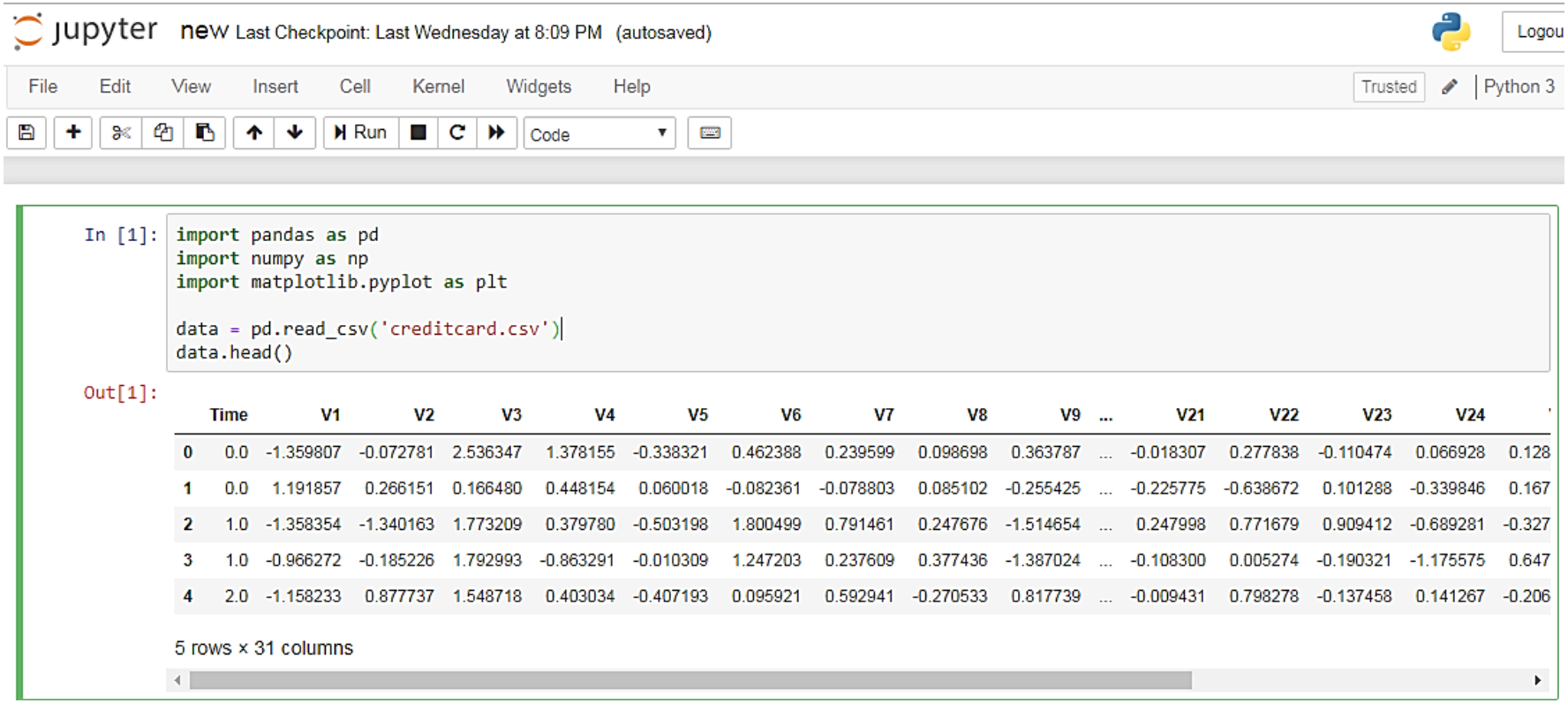


**Figure 6.** Data set details.

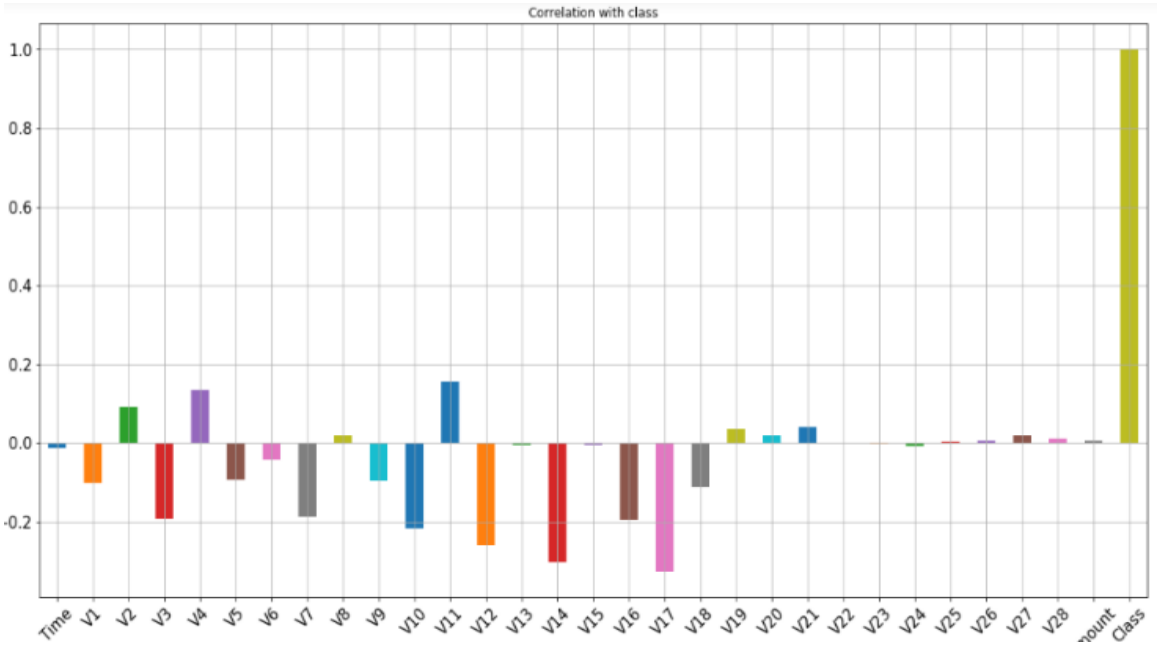


**Figure 7.** Correlation with Class.

```
#Feature Scaling

from sklearn.preprocessing import StandardScaler
data['normalizedAmount'] = StandardScaler().fit_transform(data['Amount'].values.reshape(-1,1))
data = data.drop(['Amount'],axis=1)
data = data.drop(['Time'],axis=1)
data.head()
```

| | V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 | V9 | V10 | ... | V21 | V22 | V23 | V24 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | -1.359807 | -0.072781 | 2.536347 | 1.378155 | -0.338321 | 0.462388 | 0.239599 | 0.098698 | 0.363787 | 0.090794 | ... | -0.018307 | 0.277838 | -0.110474 | 0.066928 | |
| 1 | 1.191857 | 0.266151 | 0.166480 | 0.448154 | 0.060018 | -0.082361 | -0.078803 | 0.085102 | -0.255425 | -0.166974 | ... | -0.225775 | -0.638672 | 0.101288 | -0.339846 | |
| 2 | -1.358354 | -1.340163 | 1.773209 | 0.379780 | -0.503198 | 1.800499 | 0.791461 | 0.247676 | -1.514654 | 0.207643 | ... | 0.247998 | 0.771679 | 0.909412 | -0.689281 | |
| 3 | -0.966272 | -0.185226 | 1.792993 | -0.863291 | -0.010309 | 1.247203 | 0.237609 | 0.377436 | -1.387024 | -0.054952 | ... | -0.108300 | 0.005274 | -0.190321 | -1.175575 | |
| 4 | -1.158233 | 0.877737 | 1.548718 | 0.403034 | -0.407193 | 0.095921 | 0.592941 | -0.270533 | 0.817739 | 0.753074 | ... | -0.009431 | 0.798278 | -0.137458 | 0.141267 | |

5 rows × 30 columns

**Figure 8.** Feature scaling.

```
## Decision Tree
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import f1_score

classifier = DecisionTreeClassifier(random_state = 0,
                                    criterion = 'gini',  splitter='best', min_samples_leaf=1, min_samples_split=2)
classifier.fit(X_train, y_train)

# Predicting Test Set
y_pred = classifier.predict(X_test)
acc = accuracy_score(y_test, y_pred)
prec = precision_score(y_test, y_pred)
rec = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)

results = pd.DataFrame([['Decision tree', acc, prec, rec, f1]],
               columns = ['Model', 'Accuracy', 'Precision', 'Recall', 'F1 Score'])
```

```
results
```

| | Model | Accuracy | Precision | Recall | F1 Score |
|---|---|---|---|---|---|
| 0 | Decision tree | 0.999333 | 0.835821 | 0.761905 | 0.797153 |

**Figure 9.** Model training-decision tree.

 

***Benefits of Random Forest***

*Computational Efficiency:* Each tree is built independently of others, making Random Forest computationally efficient.

***Robustness***

Random Forest exhibits robustness to overfitting and noise in the data due to its utilization of a substantial number of trees in the ensemble.

***Performance Measure of Random Forest (Figure 10)***

The performance measure of the Random Forest model includes metrics such as accuracy, precision, recall, and F1 score.

The ensemble nature of Random Forest allows for a comprehensive evaluation of its effectiveness in classifying instances.

**Accuracy of Models (Table 1)**

The table provides accuracy values for different models, showcasing the performance of each model in terms of correctly classifying instances.

**Performance of the Models (Figure 11)**

The Figure 11 illustrates the performance of different models, offering a visual representation of their accuracy and effectiveness.

```
## Randomforest
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix, accuracy_score, f1_score, precision_score, recall_score
classifier = RandomForestClassifier(random_state = 0,n_estimators = 100,
                                    criterion = 'entropy')
classifier.fit(X_train, y_train)

# Predicting Test Set
y_pred = classifier.predict(X_test)
acc = accuracy_score(y_test, y_pred)
prec = precision_score(y_test, y_pred)
rec = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
model_results = pd.DataFrame([['Random Forest (n=100)', acc, prec, rec, f1]],
               columns = ['Model', 'Accuracy', 'Precision', 'Recall', 'F1 Score'])
results = results.append(model_results, ignore_index = True)
```

```
C:\Users\abhishek\Anaconda3\lib\site-packages\ipykernel_launcher.py:8: DataConversionWarning: A column
a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
```

```
results
```

| | Model | Accuracy | Precision | Recall | F1 Score |
|---|---|---|---|---|---|
| 0 | Decision tree | 0.999333 | 0.835821 | 0.761905 | 0.797153 |
| 1 | Random Forest (n=100) | 0.999520 | 0.941667 | 0.768707 | 0.846442 |

**Figure 10.** Performance measure of random forest.

**Table 1.** Accuracy of models.

| | Model | Accuracy | Precision | Recall | F1 Score |
|---|---|---|---|---|---|
| 0 | Decision Tree | 0.999333 | 0.835821 | 0.761905 | 0.797153 |
| 1 | Random Forest (n=100) | 0.999520 | 0.941667 | 0.768707 | 0.846442 |

 

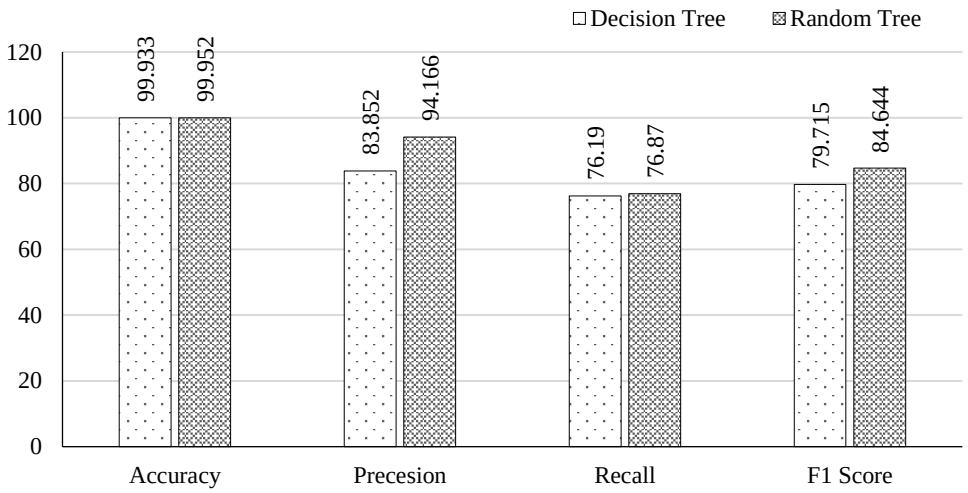


**Figure 11.** Performance of the models.

## CONCLUSION

Cloud computing stands as a highly flexible paradigm for delivering computational power, catering to diverse needs and interpretations. It offers solutions to challenges such as scalability, resource management, and cost control. In this context, utilizing a virtual machine on a private cloud, especially through third-party providers like Google Cloud Platform (GCP), emerges as a secure and scalable approach.

The preference for a private cloud is driven by security considerations, with the connection between users and virtual machines being secured by SSH. This ensures a robust and protected environment for deploying applications. The scalability of storage and processing power on a private cloud adds to the appeal, allowing dynamic adjustments based on application requirements.

As credit card usage becomes ubiquitous in various aspects of daily life, the prevalence of credit card fraud has increased. This study delves into the binary classification of imbalanced credit card fraud data using Decision Tree and Random Forest models. The emphasis is on comparing the performance of these classification models.

The comparative results highlight the superiority of the Random Forest model over Decision Tree in handling imbalanced credit card fraud data. Random Forest's ensemble approach, incorporating multiple decision trees, proves to be more effective in achieving accurate and robust predictions. As the demand for secure and efficient fraud detection mechanisms grows, the adoption of Random Forest models in credit card fraud detection becomes increasingly evident.

## REFERENCES

1. Kou Y, Lu CT, Sirwongwattana S, Huang YP. Survey of fraud detection techniques. In IEEE International Conference on Networking, Sensing and Control, 2004. 2004 Mar 21; 2: 749–754.
2. Dal Pozzolo A, Boracchi G, Caelen O, Alippi C, Bontempi G. Credit card fraud detection: a realistic modeling and a novel learning strategy. IEEE Trans Neural Netw Learn Syst. 2017 Sep 14; 29(8): 3784–97.
3. Adewumi AO, Akinyelu AA. A survey of machine-learning and nature-inspired based credit card fraud detection techniques. Int J Syst Assur Eng Manag. 2017 Nov; 8(2): 937–53.
4. Thaifur AY, Maidin MA, Sidin AI, Razak A. How to detect healthcare fraud? “A systematic review”. Gac Sanit. 2021 Jan 1; 35(Suppl 2): S441–9.
5. Al-Hashedi KG, Magalingam P. Financial fraud detection applying data mining techniques: A comprehensive review from 2009 to 2019. Comput Sci Rev. 2021 May 1; 40: 100402.

 

6. Joshi K. A review of credit card fraud detection techniques in e-commerce. Acad J Forensic Sci. 2018; 1(01): 5–11.
7. Oladejo MT, Jack L. Fraud prevention and detection in a blockchain technology environment: challenges posed to forensic accountants. International Journal of Economics and Accounting (IJEA). 2020; 9(4): 315–35.
8. Gupta A, Goswami P, Chaudhary N, Bansal R. Deploying an application using google cloud platform. In 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA). 2020 Mar 5; 236–239.
9. Parra-Royon M, Benítez JM. Delivering data mining services in cloud computing. In 2019 IEEE World Congress on Services (SERVICES). 2019 Jul 8; 2642: 396–397.
10. Awoyemi JO, Adetunmbi AO, Oluwadare SA. Credit card fraud detection using machine learning techniques: A comparative analysis. In 2017 IEEE international conference on computing networking and informatics (ICCNI). 2017 Oct 29; 1–9.
11. Bolton RJ, Hand DJ. Statistical fraud detection: A review. Stat Sci. 2002 Aug; 17(3): 235–55.
12. Dheepa V, Dhanapal R. Analysis of credit card fraud detection methods. International Journal of Recent Trends in Engineering (IJRTE). 2009 Nov 1; 2(3): 126.
13. Chan PK, Fan W, Prodromidis AL, Stolfo SJ. Distributed data mining in credit card fraud detection. IEEE Intelligent Systems and Their Applications. 1999 Nov; 14(6): 67–74.
14. Sivakumar N, Balasubramanian DR. Credit Card Fraud Detection: Incidents, Challenges and Solutions. Int J Adv Res Comput Sci Appl. 2015; 6(2): 1379-1386.
15. Srivastava A, Kundu A, Sural S, Majumdar A. Credit card fraud detection using hidden Markov model. IEEE Trans Dependable Secure Comput. 2008 Feb 8; 5(1): 37–48.

 