UJEAS

UNIZIK JOURNAL OF ENGINEERING AND APPLIED SCIENCES

UNIZIK Journal of Engineering and Applied Sciences 2(1), June 2023, 187-195
Journal homepage: https://journals.unizik.edu.ng/index.php/ujeas
PRINT ISSN: 2992-4383 || ONLINE ISSN: 2992-4391

Credit card fraud detection using logistic regression and isolation forest algorithms

Akinola Kayode E[1]., Aina Daniel A[2]., Oyedele Oluwasanya[3]., Braimah Joachim, A[4].
[1,2,3,4] Department of Physical and Computer Science, McPherson University,
Seriki Sotayo, Ogun State, Nigeria.
Corresponding Author's E-mail: Kayodewale87@yahoo.com

Abstract
Due to the rapid growth of e-commerce, the use of credit cards for online purchases has increased and unexpectedly caused an eruption in credit card fraud. Fraud detection systems come into a synopsis when the fraudsters break down every prevention initiative put in place. Fraud detection based on analysing existing purchase data of a cardholder is a promising way in minimizing fraud. The detection of credit card fraud features statistical tests and data made on user data based on those behavioural and historical data. This study focused on the use of Logistic Regression and Isolation Forest in detection of credit card fraudulent transactions. Dataset used in this study was obtained from Kaggle. In measuring the model performance: precision, recall, F1-score and AUC-ROC curve were used. From the study results, accuracy score for logistic regression algorithm yielded 99.91% for training data and 78% for testing data, while the precision, recall and F1-score were 0.95, 0.56 and 0.70 respectively. Furthermore, accuracy score for isolation forest algorithm yielded 99.82% for training data and 74% for testing data, while the precision, recall and F1-score were 0.49, 0.49 and 0.49 respectively. From the results obtained upon evaluating the dataset, finding revealed that logistic regression algorithm out-performed isolation forest algorithm.

Keywords: Anomaly detection, credit card, fraudulent, isolation, local outlier, machine learning.

1.0 Introduction

A credit card or universally known as a payment card is a small plastic card issued to various users as a system of payment. It is branded as one of the methods of carrying out transactions and have become commonplace for individual finance over the past few years. In our daily lives, credit cards are used for purchasing goods and services with the help of virtual card for online transaction or physical cards for offline transaction. The credit card has a plethora of advantages, one being its easy access to credit, purchase and offering a guaranteed method of payment and providing consumers with a way to further implement a cashless policy in transactions. Fraudulent transactions can be carried out by an attacker by stealing the card information from the cardholder. This information may include the credit card number, the validity, the Card Verification Value (CVV) which is vital for completing online transactions and the name of the card holder. After the information gathering, the attackers can then use these cards for ridiculous purchases, putting both the cardholder and the institution at risk. The good thing is, some major payment processes mine data from their card holders and their spending habits. The company builds a picture not only of where you spend the money but how much and how frequently. Some more advanced methods can track the IP addresses of where the transactions originated from. So, if a charge tied to an IP address previously used for fraud is observed, the card is flagged and immediately reported.

Machine Learning is one of the fastest growing areas of computer science, with far-reaching applications (Shalev-Shwartz & Ben-David, 2014) has a natural outgrowth at the intersection of Computer Science and Statistics which has evolved into a broad, highly successful, and extremely dynamic discipline. Machine Learning is broadly defined as computational methods using experience to improve performance or to make accurate predictions; experience refers to the past information available to the learner, which typically takes the form of electronic data collected and made available for analysis. Machine Learning entails data-driven methods capable of mimicking, understanding and aiding human and biological information processing tasks; and is closely related with Artificial Intelligence (AI), with machine learning placing more emphasis on using data to drive and adapt the model from large datasets. The motivation in machine learning is majorly to produce an algorithm that can either mimic or enhance human/biological performance (Sepp, 2013).

The implementation of Machine Learning in credit card fraud detection system involves a process of data investigation using data science and the development of a model that will provide the best results in revealing and preventing fraudulent transactions. This is achieved by putting together the meaningful features of card users' transactions. This information is then run through a trained model which analyzes patterns to be able to classify whether a transaction is fraudulent or legitimate. Credit card fraud detection is a very active area of research and learning in data science and many works have been done over the years in relation to this topic and its constituents. Table 1, below summarizes the consulted literatures on machine learning, method used, strength and limitations.

Table 1: Summary of related works

Author	Method Used	Strength	Limitations
Vengatesanet al. (2020)	Proposed a working model of the system, involving pre-processing techniques, logistical regression and KNN algorithm for production analytics	The KNN algorithm is produced best result such as statistical measure	Outlines the infinitesimal number of trades fraudulent in nature
Carcillo et al. (2019)	Taking the outlier scores completed on the dataset, involving a hybrid approach	The implementation and assessment of different levels of granularity for the definition of an outlier score	The use of global outlier scores indicated a strong deterioration in accuracy and inconsistencies in the behaviour of precision metrics used
Makki (2019)	Implementing Class Imbalance solutions like classification algorithms and a selection of performance measures	Their research was able to show that SVM and ANN are the best methods.	While these approaches improve sensitivity, it led to an increase in the number of false alarm rates
Jain et al. (2019)	Comparing the performance of different systems by using measures generated from the system in quantitative environments	Neural Networks and Naïve Bayes networks give the highest accuracy in comparison to others	ANNs are expensive to train and can easily be overstrained
Prakash et al. (2018)	Use of R programming language with RStudio and a GUI for confusion matrix decision tree algorithm analysis	Their results showed that decision tree had a higher accuracy than other algorithms	Discovered that the standard data mining algorithms did not fit well with classification problems
Tran et al. (2018)	They used data-driven approaches without anomalies in the training set	Their proposed approaches to a high-level of accuracy and a low false alarm rates	Improvement on the detection ability of the proposed system
Niu et al. (2019)	They evaluated five supervised and four unsupervised learning models to leverage transactions to determine abnormal transactions.	All models performed well, with XG Boost achieving the best performance	The label availability and data imbalance restrict the supervised learning performances
Varmedja (2019)	The use of SMOTE technique was used for oversampling	They proved that the usage of classical algorithms is as successful as deep learning	Stresses the need for feature selection for metrics such as accuracy and precision
Patil et al. (2018)	Proposed a robust framework to build a strong analytical model with the help of confusion matrix	They were able to obtain a higher performance with decision tree in terms of accuracy	They had the issue of overfitting of tree in memory as data increases
Rahmawati (2017)	They based the possibility of fraud on the event log by identifying symptoms of fraudulent activities	The method yielded a 94% accuracy.	The state probability of fraud has to be greater than the value of state probability of no fraud

From the table 1 above, literatures that are related to different methodologies of analysing dataset are reviewed in the summary table showing their objectives, methods, strengths and weaknesses are presented.

2.0 Material and methods

There is scarcity in the availability of credit card frauds data publicly, as this information will contain and include sensitive or confidential information. However, the dataset used for this work is obtained from Kaggle. This dataset contains transactions made by credit cards in September 2013 by European cardholders. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions. It contains encapsulated numerical input variables of features V1 through V28, the result of a PCA dimensionality reduction that was used in order to protect sensitive information. Principal Component Analysis (PCA) enables the execution of an exploratory data analysis to reveal the inner structure of data and explain its variations. The features which have not been transformed with PCA are 'Time' and 'Amount'. Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature 'Amount' is the transaction and this feature can be used for example-dependent cost-sensitive learning. The feature, 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise.

2.1 Software Used

The software used for this work is Jupyter Notebook, a web based interactive computing platform provided by Project Jupyter. The notebook combines live code, equations, narration text, visualization and offers a streamlined, document-centric experience. It uses the IPython variable in shell. IPython is an interactive shell that is built with Python. It provides a more useful shell environment to execute python code in REPL (Read Eval Print Loop). Below are some of the dependencies used during the course of this project study:

- NumPy: NumPy is a Python library used for working arrays. It provides a high-performance multidimensional array and tools to manipulate those arrays. It also has functions for working in the domain of linear algebra, Fourier transform and matrices.
- Pandas: The Pandas module is an open-source python library that provides high performance data structures and data analysis tools. It is used to process data from csv files for analysis and processing. Pandas is also capable of offering an in-memory 2d table object called Data Frame.
- Sklearn: This is the most robust library for machine learning in Python. It provides a vast selection of efficient tools for machine learning and statistical modeling. This includes classification, regression, clustering and dimensionality reduction. Note that importing sklearn functions have to be specified and issued at the beginning of the project data.
- Scipy: Scipy builds on NumPy, providing a large number of functions that operate on NumPy arrays. It is useful for different scientific, engineering and mathematical applications. It allows the user to manipulate and visualize data using a wide range of high-level commands.
- Matplotlib: This is a cross-platform library for making 2d plots from data in arrays, providing data visualization and graphical plotting library, and it's numerical extension NumPy. Jupyter Notebook is able to display plots if code in input cells and works seamlessly with matplotlib library.
- Pylab: This is a module that provides a namespace by importing functions from the modules NumPy and Matplotlib. It gets installed alongside matplotlib as a module.
- Seaborn: Seaborn aids in better understanding of the data by making statistical graphics in Python. It builds on top of matplotlib and integrates closely with pandas' data structure.

2.2 Method

This work employed pandas for reading .csv files, NumPy for working arrays, and some sklearn functions. Model_selection is a method for setting a blueprint to analyze data and using it to measure new data. This in conjunction with the train_test_split function which splits arrays or matrices into random train and test subsets, splitting the data into training and test data. The sklearn.linear_model function is a logistic regression classifier, a classification algorithm rather than a regression algorithm, used to estimate discrete value like 0 or 1, yes/no, true/false. It is also called "Logit" or "MaxEnt Classifier". The last dependency specified is a module that implements several loss, score and utility functions to measure classification performance. In this case, it deals with the accuracy classification score, which computes the subset accuracy. It returns the mean accuracy on the given test and data, and aids in checking the performance of the model.

2.2.1 Understanding True Positive, True Negative, False Positive and False Negative in a Confusion Matrix

Sklearn has two great functions as can be seen in (figure 1): confusion_matrix() and classification_report(). Sklearnconfusion_matrix() returns the values of the Confusion matrix. The output given is slightly different. It shows that it takes and accesses the rows as Actual values and the columns as Predicted values. The rest of the concept remains the same. Sklearnclassification_report() outputs precision, recall and f1-score for each target class.

True Positive (TP): Here, the predicted value matches the actual value. This means that the actual value was positive and the model predicted a positive value. Therefore, it can be said that the Observation is Positive, and the model classified it as Positive.
True Negative (TN): Here, the predicted did not value matches the actual value. This means that the actual value was negative and the model predicted a negative value.
Positive (FP): This is also known as the Type 1 error. In this scenario, the predicted value was falsely predicted because the actual value was negative but the model predicted a positive value. Therefore, it can be said that the Observation is Negative, but the model classified it as Positive.
False Negative (FN): This is also known as the Type 2 error. In this scenario, the predicted value was falsely predicted because the actual value was positive but the model predicted a negative value.
Accuracy: Although Accuracy is not recommended for imbalanced data, because the great number of correct predictions of the negative class will make the accuracy high, even if we have a lot of wrong predictions for the positive class.

$$\text{ACC} = \frac{\text{TP} + \text{TN}}{\text{TP}+\text{FP}+\text{TN}+\text{FN}} \quad (1)$$

```
Confusion matrix :
 [[2 2]
 [1 5]]
Outcome values :
 2 2 1 5
Classification report :
              precision    recall  f1-score   support

           1       0.67      0.50      0.57         4
           0       0.71      0.83      0.77         6

   micro avg       0.70      0.70      0.70        10
   macro avg       0.69      0.67      0.67        10
weighted avg       0.70      0.70      0.69        10
```

Figure 1: Confusion Matrix with the Scikit-learn library in Python

2.2.1.1 Precision vs. Recall
Precision gives a definite description of how many of the correctly predicted cases actually turned out to be positive.

To calculate Precision:

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

This would determine whether the model is reliable or not. Low precision means the more false positives are predicted bythe model. Recall describes how many of the actual positive cases that was able to be predicted correctly with the model.

To calculate Recall:

$$Recall = \frac{TP}{TP+FP} \quad (3)$$

Recall focuses on outlining the proportion of actual positive cases that are correctly identified. It can also be regarded as the ratio of True Positives to all the positives in the dataset. Low recall means the more false negatives the model predicts. Despite their seemingly clashing attributes, Precision and Recall are useful for imbalanced datasets, because they don't involve the true negatives. They are only concerned with the correct prediction of the positive class.

2.2.1.2 F1 Score

The F1 Score is used when both the scores of precisions and recall are needed for the evaluation of the model

$$F1 = \left(\frac{recall^{-1} + precision^{-1}}{2}\right)^{-1} = 2\ \frac{precision\ .recall}{precision+ recall} \quad (4)$$

It is the harmonic mean of precision and recall values for a classification problem. The F1 Score maintains a balance between the precision and recall for the classifier. If the precision is low, the F1 is low and if the recall is low again the F1 score is low.

2.3. Statistical / Data analysis

AUC-ROC curve (Area Under Curve — Receiver Operating Characteristic Curve)

AUC ROC indicates how well the probabilities from the positive classes are separated from the negative classes. AUC is scale-invariant. It measures how well predictions are ranked, rather than their absolute values. The ROC is a trade-off between the True Positive Rate (TPR) and False Positive Rate (FTR) for a predictive model using different probability thresholds. The True Positive Rate (TPR) is plot against False Positive Rate for the probabilities of the classifier predictions. The area under the curve is then calculated. The False Positive Rate is the probability of a false alarm (Figure 2).

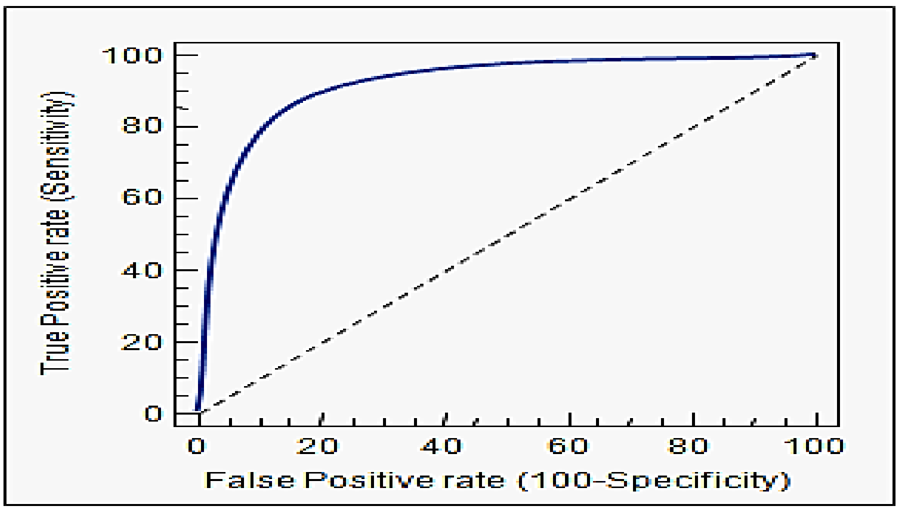

Figure 2: True Positive and False Positive relation

In figure 3, the ROC curves were used to decide on a Threshold value. The choice of threshold value will also depend on how the classifier is intended to be used.

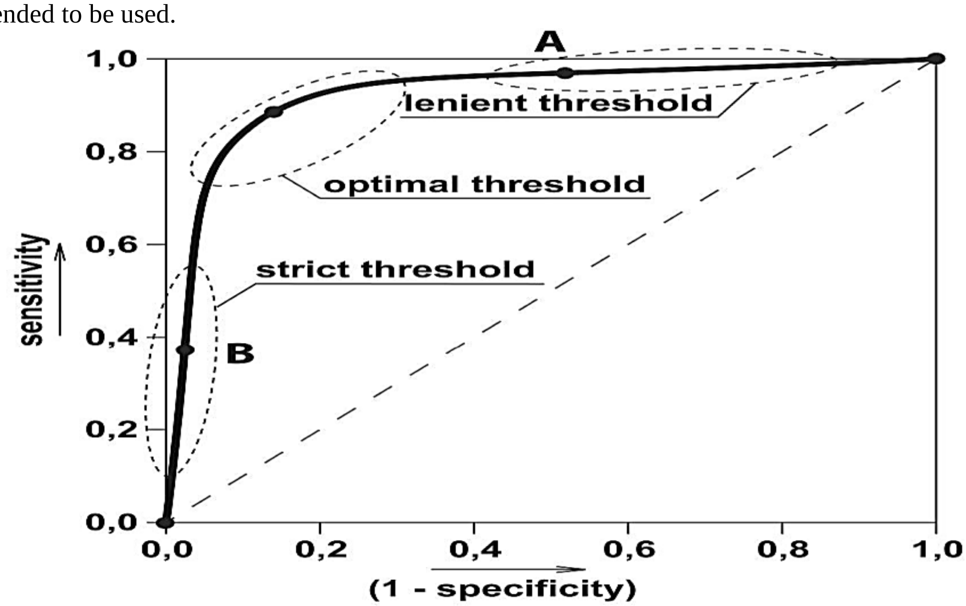

Figure 3: Threshold Specificity

2.3.1 Exploring Logistic Regression and Isolation Forest AUC

Logistic Regression is commonly used to estimate the probabilities, that an instance belong to a particular class. The class probabilities are also determined in a specific approach depending on the distance from the boundary. When dataset is bigger, it passes to ends which are (0 and 1). These probability statements do not just make logistic regression a classifier, but an efficient classifier.In this case, the model developed uses logistic regression to build the classifier to prevent frauds in credit card transactions, basically known as a binary classifier.

Logistic Regression has several hyperparameters such as; C, Solver, Penalty and Max_iter.

C: This is a control parameter that has full control of the penalty strength. The higher the value of C, the less the model is standardized.

Solver: It is of great significance to try different solvers as each solver's performance or convergence is notably different from others.

Penalty: Here, it is possible to specify regularization Techniques.

Max_iter: This is the maximum number of iterations taken.

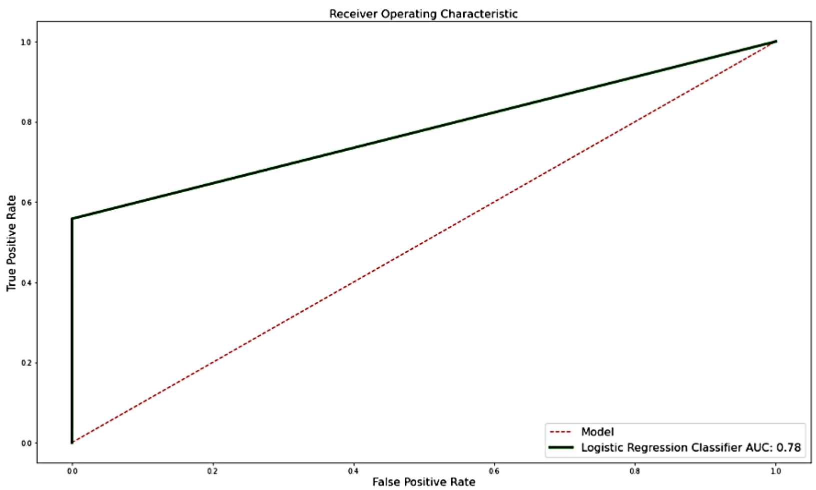

Figure 4: Output of the Logistic Regression yielding AUC accuracy of 78%

Out[49]: <matplotlib.legend.Legend at 0x1421ea70>

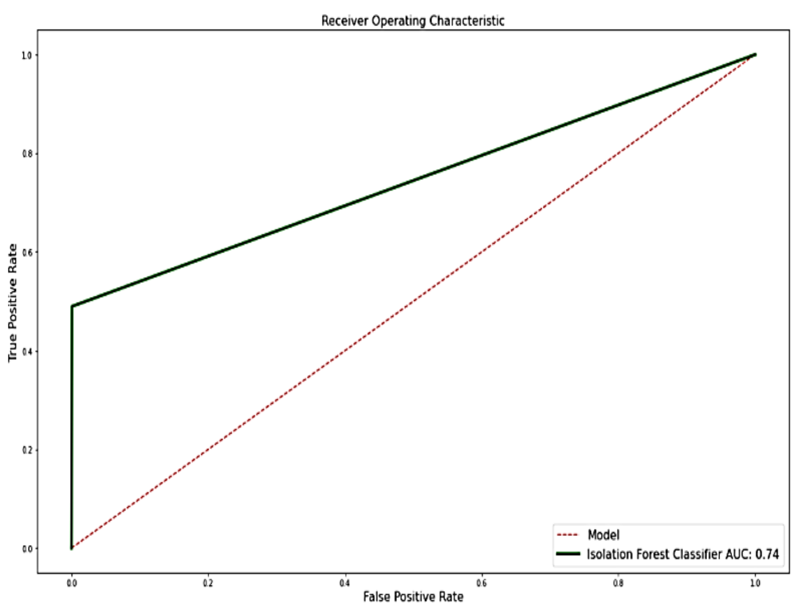

Figure 5: Output of the Isolation Forest Algorithm yielding AUC accuracy of 74%

2.3.1.1 Isolation Forest AUC

Isolation forest attempts to separate each point in the data. Here, an aberrant point could be separated in a few steps while closer normal points could take significantly more steps to be isolated. Isolation forest is a tree-base model that is developed to detect anomalies and aberrant factors. In figure 4, the AUC from the study yielded an accuracy of 78% for Logistic Regression and 74% for isolation forest in figure 5.

2.4 Comparison Results of Logistic Regression and Isolation Forest in a Distributed Data frame

Figure 6, shows that when evaluating the model using Logistic Regression, it is found that the test set has 99.91% accuracy. Despite having an accuracy of 99.91%, the model predicted 57 fraud cases incorrectly. This is known as Accuracy Fallacy.

```
Training Accuracy:  0.999133915404602
Testing Accuracy:  0.9991432824920649
[[71069     4]
 [   57    72]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     71073
           1       0.95      0.56      0.70       129

    accuracy                           1.00     71202
   macro avg       0.97      0.78      0.85     71202
weighted avg       1.00      1.00      1.00     71202
```

Figure 6, Training and Test accuracy model for a Logistic Regression data frame yields 99.91% .

Evaluating the Isolation Forest;

```
Training Accuracy:  0.9982374028728227
[[284064    251]
 [   251    241]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00    284315
           1       0.49      0.49      0.49       492

    accuracy                           1.00    284807
   macro avg       0.74      0.74      0.74    284807
weighted avg       1.00      1.00      1.00    284807
```

Figure 7, Training and Test accuracy model for a Isolation Forest data frame yields 99.82%

Figure 7, shows that when evaluating the model using Isolation Forest, it is found that the test set has 99.82% accuracy. Despite having an accuracy of 99.82%, the model predicted 251 fraud cases incorrectly. This is known as Accuracy Fallacy.

3.0 Results and Discussions

From table 2 and 3, the precision = 0.95, recall = 0.56 and F1-score = 0.70 for logistic regression were better than Isolation Forest with the precision, recall and F1-score of 0.49, 0.49 and 0.49 respectively. The work has established effectiveness and efficiency of logistic regression over isolation forest algorithm in machine learning approach to analyzing any dataset. These results are acceptable because they were generated from widely acceptable dataset and processed with standard programming software. Accessibility to financial institutions database to capture dataset proved to be a major thorn in this work; because financial institutions were not willing to expose their customers data to a third part because of the risk that is involved.

Table 2: Values from Logistic Regression Calculation

	Precision	Recall	F1 – score	Support
0	1.00	1.00	1.00	71073
1	0.95	0.56	0.70	129

Table 3: Values from Isolation Forest Calculation

	Precision	Recall	F1 – score	Support
0	1.00	1.00	1.00	284315
1	0.49	0.49	0.49	492

4.0. Conclusion

In this study, an analysis of credit card fraud identification was carried out on a publicly available dataset; utilizing machine learning approaches such as logistic regression and isolation forest model. PYTHON programming language was employed. When analyzing the dataset, the results of this study showed that logistic regression algorithm had higher accuracy when compared with isolation forest algorithm.

5.0 Recommendation

Financial Institutions should adopt more advanced security measures in protecting their customer's credit cards from fraudulent; because the hackers keep developing different techniques to break security measure put in place by financial institutions.

References

Bhusari, V., &Patil, S. 2016. Study of hidden markov model in credit card fraudulent detection. In *2016 World Conference on Futuristic Trends in Research and Innovation for Social Welfare (Startup Conclave)* (1-4). IEEE.

Carcillo, F., Le Borgne, Y. A., Caelen, O., Kessaci, Y., Oblé, F., &Bontempi, G. 2021. Combining unsupervised and supervised learning in credit card fraud detection. Information sciences, 557, 317-331.

Jain, Y., Tiwari, N., Dubey, S., & Jain, S. 2019. A comparative analysis of various credit card fraud detection techniques. Int J Recent TechnolEng, **7**(52), 402-407.

Makki, S. 2019. An efficient classification model for analyzing skewed data to detect frauds in the financial sector (Doctoral dissertation, Université de Lyon; Universitélibanaise).

Maniraj, S. P., Saini, A., Ahmed, S., & Sarkar, S. 2019. Credit card fraud detection using machine learning and data science. International Journal of Engineering Research and, **8**(09).

Niu, X., Wang, L., & Yang, X. 2019. A comparison study of credit card fraud detection: Supervised versus unsupervised. arXiv preprint arXiv:1904.10604.

Patil, S., Nemade, V., &Soni, P. K. 2018. Predictive modelling for credit card fraud detection using data analytics. Procedia computer science, 132, 385-395.

Prakash, B., Murthy, G. V. M., Ashok, P., Prithvi, B. P., & Kira, S. S. H. (2018). ATM Card Fraud Detection System Using Machine Learning Techniques. Int. J. Res. Appl. Sci. Eng. Technol, **6**(4), 5124-5129.

Rahmawati, D., Sarno, R., Fatichah, C., &Sunaryono, D. 2017. Fraud detection on event log of bank financial credit business process using Hidden Markov Model algorithm. In 2017 3rd International Conference on Science in Information Technology (ICSITech) (35-40). IEEE.

Sepp, H. 2013. *Theoretical bioinformatics and machine learning* (2nd ed.). Wellington, New Zealand.

Shalev-Shwartz, S., & Ben-David, S. 2014. *Understanding machine learning: From theory to algorithms* (1st ed.). Cambridge, USA, ISBN 978-1-107-05713-5.

Tran, P. H., Tran, K. P., Huong, T. T., Heuchenne, C., HienTran, P., & Le, T. M. H. (2018, February). Real time data-driven approaches for credit card fraud detection. In Proceedings of the 2018 international conference on e-business and applications (6-9).

Varmedja, D., Karanovic, M., Sladojevic, S., Arsenovic, M., &Anderla, A. 2019. Credit card fraud detection-machine learning methods. In 2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH) (1-5). IEEE.

Vengatesan, K., Kumar, A., Yuvraj, S., Kumar, V., &Sabnis, S. 2020. Credit card fraud detection using data analytic techniques. Advances in Mathematics: Scientific Journal, **9**(3), 1185-1196.