International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 07 | July - 2022 Impact Factor: 7.185 ISSN: 2582-3930

CREDIT CARD FAULT DETECTION USING ISOLATION FOREST AND LOCAL OUTLIER FACTOR METHOD

Alisha Gaikwad

Research Scholar

Dr. Rohit Miri

Head of department

Department of Computer Science, Dr. C.V.Raman University Kota, Bilaspur Chhattisgarh

Abstract:

The fast rise of the e-commerce industry has resulted in an exponential increase in the usage of credit cards for online purchases, resulting in an increase in fraud. In recent years, identifying fraud in the credit card system has grown extremely challenging for banks. In order to detect credit card fraud in transactions, machine learning is essential. Banks utilise a variety of machine learning approaches to forecast these transactions, as well as historical data and new variables to improve the prediction capability. The sampling strategy on the data-set, the selection of variables, and the detection algorithms utilised all have a significant impact on the performance of fraud detection The efficacy of logistic regression, decision trees, and random forests for detecting credit card fraud is investigated in this research. Kaggle provided a credit card transaction data collection with a total of 2,84,808 credit card transactions from a European bank data source. It divides transactions into two categories: "positive class" and "negative class." The data set is substantially skewed, with around 0.172 percent of transactions being fraudulent and the remainder being legitimate. We used oversampling to balance the data set in this article, which resulted in 60% fraudulent transactions and 40% legitimate transactions. The dataset is subjected to the three approaches, and the work is carried out in R. The effectiveness of the strategies is assessed depending on a variety of factors sensitivity, specificity, accuracy and error rate. Isolation Forest and Local Outlier Factor have accuracy values of 99.7 and 99.6, respectively. The Random forest outperforms the logistic regression and decision tree procedures, according to the results.

Keywords: Fraud detection, Credit card, Logistic regression, Decision tree, Random forest.

I. INTRODUCTION

A credit card is a thin, convenient plastic card that carries identity information, such as a signature or a photograph, and permits the person listed on it to charge goods or services to his account, for which he will be invoiced on a regular basis. Today, automated teller machines (ATMs), retail readers, banks, and online internet banking systems all read the information on the card. They have a one-of-a-kind card number, which is crucial. Its safety is dependent on both the physical security of the plastic card and the confidentiality of the credit card number. The quantity of credit card transactions is rapidly increasing, which has resulted in a significant increase in fraudulent activity. Theft and fraud performed with a credit card as a fraudulent source of funds in a transaction are referred to as credit card fraud. Statistical approaches and a variety of data mining algorithms are commonly utilised to tackle the fraud detection challenge. Artificial intelligence, Meta learning, and pattern matching are used in the majority of credit card fraud detection systems. Genetic algorithms are evolutionary algorithms that try to find the best methods for detecting and preventing fraud. The development of an efficient and secure electronic payment system to detect whether a transaction is fraudulent or not is given a high priority. In this paper, we'll look at credit card fraud and how to identify it. A credit card fraud happens when someone uses another person's card for their own personal usage without the owner's knowledge. When fraudsters utilise it in such instances,

it is used until its whole usable limit is spent. As a result, we want a solution that reduces the overall allowed limit on the credit card, which is more vulnerable to fraud. Furthermore, as time passes, a Genetic algorithm creates better answers. The development of an efficient and secure electronic payment system for identifying fraud is given top priority.

The methods for detecting fraud may be classified into two categories: Unsupervised procedures, in which there are no prior sets in which the status of the transactions is known to be fraud or legitimate, and supervised techniques, in which past known legitimate/fraud instances are utilised to develop a model that will provide a suspicion score for the current transactions. [2] has a brief overview of supervised and unsupervised approaches. Many of these methods, such as artificial neural networks, may be employed in both supervised and unsupervised settings.

Rule-induction approaches, decision trees, neural networks, Support Vector Machines (SVM), logistic regression, and meta-heuristics such as genetic algorithm, k-means clustering, and closest neighbour algorithms are the most often used fraud detection methods. To develop classifiers, these strategies can be employed alone or in conjunction with ensemble or meta-learning techniques. ID3, C4.5, and C&RT, which are well-known decision tree approaches, are applied for credit card fraud detection in [11- 14]. SVM is also used to detect credit card fraud in [10, 15].

Fraud detection systems assess transactions and generate a suspicion score (usually a probability between 0 and 1) that indicates the likelihood of a fraudulent transaction. The strategies utilised to generate the model/models in fraud detection systems are relevant to the computational operations of these scores. These scores are used with a predetermined threshold value to distinguish between fraudulent and genuine transactions. However, these ratings are rarely utilised directly; rather, they assist observer employees with topic knowledge in examining and attempting to discover forgeries. Because firms have a limited number of employees to handle this procedure, the ability of detection systems to provide correct suspicion ratings aids these employees in a variety of ways. Nonetheless, the detection systems' effectiveness is based on their ability to discern between fraudulent and lawful transactions by giving high-precision suspicion ratings.

A credit card fraud detection system is built in this work using a number of Isolation Forest and Local Outlier Factor approaches. Using appropriate descriptions, each account is watched independently, and transactions are attempted to be detected and labelled as authentic or normal. The classification will be based on the suspicion score generated by the established classifier models. When a new transaction occurs, the classifier can determine whether it is legitimate or fraudulent. To our knowledge, there has been no prior research comparing the performance of ISOLATION Forest and Local Outlier Factor based classifiers in the field of credit card fraud detection using real data.

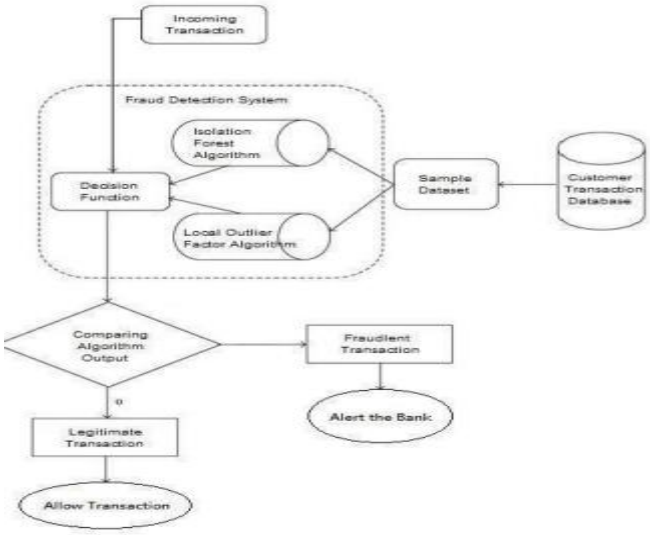

Fig1 : Block Diagram for Credit card fraud detection

II. RELATED WORKS

In the last decade, fraud detection has gotten a lot of attention. The application of hybrid machine learning techniques in the credit card fraud area is discussed in this section. A growing corpus of research has offered methods for improving fraud detection.

The authors of [3] studied a mix of diverse techniques as they presented a novel voting mechanism dubbed

OPWEM, which stands for optimistic, pessimistic, and weighted voting in an ensemble of models that can function in conjunction with rule-based systems. The authors' use of OPWEM is totally warranted, since they advise that bank management should pick one of the voting strategies based on the bank's false alarm rate strategy. If a bank wants to find as many fraud instances as possible, for example, pessimistic voting (PES) should be used. A bank that seeks for a low false alarm rate, on the other hand, should utilize the optimistic voting (OPT) technique. Furthermore, with a negligible false alarm rate, weighted voting (WGT) found more frauds than OPT. As a result, it might be a viable alternative to OPT and PES. In addition, the author [17] suggested a hybrid framework model based on a mix of unsupervised and supervised learning methods. The author's goal was to identify fraudulent transactions at a cheap cost, which included the time and effort invested by bankers to get the requisite skills in machine learning classification algorithms. The author used a simple one-class classification technique, with the addition that the data description boundary is changed dependent on account holders' purchasing behavior. A post-processing step was introduced to improve the model's output by passing the flagged accounts via rule-based filters. The one-class classification approach, according to the author, is well suited to complicated and large-scale datasets of transaction data since it aids in the development of an account group structure that gives individualized models for various types of cardholder behavior. The author claims that the utilized approach, when paired with the rule-based filters' post-processing level, produces the greatest results. The fundamental weakness of this work is that the experimental data show that the hybrid approach detects the majority of fraudulent cases that the bank's rule-based system misses, and vice versa. As a result, both procedures should be employed concurrently to gain the optimum results.

The authors also introduced a hybrid model for boosting fraud detection accuracy by merging supervised and unsupervised approaches in [22]. They presented a number of criteria for computing outlier ratings at various granularities (from high granular card-specific outlier scores to low granular global outlier scores). Then, once included as characteristics in a supervised learning technique, they assessed their additional value in terms of accuracy. Unfortunately, the results are inconclusive in terms of local and global techniques. However, in terms of Area Under the Precision–Recall Curve, the model produces a more significant result (AUC-PR).

In [23], the authors used Bayesian Classification and Association Rule Learning (ARL) to examine and find the true transaction signals of fraudulent accounts, as well as to give a fraud prevention reference for the banking industry. A fake account detection system was created based on these signals, and the signs were studied further using real-time daily transaction data. They found that the study's proposed solution is successful and efficient, and that financial institutions may utilise it to reduce the need for manual fraud account screening. Similarly, the authors suggested in [20] an intelligent model for credit card fraud detection that can detect fraud in anonymous and strongly skewed credit card datasets.

Similarly, the authors in [21] also presented a hybrid model that combines ARL and process-mining by conducting a process-mining inquiry to collect a number of fraud variables to create some association rules for fraud detection. The aim of process-mining in this context is to inspect skipped tasks, resources, throughput time, and decision points based on simple rules in the Standard operating procedure (SOP). In the first phase, they used a process-mining technique to extract the variables of fraudulent cases from the dataset. Then, an expert determines whether a case contains fraud variables. In the second phase, an Apriori algorithm is used to produce either fraud cases or legal cases. Eventually, as the detection rules, only the association rules with specific consequences such as expert judgement regarding fraudulent status are selected. The authors separated each customer's transactions into fraudulent and valid transactions, then used the Apriori algorithm to find fraudulent and legitimate transaction patterns in both groups. As a result, they proposed a matching algorithm that searches pattern databases for a match with the incoming transaction to identify fraud. Another key

factor to keep in mind is that, in order to maintain the anonymity of the data, each characteristic was evaluated equally while looking for patterns, with no priority given to any attribute. Finally, as a result of consumer fraudulent conduct shifting somewhat over time, the authors advised running the proposed model at specific time points to improve the legal and fraud pattern information.

Furthermore, the performance of the utilised classifiers was investigated using a genuine credit card dataset received from a financial institution and a twelve-machine learning algorithm in conjunction with the AdaBoost and majority voting techniques [16]. The highest Matthews correlation coefficient (MCC) score they received was 0.823, which received a majority of the votes. When AdaBoost and majority voting techniques were used, however, a perfect MCC score of 1 was produced. Noise ranging from 10% to 30% was introduced to the data samples to better analyse the hybrid models. The majority voting process yielded the best MCC score of 0.942 when 30 percent noise was added to the data set. As a result, the authors concluded that the majority vote technique works well in the presence of noise.

A more recent study used credit card datasets and machine learning classifiers including LR, Gradient Boosting (GB), RF, and voting classifiers to construct a hybrid model to identify credit card fraud [24]. The author discovered that RF and GB had 99.99 percent detection rates. Despite the fact that all of the research above were focused with fraud detection, different techniques were applied based on the dataset's nature. Various techniques, such as a single machine learning algorithm or hybrid models, have been employed to detect fraudulent transactions in the financial industry, particularly the credit card domain, as evidenced by prior attempts. However, these hybrid models merely used one model without taking into account the performance of other models to ensure that the chosen dataset is correct. As a result, the suggested approach may accidentally produce erroneous results and a lack of generality. To understand the relative effectiveness of the proposed method, a comparison of different hybrid models utilizing the same datasets is still required. This paper's main contribution is to build and analyses the usage of numerous hybrid models for the same dataset, and to select a champion hybrid model based on performance prediction evaluation.

III. METHODOLOGY

The method proposed in this study employs the most up-to-date machine learning methods to detect aberrant activity known as outliers. The following is a representation of the fundamental rough architectural diagram:

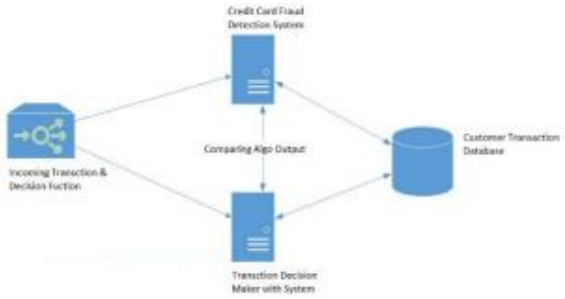

Fig 2: architecture diagram

First and foremost, we got our data from Kaggle, a data analysis service that offers datasets. There are 31 columns in this dataset, with 28 of them labelled v1-v28 to preserve sensitive information. Time, Amount, and Class are represented by the other columns. The time difference between the first and subsequent transactions is shown in this graph. The amount of money exchanged is referred to as the amount. A genuine transaction is represented by class 0, while a fraudulent transaction is represented by class 1. We use several graphs to visually understand the dataset and check for errors.

The data has been prepared and is being processed. To guarantee that the evaluation is fair, the time and money columns are standardized, and the Class column is deleted. A series of algorithms is used to process the data. The module diagram below depicts how these algorithms interact: The following outlier identification modules are applied to this data once it has been fitted into a model:

- Local Outlier Factor
- Isolation Forest Algorithm

Sklearn contains these algorithms. Ensemble-based algorithms and tools for classification, regression, and outlier identification are included in the sklearn package's ensemble module. This free and open-source Python library is made up of NumPy, SciPy, and Matplotlib modules, and it includes a number of basic and efficient data analysis and machine learning tools. It includes a number of classification, grouping, and regression methods, as well as the ability to work with numerical and scientific libraries. We created a Python programmed using the Jupiter Notebook platform to showcase the method proposed in this research. The Google Collab platform, which supports all python notebook files, may also be used to run this application in the cloud. Modules are described in detail, including pseudocodes for each one.

A. Local Outlier Factor (LOF) algorithm

Is an unsupervised anomaly detection approach that computes a data point's local density deviation in relation to its neighbors. It considers samples with a significantly lower density than their neighbors to be outliers. This example demonstrates how to utilize LOF for outlier identification, which is the estimator's default use case in scikit-learn.

Formulation:

Let k-distance(A) be the distance of the object A to the k-th nearest neighbor. Note that the set of the k nearest neighbors includes all objects at this distance, which can in the case of a "tie" be more than k objects. We denote the set of k nearest neighbors as $N_k(A)$.

Illustration of the reachability distance. Objects B and C have the same reachability distance (k=3), while D is not a k nearest neighbor

This distance is used to define what is called *reachability distance*:

$$\text{reachability-distance}(A,B)=\max\{k\text{-distance}(B), d(A,B)\}$$

In other words, the reachability distance between two items A and B is the real distance between them, but at least the k-distance between them. Objects belonging to B's k closest neighbours (the "core" of B, see DBSCAN cluster analysis) are regarded as equally distant. The objective for this separation is to get more consistent outcomes. Because it is not symmetric, this is not a distance in the mathematical sense. While it is a typical error to apply k-distance(A) every time, this results in a slightly different approach known as Simplified-LOF.

The density of local reachability of an item A is defined by

$$\text{lrd}_k(A):=1/(\Sigma_{B \in N_k(A)}\text{reachability-distance}_k(A, B)/|N_k(A)|)$$

which is the inverse of the average reachability distance of the object *A from* its neighbors. Note that it is not the average reachability of the neighbors from *A* (which by definition would be the *k*-distance(A)), but the distance at which *A* can be "reached" *from* its neighbors. With duplicate points, this value can become infinite.

The local reachability densities are then compared with those of the neighbors using

$$\text{LOF}_k(A):=\Sigma_{B \in N_k(A)}\text{lrd}_k(B)/|\text{lrd}_k(A)/|N_k(A)| = \Sigma_{B \in N_k(A)}\text{lrd}_k(B)/|N_k(A)| \cdot \text{lrd}_k(A)$$

This is the object's own local reachability density divided by the average local reachability density of its neighbors. A number of about 1 denotes that the thing is comparable to its surroundings (and thus not an outlier). A number less than 1 denotes a denser region (an inlier), but values considerably more than 1 denote outliers.

LOF(k) ~ 1 means Similar density as neighbors,

LOF(k) < 1 means Higher density than neighbors (Inlier),

LOF(k) > 1 means Lower density than neighbors (Outlier).

B. Isolation Forest:

Isolation forest is a method for detecting anomalies. Rather than modelling the normal points, it discovers anomalies using isolation (how remote a data point is from the rest of the data). Fei Tony Liu created it in 2007 as one of his original concepts for his PhD dissertation. The relevance of this study stems from its departure from

the dominant theory that underpins most current anomaly detectors at the time, in which all normal instances are profiled before anomalies are detected as examples that do not correspond to the usual distribution. Isolation forest provides a novel approach that uses binary trees to specifically separate anomalies, providing a new option for a speedier anomaly detector that directly targets abnormalities without profiling all the data. The approach has a linear time complexity, a low constant, and a small memory footprint, making it suitable for large data sets.

IV. RESULT:

The code reports the number of false positives it found and compares it to the real numbers. This is used to compute the algorithm's accuracy score and precision. The percentage of data we used for speedier testing was 10% of the whole dataset. At the end, the entire dataset is utilized, and both results are reported. These findings, as well as the classification report for each algorithm, are included in the output, where class 0 indicates that the transaction was considered to be genuine and class 1 indicates that the transaction was determined to be fraudulent. To rule out false positives, this result was compared to the class values.

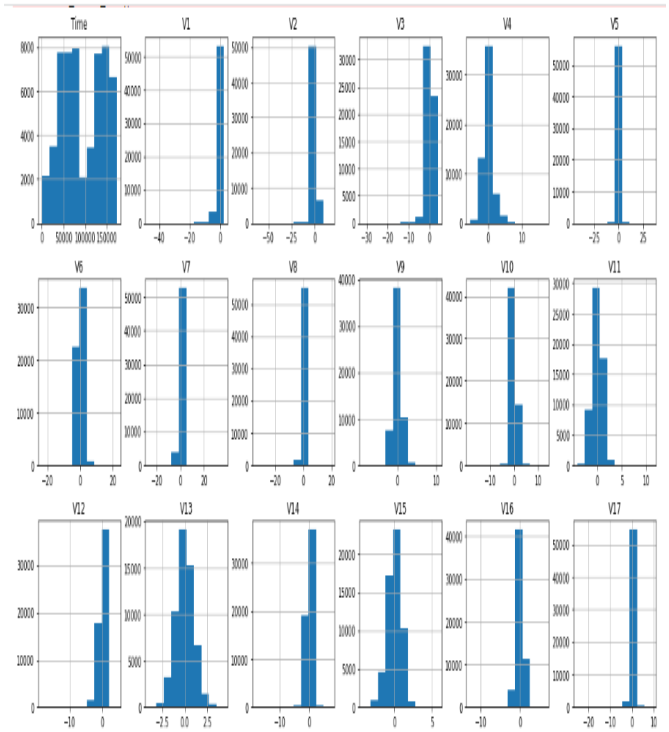

Fig3: Histogram View of each parameter.

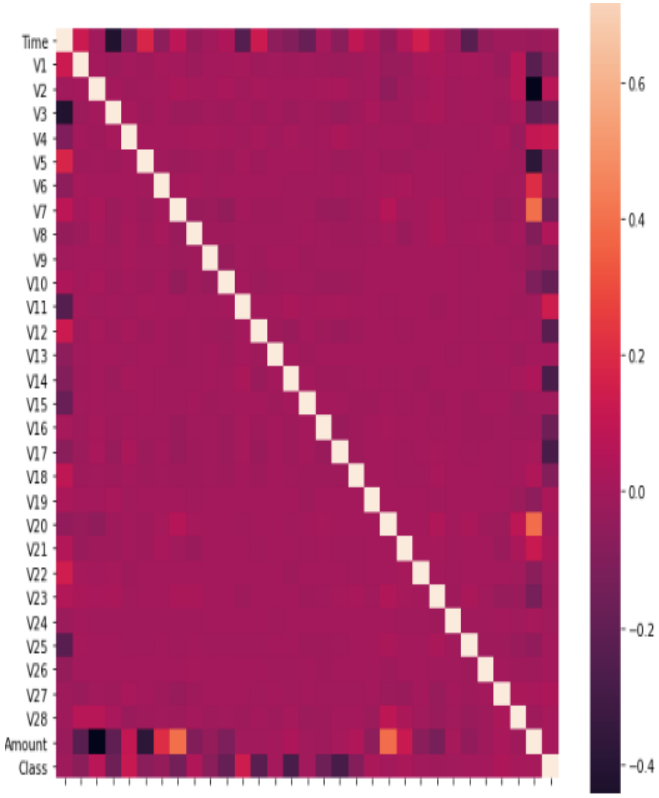

Fig 4 : Correlation Matrix

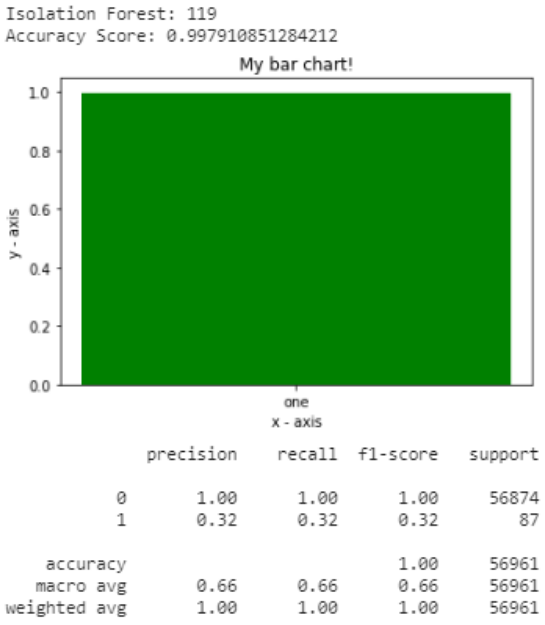

Fig 5: Accuracy Score for Isolation Forest algorithm.

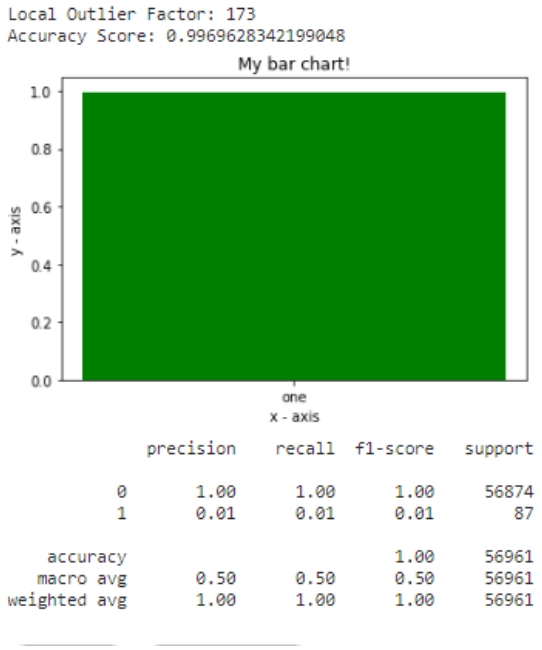

Fig 6: Accuracy Score for Local Outlier Factor.

V. CONCLUSION

Credit card fraud is without a doubt an act of Dishonesty on a criminal level. This article evaluated current results in this subject and outlined the most prevalent types of fraud, as well as how to identify them. This paper also includes a detailed description of how machine learning may be used to improve fraud detection findings, as well as the method, pseudocode, explanation, and experimentation results. While the method achieves a precision of over 99.6%, when only a tenth of the data set is considered, it only achieves a precision of 28%. When the complete dataset is given into the system, however, the accuracy increases to 33%. Due to the large disparity between the number of legitimate and authentic transactions, this high percentage of accuracy is to be expected. Because the entire dataset is made up of

REFERENCES

1. PWC. Fighting Fraud: A Never-Ending Battle; PWC: London, UK, 2020.

2. Garner, B.A. Black's Law Dictionary, (Black's Law Dictionary (Standard Edition)), 8th ed.; Thomson West: Toronto, ON, Canada, 2004; p. 1805.

3. Kültür, Y.; Ça ˘glayan, M.U. Hybrid approaches for detecting credit card fraud. Expert Syst. 2017, 34, 1–13. [CrossRef]

4. Kurshan, E.; Shen, H. Graph Computing for Financial Crime and Fraud Detection: Trends, Challenges and Outlook. Int. J. Semant. Comput. 2020, 14, 565–589. [CrossRef]

5. West, J.; Bhattacharya, M. Intelligent Financial Fraud Detection: A Comprehensive Review. Comput. Secur. 2015, 57, 47–66. [CrossRef]

6. Ethem, A. Introduction to Machine Learning, 2nd ed.; The MIT Press: Cambridge, MA, USA, 2014.

7. Mater, A.C.; Coote, M.L. Deep Learning in Chemistry. J. Chem. Inf. Model. 2019, 59, 2545–2559. [CrossRef]

8. Hossain, M.A.; Islam, S.M.S.; Quinn, J.M.W.; Huq, F.; Moni, M.A. Machine learning and bioinformatics models to identify gene expression patterns of ovarian cancer associated with disease progression and mortality. J. Biomed. Inform. 2019, 100, 103313. [CrossRef]

9. Abdelrahman, O.; Keikhosrokiani, P. Assembly Line Anomaly Detection and Root Cause Analysis Using Machine Learning. IEEE Access 2020, 8, 189661–189672. [CrossRef]

10. Khan, M.A.; Ashraf, I.; Alhaisoni, M.; Damaševi ˇcius, R.; Scherer, R.; Rehman, A.; Bukhari, S.A.C. Multimodal brain tumor classification using deep learning and robust feature selection: A machine learning application for radiologists. Diagnostics 2020, 10, 1–19. [CrossRef]

11. Cruz, J.A.; Wishart, D.S. Applications of machine learning in cancer prediction and prognosis. Cancer Inform. 2006, 2, 59–77. [CrossRef]

12. Lalmuanawma, S.; Hussain, J.; Chhakchhuak, L. Applications of machine learning and artificial intelligence for COVID-19 (SARS-CoV-2) pandemic: A review. Chaos Solitons Fractals 2020, 139, 110059. [CrossRef]

13. Angermueller, C.; Pärnamaa, T.; Parts, L.; Stegle, O. Deep learning for computational biology. Mol. Syst. Biol. 2016, 12, 878. [CrossRef]

14. Taha, A.A.; Malebary, S.J. An Intelligent Approach to Credit Card Fraud Detection Using an Optimized Light Gradient Boosting Machine. IEEE Access 2020, 8, 25579–25587. [CrossRef]

15. Khandani, A.E.; Kim, A.J.; Lo, A.W. Consumer credit-risk models via machine-learning algorithms. J. Bank. Financ. 2010, 34, 2767–2787. [CrossRef]

16. Randhawa, K.; Loo, C.K.; Seera, M.; Lim, C.P.; Nandi, A.K. Credit Card Fraud Detection Using AdaBoost and Majority Voting. IEEE Access 2018, 6, 14277–14284. [CrossRef]

17. Krivko, M. A hybrid model for plastic card fraud detection systems. Expert Syst. Appl. 2010, 37, 6070–6076. [CrossRef]

18. Alharbi, A.; Alshammari, M.; Okon, O.D.; Alabrah, A.; Rauf, H.T.; Alyami, H.; Meraj, T. A Novel text2IMG Mechanism of Credit Card Fraud Detection: A Deep Learning Approach. Electronics 2022, 11, 756. [CrossRef]

19. Behera, T.K.; Panigrahi, S. Credit Card Fraud Detection: A Hybrid Approach Using Fuzzy Clustering & Neural Network. In Proceedings of the 2015 2nd IEEE International Conference on Advances in Computing and Communication Engineering, Dehradun, India, 1–2 May 2015; pp. 494–499. [CrossRef]

20. Seeja, K.R.; Zareapoor, M. FraudMiner: A novel credit card fraud detection model based on frequent itemset mining. Sci. World J. 2014, 2014, 252797. [CrossRef]

21. Sarno, R.; Dewandono, R.D.; Ahmad, T.; Naufal, M.F. Hybrid Association Rule Learning and Process Mining for Fraud Detection. IAENG Int. J. Comput. Sci. 2015, 42, 59–72.

22. Carcillo, F.; Le Borgne, Y.A.; Caelen, O.; Kessaci, Y.; Oblé, F.; Bontempi, G. Combining unsupervised and supervised learning in credit card fraud detection. Inf. Sci. 2019, 557, 317–331. [CrossRef]

23. Li, S.H.; Yen, D.C.; Lu, W.H.; Wang, C. Identifying the signs of fraudulent accounts using data mining techniques. Comput. Hum. Behav. 2012, 28, 1002–1013. [CrossRef]

24. Sivanantham, S.; Dhinagar, S.R.; Kawin, P.A.; Amarnath, J. Hybrid Approach Using Machine Learning Techniques in Credit Card Fraud Detection. In Advances in Smart System Technologies; Springer: Singapore, 2021.

25. IEEE Computational Intelligence Society. IEEE-CIS Fraud Detection Can You Detect Fraud from Customer Transactions? 2019. Available online: https://www.kaggle.com/c/ieee-fraud-detection/overview (accessed on 5 December 2021).

26. Aoife, D.; Brian, M.; John, D.K. Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies; The MIT Press: Cambridge, MA, USA, 2015.