Real-Time Credit Card Fraud Detection Using Spark Framework

Chapter · March 2021

DOI: 10.1007/978-981-33-4046-6_28

2 authors, including:

Madhavi Alli
VNR Vignana Jyothi Institute of Engineering & Technology
9 PUBLICATIONS **12** CITATIONS

Real-Time Credit Card Fraud Detection Using Spark Framework

A. Madhavi and T. Sivaramireddy

1 Introduction

Credit card fraud transaction is an immensely popular and common issue these days worldwide. All credit-card-based companies are spending a huge amount on overcoming such fraud transactions every year but still as increasing technological advances new fraud ways are also increasing exponentially. Capturing cardholder historical behavior patterns helps derive new rules to identify a transaction as fraud or non-fraud [5]. Those behavioral patterns can be extracted in the model while implementing the algorithm. In this study, we primarily targeted on improving model performance and real-time processing of transactions instead of batch processing. The possible ways to improve model performance by feeding more pre-processed training data to an algorithm and enhance the algorithm logic to be more robust [6]. As transaction predictions should be handled in millions, we need to use distributed frameworks to drive such large-scale transactions and identify a transaction is genuine or fraud. We use Spark as a distributed framework that is similar to the Hadoop environment but can be 100 times faster compared to MapReduce processing. Though Hadoop and Spark were developed mainly to process large amounts of data we have a small difference in their selection. Hadoop is used mostly with ETL processing of batch data whereas Spark is used for ETL processing of real-time data with the tremendous performance [7]. We have used Scala programming in a Spark as it is the indigenous language of Spark and proved swift in performance over Python or java.

A. Madhavi (✉) · T. Sivaramireddy
Department of Computer Science, VNR VJIET, Hyderabad, India
e-mail: madhavi_a@vnrvjiet.in

T. Sivaramireddy
e-mail: sivaramireddy.tiyyagura@gmail.com

C. K. Mai et al. (eds.), *Machine Learning Technologies and Applications*,
Algorithms for Intelligent Systems,
https://doi.org/10.1007/978-981-33-4046-6_28

In this work, the data utilized during the experiment are indiscriminately formatted under a normal distribution. As part of pre-processing data, we scale parameters to give a good understanding of the machine learning algorithm and extract two additional features computed from five existing features which help in turning training data to random forest algorithms to give the best accuracy [8]. We used Spark streaming jobs to process collected facts in real-time and Kafka messaging systems to process data with no time these two are the best combination to handle information provided in real-time with distributed systems. Once results obtained from Spark streaming with a random forest model will display fraud transactions in fraud alert dashboard.

2 Related Work

In the existing systems, research is done on a theme study involving credit card fraud screening whereby data is balanced with sampling techniques and applied supervised machine learning algorithms [9]. The substantial problem with fraud detection is finding the best dataset to process, i.e., real-time data is practically not available due to the point of confidentiality. These concerns were not an obstacle to analysts as they were implementing their work in coordination with corporate partners and some were advised to apply a synthetic dataset of transactions [10]. Besides forthcoming genetic algorithms can be used for data generation and accomplishing the uniform data from cumbersome dataset tasks [11] As credit card safeguard technologies getting upsurge proportionally the new techniques were used by fraudsters. To overcome this situation, the data mining tool's performance is less than ideal so implementing fraud detection needs the finest enhancements like adopting big data machine learning approaches [12]. The supervised and unsupervised machine learning models are giving improved performance, in light of this we are using both the algorithms in our system, i.e., K-means clustering algorithm for data balancing and random forest classification algorithm for decision-making whether a transaction is genuine or fraud [13]. In the existing system, k-means algorithm is implemented for clustering accounts based on customer spending habits and then the XGBoost classification algorithm is applied for decision-making [14]. Most research happens on tuning algorithms and trying to improve model performance and reached an efficiency of up to 85%. We can improve the efficiency of any machine learning model with three possible solutions, they are data, algorithm, and cost-sensitive.

2.1 Data Level

The given input data is generated from the existing Sparkov data generator tool [15], by providing customer names it will generate customer data. Referring to customer data the mapped transaction data generated provides transaction category

and merchant names to the tool. The generated data for training machine learning may not have both fraud and non-fraud transactions in an equal ratio [16]. We must balance both class data and this can be achieved in multiple ways, i.e., sampling techniques. The popularly known sampling techniques are undersampling and oversampling. The undersampling method takes random samples from a majority class set that approximately match to the number of samples in the minority class set to get a balanced dataset. But we must discard large majority class samples which leads to loss of information and may reduce model potential related to the majority class sample. Conversely, the oversampling technique will synthesize the minority class set and produce samples approximately equal to the majority class which leads to more noise in newly created minority class data. Part of the work we choose undersampling not just with a random selection of samples but with the k-means algorithm to prepare balanced both class data. This data sampling can be implemented as hybrid sampling which is a combination of both undersampling and oversampling techniques [17].

2.2 *Algorithm Level*

As fraud detection problem solution classifies a transaction as a fraud or non-fraud, a classification algorithm is the best choice to use. In a classification problem, some of the known existing algorithms are implemented with naïve Bayes, k-NN and decision tree models [18]. We are concentrating on a random forest algorithm with the implementation of k-fold cross-validation to validate data with multiple trees [19].

2.3 *Cost-Sensitive*

Cost-sensitive is part of machine learning that often comes to the picture mainly for imbalanced data classification problem because the wrong prediction of positive or minority class case is a blunder than identifying wrongly for negative or majority class. This can be achieved with imbalanced dataset resampling, algorithm-level modifications, and using ensemble learning methods. Here cost means penalty awarded for the wrong prediction of positive or minority class sample [15]. This aims to minimize the cost of a model on the training dataset whether it is having imbalanced data, it can be achieved by checking the cost of tests, data instability, misclassification errors, etc. The most common and basic metric used to find the cost for an imbalanced dataset is the confusion matrix.

3 Proposed System

The current system focuses on all three solutions to achieve optimal performance, i.e., data-level, algorithm-level enhancements with fraud detection ensemble model, and cost-sensitive learning. As cost-sensitive learning can be achieved with data resampling and algorithm modifications. We mainly focus on data pre-processing and ensemble learning model.

3.1 Random Forest Model

The random forest/ensemble model-derived at the base of the decision tree gives the best results compared to other classification algorithms [16]. Decision tree is an initiative algorithm that decides on the sequence of questions on data features, but it may overfit the mode due to low variance and high bias in nature. This drawback can be overcome with a random forest algorithm having important features as Gini impurity, bootstrapping, and random selection of features for each node in the decision tree. Gini impurity is a measure used by the decision tree to decide on splitting each node, which represents a probability that a sample from a node will classify incorrectly according to the distribution of samples in a node. Bootstrapping which randomly selects samples with replacement. Random set of features while considering split for each node in a decision tree. Combining all these features, random forest is made of multiple decision trees and taking average voting to make predictions make random forest an ensemble model and also a bagging example (see Fig. 1).

We select a random forest model as the best outfit of all classification algorithms for credit card fraud recognition [20].

3.2 Architecture

In the intended system, we want to focus not just on the data and algorithm-level and also on real-time data processing. As real-time credit card transactions will happen in millions, we should process all transactions with no time and decide in millisecond time. To achieve this, we are implementing distributed architecture and deploying random forest models to achieve both performance and model efficiency at the same time.

The proposed architecture is divided into three parts data streaming, data processing, and data representation (see Fig. 2).

Data streaming. Data streaming is achieved using the Kafka tool and Spark streaming job. where Kafka is the best messaging platform to process data from web/custom apps to database/data warehouse, etc., in our context process data is nothing but transactions and each transaction is treated as a message by Kafka. As

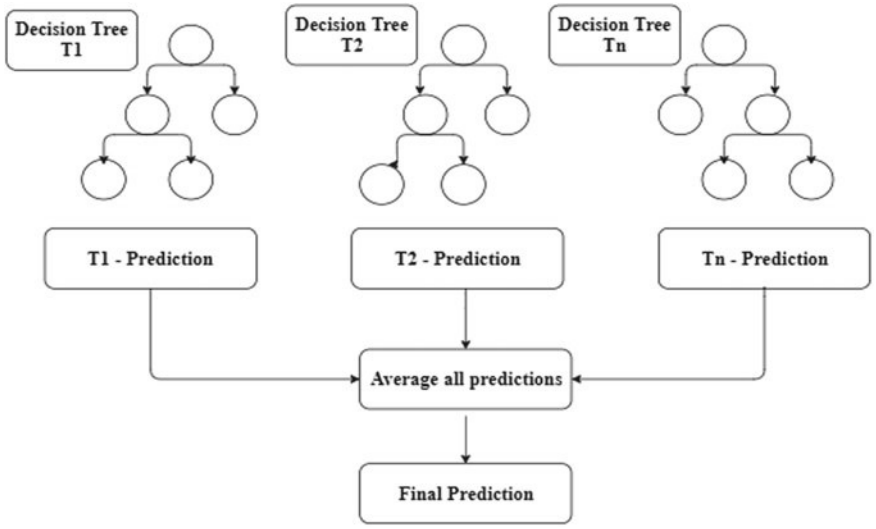

Fig. 1 Random forest model demonstration

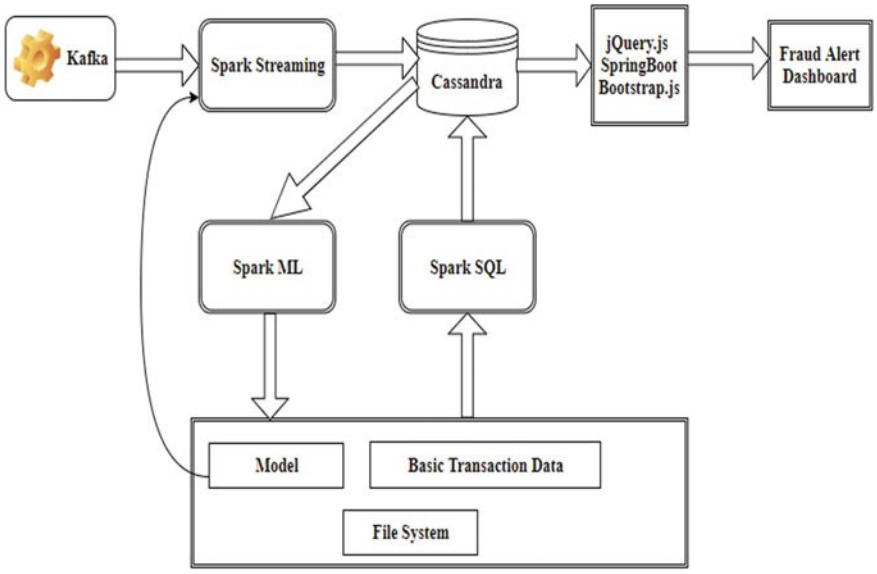

Fig. 2 Proposed system architecture

Table 1 Additional feature extraction information

Actual features	Extracted feature	Description
DOB	Age	Customer age
Customer (latitude and longitude) and Merchant (latitude and longitude)	Distance	Distance between the customer and merchant locations

part of Spark streaming, the messages received from Kafka will apply a random forest model, taking decisions will store transaction details to Cassandra an in-memory database. Cassandra is very scalable and efficient for distributed databases geared toward handling huge data volumes through commodity servers. It suits best for data processing in a distributed environment.

Data processing. Data processing is a crucial part of any machine learning algorithm. Like all algorithms, we also processing raw data in three sub-stages.

Data transformation. Data transformation includes converting string type columns values to numeric value using string indexer in Scala and scaling numeric columns converting to vector and slicing them to get normalized values which should have mean equal to zero and standard deviation to one. These transformations were applied to features before process training data to machine learning models.

Feature extraction. Extracting features is an approach of retrieving new features from existing [13]. This will improve model efficiency and helps to make the best decision on input data. In our system, we are extracting customer age based on his date of birth and distance between customer and merchant based on their latitude and longitude parameters (see Table 1). Both are important features to decide whether a transaction is a fraud or genuine, considering transaction amount value and customer spending habits like frequency of performing domestic and international transactions in the past, etc.

Balanced dataset. Any credit card company data will have 96% genuine transactions and less than or equal to 4% fraud transactions. Similarly, the dataset considered for the proposed system has 97:3 genuine and fraud transactions. Processing complete data as it is will give high variance for the machine learning model results in detecting most transactions as genuine. So, we must balance the dataset before feeding it to the machine learning model for accurate prediction. To achieve a balanced dataset, we will use the k-means clustering algorithm to make the majority class transactions count approximately equal to the minority class [21].

Data representation. We have a user interface called a fraud alert monitoring dashboard used to display any fraud transaction detected in our process. This fraud alert dashboard was developed using jQuery, Spring Boot, Bootstrap, and SockJS technologies and linked with Spark streaming job. Whenever a Spark streaming job predicts any new transaction as fraud, the fraud alert dashboard will get a notification and it will be displayed in the fraud alert monitoring dashboard.

4 Implementation

The proposed system mainly focuses on achieving the best performance while process millions of credit card transactions online. As discussed in the data processing section, the input transaction data has been transformed and applied to data balancing using the k-means algorithm. In the wake of data is balanced, it will be processed through a random forest algorithm for model creation. The streaming job will utilize the created model and will be used for processing new input data and making choices on transaction type, i.e., fraud or genuine. Corresponding fraud transaction notifications will be displayed in the UI dashboard (see Fig. 3). To achieve expected results, we used a distributed system and enhanced data processing with a random forest ensemble model. We have implemented the current system in a Linux environment with the mentioned software. Cassandra database, Spark streaming, Spark MLlib, Spark SQL, Kafka messaging system, Spring Boot, and jQuery.js for UI dashboard. Most of the projects are designed with Spark-based framework to achieve fast processing of transactions and deciding whether a transaction is a fraud or non-fraud.

Detailed process to build the proposed system.

1. Create all table schemas in Cassandra and upload data to them from the file system using Spark SQL.
2. Start pre-processing training data and prepare a balanced dataset having equal no of fraud and non-fraud transactions.
3. Train ensemble random forest model and take metrics like F1 score to decide how efficient this model to process fraud transactions better.
4. Now both the pre-processing model and ensemble model saved to the filesystem for future is used to process new incoming transactions from Kafka.
5. Start Spark streaming job which inputs both data pre-process and ensemble models and waiting for new transactions to process.
6. Kafka messaging system will start publishing new incoming transactions to the created topic.

Fraud Alert Monitoring Dashboard

cc_num	trans_time	trans_num	category	merchant	amt	distance	age
349326734419590	2020-06-28 20:18:07	e7cb35c29c41ca9...	home	Quitzon-Goyette	71	4.48	38
349326734419590	2020-06-28 20:19:39	dff60068a999de3...	entertainment	Johns Inc	47	4.63	38
5157436163845247	2020-06-28 20:19:44	2fe127c95a68344...	shopping_pos	Lynch Ltd	2013	89.07	29
5157436163845247	2020-06-28 20:19:41	8e0299d3779108...	health_fitness	Dietrich-Fadel	1774	244.45	29
5157436163845247	2020-06-28 20:19:46	80d10820173241...	shopping_pos	Hudson-Grady	1801	219.32	29
4361355512072	2020-06-28 20:19:50	c8ce99ec32c0fab...	shopping_net	Bashirian Group	89	4.32	32
5157436163845247	2020-06-28 20:21:19	c3d13d0ac25edc9...	kids_pets	Yost, Schamberge...	1978	161.33	29
5157436163845247	2020-06-28 20:21:20	f7afbdef045b1316...	home	Collier LLC	1106	112.72	29
5157436163845247	2020-06-28 20:21:29	8d5cec66e5ee72f...	food_dining	Kutch, Steuber an...	1238	60.56	29
5157436163845247	2020-06-28 20:21:25	18ebcfe1cf2da133...	grocery_pos	Hackett-Lueilwitz	2242	109.76	29

Fig. 3 Fraud notification dashboard

7. Spark streaming jobs will subscribe to the initiated topic and start consuming transactions.
8. All new transaction processing results will be stored in Cassandra fraud and non-fraud transaction tables for future machine learning model training purposes and transaction monitoring.
9. If any transaction is identified as fraud, then an alert will be receiving at fraud alerts monitoring dashboard to display the same.
10. Once in a week or month, the machine learning model will be re-trained, if model efficiency is good than the previous deployed model then the new model will be deployed stopping the old model.

5 Results

We are estimating the results of the implemented model based on evaluation metrics that were more popular known to calculate the accuracy of the machine learning classification problem [10].

5.1 Performance Metrics

The performance metrics represent the cost sensitivity of the developed model, mainly checking the wrong prediction of positive values or minority class. Current results show the very little cost for the evaluated model as we get a true positive rate equal to 1.

Confusion matrix. The error matrix helps to measure the effectiveness of machine learning classification/categorization problem. A classification problem generally differentiates between two classes and the confusion matrix will represent four different combinations of having actual and predicted values (see Table 2). Here N refers to the total number of transactions used for model evaluation. The efficiency metrics shall be evaluated against the error matrix as below.

Heat map. A heat map is a diagrammatical representation of the confusion matrix, which shows results with color presentations, i.e., color darkness will vary from minimum value to maximum value in the confusion matrix (see Fig. 4). The minimum value of the confusion matrix will have a complete light color and the maximum value

Table 2 Confusion matrix for model evaluation

Confusion matrix		
N = 2474	Actual = 1	Actual = 0
Predicted = 1	110	12
Predicted = 0	0	2352

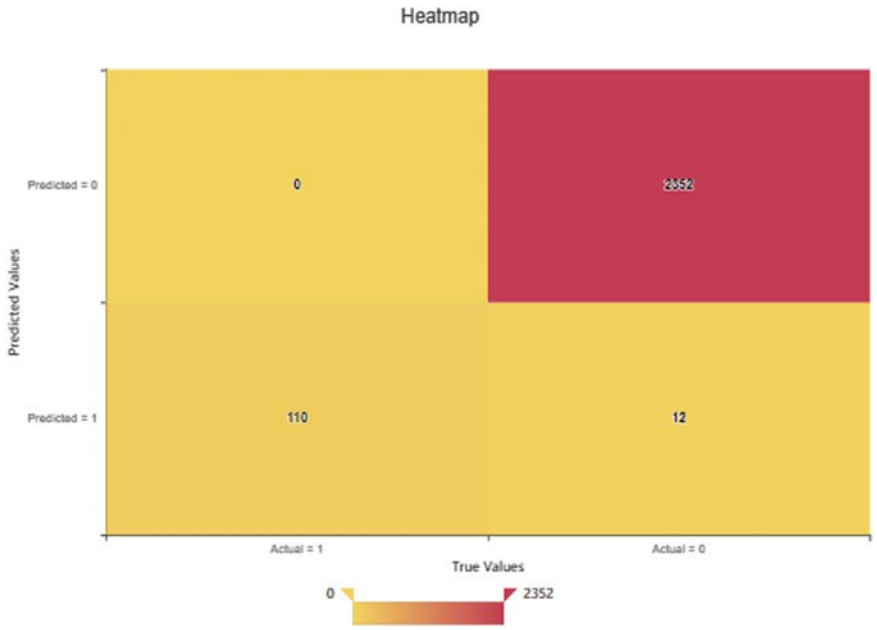

Fig. 4 Heat map representation for the confusion matrix

will be a complete darkness and the other values of color will settle between color shade from low to high based on their actual value.

All the metrics' values will be between 0 and 1 but the expected value will vary from one metric to another.

True positive rate. Total fraud transactions in the test dataset were predicted correctly as fraud. The anticipated value should be at a possible high, i.e., 1 is perfect and 100% fraud transactions were identified.

False-positive rate. Total genuine transactions in the test dataset were predicted as fraud. Presumed value should be at a possible low.

Precision and Recall. Precision is a metric to test how often the model predicts positive. Predictable value should be at a possible high, i.e., obtaining near to 1 is the finest result. The recall is nothing but a true positive rate, identifying how many fraud transactions were correctly classified.

F1 Score. F1 score is an overall extent of model accuracy which is computed using both precision and recall metrics. F1 score resembles having low false positive and false negatives so that we are finding real threats and not deviated with false alarms.

Expecting value should be at a possible high.

ROC. ROC metric is to show the performance of a classification model against problems considering all classification thresholds (see Table 3). This will be evaluated as a true positive rate against a false-positive rate. The intended value should be at a feasible high.

Table 3 Evaluation metrics summary

Measure	Formula	Result
True-positive rate/recall	TP/(TP + FN)	1
False-positive rate	FP/(FP + TN)	0.00471
Accuracy	(TN + TP)/(TP + FP + FN + TN)	0.99547
Precision	TP/(TP + FP)	0.90163
F1 score	2 * (Precision * Recall)/(Precision + Recall)	0.94827
ROC	The TP rate against the FP rate	0.99764

The evaluation of this algorithm is compared to other classification algorithms which provide efficiency up to 85% maximum, whereas the proposed model with the best pre-processing data we got 90% and above in all the times. As part of our study, we got to know if there is not much data it gives better results with classical algorithms than deep networks [22].

6 Conclusion

In the proposed manuscript, we have presented a real-time distributed fraud scrutinizing method for credit card transactions. Data pre-processing with scaling and extracting new features such as age and distance from existing data helps to prepare the best training data set for random forest ensemble learning method which is best for fraud detection out of all classification algorithms. The developed model is utilized in Spark streaming jobs that receive data from Kafka messaging system and process transactions. Transaction results will be stored in Cassandra database and if any fraud transaction comes will be displayed in the fraud alert dashboard. The random forest model gave the best accuracy with optimal scaling of data by introducing new features in it. As model selection also varies sometimes based on specific data processing, we have to decide before deploying the model with cost evaluation on the trained model.

7 Future Scope

The proposed model is the best suite for fraud detection problems though we have tested the model with few simulated datasets results that may vary slightly compared to real-time data testing and will give us more insight to decide on model performance. As an enhancement to this model, we can implement the XGBoost classification machine learning model and for processing real-time data, we can try Apache Flink to replicate real-time processing.

References

1. Popat, R.R., Chaudhary, J.: A survey on credit card fraud detection using machine learning. In: 2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI), Tamilnadu, Tirunelveli, pp. 1120–1125 (2018)
2. Kumar, P., Iqbal, F.: Credit card fraud identification using machine learning approaches. In: 2019 1st International Conference on Innovation in Info and Communication Technology (ICIICT), TN, Chennai, India, pp. 1–4 (2019)
3. Kuruwitaarachchi, N., Bhagyani, C., Mihiranga, S., Premadasa, S., Thennakoon, A.: Real-time Credit-Card Fraud Detect Using Machine Learning. https://doi.org/10.1109/CONFLUENCE.2019.8776942 (2019)
4. Rajeshwari, U., Babu, B.S.: Real-time credit card fraud detection using streaming analytics. In: 2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATCCT), Bangalore, pp. 439–444 (2016)
5. Zheng, L., et al.: A new credit card fraud detecting method based on behavior certificate. In: 2018 IEEE 15th International Conference on Network, Sensing and Control (ICNSC), Zhuhai, pp. 1–6 (2018)
6. Mittal, S., Tyagi, S.: Performance evaluation of machine learning algorithms for credit card fraud detection. In: 2019 9th International Conference on Cloud Computing, Data Science and Engineering, Noida, India, pp. 320–324 (2019)
7. ARMEL, Zaidouni, D.: Fraud detection using apache spark. In: 2019 5th International Conference on Optimization and Applications (ICOA), Morocco, Kenitra, pp. 1–6 (2019).
8. Xie, Y., Liu, G., Cao, R., Li, Z., Yan, C., Jiang, C.: A feature extraction method for credit-card fraud detection. In: 2nd International Conference on Intelligent Autonomous System (ICoIAS), Singapore, pp. 70–75 (2019)
9. Dhankhad, S., Mohammed, E., Far, B.: Supervised machine learning algorithms for credit card fraudulent transaction detection: a comparative study. IEEE Int. Conf. Info Reuse Integr. (IRI) 122–125 (2018)
10. Puh, M., Brkić, L.: Detecting credit card fraud using selected machine learning algorithms. In: 2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO) Opatija, Croatia, pp. 1250–1255 (2019)
11. Randhawa, K., Loo, C.K., Seera, M., Lim, C.P., Nandi, A.K.: Credit card fraud detection using AdaBoost and majority voting. IEEE Access **6**, 14277–14284 (2018)
12. Gyamfi, N.K., Abdulai, J.: Bank fraud detection using support vector machine. In: 2018 year IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON) Vancouver, BC, pp. 37–41 (2018)
13. Jiang, C., Song, J., Liu, G., Zheng, L., Luan, W.: Credit card fraud detection: a novel approach using aggregation strategy and feedback mechanism. IEEE Internet of Things J. **5**, 3637–3647 (2018)
14. Kasa, N., Dahbura, A., Ravoori, C., Adams, S.: Improving credit card fraud detection by profiling and clustering accounts. In: 2019 year Systems and Information Eng Design Symposium (SIEDS), VA, US, 2019-year, pp. 1–6 (2019)
15. Data generator i.e. https://github.com/namebrandon/Sparkov_Data_Generation
16. Kho, J.R.D., Vea, L.A.: Credit card fraud detection based on transaction behaviour. In: TENCON, 2017 IEEE Region 10 Conference Penang, pp. 1880–1884 (2017)
17. Dighe, D., Patil, S., Kokate, S.:Detection of credit card fraud transactions using machine learning algorithms and neural networks: a comparative study. In: 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, IN (2018)
18. Malini, N., Pushpa, M.: Analysis on credit card fraud identification techniques based on KNN and outlier detection. In: 2017 Third International Conference on Advances in Electrical, Electronics, Info, Communication, and Bio-Informatics (AEEICB), Chennai, pp. 255–258 (2017)

19. Kumar, M.S., Soundarya, V., Kavitha, S., Keerthika, E.S.: Credit card fraud detection using random forest algorithm. In: 2019 3rd International Conference on Computing and Communication Technologies (ICCCT), Chennai, India, pp. 149–153 (2019)
20. Varmedja, D., Karanovic, M., Sladojevic, S., Arsenovic, M., Anderla, A.: Credit card fraud detection—machine learning methods. In: 2019 18th International Sympos East Sarajevo, Herzegovina and Bosnia 1–5 (2019)
21. Wang, H., Zhu, P., Zou, X., Qin, S.: An Ensemble Learning Framework for Credit Card Fraud Detection Based on Training Set Partitioning and Clustering, pp. 94–98 (2018)
22. Roy, A., Sun, J., Mahoney, R., Alonzi, L., Adams, S., Beling, P.: Deep learning detecting fraud in credit card transactions. In: 2018 Systems and Information Engineering Design Symposium (SIEDS) Charlottesville, USA, pp. 129–134 (2018)