IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Ingénierie des Systèmes d'Information
Vol. 25, No. 1, February, 2020, pp. 101-106
Journal homepage: http://iieta.org/journals/isi

Machine Learning Approach on Apache Spark for Credit Card Fraud Detection

Thakur Santosh, Dharavath Ramesh*

Department of Computer Science and Engineering, Indian Institute of Technology (ISM), Dhanbad 826004, India

Corresponding Author Email: drramesh@iitism.ac.in

https://doi.org/10.18280/isi.250113

Received: 12 September 2019
Accepted: 6 December 2019

Keywords:
index terms – credit card fraud, spark, Hadoop, K-Means, decision tree

ABSTRACT

In the digital era, industries such as banking and financial organizations are facing different challenges with transaction-related activities. One of the significant challenges in financial organizations is Credit card fraud. In order to identify credit card fraud activities. In this paper, we employ an integrated hybrid approach using Apache Spark. The proposed hybrid approach is the integration of K-Means and C5.0 decision tree with an adaptive method, which is examined through Hadoop and Spark. Using K- Means, we find the closest clusters, and with the rules of a decision tree, each normal and fraud instance in the dataset is classified. This model is evaluated on one million synthetic datasets and achieved a good classification rate. We present our model with detailed experimental results and comparison with other models. This model is suitable for computationally complex datasets, and it can be applied to various fields for anomaly detection on big data.

1. INTRODUCTION

With the recent advances in internet-based applications, online transactions are being performed in a significant way at an exponential rate. The progressing numbers of online transactions are to draw illegitimate activities in various forms. These transactions leave the logs of card details. When a card is gain access to some adversary, an anomalous pattern is revealed by the transactions. Such anomalous patterns are termed as fraudulent transactions. These types of transactions are relatively less as compared to large voluminous genuine transactions. Therefore, identifying such unlawful transactions is a very problematic job that requires some relative fraud analytics. This instance created fraud analytics as a good research problem for machine learning and computational intelligence research group. Aiming at credit card fraud detection [CCFD], many methods have been proposed in recent years [1-3]. But, the analysis of the primary task of CCFD by considering, concept drift, class imbalance, feedback alert with high speed and accuracy have not been analyzed with big data technologies. Day-by-day transactions and online payments are drastically increasing; as a result, the hefty workload on the participating systems. Therefore, the computational efficiency of CCFD has become a significant factor [4]. The recently generated data, which is higher in quantity, arrives into the system with optimum velocity and can reach several Zetta Bytes. On such type of transactional data, it is difficult to work with traditional systems with greater accuracy. Apart from computational efficiency, class imbalance and concept drift have to be addressed.

In real-time, a significant delay problem is observed as a major concern in supervised samples [5]. The works proposed in recent work, do not pay attention to the *feedback recommendation* [6]. On the other hand, while designing a real-time CCFD, *sample selection bias* (SSB) may be considered since feedback is responsible for SSB [7]. This makes the additional difference between the distribution of tests and training data. One of the important factors needs to consider that companies are worried about the accuracy of the generated feedback alerts [8]. Big data technologies are the best-suited choice to answer this type of computational problems [9]. For example, service providers of credit cards could earlier analyze merely 3% of its historical data, and after each 3 to 4 days, the update was done. The latest CCFD model has the ability to analyze the complete historical data in a given time frame, with the advantage of current advanced big data technologies. At the same time, detection can be done, and updates can be created simultaneously for every 2 to 3 hrs. The attribution of big data can be done using 4 V's as; Volume, Velocity, Variety, and Veracity [4]. A huge amount of data refers to volume. The rate of speed in which data is generated as velocity. Variety refers to data heterogeneity. Finally, the accuracy of data refers to veracity.

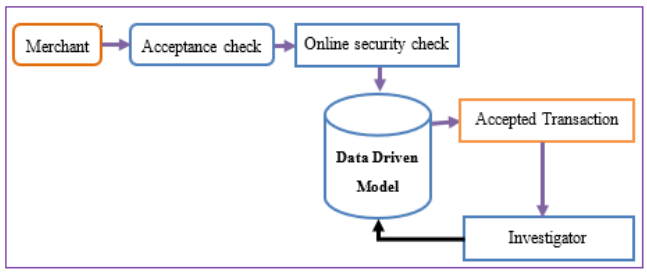


Figure 1. Credit card fraud detection process

To address the above issues, in this manuscript, we employ an integrated hybrid approach [IHA] for the detection of credit card fraud using Big Data. We implement the proposed model with the functionalities of Apache Spark on the top layer of Hadoop with a massive amount of credit card data. Here we and presented the working process of credit card fraud detection with Figure 1.

Acceptance check: The acceptance check signifies the first

mechanism level in CCFD and carries out standard authorizations of the payment process [10]. Acceptance checks consist of monitoring the PIN, balance, and an available number of attempts, credit limits, and card status. In the case of online payments, these tasks have to be performed within a few milliseconds and response has been fetched. After satisficing the above controls, then the transaction request is processed by the next layer.

Online security check: online security check is processed in different phases as transaction blocking rules, Scoring Rules, and Data Model. The transaction-blocking rules are framed by experts such as if the transaction is from the blocked site then denies the transaction. Investigators frame the rules and scoring rules manually for CCFD.

Data-Driven Model: In order to guess the probability of the features being a fraud or normal, the classifier adopts a statistical model based on historical data. Therefore, the data-driven models are trained on a set of features and cannot be inferred manually. An effective model is estimated to discover fraudulent patterns by analyzing several components of the feature. As a result, investigator experience goes beyond to find the frauds and framing rules. In this paper, we propose a new approach to emphasize on this component and improve CCFD performance to train, design, and update the data-driven model.

Investigators: Investigators are experienced professionals in CCFD analysis and responsible for the expert driven model. In precise, investigators plan transaction-blocking and scoring rules. They imagine all the notified transactions and call cardholders for verification, after having verified, they confirm “genuine” or “fraudulent” transaction, and update this information to the CCFD. These tagged variables are referred to as feedback alerts. The primary goal of data-driven model is to deliver accurate alerts. The rest of the paper is organized in the below manner.

In section 2 and section 3, we present the currently existing credit card fraud detection methods and problems. In section 4, we present the working of the proposed methodology and experimental setup, followed by results and discussion.

In recent years, credit card fraud detection attracts many reach communities and proposed various techniques. In machine learning, there are two types of methods, namely, supervised methods and unsupervised methods. The supervised learning method uses and compiles labeled training data whereas unsupervised learning compiles unlabeled training data [11]. A popular supervised learning method includes the study of decision trees, artificial neural networks, etc. A three-layer neural network model helped in detecting fraud transactions [12]. To find and learn about the behavior of fraud occurrences, Bolton used the group analysis. They identified certain breakpoint changes during the conduct of payments [13]. Abhinav et al. [11] proposed a CCFD model where incoming transactions should not possess a greater transfer rate. This transaction is considered to be a fraud by using hidden Markov model (HMM)**.** Supervised models are the most prevalent in credit fraud detection; a number of studies have stated hybrid algorithms to achieve the best performance in anomaly detection [8, 14, 15]. As an alternative of put forward new techniques in order to obtain more accuracy, we aim to achieve the desired goal by integrating the existing methods.

Here, we observe combining different algorithms and making a new stable algorithm is in earlier proposed methods. Peter et al. [12] merged the fuzzy logic with a genetic algorithm and designed the fuzzy evolutionary model to classify credit card transactions. Panigrahi et al. [16] proposed a new method by combining Dempster-Shafer and Bayesian with scoring rules. This method provides scoring optimality to identify uneven activities. To detect credit card fraud occurrence, Krishna et al. [17] proposed an efficient method by combining SSAHA and BLAST algorithms. In this model, they study past fraud transactions and classify fraud events. Ghanem et al. [18] presented a hybrid approach by applying meta-heuristic method for the detection of anomalies. This model is proved to be efficient when compared with other machine learning models proposed for CCFD.

2. PRELIMINARIES

2.1 Big data technologies

In recent days credit card payments are increasing immensely in enormous dimensions. The predominant CCFD models must be more adaptable and powerful to manage extensive volumes of information originating from differing sources. Consequently, the best answer to this issue is to make use of technologies to store and process a substantial measure of information. The most well-known open-source big data platform for storing and executing of big data is Hadoop. Hadoop Distributed file system (DFS) and MapReduce (MR) are used on a wide-scale to handle this type of information. In any case, on iterative conveyed processing, Apache Spark considered being the best in execution when contrasted with Hadoop MR [19]. In Spark's main memory uses to process the data on distributed workloads, which leverages with rapid execution. Whereas in Hadoop MR the data is stored on the disk. On iterative machine learning workloads, Hadoop MR is slower when compared to Spark [14, 20, 21]. However, Spark does not come up with its own file management system, so it requires to be integrated with some other file management system. Spark uses Resilient Distributed Datasets (RDDs), which is a distributed memory abstraction. RDDs allow the user to get the data from a distributed environment with fault tolerance.

2.2 K-means algorithm

K-Means is one of the most common and partitions based clustering algorithm. To assign the data points, K-Means works in an iterative manner. The data points are chosen randomly in a cluster and updates the cluster center. This process will continue until no change takes place in a cluster for a fixed number of iterations. The data points within the cluster are similar and dissimilar, which are outside the cluster. In K-Means, we define two measures in the form of distance between two clusters and data points, respectively. Euclidean distance is the most popular method to measure the distance between two clusters [22, 23]. Here, z and c are two points in Euclidian distance space, and it is calculated as;

$$\begin{aligned}\sqrt{\sum_{a=1}^{m}(z_{ia}-c_{ja})^2} &= \sum_{a=1}^{m}(z_{ia}-c_{ja})^2 \\ &= \sum_{a=1}^{m}(z_{ia}^2+c_{ja}^2-2z_{ia}c_{ja}) \\ &= \sum_{a=1}^{m} z_{ia}.z_{ia} + \sum_{a=1}^{m} c_{ja}.c_{ja} - 2\sum_{a=1}^{m} z_{ia}.c_{ja}\end{aligned} \quad (1)$$

The algorithmic instance of K-means is presented in the following **Algorithm 1**.

Algorithm$_1$: *K means*
Input: *Numerical (There must be a distance metric over the variable space i.e. Euclidian distance)*
Output: *The centres of each discovered cluster and the assignment of each input data to a cluster i.e. centroid*
Step$_1$: let x=(x_i,x_2,…x_n) and (c_i,c_2,…c_n)
Step$_2$: Randomly select C cluster centres
Step$_3$: ***while*** m>itr do
Step$_4$: ***for*** each x_i Distance with all centers c_j is calculated Assign x_i to nearest c_j calculate new centroid
Step$_5$: ***end while***
Step$_6$: *End*

2.3 Decision tree

The C5.0 algorithm is developed by Quinlan based on the decision tree. It is an extension of C4.5, and better than the C4.5 in terms of efficiency and memory. In the C5.0 model, samples are split on the basis of the highest information gain field [18]. C5.0 generates classifiers expressed as decision trees, but it can also construct the classifiers in a comprehensible rule set form.

The training data is segregated by using well-defined attributes, which is based on the entropy measure commonly used in the information theory [19, 24]. Most of the attributes are chosen for the classifications based on the highest information gain in the decision tree [2, 15]. The algorithmic instance of Decision Tree is presented in **Algorithm 2**.

Information Gain:

$$\begin{aligned} &\textit{Let S be a random variable } S_i,\ i\in\{1,\ldots,n\} \\ &I(S)=\sum_{i=1}^{n}-p(s_i)\log_2(p(s_i)) \\ &I(S) \text{ is continuous in } p(s_i),\ i\in\{1,\ldots,n\} \\ &G(S,N)=I(S)-\sum\frac{I(S_n)}{S}I(S_n) \\ &I(S)-\frac{I(S_1)}{S}I(S_1)-\frac{I(S_2)}{S}I(S_2)-\frac{I(S_n)}{S}I(S_n) \end{aligned} \tag{2}$$

$I(S)$ is the entropy of data set and $\sum\frac{I(S_n)}{S}I(S_n)$ is conditional entropy for the dataset given the variable N.

Algorithm$_2$: *Decision Tree*
Step$_1$: Generate a rule set
Step$_2$: Pick the most informative attribute
Step$_3$: Find the partition with the highest information gain
Step$_4$: at each result node, repeat **step$_1$** and **step$_2$**
Step$_5$: End

2.4 Real time challenges in CCFD

Class Imbalance problem: Class imbalance arises when genuine transactions far be more than the fraudulent ones. In credit card transactions, class distribution is very unbalanced, and in the meantime, frauds are naturally not more than 2% of total transactions [25]. In order to deal this issue, two different methods have been proposed as; (i) cost-based methods and (ii) sampling methods. Cost-based methods adjust the misclassification cost to the smaller class of a learning algorithm while sampling methods are used in training to maintain the stability of class distribution of the algorithm [26].

Concept Drift: With the gradual increase in credit card transactions, the spending behavior of the credit card holders is also changing gradually, which provides scope for an increase in new methods of fraudulent activities. This situation is called concept drift. Training under the concept of drift is one of the challenging tasks for data-driven models. In order to handle the concept drift, models were divided into two categories they are active and passive methods [27]. As incoming data change is detected as soon as possible, adaptation classifier updates on current supervised samples that are carefully articulated with the present state of the process. Passive methods constantly update the classifier when new supervised samples are presented without including any activating tool. Hybrid methods, sampling methods, and classifiers are learned over the current supervised methods, and they are extensively examined by using passive clarifications [1]. When data is unstable, adaptation is frequently attained by merging hybrid models.

Feedback alert and SSB: The probability of fraud is characterized by transaction feedbacks. Feedbacks signify a sort of biased training set, i.e., sample selection bias (SSB) [7]. In the learning process, in order to decrease the effect of biased samples, we use to consider weighting samples. In another approach, hybrid models have also been proposed to correct sample selection bias [28].

3. PROPOSED METHODOLOGY: INTEGRATED HYBRID APPROACH

In this section, we present the proposed hybrid model with a feedback mechanism on Spark and Hadoop to handle a large amount of data in clusters. In Hadoop Distributed file system (HDFS), initially, the data are divided into small fragments and disseminated among the nodes in clusters. In order to reduce the loss of data, the data chunk is positioned at three different replicas in HDFS. Though, the replica factor is modified according to the prerequisite. The data are accessed from Spark through Resilient Distributed datasets (RDDs).

On the other hand, K-Means is used for partitioning the dataset into K closest clusters. The Decision tree technique is then applied on each closest cluster to construct the corresponding tree for each cluster and to categorize each instance into normal form. The reason for choosing the K-Means is that its time complexity $O(ckm)$ and space complexity $O(c+k)$, where c is the number of clusters, k is the number of patterns, and m is the number of iterations. The reason for choosing C5.0 is that it is more efficient, and its decision tree is smaller in comparison with C4.5. On the other hand, the strategy of C5.0 eliminates unnecessary attributes. The integrated algorithm is depicted in **Algorithm 3**. It separately contains a classifier of feedbacks F where D indicates the test data, and $t+1$ indicates transactions at a day. At first, we train a classifier exclusively on historical data. Second, we train the classifier dynamically based on feedback and test data.

Algorithm 3: *IHA Algorithm with feedback mechanism*
Step$_1$: Test the instances $z(i)$ *where i=1…n*
Step$_2$: Read the data set
Step$_3$: Randomly select the K initial centroid of the cluster

Step4: For every instance $z(i)$ in the data set, find the closest cluster using Euclidian distance

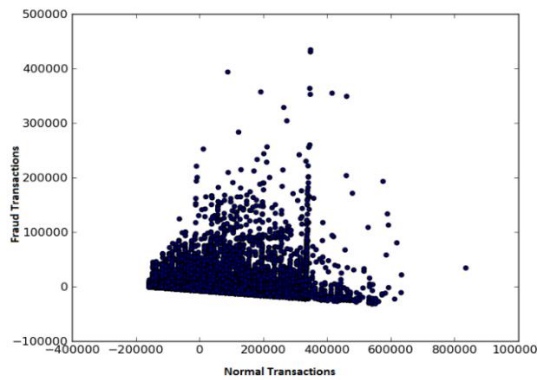
$$D((z(i),c(j), j=1\ldots k,$$
$$d((z(i),c(j))=\sqrt{\sum_{a=1}^{m}(z_{ia}-c_{ja})^2}$$

Step5: Compute the Decision Tree algorithm for closest cluster using highest information gain
Step6: Apply the test instance $z(i)$ with the use of decision tree and include it in the cluster
Step7: Classify the test instance $z(i)$ with the use of decision tree and include it in the cluster.
Step7: If $z(i)=F$
then
$z(i)$=TRAIN($D_{t+1}\ldots\ldots D_{t+n}$)
Else
$z(i)$= TRAIN({$F_{t+1}\ldots\ldots F_{t+n}$ }+ {$D_{t+1}\ldots\ldots D_{t+n}$})
Step8: Update classifer with $z(i)$
Step9: End

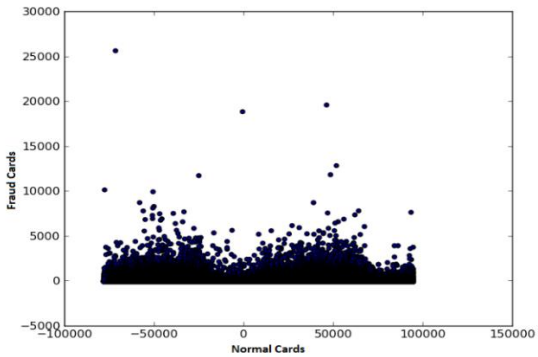
3.1 Result and experimental analysis

To perform extensive experiments, Spark 1.6.0 is installed on top of Hadoop 2.6.0 version. With host operating system as UBUNTU 14.04 LTS, the methodology is examined on three nodes. As two data nodes and one name node are the two interior components of a Spark cluster. In a cluster, there exists more than one data node and only a single name node. The work of the name node is to assign jobs to all the data nodes [23].

3.1.1 Data sets



(a). Number of fraudulent transactions



(b). Number of fraudulent cards

Figure 2. Number of fraudulent transactions and cards in the data sets

With different transaction occurrences, high velocity is seen in the resulting credit card transaction data. This results in the uncertainty or veracity in the data. The volume of the credit card fraud (*ccFraud*) dataset is somewhat complex, and processing it on a single machine is not possible. Therefore, it is executed on the Spark cluster that supports the distributed processing entity. We experiment with "*ccFraud*", data sets which in the public domain [29]. The fraudulent transaction of 2.96% in the *ccFraud* dataset renders the veracity in the data, which is why *ccFraud* dataset is highly unbalanced. This dataset contains 8 features with one million samples. From Figure 2, it is observed that there is a greater number of false transactions than false cards. However, in real-time, multiple numbers of fraud transactions are not possible with a single card on the same day [1].

To measure the fraud-detection performance consistently, we have removed the *card_id* variable from all the feature vectors to reduce the class imbalance. In the testing process, the classifier receives input as a card_id variable, which is a critical feature to detect many frauds from the same card on different days. In a real-time fraud detection system, once credit card fraud is detected, then the card is blocked. Therefore, multiple frauds are not possible on the same card.

Concept drift*:* To overcome the concept drift, initially, we trained the classifier for T days statically and never updated and compared with F, then it updates Z(i) as represented in algorithm3.

Sample Selection Bias (SSB): A typical key to correct SSB is weighting [19]. It can effectively reimburse the SSB presented by the feedback alert. For this reason, we consider the feedback F. In feedback interaction, we use weight resampling from Bayes theorem.

$$p(S\,|\,T_j)=\prod_{i=1}^{m}p(S_i\,|\,T_j)^b.(1-p(S_i\,|\,T_j))^{(1-b)}(b\in 0,1) \quad (3)$$

$\hat{p}(S_i|\,T_j)$ be the maximum-likelihood to consider risk having the grater probability of sample S_i occurs in class T_j.

$$\hat{p}(S_i\,|\,T_j)=\frac{df_{s_i,y}+1}{df_y+2} \quad (4)$$

- $df\,s_{j,y}$ the number of days in the training dataset that contain the feature and belongs to the class T_j.
- df_y is the number of days in the training dataset that belong to class T_j.
- +1 and +2 are the parameters of *Laplace smoothing*.

The selection sample s is a variable, and the value is 1 for feedback transaction f otherwise, it is 0. Hence the performance is achieved by correcting the weights to overcome the SSB and lower the influence of feedback samples for fraud detection. Let (x,y) be the probability of the sample in the training set. Here, we present the performance of f influenced by different parameters such as the number of feedback days been considered in the training model.

3.1.2 Training performance

In order to reduce the computational workloads, the number of iterations of each algorithm is fixed to a number of iterations while training the model. On Spark Cluster, the training time differs as the number of transactions increases, and the throughput also changes, but when compared to other models

such as HMM [11] and BLAST [17], the proposed model performs well. The training performance of the proposed model (IHA) is depicted in Figure 3. From Figure 3 we observe that when the transactions reach from 500000 to 1000000.We find no change in execution speed time in seconds of proposed model in hence it becomes stable.

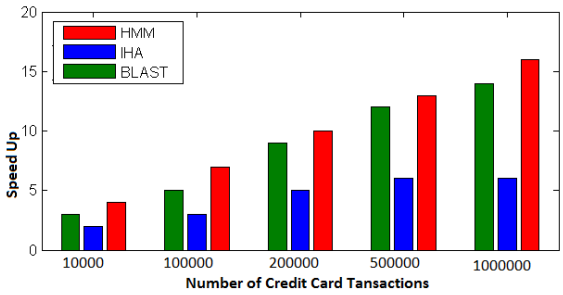


Figure 3. Training performance of the proposed algorithm (IHA) on Spark cluster

3.1.3 Streaming detection performance

In two-node cluster, when the rate of incoming payment has differed from 3000 transactions/sec to 60,000 transactions/second then the delay is observed, as shown in Figure 4. When the rate of incoming payment is less than 15,000 transactions/second, the delay is less than one second. Assume that in 48 hrs, the credit card payment system has 200,000,000, number of transactions. Then the average speed of incoming transactions is 23,148 trans/s (200,000,000 trans/48 hrs). It means our model can support near real-time detection.

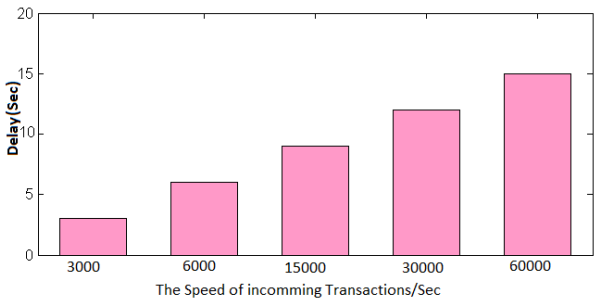


Figure 4. Delay detection of streaming on Spark

3.1.4 Receiver operating Characteristic Curve (ROC)

To check the accuracy of the prediction model, we present with ROC curve as shown in Figure 5. The ROC curve is used to see the performance of IHA model. Here, the presented results are with classification performance 94% in area under the ROC curve (AUC).

Credit card fraud detection has drawn the interest of a number of researchers, and for the research communities, it has become an interesting research problem as well. A number of CCFD techniques have been proposed in recent for the detection of credit card fraud, but each of these methods has their own advantages and disadvantages [30]. In terms of credit card fraud detection, the neural network and Fuzzy Darwinian method based CARDWATCH improves the accuracy of the system. These methods are not scalable for the high computationally complex datasets even though they have a positive response rate. The proposed model is more efficient in comparison to other models mentioned in the literature [11, 12, 17, 18]. These methods are not compatible with detection of fraud transactions for big data problems.

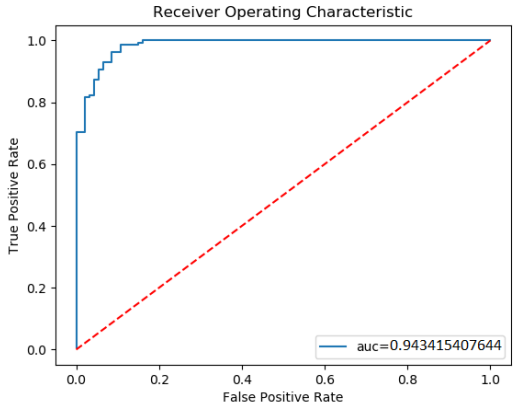


Figure 5. ROC receiver operating characteristic curve

4. CONCLUSION AND FUTURE SCOPE

In this paper, we have proposed a Spark-based integrated hybrid approach for credit card Fraud detection using big data technologies. When a large amount of data is generated by the time sequence, it turns out to be a challenging problem. The proposed integrated hybrid method is a combination of K-Means and Decision tree algorithms that can solve big data-related problems. This IHA model can also handle high-performance systems with fault-tolerant instances over big data. In this paper, we have also presented step by step experimental process to analyze the credit card fraud with big data technologies. At an average, the classification performance of the model was achieved by 94%. The proposed Spark-based IHA method is suitable for computationally complex problems. In future, this method can be applied to various fields for anomaly detection on big data. But extensive research is required to bring the model in practice.

ACKNOWLEDGEMENTS

This work is supported by Indian Institute of Technology (ISM), Dhanbad, Govt. of India. The authors wish to express their gratitude and heartiest thanks to the Department of Computer Science & Engineering, Indian Institute of Technology (ISM), Dhanbad, India for providing their continuous research support.

REFERENCES

[1] Dal Pozzolo, A., Boracchi, G., Caelen, O., Alippi, C., Bontempi, G. (2017). Credit card fraud detection: a realistic modeling and a novel learning strategy. IEEE Transactions on Neural Networks and Learning Systems, 29(8): 3784-3797. https://www.doi.org/10.1109/TNNLS.2017.2736643

[2] Behera, T.K., Panigrahi, S. (2017). Credit card fraud detection using a neuro-fuzzy expert system. In Computational Intelligence in Data Mining, Springer, Singapore, pp. 835-843.

[3] Fiore, U., De Santis, A., Perla, F., Zanetti, P., Palmieri, F. (2017). Using generative adversarial networks for

improving classification effectiveness in credit card fraud detection. Information Sciences, 479: 448-455. https://doi.org/10.1016/j.ins.2017.12.030

[4] Dai, Y., Yan, J., Tang, X., Zhao, H., Guo, M. (2016). Online credit card fraud detection: A hybrid framework with big data technologies. 2016 IEEE Trustcom/BigDataSE/ISPA, Tianjin, China, pp. 1644-1651. https://doi.org/10.1109/TrustCom.2016.0253

[5] Kamaruddin, S., Ravi, V. (2016). Credit card fraud detection using big data analytics: Use of Psoaann based one-class classification. In Proceedings of the International Conference on Informatics and Analytics, pp. 1-8.

[6] Plasse, J., Adams, N. (2016). Handling delayed labels in temporally evolving data streams. 2016 IEEE International Conference on Big Data, pp. 2416-2424.

[7] Cortes, C., Mohri, M., Riley, M., Rostamizadeh, A. (2008). Sample selection bias correction theory. In International Conference on Algorithmic Learning Theory, Berlin, Heidelberg, pp. 38-53. https://doi.org/10.1007/978-3-540-87987-9_8

[8] Paramjeet, Ravi, V., Naveen, N., Rao, C.R. (2012). Privacy preserving data mining using particle swarm optimisation trained auto-associative neural network: An application to bankruptcy prediction in banks. International Journal of Data Mining, Modelling and Management, 4(1): 39-56. https://doi.org/10.1504/IJDMMM.2012.045135

[9] Wang, S. (2010). A comprehensive survey of data mining-based accounting-fraud detection research. 2010 International Conference on Intelligent Computation Technology and Automation, Changsha, China, pp. 50-53. https://doi.org/10.1109/ICICTA.2010.831

[10] Van Vlasselaer, V., Bravo, C., Caelen, O., Eliassi-Rad, T., Akoglu, L., Snoeck, M., Baesens, B. (2015). APATE: A novel approach for automated credit card transaction fraud detection using network-based extensions. Decision Support Systems, 75: 38-48. https://doi.org/10.1016/j.dss.2015.04.013

[11] Srivastava, A., Kundu, A., Sural, S., Majumdar, A. (2008). Credit card fraud detection using hidden Markov model. IEEE Transactions on Dependable and Secure Computing, 5(1): 37-48. https://www.doi.org/10.1109/TDSC.2007.70228

[12] Bentley, P.J., Kim, J., Jung, G.H., Choi, J.U. (2000). Fuzzy Darwinian detection of credit card fraud. In the 14th Annual Fall Symposium of the Korean Information Processing Society, pp. 1-4.

[13] Bolton, R.J., Hand, D.J. (2001). Unsupervised profiling methods for fraud detection. Credit Scoring and Credit Control VII, 235-255.

[14] Hayes, M.A., Capretz, M.A. (2014). Contextual anomaly detection in big sensor data. 2014 IEEE International Congress on Big Data, Anchorage, AK, USA, pp. 64-71. https://doi.org/10.1109/BigData.Congress.2014.19

[15] Muniyandi, A.P., Rajeswari, R., Rajaram, R. (2012). Network anomaly detection by cascading k-Means clustering and C4. 5 decision tree Algorithm. Procedia Engineering, 30: 174-182. https://doi.org/10.1016/j.proeng.2012.01.849

[16] Panigrahi, S., Lenka, R.K., Stitipragyan, A. (2016). A Hybrid distributed collaborative filtering recommender engine using apache spark. Procedia Computer Science, 83: 1000-1006. https://doi.org/10.1016/j.procs.2016.04.214

[17] Kundu, A., Panigrahi, S., Sural, S., Majumdar, A.K. (2009). BLAST-SSAHA hybridization for credit card fraud detection. IEEE Transactions on Dependable and Secure Computing, 6(4): 309-315. https://www.doi.org/10.1109/TDSC.2009.11

[18] Ghanem, T.F., Elkilani, W.S., Abdul-Kader, H.M. (2015). A hybrid approach for efficient anomaly detection using metaheuristic methods. Journal of Advanced Research, 6(4): 609-619. https://doi.org/10.1016/j.jare.2014.02.009

[19] Dittrich, J., Quiané-Ruiz, J.A. (2012). Efficient big data processing in Hadoop MapReduce. Proceedings of the VLDB Endowment, 5(12): 2014-2015.

[20] Meng, X.R., Bradley, J., Yavuz, B., Sparks, E., Venkataraman, S., Liu, D., Freeman, J., Tsai, D.B., Made, M., Owen, S., Xin, D., Xin, R., Franklin, J.M., Zadeh, R Zaharia, M., Talwalkar, A. (2016). Mllib: Machine learning in apache spark. Journal of Machine Learning Research, 17(34): 1-7.

[21] Zaharia, M., Chowdhury, M., Das, T., Dave, A., Ma, J., Mccauley, M., Franklin, M., Shenker, S., Stoica, I. (2012). Fast and interactive analytics over Hadoop data with Spark. USENIX Login, 37(4): 45-51.

[22] Wu, X., Zhu, X., Wu, G.Q., Ding, W. (2014). Data mining with big data. IEEE transactions on Knowledge and Data Engineering, 26(1): 97-107.

[23] Thakur, S., Dharavath, R. (2018). KMDT: A hybrid cluster approach for anomaly detection using big data. In Information and Decision Sciences, 169-176.

[24] Gaddam, S.R., Phoha, V.V., Balagani, K.S. (2007). K-Means+ ID3: A novel method for supervised anomaly detection by cascading K-Means clustering and ID3 decision tree learning methods. IEEE Transactions on Knowledge and Data Engineering, 19(3): 345-354. https://doi.org/10.1109/TKDE.2007.44

[25] Krivko, M. (2010). A hybrid model for plastic card fraud detection systems. Expert Systems with Applications, 37(8): 6070-6076. https://doi.org/10.1016/j.eswa.2010.02.119

[26] Elkan, C. (2001). The foundations of cost-sensitive learning. In International Joint Conference on Artificial Intelligence, 17(1): 973-978.

[27] Alippi, C., Boracchi, G., Roveri, M. (2013). Just-in-time classifiers for recurrent concepts. IEEE Transactions on Neural Networks and Learning Systems, 24(4): 620-634. https://doi.org/10.1109/TNNLS.2013.2239309

[28] Fan, W., Davidson, I., Zadrozny, B., Yu, P.S. (2005). An improved categorization of classifier's sensitivity on sample selection bias. Fifth IEEE International Conference on Data Mining (ICDM'05), Houston, TX, USA. https://doi.org/10.1109/ICDM.2005.24

[29] ccFraud Dataset: Apr. (2017). http://packages.revolutionanalytics.com/datasets/, accessed on Jun. 14, 2016.

[30] Ngai, E.W.T., Hu, Y., Wong, Y.H., Chen, Y., Sun, X. (2011). The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. Decision Support Systems, 50(3): 559-569. https://doi.org/10.1016/j.dss.2010.08.006