Proceedings of the Fifth International Conference on Trends in Electronics and Informatics (ICOEI).
IEEE Xplore Part Number:CFP21J32-ART; ISBN:978-1-6654-1571-2

Credit Card Fraud Detection using Apache Spark Analysis

1st Ananthu S
P G Scholar
Department of Computer Science & IT
Amrita School of Arts and Sciences,
Amrita Vishwa Vidhyapeetham
Kochi, India
ananthusreekumar69@gmail.com

2nd Nithin Sethumadhavan
P G Scholar
Department of Computer Science & IT
Amrita School of Arts and Sciences,
Amrita Vishwa Vidhyapeetham
Kochi, India
nithin.sethumadhavan@gmail.com

3rd Hari Narayanan AG
Assistant Professor
Department of Computer Science & IT
Amrita School of Arts and Sciences,
Amrita Vishwa Vidhyapeetham
Kochi, India
hariag2002@gmail.com

2021 5th International Conference on Trends in Electronics and Informatics (ICOEI) | 978-1-6654-1571-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICOEI51242.2021.9453011

***Abstract*—Credit card fraud is an ever-growing threat to the financial industry. High dependence on internet technology leads to an increase in fraudulent transactions.Nowadays, banking sector is the most vulnerable for attacks as most of the transactions are done through credit cards and net banking.Many techniques have been proposed to tackle this problem. This paper mainly focus on recognizing fraudulent transactions by analyzing the previous set of transaction records. This research work attempts to integrate big data analytics along with machine algorithms for fast detection of large real-time data. This paper gives a brief comparison of some of the machine learning techniques using Big data.**

***Index Terms*—E-commerce,Credit Card Fraud Detection,Big Data,Machine Learning,Logistic Regression,Decision Tree Classifier,Random Forest Classifier,Spark.**

I. Introduction

Electronic commerce or commonly known as e-commerce plays a vital role in today's world.Credit card is the most common method for online purchases or payments.As the number of credit card transactions increases daily, the attacks towards these transactions also increases at an alarming rate.These attacks brings a great loss to bank and card holder that affects the development of banks. By introducing EMV pin and chip technology, the security of physical card has been boosted up but it hasn't been consistent globally and opened door for some global bandits. Thus,with the aim of achieving a good and fewer fraud detection,various firms and institutions are found investing great amount of money,time and resources for the development of an efficient algorithm and bigdata technology.Big data have the ability of using large volumes of data that is generated massively in high speed which ends up in increasing accuracy in the detection of fraud in the credit card transactions. So, this study attempts to analyze the role of big data in credit card fraud detection and how it can be effective in using machine learning algorithms for analysing the large datasets.

BIG DATA: Big Data is a computing technique which generates value from a very large dataset.A data is referred as a big data not only depending upon the size but also the context used in it.Mainly there are four characteristics for big data:

VOLUME : It refers to the quantity of data collected.

VELOCITY :Refers to the speed in which the data is generated.

VARIETY : Refers the various formats of data collected,from numeric and structured , unstructured audio,video,email e.t.c.

MACHINE LEARNING : Machine learning is a part of AI which helps the machine in learning automatically from past experiences or historical data . It has mainly three categories of algorithms.The algorithm which includes prediction of dependent variable from a set of predictors is Supervised Learning. The algorithm which do not contain a dependent variable for prediction is Unsupervised learning. Reinforcement learning is the type of algorithm in which the machine is trained to make certain decisions by learning from previous experiences and captures the best possible knowledge to make accurate business decisions.

Here, the supervised learning algorithms like logistic regression,decision tree classifier and random forest classifier are compared in order to find the best one for detecting frauds in credit card transactions.

APACHE SPARK: Apache Spark is an open-sourced distribued engine developed in 2009 at UC Berkeley AMPLab with an aim of analysing and processing the big data in real-time.It is mainly designed for the fast computation of data.It supports multiple programming languages and allows the developers to write applications in R,Java,Scala or Python.When compared to Hadoop, Spark processes data 100 times faster as it is an in-memory computing framework.RDD or Resilient Distributed Data is the building block of Spark which saves huge time taken in reading and writing operations.These RDDs are distributed among various nodes so that the failure of one worker node will no affect the RDDs and it prevents data loss.

Apache Spark framework is divided into three layers:
*Spark Core

*Spark Ecosystem

*Spark Resource Management

Spark Core is the base engine and it is widely used for parallel and distributed processing of data. Spark Ecosystem

provides another additional library and it is built on top of Spark Core to enhance query processing in real-time.

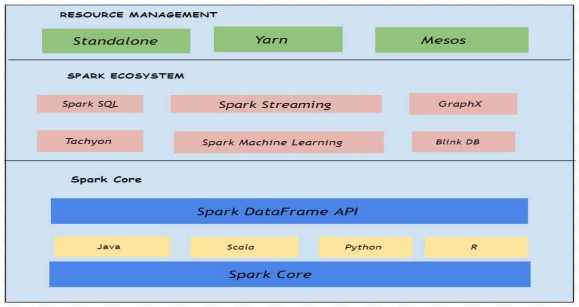

Fig. 1. Spark Architecture

II. Related Works

In 2016 S.V.Phulari et al.[1]. identifies the limitations of other related fields and also defines the types of frauds,technical nature of data and their performance metrics.

The paper "Big data analytics techniques for credit card fraud detection" was published in the year 2015 proposed by [M. Sathyapriya , Dr. V. Thiagarasu]. This paper compares the main four techniques of big data Hadoop,Spark,MapReduce and Flink based on various features and suggests Spark as the better big data analytic technique.

N Malini et al. proposed a paper "Analysis on credit card fraud detection techniques by data mining and big data approach".This paper discuss about the various types of credit card frauds involved in both physical or virtual cards ,precaution steps to avoid these frauds and also big data techniques to detect these frauds.

R Ramyakalyani and D Umadeviproposed a paper named "Credit Card Fraud Detection using Genetic Algorithm" which focuses on developing a technique for generating test data and generating fraudulent transactions using GA. It also identifies that the possibility of fraudulent transactions can be soon found out after credit card transactions by using GA.

Masoumeh Zareapoor and Pourya Shamsolmoali[12] evaluates the performance of three levels of big data processing methods with the help of bagging ensemble and decision tree algorithms. It is found that BEM takes only less time and performance is stable gradually during evaluation of credit card fraud transaction.

Poonam Salwan et al.[6] discusses the relationship of E-Governance with massive data and how a big data tool like Apache Spark can help analyze collected information precisely at a very high speed. It also gives an understanding into the Apache Spark framework as well as the implementation strategies of analyzing the government collected datasets.

Hangjun Zhou1 et al.[16], where an approach for credit card fraud detection in e-commercial transactions is proposed with four modules with the help of big data analytics tool and machine learning techniques.

Dr. MD Nadeem Ahmed, Asif Afthab, and Mohammed Mazhar Nezami [4] provides a comparison and performance of various technologies used in Apache Spark and Hadoop. It suggested that Spark and related technologies are much better than traditional databases from a processing perspective.

Salman Salloum proposed a paper called "Big Data Analytics on Apache Spark" which focuses on important properties of Apache Spark in big data analytics. It discusses the various components, features, and architecture of Spark.

III. Proposed System

The existing system uses small datasets for fraud detection.The proposed system mainly focuses on using big data analytics which can be used for implementing an algorithm with the help of machine learning techniques.

The dataset used here are transactions made by European credit cardholders .We have collected about 100GB of datasets from various sources."There were 30 features in the dataset.To protect confidentiality , 28 out of the 30 had been transformed using PCA and they had unknown labels.The output of PCA transformation is used as the numerical input variables in the dataset.The key elements obtained from the dataset are the features V1,..V28.'Amount' and 'Time' are the only features and are not transformed with PCA. 'Time' shows the time taken in seconds between each transaction and the initial transaction. 'Amount' is the Amount debited during the credit card transactions in EUROS.The feature 'Class' is the response variable which takes 1 as its value in case of fraud and takes 0 otherwise.The dataset contains transactions made by credit cards in September 2013 by European cardholders.

Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	0.090794	-0.551600	-0.617801	-0.991390	-0.311169	1.468177	-0.470401	0.207971
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	-0.166974	1.612727	1.065235	0.489095	-0.143772	0.635558	0.463917	-0.114805
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	0.207643	0.624501	0.066084	0.717293	-0.165946	2.345865	-2.890083	1.109969
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	-0.054952	-0.226487	0.178228	0.507757	-0.287924	-0.631418	-1.059647	-0.684093
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	0.753074	-0.822843	0.538196	1.345852	-1.119670	0.175121	-0.451449	-0.237033

V18	V19	V20	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
0.025791	0.403993	0.251412	-0.018307	0.277838	-0.110474	0.066928	0.128539	-0.189115	0.133558	-0.021053	149.62	0
-0.183361	-0.145783	-0.069083	-0.225775	-0.638672	0.101288	-0.339846	0.167170	0.125895	-0.008983	0.014724	2.69	0
-0.121359	-2.261857	0.524980	0.247998	0.771679	0.909412	-0.689281	-0.327642	-0.139097	-0.055353	-0.059752	378.66	0
1.965775	-1.232622	-0.208038	-0.108300	0.005274	-0.190321	-1.175575	0.647376	-0.221929	0.062723	0.061458	123.50	0
-0.038195	0.803487	0.408542	-0.009431	0.798278	-0.137458	0.141267	-0.206010	0.502292	0.219422	0.215153	69.99	0

Fig. 2. Dataset Records

Apache Spark Implementation

Apache Spark can work with different clustering technologies like Apache Mesos,Yarn and it can also work as Standalone. Spark uses a master slave architecture which

consists of a driver program that can run on a master node and it can also run on client node depending upon the configuration.It have multiple executors which can run on worker nodes.The Spark code behaves as a driver program and creates a SparkContext, which is an entry point of any Spark functionality.The driver program interacts with cluster manager and the SparkContext is responsible for taking the application request to the driver manager.Spark applications internally runs as series or set of tasks or processes in cluster and it can be Mesos,Yarn or Standalone master itself.At highlevel a job is split into multiple tasks and those tasks will be distributed over slave nodes or worker nodes so whenever any transformation take place, RDDs are created and these RDDs are distributed across multiple nodes.

In this paper Yarn is taken as the cluster manager.There are node managers which are running on multiple machines and each machine has ram and cpu allocated for these node managers.Also there are data nodes running on same machines which is used for getting Hadoop related data.So, whenever an application wants to process the data the SparkContext contacts the resource manager and these resource managers make a request for the containers (which is a combination of cpu and ram) to the node managers of the machines wherever the relevant data resides.Once the node manager approves the request, resource manager starts an extra piece of code called 'App master' which is responsible for execution of the applications. Containers have the executor process and these process is taking care of application related tasks.

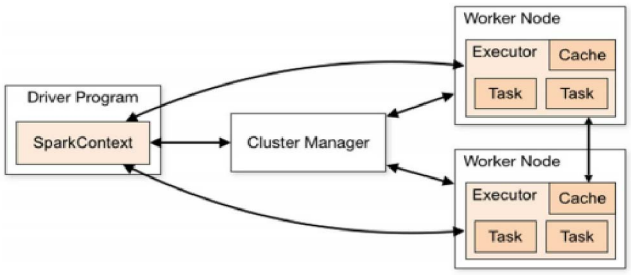

Fig. 3. Spark Implementation

A. Experiment

Machine learning algorithms comparison :

Decision Tree Classifier:

The supervised learning method which can be used for classification as well as regression models. This algorithm creates a model in which the target variable's value can be predicted by learning simple decision rules from the data features.

Decision Tree Classification

```
dt_classifier = DecisionTreeClassifier(criterion = 'entropy', random_state = 0)
dt_classifier.fit(xtrain, ytrain)
```

```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='entropy',
                       max_depth=None, max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort='deprecated',
                       random_state=0, splitter='best')
```

```
# Predicting
y_pred_decision_tree = dt_classifier.predict(xtest)
```

```
print("y_pred_decision_tree : \n", y_pred_decision_tree)
```

```
y_pred_decision_tree :
 [0 0 0 ... 0 0 0]
```

```
cm_decision = confusion_matrix(ytest, y_pred_decision_tree)
print("confusion Marix : \n", cm_decision)
```

```
confusion Marix :
 [[71052    30]
 [   25    95]]
```

Fig. 4. Code Snippet of Decision Tree Classifier

Logistic regression: Logistic Regression is a supervised learning classification algorithm in which the output of dependent variable can be predicted by using independent variables.

Random Forest Classifier:

Random Forest is a supervised learning algorithm . Both classification and regression tasks can be done by this algorithm.In Random Forest algorithm,multiple classifiers are combined to solve a complex problem and thus improves the performance of the model.As the name suggests,this algorithm creates the forest with number of decision trees and the robustness of prediction increases along with the number of trees.It can handle large set of data with higher dimensionality.

B. Result

Here we have collected about 100GB of data from different sources and compared the above machine learning algorithms and found out their accuracy,error rate,specificity and sensitivity for the given dataset as shown in Fig.7,Fig.9 and Fig.11. We are getting high perfomance using these three algorithms than the algorithms seen in paper[7].According to Fig.8,Fig.10 and Fig.12 we can see that by using Random Forest algorithm,we are getting high accuracy than the other mentioned alogorithms. The reason for high accuracy in Random Forest is that they are unbiased,as there are multiple trees in random forest and these trees are trained on a sub-group of data.The Random forest algorithm is stable and works well in

categorical features. The following are the results found during the Experiment:

LOGISTIC REGRESSION

```
[ ] from sklearn import datasets, linear_model

[ ] logistic = linear_model.LogisticRegression(C=1e5)
    logistic.fit(xtrain, ytrain)
    print("Score: ", logistic.score(xtest, ytest))

    Score:  0.9992977725344794

[ ] y_predicted = np.array(logistic.predict(xtest))
    y_right = np.array(ytest)

    cm3 = confusion_matrix(y_right, y_predicted)
    #print("Confusion matrix:\n%s" % confusion_matrix)
    print("Confusion Matrix : \n\n", cm3)

    Confusion Matrix :

     [[71071    11]
     [   39    81]]
```

Fig. 5. Code Snippet of Logistic Regression

DECISION TREE CLASSIFIER	
Accuracy_Decision	99.9227549787927
Error_rate_decision	0.0772450212072695
Specificity_decision	76.0
Sensitivity_decision	99.964826877893

Fig. 7. Findings of Decision Tree

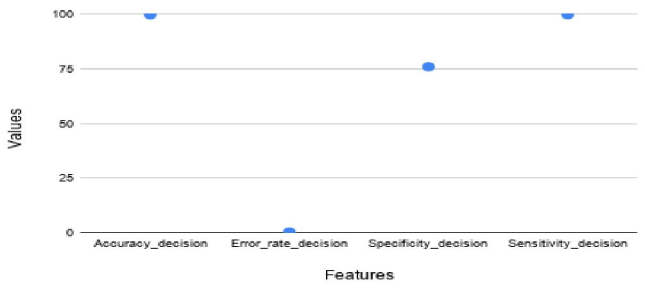

Fig. 8. Graphical Representation of Decision Tree Classifier

Random Forest Classification

```
[ ] svc_classifier = SVC(kernel = 'rbf', random_state =0)
    svc_classifier.fit(xtrain, ytrain)

    SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
        decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf',
        max_iter=-1, probability=False, random_state=0, shrinking=True, tol=0.001,
        verbose=False)

[ ] # Predicting
    y_pred2 = svc_classifier.predict(xtest)

[ ] print("y_pred_randomforest : \n", y_pred2)

    y_pred_randomforest :
     [0 0 0 ... 0 0 0]

    cm2 = confusion_matrix(ytest, y_pred2)
    print("Confusion Matrix : \n\n", cm2)

    Confusion Matrix :

     [[71077     5]
     [   44    76]]
```

Fig. 6. Code Snippet of Random Forest

Logistic Regression	
Accuracy_Logistic	99.9297772534479
Error_rate_Logistic	0.0702227465520632
Specificity_Logistic	88.0434782608696
Sensitivity_Logistic	99.94515539305301

Fig. 9. Findings of Logistic Regression

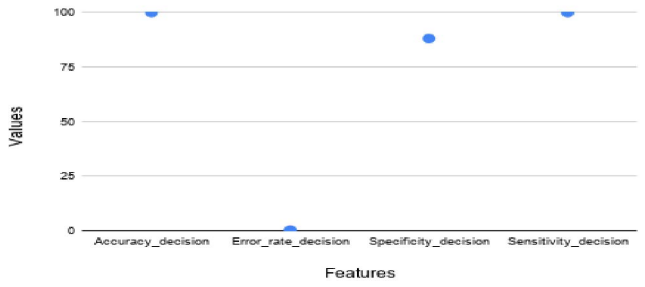

Fig. 10. Graphical Representation of Logistic Regression

Random Forest Classifier	
Accuracy_svc	99.93118170837899
Error_rate_svc	0.0688182916210219
Specificity_svc	93.8271604938272
Sensitivity_svc	99.93813360329578

Fig. 11. Findings of Random Forest

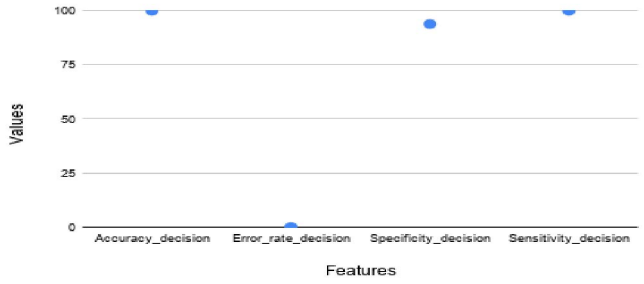

Fig. 12. Graphical Representation of Random Forest

IV. Conclusion

The credit card fraudulent is very much prevalent now-a- days. There have been a lot of cases registered recently. Hence, it's very much essential to develop a method for the detection of such fraudulent cases. In this study, we propose a method for detecting fraud in large datasets by integrating ma- chine learning algorithms with big data analytics. Here we used Apache Spark as the big data analytics tool. From our analysis , it's found that Spark is the better option than Hadoop, Flink and MapReduce.This paper gives a comparison of different machine learning algorithms like decision tree classifier, logistic regression, and random forest by using the collected dataset. From Fig.13,It is clear that Random forest was found to be performing the best in terms of accuracy and error rate . So we are suggesting a new model to detect real-time fraud transactions for high streaming real-time data by using Apache Spark with the help of random forest machine learning algorithm.

Fig. 13. Comparison of the three machine learning algorithms

V. Future Study

Here we already tried our experiment with datasets for a particular area.In future we will be trying to do the same for different locations.

References

[1] S.V.Phulari, Umesh Shantling Lamture, Sumit Vilas Madage , Kunal Tirupati Bhandari (2016),"Pattern Analysis and Fraud Detection using Hadoop Framework",International Journal of Engineering Science and Innovative Technology(IJESIT).
[2] M. Sathyapriya , Dr. V. Thiagarasu (2017) , "Big data analytics techniques for credit card fraud detection", International Journal of Science and Research.
[3] Fransisca Nonyellum and Ogwueleka , "Fraud Detection using Neural Networks".
[4] Dr. Nadeem Ahmed, Asif Afthab, and Mohammedd Mazhar Nezami (2020) ,"A Technological Survey On Apache Spark And Hadoop Technologies" ,International Journal of Scientific and Technology Research Volume 9, Issue 01.
[5] Salman Salloum, Ruslan Dautov, Xiaojun Chen and Patrick Xiaogang (2016), "Big Data Analytics on Apache Spark",Springer International Publishing Switzerland.
[6] Poonam Salwan and Veerpaul Kaur Maan (2020) ,"Integrating E-Governance with Big Data Analytics using Apache Spark"",International Journal of Recent Technology and Engineering (IJRTE).
[7] Shailesh S Ghosh , "Credit Card Fraud Detection using Hidden Markov Model".
[8] N Malini , Dr M Pushpa (2017) , "Analysis on credit card fraud detection techniques by data mining and big data approach",International Journal of Research in Computer Applications and Robotics.
[9] R Ramyakalyani and D Umadevi(2012) , "Credit Card Fraud Detection using Genetic Algorithm"International Journal of Scientific & Engineering Research Volume 3, Issue 7.
[10] I.Sadgali,N Sael and F Benabobu (2018) , "Performance of Machine Learning Techniques in detection of Financial Frauds", ScienceDirect Procedia Computer Science 148 (2019).
[11] Asha RB and Sureshkumar , "Credit Card Fraud Detection Using Artificial Neural Network" , Global Transition Proceedings.
[12] Masoumeh Zareapoor and Pourya Shamsolmoali (2015) , "Application of MasterCard Fraud Detection: Based on Bagging Ensemble", ScienceDirect Procedia Computer Science 48.
[13] Amanze BC and Onukwugha CG (2018), "Data Mining Applications in Credit Card Fraud Detection System" ,International Journal of Trend in Research and Development.
[14] Dr.Pal , "Fraud Detection in Health Insurance domain: A Big Data Application with data mining approach".
[15] Jyoti Guru, Sai Kiran, Rishab Kumar, Deepak Kataria, Naveen Kumar, and M Sharma (2018), "Credit Card Fraud Detection using Naïve Bayes model and KNN Classifier", International Journal of Advance Research, Ideas And Innovations in Technology (IJARIIT).
[16] Hangjun Zhou1, 2, *, Guang Sun1, 3, Sha Fu1, Wangdong Jiang1 and Juan Xue1,"A Scalable Approach for Fraud Detection in Online E-Commerce Transactions with Big Data Analytics".
[17] Kumar, T. Senthil. "Data Mining Based Marketing Decision Support System Using Hybrid Machine Learning Algorithm." Journal of Artificial Intelligence 2, no. 03 (2020): 185-193.
[18] Manoharan, Samuel. "Population Based Meta Heuristics Algorithm for Performance Improvement of Feed Forward Neural Network." Journal of Soft Computing Paradigm (JSCP) 2, no. 01 (2020): 36-46.