

Financial Fraud Detection adopting Distributed Deep Learning in Big Data

Priyanka Purushu[+], Jongwook Woo*
[+]AT&T
Department of Information System, California State University
Los Angeles, CA, USA
[e-mail: priyanka.purushu89@gmail.com, *jwoo5@calstatela.edu]
**Corresponding author: Jongwook Woo*

Abstract

This paper presents Financial Fraud Detection on transactional activity. We can classify the transaction as fraud by training the models using unified analytics systems. The Random Forest algorithm of the classification model is the optimal algorithm for fraud detection using Spark ML with the massive data set. We show that the Feed Forward prediction model, a distributed deep learning on Spark, presents the better accuracy in **recall** with the same data set under the similar computing time, which also leverages the existing Big Data Spark systems. It provides that integrating Deep Learning into the Big Data platform for fraud detection presents the better accuracy comparing to the legacy classification models in Big Data machine learning.

Keywords: Fraud Detection, Big Data, Deep Learning, Unified Analytics Systems, Predictive Analysis, Distributed Computing, Spark, Feed Forward

1. Introduction

We can define Big Data as non-expensive frameworks, mostly in distributed parallel computing systems, which can store a large-scale data and process it in parallel. A large-scale data means a data of giga-bytes or more, which cannot be processed or too expensive using traditional computing systems [2, 6].

Financial frauds can be a devastating issue with extensive ramifications on any business. In the data-driven world, we can track down the fraudulent transactions by analyzing the massive transaction data set with the use of Big Data platforms and data mining approaches.

Spark Big Data engine is more efficient at iterative computations and thus well-suited for the development of large-scale machine learning applications such as fraud detection than the legacy MapReduce Big Data solution [5].

We adopt *PaySim*'s synthetic dataset while financial datasets are not publicly available due to the nature of the information. A synthetic transactional data was developed by *PaySim* simulator which incorporated both: normal customer behavior and fraudulent behavior [4]. We aim at doing predictive analysis on the target value which is column "**isFraud**" and detect if a money transaction is a fraud or not.

In this paper, we adopt Spark big data architecture and develop predictive models, which is linearly scalable to compute massive data set by adding more spark nodes to the cluster with respect to the data set. We have analyzed the data with two machine learning platforms: Apache Spark ML and Spark Deep Learning (DL).

2. Related Work

Kamaruddhin implements a hybrid architecture of Particle Swarm Optimization and Auto-Associative Neural Network for one-class classification in Spark computational framework

This research was supported by the AWS Cloud Credits for Research program

to detect credit card fraud with an accuracy of 89% [3]. However, they worked on comparatively smaller dataset of 291.7MB in size that contains only 9 features.

Pryanka adopts Spark that contains the package called *MLlib*. *MLlib* provides fast, distributed implementations of machine learning algorithms, and she developed *logistc regression*, *decision tree*, and *random forest* models for fraud detection in Spark cluster and compares the accuracy and computing time, even including the traditional sequential machine [1].

In this paper, we extend *Pryanka*'s approach by adopting deep learning classification model, *Feed Forward*, and compare the accuracy and computing time with the legacy Big Data models in Spark cluster.

3. Financial Transaction Data for Fraud Detection

For the fraud detection experiment, we use a synthetic dataset generated using the simulator *PaySim* [4]. *PaySim* simulates mobile money transactions extracted from one month of financial logs from a mobile money service implemented in an African country.

The data has a size of 470 MB with 6,362,620 rows. The dataset contains 11 attributes and the target column is '**isFraud**'. The dataset provides 5 numeric attributes (*amount, oldbalanceOrg, newbalanceOrg, oldbalanceDest, newbalanceDest*), 4 categorical attributes *(step, type, isFraud, isFlaggedFraud*) and two string attributes (*nameOrig, nameDest*).

The dataset contains 98.87% non-fraud transactions and 0.12% fraud transactions which implies a big imbalance in the data. **SMOTE** is adopted for sampling the data set, which stands for Synthetic Minority Over Sampling Technique. It takes a subset of data from the minority class and creates new synthetic similar instances. . **SMOTE** helps to generate data more as increasing percent of minority class from 0.19% to 11%. That is, it helps balancing data and avoiding overfitting

As a feature engineering, we drop the attribute **step** because there is no correlation between the time for the simulation and the transactions. Furthermore, we drop the two string attributes **nameDest** and **nameOrig** because they are unique values which have no relationship to any other attributes and, thus, is not important. And, the attribute **isFlaggedFraud** has been removed, which has no impact to our model.

4. Big Data Predictive Analysis

Fig. 1 shows that the Spark Hadoop cluster can grow by adding more servers while collecting more data. *Gupta* and *Purush* et al. showed that the Big Data architecture is linearly scalable [7, 8].

Fig 1. Spark Hadoop Big Data UDA platform and the scalability

Apache Spark is a distributed parallel and cluster computing systems and supports MLlib machine learning APIs. Spark provides a unified data analytics (UDA) platform. UDA platform is an integrated system for data storage, analysis, and prediction, especially for massive datasets.

For classifying and detecting the fraud in financial data set, we consider three traditional machine learning algorithms: Logistic Regression, Decision Tree and Random Forests.

For deep learning, we adopt *feed forward* neural network, which produces many popular Convolution Neural Networks (CNN). It composes the neural network by copying the connectivity patterns of the neurons from the animal's visual cortex.

As Deep Learning grows popular, it has had many different architectures to integrate Spark and Deep Learning, for example, *DeepLearning Pipeline* for Apache Spark by Databricks, *TensorFlowOnSpark* by Yahoo, *BigDL/Analytics Zoo* by Intel.We adopt the *BigDL* architecture by Intel, which is integrated into *Analytics Zoo* using AWS EMR Big Data cloud service in the architecture of **Fig. 1**.

5. Comparing Models and Experimental Results

Table 1 shows the experimental result of our four classification models in Spark cluster: Decision Tree (DT), Random Forest (RF), Logistic Regression (LR), and Feed Forward (FF).

Table 1. Comparison of Classification Models

Model	Precision	Recall	Computing Time (mins)
DT	0.946	0.889	29
RF	0.959	0.909	53
LR	0.902	0.655	24
FF	0.880	0.938	51

RF has the best *Precision*: 0.959, with the least number of False Positive (FP) and even more True Positive (TP). And, FF has the best *Recall*: 0.938, with the least False Negatives (FN) and even more TP. *Recall* is more important for our fraud detection because the prediction should have the least FN.

Table 1 also shows that LR has the shortest computing time: 24 minutes, while DT has 29 minutes while RF and FF predict it in 53 and 51 minutes respectively.

6. Conclusions

We investigated a dataset containing fraudulent and non-fraudulent transactions to predict frauds. Since the dataset was about 470 MB, which takes several hours to predict using traditional systems.

We adopt big data technologies using Amazon AWS with Spark ML to compute the entire data set. In Big Data cluster using Spark ML and DL, the Random Forest Classifier scored the best **precision** accuracy with 0.959 and **recall** with 0.909. The Feed Forward DL Classifier scored the best **recall** accuracy with 0.938. The Logistic Regression Classifier scored the quickest **computing time** with 24 minutes.

Our RF and FF models should be acceptable in predicting fraudulent transactions with the computing times, 51 and 53 minutes respectively, comparing to tens of hours in the traditional systems. And, FF model should be more preferable as it has the best **recall** even though it has the worst accuracy in **precision**.

References

[1] Priyanka Purushu, Niklas Melcher, Bhagyashree Bhagwat, Jongwook Woo, "Predictive Analysis of Financial Fraud Detection using Azure and Spark ML", Asia Pacific Journal of Information Systems (APJIS), VOL.28, NO.4, December 2018, pp308~319

[2] Woo , Jongwook, & Xu , Yuhang (July 18-21, 2011), Market Basket Analysis Algorithm with Map/Reduce of Cloud Computing, The 2011 international Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA 2011), Las Vegas.

[3] Kamaruddhin, SK., & Ravi, Vadlamani (2016). Credit Card Fraud Detection using Big Data Analytics: Use of PSOAANN based One-Class Classification. ICIA-16 Proceedings of the International Conference on Informatics and Analytics 2016. Article No. 33

[4] Lopez-Rojas, E. A., Elmir, A., & Axelsson, S. (2016). PaySim: A financial mobile money simulator for fraud detection. The 28th European Modeling and Simulation Symposium-EMSS

[5] Meng, X., Bradley, J., Yavuz, B., Sparks, E., Venkataraman, S., Liu, D., Freeman, J., Tsai, D.B., Amde, M., Owen, S., & Xin, D. 2015. MLlib: Machine learning in apache spark. arXiv preprint arXiv:1505.06807

[6] Woo, Jongwook, Market Basket Analysis Algorithms with MapReduce, DMKD-00150, Wiley Interdisciplinary Reviews Data Mining and Knowledge Discovery, Oct 28 2013, Volume 3, Issue 6, pp445-452, ISSN 1942-4795.

[7] Gupta N, Le H, Boldina M, Woo J (2019) Predicting fraud of AD click using Traditional and Spark ML. KSII The 14th Asia Pacific International Confer-ence on Information Science and Technology (APIC-IST), pp24-28

[8] Purushu P, Melcher N, Bhagwat B, Woo J (2018) Predictive Analysis of Finan-cial Fraud Detection using Azure and Spark ML. Asia Pacific Journal of In-formation Systems (APJIS), VOL.28 | NO.4 | December 2018, pp308~319