Asia Pacific Journal of Information Systems
Vol. 28 No. 4 (December 2018), 308-319

ISSN 2288-5404 (Print) / ISSN 2288-6818 (Online)
https://doi.org/10.14329/apjis.2018.28.4.308

Predictive Analysis of Financial Fraud Detection using Azure and Spark ML

Priyanka Purushu[a], Niklas Melcher[b], Bhagyashree Bhagwat[c], Jongwook Woo[d,*]

[a] *Big Data Analyst, AT&T, USA*
[b] *Student, California State University, Los Angeles, USA*
[c] *System Analyst, Los Angles Metro, USA.*
[d] *Professor, California State University, Los Angeles, USA*

ABSTRACT

This paper aims at providing valuable insights on Financial Fraud Detection on a mobile money transactional activity. We have predicted and classified the transaction as normal or fraud with a small sample and massive data set using Azure and Spark ML, which are traditional systems and Big Data respectively. Experimenting with sample dataset in Azure, we found that the Decision Forest model is the most accurate to proceed in terms of the recall value. For the massive data set using Spark ML, it is found that the Random Forest classifier algorithm of the classification model proves to be the best algorithm. It is presented that the Spark cluster gets much faster to build and evaluate models as adding more servers to the cluster with the same accuracy, which proves that the large scale data set can be predictable using Big Data platform. Finally, we reached a recall score with 0.73, which implies a satisfying prediction quality in predicting fraudulent transactions.

Keywords: Fraud Detection, Spark, Azure, Machine Learning, Hadoop, Big Data

Ⅰ. Introduction

Financial frauds can be a devastating issue with extensive ramifications on any business, finance industry, corporate and government segments and for individual consumers ("Financial Transactions & Fraud Schemes", n.d.). With technological advancements, these transaction frauds are becoming more intricated. Today 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.

Big Data is defined as non-expensive frameworks, mostly on 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 well or expensive using traditional computing systems

*Corresponding Author. E-mail: jwoo5@calstatela.edu Tel: 13233432916

(Woo and Xu, 2011; Woo and Xu, 2013).

While carrying research on this topic, we encountered challenges in finding a dataset on financial fraud detection, as these kinds of 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 (Lopez-Rojas et al., 2016). We aim at doing predictive analysis on the target value which is column "isFraud" which detects if a money transaction is a fraud or not. The dataset size is approximately 470MB and it has eleven features – it is acknowledged that the 470MB is not Giga- or Tera-bytes of massive data set. However, we would like to adopt and provide the spark big data architecture and predictive model, which is linearly scalable to compute massive data set by adding more spark nodes to the cluster with respect to the data set. In addition to this, Spark-in-memory processing in Python is much faster than the traditional sequential Python approach. We want to predict if a money transaction is a fraud or not using classification models. In this paper, we have analyzed the data with two machine learning platforms: Microsoft Azure ML and Apache Spark ML, which are sequential and distributed parallel computing respectively.

While working on this project, we underwent extensive research on similar papers about financial fraudulent activities. The most common problem we noticed was that many researchers were having hard time finding an appropriate dataset for analysis. Also, PaySim simulator fixed the problem for most researchers (Lopez-Rojas et al., 2016). We could relate our work on the same lines just like other researchers. For us, finding the dataset was not much difficult due to the availability of PaySim's synthetic dataset. Where others research paper was more focused on creating a synthetic financial dataset (Lopez-Rojas et al., 2016), ours was primarily targeted on detecting the fraudulent transactions from the synthetic dataset. Besides, traditional and Big Data Platforms - Azure and Spark ML - are adopted in this paper.

Ⅱ. Related Work

The research works related to financial fraud transaction with the implementation of various data mining techniques and machine learning algorithms is a well-studied area. Based on the extensive study conducted on various academic papers we noted some prominent differences with our paper. A hybrid architecture of Particle Swarm Optimization and Auto-Associative Neural Network for one-class classification in Spark computational framework was implemented in (Kamaruddhin and Ravi, 2016) to detect credit card fraud. They were able to achieve an accuracy of 89% which is much higher than what we achieved. However, unlike us, they worked on comparatively smaller dataset of 291.7MB in size that contains only 9 features.

Another work in this field of financial frauds is by the model proposed for credit card fraud detection system, which is aimed to improve the current risk management by adding an Artificial Immune System's algorithm to fraud detection system in (Hormozi et al., 2013). They have used the negative selection algorithm on the cloud platform such as apache hadoop and mapreduce on a dataset with 300,000 rows which is comparatively smaller than our dataset which has 6.362.620 rows. In our project we have used Spark which contains the package called MLlib. MLlib provides fast, distributed implementations of common learning algorithms, including - but not limited to: various linear models,

naive Bayes, and ensembles of decision trees for classification and regression problems; alternating least squares with explicit and implicit feedback for collaborative filtering; and k-means clustering and principal component analysis for clustering and dimensionality reduction (Meng et al., 2015). Spark is efficient at iterative computations and thus well-suited for the development of large-scale machine learning applications whereas MapReduce's scheduling overhead and lack of support for iterative computation substantially slow down its performance on moderately sized datasets (Meng et al., 2015).

Ⅲ. Financial Fraud Detection using Azure ML and Spark ML

Here is the background on which we started the paper: starting from determining the type of problem, understanding the importance of machine learning and determining the algorithms. Two main problems machine learning is trying to solve: Classification & Regression problems. Mathematically speaking, regression is a combination of multidimensional feeding and function interpolation. Regression is a statistical methodology used to reveal the relationship between one or more independent variables and a dependent variable, which is continuous-valued (Han and Kamber, 2006). With a regression problem, you are trying to find a function approximation with a minimal error deviation or cost function. In other words, regression is to predict numeric dependency – a function value, for example, price of a house – from a set of input parameters like square footage, age, number of bedrooms and so on.

Classification builds up from the training set and utilizes a model on the target set to predict the categorical labels of unknown objects to distinguish between objects of different classes. These categorical labels are predefined, discrete and unordered (Han and Kamber, 2006). Classification is a different type of problem which identifies group membership. That means that if you have multiple events characterized by input parameters, which can be labelled differently, and you want your system to predict which label should be used, this is the classification problem. Let us consider an example of spam filters, emails in your inbox are processed by the machine learning algorithm. And if some criteria are met, emails are labelled as spam.

Machine learning is a fascinating topic as it incorporates substantial parts of different fields – statistic, artificial intelligence theory, data analytics and numerical methods. In simple words, machine learning is an application that can improve its prediction results with successive iterations.

For classifying and detecting the fraud in financial data set, we consider two algorithms: Decision Tree and Random Forests. Decision tree is an analytical tool which supports decision making by including event outcomes or their possible consequences. Predictions are represented by leaves and the conjunctions of features by branches. Decision trees are commonly used in credit card, automobile insurance, and corporate fraud (Anuj and Prabin, 2013). Random forest can be expressed as a set of de-correlated decision trees. The example of random forest can be a data set which contains different random values and their class. Then we divide the data set into lot of subsets with random values and random classes. After the division, the algorithm decides and allocates different classes to each of the independent forest.

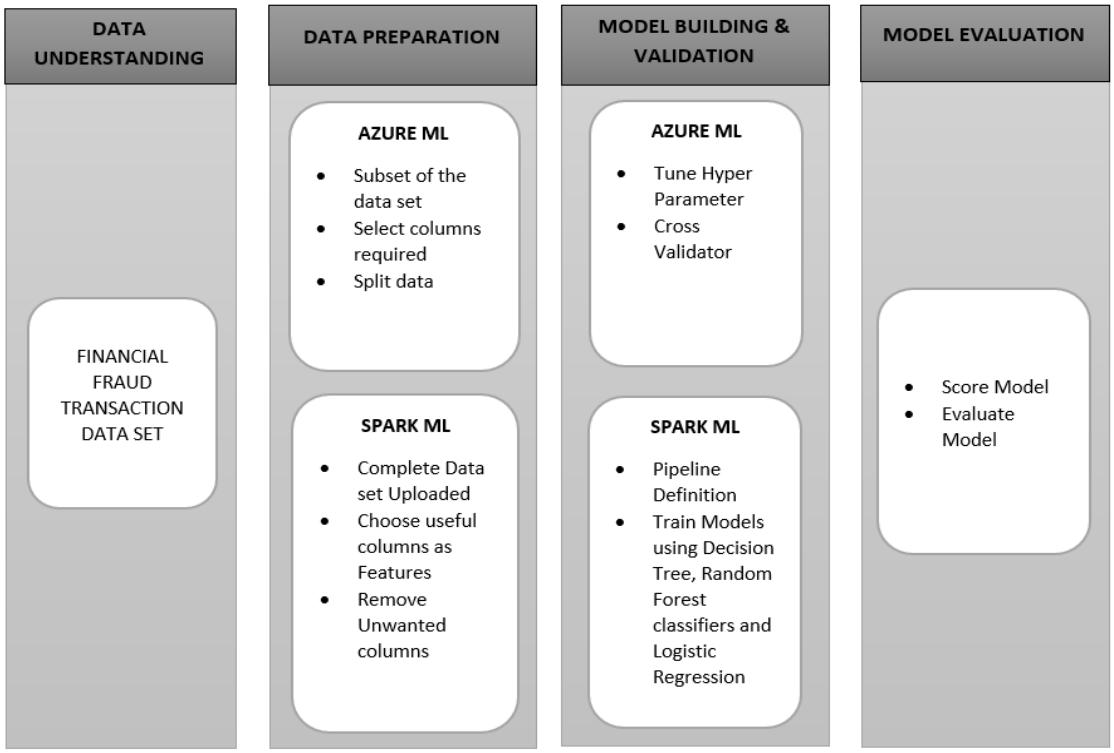
This can be used for predictive analysis as the algorithm assigns classes to each forest, and predicts the class which is repeated the most in the classification.

3.1. Method

<Figure 1> illustrates the workflow of the experimental systems to detect fraud. It is composed of four stages: Data Understanding, Data Preparation, Model Building & Validation, and Model Evaluations. Data Understanding and Preparation stages are so called data engineering process. Model Building, Validation, and Evaluation stages are known as data analysis and science processes. Additionally, we even adopted traditional and big data systems to build models to detect fraud.

One of the most pertinent stage in any data analysis and prediction is understanding the data we have. The Financial Fraud Transaction dataset was categorized into Numeric, Categorical and String attributes and a correlation was done between attributes. This is necessary to eliminate columns that are less important to our analysis such that we are left with strong contributing features. This would provide us with a better understanding on the type of machine learning algorithm that can be implemented. Data preparation was performed on Azure ML and Spark ML.

In Azure ML we have limited the data to a subset of the original data set but in Spark ML due to the high computational speed we have uploaded the complete data set. In this stage, we drop the unnecessary columns and we split the remaining data as test and train data. In Azure ML under model building and validation we pass the split data into Tune Hyperparameters – a process of tuning and parameter sweep to find the best combination and use cross validator to understand the variability of



<Figure 1> Workflow of the Systems

the dataset and the reliability of the model trained using that data.

In Spark, MLlib provides efficient functionality for a wide range of learning settings and includes several underlying statistical, optimization, and linear algebra primitives (Meng et al., 2015). Using Spark ML we define a pipeline in which we pass our features and the models to specify a machine learning flow. The final stage is to evaluate the model and determine the best model. This is based on the results of accuracy, recall and precision. All these stages are explained in detail in the later part of this paper.

3.2. Dataset

For this experiment, we use a synthetic dataset generated using the simulator PaySim (Lopez-Rojas et al., 2016) as an approach to such a problem. PaySim uses aggregated data from the private dataset to generate a synthetic dataset that resembles the normal operation of transactions and injects malicious behavior to later evaluate the performance of fraud detection methods. All in all, PaySim simulates mobile money transactions extracted from one month of financial logs from a mobile money service implemented in an African country. The original logs were provided by a multinational company, who is the provider of the mobile financial service which is currently running in more than 14 countries all around the world.

The data has a size of 470 MB with 6.362.620 rows. The dataset contains 11 attributes and the target column is 'isFraud'. A transaction can either be non-fraudulent, indicated by a 0, or fraudulent, indicated by a 1, which makes this to a binary classification problem.

Ⅳ. Attributes of the Dataset

We can list a sample row of the dataset: (1, PAYMENT, 1060.31, C429214117, 1089.0, 28.69, M15916 54462, 0.0, 0.0, 0,0). And, the attributes of the dataset with metadata has been explained in further detail below:

- **Step**: maps a unit of time in the real world. In this case 1 step is 1 hour of time. Total steps 744 (30 days simulation).
- **Type**: CASH-IN, CASH-OUT, DEBIT, PAYMENT and TRANSFER. amount: amount of the transaction in local currency.
- **nameOrig**: customer who started the transaction.
- **oldbalanceOrg**: initial balance before the transaction.
- **newbalanceOrig**: new balance after the transaction.
- **nameDest**: customer who is the recipient of the transaction.
- **oldbalanceDest**: initial balance recipient before the transaction. Note that there is not information for customers that start with M (Merchants).
- **newbalanceDest**: new balance recipient after the transaction. Note that there is not information for customers that start with M (Merchants).
- **isFraud**: This is the transactions made by the fraudulent agents inside the simulation. In this specific dataset, the fraudulent behavior of the agents aims to profit by taking control or customers' accounts and try to empty the funds by transferring to another account and then cashing out of the system. The attribute is binary, either 0 or 1.
- **isFlaggedFraud**: The business model aims to control massive transfers from one account to another and flags illegal attempts. An illegal attempt in this dataset is an attempt to transfer more than 200.000 in a single transaction.

Ⅴ. Data Structure and Correlations

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. In the next step, we need to understand the dataset and try to recognize certain patterns that would be helpful for our experiment. We can see most of the transactions are made with CASH_OUT and PAYMENT (see <Figure 2> to understand the relation between the amount of transactions grouped by the type).

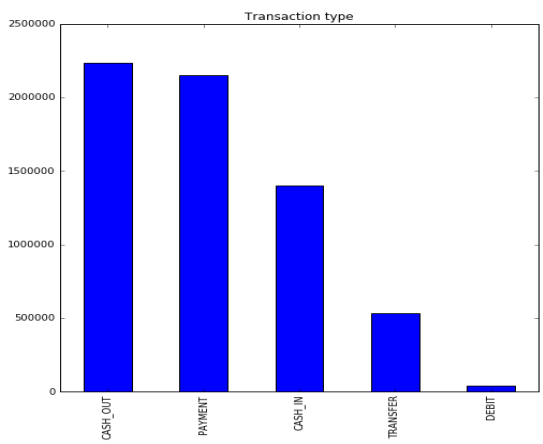
Next, we want explore which transaction types are vulnerable to fraud (see <Figure 3>). It shows clearly that fraud transactions are only made with the type CASH_OUT and TRANSFER. This is an interesting fact since the type TRANSFER is in the fourth place when it comes to the number of transactions.

Furthermore, there is an interesting attribute in the dataset called isFlaggedFraud. This attribute is supposed to flag suspicious transactions as a fraud to help the system detecting them. Unfortunately, out of the 6.362.620 rows there are only 16 transactions flagged as fraud which makes this attribute useless for our data model. This is one of the challenging task to reduce the false positive and false negative rates in order to optimize detection of fraud transactions.

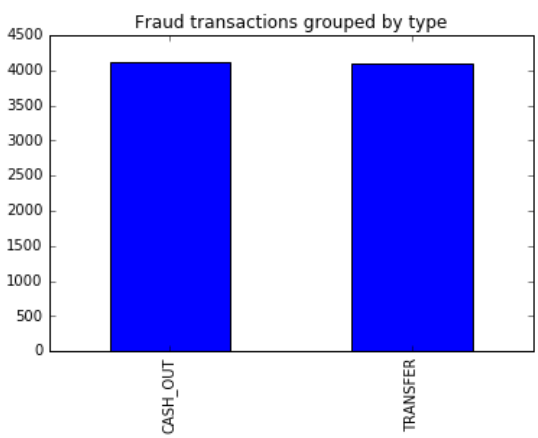
Next, we want to explore correlations between attributes which would be useful for our model (see <Figure 4> and 5 to understand relationships between newbalanceDest and oldbalanceDest, and newbalanceOrig and oldbalanceOrg, because there are strong positive correlations).

The next step is to drop useless attributes for the model (see <Table 1> – it shows the attributes we drop and the attributes we keep.)

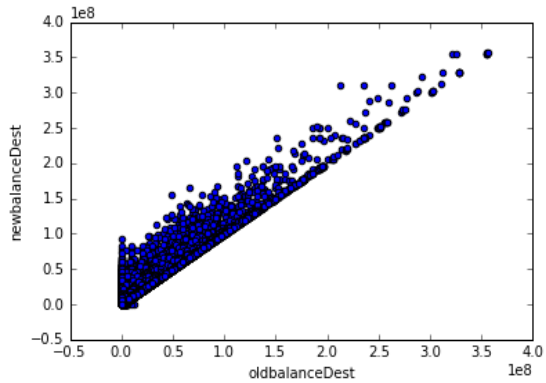
We drop the attribute step because there is no correlation between the time for the simulation and the transactions. Furthermore, we drop the two string attributes nameDest and nameOrig because they are unique values which have no relationship to any other attributes and is thus not helpful. As already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



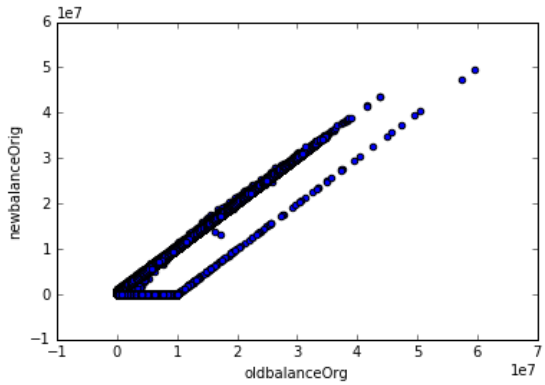
<Figure 2> Amount of Transactions Grouped by Type



<Figure 3> Fraud Transactions Grouped by Type



<Figure 4> Scatterplot between Newbalancedest and Oldbalancedest



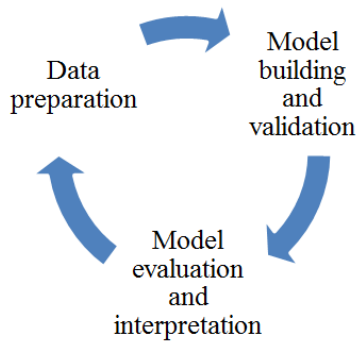
<Figure 5> Scatterplot between NewbalanceOrig and OldbalanceOrg

<Table 1> Columns to Drop and Columns to Kep from the Dataset

Columns dropped	Columns kept
step	amount
nameDest	oldbalanceOrg
nameOrig	newbalanceOrig
isFlaggedFraud	newbalanceDest
	oldbalanceDest

5.1. Experiments with the Traditional and Big Data Systems

In the first part, we build the model in Azure ML. Mainly to try a sample data set as a subset of the original data and to adopt different classification models on it. In the second part, we build our model with Spark ML in Databricks, which is a leader in developing Spark and support Spark cloud computing services. Here, we run our model with the best algorithms from the first part using a sample dataset and try to improve our result taking the whole dataset into consideration. (see <Figure 5>). First, we need to prepare our data. As already mentioned in above section, we now which columns are useful for our model and which we can drop. This step also contains the process of make attributes categorical and numeric, as we need to do this procedure with the attribute type. Additionally, the dataset must be normalized and split into a train and test set, respectively 70% and 30%. Afterwards, we build the model with different algorithms, tune the hyper-parameters and cross validate each model. Finally,



<Figure 6> Raw Description for the Workflow

we evaluate the model and interpret the results. The whole procedure is an iterative process and can be done several times before finding the best model.

There are three main metrics which are important in terms of evaluating the model. The accuracy, precision and recall. In our case, the recall is the most important metric because the aim of this classification problem is to detect fraudulent transactions. A good recall implies that the model is correct in predicting a transaction as a fraud when it is actually fraud.

Recall is the fraction of the relevant documents that are retrieved, which can be defined when TP: True Positive and FN: False Negative:

$$\frac{TP}{(FN + TP)}$$

5.2. Experiment with the Traditional Systems: Azure ML.

We took a small subset of our dataset with approximately 10,000 rows (359 KB) and tried four different classification algorithms following the procedure displayed by <Figure 5>. Overall, it was easy to realize because Azure ML is mostly a drag and drop tool with elements to configure.

Since we have a binary classification problem, we use two class classification algorithms: Two Class Logistic Regression (LR), Two Class Decision Forest (DF), Two Class Decision Jungle (DJ) and Two Class Support Vector Machine (SVM) (see <Table 2> – it shows the summarized results of the experiments).

Clearly, the DF is the best algorithm for our model since it has the highest recall score. The DJ has a good performance as well but needed approximately five times longer than the DF to calculate. Based on this results we will continue building our model in Databricks mainly with the DF.

The execution time to build and measure the models with the small data set are about 11 secs for all models. However, it takes more than 24 hours when adopting the data set of 470MB for the models.

5.3. Experiment with the Big Data: Databricks with Spark ML.

We took the whole dataset and tried three different classification models following the procedure displayed by <Figure 5>. This time we used a train validation split instead of cross validation for every model because it takes much less time to train the model with the train validation split. We used the Random Forest Classifier (RF), the Decision Tree Classifier (DT) and Logistic Regression (LR). Although the result of LR was very bad in the Azure ML experiment, we gave it another try in Databricks, because we wanted to examine the LR's performance using the whole dataset (see <Table 3> – it shows a summarized result of the experiment. We added the Receiver Operating Characteristic (ROC) with the Area Under Curve (ROC AUC) as another metric to better visualize the performance of a binary classi-

<Table 2> Results for the Small Subset with Different Classification Algorithms using Azure ML

Model	Accuracy	Precision	Recall
LR	0.991	1.000	0.100
DF	0.995	0.727.	0.800
DJ	0.997	1.000	0.700
SVM	0.993	1.000	0.300

<Table 3> Results for the Whole Dataset with Different Classification Algorithms using Spark ML

Model	ROC AUC	Precision	Recall
RF	0.860	0.927	0.719
DT	0.829	0.967	0.679
LR	0.726	0.846	0.453

<Table 4> Confusion Matrix using the RF

Confusion matrix	Predicted: NO	Predicted: YES
Actual: NO	TN = 1,905,940	FP = 78
Actual: YES	FN = 644	TP = 1,761

fier summarized in a single number.

Clearly, the RF has the best recall and ROC AUC score which indicates that this model is the best compared to the other two. To understand the results better (see <Table 4> – it shows a summarization of the confusion matrix).

The confusion matrix shows that our model is good in predicting non-fraudulent transaction when they are actual not a fraud, indicated by the high number of the true negative (TN) and small amount of false positive (FP) numbers (Specificity = 0.99). Nevertheless, when predicting fraudulent transactions, we still have some errors because the number of false negative results (FN) is still high.

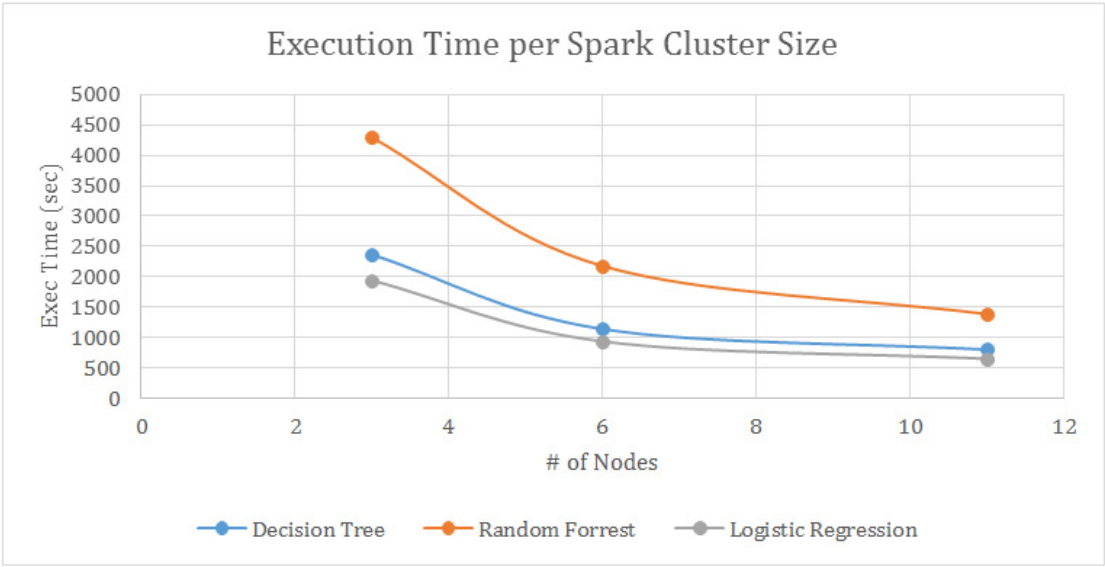
The models are built in Amazon AWS cloud computing service: EMR 12.1 (m3.xlarge) with Spark 2.2.1 on Hadoop 2.8.3 YARN with Ganglia 3.7.2 and Zeppelin 0.7.3. m3.xlarge instance is composed of Memory: 15.0 GiB, CPU: 4 vCPUs, and Storage80 GiB (2 * 40 GiB SSD). The EMR cluster is executed with 3 different number of nodes that are servers: 3, 6, 11

The execution time to build and evaluate models are listed in the chart below. It shows that the Spark cluster with 6 nodes are about 2 times faster than the cluster with 3 nodes. And, the Spark cluster with 11 nodes are about 3 times faster than the cluster with 3 nodes:

Decision Forest algorithm in Azure ML and Random Forest in Spark ML show the best result in the experiment. Random Forest – Random Decision Forest – is an extension of Decision Forest, which in Spark, it only supports Random Forest algorithm. Thus, basically both shows that random forest algorithm computes the best result.

Ⅵ. Conclusion

We investigated a dataset containing fraudulent and non-fraudulent transactions which made it to a binary classification problem. Since the dataset was about 470 MB we adopt big data technologies, Databricks with Spark ML to compute all data set while executing sample data set in the traditional Azure ML systems. We showed the model building process, both in traditional and Big Data systems respectively: Azure ML and Spark ML. For Azure ML, we use a small subset to examine the results of several different classification algorithms. The Decision Forest Classifier scored the best recall score with 0.800.



<Figure 7> Execution Time of the Models in EMR

In the next step, we use Databricks cloud computing service with Spark ML to train three different classification algorithms on the whole dataset. The Random Forest Classifier scored the best recall score with 0.719 and a specificity of 0.99. From this it can be concluded that our model is good enough in predicting non-fraudulent transaction when they are actual non-fraudulent. But we still have some errors predicting fraudulent transaction when they are actually fraudulent. This can be explained by the misbalanced data since 98.7% of our data contains non-fraudulent transactions which makes it hard to train a model properly. Nevertheless, our model can be acceptable in predicting fraudulent transactions.

Finally, the experimental results have presented with the different number of nodes in Spark clusters – 3, 6, and 11. The result shows that the accuracy and recall are the same but the performance to build and evaluate models are about 3 times faster when using 11 nodes than 3 nodes. It should be recognizable as the large scale data set can be processed with the Spark ML cluster. That is, the larger the data set, the more nodes we can add to build models in Spark ML.

Acknowledgement

This research work was supported by AWS in Education Grant award.

<References>

[1] Financial Transactions & Fraud Schemes (n.d). Retrieved from https://www.acfe.com/financial-transactions-and-fraud-schemes.aspx

[2] Hwang, K., and Wiley, John (1997). Computer Arithmetic

[3] Han, J., and Kamber, M. (2006). *Data Mining: Concepts and Techniques*, Second edition. Morgan Kaufmann Publishers.

[4] Hormozi, H., Akbari, M. K., Hormozi, E., and Javan, M. S. (2013). Credit cards fraud detection by negative selection algorithm on hadoop (To reduce the training time), *The 5th Conference on Information and Knowledge Technology*, 40-43.

[5] Jones, T. A. (2002). Writing a good paper. IEEE Trans. *On General Writing, 1*(2), 1-10.

[6] Kamaruddhin, S., and Ravi, V. (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.

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

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

[9] Sharma, A., and Panigrahi, P. K. (2013). A Review of Financial Accounting Fraud Detection based on Data Mining Techniques. *International Journal of Computer Applications* .

[10] Synthetic Financial Datasets for Fraud Detection (n.d). Retrieved from https://www.kaggle.com/ntnu-testimon/paysim1

[11] Woo , J., and Xu , Y. (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 (July 18-21, 2011).

[12] Woo, J. (2013). Market Basket Analysis Algorithms with MapReduce. *Wiley Interdisciplinary Reviews Data Mining and Knowledge Discovery, 3*(6), 445-452.

◆ About the Authors ◆

Priyanka Purushu

Ms Purushu received her MS from California State University Los Angeles. She is a big data analyst in AT&T. She has worked as a data analyst at City of Los Angeles and HiPIC center. She presented number of papers and talks about Big Data Analysis and Prediction.

Niklas Melcher

Mr Melcher was an exchange student of the graduate program at California State University Los Angeles.

Bhagyashree Bhagwat

Ms Bhagwat received her MS and BS from California State University Los Angeles and University of Mumbai respectively. She is a system analyst in Los Angles Metro. She has worked as a BI analyst at many companies.

Jongwook Woo

Dr. Jongwook Woo received his Ph.D. from USC and went Yonsei University. He is a Professor at CIS Department of California State University Los Angeles and serves as a Technical Advisor of Isaac Engineering, Council Member of IBM Spark Technology Center and as a president at KSEA-SC. He has consulted companies in Hollywood: CitySearch, ARM, E!, Warner Bros, SBC Interactive. He published more than 40 papers and his research interests include Big Data Analysis and Prediction. He awards Teradata TUN faculty Scholarship and received grants from Amazon, IBM, Oracle, MicroSoft, DataBricks, Cloudera, Hortonworks, SAS, QlikView, Tableau. He is a founder of Hemosoo Inc and The Big Link.

Submitted: July 5, 2018; 1st Revision: October 5, 2018; Accepted: November 6, 2018