International Journal of Engineering & Technology, 7 (4.32) (2018) 93-97

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

A Proposed Architecture for Real Time Credit Card Fraud Detection

Soumaya Ounacer [1]*, Soufiane Ardchir [2], Zahira Rachad [3], Mohamed Azouazi [4]

[1]Laboratoire Mathématiques Informatique et Traitement de l'Information MITI
Hassan II University, Faculty Of Sciences Ben m'Sik Casablanca, Morocco
[2] Laboratoire Mathématiques Informatique et Traitement de l'Information MITI
Hassan II University, Faculty Of Sciences Ben m'Sik Casablanca, Morocco
[3] Laboratoire Mathématiques Informatique et Traitement de l'Information MITI
Hassan II University, Faculty Of Sciences Ben m'Sik Casablanca, Morocco
[4]Laboratoire Mathématiques Informatique et Traitement de l'Information MITI
Hassan II University, Faculty Of Sciences Ben m'Sik Casablanca, Morocco
**Corresponding author E-mail: soumayaounacer@gmail.com*

Abstract

The world has known a huge evolution especially in the field of e-commerce. Owing to this technology, everything has become available to the user the thing that resulted in a significant increase of the credit card use. As a consequence, it has become the most popular tool for online payment transactions. Nevertheless, the risk of credit card transactions constitutes a major problem as it plays a crucial role in criminal activities. Thanks to e-commerce, the online payment does not require the physical card or the cardholder's presence. However, it is a two faced coin; anyone who knows the details of a certain card can easily make fraud transactions and the cardholder comes to know only after the fraud transaction is carried out. When it comes to security, all actors are affected namely the cardholder, the bank and the merchants. Hence, the urgent need for a powerful system that allows fraud transactions to be detected and processed in real time. In the work at hand, we are going to propose a system that "shields" the credit card from any kind of fraud. In other words, we intend to create a system which detects the fraud based on big data technology; precisely Apache storm and Machine Learning in order to minimize the latency and to process transactions data in real time.

Keywords: *Big data, Hadoop, Machine learning, Real time credit card fraud detection, Spark, Storm.*

1. Introduction

The global electronic commerce is in full expansion. According to a report by the United Nations Conference on Trade and Development (UNCTAD), sales from e-commerce generated a turnover of almost 25,000 billion dollars in 2015[1]. If we take Europe as an example, this growth continued to exist and resulted in sales around 510 billion euros in 2016, 598 billion euros in 2017 and is expected to reach 660 billion euros in 2018[2]. Thanks to this technology, everything has become available to the user which led to a significant increase in the use of the credit card. As a result, the credit card has become the most popular payment tool and this boost in its use has led to an increase in fraudulent activities. Online transactions have become the major cause of credit card fraud and the trend is growing at an exponential rate. As maintained by The Nilson Report, the global damage caused by fraud credit card has reached 21 billion dollars (18.4 billion euros) in 2015, a number that is expected to rise above 31 billion dollars (27 billion euros) by 2020[2]. With the growing number of users and payment transactions, the systems struggle through the overload of work. That is to say, the huge numbers of transactions are very hard to deal with; they represent a serious challenge. Therefore, there is a need for a powerful system that is capable of processing real-time data transactions with low latency.

Credit card frauds need to be detected in real time in order to quickly react to it. In this paper we will use Big data technologies and Machine learning algorithms.

Big data are based on frameworks that allow the processing of stream data with low latency and the detection of suspicious transactions in real time. Machine learning allows minimizing the time of detection of fraud by using algorithms. It also learns from historical data and understands the cardholder's habits so as to detect whether the transactions are fraudulent or not.

In the second section of this work, the variant types of credit card fraud and the basics of big data are going to be presented. The third section will be devoted to a survey of some related works done on credit card fraud detection. In the fourth section, we shall present an overview of the technologies to be used to implement the proposed architecture and a brief comparison of the real-time stream processing tool used. Last but not least, in the fifth section, we shall propose a new architecture based on Apache Storm and Machine learning so as to detect credit card fraud in real time.

2. Variant Types of Credit Card Fraud

Credit card fraud is a kind of theft or unauthorized activity to make a credit card payment in an electronic payment system. It is an illegal use of card information or card without the owner's permission[3].

Copyright © 2018 Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As shown in Figure 1, there are a many methods to commit credit card fraud, namely[4][5] ,
1) Identity theft, which is the most common one, is done though using someone's personal information or by entering the existing account.
2) False Cards, also called fake cards, are developed by skimming the actual data from genuine card that has been swiped on an EDC machine.
3) Stolen/Lost Cards are also misused if found by dishonest people or even criminals.
4) Fraud CNP is a type of fraud where the criminal requires minimal information such as card number and expiry date.
5) Clean Fraud is when the purchases are made with stolen cards and later the transactions are changed finding a way around the SDS.
6) Friendly Fraud is when the actual cardholder makes the purchases, pays for them, then files a complaint indicating the loss of the card and claims the refund.
7) Affiliate Fraud is done through making purchases using a fake account or a program that are designed to conduct fraud activities.
8) Triangular fraud which involves three main steps; creating a fake website, providing real or fake offers then using stolen or counterfeit cards to make payments.

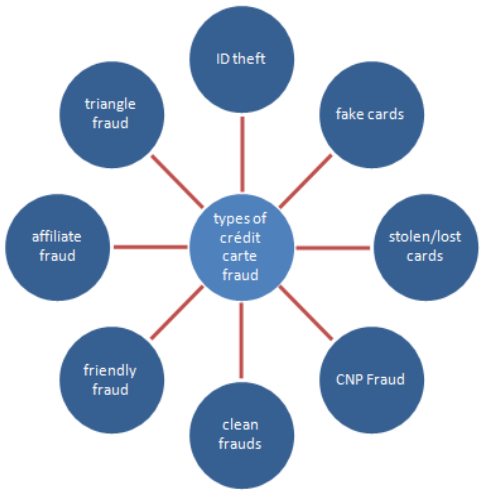

Fig. 1: Types of credit card Fraud

3. Big Data an Overview

As noted above, there are different types of credit card fraud. It is a solid proof that it has become one of the most common frauds having a huge impact on the financial industry. Financial institutions have been faced with the challenge of dealing with fraudulent transactions and detecting fraud as it occurs. There are several methods to partially remedy this problem, yet they remain limited, hence the need for Big data technology. Big data is an adequate and efficient technology that has real-time processing capability and is proving to be very useful for credit card fraud detection.
According to Gartner [6] "Big data is high volume, high-velocity and high-variety information assets that demand cost-effective, innovative forms of information processing for enhanced insight and decision making." Big data or megadata refer to all digital data produced by the use of new technologies for personal or professional purposes. This includes corporate data (e-mails, documents, databases, business processor histories, etc.) as well as sensor data, content published on the web (images, videos, sounds, texts), e-commerce transactions, exchanges on social networks, data transmitted by connected objects (electronic tags, smart meters, smartphones, etc.), geolocalized data, and many others.
By the term Big data one expects a processing of a large volume of data, of various sources in a limited time. And according to Gartner, Big data must answer a triple problem: Volume, Variety, and Velocity. [7]Volume is the important mass of data to recover, centralize, store and request. Variety is the diversity of information; what Big Data brings is the possibility of processing any type of data, in its original form, by integrating new modes of expression, measurement and interactions. Velocity, on the other hand, is a certain level of speed to reach; i.e. the exploitation of this data requires a great Velocity by a dynamic link in real time between the produced information and the operational action which results from it.[8]
In the case of credit card processing, we are faced with voluminous transactions that occur in every second representing the first V of big data Volume. The second V, which is the Variety, corresponds of the type of transactions data used. And the third V Velocity corresponds to the way used to process transactions from a high speed. The processing of these transactions requires these factors to detect fraud and find immediate solutions to this problem.

4. Related Works

Credit card fraud continues to increase and causes a significant loss in financial institutions. The need for a powerful system and adequate technologies is of crucial importance so as to solve this fraud problem. Several studies have been conducted to process, detect and prevent fraud in financial industries. Avinash Ingole et al. [9] work is based on the use of Hidden Markov Model (HMM) that models the sequence of transactions in credit card processing. A HMM is trained with Baum-welch algorithm for each cardholder. If the incoming transaction does not have a high probability of transfer, it will be considered a fraud. An HMM can detect if the transaction is fraudulent or not. The system performance is calculated using the TP (true positive) and FP (False positive) measures. The accuracy of the system is close to 75%. In another related work, Ruchi Oberoi [9] uses the Genetic Algorithm which is appropriate in credit card fraud detection as it leads to reduce the false alert and generates a successful result.
Fraud detection which is based on neural networks is also one of the most popular methods. S. Ghosh et al. [10] have implemented a neural network based system to detect fraudulent credit card transactions. This model is able to learn from the past because it is formed by all types of transactions over a period of time. Azeem Ush Shan Khan et al. [3] have developed a system on a neural network when trained with simulated annealing algorithm. But the problem is that the user's activity is different in each transaction which makes it difficult to form any ANN. Next is the clustering technique [11]which analyzes the spending behavior of the credit card user to prevent and detect fraud. In other words, when a transaction violates a certain account behavior in an unusual manner, an alarm is triggered and the transaction is declared fraudulent. These behaviors can be seen through the unusual amount of money that the cardholder isn't used on using at once, the expenses and items purchased and many others.
Another model has been proposed by K.R. Seeja and Masoumeh Zareapoor [12]falling into the same topic of credit card fraud detection. This model detects fraud from an unidentified and imbalanced credit card transactions datasets. In order to discover if the incoming transactions of the clients are legal or illegal, a parallel algorithm is proposed. Another architecture which deals with the same issue uses HDFS to store and rapidly access the logs is Anushree and Kalyani Phumamdikar's[13]. They suggested in their work that using a Bayes minimum risk classifier sheds light on better fraud detection results in the sense of higher savings.

5. Real Time Stream Processing

The big data system is based on Hadoop technology. Apache hadoop is an open source framework for data storage and processing. The heart of Hadoop is composed of two main subsystems: Hadoop Distributed FileSystem (HDFS) and MapReduce. HDFS is used for storing data and MapReduce for processing data. MapReduce processes only a finite set of data and is not convenient for real time stream processing. The system based on hadoop relies on

batch processing [14] which can give great insight into what has happened in the past; however, they do not have the capacity to process what is happening now[15]. In the context of credit card fraud detection, transactions data need to be processed in minimal time and low latency. For this reason, two systems have appeared to solve the problem of batch processing namely Apache spark and Apache storm. The upcoming section will be devoted to an overview of the two systems as well as a comparison that shall be used to build the proposed architecture.

5.1. Apache spark

Apache Spark[16] is an Open Source Big Data Processing Framework built on Hadoop MapReduce to perform sophisticated analysis and is designed for speed and ease of use. It was originally developed by UC Berkeley University in 2009 and went Open Source as an Apache project in 2010 [16]. Beside Spark's main APIs, the ecosystem contains additional libraries that allow working in the field of BigData analysis and Machine learning. These libraries include Streaming, SQL, MLlib and GraphX.
Spark Streaming [17] can be used for real-time stream processing. It is based on a micro-Batch processing mode and uses abstraction to efficiently reuse data in a large family of applications called RDD which stands for Resilient Distributed Dataset. RDDs are fault tolerant and offer parallel data structures that allow users to explicitly persist intermediate data in memory, control their partitioning to optimize data location and manipulate data using a large set of operators.
Spark has several advantages over Hadoop technology. First, Spark offers a complete and unified framework to meet the needs of Big Data processing for various data sets that different by their nature as well as by the type of source (batch or real-time). Then Spark allows applications on Hadoop clusters to run up to 100 times faster in memory and 10 times faster on disk. It allows the user to quickly write applications in Java, Scala or Python and includes a set of over 80 high-level operators. In addition, it can be used interactively to request data from a Shell command window. Finally, in addition to the"map" and"reduce" functions, Spark supports SQL queries and data flow and offers Machine Learning and graph-oriented processing features. Developers can use these capabilities separately or in combination into a complex processing chain. Thanks to the arrival of Spark for Streaming, the concept of micro-Batchs has attracted the attention of developers. This consists of splitting real-time streams to be processed into smaller processes at intervals between 500 ms and 5000 ms.
Spark implements the concept of micro-Batchs in its operation. On the other hand, it should not be considered as a real-time stream processing engine. This is indeed the biggest difference between Spark and Storm.
Despite all Spark's advantages over Hadoop, it still faces a series of limitations that prevent its access to more maturity. Among these limitations is streaming. In all the searches carried out on spark, its Streaming module is put forward as being a simple and easy to manage solution. This may be true within a very limited perimeter. Unfortunately, in reality it is very complicated to set up a real-time stream processing tunnel that runs 24 hours a day and 7 days a week. Spark patches improve this feature little by little but we are still far from a solution that is both effective and convivial. For the time being, it is preferable to adopt Storm as the solution for this specific use.

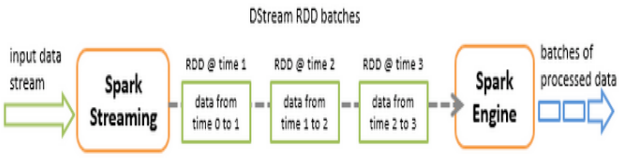

Fig.2: Spark streaming operation

5.2. Apache storm

Storm [18] is a distributed framework for reliable and real time stream processing developed by Nathan Marz in December 2010. It is derived from Hadoop providing a real-time workflow solution capable of covering a wide range of employment cases, including real-time analysis, automatic learning, continuous calculations and others. Storm also has fast processing capabilities, so that it can process one million data records per second per node within a cluster very fast and offers low latency with guaranteed data processing. It provides an architecture to build applications called Topologies. A topology is a directed acyclic graph of producers called spouts and operators called bolts that are connected with stream groupings as shown in figure 2. A stream grouping defines how an output stream should be partitioned among the bolt tasks.
In storm topologies, there is one master node which is called Nimbus and one or more worker nodes which are called Supervisor. Nimbus is responsible for assigning works to supervisors and each worker node runs a Supervisor which processes assigned tasks.
In addition, the coordination between these two entities is done through Zookeeper that is used also to give all information of supervisor's state to nimbus and managed nodes in storm cluster.

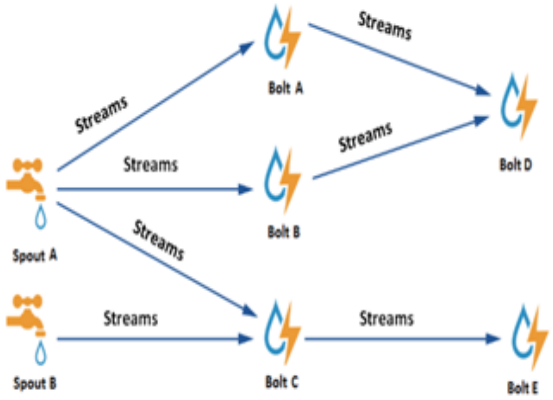

Fig.3: Topology of storm

Trident is an extension of Storm [19] developed by Twitter that provides high-level abstraction throughout real-time stream processing with distributed queries. Trident has an API to launch through an input stream, a set of operations such as filter, aggregation and grouping.
In this section both Spark and Storm projects have been described in terms of advantages over the standard version of Hadoop. Table 1 presents a comparison of the different technologies explained and their use.

Table 1: A comparison of the different Big data technologies

Functionalities	Spark Streaming	Storm
Programming languages	Java, Scala, Python	Java, Scala, Python
Stream of data	HDFS	Tuple
Processing Model	Micro-batch	Streaming
Management State	supported	supported
Reliability	One-time processing	Other modes of processing less equal to or greater than one time
Persistence	RDD	MapState
Provision	Basic monitoring	Advanced Monitoring
Resource management	YARN	YARN
Development costs	Reusability	No code reuse possible
achievable latency	A few seconds(< 1s) - More restrictions	Milliseconds (< 100ms)
Persistence to failures	Supported	Supported
Message delivery guarantee	Possible for one-time processing	Possible for one-time processing

This comparison clarifies that Apache Storm is the right tool for real-time processing especially in the case at hand: the credit card fraud.

6. Proposed Architecture

Payment transactions arrive continuously and quickly, so you need a system that can process them once they arrive. In case the transactions are not processed at the time of their production, the system will be overloaded with unprocessed data which will overcome the fraud. Our proposed architecture is based on a former work[15]. Its aim is to process the transactions that were carried out by the credit card in real time. This architecture process incoming transactions of payment with low latency. Also, the system must detect credit card fraud very quickly. For that, we must single out very efficient tools. Figure 4 represents our proposed architecture; it is divided into four main layers namely the integration layer, the real-time processing layer, the storage layer and the presentation layer. Our work focuses more on the real-time processing layer which will be based on Storm and Machine learning technologies. Storm, which is the most popular and scalable streaming processing engine, is adapted for real time. Whereas Machine learning learns from historical transactions in order to distinguish between fraudulent or normal transactions.

The scheme of our architecture is represented as follows; large data transactions come from different sources such as websites, ATM transactions, social media etc… These transactions, which come in the form of a stream, are acquired by the integration layer in real time using Apache Kafka. Kafka is a high throughput distributed messaging system which segregates the data and delivers. After being ingested, the transactions will be recovered in real time by the processing layer which is in charge of processing the payment transactions in real time and minimum latency. This layer represents two used technologies specifically storm and machine learning. The use of storm is based on the previous comparison. In other words, since Storm is based on a topology it is a network of spouts and bolts. While Spouts represent the source of the transactions, bolts contain several features. These features allow to independently calculate the fraud score, gather the fraud scores from different bolts, and decide whether the transaction is fraudulent or genuine. Also, we can apply map function in Bolt in order to mark ID card of the stream. This stream which comes from Bolt 'Map' proceeds into the following Bolt which implements the 'Reduce' function so as to aggregate together the same ID card since they correspond to the historical transactions of the same cardholder. This operation is done so as to create a model on which this work is going to be based. On the other hand, Machine learning is applied to estimate the amount of fraud and detect it as well as possible in a minimal time. Machine learning uses algorithms that learn the model from the cardholder while based on his or her habits, and allow calculating the probability in order to know whether the transaction is fraudulent or not. The characteristics that we take into consideration while controlling the cardholder history transaction are the location and time difference between two transactions, the number of transactions per day, the average of transaction amount per day, the transaction currency and many others. After the processing stage, the data will be stored in HBase and used for reporting and visualization. The proposed architecture is a continuous process to understand normal and abnormal cardholder behaviors so as to act in real time.

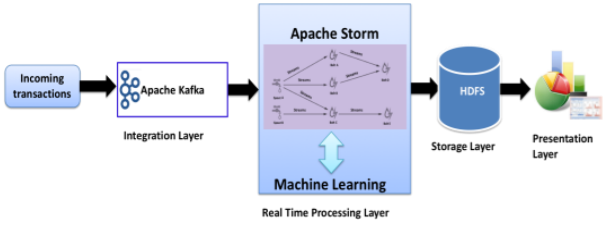

Fig.4: Proposed Architecture

7. Conclusion

Credit card fraud has increased exponentially in recent years. Accordingly, one of the main tasks of the financial industries is to develop an accurate and easy credit card fraud detection system. There are different categories of transaction scams as well as several methods to detect them. However, all these methods remain insufficient. Therefore, Big data technology helps overcome the problem of credit card fraud by identifying possible fraud attempts, suspicious activities, and studying historical data to predict fraud. Our work's ultimate aim is to provide a strong architecture that permits the credit card fraud detection in real time. For these reason, we based our study on storm technology and machine learning.

Apache storm is an effective technique that helps implement a credit card fraud detection system. It has the ability to work with large amounts of real-time transaction data and helps reduce processing time to milliseconds. The choice of storm was not arbitrary, it was mainly based on the comparison of the two technologies namely Spark streaming and Storm. These two are popular scalable streaming processing engines. Taking into consideration the scalability constraint and low delay, Storm is the one which can process a tuple at once and reduce the delay in milliseconds while spark processes incoming data in mini-batch mode and the delay can be in seconds. Machine learning, on the other hand, learns continuously from new coming transactions of data which facilitates processing. In addition to that, it generates a model that is built to classify future transactions be it fraudulent or not based on each cardholder's transaction history. This proposed architecture is in the implementation and validation phase.

References

[1] "E-commerce _ International _ le commerce électronique, nouveau thème des stratégies de développement - Le Moci - Actualité du Moci." .

[2] T. H. E. Leading, P. Covering, and P. Systems, "FAST," no. 1109, 2017.

[3] A. Khan, N. Akhtar, and M. Qureshi, "Real-Time Credit-Card Fraud Detection using Artificial Neural Network Tuned by Simulated Annealing Algorithm," nt. Conf. Recent Trends …, 2014.

[4] Rajeshwari U and B. S. Babu, "Real-time credit card fraud detection using Streaming Analytics," 2016 2nd Int. Conf. Appl. Theor. Comput. Commun. Technol., pp. 439–444, 2016.

[5] A. Delamaire, "Credit card fraud and detection techniques : a review Title Credit card fraud and detection techniques : a review Credit card fraud and detection techniques: a review," Banks Bank Syst., vol. 4, no. 2, 2009.

[6] Gartner Inc., "What Is Big Data? - Gartner IT Glossary - Big Data," Gartner IT Glossary. p. 1, 2013.

[7] A. Z. Santovena, "Big Data : Evolution , Components , Challenges and Opportunities by," p. 126, 2013.

[8] C. L. Philip Chen and C. Y. Zhang, "Data-intensive applications, challenges, techniques and technologies: A survey on Big Data," Inf. Sci. (Ny)., vol. 275, pp. 314–347, 2014.

[9] A. Ingole and R. C. Thool, "Credit Card Fraud Detection Using Hidden Markov Model and Its Performance," Int. J. Adv. Res. Comput. Sci. Softw. Eng., vol. 3, no. 6, pp. 626–632, 2013.

[10] K. K. Tripathi and M. A. Pavaskar, "Survey on Credit Card Fraud Detection Methods," Int. J. Emerg. Technol. Adv. Eng., vol. 2, no. 11, p. 721, 2012.

[11] R. J. Bolton, D. J. Hand, and D. J. H, "Unsupervised Profiling Methods for Fraud Detection," Proc. Credit Scoring Credit Control VII, pp. 5–7, 2001.

[12] K. R. Seeja and M. Zareapoor, "FraudMiner: A Novel Credit Card Fraud Detection Model Based on Frequent Itemset Mining," Sci. World J., vol. 2014, pp. 1–10, 2014.

[13] A. Naik, K. Phulmamdikar, S. Pradhan, and S. Thorat, "Real Time Credit Card Transaction Analysis," vol. 1, no. 11, pp. 1663–1666, 2016.

[14] "MapReduce." [Online]. Available: https://fr.hortonworks.com/apache/mapreduce/.

[15] S. Ounacer, M. A. Talhaoui, S. Ardchir, A. Daif, and M. Azouazi, "A New Architecture for Real Time Data Stream Processing," Int. J. Adv. Comput. Sci. Appl., vol. 8, no. 11, pp. 44–51, 2017.

[16] W. Is et al., “Welcome to ApacheTM HadoopTM!,” Innovation, no. November 2008. pp. 2009–2012, 2012.
[17] Apache Spark, “Apache SparkTM - Lightning-Fast Cluster Computing,” Spark.Apache.Org. 2015.
[18] Apache Storm, “Storm, distributed and fault-tolerant realtime computation,” Storm.Apache.Org. 2015.
[19] L. Wall et al., “simple easy learnin Storm,” p. 2, 2015.