Megha Tatti **A20427027**

Term Research Paper-CSP554

Topic: "Big Data Technologies: A Review of the techniques for Credit Card Fraud Detection"

Abstract

People tend to make multiple transactions daily. There are several modes of transactions amongst those credit card-based transactions that are of great variety. With the advent of newer and advanced methodologies, illicit use of the credit card-based system has been growing. Due to this, many banking and financial industries face severe cyber fraud challenges like credit card frauds. The fraudulent transactions are scattered with genuine transactions because of which easy pattern matching techniques are insufficient to sight them accurately. To beat this, we have Big Data that gives better methodologies and algorithms. Big data helps in building an analytical model that can be integrated with Hadoop for storage. This will further be feasible to implement pattern recognition algorithms along with the help of a couple of machine learning algorithms to predict fraudulent patterns.

Here in this paper, the discussion will be regarding the challenges faced by the current fraud detection systems and then about numerous strategies and Big Data tools utilized in credit card fraud detection and compare data mining vs big data techniques for fraud detection. Also, explore numerous techniques like Apache Hadoop, MapReduce, Apache Spark, and Apache FLINK that help in fraud detection and compare each technique with one another. This paper reflects the various techniques for better accuracy rates in fraud detection in comparison to the other existing techniques.

Introduction

These days, credit cards have comparatively cut down the trouble concerned in making transactions. Moreover, online credit card transactions have made the task far easier. The credit card payments scale back the complexness of the payment system by eliminating physical paper in use as cash or cheque. The transaction value has nearly doubled within the last few years. The growth in credit card transactions additionally attracts the eye of fraudsters. Fraudsters have become more adept and creative in innovating new strategies to commit fraud frequently. Therefore, quick fraud detection and remedy are vital for maintaining a good relationship between the bank and the client. Different algorithms are used to verify the likelihood of fraud by analyzing purchasing habits and comparing every transaction with what preceded it. Big data analytics is one of the most effective techniques which may show relationships among fraudulent activities, together with several uncertain or distrust activities in a single account or patterns of comparable activities in numerous accounts. It helps to provide the analysis result quickly and accurately.

The different kinds of credit card frauds are as in fig 1.1 [2]:

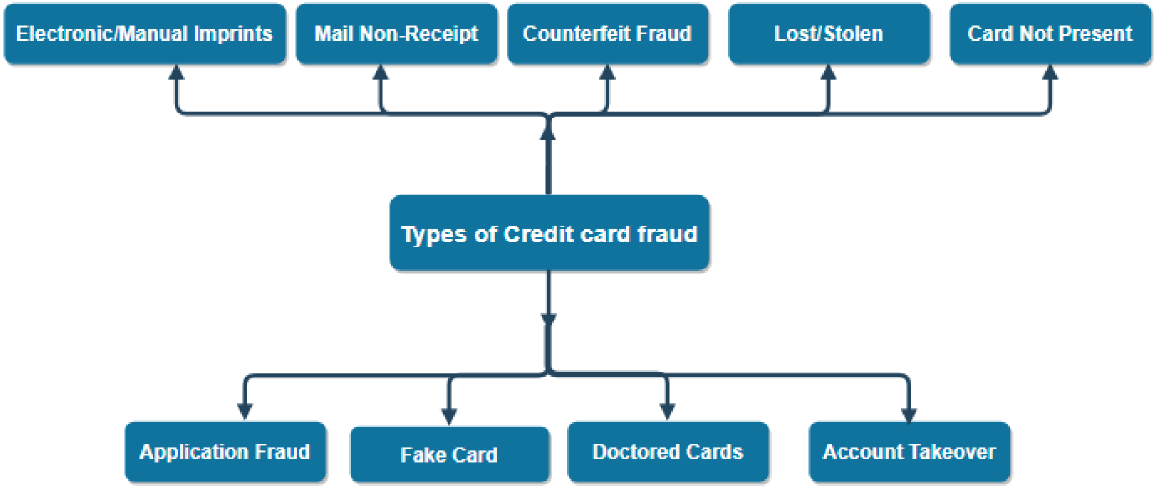

Fig 1.1 Various Types of Credit Card Fraud

1. Application Fraud: This type of fraud occurs when a fraudster uses another person's personal information to apply for credit or a new credit card. It happens by stealing the supporting documents followed by using them to authenticate their fraudulent application.

2. Fake Card: In this case, fraudsters are skilled in forge duplicate cards suing fake names and account numbers and then make transactions.

3. Doctored Cards: This type occurs when fraudsters manage and modify the details of the card itself by using a strong magnet to erase the metallic stripe.

4. Account Takeover: Fraudsters gather all required information regarding the credit card and the cardholders to contact the card issuer and pretense as a genuine user and request modifying the billing address or report a lost card and request a replacement.

5. Electronic or Manual Credit Card Imprints: This happens when a single transaction is recorded multiple times on credit card imprint machines.

6. Mail Non-Receipt Card Fraud: This is also known as never received issue or intercept fraud.

7. Counterfeit Card Fraud: This happens through skimming. Fake magnetic swipe cards are used that hold all the card details and then fully functional fraudulent cards are created.

8. Lost or Stolen Card Fraud: This happens when a card is stolen, and the thief uses it for his purchases.

9. CNP (Card Not Present) Fraud: This happens when someone knows the expiry date and account number of your card. They can easily commit to CNP fraud.

The above are a few credit card frauds. One of the best ways to tackle these problems is through Big Data Technologies.

What is Big Data?

By the definition of Gartner, circa 2001- "Big data is data that contains greater variety arriving in increasing volumes and with ever-higher velocity"[4]. The origins of enormous data sets go back to the 1960s and 1970s when the world of data was simply getting started with the primary data centers and therefore the development of the relational database. Big data can help in addressing a range of business activities, from customer experience to analytics. The three Vs of Big Data also known as the characteristics are Volume, Velocity, and Variety [4]. With Big Data, "Variety" of high "Volumes" of low-density, unstructured or semi-structured data will be processed with data being received/acted upon at a faster rate ("Velocity"). Big Data helps in gaining more complete answers because of the large amount of data collected which can give more confidence and a different approach to tackling problems. Big Data involves the following three key actions [4]:

- Integrate
- Manage
- Analyze

Integrate: Traditional data integration mechanisms like extract, transform, and load (ETL) usually aren't up to the task. During the integration process, the data must be brought from many disparate sources and applications, process it and format it in such a way that the analysts can get started with it.

Manage: The desired processing requirements and necessary process engines can be brought to those data sets stored either in the cloud, on-premise or both on an on-demand basis. The cloud is gradually gaining popularity.

Analyze: Analyzing or acting upon the data collected is very essential. Visual analysis can be carried out on the data sets to get new clarity. Build models using various machine learning and artificial techniques.

Credit Card Fraud Detection

Big Data helps financial establishments to approach fraud in numerous ways and probably get different results. For credit card fraud detection, we need the bank, customer data, and transactions. Some of the examples include data associated with customer transactions like account number, date, time, amount transacted, etc., data associated with cardholder like card type, number, expiration date, etc. and data associated with transaction history. The following two phases are generally used while detecting credit card fraud [2].

- Building the model with training features and labels.
- Testing the model to get prediction with test features.

The test predictions are then compared with test labels, looped until the model accuracy is satisfied. The model fitting parameters, features and/or machine learning algorithms are adjusted and the tests are repeated.

Fig 1.2 represents the generic model for the real-time fraud detection model.

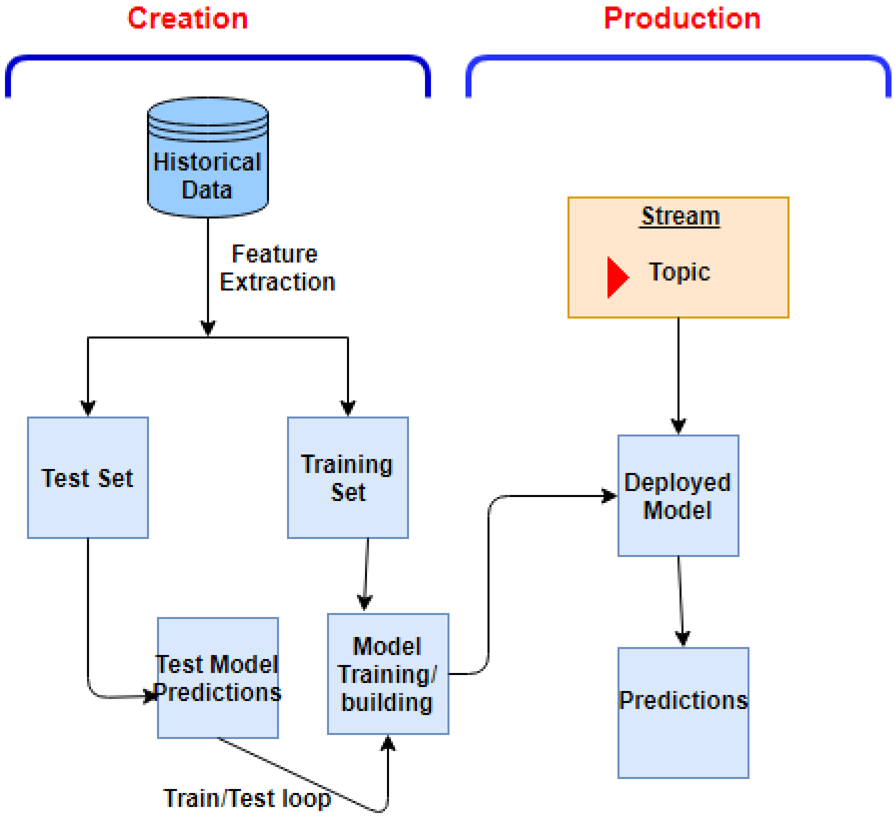

Fig 1.2 Real Time Fraud Detection Model

The following are a few techniques/methods used in credit card fraud detection. These techniques help the investigator to extract meaningful forensic evidence for detecting frauds from the large datasets [1].

- Apache Hadoop
- MapReduce
- Apache Spark
- Apache FLINK
- Pattern recognition in Fraud Detection using Big data

Apache Hadoop

Apache Hadoop is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models [5]. It was co-founded by Doug Cutting and Mike Cafarella in 2005. It was originally developed to support the Nutch search engine. Hadoop consists of the Hadoop common package. This provides the operating system and file system level abstractions (MapReduce engine and Hadoop Distributed File System (HDFS)). These components were originally derived from Google's MapReduce and

Google File System (GFS) paper respectively. The Hadoop common package also contains Java Archive (JAR) scripts and files to start Hadoop. The high-level architecture of Hadoop is as shown in Fig 1.3. [9]

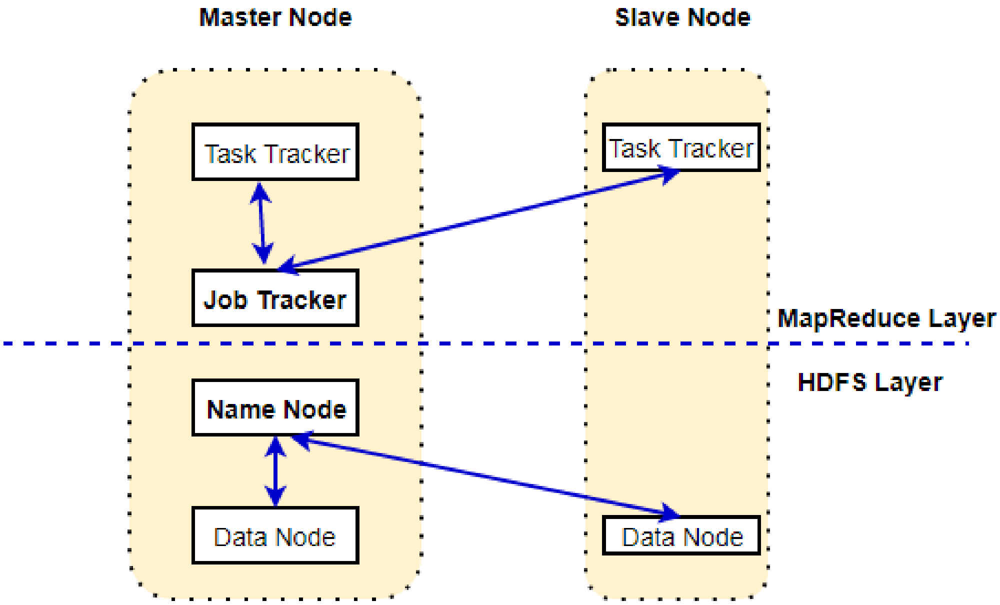

Fig 1.3 High Level Architecture of Hadoop

The key component of Hadoop is the Hadoop Distributed File System (HDFS). It is a scalable, distributed and portable file system and is written in Java for the Hadoop framework. It manages data speed across various servers. Each node in a Hadoop instance generally consists of a single name node and a cluster of data nodes that form the HDFS cluster. The file system uses the TCP/IP layer for communication. The Remote procedure call (RPC) is used by clients to communicate with each other. HDFS manages many serves in parallel [9]. Since HDFS is file-based, it doesn't require a data model to store and process data.

Hadoop can store any kind of data from any source even on a very large scale. It is capable of performing a very sophisticated analysis of data quickly and with ease. Hadoop is a powerful platform for dealing with fraudulent activities. It does that by storing all the data-message content, patterns of activity, relationships among people and computers, etc. It then runs sophisticated detection and prevention algorithms. It then creates complex models from historical data to monitor real-time activity. By monitoring real-time activity, Hadoop can detect any fraudulent activities. The advantages of using Hadoop include a distributed functionality that makes the networks more robust. If one cluster fails, it continues to run. It is efficient and provides linear scaling in the ideal case enabling easier design [1].

MapReduce

Map-reduce is a programming model for processing large data sets with a parallel, distributed algorithm on a cluster. Map-reduce, when coupled with HDFS, can be used to handle big data [6]. It is used for processing huge datasets in parallel. MapReduce uses the concept of key-value pair. Before feeding data to the MapReduce model, all the data has to be translated into a key-value pair. It consists of a Map function and a Reduce function and the computation on an input occurs in three stages namely the map stage (distributes the data), the shuffle stage (distributes the data) and the reduce stage (performs computation) [6]. Auxiliary phases like sorting, partitioning and combining values can take place between the Map and Reduce phases [1]. The following Fig 1.4 represents the working of MapReduce [10].

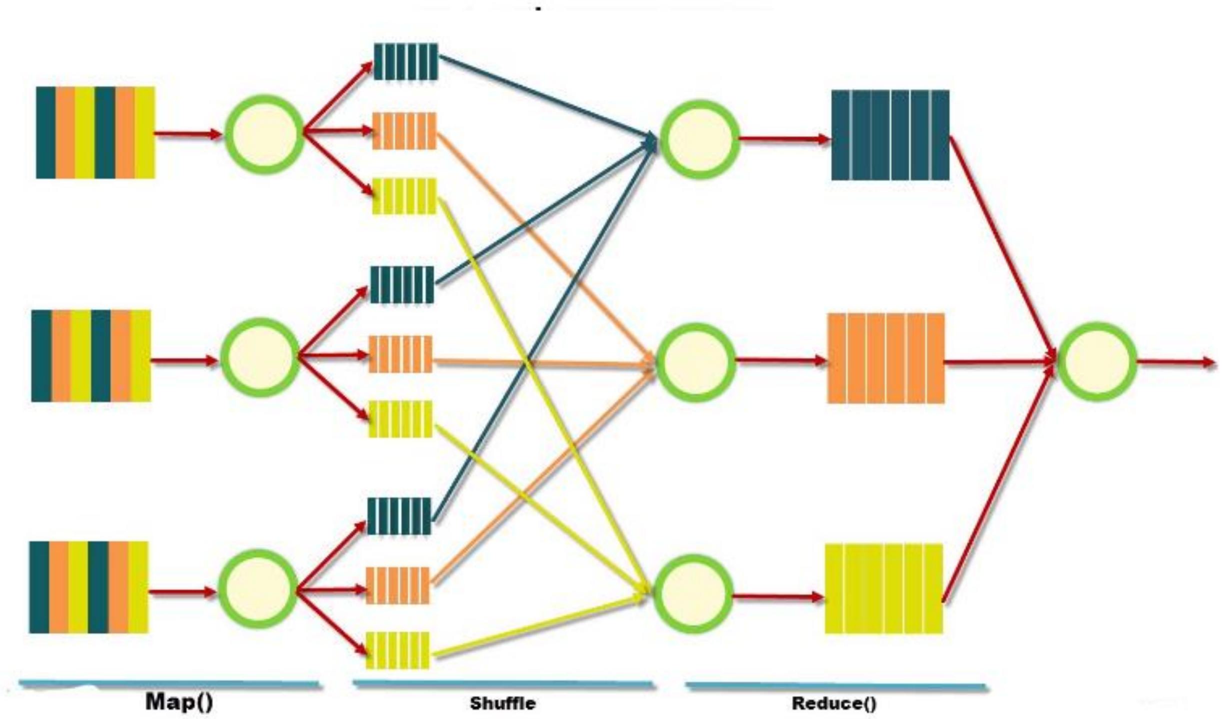

Fig 1.4 MapReduce

MapReduce extracts and transforms useful data from large unstructured/structured datasets and then distributes the data to the various servers where processing occurs. It then stores the results into a smaller and easy to analyze the file. It basically divides a task into subtasks and handles the subtasks in parallel and then aggregates the results to obtain the final output [1].

For credit card fraud detection, MapReduce makes use of transaction data as input and then translates into a key-value pair and passes it to Map Stage where the data is distributed for processing. It is then reduced by key in Reduce stage. If the convergence conditions are satisfied the output models are created. To detect fraudulent activity, comparisons are made to the original model created using genuine customer transactions [11].

Apache Spark

Apache Spark is an open-source distributed general-purpose cluster computing framework. It was originally developed at the University of California, Berkeley's AMPLab in 2009. Spark relies on Resilient Distributed Datasets (RDDs) and can be used to interactively query a large amount of data in less than a second. Apache Spark ecosystem includes Spark SQL plus Data Frames, Streaming, MLlib, GraphX and Spark Core API including R, SQL, Python, Scala, and Java. Spark SQL is a Spark module for structured data processing that enables unmodified Hadoop Hive queries to run up to 100 times faster on the existing deployments and data. While inheriting Spark's ease of use and fault tolerance characteristics, Spark Streaming enables powerful interactive and analytical applications across both streaming and historical data. MLlib is a scalable machine learning library that delivers both high-quality algorithms and tremendous speed. GraphX is a graph computation engine that enables users to interactively build, transform and reason about graph-structured data at scale [12].

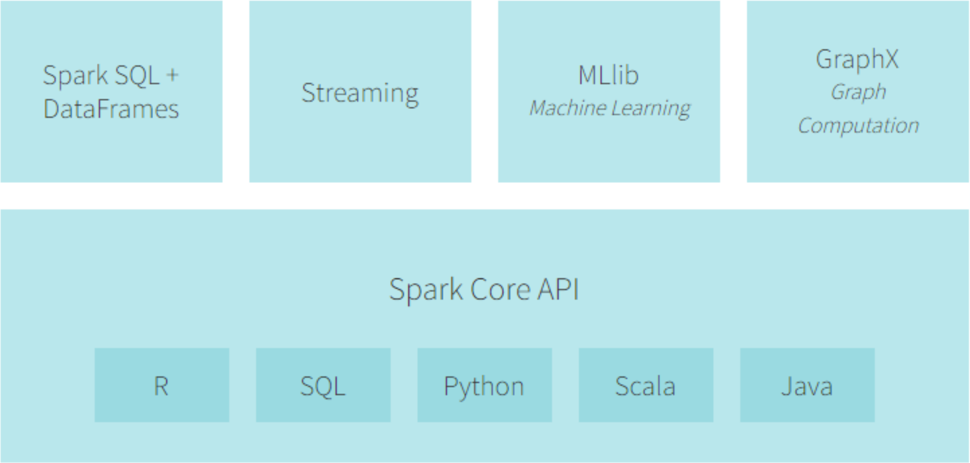

Fig 1.5 Apache Spark Ecosystem

Similar map/reduce task used in Hadoop to run analytics on large data sets are used by Spark programs with the difference being that Hadoop map/reduce run in batch mode while spark can be used to run on real-time data as well as batch mode.

Spark is used to process data or to run machine learning algorithms and then HDFS is used to store the data. Combining these 2 frameworks, we can solve credit card fraud using big data analytics. If a worker fails in spark streaming, the system can be recomputed to the lost state from the input data. This is done by following all the RDD transformations that preceded the point of failure. Advantages of Apache Spark include integrated advanced analytics, parallel processing, efficient that MapReduce, faster than Hadoop for certain cases and continuous micro-batch processing [1].

Apache FLINK

Apache FLINK is a framework and distributed processing engine for computations over unbounded and bounded data streams. Unbounded data streams have a start but no defined end and hence must be continuously processed to arrive at a particular result. Bounded data streams have a defined start and end. Ordered ingestion of data is not required in this case as they can always be sorted [8]. Flink has been designed to run in all common cluster environments, perform computations at in-memory speed and any scale. Any kind of data is produced as a stream of events. The Apache FLINK, in a nutshell, can be represented as in fig 1.6 [13].

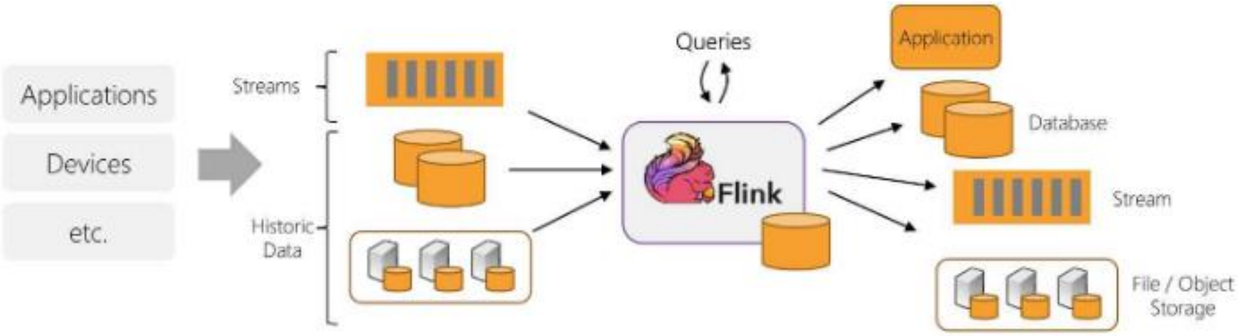

Fig 1.6 Apache FLINK

Stateful Flink applications are optimized for local state access. The task state is always maintained in memory. If the state size exceeds the available memory, then the task state is maintained in access to efficient on-disk data structures. Hence, tasks perform all computations yielding very low latencies by accessing local, often in-memory state. Flink guarantees exactly-once state consistency in case of failures. It does that by periodically and asynchronously checkpointing the local state to durable storage [8]. A checkpoint is an automatic, asynchronous snapshot of the state of an application and the position in the source stream.

Advantages of FLINK include a true stream processing framework, use of algorithms in both batch and streaming modes and aggressive optimization engine.

Pattern recognition in Fraud Detection using Big-data

Pattern recognition is a trained mechanism to recognize patterns and anomalies in a given data set using several approaches. Machine Learning is one of the approaches where the input is mapped to predefined class labels to arrive at a decision. Here, we come across two terms, supervised learning and unsupervised learning. When the pattern matching systems are trained using the previously trained data, we call it Supervised Learning and when there is no previously trained data to train the data, we call it Unsupervised Learning. The above-mentioned techniques like Apache Hadoop, Spark, MapReduce, and Flink help to obtain the pre-processed data for further pattern recognition. A learning function is generated within the system such that the input training data is mapped to the predefined class labels. Here the two class labels are fraud and non-fraud. The transaction patterns of genuine customer transactions are compared and mapped with the available pre-processed data. This mapping is usually done through Classification [3].

In this paper, the classifier we are using is Random Forest. Random Forest refers to the construction of a mass of decision trees with randomly chosen features as the root nodes. The final output is obtained by calculating the mode of the outputs derived from each tree. The random forest can deal with noise and outliers within the dataset. To build a decision tree, each tree is trained individually based on randomly chosen samples. Each decision tree is then extended based on randomly chosen features from the features present in the dataset. The working model of credit card detection is as shown in Fig 1.7 [3].

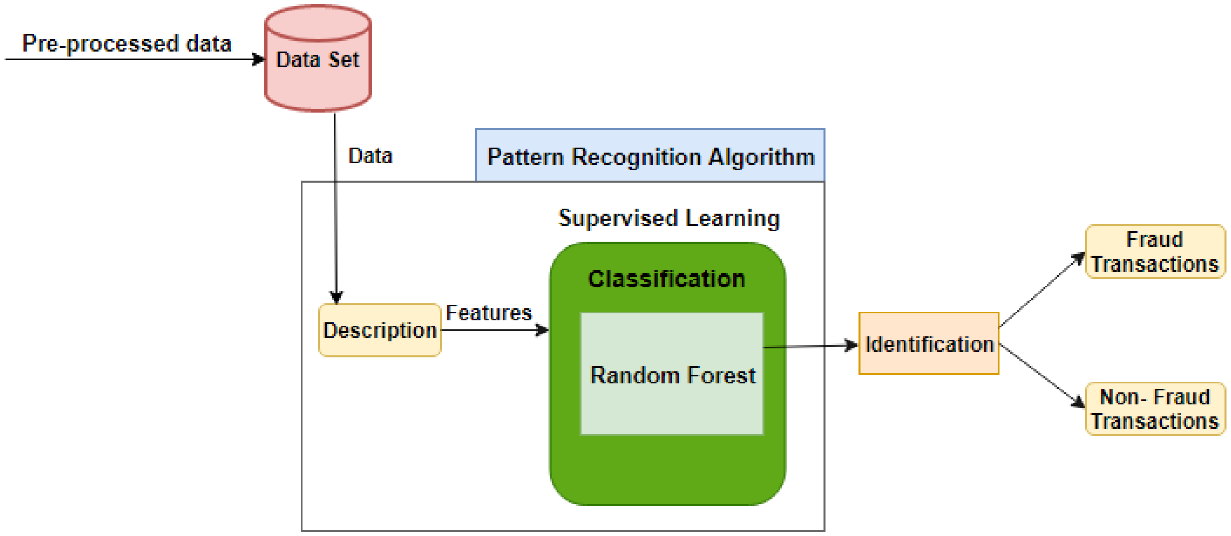

Fig 1.7 Working model of Credit Card Fraud Detection

Comparison of Big-Data Techniques

Certain factors have to be considered for the evaluation of the techniques for better selection and performance. These factors should be closely analyzed before carrying out the implementation.

The factors that need evaluation are:

- Processing Speed
- Latency
- Fault Tolerance
- Performance
- Scalability

Processing Speed

Apache Hadoop has medium processing speed. MapReduce has a slow processing speed. Apache Spark and Apache FLINK both have fast processing speed when compared with one another. To understand this better, fig 2.1 can be referred [1].

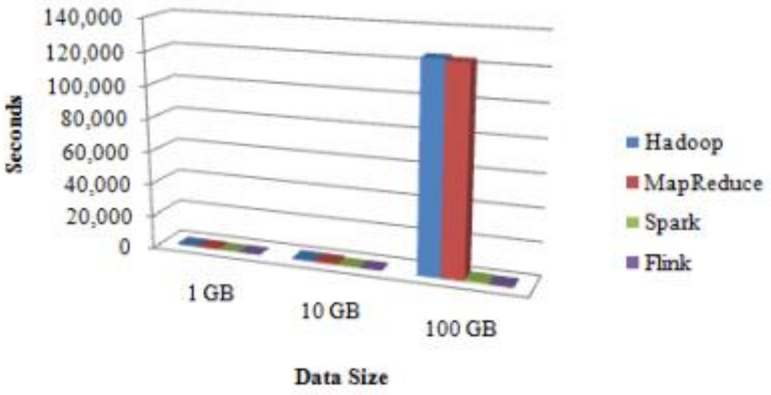

Fig 2.1 Processing Speed

From the figure above, it can be observed that the processing speed for Apache Spark and Apache FLINK remains the same throughout even with an increase of the size of datasets while the processing speed for Apache Hadoop and MapReduce increases with increase in the size of datasets.

Latency

Apache Hadoop and MapReduce both have high latency while Apache Spark and Apache FLINK have low latency comparatively. To understand this better, refer fig 2.2 [1].

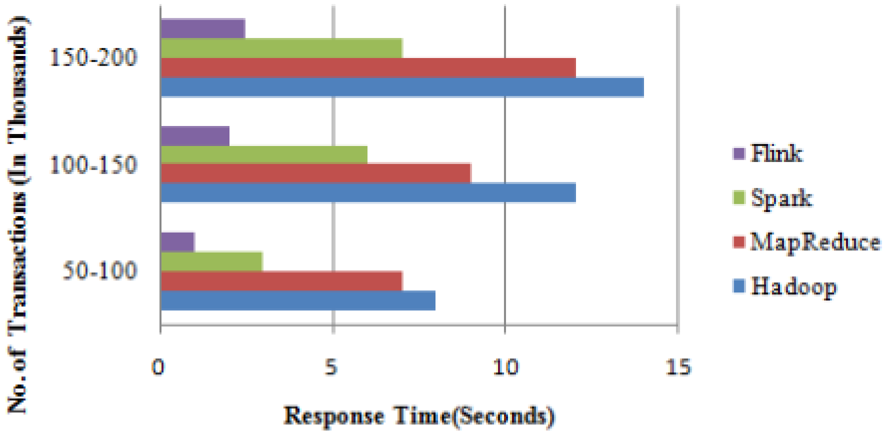

Fig 2.2 Latency

To obtain a better model, the response time has to be the lowest/minimum while executing the data or transactions. When compared with other techniques, it is clear from the graph above that Apache FLINK has low latency while processing Big Data.

Fault Tolerance

The fault tolerance for all four techniques is high but Apache Spark has a better fault tolerance system comparatively. It replicates the input data in memory because of which when the data is lost due to failure, it can be recomputed from the replicated input data.

Performance

Apache Hadoop and MapReduce have slow performance when compared to Apache Spark and Apache Flink. To understand this better, refer fig 2.3 [1].

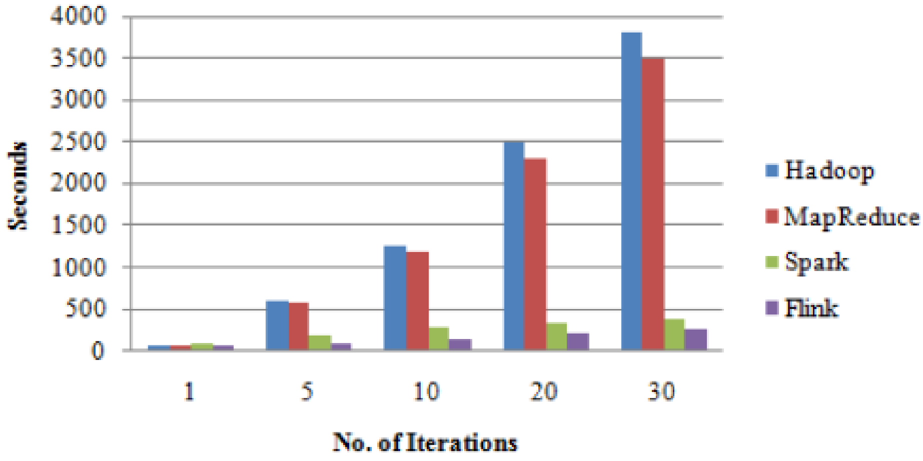

Fig 2.3 Performance

Comparing all other techniques, Apache Spark has a small variability in the execution time. So Spark is better in terms of performance.

Scalability

Apache Hadoop and MapReduce have medium scalability compared to Apache Spark and Apache FLINK which has high scalability. To understand this better, refer fig 2.4 [1].

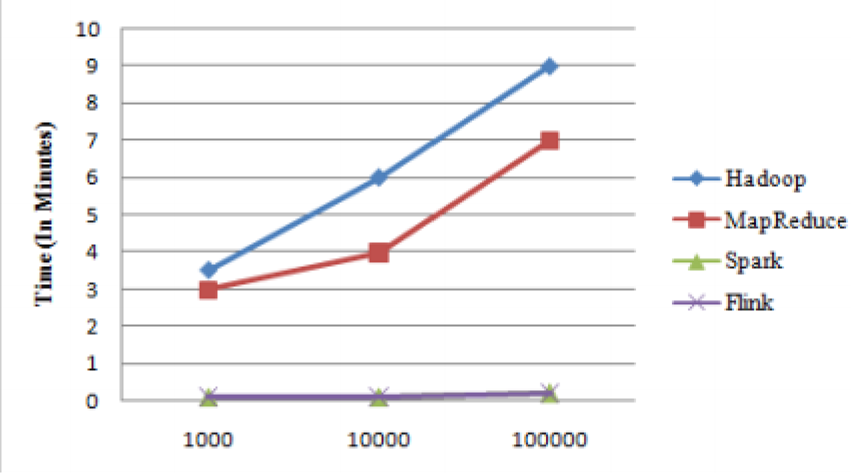

Fig 2.4 Scalability

It is clear from the figure that Apache Spark and Apache FLINK processes the data very smoothly and in less time even when the size of the data is increased while that is not the case for Apache Hadoop and MapReduce. Spark has better scalability comparatively.

Conclusion

In this paper, we discussed how credit card frauds are committed, the types of frauds. We also discussed a few big data methodologies that can be used to determine a feasible solution for the credit card fraud detection problem. Pattern recognition using the Random Forest technique gives a better understanding of how the data pre-processed by big data techniques can be used for further detection of fraudulent activities. We also compared each method with one another by evaluating the methodologies based on factors such as processing speed, latency, fault tolerance, performance, and scalability. Comparing all the techniques, Apache Spark stands out as a better and efficient model for credit card fraud detection.

References

1) https://www.ijsr.net/archive/v6i5/ART20173111.pdf
2) https://www.ijrcar.com/Volume_5_Issue_5/v5i508.pdf
3)https://www.researchgate.net/publication/332369280_Application_of_Big_Data_Analytics_and_Pattern_Recognition_Aggregated_With_Random_Forest_for_Detecting_Fraudulent_Credit_Card_Transactions_CCFD-BPRRF (the pdf version of the full paper).
4) https://www.oracle.com/big-data/guide/what-is-big-data.html
5) https://hadoop.apache.org/
6) https://www.analyticsvidhya.com/blog/2014/05/introduction-mapreduce/
7) https://en.wikipedia.org/wiki/Apache_Spark
8) https://flink.apache.org/flink-architecture.html
9) https://opensource.com/life/14/8/intro-apache-hadoop-big-data
10) https://developerzen.com/introduction-to-mapreduce-for-net-developers-1030e070698a

11)https://www.google.com/imgres?imgurl=https%3A%2F%2Fd3i71xaburhd42.cloudfront.net%2Fbeb1915b2e7cea34dfe47945bdf52e2472f3694f%2F5-Figure4-1.png&imgrefurl=https%3A%2F%2Fwww.semanticscholar.org%2Fpaper%2FOnline-Credit-Card-Fraud-Detection%253A-A-Hybrid-with-Dai-Yan%2Fbeb1915b2e7cea34dfe47945bdf52e2472f3694f%2Ffigure%2F3&docid=kz9yZOLMa5RpAM&tbnid=oIpS8DUY-Fo-OM%3A&vet=10ahUKEwi1o5jC5ZnmAhVS11kKHdRcByEQMwhRKBkwGQ..i&w=434&h=466&safe=active&bih=881&biw=1280&q=credit%20card%20fraud%20detection%20method%20using%20mapreduce&ved=0ahUKEwi1o5jC5ZnmAhVS11kKHdRcByEQMwhRKBkwGQ&iact=mrc&uact=8

12) https://databricks.com/spark/about

13)https://www.google.com/imgres?imgurl=https%3A%2F%2Fres.infoq.com%2Fpresentations%2Fflink-stateful-streaming%2Fen%2Fslides%2Fsl6-1525305347260.jpg&imgrefurl=https%3A%2F%2Fwww.infoq.com%2Fpresentations%2Fflink-stateful-streaming%2F&docid=T6kgZidxFwJevM&tbnid=sulJ7dExJNwP6M%3A&vet=1&w=1680&h=945&safe=active&bih=881&biw=1280&ved=2ahUKEwiQ3PfB9ZnmAhVGrZ4KHWjkBLgQxiAoAnoECAEQGw&iact=c&ictx=1