Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 7.056

IJCSMC, Vol. 12, Issue. 12, December 2023, pg.62 – 78

DETECTION OF CREDIT CARD FRAUD IN REAL TIME USING SPARK ML

Dr. Nwankwo, Ugochukwu. Cornelius
Lecturer, Department of Computer Science, Caritas University, Amorji Nike Enugu, Enugu State, Nigeria

Dr. Onuora, Josephone Nneka
Lecturer, School of General Studies, Caritas University Amorji-Nike, Enugu State, Enugu State, Nigeria

Dr. Obi, Justina Nwamalubia
Lecturer, Department of Mass Communication, Caritas University Amorji-Nike, Enugu State, Enugu State, Nigeria

Dr. Elizabirth, Nkechi Obiukwu
Lecturer, Department of English, Caritas University Amorji-Nike, Enugu State, Enugu State, Nigeria

Uche, Ezekiel Okore
Lecturer, Department of Economic, Caritas University Amorji-Nike, Enugu State, Enugu State, Nigeria

Correspondence Email Address: nwankwo.ugochukwu@caritasuni.edu.ng; systemgburugburu@gmail.com

DOI: https://doi.org/10.47760/ijcsmc.2023.v12i12.006

ABSTRACT: *Assume you have access to a credit card. Your prior spending habits will be investigated. For example, how much money you spend, where it is spent, how frequently it is spent, and what you buy. If your current credit card transaction differs from your historical spending patterns, it is suspected of fraud; otherwise, it is considered as a genuine transaction, and fraud transactions are warned in the dashboard. The predictions are expected to be based on millions of transactions. As result, distributed frameworks that can expand as the number of transactions grows are used. Spark this system for real-time identification of credit card theft is built with Kafka and Cassandra. Preprocessing is accomplished through the use of Spark Machine Learning Pipeline Stages such as String Indexer and Vector.*
Keywords: *Kafka, Cassandra, Stream processing, Transaction monitoring, Anomaly detection*

I. INTRODUCTION

There have always been those who would develop new ways to illegally access an individual's funds since payment methods were invented. This is a huge issue in the present world, because all purchases can be readily accomplished online by just inputting your credit card information. Chuprina et al., 2021). Fraud is as ancient as humanity and may take on an infinite number of various forms (Lakshmi, Mettu, & Hameed, 2020). As a result, detecting fraud has become a crucial and pressing responsibility for organisations. To prevent this fraud, relevant

© 2023, IJCSMC All Rights Reserved, ZAIN Publications, Fridhemsgatan 62, 112 46, Stockholm, Sweden

preventive steps should be carried out, and the behaviour of such fraudulent activities can be researched to minimise it and defend against such occurrences in the future. The remedies to the scam can be classified as follows: • Credit Card Frauds: Online and Offline •Card Theft • Account Bankruptcy • Device Intrusion • Application Fraud •Counterfeit Card •Telecommunication Fraud.

The contemporary era's fast technological advancement has increased desire for more creative payment solutions. Previously, payment methods such as checks and cash were used. Credit card use has skyrocketed over the world, and many individuals now believe in being cashless and relying only on online purchases. Digital transactions have grown simpler and more accessible as a result of debit cards. The theft of credit cards happens whenever a financial card, and such as a credit card as well as a debit card, was used for the purpose of fraud. The purpose might be to deposit funds into another criminally controlled account or to receive goods or services. Credit card fraud can occur when a genuine customer uses their own credit card to make a payment to a merchant.

There are two kinds of card fraud:

- card-present fraud
- card-not-present fraud

The compromise can occur in several ways and can usually occur without the knowledge of the cardholder. A compromised account's credentials may be kept by a fraudster for months prior to any theft, making it challenging to pinpoint the point of penetration. Stolen cards can be reported swiftly by cardholders. Unauthorized use might not be discovered by the cardholder until they get a statement. Until the cardholder contacts the issuing bank and the bank places a block on the account, the credit card can be used for unauthorized purchases. For early reporting, most banks offer free, 24-hour telephone numbers. Even so, before the card is cancelled, a thief may use the card to make illicit purchases.

There are various types of payment card fraud:

Application fraud: Application fraud takes place when a person uses stolen or fake documents to open an account in another person's name. To create a personal profile, criminals may steal or fabricate papers like utility bills and bank accounts. The fraudster could take money out of the account or apply for credit in the victim's name after opening the account with bogus or stolen documents.

Account takeover: An account takeover refers to the act by which fraudsters will attempt to assume control of a customer's account (i.e. credit cards, email, banks, SIM card and more). Due to its varied characteristics, including class imbalance, this issue is particularly difficult to solve from the standpoint of learning. Fraudulent transactions are vastly outnumbered by legitimate ones, and transaction patterns frequently shift over time.

Skimming: Skimming is the stealing of private data that was used in a typically routine transaction. Employing simple techniques like duplicating receipts or more sophisticated ones like using a small electrical gadget (skimmer) to swipe and save several victims' card numbers, the thief can obtain the victim's card number.

Unexpected repeat billing: Repeat billing, also referred to as "repeated bank charges," is a result of

© 2023, IJCSMC All Rights Reserved, ZAIN Publications, Fridhemsgatan 62, 112 46, Stockholm, Sweden

online bill payment or online transactions made with a bank account. These are banker's orders or standing orders from customers to honor and pay the payee a specific sum each month.

In Big computing communities such as machine learning and data science the solution to this problem can be automated. Due to its varied characteristics, including class imbalance, this issue is particularly difficult to solve from the standpoint of learning. Fraudulent transactions are vastly outnumbered by legitimate ones, and transaction patterns frequently shift over time. stical properties over the course of time, and considering the huge traffic of transaction data, and it is not possible for humans to check manually every transaction one by one if it is fraudulent or not.

Algorithms for machine learning are used to analyze all permitted transactions and flag any that seem suspect. Professionals look into these reports and get in touch with the cardholders to confirm whether the transaction was legitimate or fraudulent.

STATEMENT OF PROBLEM

The ease of payment credit cards has brought to the marketing world is immeasurable. Transactions can now be made without any hassle. The world at large is taking advantage of this technology and now, almost all transaction are made with credit cards. This plight has brought about a need for more a secure way to handle these transactions. Fraudsters have been taking advantage of the technology to rob people of their money and they need to be stopped. When a credit card is copied or stolen, the transactions made by them are labeled as fraudulent. These fraudulent transactions should be prevented or detected in a timely manner otherwise the resulting losses can be huge. There has been a growing amount of financial losses due to credit card frauds as the usage of the credit cards become more and more common. As a result, numerous articles [2, 20] detailed significant losses in various nations, I. Elikucuk and E. Duman. There are numerous ways to commit credit card fraud, including straightforward theft, application fraud, using fake cards, never receiving issue (NRI), and online fraud (where the card holder is not present). In online fraud, the transaction is made remotely and only the card‘s details are needed. A manual signature, a PIN or a card imprint are not required at the time of purchase. Though prevention mechanisms like CHIP&PIN decrease the fraudulent activities through simple theft, counterfeit cards and NRI; online frauds (internet and mail order frauds) are still increasing in both amount and number of transactions. Online scams accounted for around 50% of all credit card fraud losses in 2008, according to Visa reports about European countries. When the fraudsters obtain a card, they usually use (spend) all of its available (unused) limit. Statistics show that people accomplish this in four to five transactions on average [18]. Thus, although the standard predictive modeling performance metrics are extremely essential for the fraud detection problem, as specified by bank authorities, a performance criterion, assessing the loss that can be avoided on the cards whose operations are identified as fraudulent, is more prominent. In other words, identifying fraud on a bigger available limit card is more beneficial than detecting fraud on a smaller available limit card.

The specific objectives are:

1. to create a credit card fraud detection system 2. to enable the credit card fraud detection system to detect fraud in real-time transaction 3.to automate both Machine Learning and Real-time Streaming Job using Apache Airflow 4.to create a fraud alert dashboard and display all the predicted fraudulent transactions 5. The system will be compared with existing system.

© 2023, IJCSMC All Rights Reserved, ZAIN Publications, Fridhemsgatan 62, 112 46, Stockholm, Sweden

II. REVIEW OF ELATED WORKS

Over the years, credit card fraud detection has drawn a lot of by researchers 'interest and several techniques have been suggested. (Aman, 2021) Aman, Arpit Mishra, Ashish Kumar and Naveen Pandey from Inderprastha Engineering College in their research ―Credit Card Fraud Detection using Machine Learning and Data Science‖ illustrated the modelling of a data set using machine learning. Their objective was to detect 100% of fraudulent transaction while minimizing the incorrect fraud classification. They achieved this by concentrating on the analysis and pre-processing of data sets, as well as the application of numerous anomaly detection methods, including Local Outlier Factor and Isolation Forest algorithm on the PCA converted Credit Card Transaction data. In their results, the code prints out the number of false positives transactions it detected and compares it with the actual values. This is done to determine how accurate and precise the algorithms are. While the algorithm does reach over 99.6% accuracy, its precision remains only at 28% when a tenth of the data set is taken into consideration. However, when the entire dataset is fed into the algorithm, the precision rises to 33%. The large percentage of correctness was anticipated due to the large disparity between the number of legitimate and authentic transactions. As a result, they encountered an issue with the dataset's imbalance. (Vaishnavi Nath Dornadulaa, 2019) Vaishnavi Nath Dornadula and Geetha from the Vellore Institute of Technology in India created a novel fraud detection method for Streaming Transaction Data in their study on Credit Card Fraud Detection Using Machine Learning Algorithms with the aim of examining past transaction details of the customers and extracting behavioral patterns. Based on the value of their transactions, cardholders were divided into various groups. Then, they combined the transactions performed by cards from various groups in order to extract the appropriate behavioral patterns for each group using the sliding window technique. The groups are then used to train various classifiers individually in the future. The classifier with the highest rating score was then picked as one of the most effective ways to detect fraud. They observed that the Matthews Correlation Coefficient was the better parameter to deal with imbalance dataset. Although it was not the only solution. By applying the SMOTE, they tried balancing the dataset and found that the classifiers were performing better than before. They also stated another way of handling imbalance dataset which is to use one-class classifiers like one-class SVM. Finally, they stated that Logistic regression, decision tree and random forest are the algorithms that gave best results.

In July 2018, Amanze, B.C and Onukwugha, C. G (Amanze & Onukwugha, 2018) stated that every cardholder has a certain shopping behavior, which establishes an activity profile for him. They discussed the ineffectiveness of existing fraud detection systems which try to capture behavioral patterns as static rules, when cardholder develops new patterns these rules become ineffective. Utilizing adaptive data mining and intelligent agents, they created a system for Nigerian financial industry that could incorporate evidence from both past and present behavior to assess the level of suspicion associated with each incoming transaction. The research focused on the predictive model using data mining that scores each transaction with high or low risk of fraud and those with high risk generate alerts. Predictive data mining perform interference on the current data to make predictions. The intelligent agents check those alerts and provide a feedback for each alert i.e. true positive (genuine) or false positive (fraud). (Vaishnave Jonnalagadda, 2019) VaishnaveJonnalagadda, Priya Gupta and Eesita Sen from the SRM Institute of Science and Technology, Chennai, Tamil Nadu, worked on a research that focused on credit card fraud detection in

© 2023, IJCSMC All Rights Reserved, ZAIN Publications, Fridhemsgatan 62, 112 46, Stockholm, Sweden

real-world scenarios. In this research they designed a model to detect the fraud activity in credit card transactions. Initially, they collected the credit card usage data-set by users and classified it as trained and testing dataset using a random forest algorithm and decision trees. They examined the broader data set and user-provided current data set using this workable technique. The accuracy of the outcome data was then improved. Applied processing to some of the provided properties that may have an impact on fraud detection when viewing the graphical data visualization model. Accuracy, sensitivity, and specificity were used to evaluate how well the procedures performed. They obtained the outcome of an accurate credit card fraud detection value. i.e. 0.9994802867383512 (99.93%) using a random forest algorithm with new enhancements. The Random forest algorithm will provide better performance with many training data, but speed during testing and application will still suffer. Usage of more pre-processing techniques would also assist. They proposed that their future work will try to represent this into a software application and provide a solution for credit card fraud.

2.1 MACHINE LEARNING: WHAT IS IT?

The goal of machine learning is to derive knowledge from data. Also referred to as statistical analysis or predictive analysis, this area of research sits at the nexus of computer science, artificial intelligence, and statistics. According to Sarah and Muller (2017), it has had a significant impact on how data-driven research is conducted nowadays. Another definition of machine learning is the area of artificial intelligence that gives systems the capacity to autonomously learn from experience without requiring for human involvement. It uses a variety of algorithms and models for predicting future events as accurately as possible. Shakya (2018).

2.2 Different types of machine learning algorithms: reinforcement learning, semi-supervised learning, unsupervised learning, and supervised learning. Nonetheless, both supervised and unsupervised learning are the most frequently utilized methods**.**

2.3 WHAT IS SUPERVISED LEARNING

Supervised learning algorithms are Machine learning algorithms that learn from input/output pairs. The algorithm receives the required solutions, or labels, as part of the training set. When an input/output pair exists and we want to forecast a specific outcome from that input, we use it. We use these input/output pairs as the training set for our machine learning model. Our objective is to accurately predict fresh, unforeseen facts. Building the training set is frequently labor-intensive, but once it is done, supervised learning automates and frequently accelerates a time-consuming or impossible operation. (Sarah & Muller, 2017). Mathematically, supervised learning can be represented as Y = f(x). Where x represents the input variables, Y denotes an output variable and f(X) is a mapping function. The goal is to approximate mapping function such that when an unseen input is given to the mapping function, it can predict the output variable (Y) correctly. (Shakya, 2018) Supervised learning is commonly used in real world applications, such as face and speech recognition, products or movie recommendations, and sales forecasting. Supervised learning can be further classified into two types: Regression and Classification. The objective of a classification task is to forecast a class label, which is a choice from a preset list of alternatives, where the output variable is a category (for example, fraud or real, rainy or sunny, etc.). The objective of a

© 2023, IJCSMC All Rights Reserved, ZAIN Publications, Fridhemsgatan 62, 112 46, Stockholm, Sweden

regression problem is to forecast a continuous number, or in programming terms, a floating-point number, where the output variable is a real value (e.g., the price of a house, temperature, etc. (or real number in mathematical terms). The classification issue at the center of this endeavor.

2.4 Classification

A machine learning technique called classification uses existing categories to determine how new data should be categorized into them. A predicted class, which takes the form of a discrete category, is produced by classification models. A discrete category prediction is necessary in order to reach a conclusion for the majority of real-world applications. (Modeling Applied Predictive, 2013) In classification tasks, the program must learn to predict discrete values for the dependent or output variables from one or more independent or input variables. That is, the program must predict the most probable class, category or label for new observations. For example, fraud detection can be identified as a classification problem. In this case, the goal is to predict if a given transaction is fraud or genuine.

Generally, there are three types of classification: binary classification, where there are two output labels (e.g., classifying a transaction which may be fraud or genuine), multi-class classification, where there are more than two output labels (e.g., classifying a set of images of flowers which may be Rose or Lilly or Sunflower) and multi-label classification, where the data samples are not mutually exclusive and each data samples are assigned a set of target labels (e.g., classifying a crab on the basis of the sex and color in which the output labels can be male/female and red/black). (Shakya, 2018) This research deals with the binary classification problem where the output label is either normal or fraud.

2.5 Binary Classification

We can think of binary classification as trying to answer a yes/no question. In binary classification we often speak of one class being the positive class and the other class being the negative class. Here, positive doesn‘t represent having benefit or value, but rather what the object of the study is. Classifying emails as either spam or not spam is an example of a binary classification problem. In this binary classification task, the yes/no question being asked would be ―I this email spam? When working on classification predictive modeling problems, we must collect a training dataset.

2.6 Training Dataset

A training dataset is several examples from the domain that include both the input data (e.g. measurements) and the output data (e.g. class label). Depending on the complexity of the problem and the types of models we may choose to use, we may need tens, hundreds, thousands, or even millions of examples from the domain to constitute a training dataset. In order to appropriately prepare the input data for modeling, the training dataset is used to better comprehend it. Additionally, it is used to assess a variety of modeling approaches. It is used to fine-tune a model's hyperparameters. The last step involves using the training dataset to create a model that can be applied to all data and used to predict future examples from the issue area. (Brownlee, 2019)

© 2023, IJCSMC All Rights Reserved, ZAIN Publications, Fridhemsgatan 62, 112 46, Stockholm, Sweden

2.7 Imbalanced Classification Problems

The distribution of examples into classes is referred to as the class distribution. An imbalance classification problem is one where there are not an equal number of examples in the training dataset for each class label. That is, when the distribution of students by class is biased or skewed rather than equal or nearly equal. (Brownlee, 2019). The distribution of classes in a given training dataset identifies a problem's imbalance. The definition of class imbalance must consider a dataset or distribution. Class imbalance is often measured in relation to the training distribution since class labels are necessary to establish the level of class imbalance (Imbalanced Learning: Foundations, 2013). The class distribution in fraud detection is already unbalanced.

2.8 Combating Imbalanced Training Data

Most machine learning algorithms work best when the number of samples in each class are about equal. This is so because algorithms are created to minimize error and increase accuracy. This section will cover potential solutions to the categorization imbalance issue that plagues the detection of credit card fraud. This is so because algorithms are created to minimize error and increase accuracy. This section will cover potential solutions to the categorization imbalance issue that plagues the detection of credit card fraud.

2.9 Resampling approach

To address the problem of imbalanced learning, many resampling techniques have been created. Resampling techniques include

- oversampling Under-sampling
- combining oversampling and under sampling techniques Ensemble sampling.

Oversampling and under sampling both try to change the proportions between the majority and minority classes. Using oversampling and under sampling approaches together results in a more balanced new dataset. Resampling allows different classes to have about the same influence on the classification model's outputs by making the training data more balanced. (Bagui &Kunqi, 2021)

2.9.1 SELECTED MODEL

In this section, we will discuss the different models selected for the predictive analysis. Depending on the nature of the classification problem, we chose two very popular predictive model.

2.9.2 Random forest

Random forest is an ensemble learning algorithm, which can be used for both regression and classification task. In a random forest, the weak learners are decision trees. So, before going into the details of the random forest, we will try to understand the basics of decision trees. An approach for supervised learning that may be applied to both regression and classification is the decision tree. But classification issues are where it's most frequently used. It is made up of a number of internal nodes, where each node stands for a test of a characteristic (such as whether it will be sunny, cloudy, or rainy tomorrow). The tree's leaf nodes reflect the end result, while each branch shows the results of the test (class label). It involves breaking down a training set into several subsamples. A decision tree involves a series of if-then conditions that are used to make the final decision.

The splitting of the total training data into subsets, which is done in each internal node according to some

© 2023, IJCSMC All Rights Reserved, ZAIN Publications, Fridhemsgatan 62, 112 46, Stockholm, Sweden

condition, is necessary for the construction of a decision tree. The Gini impurity and information gain metrics, among others, are used by the decision tree method to find the appropriate split of each node. Gini impurity is an indicator of how frequently a randomly selected element from the set would be mislabeled if it were randomly classified in accordance with the distribution of labels in the subset. At each stage of the tree-building process, the decision on which feature to split on is made using information gained. The process of splitting continues until the internal node has the class label value. Although decision trees are easy to understand and perform well in some datasets, they tend to have a high variance because of the greedy approach of the algorithm where the tree tends to always select the best split at each level and it cannot see far behind the current level. Due to this reason, there may be the possibility of overfitting, where the model only performs better in the training set and fails to perform well in test sets.

Random forest algorithm mitigates the overfitting problem well by using the bootstrap concept. In simple language, the random forest builds multiple decision trees and combines them to improve the performance of the model. Bootstrapping is the process of sampling the training data randomly with replacement.

Random forest utilizes bootstrapping such that each decision tree will be trained with different subsamples of data. Moreover, the random forest uses random subsets of features. For example, if there are 50 features in the data, random forest will only choose a certain number of them, let‘s say 10, to train on each tree. Thus, each tree will have 10 random features that will be used for training including finding the best split of each node of the tree. Once we have the collection of decision trees, the results of each tree will be aggregated to get the final result (vote). The model trained in such a way will ensure generalization since not one, but multiple decision trees are used for making the decision, and moreover, each tree is trained with different subsections of data.

EVALUATION METRICS

In many cases, we tend to believe that the model with the highest accuracy should be considered the best choice for prediction. However, there are situations where selecting a model with lower accuracy can be more desirable because it provides enhanced predictive power for the specific problem at hand. This phenomenon is commonly referred to as the Accuracy Paradox (Lamba, 2020).

In the field of machine learning, the process involves training the model with the available training data and then assessing its ability to generalize by evaluating its performance on unseen data. In simpler terms, we examine how well the model performs when tested on data that it has not encountered during training. To measure the performance of the model, we rely on evaluation metrics that are specific to the nature of the problem, such as regression or classification. For the purpose of this section, we will focus solely on the evaluation metrics relevant to the classification problem (Shakya, 2018).

2.9.3 Confusion matrix

Of course! Please provide the content of the page you would like me to rephrase, and I'll be happy to assist you in rephrasing it.

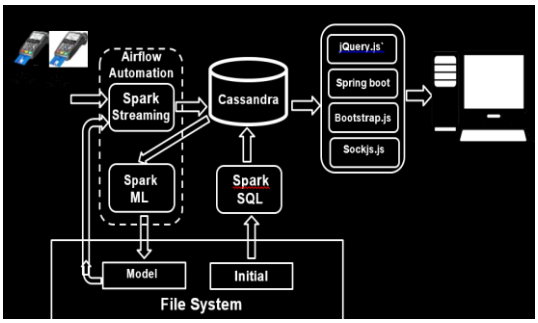
2.9.4 F1 Score

The harmonic mean of recall and precision is known as the F1 Score, sometimes spelled F Score or F-measure. It represents the weighted average of recall and precision. Its value is between 0 and 1, with 1 being the best and 0 being the worst.

© 2023, IJCSMC All Rights Reserved, ZAIN Publications, Fridhemsgatan 62, 112 46, Stockholm, Sweden

III. IMPLEMANTATION AND SYSTEM DESIGN

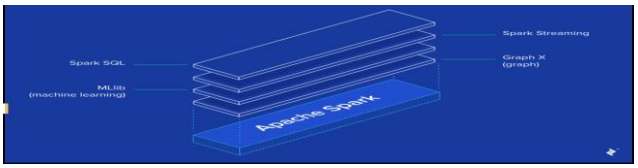
A real-time system for detecting credit card fraud will be created and automated as part of this research. Spark, Kafka, and Cassandra are used in its implementation.



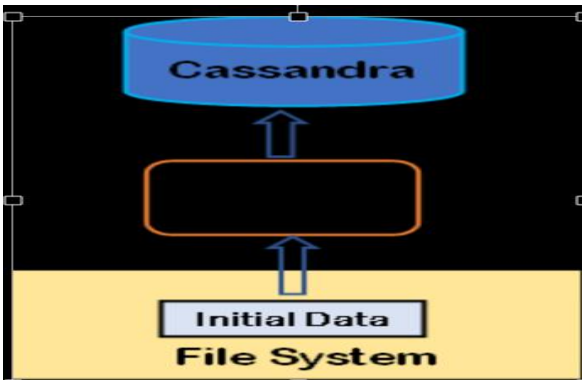
File System

3.1 Apache Spark

The study conducted by Apache According to its marketing, Spark provides "lightning-fast cluster computing." This multilingual engine may be utilized on cluster or single-node computers for machine learning, data science, and data engineering applications. The Unified Engine for large-scale data analytics is Spark, Apache SparkTM (n.d.). As of right now, it is the most active branch of Apache research, with a thriving open-source community. Spark provides a framework for organizing information more quickly and creatively. Spark offers significant performance advantages over its competitor, Hadoop, allowing applications to execute up to a hundred times faster in memory or ten times faster on disk. At the core of Spark lies a powerful engine designed for decentralized and highly parallel data processing. It is complemented by a range of high-level libraries that seamlessly integrate within the same application. These libraries encompass GraphX, MLlib (for machine learning tasks), SparkSQL, and Spark Streaming, enhancing the capabilities and versatility of Spark for various data processing needs.



3.2 Spark SQL Job

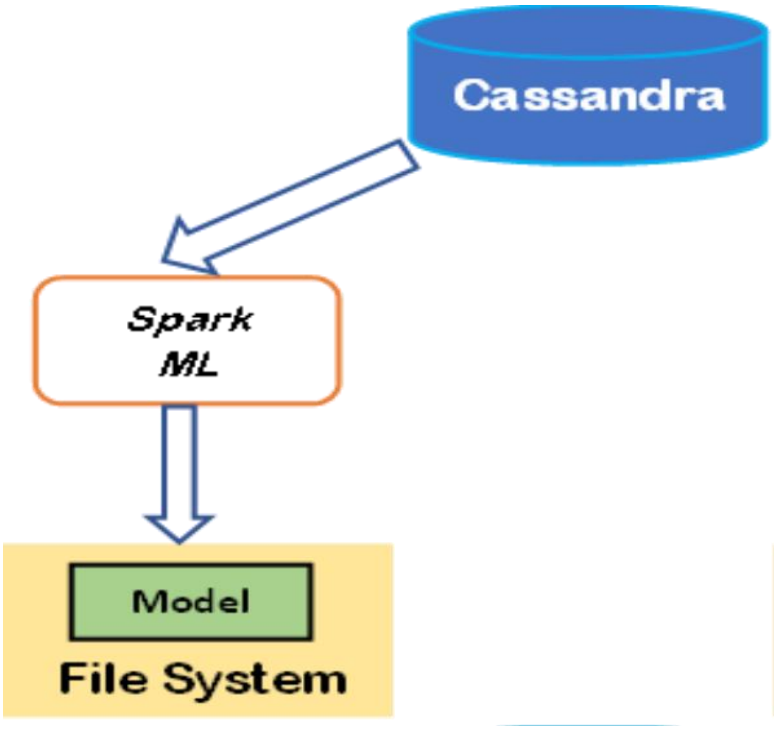


A Spark module called Spark SQL is used to manage structured data. Spark has access to more details about the structure of the data and the computation being done because of the APIs that Spark SQL provides. With the help of this additional data, Spark SQL internally improves. The SQL Programming Guide by Spark, n.d.In this

© 2023, IJCSMC All Rights Reserved, ZAIN Publications, Fridhemsgatan 62, 112 46, Stockholm, Sweden

study, the Cassandra database will use the Spark SQL Job to get all information about transactions from the file system. False transactions are imported into a fraud table while non-fraudulent transactions into a non-fraudulent table using this process.

3.3 Spark Machine Learning Job



In this research, the Spark ML Job will read fraud and non-fraud transactions from Cassandra. It will train on this data and it will create a model.

This model will be saved to the file system.

A new package called spark.ml was added to Spark 1.2 with the intention of offering a standardized set of high-level APIs to assist users in building and fine-tuning useful machine learning pipelines. . (Apache, Spark ML Programming Guide - Spark Documentation (apache.org), n.d.)

To make it simpler to incorporate many algorithms into a single pipeline, or workflow, Spark ML standardizes machine learning APIs. The Spark ML API's primary principles are covered in this section. (Apache, Spark ML Programming Guide - Spark Documentation (apache.org), n.d.)

ML Dataset: Spark ML uses the SchemaRDD dataset from Spark SQL, which accommodates a variety of data formats. For instance, a dataset could include several columns for text, feature vectors, actual labels, and predictions.

Transformer: An algorithm called a Transformer can change one Schema RDD into another. An RDD with features can be transformed into an RDD with predictions using an ML model, for instance.

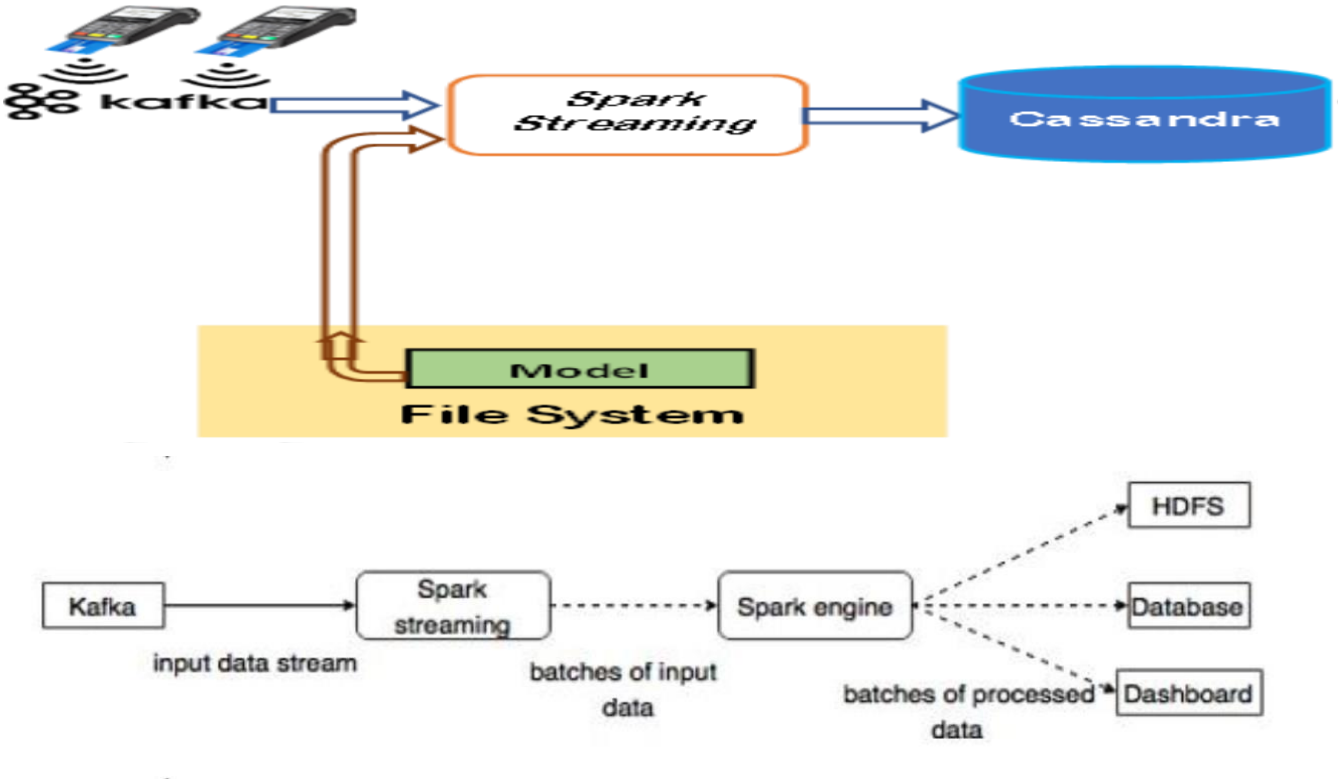
Estimator: A program called an Estimator can be used to generate a Transformer by fitting it to a Schema RDD. An Estimator, for instance, is a learning algorithm that trains on data and builds models.

Pipeline: To specify an ML workflow, a pipeline links numerous transformers and estimators together.

Param: The specification of parameters is now standardized across all Transformers and Estimator.

© 2023, IJCSMC All Rights Reserved, ZAIN Publications, Fridhemsgatan 62, 112 46, Stockholm, Sweden

3.4 Spark Streaming Job



The model will be loaded from the file system via the Spark Streaming Job. After that, it will begin consuming Kafka messages about credit card transactions. This model can be used to determine whether a transaction is fraudulent or not. The forecasts will then be saved to the Cassandra database. Non-fraudulent transactions are saved to a separate table from fraudulent transactions, which are saved to the fraud table.

Based on the Spark SQL engine, Spark Structured Streaming is a fault-tolerant and scalable stream processing engine. The same syntax that is used for batch computations on static data can also be used to define streaming computations.

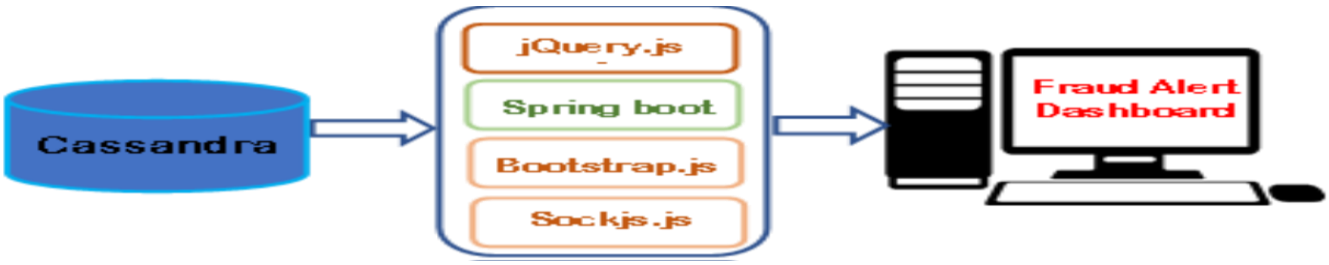
The Spark SQL engine will take care of running it continuously and incrementally, updating the outcome as fresh streaming data is received. (Apache, Structured Streaming Programming Guide - Spark 3.2.1 Documentation (apache.org), n.d.). Internally, by default, Structured Streaming queries are handled by a micro-batch processing engine.

This engine breaks up large data streams into smaller batches to process them, resulting in end-to-end latencies as low as 100 milliseconds and exact-once fault-tolerance guarantees. With at least-once guarantees, our Continuous Processing low-latency processing option, introduced with Spark 2.3, may reach end-to-end latencies as low as 1 millisecond. Without changing the Dataset/Data Frame operations in your queries, you will be able to choose the mode based on your application requirements. (Apache, Structured Streaming Programming Guide - Spark 3.2.1 Documentation (apache.org), n.d.) With the same and well-known Spark APIs, Spark Structured Streaming makes it simple to create streaming applications and pipelines.

© 2023, IJCSMC All Rights Reserved, ZAIN Publications, Fridhemsgatan 62, 112 46, Stockholm, Sweden

Integration of Kafka with Spark

Kafka may be used by Spark streaming as a platform to facilitate communication and integration. Spark Streaming uses complex techniques to efficiently handle real-time data flows by utilizing Kafka as its centralized hubs. After processing the data, Spark Streaming may store the outcomes in a variety analytics locations, such HDFS, databases in addition or dashboards, or broadcast the results onto a new Kafka topic. The following image below indicates the way the technique is conceptually organized.



3.5 Fraud-alert dashboard

The Scam Alert Panel will do an internal query on the Mongodb database to retrieve some of the most recent examples of fraudulent transactions. Every transaction suspected of being fraudulent is going to be shown on the dashboard. We will incorporate Bootstrap and jQuery prerequisites so as to make use of Spring Boot to construct the fraudulent alert panel web application. These dependencies will be bundled into webjars, that are JAR (Java Archive) files that manage client-side dependencies in web applications running on the Java Virtual Machine (JVM) in an easy and explicit manner.

What is jQuery?

jQuery is a JavaScript library that aims to simplify JavaScript usage on websites, following the principle of "write less, do more." It is known for its compact size and efficiency. By utilizing jQuery's methods, you can perform various essential tasks with minimal JavaScript code. Some of the complex JavaScript features that jQuery significantly simplifies include AJAX queries and DOM manipulation. The jQuery library offers features such as HTML/DOM manipulation, CSS manipulation, HTML event methods, effects and animations, AJAX, and utilities.

Why Choose jQuery?

While there are multiple JavaScript libraries available, jQuery stands out as one of the most popular and versatile options. It eases traversing HTML pages, managing events, implementing animations, and handling Ajax interactions, making web development faster. Compared to plain JavaScript and other JavaScript libraries, jQuery provides a more user-friendly approach and requires fewer lines of code.

Spring Boot

Spring Boot is an open-source Java-based framework used for building microservices, which allows developers to develop and deploy services independently. It is created by the Pivotal Team. Spring Boot enables Java programmers to build production-ready, standalone Spring applications with ease.

© 2023, IJCSMC All Rights Reserved, ZAIN Publications, Fridhemsgatan 62, 112 46, Stockholm, Sweden

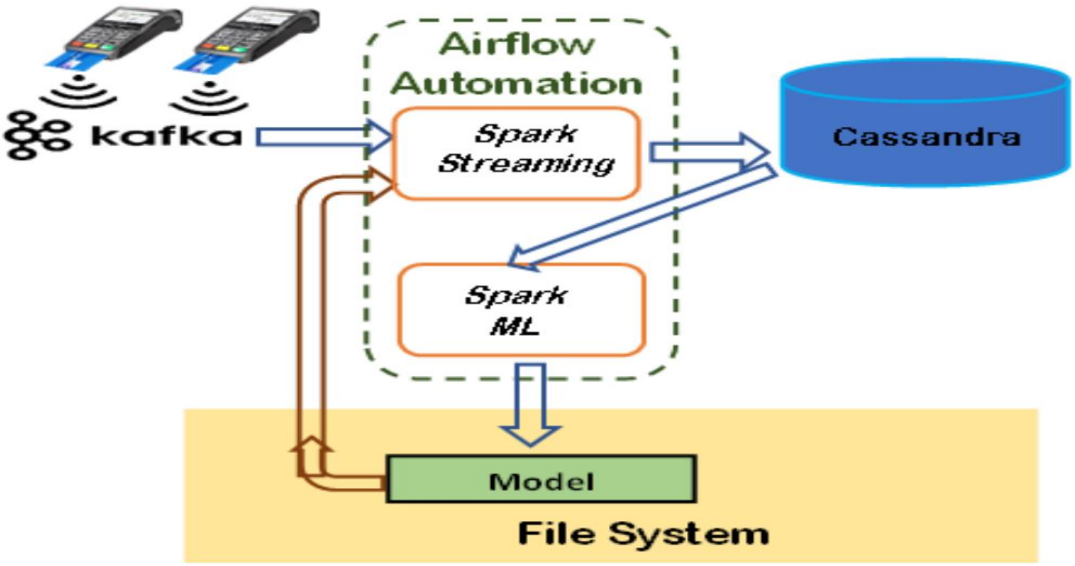
Why Use Spring Boot?

Spring Boot offers a flexible way to configure Java Beans, XML configurations, and database transactions. It provides robust batch processing and maintains REST endpoints. With Spring Boot, most configurations are automatically handled without manual intervention. It simplifies dependency management and supports the use of annotations in Spring applications.

What is SockJS?

SockJS is a JavaScript package designed for browsers that offers a WebSocket-like object. The WebSocket API is a cutting-edge technology that facilitates interactive, two-way communication sessions between a user's browser and a server. It allows sending messages to a server and receiving event-driven responses without the need for continuous server polling. SockJS provides a uniform, cross-browser JavaScript API that establishes a low-latency, full-duplex, cross-domain communication channel between the browser and the web server, regardless of whether WebSocket is supported or not.

3.6 Airflow Automation



Airflow automation is employed to stop and start a new Spark Streaming job each time a new model is constructed. In a production environment, the Data Science Team manually runs the Spark ML Job. The development of a new model involves manual execution on a weekly or monthly basis. The effectiveness of the new model is assessed and compared to the previous model. If the new model outperforms the current one, it is implemented. To select the new model, the currently running Spark Streaming Job is halted, and a new one is initiated.

Airflow serves as a task automation tool, enabling the scheduling of jobs to ensure timely completion within businesses. This eliminates repetitive work for employees. Apache Airflow is specifically utilized for planning and coordinating data pipelines or operations. With Airflow, users can automate scripts to perform tasks. It incorporates a scheduler that manages various tasks while adhering to predetermined dependencies. Airflow's comprehensive command-line tools (DAGs) facilitate easy interaction with directed acyclic graphs, simplifying task ordering and management for enterprises.

© 2023, IJCSMC All Rights Reserved, ZAIN Publications, Fridhemsgatan 62, 112 46, Stockholm, Sweden

Read data from Cassandra

```
val fraudTransactionDF =
DataReader.readFromCassandra(CassandraConfig.keyspace,
CassandraConfig.fraudTransactionTable)
      .select("cc_num" , "category", "merchant", "distance", "amt", "age",
"is_fraud")

val nonFraudTransactionDF =
DataReader.readFromCassandra(CassandraConfig.keyspace,
CassandraConfig.nonFraudTransactionTable)
      .select("cc_num" , "category", "merchant", "distance", "amt", "age",
"is_fraud")

val transactionDF = nonFraudTransactionDF.union(fraudTransactionDF)
    transactionDF.cache()
```

```
+----------------+-------------+-----------------------------+--------+-----+---+--------+
|cc_num          |category     |merchant                     |distance|amt  |age|is_fraud|
+----------------+-------------+-----------------------------+--------+-----+---+--------+
|5157595343543285|gas_transport|Schmitt Inc                  |0.73    |111.0|46 |0.0     |
|5157595343543285|kids_pets    |Beer-Jast                    |1.73    |66.0 |46 |0.0     |
|5157595343543285|grocery_pos  |Hudson-Ratke                 |1.23    |220.0|46 |0.0     |
|5157595343543285|entertainment|Kassulke Inc                 |2.17    |100.0|46 |0.0     |
|5157595343543285|shopping_net |Boyer PLC                    |1.33    |80.0 |46 |0.0     |
|5157595343543285|entertainment|Spencer PLC                  |1.29    |141.0|46 |0.0     |
|5157595343543285|entertainment|Brown-Greenholt              |0.8     |221.0|46 |0.0     |
|5157595343543285|travel       |Kovacek Ltd                  |1.5     |91.0 |46 |0.0     |
|5157595343543285|gas_transport|Kutch LLC                    |0.71    |57.0 |46 |0.0     |
|5157595343543285|kids_pets    |Lowe, Dietrich and Erdman    |1.41    |158.0|46 |0.0     |
|5157595343543285|kids_pets    |Bednar PLC                   |1.99    |211.0|46 |0.0     |
|5157595343543285|kids_pets    |Hammes-Beatty                |0.88    |42.0 |46 |0.0     |
|5157595343543285|kids_pets    |Hilpert-Conroy               |0.97    |103.0|46 |0.0     |
|5157595343543285|gas_transport|Robel, Cummerata and Prosacco|1.26    |100.0|46 |0.0     |
|5157595343543285|grocery_pos  |Strosin-Cruickshank          |1.02    |75.0 |46 |0.0     |
|5157595343543285|grocery_pos  |Schultz, Simonis and Little  |1.8     |65.0 |46 |0.0     |
|5157595343543285|home         |Beier and Sons               |1.66    |112.0|46 |0.0     |
|5157595343543285|misc_pos     |Turner, Ruecker and Parisian |1.81    |27.0 |46 |0.0     |
|5157595343543285|entertainment|Howe PLC                     |1.56    |151.0|46 |0.0     |
|5157595343543285|entertainment|Bauch-Blanda                 |1.88    |126.0|46 |0.0     |
+----------------+-------------+-----------------------------+--------+-----+---+--------+
only showing top 20 rows
```

Training

This situation can be classified as a misclassification, and it has the potential to be classified as fraud, although it is not certain. The current project framework utilizes the random forest algorithm. In a real-world scenario, multiple classification methods would typically be tested. To evaluate these algorithms, metrics such as the confusion matrix, AUC, and ROC would be utilized. However, in this specific case, we are solely using the Random Forest algorithm and not exploring other approaches. The fit() method must be called in order to use the Random Forest algorithm as an estimator. The Random Forest technique begins training on the given data when the fit() function is invoked, and it finally delivers a model that serves as a transformer. Here, you may observe the random forest model Model that we use in this particular case. We can use this trained model to make forecasts about the test data through the use of the convert() method.

© 2023, IJCSMC All Rights Reserved, ZAIN Publications, Fridhemsgatan 62, 112 46, Stockholm, Sweden

RESULTS AND DISCUSSIONS

```
def randomForestClassifier(df: org.apache.spark.sql.DataFrame)(implicit
sparkSession:SparkSession) = {
    import sparkSession.implicits._
    val randomForestEstimator = new
RandomForestClassifier().setLabelCol("is_fraud").setFeaturesCol("features").setMax
Bins(700)
    val model = randomForestEstimator.fit(df)
    model
  }
```

4.1 Confusion Matrix

True Positive (TP) = Model's prediction of 91 fraudulent transactions

False Positive (FP) = System that misclassifies legitimate transactions as fraudulent = 83

Model predicting fraudulent transactions as being legitimate = false negative (FN) = 0.

The model's forecast of 90 fraudulent transactions is the true negative (TP).

False Positive (FP) = System that incorrectly flags for fraudulent 82 genuine activities

The fact that false negative (the FN) equal to zero indicates that the predictive algorithm predicts a fraud that are legitimate.

Output

```
================== Confusion matrix ============================
##############| Actual = 1                                  Actual = 0
--------------+-------------------------------------------------------
Predicted = 1| 91.000000                                   83.000000
Predicted = 0| 0.000000                                    2548.000000
================================================================

True Positive Rate: 1.0
False Positive Rate: 0.03154694
Precision: 0.5229885
```

True Positive Rate = TP/TP + FN = 91.0/ (91.0 + 0) = 1.0

Out of all the fraud transactions, how much we predicted correctly, It should be high as possible.

False Positive Rate (FPR) = FP/FP + TN

Out of all the genuine transactions (not fraud), how much we predicted wrong (predicted as fraud). It should be low as possible Precision = TP/(TP + FP) = 91/(91.0 + 83.0) = 0.5229885

Out of all the fraud transactions that we have predicted correctly, how many are actually fraud.

4.2 Spark Streaming

Both the Preprocessing Model and the Random Forest model that were produced by the Spark ML Job will be loaded by the Spark Streaming Job. It will start consuming Kafka messages about credit card transactions.

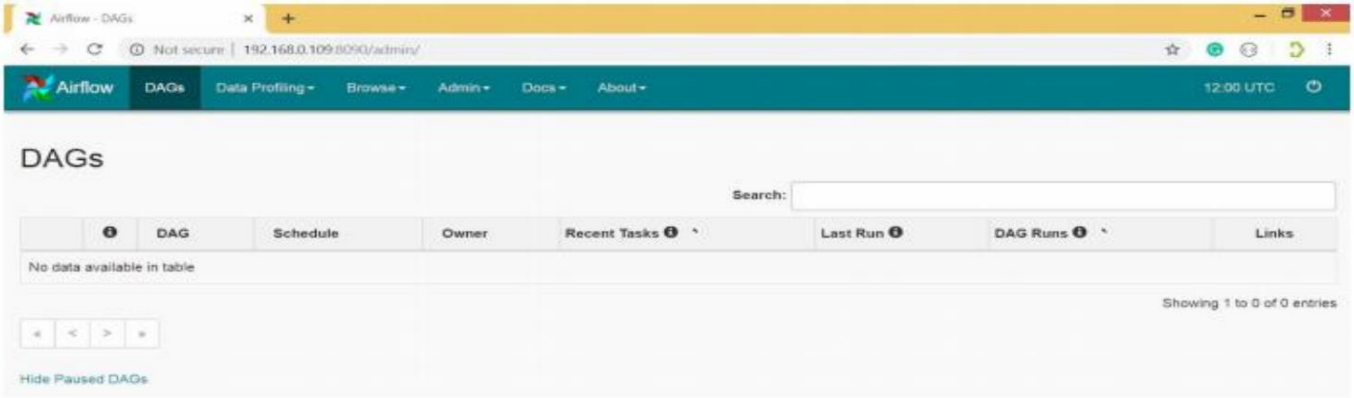
All the necessary columns (merchant, category, distance, amt, age) will be converted to feature columns using the preprocessing model. It will determine using the Random Forest Model if a transaction is fraudulent or not. Along with saving projected transactions to Cassandra, we are also saving the Kafka offset. Predicted

© 2023, IJCSMC All Rights Reserved, ZAIN Publications, Fridhemsgatan 62, 112 46, Stockholm, Sweden

transactions will be stored to Cassandra Key space. Fraud detection requires exactly-once semantics, which means that transactions cannot be lost and cannot be duplicated. By manually committing offset, we are able to achieve exactly-once semantics in this case.

4.3 Airflow Automation

A framework for automation is Apache Airflow. Jobs are automated using it. In place of XML workflows, Python can be used to write workflows. Here, both the Spark Streaming and Spark ML jobs are automated. However, a Spark ML Job is manually executed to construct a Model in a production context. The model's effectiveness will be assessed. The Model will be implemented if it is effective. We need to rebuild the model utilizing both the old data and the new data as fresh transaction data is recorded. Therefore, a new model will often be developed once a week or once a month. However, we execute the Spark ML Job every 20 minutes for the sake of demonstration. Upon creation of the new model, Spark Streaming Job must select.



CONCLUSION

Apache Airflow provides an automated platform. It is used to automate tasks. The Python workflow can be written in place of XML workflows. In this case, the Spark machine learning as well as Spark streaming activities were automated. However, to build a model in a production setting, an Apache Spark machine learning Task needs to be explicitly completed. The efficacy of the model will be evaluated. The proposal will be implemented if it proves to be successful. It is necessary to rebuild the structure using both the old and new data upon recording new details about the transaction. As a result of this, novel concepts can be produced on a weekly or monthly basis. We run a Spark machine learning job once every twentieth minute as an example. The Spark Streaming Task requires to choose whenever an updated version is produced.

REFERENCES

[1]. Aman, A. M. (2021, july). Credit Card Fraud Detection using Machine Learning and Data Science. International Journal for Research in Applied Science & Engineering Technology (IJRASET), 9 (vii).

[2]. Amanze, B. C., & Onukwugha, C. G. (2018, DECEMBER). AN ENHANCED MODEL FOR BANK FRAUD DETECTION IN NIGERIAN. INTERNATIONAL EDUCATIONAL JOURNAL OF SCIENCE AND ENGINEERING (IEJSE), 1(5).

[3]. Apache. (n.d.). Spark ML Programming Guide - Spark 1.2.2 Documentation (apache.org). Retrieved from https://spark.apache.org/docs/1.2.2/ml-guide.html.

[4]. Apache. (n.d.). Structured Streaming Programming Guide - Spark 3.2.1 Documentation (apache.org). Retrieved from https://spark.apache.org/docs/latest/structured-streaming programming-guide.html

[5]. Bagui, S., & Kunqi , L. (2021). Resampling imbalanced data for network intrusion detection datasets. Journal of Big Data.

[6]. Brownlee, J. (2019, December 23). A Gentle Introduction to Imbalanced Classification. Imbalanced Classification.

[7]. Chuprina, R. (2021, feburary 25). Credit Card Fraud Detection: Top ML Solutions in 2021.Imbalanced Learning:

© 2023, IJCSMC All Rights Reserved, ZAIN Publications, Fridhemsgatan 62, 112 46, Stockholm, Sweden

Foundations, A. a. (2013). Imbalanced Learning: Foundations, Algorithms, and Applications. In Imbalanced Learning: Foundations, Algorithms, and Applications (p. 16).

[8]. Jesus, A. d. (2019, November 22). Machine Learning for Credit Card Fraud – 7 Applications for Detection and Prevention. Discover the critical AI trends and applications that separate winners from losers in the future of business.

[9]. Lakshmi, M. G., Mettu , V., & Hameed, K. (2020). Credit Card Fraud Detection Using Random Forest. Journal of Information and Computational Science, 10(3). Retrieved from www/joics.org

[10]. Lamba, H. (2020). CREDIT CARD FRAUD DETECTION IN REAL TIME. Thesis, CALIFORNIA STATE UNIVERSITY SAN MARCO. 42 Modeling Applied Predictive. (2013). Applied Predictive Modeling.

[11]. PROSHARE. (2020, AUGUST 31). Fraudulent E-transactions Involving Credit and Debit Cards: Who is Liable? - A Case Study. Sarah, G., & Muller, A. C. (2017). Introduction to Machine Learning. United States of America: O'Reilly Media, Inc.

[12]. Shakya, R. (2018). Application of Machine Learning Techniques in Credit Card Fraud detection. University of Nevada, Las Vegas, Department of Computer Science.

[13]. Spark, A. (n.d.). Apache Spark™ - Unified Engine for large-scale data analytics. Retrieved from https://spark.apache.org/

[14]. Spark, A. (n.d.). Sql Programming Guide. Retrieved from https://spark.apache.org/docs/latest/sqlprogramming-guide.html

[15]. TutorialsPoint. (2015). Machine learning with python. Tutorials Point (I) Pvt. Ltd.

[16]. Vaishnave Jonnalagadda, P. G. (2019). Credit card fraud detection using Random Forest Algorithm. International Journal of Advance Research, Ideas and Innovations in Technology, 5(2). Retrieved from www.ijariit.com

[17]. Vaishnavi Nath Dornadulaa, G. S. (2019). Credit Card Fraud Detection using Machine Learning Algorithms. INTERNATIONAL CONFERENCE ON RECENT TRENDS IN ADVANCED COMPUTING 2019, ICRTAC 2019.

[18]. Wikipedia. (2021, June 23). Credit card fraud the free encyclopedia.

© 2023, IJCSMC All Rights Reserved, ZAIN Publications, Fridhemsgatan 62, 112 46, Stockholm, Sweden