End-to-End Real-time Architecture for Fraud Detection in Online Digital Transactions

ABBASSI Hanae, BERKAOUI Abdellah, ELMENDILI Saida, GAHI Youssef
Engineering Sciences Laboratory-National School of Applied Sciences, Ibn Tofail University, Kenitra, Morocco

***Abstract*—The banking sector is witnessing a fierce concurrence characterized by changing business models, new entrants such as FinTechs, and new customer behaviors. Financial institutions try to adapt to this trend and invent new ways and channels to reach and interact with their customers. While banks are opening their services to avoid missing this shift, they become naturally exposed to fraud attempts through their digital banking platforms. Therefore, fraud prevention and detection are considered must-have capabilities. Detecting fraud at an optimal time requires developing and deploying scalable learning systems capable of ingesting and analyzing vast volumes of streaming records. Current improvements in data analytics algorithms and the advent of open-source technologies for big data processing and storage bring up novel avenues for fraud identification. In this article, we provide a real-time architecture for detecting transactional fraud via behavioral analysis that incorporates big data analysis techniques such as Spark, Kafka, and h2o with an unsupervised machine learning (ML) algorithm named Isolation Forest. The results of experiments on a significant dataset of digital transactions indicate that this architecture is robust, effective, and reliable across a large set of transactions yielding 99% of accuracy, and a precision of 87%.**

***Keywords*—Online fraud; big data analytics; fraud detection; behavior analysis; isolation forest**

I. Introduction

Digital transaction fraud happens when an entity gains illegal entry to a banking account and utilizes it to make online transactions. Fraud detection techniques seek to exploit two fundamental limits that fraudsters experience while committing online transaction theft. First, most fraud techniques are susceptible to restricted time limits since consumers and banks block account access immediately with fraud discovery. Therefore, fraudsters are needed to hit the credit limit on the account in a brief amount of time, and as a result, their act is revealed in the shape of suspicious transactions over these shorter periods. In addition, the second type of restraint is created by the variety of digital transactions exposed by financial institutions and the variety of customer behaviors and awareness regarding fraud threats throughout these channels. Fortunately, monitoring measures applied from the financial institutions' side added to device security measures used from the customer's side may impede, in many cases, fraud attempts. Such impediments push fraudsters to target a small niche of customers that don't frequently interact through digital channels or aren't aware of security measures [1] that should be applied. As a result of this condition, fraudulent transactions are made at a few specific accounts that vary from the area set of customers to which the requirements above are applied.

Conventional methods for detecting system fraud are rule-based [2] [3]. Although rule-based solutions help prevent many fraudulent transactions, they remain static and don't adapt to fraud trends changes even when humans adjust continuously. As a result, machine learning-based algorithms have emerged as a non-deterministic approach employed for digital payment fraud detection in recent years. This area of study, however, has received little attention in the literature. Many of the solutions that have been examined propose a technique based on intelligent field knowledge features or make a fundamental assumption about transaction chronology [4]. To conduct a digital payment transaction, a fraudster should first login into the banking system. The payment and login are two separate processes that must be performed within this sequence. Anyway, this simple procedure follows a typical series of events. Although the bulk of fraud efforts are more complicated, the utmost existing systems focus on records from the most contemporary transactions, depending on hand-engineered characteristics that will presumably identify broad connections.

The target of this study is real-time fraud detection in digital banking. In this context, a fraud detection scheme aims to determine the risk within each item as in kind of fraud likelihood in real-time. The bank may then opt to authorize the transaction, refuse it, or demand a specific type of authentication on the consumer after completing it. To tackle this issue, we present a real-time architecture for detecting transactional fraud through behavioral analysis, which combines big data analysis tools (Kafka, Spark, and h2o) with the Isolation Forest algorithm to see suspicious transactions and provide excellent detection performance. The experimental findings are provided to confirm the efficacy of our strategy.

Concisely, the following are the foremost contributions of the present research:

1) Establishing an advanced fraud detection architecture for digital transactions by combining an unsupervised learner with big data analytics kits for real-time detection and training time reduction, enabling it to identify fraud in a typical online transaction context,

2) Using efficient feature engineering methods on the raw data. This involves producing aggregate features based on transaction frequency and isolating complex characteristics including the transaction's date to month, day, location, and so on. Because of the variety of features, our model can detect patterns that individuals or primitive machine learning algorithms cannot,

3) Applying the isolation forest model for identifying suspicious transactions,

4) Extensive evaluations have been carried out to assess the efficacy of the suggested architecture.

Our suggested architecture outperforms crucial benchmarks on online transaction fraud data encompassing over a hundred million transactions in a thorough experimental examination. More precisely, we show how our method can be used to meet strict operating time limitations while still maximizing prediction performance requirements relevant.

The remnant of this article is structured as following: in the second part, we present a review of the online transactions' fraud detection literature. Section III presents the research summary. Section IV offers the Isolation Forest learner. Section V describes our suggested architecture for online fraud transaction detection. And Section VI outlines the end-to-end data pipeline and the dataset used and explains the findings. Further Section VII discusses our results providing comparison with state-of-the-art studies. Finally, Section VIII concludes the work with suggestions for further research.

II. Fraud Detection Techniques for Online Transactions

World banking services and industries have been subjected to massive e-frauds, which have resulted in the overturning of whole organizations, enormous investment losses, and considerable litigation expenditures. As a result, companies and scientific studies have shown a keen interest in detecting online fraud. This section examines various significant study topics relevant to our work.

A. Outline of Extant Banking Fraud Recognition Methods and their Drawbacks

For many years, the general strategy in the cybersecurity business has been to prevent hypothetically fraudulent transactions by enforcing a set of strict criteria. A fraudulent identification rule-based system is designed to detect only elevated abusive transactions [4] [5]. This strategy efficiently reduces scam attempts and provides clients with a wisdom of security by uncovering well-known fraud trends. Nonetheless, rule-based detecting fraud technologies have shown in the arena that they are unable to go on with the gradually complex strategies used by cheats to jeopardize important properties: Cybercriminals may readily counteract a set of predetermined levels [6], [7] and fixed criteria are useless for identifying developing risks and adapting to previously undisclosed fraudulent transactions.

The miss of information to examine is another significant drawback of rule-based detecting systems more inventive the fraudulent strategy, the less the info you will get in examined trades [8]. This dearth of information might indicate that valuable data are not being gathered and saved, that data is available although lacking crucial points [7], or that data cannot relate to particular other info.

Many methods have been created and tried over time to increase the efficacy of rule-based detection strategies. However, recent trends indicate that deploying analytics regularly on a flexible data architecture and reliable machine learning algorithms might yield promising outcomes.

B. On Machine Learning-based e-Banking Fraud Detection

Recently, machine learning Fraud Detection has risen to prominence [9] [10] [11] [12]. Because of its more accurate findings, the anti-fraud domain is shifting from rule-based fraud identification to ML fraud detection. We present here some online fraud detection studies.

By using supervised machine learning methods, [13] have wanted to construct a transactional fraud detection algorithm capable of efficiently classifying an online transaction as illegitimate or legitimate. A credit-card fraud classifier was created utilizing three supervised machine-learning (ML) algorithms. SVM, LR (logistic regression), and neural networks are among these methods. All the classifiers attain about the same classification accuracy. The results show that the support vector machine beats the others.

Along with this, two algorithms, namely XGBoost, and Fully Connected Neural Network (FCNN), whose AUC merits may reach 0.912 and 0.969 correspondingly, have been developed by [14]. In the meantime, they have developed an interactive method for identifying online transaction fraud relying on the XGBoost model to evaluate submitted transaction data autonomously and provide customers with fraud detection findings. On the other side, to increase detection performance and quicken the convergence of identification, [15] has suggested an online transactional fraud detection approach using unbalanced data relying on the semantic integration of wo unsupervised learners such as an artificial bee colony model and k-means. In the suggested method, ABC functions as a secondary classification level to handle the k-means classifier's inability to investigate the real bunches since it is susceptible to the beginning circumstance. The experiment results showed up to 100% True positive and less than 2% False Positive. In the same context [16] have offered a tailored alert model for detecting fraud in online transactions by mining a set of instances in each customer's regular transaction log. The suggested methodology segmented every consumer's log into transactions extracted a collection of chronological sequence arrangements and used it to identify if a novel transaction is malicious. The entering transaction is separated within many windows, and an alert is raised if the typical behaviors are not discovered in the subsequent windows. According to the experimental outcomes, the suggested approach beats the rule-based paradigm and the Markov chain method.

Moreover, FinDeepBehaviorCluster has been proposed by [17]. They have used temporal attention-based Bi-LSTM to determine sequential embed and handled click data in real-time as an event sequence to exploit the behavior sequence data. Handmade features reflecting domain expertise are produced to improve the system's interpretability. By integrating the two sorts of traits, a hybrid behavior interpretation has been created. Then, to group transactions with similar behavior, a pHDBSCAN (i.e., GPU-powered HDBSCAN) is used. The results show that FinDeepBehaviorCluster successfully detects lacking suspicious transactions having excellent business value. By merging machine learning with big data analytics

tools, [18] have presented a robust method for detecting fraud in online-based transactions. To notice whether electronic transfer behaviors are aberrant, the big data of internet-based e-transactions, which includes (credit card details data and trading), is first refined in the transaction pre-processing stage module, and then transferred to a rule-based specialist system module, which would be achieved with Spark streaming and divvied up platform Kafka. The regular records from the expert system module are then applied in the machine learning fraud prevention module to execute behavioral analysis via DT (Decision Tree), CNN, and SVM algorithms. The findings exhibit that the proposed strategy produces satisfactory results. Also, to identify Internet financial fraud, [8] have proposed a sophisticated and scattered Big Data approach. They have used Hadoop and Spark GraphX to identify and express every vertex's topologic feature in a dense lowly-dimensional vector using the graph embedding technique Node2Vec. The suggested approach seeks to anticipate the dataset's spurious entries. The findings indicate that the proposed strategy enhances the precision and accuracy of Online fraudulent transaction detection systems. In that same vein [19] have created a real-time scam detection for credit cards system utilizing big data technologies such as Microsoft Azure. The given outcomes are pretty accurate by applying a variety of ML learners, such as Extreme Random Trees and Stochastic Gradient Descent.

Recently, the isolation forest learner has been applied in online banking transactions fraud detection, given its reputation as one of the most powerful algorithms. In fact, [20] have examined two unsupervised learners for CCFD i.e. credit card fraud detection (isolation forest (IForest) and local outlier factor (LOF)). When comparing precision and recalls for the two models, the findings show that Isolation Forest beats the local outlier factor. Additionally, the fraud detection percentage is about 0.27, whereas the LOF (local outlier factor) discovery rate is barely 0.02. The accuracy of the Isolation Forest is 0.99774 higher than that of the local outlier factor. Similarly, the IForest and LOF techniques were employed by [21] to detect fraudulent credit card transactions. The experiments provide good results.

All the studies discussed here are fascinating and revolve around fraud detection in large data circumstances. They offer trustworthy and promising prediction algorithms for preventing fraud. We present the comparison of all these models in the Table I.

TABLE I. PREVIOUS FINDINGS FOR OTHER STUDIES

Paper	Used dataset	Techniques used	Performance	Limits
[13]	Credit card dataset	SVM, LR, and neural networks	The support vector machine beats the others	The precision of the ANN is around twelve percent less than that of both of the models
[14]	Online transactions	XGBoost, and Fully Connected Neural Network (FCNN)	XGBoost reaches 0.912 and FCNN 0.969	The system can't identify malicious transactions in real-time as they occur
[15]	Online transactions	artificial bee colony model and k-means	Results showed up to 100% True positive and less than 2% False Positive	Quadratic Discriminant Analysis give the fewer accuracy
[16]	Online transactions	Tailored alert model for detecting fraud in online transactions	The suggested approach beats the rule-based paradigm and the Markov chain method.	The suggested methodology detects fraud by using regular patterns; however, it will only identify scams when individuals display considerably different trading habits than typical.
[17]	real-world e-commerce transaction data	temporal attention-based Bi-LSTM, pHDBSCAN	Results show that the proposed method successfully detects lacking suspicious transactions having excellent business value.	Unable to identify low frequency of fraudulent transaction
[18]	internet-based e-transactions (credit card details data and trading)	Spark streaming and Kafka. DT, support vector machine, and CNN	The findings show that the proposed strategy produces satisfactory results.	The outcomes need to be improved
[8]	Credit card dataset	Spark GraphX, Hadoop, and graph embedding technique Node2Vec	The findings indicate that the proposed strategy enhances the precision and accuracy of Online fraudulent transaction detection systems.	The suggested model will be enhanced to successfully learn the newly generated features, resulting in better identification of fraud.
[19]	Credit card dataset	Microsoft Azure, Extreme Random Trees, and Stochastic Gradient Descent.	Good accuracy	Does not handle the class imbalance problem
[20]	Credit card dataset	IForest and LOF	The findings show that Isolation Forest beats the local outlier factor within 0.99774 of accuracy. The fraud detection percentage is about 0.27, whereas the LOF discovery rate is scarcely 0.02.	The LOF learner yield low performance
[21]	Credit card dataset	IForest and LOF	The experiments provide good results.	LOF give the worst results

The upcoming section will highlight the comparative study of this literature reviewed methods for online fraud detection and present the motivations of our paper.

III. Summary and Motivations

Based on our literature review analysis in the previous section, we noticed that researchers have proposed several machine Learning approaches particularly supervised ones involving SVM, LR, DT, and NB algorithms, for detecting online transaction fraud. The majority of the approaches examined have shown to be beneficial in the process; nonetheless, due to changes in the fraudster's behavioral patterns, real-time fraud detection is always difficult, and algorithms fail to give better accuracy.

As the outcomes reveal, systems for detecting fraud that utilize SVM and LR offer good accuracy yet suffer from considerable overhead when handling huge datasets. Additionally, because the fraudulent act is shifting, these learners are just assisting in learning current trends in fraud. From another viewpoint, ANN, decision tree, and NB provide moderate accuracy and mid-scope at the expense of high prices.

Another limitation of these related studies is that most of them establish a profile of regular cases and then detect anything that does not fall within the usual profile as an abnormality; leading to misclassification, a high false positive rate, and also, they are not adept at handling real-time detection. In contrast with that, IForest segregates observations by picking a property and then erratically determining a splitting point between the selected property's maximum and minimum values [22]. The amount of splits required to isolate a trial equals the path length from the root node to the ending node [23]; by giving high fraud detection accuracy over large datasets, with the least false positive rates.

This study suggests a new end-to-end real-time architecture for online transaction fraud detection based on isolation forest learners. Combining the advantages of big data analytics tools and the unsupervised isolation forest with the aim to overcome the existing approaches' limitations and dealing with real-time detection and prevention of digital transactions while minimizing false positive rate, and false alarms, regulating latency in addition to speed, and dependability.

IV. Isolation Forest

Isolation Forest is defined as an unsupervised ML learner. It employs a similar technique as the (RF) Random Forest algorithm and is based on the notion of decision trees. Rather than using the typical properties of data points, the isolation forest algorithm's basic idea and approach are to detect abnormalities — for example, fraudulent transactions [24] [25].

Isolation forest outperforms other techniques in anomaly detection algorithms due to several advantages. First, it requires tiny samples from considerable datasets to generate an anomaly detection algorithm, making it rapid and robust. Secondly, no examples of abnormalities in the training sample are required. Furthermore, the tree depth serves as the foundation for its distance threshold for detecting anomalies independent of the sample dimensionality scale. It may function as both a supervised and unsupervised learner, and its goal is for irregularities to be less frequent than everyday observations and to differ from their values.

To build the IForest (Isolation Forest), determine the amount of (Itrees) isolation trees within the forest. Next, for every isolation tree, the following procedures are taken [26] [27]:

- Select n instances at random from the training dataset.
- Pick an attribute at random to divide on.
- At random, select a separated value from a uniformly distributed covering the minimum to the most significant rate of the feature set in Step 2.

Assuming a dataset has n instances, h(x) is the route length as x. The average path length c(n) is afterward used to normalize the value of the path length h(x). As Itrees have the same shape as the BStree (Binary Search Tree), the following equation is used to obtain the value of c(n), where H(i) is the harmonical number that may be obtained via [28] [29] :

$$c(n) = 2H(n-1) - \frac{2(n-1)}{n} \quad (1)$$

The abnormality score within each data point x in a database with n occurrences is obtained by using the following:

$$s(x,n) = \ 2^{-\frac{E(h(x))}{c(n)}} \quad (2)$$

with (E(h(x)) is the mean of h(x) across a set of Itrees).

The nearer a data point's anomaly score is near 1, the more likely that data point is an outlier. On the other hand, the records point is more probable to be regular if the abnormality count is near zero.

V. End-to-End Fraud Detection Solution Architecture

This section proposes a real-time scalable architecture for preventing and detecting fraud in online transactions using big data analytics algorithms to improve the capacity to manage highly complex online transaction fraud instances. In this part, we will present a fraud detection pipeline as a sequence of steps applied to every transaction to mitigate the risk of fraud occurrence. This pipeline will consequently drive the suggested architecture and the technology stack used for its implementation.

A. Fraud Detection Pipeline

Let's say the banking account provider receives an authorization request for a transaction. Initially, the Online Detecting Fraud system captures the transaction data and its context in real-time. To prevent fraud, deterministic rules are positioned as barriers that should be imperatively checked before effectively executing the transaction. Given that these rules are implemented as part of the transaction, criteria such as low latency should be a real concern. Therefore, enforcing these rules must be performed in milliseconds. Otherwise, the customers will notice a significant delay while interacting with the bank application. Once these barriers are overcome, the customer transaction is executed. Next, we move forward with

fraud detection using more sophisticated and non-deterministic data analytics techniques.

At this stage, the goal is to detect suspicious transactions based on customers' past interactions with the bank's application. To see these transactions, customer data would be processed in real-time and fed to a pre-trained isolation forest model. This model makes predictions and produces suspicious transactions with an associated score. Transactions with a score over a predefined threshold would be displayed in a fraud monitoring application for human supervisors who will investigate customer behavior to confirm or reject those cases. The transaction monitoring agents might perform some curative actions and notify account holders of the occurrence of these high fraud-risk transactions by "mobile app alerts, e-mail or SMS." The fraudulent instances observed by the transaction monitoring and customer care departments are gathered, and the associated transactions in the database are tagged as suspicious. To sum up, any customer transaction will go through the pipeline below in Fig. 1:

Fig. 1. Customer transaction pipeline.

Each of the presented steps has different prerequisites to balance user experience and prevention from potential fraudsters. Consequently, implementation choices and used technologies were driven by these requirements. The Table II presents each step, along with its prerequisites and implementation choices:

TABLE II. FRAUD DETECTION STEPS WITH THERE PREREQUISITES

Layer	Description & prerequisites	Implementation choices
Events streaming	Refers to events streaming from digital banking applications. This component must publish events as soon as they occur.	Kafka-connect. Kafka producer API
Data capture	Refers to events captured in a resilient way as well as making them available to different consumers.	Apache Kafka
Fraud prevention	Refers to real-time fraud prevention while transactions are in motion. This step must respond with a significantly reduced latency, given that the end-user would be blocked until this prevention is performed.	Apache Kafka Streams
Fraud detection	Refers to detection of fraud in a non-deterministic way, affecting a score to each transaction and persisting information about suspicious transactions.	Apache Spark Spark Streaming H2O PostgreSQL
Monitoring	Refers to making potential fraud alerts available to human supervisors that could analyze and eventually contact end-users and perform curative actions accordingly.	React NodeJS
Alerting	Refers to raising alerts once a suspicious alert is confirmed to be fraudulent. These alerts could be consumed afterward by third-party consumers for actions such as account blocking and SMS notifications...	Kafka-connect. Kafka producer API

B. End-to-End Solution Architecture

The Fig. 2 below presents an overview of the suggested architecture after gluing together the building blocks exposed previously:

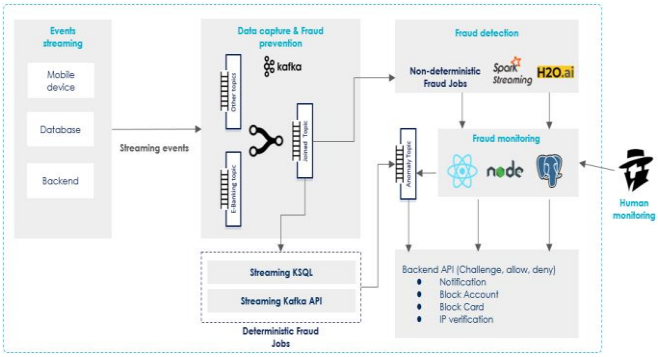

Fig. 2. End to end real time fraud detection architectur.

The prevention layer will be built with Kafka and KSQL. Kafka, the most ubiquitous and extensible stream processing platform, has been optimized for real-time use [30] [31]. KSQL, on the other hand, is a continuous query language. While it may be used for interactive data exploration, its primary aim is to construct stream processing apps. Our architectural scheme is as follows; Large data transactions originate from various sources, including websites, neobanks, social media, etc. These transactions are obtained in real-time using Apache Kafka and KSQL in the form of a stream. For instance –, if we have the same account number as that of the previous transaction at a different location in fewer than ten-minute period later, the system will deem it suspicious and reject it instantly without sending a verification email or SMS to the account's owner.

Real-time fraud detection consists of several layers, including a real-time transaction ingestion layer, a processing layer for handling massive amounts of information in storage for increased reliability and fault-tolerant, and fraud notifications via visual representation. First, the vast data of online transactions is ingested. Then, the processing layer retrieves the transactions in real time, which can handle the transaction data quickly and efficiently. This layer notably depicts two commonly used techniques. Spark streaming and Sparkling water for deploying the predictive model and its integration with the Spark distributed processing engine.

On the other hand, an isolation forest is used to predict the degree of fraud and identify it as accurately as possible in the shortest period. To verify whether a transaction is illegal, isolation forest learns the model from the account holder's behavioral patterns. We examine the location and time gap among different transactions, the frequency of transactions, and other criteria while regulating the account holder's history of transactions. The transactional data will then be saved and utilized for monitoring in a frontend application, which exposes visualizations and curative actions connected with backend APIs.

C. Solution Infrastructure Deployment

At the core of this architecture implementation, we relied on a distributed cluster on which model training and spark streaming data processing and inference jobs were deployed. In addition, other components, namely Kafka and fraud monitoring applications, were deployed separately on other servers. The Table III shows the servers used for each element:

TABLE III. USED SERVERS

Component	Servers / Characteristics
Spark streaming / H2O	Driver : CPU: 1 core RAM: 4 Go Storage: 50 Go
	Worker 1 : CPU: 2 cores RAM: 8 Go Storage: 50 Go
	Worker 2 : CPU : 2 cores RAM : 8 Go Storage : 50 Go
	Worker 3 : CPU : 2 cores RAM : 8 Go Storage : 50 Go
Kafka	Broker 1 : CPU : 2 cores RAM : 8 Go Storage : 50 Go
	Broker 2 : CPU : 2 cores RAM : 8 Go Storage : 50 Go
	Broker 3 : CPU : 2 cores RAM : 8 Go Storage : 50 Go
Monitoring application	Application server / Database: CPU : 2 cores RAM : 8 Go Storage : 50 Go

In the next section, we will focus on the used dataset as well as the model implementation (ie spark streaming job and its integration with H2O isolation forest implementation). For these two components we will expose the approach along with key results and metrics.

VI. SIMULATION AND RESULTS

This section depicts the database and the evaluation criteria that were utilized in our study. The outcomes of the suggested method's experiments are then provided.

A. Dataset

The database used in our work contains online transactions generated with an approach that simulated real customer behavior. The generated dataset contains more than 100 million rows following the structure below:

- User_id: identifier of the user connected to the portal.
- Account: account number of the customer connected to the portal.
- Event_type: type of event captured by the audit trail.

- Event_payload: payload containing event attributes.
- Event_description: descriptive text of the event
- Device_id: mac address of the device used for the action.
- Ip_address: IP address
- Timestamp.

The event attribute reflects various actions that customers could typically perform in a digital banking platform:

- LOGIN_ATTEMPT
- LOGIN_SUCCESSFULL
- LOGIN_FAILED
- LOGOUT
- VIEW_ACCOUNT_BALANCE
- VIEW_ACCOUNT_HISTORY
- VIEW_ACCOUNT_OPERATION
- MONEY_TRANSFERT
- ADD_BENEFICIARY
- REMOVE_BENEFICIARY
- PROVISION_CARD
- BILL_PAYMENT
- VIEW_CONTRACT
- VIEW_CARD

To reflect real customer behavior, data was generated concerning the sequence of events that could occur from a user interacting with the bank application. For example, the interaction sequence should start with LOGIN_ATTEMPT event followed by LOGIN_SUCCESSFUL or LOGIN_FAILED. Once the customer is logged in, they can view the account balance, add beneficiary to make money transfer to them, pay bills, or any other event reflecting the exposed services by the bank. To integrate suspicious events, the data generation script randomly picks some users for which money transfers are performed at an unusual rate or failed login attempts are performed from unknown devices. Those fraudulent transactions are then labeled and saved separately as a baseline for later model evaluation. The generated data served for model training and was published afterward to a Kafka topic using scripts relying on Kafka producer API.

B. Model Training and Inference

Before model training, generated events were processed as part of the feature engineering step to extract relevant features for our context. Below are key features used to train the model:

- User_id
- Account
- Login_attempts_count
- Last_login_timestamp
- Last_transaction_amount
- Beneficiary_account
- Transactions_sum
- Transaction_to_max
- Device_id
- Device_id_last_timestamp
- Device_id_bill_payment

Once the features were extracted, the model was trained on the provisioned cluster using h2O integrated with an Apache Spark job. The integration was done using the Sparkling water package, which was installed and used afterward to create an h2OContext employed to train our model in a distributed way. In our model training, we sought to optimize isolation forest hyperparameters such as the number of trees and tree depth that would allow us to detect all the fraudulent transactions while minimizing false positives. During our training, we reached an optimal performance with values of 200 as the number of trees and 18 as tree depth. The performance of our model with these parameters is exposed through classification metrics in the section below.

C. Experimental Criteria

During this work, we used our dataset partitioned into five sets to train the isolation forest. While the training set is made up of 80% of the data.

The experiment outcomes are assessed using accuracy, the F1-S, precision, and recall, as specified in Table IV. The Accuracy metric represents the overall performance of fraud detection. Precision is another word for a predictive value that is positive. A true positive rate is identical to the recall. The harmonious mean of (recall - accuracy) is the "F1-score". The (True Positive i,e. TP) alludes to the amount of accurately anticipated suspicious transactions within all right suspect transactions, false positive (i.e. FP) alludes to the total of regular transactions that are wrongly identified as suspicious, (TN i.e. true negative) relates to several precisely indicated normal actions for all right regular operations, and false negative i.e. FN refers to the amount of suspicious transactions that are erroneously marked as regular ones.

TABLE IV. EVALUATIONS METRICS

Performance metrics	Formulas	
Precision:	$\mathbf{TP}/\mathbf{TP}+\mathbf{FP}$	(3)
Recall:	$\mathbf{TP}/(\mathbf{TP}+\mathbf{FN})$	(4)
Accuracy:	$((\mathbf{TP}+\mathbf{TN})/(\mathbf{TP}+\mathbf{TN}+\mathbf{FP}+\mathbf{FN}))$	(5)
F1 score:	$\mathbf{2}\times\frac{(\mathbf{precision}\times\mathbf{recall})}{(\mathbf{precision}+\mathbf{recall})}$	(6)

The ROC curve is accompanied by a graphical display that compares the TP to the FP at several criteria. We additionally employ the AUC i.e., area under the ROC curve alongside the abovementioned measurements as a comprehensive performance measure. Since it does not depend on a criterion

value, the AUC is considered a superior performance metric to accuracy. The nearer the AUC number is to one, the finer a model's overall efficiency.

D. Experimental Results

Utilizing our private dataset, we experimented with the isolation forest using Python and a sparkling water engine. The output model was packaged and integrated with Spark Streaming through Sparkling Water to perform fraud detection in real-time as per previously exposed architecture. Table V and Table VI summarize the obtained results after completing the training iteration.

TABLE V. TRAINING ITERATIONS

	Events	transactions	Labeled Fraud attempts
Iteration 1	20000000	187234	151
Iteration 2	20000000	234567	213
Iteration 3	20000000	198654	195
Iteration 4	20000000	272647	286
Iteration 5	20000000	324546	323

The Table VI presents the mean of critical metrics after training the model against the above mentioned dataset.

TABLE VI. MODEL METRICS

	Accuracy	Precision	Recall	F1-score
Metrics	0,99	0,87	0,97	0,91

VII. DISCUSSION

We discussed the procedures needed to set up an online transaction fraud detection architecture in real-time utilizing Spark, Kafka, and h2O in this article. After that, the experimentation kit was utilized to build an isolation forest-based machine-learning model. The system was able to expedite its analysis by combining real-time and batch-time analysis, yielding promising results. We also looked at the efficacy of the isolation forest model. Our model's performance has been evaluated using four distinct metrics such as accuracy, recall, f1-score, and precision.

On top of that, we compare the outcomes of the presented study's work to the current fraud detection techniques. As an example [32] have employed the SVM, apriori algorithm, and SVMIG (i,e SVM with Information Gain) to handle transactional fraud detection. The outcomes give an accuracy of 0.94. Authors of [33] have applied six ML learners involving LR, XGBoost, DT, SVM, ET (Extra Tree), and the RF on the European cardholder database. These learners were integrated with AdaBoost to boost their performance of fraud classification. The experiments yield more than 98% of accuracy.

Along with that, [34] have suggested an hybrid model named AED-LGB (AE with probabilistic LGBM) to detect fraudulent transactions using real word transactional dataset. Experimental evaluation shown around 0.98 of accuracy. Also, [35] have utilized the Naïve Bayes Based classifier for transaction fraud detection on a credit card dataset. They have compared the proposed model with the state-of-the-art ML methods. The finding reveals that the NB beat the others with an accuracy of 0.97.

In line with the findings in the present article and the findings in current state-of-the-art systems for detecting fraud this study provides a high digital transaction fraud detection accuracy (0.99) using relevant big data analysis tools to speed up model analysis and training and also to detect suspects' transactions as soon as they arise. in contrast to the research published in [32], [33], [34], and [35].

VIII. CONCLUSION AND FUTURE WORK

Fraud screening is critical in digital transactions, and the most significant difficulty is the financial burden of fraud if it is investigated, detected, or prevented. As though transactions happen in real-time, there is a need for a method that takes no time and remains as effective as the scope and structure of the bank that handles it. In this study, we presented an end-to-end real-time architecture using behavioral analysis for digital transactions fraud detection centered on combining the isolation forest algorithm and current big data analytics technologies. This technique aims to regulate latency, speed, and reliability by employing batch processing to give complete and precise interpretations of batch sets alongside immediate stream analysis to provide observations of live data. In our scenario, the batch layer handles data preparation and model training providing effective outcomes on a real dataset. The F1-score and recall of our model is about 91% and 97% correspondingly.

We want to do more study in two areas in the further works. The first looks at the computing requirements of a real-time suspicion detection technology. The second goal is to investigate the use of increasingly sophisticated ML techniques and the combination of DL (deep learning) algorithms and relevant big data tools in fraud detection.

REFERENCES

[1] Y. Gahi and I. El Alaoui, “A secure multi-user database-as-a-service approach for cloud computing privacy,” Procedia Computer Science, vol. 160, pp. 811–818, 2019.

[2] A. Singla and H. Jangir, “A Comparative Approach to Predictive Analytics with Machine Learning for Fraud Detection of Realtime Financial Data,” in 2020 International Conference on Emerging Trends in Communication, Control and Computing (ICONC3), Lakshmangarh, India: IEEE, Feb. 2020, pp. 1–4. doi: 10.1109/ICONC345789.2020.9117435.

[3] E. F. Malik, K. W. Khaw, B. Belaton, W. P. Wong, and X. Chew, “Credit Card Fraud Detection Using a New Hybrid Machine Learning Architecture,” Mathematics, vol. 10, no. 9, p. 1480, Apr. 2022, doi: 10.3390/math10091480.

[4] I. Achituve, S. Kraus, and J. Goldberger, “Interpretable Online Banking Fraud Detection Based On Hierarchical Attention Mechanism,” in 2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP), Pittsburgh, PA, USA: IEEE, Oct. 2019, pp. 1–6. doi: 10.1109/MLSP.2019.8918896.

[5] P. Vanini, S. Rossi, E. Zvizdic, and T. Domenig, "Online payment fraud: from anomaly detection to risk management," Financial Innovation, vol. 9, no. 1, p. 66, Mar. 2023, doi: 10.1186/s40854-023-00470-w.

[6] Y. Gahi, M. Guennoun, Z. Guennoun, and K. El-Khatib, "Encrypted processes for oblivious data retrieval," in 2011 International Conference for Internet Technology and Secured Transactions, IEEE, 2011, pp. 514–518.

[7] M. Aschi, S. Bonura, N. Masi, D. Messina, and D. Profeta, "Cybersecurity and Fraud Detection in Financial Transactions," in Big Data and Artificial Intelligence in Digital Finance: Increasing Personalization and Trust in Digital Finance using Big Data and AI, J. Soldatos and D. Kyriazis, Eds., Cham: Springer International Publishing, 2022, pp. 269–278. doi: 10.1007/978-3-030-94590-9_15.

[8] H. Zhou, G. Sun, S. Fu, L. Wang, J. Hu, and Y. Gao, "Internet Financial Fraud Detection Based on a Distributed Big Data Approach With Node2vec," IEEE Access, vol. 9, pp. 43378–43386, 2021, doi: 10.1109/ACCESS.2021.3062467.

[9] H. Wang, W. Wang, Y. Liu, and B. Alidaee, "Integrating Machine Learning Algorithms With Quantum Annealing Solvers for Online Fraud Detection," IEEE Access, vol. 10, pp. 75908–75917, 2022, doi: 10.1109/ACCESS.2022.3190897.

[10] Z. Ullah and M. Jamjoom, "A smart secured framework for detecting and averting online recruitment fraud using ensemble machine learning techniques," PeerJ Comput. Sci., vol. 9, p. e1234, Feb. 2023, doi: 10.7717/peerj-cs.1234.

[11] M. Z. Khan et al., "The Performance Analysis of Machine Learning Algorithms for Credit Card Fraud Detection," International Journal of Online and Biomedical Engineering (iJOE), vol. 19, no. 03, Art. no. 03, Mar. 2023, doi: 10.3991/ijoe.v19i03.35331.

[12] Y. Gahi and I. El Alaoui, "Machine learning and deep learning models for big data issues," Machine Intelligence and Big Data Analytics for Cybersecurity Applications, pp. 29–49, 2021.

[13] S. Khan, A. Alourani, B. Mishra, A. Ali, and M. Kamal, "Developing a Credit Card Fraud Detection Model using Machine Learning Approaches," IJACSA, vol. 13, no. 3, 2022, doi: 10.14569/IJACSA.2022.0130350.

[14] B. Liu, X. Chen, and K. Yu, "Online Transaction Fraud Detection System Based on Machine Learning," J. Phys.: Conf. Ser., vol. 2023, no. 1, p. 012054, Sep. 2021, doi: 10.1088/1742-6596/2023/1/012054.

[15] S. M. Darwish, "A bio-inspired credit card fraud detection model based on user behavior analysis suitable for business management in electronic banking," J Ambient Intell Human Comput, vol. 11, no. 11, pp. 4873–4887, Nov. 2020, doi: 10.1007/s12652-020-01759-9.

[16] J. Kim, H. Jung, and W. Kim, "Sequential Pattern Mining Approach for Personalized Fraudulent Transaction Detection in Online Banking," Sustainability, vol. 14, no. 15, p. 9791, Aug. 2022, doi: 10.3390/su14159791.

[17] W. Min, W. Liang, H. Yin, Z. Wang, M. Li, and A. Lal, "Explainable Deep Behavioral Sequence Clustering for Transaction Fraud Detection." arXiv, Jan. 11, 2021. Accessed: Dec. 29, 2022. [Online]. Available: http://arxiv.org/abs/2101.04285

[18] H. Zhou, G. Sun, S. Fu, W. Jiang, and J. Xue, "A Scalable Approach for Fraud Detection in Online E-Commerce Transactions with Big Data Analytics," Computers, Materials & Continua, vol. 60, no. 1, pp. 179–192, 2019, doi: 10.32604/cmc.2019.05214.

[19] L. Sahai and K. Gursoy, "Real-time credit card fraud detection," 2019, doi: 10.7282/T3-QE71-9791.

[20] H. Rajeev and U. Devi, "Detection of Credit Card Fraud Using Isolation Forest Algorithm," in Pervasive Computing and Social Networking, G. Ranganathan, R. Bestak, R. Palanisamy, and Á. Rocha, Eds., in Lecture Notes in Networks and Systems, vol. 317. Singapore: Springer Nature Singapore, 2022, pp. 23–34. doi: 10.1007/978-981-16-5640-8_3.

[21] V. Palekar, S. Kharade, H. Zade, S. Ali, K. Kamble, and S. Ambatkar, "Credit Card Fraud Detection Using Isolation Forest," vol. 07, no. 03, 2020.

[22] L. V. Utkin, A. Y. Ageev, and A. V. Konstantinov, "Improved Anomaly Detection by Using the Attention-Based Isolation Forest." arXiv, Oct. 05, 2022. Accessed: May 05, 2023. [Online]. Available: http://arxiv.org/abs/2210.02558

[23] D. Prusti, D. Das, and S. K. Rath, "Credit Card Fraud Detection Technique by Applying Graph Database Model," Arab J Sci Eng, vol. 46, no. 9, pp. 1–20, Sep. 2021, doi: 10.1007/s13369-021-05682-9.

[24] H. Bodepudi, "Credit Card Fraud Detection Using Unsupervised Machine Learning Algorithms," International Journal of Computer Trends and Technology, vol. 69, pp. 1–3, Aug. 2021, doi: 10.14445/22312803/IJCTT-V69I8P101.

[25] Y.-F. Zhang, H.-L. Lu, H.-F. Lin, X.-C. Qiao, and H. Zheng, "The Optimized Anomaly Detection Models Based on an Approach of Dealing with Imbalanced Dataset for Credit Card Fraud Detection," Mobile Information Systems, vol. 2022, p. e8027903, Apr. 2022, doi: 10.1155/2022/8027903.

[26] M. T. R. Laskar et al., "Extending Isolation Forest for Anomaly Detection in Big Data via K-Means." arXiv, Apr. 27, 2021. Accessed: Dec. 26, 2022. [Online]. Available: http://arxiv.org/abs/2104.13190

[27] Y. Xu, H. Dong, M. Zhou, J. Xing, X. Li, and J. Yu, "Improved Isolation Forest Algorithm for Anomaly Test Data Detection," Journal of Computer and Communications, vol. 9, no. 8, Art. no. 8, Aug. 2021, doi: 10.4236/jcc.2021.98004.

[28] J. Lesouple, C. Baudoin, M. Spigai, and J.-Y. Tourneret, "Generalized isolation forest for anomaly detection," Pattern Recognition Letters, vol. 149, pp. 109–119, Sep. 2021, doi: 10.1016/j.patrec.2021.05.022.

[29] Y. Chabchoub, M. U. Togbe, A. Boly, and R. Chiky, "An In-Depth Study and Improvement of Isolation Forest," IEEE Access, vol. 10, pp. 10219–10237, 2022, doi: 10.1109/ACCESS.2022.3144425.

[30] "Apache Kafka," Apache Kafka. https://kafka.apache.org/intro (accessed Jan. 06, 2023).

[31] I. El Alaoui, G. Youssef, R. Messoussi, A. Todoskoff, and A. Kobi, "Big Data Analytics: A Comparison of Tools and Applications," in Lecture Notes in Networks and Systems, 2018, pp. 587–601. doi: 10.1007/978-3-319-74500-8_54.

[32] K. Poongodi and D. Kumar, "Support Vector Machine with Information Gain Based Classification for Credit Card Fraud Detection System," IAJIT, vol. 18, no. 2, 2021, doi: 10.34028/iajit/18/2/8.

[33] E. Ileberi, Y. Sun, and Z. Wang, "Performance Evaluation of Machine Learning Methods for Credit Card Fraud Detection Using SMOTE and AdaBoost," IEEE Access, vol. 9, pp. 165286–165294, 2021, doi: 10.1109/ACCESS.2021.3134330.

[34] N. S. Alfaiz and S. M. Fati, "Enhanced Credit Card Fraud Detection Model Using Machine Learning," Electronics, vol. 11, no. 4, Art. no. 4, Jan. 2022, doi: 10.3390/electronics11040662.

[35] R. O. Ogundokun, S. Misra, O. J. Fatigun, and J. K. Adeniyi, "Naïve Bayes Based Classifier for Credit Card Fraud Discovery," in Information Systems, Springer, Cham, 2022, pp. 515–526. doi: 10.1007/978-3-030-95947-0_37.