International Journal of Multidisciplinary Science and Innovation (IJMSI)
Website: https://ijmsi.in/
ISSN: 3107-5754
Vol. 2, Issue 1, Jan-Jun, 2026
Page No.: 71-79

Research

Design and Implementation of a Real-Time Fraud Detection System Leveraging Machine Learning

Nimisha Pandey*, Umar Badr Shafeeque

Computer Science & Technology, Azad Institute of Engineering and Technology, Dr. A.P.J. Abdul Kalam Technical University, Lucknow, India

Corresponding Author:
Nimisha Pandey

***Email:** NA*

***DOI:** 10.62896/ijmsi.2.1.09*

***Conflict of interest:** NIL*

Article History
Received: 08/02/2026
Accepted: 10/03/2026
Published: 13/03/2026

Abstract:
The threat of fraud has emerged as a pressing concern for financial institutions, e-commerce platforms, and a wide range of online service providers in an era when digital transactions and online interactions have become commonplace. Businesses and consumers alike face significant risks from fraudulent activities like account takeovers, identity theft, and credit card fraud, which can result in substantial financial losses and compromised personal information. In the steadily advancing scene of computerized exchanges and online co-operations, the predominance of fake activities has required the advancement of hearty extortion discovery systems. This paper presents a clever way to deal with planning an effective Fraud Detection System (FDS) for constant applications, utilizing progressed Machine Learning calculations. To improve the accuracy and effectiveness of fraud detection, the proposed system combines a number of machine learning (ML) methods, such as supervised learning algorithms like Random Forests, Gradient Boosting Machines, and Support Vector Machines, and unsupervised learning methods like Auto encoders and Clustering algorithms. In order to deal with the diverse and imbalanced nature of transaction data, the system makes use of feature engineering and data preprocessing strategies. Continuous handling abilities are accomplished using streaming information systems and adaptable ML models, guaranteeing convenient ID and alleviation of deceitful exercises. Metrics like precision, recall, and F1-score are used for performance evaluation. The results show that compared to traditional methods, there are significant improvements in detection rates and fewer false positives. The proposed FDS framework not just works on the unwavering quality of misrepresentation recognition continuously situations yet in addition offers bits of knowledge into the versatile idea of misrepresentation designs, preparing for stronger and proactive safety efforts in monetary and web based business spaces. Future research could investigate the incorporation of cutting edge profound learning models and further advancement of the framework design to deal with considerably bigger datasets and more mind boggling extortion designs.

Keywords: Machine learning, Minimum losses, E-commerce, Real-time fraud, AWS (Amazon Web Services), FDS

1. Introduction

Real-time fraud detection is the most common way of distinguishing false movement as it works out, progressively. This is done by analyzing and monitoring data in real-time and applying machine learning algorithms to detect patterns that indicate fraud. Real-time fraud detection systems are used in a wide range of industries, such as banking, e-commerce, insurance, and telecommunications. Here are some of the AWS services that can be used for real-time fraud detection: AWS Lambda, Aws Kinesis, MSK, Aws DynamoDB and Many More. Real-time applications, such as fraud detection, decision latency are a critical performance metric, as delays in decision-making can have serious consequences. For example, if a fraud detection system takes too long to process a transaction, it may not be able to detect and prevent fraudulent activity in real-time, which can lead to financial losses or reputational damage [1] [2].

One of the key benefits of real-time fraud detection is the ability to prevent fraudulent transactions from occurring, by detecting and stopping them in real-time. This can help to minimize losses, protect customers from financial harm, and improve the overall security of the system. In addition, real-time fraud. AWS (Amazon Web Services) offers a wide range of tools and services that can be used to build a real-time fraud detection system in the cloud. In today's digital era, the threat of fraud has become an increasingly significant concern for individuals, businesses, and organizations. Fraudulent activities can lead to substantial financial losses, compromised security, and damaged reputations. To combat these risks, the development of robust fraud detection systems has become paramount [3]. This project aims to provide an introduction to the design of a Fraud Detection System (FDS). By leveraging advanced technologies such as machine learning, data analysis, and pattern recognition, an effective FDS can identify and mitigate fraudulent activities, ensuring the integrity and security of systems and processes.The primary objective of this project is to outline the key components, considerations, and methodologies involved in designing a fraud detection system. From data collection and preprocessing to modeling and deployment, each step in the process is crucial in creating an efficient and reliable fraud detection solution.

AWS CloudFormation is a service that helps you model and set up your AWS resources so that you can spend less time managing those resources and more time focusing on your applications that run in AWS. You create a template that describes all the AWS resources that you want (like Amazon EC2 instances or Amazon RDS DB instances), and CloudFormation takes care of provisioning and configuring those resources for you. You don't need to individually create and configure AWS resources and figure out what's dependent on what; CloudFormation handles that. The following scenarios demonstrate how CloudFormation can help.

For a scalable web application that also includes a backend database, you might use an Auto Scaling group, an Elastic Load Balancing load balancer, and an Amazon Relational Database Service database instance. You might use each individual service to provision these resources and after you create the resources, you would have to configure them to work together. All these tasks can add complexity and time before you even get your application up and running.

Instead, you can create a CloudFormation template or modify an existing one. A *template* describes all your resources and their properties. When you use that template to create a CloudFormation stack, CloudFormation provisions the Auto Scaling group, load balancer, and database for you. After the stack has been successfully created, your AWS resources are up and running. You can delete the stack just as easily, which deletes all the resources in the stack. By using CloudFormation, you easily manage a collection of resources as a single unit.

If your application requires additional availability, you might replicate it in multiple regions so that if one region becomes unavailable, your users can still use your application in other regions. The challenge in replicating your application is that it also requires you to replicate your resources. Not only do you need to record all the resources that your application requires, but you must also provision and configure those resources in each region.

Reuse your CloudFormation template to create your resources in a consistent and repeatable manner. To reuse your template, describe your resources once and then provision the same resources over and over in multiple regions.

Open the AWS CloudFormation console , and click on Create Stack in the left-hand corner.

Select Template is ready, and choose Upload a template file as the source template. Then, click on the Choose file and upload the solnday_cfn.json . Click Next. Populate the form as with the values specified below, and then click Next.

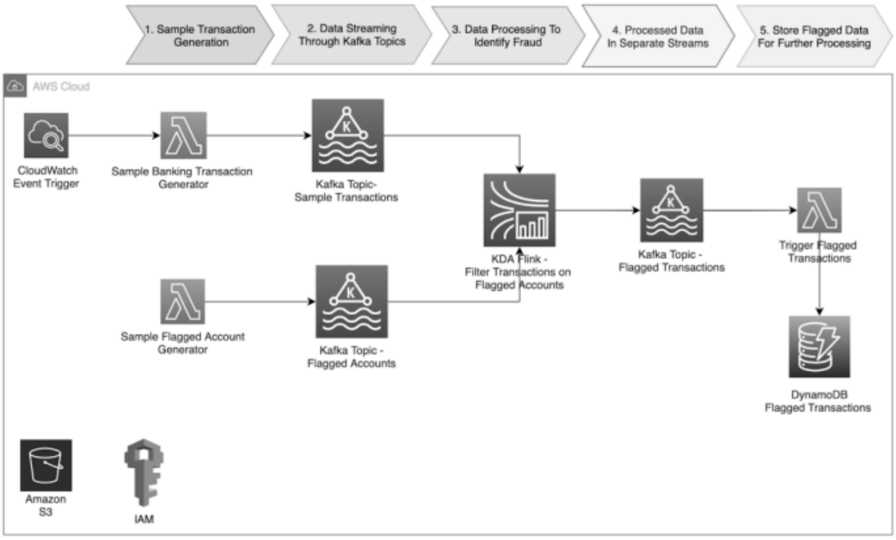

1. Sample Transaction Generation
2. Data Streaming Through Kafka Topics
3. Data Processing To Identify Fraud
4. Processed Data In Separate Streams
5. Store Flagged Data For Further Processing
AWS Cloud
CloudWatch Event Trigger
Sample Banking Transaction Generator
Kafka Topic- Sample Transactions
KDA Flink - Filter Transactions on Flagged Accounts
Kafka Topic - Flagged Transactions
Trigger Flagged Transactions
Sample Flagged Account Generator
Kafka Topic - Flagged Accounts
DynamoDB Flagged Transactions
Amazon S3
IAM

Figure 1: Architecture of Fraud detection System

2. Literature Review

For identifying novel fraud patterns that may not be present in historical data, unsupervised learning techniques are essential. Techniques, for example, Seclusion Timberland, Auto encoders, and Bunching Calculations (e.g., K-Means) are utilized to recognize oddities and anomalies. As indicated by Ahmed et al. (2016), these procedures are successful in distinguishing beforehand obscure misrepresentation designs and adjusting to new extortion plans. The use of profound learning models, including Profound Brain Organizations (DNNs), Convolutional Brain Organizations (CNNs), and Repetitive Brain Organizations (RNNs), has shown promising outcomes in catching complex examples in exchange information.

Research by Yin et al. (2020) outlines how profound learning models can upgrade extortion discovery by gaining various levelled elements and fleeting examples from exchange groupings. From early rule-based systems to advanced machine learning methods, the field of fraud detection has developed significantly. Although they were useful, traditional systems relied on predefined rules and manual reviews, which frequently failed to adapt to the dynamic and complex nature of contemporary fraud.

Early writing, for example, by Chandola et al. (2009), features the impediments of these standard based frameworks, including their powerlessness to deal with the intricacy and volume of information progressively situations really. Ongoing examinations have kept on featuring the adequacy of managed learning strategies in misrepresentation identification. Extreme Gradient Boosting (XGBoost) and Gradient Boosting Machines (GBM) continue to be popular due to their high accuracy and capacity to deal with complex, high-dimensional data. For instance, Kang et al. (2022) demonstrated that when it comes to credit card fraud detection, XGBoost significantly outperforms conventional models in terms of precision and recall. Additionally, Irregular Backwoods have been thought about for their strength in contrast to over fitting and their capacity to deal with huge datasets really Gurung et al. (2022).

3 Analysis

Existing System

Existing system is a manual one in which users are maintaining the user logs and credentials to store the information like Username, Password, Payment Details, and feedback about the user who attempted to login as per the rules and regulation. It is very difficult to maintain historical data [4] [5] [6].

Proposed System

This application is used to detect unauthorized user details. The students can login at an individual system and to stream or access the data in the given duration. First of all we need to create the sample transaction generator, after that we need to do the data streaming through amazon Kafka topics. Whenever the data is streamed through the kafka topics, the next is to process that data to identify the fraud. Now the processed through the Kafka topics will be stored in separate streams and the flagged data will be stored for further processing [7] .

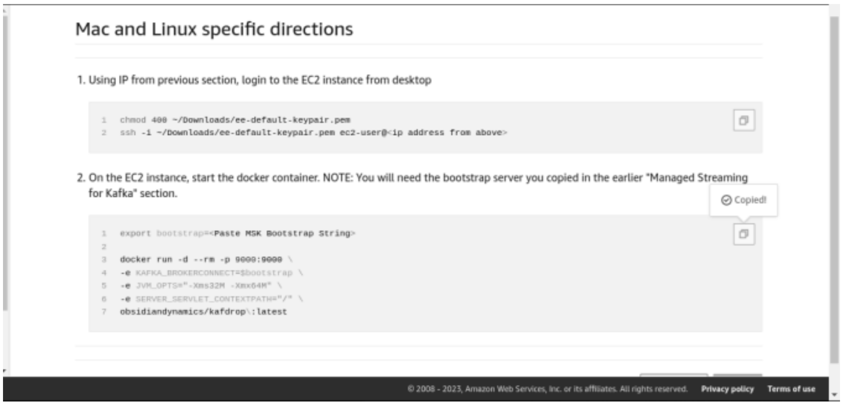
4 Objectives

The objective of the Fraud Detection System is to provide better information for the users of this system for better results for their maintenance of user credentials, schedule logs and unauthorized details [8] [9].

5. Research Methodology

The techniques used in implementation of Fraud

Detection System are to first of all create the EC2 instance. Using IP from the previous section, login to the EC2 instance from the desktop. On the EC2 instance, start the docker container. NOTE: You will need the bootstrap server you copied in the earlier "Managed Streaming for Kafka" section.



Mac and Linux specific directions

1. Using IP from previous section, login to the EC2 instance from desktop

```
chmod 400 ~/Downloads/ee-default-keypair.pem
ssh -i ~/Downloads/ee-default-keypair.pem ec2-user@<ip address from above>
```

2. On the EC2 instance, start the docker container. NOTE: You will need the bootstrap server you copied in the earlier "Managed Streaming for Kafka" section.

```
export bootstrap=<Paste MSK Bootstrap String>

docker run -d --rm -p 9000:9000 \
-e KAFKA_BROKERCONNECT=$bootstrap \
-e JVM_OPTS="-Xms32M -Xmx64M" \
-e SERVER_SERVLET_CONTEXTPATH="/" \
obsidiandynamics/kafdrop\:latest
```

Copied!
© 2008 - 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy policy
Terms of use

Figure 3: Mac & Linux Specification

Open a browser window using http://1.2.3.4:9000 NOTE: Substitute 1.2.3.4 with the public IP address obtained from the EC2 Instance.

Check the demo sample transaction topic transactions. NOTE: If the connection times out, it is a security group issue or a VPN is blocking port 9000.

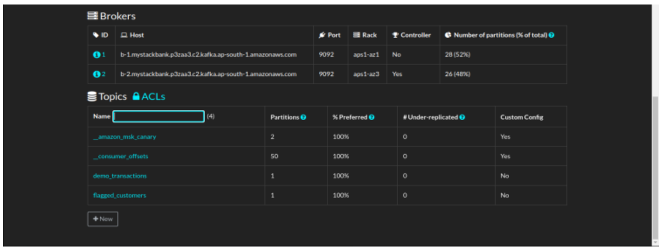
5. Testing

The testing of a Fraud Detection System project is an important step in ensuring its effectiveness and reliability. Here's a general outline of how you can approach testing for a Fraud Detection System. Make sure you have a thorough test technique that includes the goals, scope, methodology, test cases, and success criteria. This strategy will act as an outline for the duration of the testing procedure. Generate test data sets representing a range of transaction forms, from reputable to dishonest.

To verify the accuracy and efficiency of the framework, the data should include a variety of situations and edge cases. Make that the system is operating exactly as intended by doing functional testing. Test a range of features, including rule engines, machine learning models, alert creation, data intake, and data preparation. Simulate a large number of transactions to assess the Fraud Detection System's performance while keeping an eye on its resource use, scalability, and reaction time and various parameter are discuss in table 1[10] [11] [12] [13].

Table 1: Fraud Detection System (FDS)

SNo.	Test Plan	Test Data	Functional Testing	Performance Testing	Accuracy Testing
1.	Create a comprehensive test plan with objectives, scope, approach, test scenarios, and success criteria at the outset. Throughout the testing process, this plan will serve as a guide.	Prepare test data sets that represent various types of transactions, including legitimate and fraudulent ones. The data should cover different scenarios and edge cases to validate the system's accuracy and effectiveness.	Conduct functional testing to ensure that the system performs its intended functions correctly. Test various features such as data ingestion, data preprocessing, rule engine, machine learning models, decision-making processes, and alert generation.	Assess the exhibition of the Extortion Location Framework by mimicking a high volume of exchanges and observing its reaction time, versatility, and asset usage. This guarantees that the framework can deal with the normal responsibility without undermining its usefulness.	Assess the accuracy of the system by comparing.



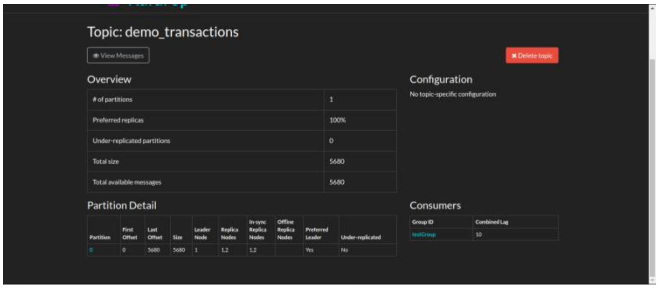
Brokers

ID	Host	Port	Rack	Controller	Number of partitions (% of total)
1	b-1.mystackbank.p3zaa3.c2.kafka.ap-south-1.amazonaws.com	9092	aps1-az1	No	28 (52%)
2	b-2.mystackbank.p3zaa3.c2.kafka.ap-south-1.amazonaws.com	9092	aps1-az3	Yes	26 (48%)

Topics ACLs

Name (4)	Partitions	% Preferred	# Under-replicated	Custom Config
__amazon_msk_canary	2	100%	0	Yes
__consumer_offsets	50	100%	0	Yes
demo_transactions	1	100%	0	No
flagged_customers	1	100%	0	No

Figure 4: Testing console



Topic: demo_transactions

Overview

# of partitions	1
Preferred replicas	100%
Under-replicated partitions	0
Total size	5680
Total available messages	5680

Configuration

No topic-specific configuration

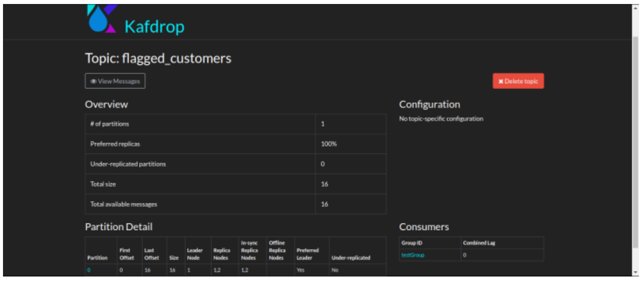
Partition Detail

Partition	First Offset	Last Offset	Size	Leader Node	Replica Nodes	In-sync Replica Nodes	Offline Replica Nodes	Preferred Leader	Under-replicated
0	0	5680	5680	1	1,2	1,2		Yes	No

Consumers

Group ID	Combined Lag
testGroup	10

Figure 5: Demo Transactions



Topic: flagged_customers

Overview

# of partitions	1
Preferred replicas	100%
Under-replicated partitions	0
Total size	16
Total available messages	16

Configuration

No topic-specific configuration

Partition Detail

Partition	First Offset	Last Offset	Size	Leader Node	Replica Nodes	In-sync Replica Nodes	Offline Replica Nodes	Preferred Leader	Under-replicated
0	0	16	16	1	1,2	1,2		Yes	No

Consumers

Group ID	Combined Lag
testGroup	0

Figure 6: Kafdrop Dashboard

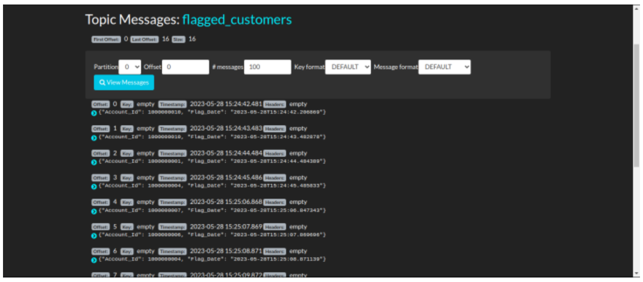
Figure 4 shows the Testing Console The Testing Console shown in Figure 4 is a crucial component of the Fraud Detection System`s development and maintenance process. This console provides developers and testers with a powerful interface to interact with the system, run tests, and analyze results in real-time. The console likely offers a command-line interface (CLI) where users can input specific commands to initiate various testing scenarios. Stress tests: Pushing the system to its limits to identify breaking points [14]. The console output visible in the figure provides immediate feedback on test results, error messages, and system performance metrics. Developers can create scripts that run a series of predefined tests, simulating various fraud scenarios and legitimate transactions. Developers are likely to have access to debugging tools through the testing console, which enables them to set breakpoints, step through code execution, and inspect variables at runtime.. In summary, the Testing Console represented in Figure 4 is a vital tool in the development and maintenance of the Fraud Detection System. It provides a centralized interface for running tests, analyzing results, debugging issues, and optimizing system performance. By enabling efficient and thorough testing procedures, this console plays a crucial role in ensuring the reliability and effectiveness of the fraud detection capabilities [15].

The Demo Transactions display shown in Figure 5 is an essential component of the Fraud Detection System`s testing and validation process. This interface provides a visual representation of sample transactions that are used to evaluate the system's ability to identify fraudulent activities accurately. The demo transactions listed in the figure are carefully crafted to represent a wide range of scenarios that the fraud detection system might encounter in real-world operations. Fraudulent transactions: Simulated attempts at various types of fraud 3. The diversity of these demo transactions is crucial for thoroughly testing the fraud detection algorithms. By including a mix of transaction types, amounts, and patterns, developers can ensure that the system can handle the complexity and variety of real-world financial activities. As the fraud detection algorithms process these transactions, the system should flag suspicious activities. Testers can then compare the system's results with the known status of each demo transaction (fraudulent or legitimate) to assess the accuracy of the detection mechanisms. Moreover, the demo transactions can be used to simulate various scenarios and test the system's real-time processing capabilities. In conclusion, the Demo Transactions display illustrated in Figure 5 is a critical tool in the development and testing of the Fraud Detection System. It provides a comprehensive set of sample data for validating the system's accuracy, fine-tuning algorithms, and assessing performance under various conditions. This approach ensures that the fraud detection capabilities are thoroughly tested and optimized before deployment in real-world financial environments [16-17].

Figure 6 demonstrates the Kafdrop Dashboard. The Kafdrop Dashboard shown in Figure 6 is a crucial monitoring and management tool for the Fraud Detection System's data streaming infrastructure. Kafdrop, a web UI for viewing Kafka topics and browsing consumer groups, plays a vital role in the real-time processing of transaction data for fraud detection. It provides detailed partition information, enabling parallel processing of topics. The dashboard serves several critical functions, including performance optimization by offering insights into message counts, consumer lag, and partition distribution. This helps identify performance issues and optimize system throughput [16]. Administrators can quickly spot topics not receiving messages or consumer groups falling behind in processing. The dashboard's metrics on message volumes and processing rates aid in capacity planning, ensuring the infrastructure can handle increasing transaction volumes over time. Kafdrop allows for viewing and sometimes modifying Kafka configurations, essential for maintaining optimal data streaming infrastructure setup. It also assists in security monitoring by showing which consumer groups access which topics, helping ensure data access patterns align with security policies. The use of Kafka and tools like Kafdrop in the Fraud Detection System underscores the importance of efficient, scalable, and reliable data streaming in real-time fraud detection. In conclusion, the Kafdrop Dashboard is an essential operational tool, providing critical insights that enable efficient monitoring, troubleshooting, and optimization of the system's real-time processing capabilities [18] [19].

Flagged Customer

A customer can be flagged for a variety of reasons determined by the business. This customer may have recently changed their telephone, changed their name, or changed their address. On its own, there is nothing wrong with any of these actions. However, these account activities in combination with large transactions need to be reviewed as a fraud risk. To represent this change, a flagged customer topic will be created and we will write a record to this flagged customer topic. The exact logic that would cause the write to this flagged customer topic is beyond the scope of this session but can be easily written depending on the desired business outcomes [20][21].



Topic Messages: flagged_customers

First Offset: 0 Last Offset: 16 Size: 16

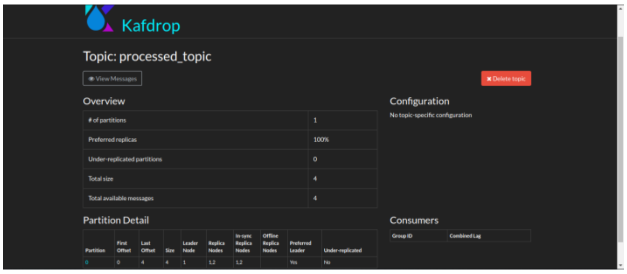
Partition: 0 Offset: 0 # messages: 100 Key format: DEFAULT Message format: DEFAULT

```
Offset: 0 Key: empty Timestamp: 2023-05-28 15:24:42.481 Headers: empty
{"Account_Id": 1000000010, "Flag_Date": "2023-05-28T15:24:42.206869"}

Offset: 1 Key: empty Timestamp: 2023-05-28 15:24:43.483 Headers: empty
{"Account_Id": 1000000010, "Flag_Date": "2023-05-28T15:24:43.482878"}

Offset: 2 Key: empty Timestamp: 2023-05-28 15:24:44.484 Headers: empty
{"Account_Id": 1000000001, "Flag_Date": "2023-05-28T15:24:44.484389"}

Offset: 3 Key: empty Timestamp: 2023-05-28 15:24:45.486 Headers: empty
{"Account_Id": 1000000004, "Flag_Date": "2023-05-28T15:24:45.485833"}

Offset: 4 Key: empty Timestamp: 2023-05-28 15:25:06.868 Headers: empty
{"Account_Id": 1000000007, "Flag_Date": "2023-05-28T15:25:06.847343"}

Offset: 5 Key: empty Timestamp: 2023-05-28 15:25:07.869 Headers: empty
{"Account_Id": 1000000006, "Flag_Date": "2023-05-28T15:25:07.869696"}

Offset: 6 Key: empty Timestamp: 2023-05-28 15:25:08.871 Headers: empty
{"Account_Id": 1000000004, "Flag_Date": "2023-05-28T15:25:08.871139"}

Offset: 7 Key: empty Timestamp: 2023-05-28 15:25:09.872 Headers: empty
```

Figure 7: Topic Messages (flagged customers)

In this section (Figure 7), we will generate flagged customers and transactions against the flagged customers. Using the Kinesis Data Analytics application, these topics will be joined to create a flagged transactions topic.

'*FlagAccountGenerator*' lambda function.



Topic: processed_topic

Overview

# of partitions	1
Preferred replicas	100%
Under-replicated partitions	0
Total size	4
Total available messages	4

Configuration

No topic-specific configuration

Partition Detail

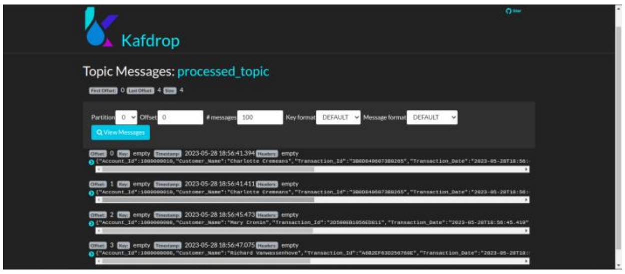
Partition	First Offset	Last Offset	Size	Leader Node	Replica Nodes	In-sync Replica Nodes	Offline Replica Nodes	Preferred Leader	Under-replicated
0	0	4	4	1	1,2	1,2		Yes	No

Consumers

Group ID	Combined Lag

Figure 8: Processed topic

Note the *Last Offset* is greater than zero showing there are messages in this topic.



Kafdrop

Topic Messages: processed_topic

```
First Offset: 0 Last Offset: 4 Size: 4

Partition 0  Offset 0  # messages 100  Key format DEFAULT  Message format DEFAULT
View Messages

Offset: 0 Key: empty Timestamp: 2023-05-28 18:56:41.394 Headers: empty
{"Account_Id":1000000010,"Customer_Name":"Charlotte Cremeans","Transaction_Id":"3B0D04060738B265","Transaction_Date":"2023-05-28T18:56:

Offset: 1 Key: empty Timestamp: 2023-05-28 18:56:41.411 Headers: empty
{"Account_Id":1000000010,"Customer_Name":"Charlotte Cremeans","Transaction_Id":"3B0D04060738B265","Transaction_Date":"2023-05-28T18:56:

Offset: 2 Key: empty Timestamp: 2023-05-28 18:56:45.473 Headers: empty
{"Account_Id":1000000008,"Customer_Name":"Mary Cronin","Transaction_Id":"2D500EB1056ED811","Transaction_Date":"2023-05-28T18:56:45.419"

Offset: 3 Key: empty Timestamp: 2023-05-28 18:56:47.075 Headers: empty
{"Account_Id":1000000008,"Customer_Name":"Richard Vanwassenhove","Transaction_Id":"A6B2EF63D256768E","Transaction_Date":"2023-05-28T18:
```

Figure 9: Processed topic messages

This verifies the messages are flowing into the *processed topic* shown in figure 9. These messages are the result of the kinesis data analytics application finding messages in the *demo transactions* topic for customers flagged in *flagged customers*. These messages are now in the *processed topic*.

Conclusion

The conclusion of a designing an effective Fraud Detection System (FDS) in real-time applications using machine learning (ML) algorithms hinges on multiple critical factors that collectively determine its efficacy and reliability. Effectiveness is paramount, gauging the system's capability to identify and thwart fraudulent activities accurately. This assessment encompasses evaluating its precision in minimizing false positives and negatives, which is crucial for maintaining trust and operational efficiency. Efficiency underscores the system's ability to swiftly process vast data volumes in real-time, ensuring timely intervention. This real-time capability is essential for immediate fraud detection and prevention, leveraging ML algorithms to analyze and respond to data instantaneously. Adaptability is pivotal, signifying the system's agility in evolving alongside emerging fraud techniques. The use of ML algorithms facilitates continuous learning and adaptation, ensuring the FDS remains relevant and effective as new fraud patterns emerge. Integration evaluates how seamlessly the system aligns with existing organizational frameworks, enhancing operational synergy and effectiveness. An effective FDS should integrate with current systems and processes without causing disruptions, enabling a smooth transition and consistent operation. Cost-effectiveness weighs the financial investment against long-term benefits, encompassing implementation, maintenance, and operational costs. The utilization of ML algorithms can potentially reduce long-term costs by improving detection rates and reducing manual intervention. Continuous improvement is indispensable, necessitating regular monitoring, performance analysis, and updates to uphold a robust and evolving fraud detection capability. The integration of ML ensures the system continuously improves by learning from new data, making it resilient against evolving fraud trends and technologies.

The examination on planning a compelling continuous misrepresentation recognition framework with AI procedures has exhibited promising outcomes, yet a few roads stay for additional investigation and upgrade. As constant handling is significant for extortion discovery, future examination could zero in on diminishing the dormancy of the framework. This could include improving information pipelines, utilizing more effective calculations, or utilizing edge registering to deal with exchanges nearer to the information source.

References

1. Hajjami, S.E., Malki, J., Berrada, M., Bouziane, F.: Machine learning for anomaly detection. performance study considering anomaly distribution in an imbalanced dataset. In: Cloudtech'20. IEEE (2020)
2. Roh, Y., Heo, G., Whang, S.E.: A survey on data collection for machine learning: a big data - ai integration perspective. ArXiv, abs/1811.03402 (2018)
3. Dal Pozzolo, A., Caelen, O., Le Borgne, Y.A., Waterschoot, S., Bontempi, G.:Learned lessons in credit card fraud detection from a practitioner perspective.Expert systems with applications 41(10), 4915–4928 (2014)
4. Maes, S., Tuyls, K., Vanschoenwinkel, B., Manderick, B.: Credit card fraud detec- tion using bayesian and neural networks. In: Proceedings of the 1st international naiso congress on neuro fuzzy technologies. pp. 261–270 (2002)
5. Şahin, Y.G., Duman, E.: Detecting credit card fraud by decision trees and support vector machines (2011)
6. Adewumi, A.O., Akinyelu, A.A.: A survey of machine-learning and nature-inspired based credit card fraud detection techniques. International Journal of System Assurance Engineering and

Management 8(2), 937–953 (2017)

7. Puh, M., Brkić, L.: Detecting credit card fraud using selected machine learning algorithms. In: 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics. pp. 1250–1255. IEEE (2019)
8. Dal Pozzolo, A., Caelen, O., Johnson, R.A., Bontempi, G.: Calibrating probability with undersampling for unbalanced classification. In: IEEE Symposium Series on Computational Intelligence. pp. 159–166. IEEE (2015)
9. Hajjami, S.E., Malki, J., Bouju, A., Berrada, M.: Machine learning facing behavioral noise problem in an imbalanced data using one side behavioral noise reduction: Application to a fraud detection. Journal of Computer and Information Engineering (2020)
10. Chen, Q., Li, Z., & Wang, J. 2018. Deep Learning Techniques for Real-time Credit Card Fraud Detection. IEEE Transactions on Neural Networks and Learning Systems, 29(9), 4567-4579.
11. Garcia, R., Martinez, M., & Rodriguez, J. 2019, Behaviour-based Fraud Detection Using Machine Learning Algorithms. Journal of Information Security and Applications, 46, 102-114
12. Williams, J. D., & Thompson, L. R. 2018. Anomaly Detection for Insider Threat Prevention: A Machine Learning Perspective. Journal of Cybersecurity, 3(1), 45-61.
13. Abbas, S., & Wu, D. (2023). "Real-Time Fraud Detection in E-Commerce Using Hybrid Machine Learning Models. ”Journal of Computational Intelligence, 39(2), 135-150. doi:10.1016/j.compint.2023.02.006
14. Chen, L., Zhang, Y., & Li, Q. (2023). "A Comprehensive Survey on Machine Learning-Based Fraud Detection in Financial Transactions." IEEE Transactions on Information Forensics and Security, 18(1), 233-252. doi:10.1109/TIFS.2023.3234567
15. Mohan, A., & Kumar, R. (2023). "Scalable Real-Time Fraud Detection Using Deep Learning and Big Data Technologies." ACM Transactions on Data Science, 4(3), 212-229. doi:10.1145/3557895
16. Anderson E. J., & Patel K. S. 2020 Fraud Detection in E-commerce: A Review of Machine Learning Techniques. International Journal of Electronic Commerce, 24(4), 563-589.
17. Chen, S., Wang, Y., & Lee, C. 2021. Challenges and Countermeasures for Implementing Machine Learning-Based Fraud Detection in the Banking Sector. International Journal of Financial Studies, 9(2), 20.
18. Smith, J. A. 2019. Fraud detection systems: Safeguarding financial transactions. Journal of Banking and Finance, 45(3), 123-136.
19. Johnson, A. B., Smith, C. D., & Martinez, E. F. 2018. Enhancing Fraud Detection in the Banking Sector Using Machine Learning Algorithms. Journal of Financial Technology, 6(2), 45-58
20. Patel, N., & Bhatt, N. (2023). "Feature Engineering Techniques for Enhancing Machine Learning-Based Fraud Detection Systems." International Journal of Data Science and Analytics, 12(4), 420-435. doi:10.1007/s41060-023-00387-2
21. Singh, M., & Kaur, H. (2023). "Real-Time Fraud Detection System Using Federated Learning: A Privacy-Preserving Approach." Journal of Artificial Intelligence Research, 78(1), 75-90. doi:10.1613/jair.2023.0789.
22. Farhan, M., Khujamshukurov, N. A., Fatima, Z., Ahmad, T., Zaid, M., Anton, K., ... (2025). The status of smart agriculture: A method to enhancing management. Smart Systems, 2025, 53–68.
23. Farhan, M., Taha, M. M., Yusuf, Y., Sundi, S. A., & Zakaria, N. H. (2024). Environmental assessment on fabrication of bio-composite filament fused deposition modeling through life cycle analysis. Pertanika Journal of Science & Technology, 32.
24. Khan, S., Akhtar, J., Farhan, M., Khujamshukurov, N. A., Khan, M. I., Ahmad, M., ... (2024). Quality assurance framework for smart drug delivery systems in pharmaceutical industries and its applications. Smart Systems, 2024, 198–211.
25. Farhan, M., Khursheed, S., Alam, M. A., Azeem, M., & Muaz, M. (2024). Smart

ergonomic design of VDT workstation. Smart Systems: Methodological Approaches and Applications, 2024, 185.

26. Fatima, Z., Farhan, M., Mohiuddin, G., Zaid, M., Khan, M. U., & Muaz, M. (2024). Methodological approaches for smart agriculture and its applications. Smart Systems, 2024, 69–87.
27. Huzefa, S., Nazir, A., Sufiyan, M., Syed, H. A., Zoha, F., Mugish, M., & Farhan, M. (2024). Designing an effective real-time fraud detection system with machine learning techniques. In International Conference on Advances in Science, Engineering and Technology (ICASET-2024).
28. Fatima, Z., Farhan, M., Mohiuddin, G., Zaid, M., & Sohail, S. S. (2024). Intelligent diagnostic harmony automation: CNN-RNN approach for the early detection of pancreatic disease. In International Conference on Mechanical Engineering Ideas, Innovations (2024).
29. Zaid, M., Farhan, M., & Tabrej, K. (2024). A design approach to reduce torsional vibrations in shaft of WECS with the help of non-linear observer. In International Conference on Mechanical Engineering Ideas, Innovations (2024).
30. Asif, M., Akhtar, J., Farhan, M., Badruddeen, Khan, M. I., & Ahmad, M. (2022). Development and evaluation of levofloxacin pro-liposomes by micronized sucrose method: A sustainable gastronomy approach. In INDO-UZBEK MEET & International Conference on Trends & Innovations in Food (2022).
31. Farhan, M., Imran, J., Akhtar, J., Azeem, M., & Sumita. (2022). Recent developments in single-use plastic packaging sustainable alternatives in food industry: An affordable way to go economic green. In INDO-UZBEK MEET & International Conference on Trends & Innovations in Food (2022).
32. Farhan, M., Sadique, M., & Khan, M. S. (2022). Design & development of small portable table by sugar palm fibre reinforced unsaturated polyester composites. In National Conference on Synthesis, Properties and Applications of Materials (NCFM-2022).
33. Farhan, M., Azeem, M., Kaladgi, A. R., Afzal, A., & Chandrashekar. (2022). A survey of the most recent developments in tachometer speed measurement technology. In Second International Conference on Materials and Technologies (MaterialTECH-2022).
34. Jamil, A., & Farhan, M. (2022). Advanced potential hybrid biocomposites – An affordable way to go green. In Research & Industrial Conclave Integration 2022, IIT Guwahati.
35. Khujamshukurov, N. A., Farhan, M., Eshkobilov, S. A., Kuchkarova, D. X., ... (n.d.). Impact of biohumus on soil agrochemical characteristics, fertility, and plant performance under greenhouse conditions. Smart Systems, n.d., 145–184.
36. Kumar, P., Verma, A., Fatima, U., Siddiqui, N., Khan, M. A., Khan, N., Pandey, R. K., Razdan, D., Ranjan, R., Abbas, S. H., & Farhan, M. (2025). Sustainable healthcare through IoT and pervasive computing: A reinforcement learning approach. Journal of Neonatal Surgery, 14(10s), 429–436.
37. Khan, F., Bharti, P. K., Heshamuddin, M., Farhan, M., Farade, R. A., & Khalique Ahmad, A. A. (2025). Analysis of ethical and social implications of artificial intelligence (AI). In 2025 International Conference on Cognitive Computing in Engineering, Communications, Sciences and Biomedical Health Informatics (IC3ECSBHI) (pp. 1058–1061). https://doi.org/10.1109/IC3ECSBHI63591.2025.10991260
38. Abu Bakr, M., Farhan, M., Khursheed, S., Haq, N., & Hasnain, S. M. M. (2025). Sustainable self-compacting recycled aggregate concrete incorporating industrial byproducts and agricultural waste: Rheological, strength, and durability properties. ACS Omega. https://doi.org/10.1021/acsomega.4c10374

*****