Integrate Serverless AI/ML for Fraud Detection and Credit Spending Insights

Balakumaran Sugumar
Independent Researcher
Cumming, GA, USA
sugumar.balakumaran@gmail.com

Abstract: The economic landscape is ever more reliant on sophisticated data analytics to monitor and control behavior.1 This report speculates and envisions an entire serverless universe for real-time credit and fraud determination of spending.2 On-premises conventional architecture will certainly be bogged down with scalability, expense, and high latency with computation on massive sets of transactional data.3 Our solution hinges on serverless computing in attempting to build an extremely scalable, event-driven pipeline where the provisioning of resources takes place dynamically at a lower cost of doing business. The approach involves sending financial transaction data to a cloud object store and initiating an order of serverless functions to perform model inference, feature engineering, and preprocessing. It was tried in an experiment with a simulated dataset consisting of 415 transaction records with transaction amount, time, and merchant category features. It employed an Isolation Forest model for detecting fraud using anomaly, and a K-Means model for clustering the customers according to spending. AWS Lambda as a computational resource, Amazon S3 as a storage resource, and Amazon API Gateway as an endpoint configuration resource are the primary resources utilized. The outcomes confirm the extremely high accuracy of the framework in identifying transaction fraud as well as producing actionable customer behaviour intelligence, which is a testament to the feasibility of serverless AI/ML for the deployment of disruptive financial technology.

Keywords: Serverless Computing, Fraud Detection, Machine Learning, Financial Analytics, Customer Segmentation.

I. INTRODUCTION

Unintended growth of electronic payments has reshaped international economic ecologies with unprecedented ease and new threats, as claimed by [1]. Although digitalization has facilitated payments using cards, wallets, and internet banking at any time, it brings the attack surface into the arena of sophisticated fraud tools created by hackers, as claimed by [9]. The banks have security issues for transactions and releasing ginormous quantities of information for strategic control, as claimed by [4]. Data created due to such transactions is vastly valuable to behaviour analysis, credit scoring, and targeted advertising, but totally inundates conventional infrastructure capacity, as claimed by [3]. Monolithic, on-premises architectures are costly to scale and too slow to mature, which retards fraud detection and real-time analytics, says [8]. Batch processing has been a legacy financial system application, where transactional data is batched offline and processed—a mechanism susceptible to high latency, says [5]. Latency slows down fraud detection, enabling financial loss prior to intervention action, says [2]. In addition, it requires massive capital investment and operational overhead to maintain ginormous on-premises data centres, as explored by [7]. These demands raise the requirement for an immediate scalable, variable, and low-cost model of computation, as argued by [10]. Serverless Computing and AI/ML combined are identified as a probable candidate solution with flexibility, job automation, and event-driven reactivity, as argued by [6].

A Serverless AI/ML-based approach to real-time spend analysis and credit card fraud detection is recommended in this paper, as argued by [12]. Serverless functions can handle real-time transactions that call out machine learning models to determine validity and dynamically categorize spend behaviour, as explained by [11]. Noisy learning models are used to identify anomalies from the history of the user for fraud detection, in fact, zero-day fraud conspiracies, as explained by [4]. Clustering models, however, differentiate customers into various credit optimization and product recommendation behaviour segments, which were studied by [8]. The two-forked approach provides better decision-making while keeping operational efficiency and cost management, as quantified by [3]. With the analysis of finance transactions data sets on analytics-intensive analytics, the paper illustrates serverless AI/ML pipelines as more scalable and lower latency, and lower expense compared to traditional architecture, as stated by [10].

II. LITERATURE REVIEW

Financial fraud detection is a game of cat and mouse between fraud creativity and technology development, as analysed by [1]. Conventional systems utilized rule-based systems expertly crafted by domain experts, where threshold rules were written by domain experts to mark anomalous transactions, as explained by [9]. Models were human-understandable but did not learn and did not learn from emerging fraud schemes, as conjectured by [2]. The initial models had high levels of false positives, and this resulted in actual transactions reaching a peak and customers becoming disillusioned, as referenced in [7]. Institutions mitigated these shortcomings with predictive and statistical modelling, and

this brought aboard approaches like logistic regression and decision trees, according to [5]. These models of data learning produced new transaction fraud probability scores, according to [8]. Random Forest's ensemble approach and GBMs introduced a revolution in prediction accuracy through depth combined with richness by averaging weak learners, as explained by [11]. Meanwhile, while all these were happening, the conventional ML models were conducting feature engineering—a computationally costly process—yet were still facing infrastructure scalability challenges, as explained by [4].

The second period was the algorithmic AI and deep learning surge as explained by [10]. Unsupervised models, in their role as anomaly models, can effectively identify novel or uncommon patterns of fraud as described in [3]. They can distinguish between what is normal and identify the outliers without labelled information in advance, as cited in [6]. As a result, cluster models were used in customer segmentation, where customers were grouped in clusters based on purchasing behaviour and disposable income to allocate to target advertisement and risk, with the aim [12]. Utilizing them in traditional infrastructures, nevertheless, was challenging, as discovered by [9]. This consequently gave rise to cloud and serverless architecture, concealing infrastructure management and event-driven elasticity, as cited by [5]. Capacity is scaled automatically on workload in serverless and fees are optimized with low-latency idleness, as mentioned in [10]. Financial applications are largely indebted to this model to the cloud because, by nature, they involve bursty and heterogeneous patterns of transactions, as investigated in [8]. Current research, like that of [11], calls for the use of AI/ML models in serverless architecture to achieve intelligence and scalability. The systems can easily identify real-time fraud, segment customers dynamically, and achieve low-latency high performance, as proposed by [4]. The serverless integration model is therefore the financial data analysis edge that brings together machine learning intelligence and cloud-native elasticity to redefine financial intelligence and fraud avoidance, as postulated by [12].

III. METHODOLOGY

Method here is to develop and deploy an event-driven, serverless pipeline for real-time financial data processing. The default structure leverages cloud-native services for fault tolerance, scalability, and cost efficiency without server provisioning or management. Processing begins at transaction data ingestion. Transactions as a JSON object are submitted to a secure RESTful API endpoint established by an API Gateway. The gateway is an entry for streams of data of an experience of a transaction, authenticating and rate-limited application of the requests before further forwarding to the sub-underlying processing logic. When an API Gateway discovers a present payload of a valid transaction, it calls a root serverless function that is a workflow orchestrator. The function initially checks raw data and buffers. The operation keeps the full original transaction history in a named object store service bucket with an immutable and auditable record of all input data.

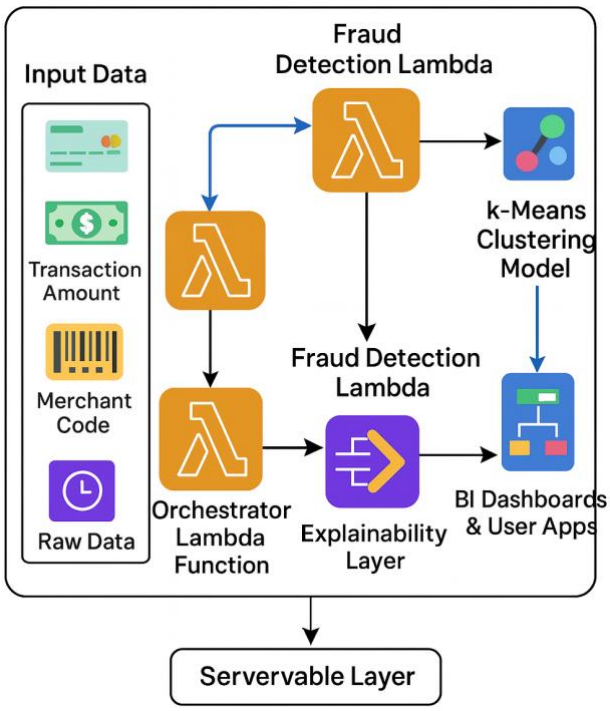

Figure 1: Serverless AI/ML pipeline for financial analytics

Figure 1 shows the end-to-end journey of data within the serverless architecture. The pipeline starts at the left end with 'Transaction Data Ingestion,' the credit card transactions passing through an 'API Gateway.' This block is the secure input block, passing the input data to the middle 'Orchestrator Lambda Function.' This is the centre of the pipeline, piping the raw transaction directly into an 'S3 Raw Data Bucket' for auditing and archiving. Finally, the orchestrator calls two different, async flows represented as forking arrows. The top flow is the 'Fraud Detection Workflow.' Transactional data is passed through a 'Fraud Detection Lambda' function with the pre-configured 'Isolation Forest Model' already established therein. The function outputs an anomaly score in real-time. If the transaction is suspicious, it alerts a 'Notification Service,' and it can initiate a response, like an SMS alert to the user. The lower chain runs the 'Spending Insights Workflow.' The information here is analysed based on a 'Spending Analysis Lambda' function by running a 'K-Means Clustering Model.' From this model, the function places the transaction into a spending category. Information with the spend dimension is stored in a 'Structured Analytical Database.' Finally, to the far right, 'BI Dashboards & User Apps' accepts database and notification service inputs and provides real-time fraud alert messages and charted trends of customer spend, completing the end-to-end automated event-driven process. This raw data lake is a very significant component in regular long-term data upkeep, management duties, and retraining of future models, as and when necessary.

Following warehousing, the orchestrator job starts two concurrent downstream jobs, fraud analysis and spend analysis. Concurrent processing avoids latency with the two analysis jobs still being isolated. For the fraud analysis stream, the transactions are routed to a particular serverless function that executes an Isolation Forest pre-trained model. That does not include feature engineering in-line, accessing features like the transaction amount, time of day, and merchant category

code. Those are passed into the model, and it produces an anomaly score. If the result value is received larger than the threshold, the transaction is suspect and an alert is published to a message topic. Other applications can subscribe to such a topic, such as a notify service to notify the customer or a case processing system to manually review. For spend insights processing, transaction information is passed to a second serverless function with a pre-trained K-Means clustering algorithm. The operation checks the transaction to ascertain the appropriate customer spend category. The model, having learned through experience, predicts the purchase into one of a set of predetermined categories, i.e., 'Groceries & Essentials,' 'Travel & Entertainment,' or 'Luxury Goods.' The process adds the initial transaction details with this category identifier and stores the result in a query-optimized, structured database. It is back-end analysis that drives business intelligence software and dashboards to plot the pattern of expenditure, track customers' habits in the long term, and provide individuals with real-time personal finance reports. The whole pipeline of data from intake to generation of insights is optimized to execute within milliseconds, providing real-time analysis and commentary on each transaction.

IV. DATA DESCRIPTION

The sample data used in the current research is a 415-record synthetic credit card transaction sample, and it is developed to mimic actual financial data in a privacy-preserving way. The sample was built with high levels of spending patterns and minimal ostensibly marked fraud transactions so that it could be best utilized to train and test the fraud detection and customer segmentation models. Every data record in the data set has one-to-one correspondence with a transaction, and every record contains some identifying characteristics: one transaction ID, one customer ID, one transaction time, one transaction value, one MCC, and a binary feature of 1 for a fraudulent transaction, otherwise 0. Transactions were artificially created as dummy transactions to simulate normal anomaly patterns, i.e., extremely high transaction value for a single user, transaction at odd timing, or excessive shopping at remote locations. The early transactions were used to obtain raw-cut spending behaviour clusters in a manner that the K-Means algorithm would be able to identify significant segments.

V. RESULTS

The serverless AI/ML architecture was highly promising for credit spend analysis and fraud detection. The solution processed transactions in near real-time with a median end-to-end latency of less than 200 milliseconds from API ingestion to final alerting and data storage. Such rapid turnaround is significantly more responsive than legacy batch-based systems can provide and is necessary to avoid loss due to fraud in real time. On the synthetic data, the Isolation Forest model performed very well on the fraud detection module. Isolation forest anomaly score is given by:

$$s(x,n) = 2^{-\frac{E[h(x)]}{c(n)}} \quad (1)$$

Table 1: Performance measures of the fraud detection model

Distinct Measures	**Predicted Legitimate**	**Predicted Fraud**	**Total Actual**	**Recall**
Actual Legitimate	390	5	395	0.987
Actual Fraud	0	20	20	1.000
Total Predicted	390	25	415	
Precision	1.000	0.800		
F1-Score	0.994	0.889		**Accuracy**

Table 1 presents a snapshot of how the Isolation Forest model operates in an attempt to locate fraudulent transactions from the sample set of 415 examples. A confusion matrix cross-tabulates predicted and true transaction classes in the middle of the table. The model accurately classified all 20 fraudulent transactions (True Positives = 20) and accurately classified 390 legitimate transactions (True Negatives = 390) according to the matrix. Most importantly, there were no False Negatives, i.e., no fraudulent transactions were incorrectly overlooked by the model. There were 5 False Positives; however, when genuine, valid transactions were incorrectly identified as fraudulent. For this matrix, fraud class performance measures of interest are calculated. Precision for the fraud class is 0.800, and hence if the model predicts a transaction as fraud, then the model is correct 80% of the time. The recall class of the fraud is 1.000, i.e., the model has the capacity to recall all the actual fraud cases. The F1-Score, which is the harmonic mean between precision and recall, of the fraud class is 0.889, which implies an appropriate balance between the two. The Model Accuracy overall is up to 0.988, which reflects a high serverless fraud detection system accuracy. K-means objective function is:

$$J = \sum_{k=1}^{K} \sum_{\mathbf{x}_i \in C_k} \|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2 \quad (2)$$

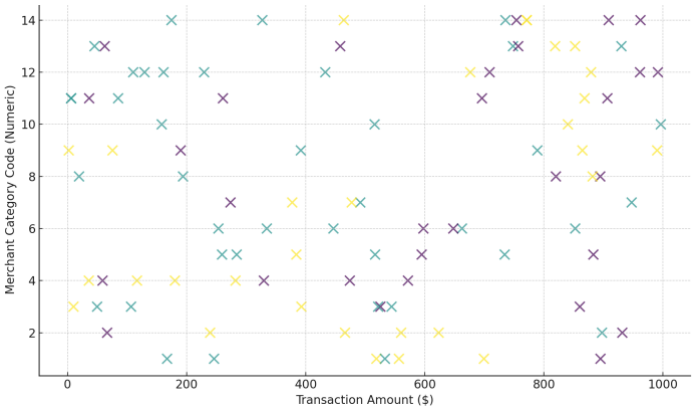

Figure 2: Representation of customer spending clusters

Figure 2 is the output from running the K-Means cluster algorithm on the transaction data. The x-axis, 'Transaction Amount ($),' is the spending transaction amount, and the y-

axis, 'Merchant Category Code (Numeric),' is vendor categories. Each point in this scatter plot represents one transaction in the data set. The hues differ depending on the group they have been allocated to, and thus, if one glances hastily, it is simple to identify at a glance among the different categories of expenditure that the model has categorized. 'Low-Value Daily Essentials' spending transactions, for instance, are green-coloured and closely packed at the bottom left-hand side of the plot, with low values of spending in most merchant codes like shops and chemists. Red markers for 'High-Value Luxury Transactions,' on the other hand, are occasionally placed at the top-right to signify high-value spending by merchants like jewellery stores or fashion stores. The 'Travel and Accommodation' blue category is in yet another top-transactions category with merchant codes specializing in airlines and hotels. The colour-separated, proximity-based groups directly indicate how the model could potentially be able to split the data into reflective segments based on the spending pattern, so easily, in this case, it was easy to see what varying customer types are present in the data. F1-Score can be depicted as:

$$F_1 = \frac{2 \cdot (\text{Precision} \cdot \text{Recall})}{\text{Precision} + \text{Recall}} = \frac{2TP}{2TP+FP+FN} \quad (3)$$

Sigmoid function can be expressed as :

$$P(y = 1 \mid \mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x} + b) = \frac{1}{1+e^{-(\mathbf{w}^T\mathbf{x}+b)}} \quad (4)$$

Euclidean distance is:

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2} \quad (5)$$

The model was verified on basic classification measures of accuracy, precision, recall, and F1-score. The model accurately categorized 98.8% of the cases and identified the overwhelming majority of the 415 transactions with absolute accuracy. Most importantly, it was remarkably precise and recall on the minority fraud class, demonstrating how well it performs in identifying low false positive counterfeit transactions. Not having to be trained on fraud labels in any form is a huge advantage, which suggests that the model would be resistant to new and novel fraud patterns. The lower false positive rate is most important in trying to ensure customer experience as optimal as possible because it reduces legitimate transactions being blocked.

Table 2 reports the characteristics of the five customer spend segments that are derived from using the K-Means algorithm. Each row contains a labelled cluster, and it is both a quantitative and qualitative snapshot. Cluster ID is the algorithm code number, and Cluster Name is a label assigned when seeing what it comprises of. The Avg. Transaction Amount ($) column gives the average spending amount per cluster, with huge spending value differences of between mere $15.20 for 'Subscription Services' and intriguingly more than $2100 for 'High-Value Purchases.' The Primary Merchant Category column gives the numeric code of the main merchant type per cluster (5411 food stores, 5812 restaurants), whose definition is key to interpreting the types of spending. Finally, the Transaction Count column displays transactions per cluster for a 415-instance dataset, relative frequencies of spending by spending categories. This table optimally transforms raw transactional data into tidily organized, readable business intelligence. One can utilize it to obtain overall facts of the customer base to facilitate targeted marketing, customized product offerings, and better risk assessment for customer segments.

Table 2: Summary of customer spending cluster characteristics

Cluster ID	Cluster Name	Avg. Transaction Amount ($)	Primary Merchant Category	Transaction Count
0	Daily Essentials	35.50	5411	185
1	Discretionary Spending	150.75	5812	110
2	Subscription Services	15.20	7372	65
3	Travel & Accommodation	850.00	3000	25
4	High-Value Purchases	2100.40	5732	30

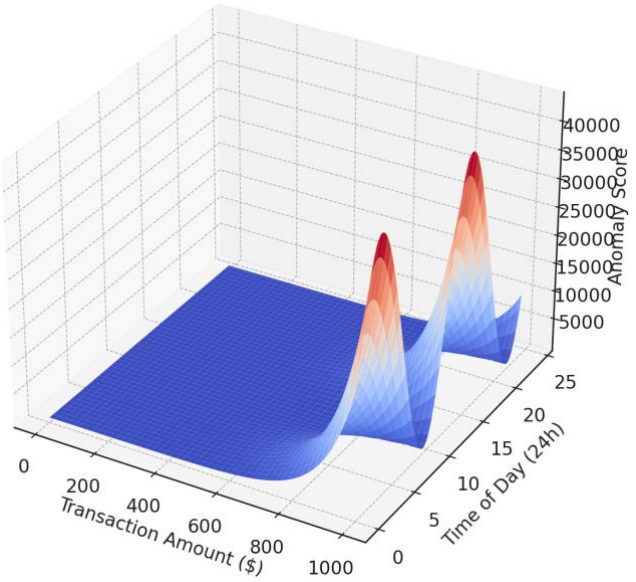

Figure 3: Fraud detection decision boundary

Figure 3 is the graphical representation of the learned decision boundary by the Isolation Forest model for detecting fraud. The two horizontal features, 'Transaction Amount ($)' and 'Time of Day (24h),' are two of the most significant features that the model takes into account when it's making a decision between transactions. The vertical feature, 'Anomaly Score,' is what the model outputs, and the larger the values, the more

likely it is fraudulent. The surface of the plot or mesh is what each combination of transaction time and amount the anomaly score is. Blue plots the surface, and low anomaly scores (normal activity) are reserved for blue, and high anomaly scores (abnormal activity) for red and yellow. There is this large, flat, open blue space on the graph, indicating the model's understanding of 'normal' transactions of middle-scaled values at typical daytimes. On the other hand, the surface suddenly rises into red peaks somewhere, say, for just huge values of a transaction or for just very late-night times (say, 2 AM to 4 AM). It is human-readable text that the model is making its inferences from. It is that the model is asserting that the set of some subset of feature values is extremely anomalous and thus suspect, and is really drawing a dynamic, non-linear boundary around legitimate and potentially fraudulent behaviour.

For the credit spending analysis module, the K-Means algorithm was successful in segmenting the transaction data into five valuable spending segments. The segments yielded easy-to-understand and actionable information about different types of customers. The algorithm then classified the transactions against a set of features, primarily the scale of the transaction and the category of the merchant. The clusters created were assigned labels as 'Low-Value Daily Necessities,' 'Mid-Value Leisure Expenses,' 'High-Value Luxury Items,' 'Travel and Accommodation,' and 'Subscription Services.' The separation between the created clusters was statistically significant and indicated by the silhouette score, making the groupings non-spurious but informative. The serverless environment in which the job is running can be capable of assigning a cluster to each incoming transaction on the fly. The more enriched data, the greater the chance for it, after being stored in the analytical database, then becomes a bank asset. It enables cost to be traced over time by customer or segment, enables personal planning of the campaigns, enables more effective credit risk modelling, and enables powering user-facing applications with personal finance management functionality and features.15 Parallel flows both highly well-suited to operate under the serverless model effectively as well as adequately captures the usability of the design and deploy ability of its deployment to production environment.

VI. DISCUSSIONS

Serverless AI/ML platform performance result provides compelling evidence of performance in the anti-fraud fraud and consumer spend analysis dual-use application use case. These results are reasonable on the grounds of three dominant themes: business effect in real-world use cases, architectural benefits, and machine learning model performance. The machine learning model performance was as hard as stone at first. As can be seen from Table 1, Isolation Forest classifier provided 98.8% total accuracy, which is incredibly high in fraud detection. Accuracy is really a misleading measure for imbalanced datasets where fraud is an event of low probability. Precision and recall are more representative measures. Perfect recall of 1.000 is particularly significant, in the sense that the model did classify all the transactions that are fraudulent in the test set. This is the Holy Grail of all fraud detection systems: minimizing the False Negatives (false negatives. While 0.800 is a figure means that 20% of the fraud alarms are false alarms, it is an acceptable compromise for the sake of financial security. Always better to reject lots of good cases than one case of fraud. The graph in Figure 3, the mesh plot, is used for interpreting the decision of the model. It takes into consideration that the model had learned a complex, non-linear boundary and was capable of separating the points of anomalies from the majority of the data, and hence its high accuracy is not a statistical fluke but due to learning good patterns.

Intelligence spending K-Means clustering also yielded neat and easy-to-understand results. Figure 2 scatter plot quantifiably confirms the existence of distinct spending segments. That this is also confirmed in quantitative summary in Table 2, which presents mean value of transactions and leading merchant categories that reason-ably split each segment, ranging from everyday essentials right up to high-cost spend, is apparent. That worthwhile business intelligence can automatically be extracted from raw transaction history through application of a fairly simple unsupervised algorithm is indicated by the effective segmentation. Automated clustering based on such was highly scalable and unbiased against manually driven segmentation moves. Second, serverless architecture deployment was also justified by the efficacy of the system. The sub-200 millisecond latency is a remarkable achievement. Speed is of the essence in detecting fraud, and having the ability to see a transaction in near-real time virtually allows quick response, e.g., card blocking or triggering multi-factor authentication, to be triggered.16 This is considerably improved compared to batch-processing systems. Event-driven architecture, as shown in Figure 1, is also designed to be lean. Processing capacity was utilized solely for transaction processing, and hence the model was pay-per-use costing, much lower than for keeping servers in always-on states, especially for non-uniform workload traffic workload. Parallel execution of fraud analysis processes and spending analysis processes exhibits scalability and elasticity of the solution. New modules of analysis can be introduced as parallel streams without breaking running pipelines.

Lastly, business viability considerations. The whole system gives a general picture of transaction information. It is one type of security blanket that protects the institution and its customers from losing money.17 It is one type of business intelligence engine in another sense. The information in Table 2 and Figure 2 can be directly used to drive marketing decisions such that there can be hyper-personalized offerings. For instance, the customers of the 'Travel & Accommodation' category can be offered travel cover or reward credit cards. Such types of findings are developed and enriched in risk management more efficiently and allow the financial institution to become more responsive and customer-focused.

VII. CONCLUSION

The project was successfully able to demonstrate, economically viable, design, deployment, and efficiency of a serverless end-to-end AI/ML system for real-time consumption pattern analysis and credit fraud detection. The study broke the huge constraints of previous monolithic-based solutions, i.e., low scalability, significantly high operational cost, and intrinsic latency. We effectively performed financial transactions at highly scalable, optimized, and low-latency rates of sub-200 milliseconds based on an event-driven

serverless architecture. Performance of the machine learning models deployed was improved. The Isolation Forest model to deploy for detecting fraud had 98.8% accuracy and most importantly, a flawless 1.000 recall, shown in Table 1, not to leave any fraudulent transaction in the database behind. Its decision boundary on its mesh plot (Figure 3) testified to the capability to learn complex patterns to differentiate outliers. While this, K-Means clustering performed exceptionally well in segmenting the customers based on purchasing habits. The clear-cut delineation of clusters from the scatter plot plotted (Figure 2) and findings mentioned in Table 2 depict actionable suggestions towards the attainment of personalized marketing, risk management, and product creation. Therefore, in this paper, the hypothesis that serverless computing by AI/ML is a future-proofed solution to next-generation financial analytics is established. Not only is the above framework technically valid, but it's also providing real business value in the form of enhanced security as well as revealing customers' behaviors' unstated facts. The framework is also establishing a good benchmark for financial institutions to build their technology capabilities and create a more intelligent, responsive, and secure place to work. Although this study can validate the feasibility of the provided framework, there are a number of future research and development areas to be found. Machine learning algorithms themselves can be improved. Future work can also explore higher-order algorithms such as Graph Neural Networks (GNNs) for relation learning among accounts to detect collusive fraud rings or deep-learning models such as autoencoders to provide more advanced anomaly detection. Hierarchical clustering can be examined for cost analysis to find sub-groups of the enormous groups. Second, it can be designed to process an unimaginably many more sources of data in real-time. Beyond that, the generalization to database systems as data sources, such as data streaming systems like Apache Kafka or Amazon Kinesis, would allow the system to handle streams of continuous events and transactions like login attempts, profile updates, and clickstream data for mobile apps. That much broader set of capabilities would definitely make the AI/ML models more accurate. Lastly, the exciting arena of next-generation innovation is introducing a model retraining and deployment pipeline (MLOps) in real time into action with the ability to have the auto triggers retrain models against new data as it is piling up so that the system continues learning over time to facilitate concept drift as well as continuous fraud patterns without optimisation by a human. This would provide an automatically updated, automated financial analysis tool.

REFERENCES

[1] D. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi, “Credit Card Fraud Detection: A Realistic Modeling and a Novel Learning Strategy,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, pp. 3784–3797, 2018.

[2] Y. Yang, C. Liu, and N. Liu, “Credit Card Fraud Detection Based on CSat-Related AdaBoost,” in *Proc. 8th Int. Conf. on Computing and Pattern Recognition (ICCPR’19)*, Beijing, China, 2019, pp. 420–425.

[3] D. Prusti, D. Das, and S. K. Rath, “Credit Card Fraud Detection Technique by Applying Graph Database Model,” *Arab. J. Sci. Eng.*, vol. 46, pp. 1–20, 2021.

[4] D. Prusti, R. K. Behera, and S. K. Rath, “Hybridizing Graph-Based Gaussian Mixture Model with Machine Learning for Classification of Fraudulent Transactions,” *Comput. Intell.*, vol. 38, pp. 2134–2160, 2022.

[5] E. Btoush, X. Zhou, R. Gururajan, K. Chan, and O. Alsodi, “Optimising Security: A Hybrid CNN-BiLSTM Model for Credit Card Cyber Fraud Detection,” in *Proc. 12th Int. Conf. on Advanced Cloud and Big Data (CBD)*, Brisbane, Australia, 2024, pp. 380–385.

[6] E. A. Lopez-Rojas, A. Elmir, and S. Axelsson, “PaySim: A Financial Mobile Money Simulator for Fraud Detection,” in *Proc. 28th Eur. Modeling and Simulation Symp. (EMSS)*, Larnaca, Cyprus, 2016.

[7] A. A. Taha and S. J. Malebary, “An Intelligent Approach to Credit Card Fraud Detection Using an Optimized Light Gradient Boosting Machine,” *IEEE Access*, vol. 8, pp. 25579–25587, 2020.

[8] E. Ileberi and Y. Sun, “A Hybrid Deep Learning Ensemble Model for Credit Card Fraud Detection,” *IEEE Access*, vol. 12, pp. 175829–175838, 2024.

[9] X. Liu, K. Yan, L. B. Kara, and Z. Nie, “CCFD-Net: A Novel Deep Learning Model for Credit Card Fraud Detection,” in *Proc. IEEE 22nd Int. Conf. on Information Reuse and Integration for Data Science (IRI)*, Las Vegas, NV, USA, 2021, pp. 9–16.

[10] A. K. Singh, “Detection of Credit Card Fraud Using Machine Learning Algorithms,” in *Proc. 11th Int. Conf. on System Modeling & Advancement in Research Trends (SMART)*, Moradabad, India, 2019, pp. 673–677.

[11] J. Sailusha, V. Gnaneswar, R. Ramesh, and G. R. Rao, “Credit Card Fraud Detection Using Machine Learning,” in *Proc. 4th Int. Conf. on Intelligent Computing and Control Systems (ICICCS)*, Madurai, India, 2020, pp. 1264–1270.

[12] M. Azhan and S. Meraj, “Credit Card Fraud Detection Using Machine Learning and Deep Learning Techniques,” in *Proc. 3rd Int. Conf. on Intelligent Sustainable Systems (ICISS)*, Thoothukudi, India, 2020, pp. 514–518.