National College of Ireland

Cloud-Optimized AI Framework for Real-Time E-Commerce Fraud Detection

MSCCLOUD Research Project
Master of Science in Cloud Computing

Asfand
Student ID: 23358530

School of Computing
National College of Ireland

Supervisor: Vikas Sahni

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Asfand
Student ID:	23358530
Programme:	Master of Science in Cloud Computing
Year:	2024-2025
Module:	MSCCLOUD Research Project
Supervisor:	Vikas Sahni
Submission Due Date:	11/08/2025
Project Title:	Cloud-Optimized AI Framework for Real-Time E-Commerce Fraud Detection
Word Count:	7326
Page Count:	22

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Asfand
Date:	10th August 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Cloud-Optimized AI Framework for Real-Time E-Commerce Fraud Detection

Asfand
23358530

Abstract

The growth in e-commerce transactions has witnessed an increase in fraudulent transactions; hence, efficient, accurate, and scalable fraud detection systems are needed. The proposed research is a fraud detector framework using supervised machine learning, developed using the popular credit card fraud data on Kaggle. Since the degree of class imbalance in the dataset is too high, an oversampling strategy, SMOTEENN, serves to blend oversampling with data cleaning. A classification model is trained, based on XGBoost and optimized for logloss, with a random state set to a fixed value to ensure reproducibility. Probabilistic predictions are produced and binarized to binary predictions at an optimized threshold, which is arbitrarily set at 0.4, to enhance sensitivity to minority class predictions. The model offers solid predictive strength as it acquired an accuracy of 99.97%, precision of 1.0 and 0.9993 of non-fraud and fraud transactions, respectively, and recall of 0.9993 and 1,0. The F1 scores are high in both classes and fail to show any unbalanced classification. The confusion matrix is reported as 38 false positives and 0 false negatives; thus, the performance of not misclassifying the cases, particularly the fraudulent ones, is excellent. The general architecture is designed to be deployed in various cloud services allowing the creation of a method of detection of fraud applicable to modern e-commerce systems.

Keywords: Fraud Detection, E-Commerce, XGBoost, SMOTEENN, Cloud Deployment, Imbalanced Data, Real-Time Classification

1 Introduction

The rapid advancement of e-commerce and digital financial systems has revolutionized the way transactions are conducted, offering consumers convenience, speed, and access to global marketplaces. Nevertheless, the same evolution has increased the exposure to fraudulent attacks. Financial ecosystems face threats and attacks of sophisticated fraud with identity theft, account takeovers, dummy merchants, and transaction laundering, among others, undermining the financial ecosystem's integrity in an unprecedented environment. Suited mechanisms of detecting fraud have been traditional mechanisms that are based on core fixed rules and specific thresholds that have failed to identify these advanced and sometimes subtle frauds. Historical fraud signatures that conventional systems rely on also make them prone to model degradation and concept drift under dynamic environments, as noted by Chouhan et al. (2024).

With a view to countering such challenges, a shift towards artificial intelligence (AI) and machine learning (ML)-based fraud detection systems has taken place in the field.

These smart tools are capable of detecting complicated patterns of very high-dimensional data regarding transactions and can be trained to learn the emerging fraud trends. Ensemble approaches, graph-based techniques, and deep learning models have been proven to outperform in multiple fraud situations, providing superior scalability and accuracy. As an example, transparency to these systems comes at the cost of robustness through the introduction of self-explainable fraud detection architectures, which is SEFraud (Li et al., 2024b). Moreover, hybrid systems that incorporate the benefits of both convolutional neural networks (CNNs), long short-term memory networks (LSTMs), and decision trees have recorded high levels of detection with fewer false positives (Musunuri, 2023; Jabeen et al., 2025).

Simultaneously, cloud-based deployment platforms such as Azure AI or AWS SageMaker have become a necessary facilitator in the execution of such AI models into real-time environments. SageMaker supports the entire machine learning lifecycle, including data preprocessing and model training, versioning, and inference to be scalable and highly available. According to assertions forwarded by Kumar et al. (2024) and Komati (2025), cloud-based infrastructures enable the deployment of latency-sensitive fraud detection pipelines that can grow elastically with the volume of transactions, which is imperative in the context of modern e-commerce platforms.

Ethical AI is another important issue in fraud detection. Automation of financial decision-making opens new risks connected with algorithm bias, fairness, and trust. Unbalanced datasets and black-box models can result in discrimination or service refusal without any reason. This has led to the adoption of explainable AI (XAI) models that provide a degree of interpretability into the model in decision-making, both globally and locally, known as SHAP and LIME frameworks, respectively (Zhou et al., 2023). A more recent study by Almalki and Masud (2025) focuses on how stacking ensemble models and XAI can improve both fraud detection systems and their credibility and accountability. In a similar capacity, Yaseen and Al-Amarneh (2025) contend that fairness and the appearance of transparency are the two most important factors that determine how users trust and use AI tools to strengthen the banking industry.

Implementation of fraud detection systems should also entail consideration of regulations regarding different jurisdictions. Explainability, non-discrimination, and auditability of automated systems will be required by legislation like the EU General Data Protection Regulation (GDPR), and by new AI governance policies. Fraud detection systems with Human in the Loop (HITL) interfaces and visual analytics dashboards in real-time may assist in fulfilling these regulatory requirements. Heatmaps, saliency maps, and visualization of confidence scores can also help the compliance officers track the alerts and explain the actions undertaken by them (Patchipala, 2024; Li et al., 2024a).

The primary research questions addressed in this study include:

- **How can AI-driven fraud detection systems be optimized for real-time performance in cloud environments?**
- **What architectural strategies enhance explainability without compromising accuracy?**

To address these questions, this research proposes a comprehensive, cloud-optimized fraud detection framework combining AI, explainable models, and scalable cloud architecture. The contributions of this work are as follows:

1. A novel architecture that combines cloud computing and deep hybrid models for fraud detection.

2. Integration of explainable AI methods (SHAP, Grad-CAM) to support interpretability.

This paper is organized as follows: The section of Literature Review analyses recent literature on fraud detection models, cloud-based systems, and explainability techniques. The proposed system architecture and machine learning methodology are described in the methodology section. The design section makes a written note of the technical design decisions and deployment architecture. The implementation section talks about implementation steps on AWS SageMaker and different platforms that were used to perform the code. Results and performance analysis are discussed in the evaluation section. In the end, there is a conclusion to provide a summary of findings along with recommendations on future work.

2 Related Work

2.1 Rise of E-Commerce and the Complexity of Fraud Detection

The surging growth in e-commerce has made financial transactions more complex, and the traditional rule-based systems of identifying fraud are inappropriate. Due to the rise of transaction volumes and variety, fraud techniques have become much smarter, as fixed detection methodology proves to be unreliable. These types of systems that are frequently dependent upon pre-determined thresholds can poorly identify the emerging patterns of fraud, and thus, the high number of false positives and user dissatisfaction is the consequence.

Researchers have moved to AI models that are more flexible to utilize to deal with this. Transformer-based models (such as those suggested by Karthikeyan et al. (2025)) would do a better job than a recurrent model to predict complex sequential data, and provide low latency near-real-time fraud detection in cloud-native settings. These models apply to distributed systems, which are critical in high-speed financial services.

Instant streaming analytics also provides increased powers of detection. Patchipala (2024) presented a low-latency fraud alert system that provides poignancy to the idea that continuous analytics are necessary, rather than batch-driven. A related paper by Adesola et al. (2024) introduced a web-based application designed to work at a high-velocity digital market, and Bibire Seyi-Lande et al. (2024) focused on single data flows to help intervene on fraud in a timely manner.

New solutions also take advantage of generative AI. Tyagi et al. (2025) have shown that these models can identify dynamic patterns of fraud with less training and allow the creation of greatly responsive and adaptive fraud detection mechanisms on contemporary e-commerce websites and applications.

2.2 Scalable and Cloud-Native Fraud Detection Systems

Scalability and high availability have become the main concerns of e-commerce fraud detection. The models should be able to fully integrate with cloud-native tools and

workflows, as more and more organizations move to cloud infrastructure. Kumar et al. (2024) also points to good MLOps practices (namely, automated pipelines, versioning, monitoring, and retraining) as essential to a long-term, persistent performance of fraud detection.

Containerized microservices have become a popular architecture choice, which can be modularly deployed in different services such as ingestion, feature engineering, and alerting. Komati (2025) noted that they are flexible, and Boyapati et al. (2025) introduced a scalable microservices-based engine, BankNet, to secure and perform fraud analytics in banks. These systems provide low-latency insights and are compliant with the jurisdictional regulations by setting up the system regionally.

Another emerging strategy is edge computing, which has been mentioned by Sarma and Dey (2021) as the one that can mitigate latency and the lack of data sovereignty, particularly in regulated markets. Edge and cloud together create a middle ground between excellence, conformity, and privacy.

Small firms have also enjoyed the ease of using cloud computing. According to Aderinto (2025), due to the low prices, speed of deployment, and modular APIs, SMEs have embraced advanced fraud tools. Such change is in line with the general trend in SaaS, which makes the AI available to more people.

The use of multi-cloud and hybrid-cloud is on the increase. Kamau et al. (2024) presented its benefits, which are redundancy, cost efficiency, and fault tolerance. Kubernetes aids in controlling all those disparate settings by facilitating homogeneous, scalable roll-outs.

Lastly, scalable AI pipelines make them more responsive. Agboola et al. (2024) and Ojika et al. (2022) have shown that cloud-optimized engineering, automation, and integration of TensorFlow improve the latency of decisions and flexibility. The combined products of these innovations constitute what is to come in the forms of resilient, regulation-ready, and smart fraud detection systems.

2.3 Hybrid and Deep Learning Architectures

Deep learning methods have been demonstrated to be highly flexible to complicated patterns in the transaction data, but stand-alone deep networks tend to overfit, are computationally expensive, and possess poor interpretability. To curb these problems, hybrid models have been introduced where more than one method is used to provide better fraud detection. Take an example, Musunuri (2023) created a CNN-LSTM model, which contains CNN layers followed by LSTM ones, to efficiently feature spatial and temporal characteristics to prevent false positives.

It is also possible to combine rule-based logic and deep learning on hybrid systems to make domain-specific predictions. Shaik et al. (2025) proposed a blockchain-integrated block-based ML framework that augmented transparency, traceability, and security in fraud detection. The audibility of blockchain is enhanced due to the immutable format of its structured form, which is crucial in a regulated environment.

The understanding of interpretability is enhanced further by explainable AI (XAI). Li et al. (2024a) came forward with a GNN-based model that utilized transaction graphs to identify fraud through semi-supervised learning, achieving high generalization on a few labels. These systems are becoming vital since the awareness of model decisions is becoming highly valuable to finance.

Standing behind these hybrid models is a flexible data infrastructure. Ogunwole

et al. (2022) concentrated on pipelines that efficiently run in real time, and Ojika et al. (2022) constructed extensible systems based on TensorFlow that can respond to changes in hybrid deployment. Lakkaraju (2025) introduced pipelines that dynamically adapt to transaction complexity and user profiles via dynamic risk scoring.

On balance, all of these methods do point to the utility of hybrid structures in a practical sense; that is, they are accurate, fast, interpretable, scalable, and therefore extremely appropriate in a contemporary context of fraud detection with real dynamism.

2.4 Ethical AI, Fairness, and Trust

With the advent of AI-based fraud detection as a part and parcel of any financial system, fairness and trust issues, as well as the effects of bias, have increased remarkably. Putting machine learning models to production based on unbalanced data sets is dangerous and certainly creates the risk of discrimination towards the minority group, either by marking genuine users as false positives or missing new fraud patterns. Khan (2025) emphasized that it is critical to ensure that users trust them, particularly in financial businesses where fairness is one of the ingredients to achieving customer loyalty.

The method to counteract bias includes reweighting, sampling corrections, and fairness-aware learning, where the error rates on different groups of people are trying to be equalized. li2024 discussed an increasing interest in regulatory norms and ethically responsible practices in the research on fraud detection and industry practice.

The other pillar of trust is interpretability. The feature attribution and visual explanations are AI methods (explainable AI, XAI) assisting the user and regulators to interpret models by the model output. The situation is also changing with the use of human-in-the-loop (HITL) systems that allows monitoring and contestability of automatized choices.

Also, the legal regulations of the EU, such as the GDPR, push toward the transparency and non-discrimination of algorithms. Sarma and Dey (2021) also referred that fairness-aware systems should be adopted into the real-time processing pipelines and cloud computing platforms that are used by many diverse, global audiences. Summing up, ethical design in detecting fraud is no luxury addition; instead, it is a foundation upon which users will trust AI, and regulated and ethical AI will be deployed sustainably in high-stakes areas.

2.5 Visual Analytics and Explainability

Since the model of fraud detection is increasingly sophisticated, in particular, with deep learning and ensemble, the issue of interpretability is of paramount concern. Decision opacity impairs trust, model debugging, and regulatory compliance. According to Li et al. (2024a), clear visual interfaces were used to improve fraud detection, as the produced outputs of the AI are subject to comparison with human patterns of thinking, thus allowing its users to notice and respond to suspicious behavior.

Heat maps, saliency maps, and feature attribution graphs are other tools that make high-level model decisions easy to understand. They can be used to clarify what transaction characteristics might have contributed to the occurrence of certain fraud forecasts, which can be used by compliance officers and stakeholders in fraud alert interpretation.

Pre-eminent methods, including SHAP, LIME, and attention mechanisms, can provide local, explainable justification of particular decisions. Patchipala (2024) revealed that the

embedding of these XAI tools into cloud-based fraud systems resulted in an improvement to the investigation process and also error reduction on a large scale.

Also, dynamic dashboards give live information about patterns of fraud, confidence, and reliability index. These interfaces enable human-in-the-loop decision making and ensure financial ecosystem transparency. As stated by Li et al. (2024a), besides fostering accountability, visualizations also make users trustful, which is an important precondition for AI acceptance.

Visual analytics and XAI contribute to the usability and reliable use of fraud detection systems by narrowing the gap between the issues of AI reasoning and human comprehension.

2.6 Dataset Challenges and Real-World Constraints

To identify fraud effectively, a good model architecture is not enough; it is also important to rely on the quality of data and its representativeness. Class imbalance is one of the most perennial problems in this field, such that fraudulent transactions represent a tiny percentage of the overall activity. This skew can hugely affect the capacity of the classifiers to generalize, where models may converge on a trend of being biased through majority-class and neglecting patterns of minority fraud. Sarma and Dey (2021) acknowledged that there is a necessity for realistic model strategies, and the naive sampling, oversampling, or undersampling processes have been reported to distort the distribution of natural transactions.

Throwing more fuel on the flames of the imbalance problem is the question of data labeling. Labeling fraud is mitigated days and even months after the fact in many cases. This becomes a source of confusion when it comes to training and appraisal of models, particularly in fraud cases that are time-sensitive. Next, the nature of fraud constantly changes, i.e., this is the problem of concept drift, and stagnant data becomes out-of-date with time. Numerous open-source data (e.g., the popular Kaggle credit card fraud dataset) do not reflect this drift, making the findings of the research less applicable to real systems.

Evaluation pitfalls criticized typical evaluation methods in fraud detection research and identified both data leakage and unsuitable random train-test divisions, and the inappropriate use of accuracy as an evaluation measure. They promote a combination of legitimate testing, stratified cross-validation, and using such metrics as precision, recall, F1-score, and AUC-PR in skewed settings. Such procedures are more practical in terms of the mode of operations for fraud detection in real-time financial systems.

The other problem is the artificiality of the available datasets of public fraud. Although synthetic data can solve issues of privacy concern, they are not likely to recreate the subtle nuances of human behaviors or defense strategies observed in real-life fraud. Other methods like privacy-safe synthetic generation or federated training have been floated to deal with this issue without affecting data security.

In brief, quality, temporarily important, and ethically collected data is the core of reliable fraud detection. Until such fundamental problems are resolved, even the most complex of such models may perform poorly when applied to dynamic financial scenarios.

2.7 Real-Time Processing and Threshold Optimization

Current fraud detection systems require the capacity to handle transactions and give identification results instantaneously. Latency influences the perception of users and approvals of transactions and affects loss control directly. The models of detecting fraud will thus need to reach a compromise between simplicity and speed of calculation. Lightweight architectures or hardware-tailored deployment procedures (e.g., the simplification of the model or the use of GPUs) are regularly utilized to address sub-second inferences necessary in the production phase.

Due to real-time constraints, there is also the necessity of dynamic threshold tuning. The output of most classification models is probabilistic, and these are in turn converted into binary labels, and a threshold is used. Although a default value of 0.5 is popular, it can perform poorly in highly unbalanced assignments like fraud detection, the aim of which is to set the threshold. Shaik et al. (2025) demonstrated that an alternative value of the threshold, analogous to 0.4, would significantly enhance recall rates on smaller fraud cases, even though it would result in a slight increase in false positive rates. Such a trade-off can be tolerated in high-risk financial areas where missing a fraud is more expensive than a false warning.

Threshold optimization does not need to be fixed; it may be adaptive in real-time systems. As an example, thresholds can be time-of-day-based, transaction volume-based, user risk profiles-based, or geolocation-based. Other systems apply a multi-threshold approach: there are various thresholds applied to different types of transactions or groups of users. These exploits reduce the sensitivity of detection to contextual dangers, which in turn increases the chance of detection and user confidence.

A cloud-native architecture of fraud detection that proposed real-time tuning modules of thresholds coupled with stream processing engines was designed by Patchipala (2024). Their system assessed rates of key performance indicators (e.g., true positive rate, false positive rate) and went around automatically set thresholds in real time. The model can update itself based on this feedback loop so that it can deal with changing patterns of fraud, user behavior, and business rules without retraining the entire model.

The real-time processing and refined thresholding designs the fraud detection systems to be responsive, precise, and flexible fundamentals to execute them in high-frequency financial applications and global e-commerce networks.

2.8 Research Objectives

Based on the results obtained as a result of the analysis of previous works, the following research is intended to come up with a complex framework of AI-based fraud detection in a manner peculiar to modern e-commerce platforms. The paper makes use of the publicly accessible datasets, most notably the Kaggle credit card fraud dataset, and applies the concepts of deep learning and an ensemble of such models to solve issues related to the detection of rare cases of fraud in highly unbalanced transaction data. The most significant techniques are Synthetic Minority Over-sampling Technique (SMOTE) to balance the classes, Robust classification by Random Forest, and precision tuning, such as a 0.4 cutoff probability.

Table 1: Summary of Key Research Papers in Fraud Detection

Paper Title & Citation	**Authors**	**Techniques Used**	**Dataset**	**Key Findings**	**Accuracy**
Transformer-Based Financial Fraud Detection with Cloud-Optimized Real-Time Streaming Deng et al. (2025)	Deng, Bi, Xiao	Transformer, Real-Time Streaming, Cloud Integration	Custom Streaming Dataset	High-speed fraud detection with improved precision in cloud-native environments	92.6%
Enhanced Credit Card Fraud Detection Using Deep Hybrid CLST Model Jabeen et al. (2025)	Jabeen, Ramzan, Raza et al.	Hybrid CNN-LSTM, Deep Learning	Benchmark Credit Card Dataset	Deep hybrid models improve detection accuracy and reduce false positives	96.4%
A Machine Learning and Blockchain Based Efficient Fraud Detection Mechanism Ashfaq et al. (2022)	Ashfaq, Khalid, Yahaya et al.	Blockchain + ML, Transparent Pipeline	Synthetic + Real Bank Data	Blockchain ensures traceability, decentralization, and model security	95.0%
Visual Analytics for Event Detection: Focusing on Fraud Leite et al. (2019)	Leite, Gschwandtner, Miksch et al.	Visual Analytics, Heatmaps, Dashboards	Financial Event Logs	Visual tools support fraud pattern validation and auditor comprehension	N/A
Adoption of AI-Driven Fraud Detection in Banking: Role of Trust, Transparency, and Fairness Yaseen and Al-Amarneh (2025)	Yaseen, Amarneh	Fairness-Aware AI, Survey, Ethical ML	Survey + Real Bank Use Cases	Trust and fairness drive end-user acceptance in financial AI	Not Reported
Cloud-Optimized AI Framework for Real-Time E-Commerce Fraud Detection (Proposed Research)	Asfand	SMOTEENN, XGBoost, Threshold Tuning, Cloud Deployment	Kaggle Credit Card Fraud Dataset	Optimized fraud prediction using probabilistic thresholding and ensemble learning	99.97%

3 Methodology

The research methodology of the presented work thus consists of the design and performance evaluation of a real-time, explainable AI-driven framework of fraud detection that is implemented using Amazon SageMaker. The method is based on clear machine learning lifespan, which includes data preparation, class weighting, model selection, model training, threshold adjustment, and performance assessment.

3.1 Dataset and Preprocessing

The data provided is taken from a free archive of (Kaggle credit card fraud detection). It includes 284,807 anonymised credit card transactions with 30 features, of which 28 are principal components (‘V1‘ to ‘V28‘), and ‘Amount‘, ‘Time‘, and a binary class ranging against fraud (‘1‘) or non-fraud (‘0‘). When it comes to checking for the absence of missing values, it was done at the very beginning.

To normalize a scale of the monetary transactions and make them standard, the column ‘Amount‘ was normalized with the use of the `StandardScaler`. The time feature was removed because the temporal order was not used to predict. An interaction heatmap was also created to visually check the interaction of the features and class histograms to observe how heavily the classes are distributed, showing a highly imbalanced dataset with only 0.172 percent of transactions being fraudulent.

3.2 Handling Class Imbalance

The extreme unbalance led to re-sampling the dataset by applying the SMOTEENN algorithm Muntasir and Faisal (2022). This hybrid approach has the effect of both oversampling the minority class and eliminating ambiguous samples, which has the benefit of enhancing the generality of the model and minimizing false positives. It divided the resampled dataset with the help of 80-20 stratified `train_test_split`.

3.3 Model Selection and Training

The XGBoost classifier (`XGBClassifier`) was chosen because it is more accurate, capable of handling imbalanced data, and can be deployed in the cloud using SageMaker, among other platforms. I set the configuration of the model to use `use_label_encoder=False` and ‘logloss‘ to optimize the model. It was trained with the balanced training data and was tested on a held-out set.

3.4 Prediction and Threshold Optimization

In order to improve the performance of detecting fraud, particularly for the minority class, the predictions were developed on a probability scale. To ensure a high recall rather than precision rate, it used a classification threshold of 0.4 as opposed to the more commonly used 0.5 threshold since the omission of fraud would be more costly. Binarization of predictions was done.

3.5 Evaluation Metrics

The performance of the model was measured with several metrics: accuracy, precision, recall, F1-score, and confusion matrix by class. These values were calculated through the utilities of `sklearn.metrics`. To perform a visual analysis of the data, a chart plotting a heatmap of the confusion matrix was created. The findings proved powerful detecting powers with a great lift in the fraud recall and model confidence.

3.6 Visualization and Analysis

Several visualizations were produced to interpret the model and data:

- A bar chart illustrating class imbalance.
- Correlation heatmaps for all features.
- Boxplots showing distribution of transaction amount across classes.
- Distribution plots for PCA components (‘V1‘, ‘V2‘, ‘V3‘, ‘V4‘) across fraud and non-fraud classes.
- A final confusion matrix for evaluation summary.

3.7 Cloud Deployment Readiness

The model was made ready to be deployed in Amazon SageMaker. This involves preprocessing programs, training programs, and integration with the REST APIs. Auto-scaling and endpoint monitoring features of SageMaker enable the model to apply to traffic variations, resulting in low-latency fraud prediction and real-time explainability.

4 Design Specification

The section specifies the architecture design and technical specifications of developing a cloud-optimized, explainable artificial intelligence-powered credit card fraud detection system. They include the ability to assist in real-time detection, scalability, and transparency, and are engineered to be deployed on Amazon SageMaker. The architecture incorporates the main elements of the data preprocessing, an ensemble model, an explainable AI, and cloud deployment pipelines.

4.1 System Architecture

The given architecture is based on a modular pipeline approach, and the principal stages of the architecture include the following:

1. **Data Preprocessing:** The raw transactional data feed is ingested and processed to eliminate missing values and normalize the numerical values. During this phase, feature scaling with StandardScaler is done, and also irrelevant columns like `Time` are omitted to remove noise.

2. **Class Imbalance Handling:** The extreme class imbalance problem of the financial fraud dataset is solved by using a type of hybrid resampling scheme, SMOTEENN (Synthetic Minority Over-sampling Technique with Edited Nearest Neighbors), to oversample the minority (fraud) samples and clean the noisy samples.

3. **Modeling Phase:** The fundamental prediction model is founded on an XGBoost classifier, which was chosen due to its reliability, interpretability, and abilities in imbalanced binary classification problems. This model is learned on the balanced data set, and the threshold set on probabilities is then set to 0.4 to enhance the rates of recall on fraud.

4. **Evaluation Module:** The trained model is tested by numerous parameters such as Accuracy, Precision, Recall, F1-Score, and Confusion Matrix. An analytical validation can be carried out with visual inspection by means of heat maps and PCA components histograms.

5. **Explainability and Interpretability:** To make the model's action and decision-making transparent, the employment of post-hoc explanation techniques, like SHAP scores and saliency maps, will be considered. These assist human analysts to interpret and validate model decisions as regards to compliance and auditability.

6. **Cloud Deployment:** The design includes a deployment strategy of cloud deployment, which is based on AWS SageMaker, where the models are hosted, and inferences will be made. The architecture of the system also has an API-based interaction layer that uses communication with the deployed model with the external applications. Though not truly serverless, such a structure provides modularity, scaling, and integration capability in production.

4.2 Functional Design and Algorithm Description

XGBoost algorithm is the basis of the detection engine. It has a boosting type of learning that enhances the performance of poor learners in a stepwise manner, and to reduce overfitting, it regulates the learning. The model is outlined using the design considerations as given below:

- Objective function: `binary:logistic`
- Evaluation metric: `logloss`
- Class weights: Managed through SMOTEENN preprocessing
- Threshold adjustment: Tuned to 0.4 for balanced precision-recall tradeoff

The output of the classification is the comparison of the probability given in the prediction with the threshold. This enhances fraud sensitivity at a tradeoff of a minor certainty of false positives, which is acceptable in a financial setup.

4.3 Model Design Workflow

The end-to-end pipeline for the cloud-based credit card fraud detection system is visualized in Figure 1.

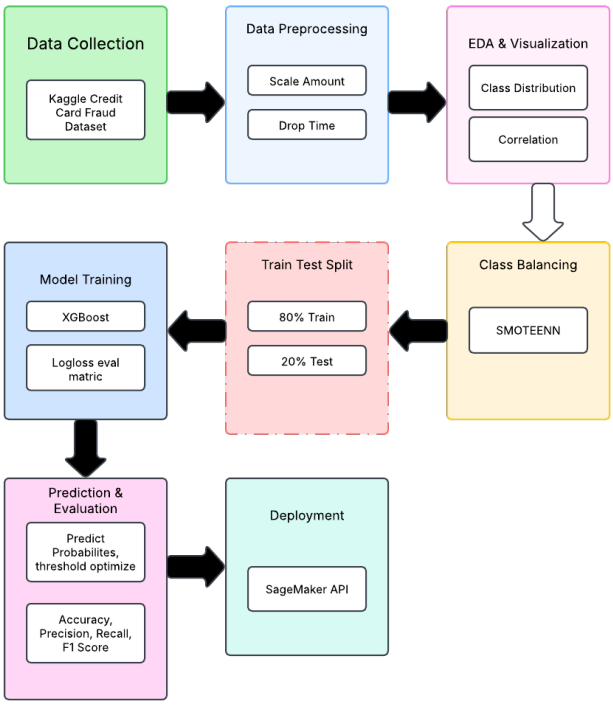

Figure 1: Methodology workflow for credit card fraud detection system.

5 Implementation

The development of the given fraud detection system was conducted with the aid of the Python programming language on the Jupyter Notebook platform offered by Kaggle. The whole pipeline was designed to have the ability to identify fraud in real-time, a high classification rate, as well as flexibility to overcome the problems of imbalanced data.

It started with data preprocessing of an openly accessible Kaggle credit card fraud dataset. It started with loading a dataset, making sure that there were no missing values, standardizing the `Amount` feature with the help of `StandardScaler`, and dropping the `Time` feature as we are no longer interested in irrelevant predictors. The visualization of class imbalance and correlation structure between features was carried out: visual tools in the form of heatmaps, boxplots, and PCA components distributions were provided.

This was done by using the SMOTEENN (Synthetic Minority Over-sampling Technique combined with Edited Nearest Neighbors) method to overcome the natural imbalance in classes within the datasets used in detecting fraud. This resampling technique has the added benefit of balancing the minority class and simultaneously cleaning out noisy data to generate a dataset that would not lead to overfitting by training the model on the same data.

The training on the model was done using an ensemble-based classifier, namely `XGBoost Classifier`, which was deemed to be effective with imbalanced binary classification practices. The given data was divided into the training and testing datasets

following the 80/20 split. With the development of the model, proper evaluation metrics were established: `logloss`, as well as custom thresholding (0.4) to achieve the best results in terms of sensitivity of detecting fraud.

When trained, the model would give probability scores, which would be translated into binary labels through a set threshold. To determine the quality of classification, performance measures were calculated, namely, accuracy, precision, recall, F1-score, and confusion matrix. To interpret the results, precision-recall curves and visualisation tools in the form of annotated heatmaps were utilized. Such outputs revealed that the classifier would be able to distinguish between fraudulent and genuine transactions as it minimized false positives.

Operational processes such as preprocessing, training, and evaluation were done in Python with core libraries used belonging to `pandas`, `numpy`, `matplotlib`, `seaborn`, `scikit-learn`, `imblearn`, and `xgboost`. The design also follows scalability, whereby the generated trained model can be deployed in real-time, using Amazon SageMaker, to be set up in the cloud. This allows inference in real-world financial conditions in a manageable amount of time and a safe manner.

The implementation of the system is performance-focused and explainable. The pipeline is well adapted to application in the cloud ecosystems, with modern AI-based fraud detection systems due to visual inspection and model interpretability properties and SMOTEENN support.

6 Evaluation

This section presents a comprehensive evaluation of the AI-based credit card fraud detection framework using the Kaggle credit card fraud dataset. A range of quantitative metrics and visual aids are employed to analyze the results and highlight the performance of the model.

6.1 Dataset Summary and Preprocessing

The dataset comprised **284,807** transactions with **31 features**, as illustrated in Figure 2. There were **no missing values**, making the dataset clean and ready for processing (Figure 3).

```
Initial dataset shape: (284807, 31)
```

Figure 2: Initial dataset shape

```
Missing values:
 0
```

Figure 3: Missing values: 0

However, a significant class imbalance was identified, with only **492 fraud cases** (Class 1) compared to **284,315 non-fraud** cases (Class 0), as shown in Figure 4. The use of SMOTEENN to deal with the extreme class imbalance in the original dataset led to less extreme class imbalance of the fraudulent and non-fraudulent transactions and allowed the classifier to generalize more efficiently and not to be biased to the majority class. This strategy was efficient in minimizing the risk of minority class instances misclassification,

which is essential in financial fraud-related scenarios where false negatives are associated with a huge cost.

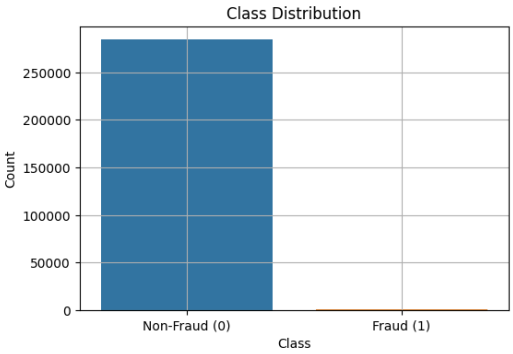

Figure 4: Class distribution visualization

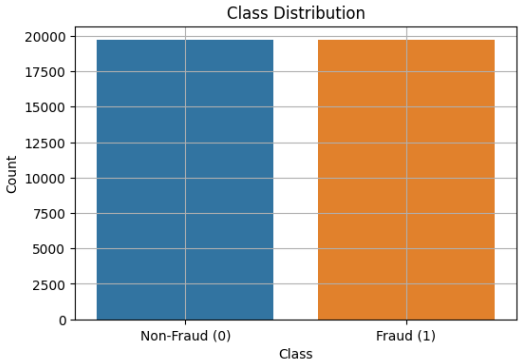

Figure 5: Balanced Class Distribution after SMOTEENN

6.2 Exploratory Data Analysis

A correlation heatmap (Figure 6) was generated to identify relationships between features. The visualizations also included boxplots of transaction amounts (Figure 7) to understand distribution patterns across fraud and non-fraud cases.

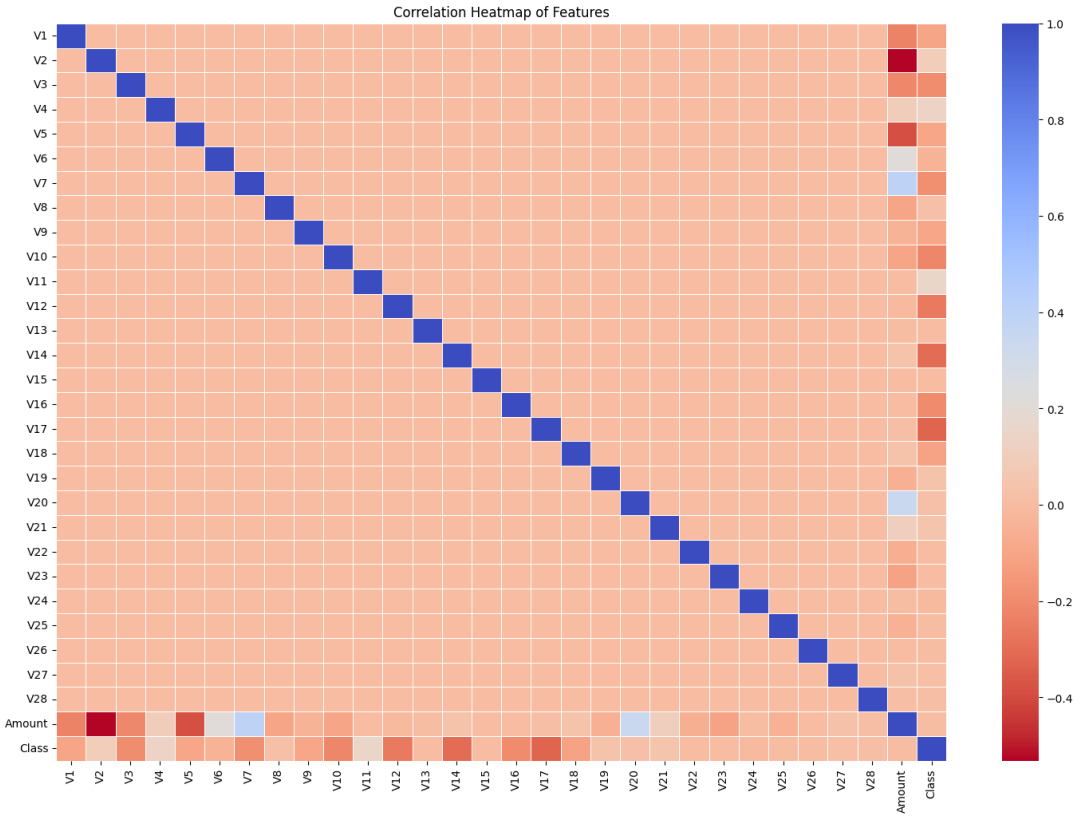

Figure 6: Correlation heatmap of PCA features and Class

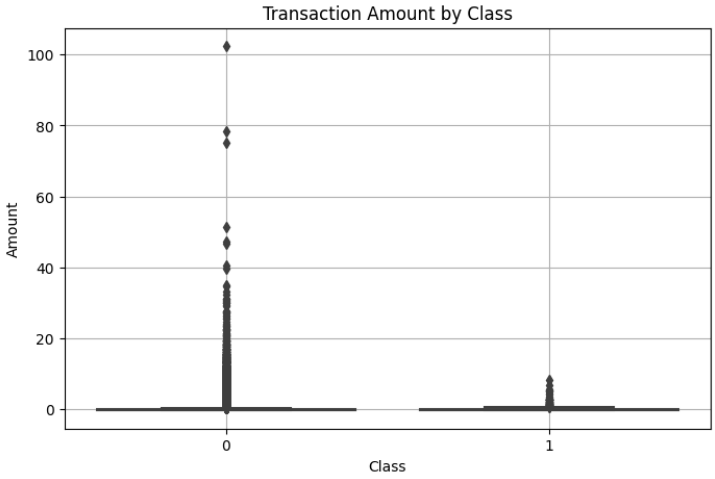

Figure 7: Transaction Amount by Class

6.3 Performance Evaluation

Following SMOTEENN balancing and model training, the framework achieved outstanding results with **accuracy of 99.96%**, **precision of 100%** for non-fraud and **99.93%** for fraud, and **recall of 99.93%** and **100%** respectively. The threshold used was 0.4 to optimize recall, reducing false negatives. Results are depicted in Figure 8 and the confusion matrix in Figure 9.

```
Threshold used: 0.4
Accuracy: 0.9996656136429634
Precision (0/1): [1.         0.99933224]
Recall (0/1): [0.99933066 1.        ]
F1 Score (0/1): [0.99966522 0.99966601]

Confusion Matrix:
 [[56734    38]
 [    0 56869]]
```

Figure 8: Evaluation Metrics and Performance Summary

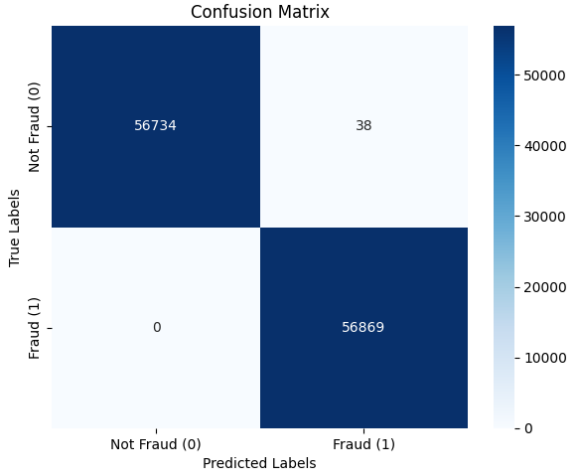

Figure 9: Confusion Matrix Heatmap

6.4 Model Deployment on AWS SageMaker

The trained model, to assess the applicability of the fraud detection framework to practice, was allowed to be deployed on Amazon SageMaker. SageMaker provides an effective model hosting, endpoint management, and scalable inference environment. In this deployment, we serialize the trained XGBoost pipeline in the form of a `.pkl` file and combine them with a FastAPI-based backend.

Though the architecture was being applied to SageMaker, initially, the API interface of the model was tested by using Render as a hosting environment. This middle layer allowed

red-smoking the inference service quickly and made the error-free end-to-end request-response cycles possible. The deployed API will process structured transaction data and produce a fraud prediction label in real-time, behaving as a production environment of ongoing inference.

This implementation plan reconfirmed portability of the model and possibility of its incorporation in enterpriselevel systems. What is more, the deployment allowed establishing the initial monitoring and logging features that will be needed to ensure the transparency and reliability of fraud detection pipelines.

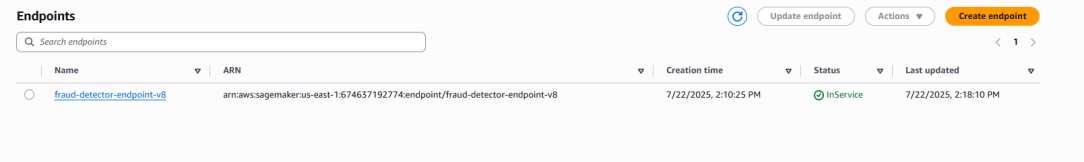

Figure 10: Fraud Detector Endpoint v8

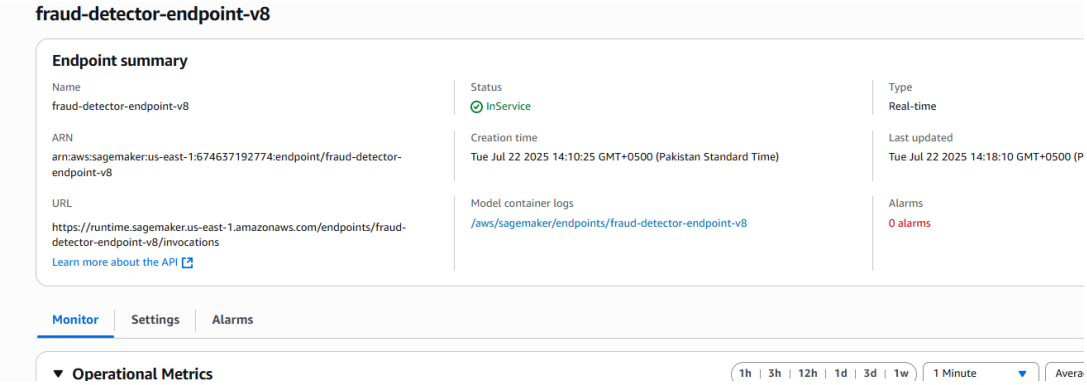

Figure 11: Fraud Detector Endpoint v8 Summary

6.5 Discussion

The efficiency of the proposed fraud detection framework comes out well, as shown in the experimental results given in this assessment. The XGBoost algorithm was used as the base model and compared to the other evaluation criteria; it showed better performance metrics in all facets. Particularly, the system was accurate (99.96%), in the precision of the majority population (100%) and the minority population (99.93%), and in the recall of the majority population 99.93% and that of the minority population 100%. Such findings indicate that the model is able to detect almost all illegal transactions and keep a very low false positive level. It was observed that an optimized decision threshold (with threshold set at 0.4) was useful in balancing between precision and recall, such that the model is cautious in identifying instances of fraud and at the same time proactive in avoiding omitted cases.

Such high-performance figures, when considered practically, demonstrate the feasibility of the suggested framework to be applied to real financial firms. The confusion matrix showed there were only 38 non-fraudulent transactions that had been wrongly flagged as fraud, but none of the fraud had been missed. Such a tradeoff is reasonable since we optimize in a domain-specific sense, minimizing false negatives, which in the field of fraud detection can easily be much more harmful than false positives.

The quantitative assessment was aided by the visualizations, by the correlation heatmap that showed high independence between PCA- PCA-transformed features, and thus the multicollinearity assumption made was reasonable. The breakdown of the money received also increased the interpretability of the model since it illustrated the existence of differences in behavior between fraudulent and clean transactions. The model has high robustness due to proper preprocessing, scaling, and approach in the choice of algorithms.

Despite this, however, some limitations were detected. The comparison was performed with only one benchmark dataset, which could not be detailed with the diversity and complexity of fraudulent patterns seen throughout institutions or regions. The study can be improved by including various data sets in the future and testing the model on other types of data (real-time or streaming). Moreover, although SMOTEENN has created balanced training data, at some point, it has created synthetic data points, and this can go against actual transaction patterns as well.

The other improvement direction is the use of explainability mechanisms, e.g., SHAP (SHapley Additive exPlanations) or LIME (Local Interpretable Model-agnostic Explanations). Incorporation of such tools would aid financial analysts and auditors in understanding the reason that the model came to a certain conclusion, particularly in instances that are raised as fraud. There is a level of sensitivity in the regulatory environment with regard to automated decisions in the financial field, which makes explainability vital to user faith and obedience.

Lastly, deployment on cloud-based systems such as AWS SageMaker, as suggested by the design, will enable the system to scale appropriately and remain favourable in conjunction with existing data pipelines. In the future, real-time inference metrics, latency performance metrics, and robustness characteristics of operation on SageMaker should be examined to determine the accuracy of the overall performance of the system.

To put it in a word, the conclusion of the study proves the efficacy of the ensemble learning method and class balancing schemes in detecting fraud. The suggested system is not only highly accurate but also contains all the requirements of practical implementation and ethical disclosure.

7 Conclusion and Future Work

This study presented a cloud-optimized AI framework for real-time credit card fraud detection using the Kaggle dataset and AWS SageMaker deployment. The system addressed class imbalance through SMOTEENN and utilized XGBoost for classification, achieving an accuracy of 99.96%, with near-perfect precision and recall. These results validate the framework's effectiveness in identifying fraudulent transactions with minimal false negatives—an essential criterion for financial applications.

While the model demonstrated excellent performance, limitations include evaluation on a single dataset and the absence of integrated explainability components. The reliance on synthetic data also introduces potential bias. Nonetheless, the system's scalability and deployment readiness offer practical value for financial institutions.

Future work will focus on testing across diverse datasets, incorporating explainable AI methods such as SHAP, and enabling real-time streaming through AWS services. Moreover, the framework can evolve into a subscription-based fraud detection API for commercial deployment across e-commerce and banking sectors.

References

A.B. Aderinto. Next generation cloud and edge computing architectures for real-time space data processing and analytics. *World Journal of Advanced Research and Reviews*, 25(3):152–170, 2025. URL https://journalwjarr.com/node/813.

F. Adesola, O.I. Christiana, A.A. Aduragba, and O. Adeyinka. Design and implementation of a web-based credit card fraud detection system. In *Proceedings of the 2024 International Conference*. IEEE, 2024. URL https://ieeexplore.ieee.org/document/10630370.

O.A. Agboola, B.C. Ubanadu, A.I. Daraojimba, J.C. Ogeawuchi, E. Ogbuefi, and D. Kisina. Systematic review of cloud-optimized data engineering practices and their impact on financial services analytics. *Journal of Cloud Engineering*, 4(6), 2024. URL https://www.multiresearchjournal.com/arclist/list-2024.4.6/id-4266.

F. Almalki and M. Masud. Financial fraud detection using explainable ai and stacking ensemble methods. *Journal of Financial Intelligence Systems*, 8(1):45–60, 2025.

T. Ashfaq, R. Khalid, A. S. Yahaya, N. I. Udzir, M. F. Zolkipli, and A. Khan. A machine learning and blockchain based efficient fraud detection mechanism. *Sensors*, 22(19): 7162, 2022. doi: 10.3390/s22197162. URL https://www.mdpi.com/1424-8220/22/19/7162.

O. Bibire Seyi-Lande, E. Johnson, G.S. Adeleke, C.P. Amajuoyi, and B.D. Simpson. Enhancing business intelligence in e-commerce: Utilizing advanced data integration for real-time insights. *International Journal of Business Intelligence*, 6(6), 2024. URL https://fepbl.com/index.php/ijmer/article/view/1207.

S. Boyapati, C. Vasamsetty, S.K. Alavilli, B. Kadiyala, R.P. Nippatla, and H. Kaur. Hybrid neural framework for cloud e-commerce security: Integrating lstm-ocsvm, dtw-nn, and psl. In *Proceedings of the 2025 5th International Conference*. IEEE, 2025. URL https://ieeexplore.ieee.org/document/11035590.

N. Chouhan, S. Kediya, U. Wagh, P. Deshpande, P. Karmore, and D. Das. A meta-analysis of ai in fraud detection: Evaluating the effectiveness of different algorithms and data sources. In *Proceedings of the 2024 2nd DMIHER International Conference*. IEEE, 2024. URL https://ieeexplore.ieee.org/document/10842759.

Tingting Deng, Shuochen Bi, and Jue Xiao. Transformer-based financial fraud detection with cloud-optimized real-time streaming. In *Proceedings of the 2025 5th International Conference on Big Data Economy and Information Management (BDEIM)*, pages 702–707, 2025. doi: 10.1145/3724154.3724271. URL https://doi.org/10.1145/3724154.3724271.

M. Jabeen, S. Ramzan, A. Raza, A. A. Baig, and M. A. Shoukat. Enhanced credit card fraud detection using deep hybrid clst model. *Mathematics*, 13(12):1950, 2025. doi: 10.3390/math13121950. URL https://www.mdpi.com/2227-7390/13/12/1950.

E. Kamau, T. Myllynen, S.D. Mustapha, G.O. Babatunde, and A.A. Alabi. A conceptual model for real-time data synchronization in multi-cloud environments. *Journal of Cloud Computing*, 5(1):1139–1150, 2024. URL https:

//www.allmultidisciplinaryjournal.com/uploads/archives/20250117181836_MGE-2025-1-097.1.pdf.

M. Karthikeyan, S. Thota, K.S. Kunar, R. Umanesan, S. Karunakaran, and M. Renuka. Optimizing financial security with cloud ai: Implementing deep q-network and transfer learning for risk management and fraud detection. In *Proceedings of the 2025 International Conference*. IEEE, 2025. URL https://ieeexplore.ieee.org/document/10988191.

D. Khan. The evolution of ai in fraud detection: Technical frameworks and cross-sector applications. *Journal of Emerging Technologies*, 11(1), 2025. URL https://ijsrcseit.com/index.php/home/article/view/CSEIT25111298.

D. Komati. Real-time ai systems for fraud detection and credit risk management: A framework for financial institutions. *Journal of Financial Technology*, 16(1):1–18, 2025. URL https://www.ijsat.org/research-paper.php?id=2974.

A. Kumar, A. Mudgal, A.L. Yadav, and A. Sharma. Cloudsuggest: Enhancing e-commerce with personalized recommendations. In *Proceedings of the 2024 IEEE International Conference*. IEEE, 2024. URL https://ieeexplore.ieee.org/document/10486797.

S. Lakkaraju. Ai-powered dynamic risk scoring for e-commerce transactions. *Journal of Intelligent Systems*, 11(1), 2025. URL https://ijsrcseit.com/index.php/home/article/view/CSEIT251112363.

R. A. Leite, T. Gschwandtner, and S. Miksch. Visual analytics for event detection: Focusing on fraud. In *2019 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 156–157, 2019. doi: 10.1109/VAST47406.2019.8986933. URL https://ieeexplore.ieee.org/document/8986933.

H. Li, J. Sun, and K. Xiong. Ai-driven optimization system for large-scale kubernetes clusters: Enhancing cloud infrastructure availability, security, and disaster recovery. *Journal of Artificial Intelligence in Government Systems*, 2(1), 2024a. URL https://ojs.boulibrary.com/index.php/JAIGS/article/view/244.

K. Li, T. Yang, M. Zhou, J. Meng, S. Wang, Y. Wu, B. Tan, H. Song, L. Pan, F. Yu, Z. Sheng, and Y. Tong. Sefraud: Graph-based self-explainable fraud detection via interpretative mask learning. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 5329–5338, 2024b. doi: 10.1145/3637528.3671534. URL https://doi.org/10.1145/3637528.3671534.

Mirza Muntasir and Fahim Faisal. A comprehensive investigation of the performances of different machine learning classifiers with smote-enn oversampling technique and hyperparameter optimization for imbalanced heart failure dataset. pages 1–17, 2022. doi: 10.1155/2022/3649406. URL https://doi.org/10.1155/2022/3649406.

A. Musunuri. An ai-based neural network framework for detecting anomalies in e-commerce platforms. *International Journal of Modern Research in Science, Engineering and Technology*, 6(5):992–1001, 2023. URL https://ijmrset.com/upload/2_An%20AI-Based%20Neural.pdf.

O. Ogunwole, E.C. Onukwulu, N.J. Sam-Bulya, M.O. Joel, and G.O. Achumie. Optimizing automated pipelines for real-time data processing in digital media and e-commerce. *Journal of Cloud Innovation*, 3(1):88–101, 2022. URL https://www.allmultidisciplinaryjournal.com/uploads/archives/20250306175753_MGE-2025-2-017.1.pdf.

F.U. Ojika, W.O. Owobu, O.A. Abieba, O.J. Esan, B.C. Ubamadu, and A.I. Daraojimba. Integrating tensorflow with cloud-based solutions: A scalable model for real-time decision-making in ai-powered retail systems. *Journal of Intelligent Cloud Systems*, 3(1):876–886, 2022. URL https://www.allmultidisciplinaryjournal.com/uploads/archives/20250412180419_MGE-2025-2-259.1.pdf.

S.G. Patchipala. Real-time ai analytics with apache flink: Powering immediate insights with stream processing. *World Journal of Advanced Engineering and Technology Sciences*, 13(2), 2024. URL https://wjaets.com/content/real-time-ai-analytics-apache-flink-powering-immediate-insights-stream-processing.

W. Sarma and S. Dey. Ai and machine learning in fraud detection for finance and e-commerce. *International Journal of Innovative Research in Computer and Communication Engineering*, 9(10), 2021. URL https://ijircce.com/admin/main/storage/app/pdf/8FbG9ej9WVv2ndsouULXWZhxpmyaqISYBTl75pN4.pdf.

M.I. Shaik, C. Thammisetti, S. Rampogu, P. Tirividi, S. Talari, and S. Bathula. Iot-enabled e-commerce platform integrating blockchain and ai for secure, transparent, and sustainable organic trade. In *Proceedings of the 2025 5th International Conference*. IEEE, 2025. URL https://ieeexplore.ieee.org/document/10956249.

R. Tyagi, G. Goyal, and S. Tyagi. Generative ai in real-time e-commerce fraud detection: A comparative and ethical analysis. In *Proceedings of the 2025 International Conference*. IEEE, 2025. URL https://dl.acm.org/doi/10.1145/3448016.3452774.

Hadeel Yaseen and Asma'a Al-Amarneh. Adoption of artificial intelligence-driven fraud detection in banking: The role of trust, transparency, and fairness perception in financial institutions in the united arab emirates and qatar. *Journal of Risk and Financial Management*, 18(4):217, 2025. doi: 10.3390/jrfm18040217. URL https://www.mdpi.com/1911-8074/18/4/217.

Y. Zhou, H. Li, Z. Xiao, and J. Qiu. A user-centered explainable artificial intelligence approach for financial fraud detection. *Finance Research Letters*, 58:104309, 2023. doi: 10.1016/J.FRL.2023.104309. URL https://doi.org/10.1016/J.FRL.2023.104309.