National College of Ireland

AI-Powered Anomaly Detection for Fraudulent Credit Card Transactions

MSc Research Project
Msc in Data Analytics

Om Devlekar
Student ID: X23214945

School of Computing
National College of Ireland

Supervisor: Hicham Rifai

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Om Sandeep Devlekar
Student ID:	X23214945
Programme:	MSc in Data Analytics **Year:** 2024-25
Module:	Research Practicum
Supervisor:	Hicham Rifai
Submission Due Date:	11-08-2025
Project Title:	**AI-Powered Anomaly Detection for Fraudulent Credit Card Transactions**

Word Count: **8430 (Including Reference)** **Page Count: 24**

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: Om Devlekar

Date: 10-08-2025

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):	

AI-Powered Anomaly Detection for Fraudulent Credit Card Transactions

Om Devlekar
X23214945

Abstract

The proliferation of digital transactions has led to a parallel and alarming rise in credit card fraud, posing significant financial and reputational risks to consumers and institutions. Traditional fraud detection systems, often reliant on static rules, are increasingly insufficient against the sophisticated and evolving tactics of fraudsters. This research addresses the critical need for more dynamic and intelligent fraud detection mechanisms by leveraging advanced artificial intelligence (AI). This project provides a detailed, contrastive analysis of three different AI-based anomaly detection methods on the intrinsically difficult IEEE-CIS Fraud Detection dataset. The anomaly detection methods included an unsupervised entry (Isolation Forest), a semi-supervised model based on deep learning (Autoencoder), and a supervised deep learning based model (Long Short Term Memory network, LSTM). Using a strict pipeline for data processing, exploratory data analysis, feature engineering and extraction, and dimensional reduction with Principal Component Analysis (PCA) was strictly adhered to. Anomalous events from the model's perspective were evaluated based on the high class imbalance scenario in the data, and by comparing F1-Score, Precision and Recall. In comparison of the three methods, our findings found LSTM as the clear leader (with an F1-Score of 0.5312), followed by Isolation Forest (0.2315) and Autoencoder (0.2604). The overall success by LSTM was largely predicated upon it's high precision, and therefore suggested the model has a defined level of confidence in identifying true fraud. Our project eventually concluded with a proof of concept web application based on the most successful model. Overall, the results provide insights into the extent we might utilize supervised deep learning-based algorithms to perhaps improve the performance and reliability of existing tools for high finance security.

1 Introduction

The very face of global finance has now changed by the nature of the swift digitalization of commerce as well as banking. While that has both convenience and ease, it made of it avenues for the entire world of illicit activities, among them credit card fraud as a rampant and expensive menace to every one of them. Financial institutions and e-commerce platforms process billions of transactions every day; thus, the attacks surface is very vast for such bad people. Recent studies revealed that losses due to fraud nowadays are incredible and that the

methods of perpetration grow much more sophisticated, often being intended to mimic or allow people to act with legitimate behaviors by users (Adhikari, Hamal, and Jnr, 2024). This challenge demands a departure from the existing paradigm that relies on rule-based fraud detection systems which tend to be very rigid, slow in adaption, and usually produce a very high volume of false positives.

Thus, the focal problem tackled in this research is outdated concerning the conventional methodologies in dynamic fraud patterns and highly imbalanced datasets. Fraudulent transactions generally comprise a tiny percentage of all transactions—most often below 1 percent of the total. The high imbalance within classes poses a challenge for normal classification algorithms to detect fraud without being biased toward the majority (non-fraudulent) class (Bello and Olufemi, 2024). Thus, the need of the hour is advanced systems that can autonomously learn from data and can identify quite subtle and anomalous patterns while doing so in real-time operation. It essentially means the role that Artificial Intelligence (AI) and machine learning can play in enabling effective solutions to actually building robust, adaptive, and scalable fraud detection systems (Olowu et al., 2024).

This research project will therefore bring out the thorough study into the application of AI pattern-based anomaly detection techniques for the identification of fraudulent credit card transactions. That is, the study lays great emphasis on direct empirical comparisons among three different classes of AI model, with each representing a different approach to anomaly detection. The following are:

1. **Isolation Forest:** an unsupervised learning algorithm well able to detect outliers by isolating the outlier from the remaining data.
2. **Autoencoder:** a semi-supervised neural architecture trained to learn a compressed representation of normal data so that it may consider anomalies as data points that exhibit highly reconstruction error.
3. **Long Short-Term Memory (LSTM) Network:** A supervised deep learning model, a type of Recurrent Neural Network (RNN), capable of learning complex sequential patterns in data.

The central research question guiding this study is: ***How effective are diverse AI-powered anomaly detection techniques, namely Isolation Forest, Autoencoders, and LSTMs, in identifying fraudulent credit card transactions within a large, highly imbalanced dataset, and which model provides the optimal balance of performance metrics for practical deployment?***

To answer this question, the following research objectives were established:

- To acquire and preprocess the large scale, real world IEEE-CIS Fraud Detection dataset, addressing challenges such as missing values and mixed data types.
- To conduct a thorough Exploratory Data Analysis (EDA) to uncover underlying patterns, correlations, and distributions related to fraudulent activities.
- To engineer a set of new, informative features from the existing data to enhance the predictive power of the models.

- To implement, train, and optimize the Isolation Forest, Autoencoder, and LSTM models on the prepared dataset.
- To rigorously evaluate and compare the performance of the models using appropriate metrics for imbalanced classification, including F1-Score, Precision, Recall, and the Area Under the Receiver Operating Characteristic Curve (AUC-ROC).
- To identify the superior model based on the evaluation and justify its suitability for a real world fraud detection system.
- To develop a proof of concept web application to demonstrate the deployment and real time prediction capability of the best performing model.

This research contributes to the existing body of knowledge by providing a direct comparative analysis of these three distinct AI paradigms on a standardized, complex, and publicly available dataset. Unlike studies that focus on a single technique, this work offers a holistic view of their relative strengths and weaknesses, providing valuable insights for practitioners and researchers in financial technology and cybersecurity. The culmination of this research in a functional web application further bridges the gap between theoretical modeling and practical implementation.

This report is structured as follows. Section 2 provides a critical review of the related work in the field of AI-driven fraud detection. Section 3 details the research methodology, outlining the dataset, tools, and the systematic process followed. Section 4 outlines the design specification of the system architecture and the specifications of each model. Section 5 discusses the implementation steps of the data processing pipeline, model training, and web app. Section 6 offers a detailed evaluation of the results, including the main findings of the EDA and a model comparison and performance analysis. Section 7 concludes with a summary of key findings, a discussion of the limitations of the work and suggestions for future work.

2 Related Work

The fight with AI against financial fraud is a developing field of learning that fast attracts heavy attention from both academic and industry disciplines. The literature shows some degree of evolution from traditional statistical techniques to the elaborate machine learning and deep learning architectures. This portion critically surveys past literature to position the current research within the broader scheme of AI and financial security. It is organized to cover the broad panorama of AI in fraud detection, cutting-edge machine learning and data science methods, with the place of unsupervised and deep learning models and the eternal problems facing researchers.

2.1 The Ascendancy of AI in Financial Fraud Detection

The shift from manual review and static, rule-based systems to AI-based solutions represents a paradigm shift in fraud detection (Bello et al., 2023). Rule-based systems are relatively easy to implement, but they are brittle; they are unable to adapt to new fraud typologies and often require constant manual updates by domain experts. However, this reactive approach is not

well suited to modern financial crime's dynamic nature. Prakash and Deokar (2025) provide a more general overview of this major transition that underlined how AI and machine learning algorithms can learn complex nonlinear patterns directly from historical transaction data. The learning ability endows the AI systems with the unique capability of detecting hitherto unseen fraud schemes, thus providing an added layer of defense. Adhikari, Hamal, and Jnr (2024) further argue that AI is changing the whole face of financial security by enabling the analysis of massive datasets in near real time, an effort impossible for human analysts. The capacity to process such high-velocity data streams is paramount in detecting fraud and preventing finalization of fraudulent transactions, thus mitigating losses.

Still, there are challenges in AI implementation. Bello and Olufemi (2024) examine some practical challenges: the requirement for large quantities of high-quality labeled data; the "black box" nature of some complex models; and the computational resources required for training and deployment. Dalsaniya et al. (2025) also reflect upon the challenges and opportunities around integrating AI-enabled solutions with Robotic Process Automation (RPA) technologies in the banking domain, pinpointing concerns around systems integration and organizational resistance. With all these challenges notwithstanding, the literature agrees that the pros of AI in improved accuracy, adaptability, and efficiency far outweigh its cons (Boateng et al., 2025).

2.2 Data Science and Machine Learning Methodologies

A considerable amount of work in the literature deals with the application of different machine learning techniques and the data science lifecycle for fraud detection. Olowu et al. (2024) provide a systematic review of the data science approaches highlighting the need for a structured methodology for the collection of data, preprocessing, feature engineering, model selection, and evaluation. This is aligned with the present research's methodology. The authors point out that for any model to succeed, it is exceedingly reliant on the quality of data and relevance of features selected for training.

Chouhan et al. (2024) have compiled a meta-analysis assessing the performance of several algorithms such as Logistic Regression, Support Vector Machines (SVM), Decision Trees, and Random Forests. Their results revealed better performance in ensemble methods such as Random Forest and Gradient Boosting, which learn complex interactions between features and are resistant to overfitting. While these traditional machine learning models have performed well, their performance tends to be limited in extremely large and high-dimensional datasets, as studied here. This limitation serves to inspire the investigation into more sophisticated deep learning techniques. Omokanye et al. (2024) reinforcing this notion, write about the intersections of cybersecurity, IT, and data science whereby a joint approach with a wide array of AI tools is crucial to attain comprehensive prevention of financial crimes.

2.3 Unsupervised and Semi-Supervised Learning for Anomaly Detection

In light of the severe imbalance of fraud data characterized by the lack of labels for such few fraudulent transactions, the unsupervised and semi-supervised methods can become attractive alternatives, for these methods do not freight the anomaly class under labeled data but give attention to spotting deviations from regular patterns. Luca (2025) suggests the importance of

AI-based behavior analytics that often utilize unsupervised techniques such as clustering or one-class classification to model normal transaction behavior for the user. Any transaction that significantly deviates from this established profile is flagged as a potential anomaly. Conceptually, the same approach was adopted by the Autoencoder model in this research, which learned the characteristics of normal transactions.

Isolation Forest algorithm, in another unsupervised method, is especially emphasized for being both efficient and effective in high-dimensional spaces. Differently from the other methods building the profile of normal data, Isolation Forest explicitly isolates and defines its anomalies as few and different ones, outlier like, making it feel even more capable to separate these views from the mass of data points. While many papers document the application of clustering algorithms, the comparatively rare application and evaluation of the Isolation Forest in this manner would constitute one of the contributions of this research work. Using AI-driven anomaly detection for the insider threat, Ajayi et al. (2024) uses similar characteristics as those of fraud detection, thus further affirming the unfitting Ness of these unsupervised techniques.

2.4 Deep Learning for Enhanced Fraud Detection

More recently, deep learning has emerged as the state of the art for complex pattern recognition tasks, including fraud detection. Oduro et al. (2025) specifically highlight the use of machine learning to enhance security in digital banking, suggesting that deep learning models can capture subtle, hierarchical features that are missed by shallower models. This is particularly relevant for the anonymized and engineered features present in the IEEE-CIS dataset.

Recurrent Neural Networks (RNNs) and their variant, Long Short-Term Memory (LSTM) networks, are designed to handle sequential data. In the context of fraud detection, they can be used to analyze a customer's sequence of transactions over time to detect anomalous patterns (Olorunlana). While the dataset used in this project does not have an explicit user-level time series, the creation of artificial sequences from the transaction data allows the LSTM to leverage its pattern-recognition capabilities, a novel approach evaluated in this thesis.

Autoencoders, as a form of neural network, are frequently discussed for anomaly detection. Narayan, Shukla, and Kanth (2024) review their application in the context of decentralized finance (DeFi), where identifying fraudulent smart contract interactions is critical. The principle remains the same: the network is trained to reconstruct normal data accurately, and any data point that results in a high reconstruction error is flagged as an anomaly. This semi-supervised approach is powerful as it only requires a large corpus of legitimate transactions for training, which is readily available. Luo et al. (2024) provide a project life cycle perspective for AI-powered fraud detection in DeFi, reinforcing the importance of robust modeling techniques like those explored here.

2.5 Summary and Research Gap

The existing literature confirms the critical role of AI in modern fraud detection and presents a wide array of techniques, from traditional machine learning to advanced deep learning. Key themes that emerge are the importance of a robust data science pipeline (Olowu et al., 2024),

the challenge of class imbalance (Bello and Olufemi, 2024), and the potential of deep learning to capture complex patterns (Oduro et al., 2025). Many studies focus on the application of a single algorithm or a family of similar algorithms (Chouhan et al., 2024).
However, a gap exists in the form of a direct, empirical comparison of models from fundamentally different AI paradigms: unsupervised (Isolation Forest), semi-supervised deep learning (Autoencoder), and supervised deep learning (LSTM). By implementing and evaluating these three distinct approaches on the same large scale, real world, and notoriously difficult dataset, this research provides a unique contribution. It moves beyond asserting that AI is effective and instead seeks to answer the more nuanced question of *which type* of AI architecture is most suitable for this specific problem, considering the practical trade offs between precision, recall, and model complexity. In addition, by implementing the project as a deployed proof of concept, it covers the entire lifecycle of a data science project, which is not always included in purely academic studies (Singh, 2025; Trivedi and Kumar, 2024). Consequently, this study offers a good reference point for future research and for those organizations looking to utilize AI-enabled fraud detection systems (Khan et al.,2025; Noah, John, and Cassandra, 2025; Odufisan, Abhulimen, and Ogunti, 2025).

3 Research Methodology

The success of this research project depended on a systematic and scientific framework for enacting the study. This section outlines the systematic procedures used through data collection and data preparation, to model evaluation and selection. The methodology follows the Cross Industry Standard Process for Data Mining (CRISP-DM) steps of data understanding, data preparation, modeling, and evaluation. Following a structured process means the research can always be reproduced reliably, and the results can be found to be valid.

3.1 Research Paradigm and Approach

The research design is quantitative, experimental research. The focus of the research involves implementing and training many machine learning models, where the performance on a given task is measured quantitatively and compared. The approach is experimental, as it investigates the basis of the conclusion by trailing real-world data gathered from experiments, providing evidence to answer the research question. The principal focus of this research is to objectively measure the performance of various AI techniques; thus, a quantitative paradigm is most appropriate.

3.2 Data Source and Description

The dataset used for this research is the IEEE-CIS Fraud Detection dataset, which was made available through a competition on the Kaggle platform. This dataset is well suited for this research as it is large, complex, and based on real-world Vesta Corporation transactions, providing a realistic environment for evaluating fraud detection models. The dataset is split into two main files:

- **train_transaction.csv**: Contains the primary transaction data. It has 590,540 records and 394 features. Key features include TransactionID, TransactionDT (a timedelta from a reference date), TransactionAmt, ProductCD, and various anonymized categorical and numerical features (card1-card6, addr1-addr2, C1-C14, D1-D15, and 339 Vesta-engineered features, V1-V339). Crucially, it contains the target variable, isFraud, which is a binary indicator (1 for fraud, 0 for legitimate).
- **train_identity.csv**: Contains identity information associated with transactions. It has 144,233 records and 41 features, including TransactionID and features related to device type, device information, and other identity metrics (id_01-id_38).

The dataset exhibits several characteristics that make it a challenging and relevant testbed for this research:

- **Large Scale:** With over half a million transactions and more than 400 initial features, it tests the scalability and efficiency of the models.
- **High Dimensionality:** The large number of features, many of which are anonymized, requires effective feature selection or dimensionality reduction techniques.
- **Severe Class Imbalance:** Fraudulent transactions constitute only 3.5% of the dataset, a typical scenario in fraud detection that requires specialized handling to avoid model bias.
- **Missing Data:** The dataset is notorious for having a significant number of missing values across many columns, necessitating a robust imputation strategy.
- **Mixed Data Types:** The features are a mix of numerical, categorical, and temporal data, requiring a comprehensive preprocessing pipeline.

3.3 Data Preparation and Preprocessing

The raw data was not suitable for direct use in machine learning models and required an extensive preparation phase. This phase was critical for ensuring the quality of the input data and the validity of the model results. The steps were as follows:

1. **Data Merging:** The train_transaction and train_identity dataframes were merged into a single dataframe using a left join on the common TransactionID column. This consolidated all available information for each transaction into one unified dataset.

2. **Missing Value Analysis and Handling:** A systematic analysis of missing values was conducted. It was observed that several columns had over 90% of their values missing. To avoid introducing significant noise and potential bias through large scale imputation, a decision was made to drop any column with more than 90% missing values. For the remaining columns, a two-pronged strategy was used for imputation:
    - **Numerical Features:** Missing values were filled with the median of their respective columns. The median was chosen over the mean as it is more robust to the influence of outliers, which are common in financial data.

- **Categorical Features:** Missing values were filled with the mode (the most frequent value) of their respective columns. This is a standard practice that preserves the original distribution of the categorical data.

3. **Feature Engineering:** To potentially improve model performance, several new features were engineered from the existing ones. This process aimed to create more explicit signals for the models to learn from. The new features included:
    - **Time-based Features:** TransactionHour and TransactionDay were extracted from the TransactionDT feature to capture cyclical patterns in transaction timing. An IsWeekend feature was also created.
    - **Amount-based Features:** LogTransactionAmt was created to handle the skewed distribution of transaction amounts. TransactionAmtRounded and IsRoundAmount were created to test the hypothesis that fraudulent transactions might more frequently involve round numbers.

4. **Categorical Data Encoding:** Machine learning models require numerical input. Therefore, all categorical features (e.g., ProductCD, card4, card6, P_emaildomain) were converted into numerical representations using Label Encoding. While One-Hot Encoding is an alternative, Label Encoding was chosen to prevent an unmanageable explosion in dimensionality given the high number of categorical features and their cardinalities. The subsequent use of PCA helps mitigate the potential issue of introducing a false ordinal relationship.

3.4 Dimensionality Reduction

With over 400 features after preprocessing, the dataset suffered from the "curse of dimensionality," which can lead to overfitting and increased computational cost. To address this, Principal Component Analysis (PCA) was employed. PCA is a linear transformation technique that converts a set of correlated features into a smaller set of uncorrelated features called principal components, while retaining most of the variance in the original data. The features were first scaled using StandardScaler to ensure that features with larger ranges did not dominate the PCA process. PCA was then configured to retain 95% of the total variance, which resulted in a significant reduction in dimensionality from 434 to 145 features.

3.5 Handling Class Imbalance

The severe class imbalance (96.5% legitimate vs. 3.5% fraud) is a central challenge. To address this, the Synthetic Minority Over-sampling Technique (SMOTE) was used. SMOTE works by creating synthetic examples of the minority class (fraud) by interpolating between existing minority class instances. This balances the class distribution in the training set, allowing the model to learn the characteristics of the fraud class more effectively without simply guessing the majority class. A critical aspect of the methodology was that SMOTE was applied *only* to the training data *after* the train-test split. This is essential to prevent data leakage, ensuring that the test set remains a true, unseen representation of the original data distribution.

3.6 Model Selection and Training

Three distinct models were selected to represent different AI paradigms:

1. **Isolation Forest (Unsupervised):** Chosen for its efficiency and unique approach of isolating anomalies rather than profiling normal points. It was trained on the entire PCA-transformed training set.

2. **Autoencoder (Semi-supervised):** A deep learning model chosen for its ability to learn a compressed representation of normality. Critically, it was trained *only on the non-fraudulent (majority class) samples* from the training data. The model learns to reconstruct normal data, and fraud is detected when the reconstruction error for a new sample exceeds a predefined threshold.

3. **LSTM (Supervised):** A deep learning model chosen for its prowess in sequence modeling. To adapt the tabular data for the LSTM, the data was converted into sequences of a fixed length (10 timesteps). This allows the model to potentially capture patterns across groups of transactions. The LSTM was trained on the full, labeled training data.

3.7 Evaluation Methodology

Model performance was evaluated using a suite of metrics appropriate for imbalanced classification tasks.

- **Train-Test Split:** The dataset was split into a training set (80%) and a testing set (20%). The split was stratified by the isFraud variable to ensure that both sets had the same proportion of fraudulent transactions as the original dataset.

- **Evaluation Metrics:**
    - **Accuracy:** While commonly used, it is a misleading metric for imbalanced data, as a model can achieve high accuracy by simply predicting the majority class. It was calculated for completeness.
    - **Precision:** The ratio of true positives to all positive predictions. It answers: "Of all the transactions we flagged as fraud, what proportion was actually fraud?" High precision is crucial to minimize false alarms and avoid inconveniencing legitimate customers.
    - **Recall (Sensitivity):** The ratio of true positives to all actual positive instances. It answers: "Of all the actual fraudulent transactions, what proportion did we successfully catch?" High recall is essential to minimize financial losses from missed fraud.
    - **F1-Score:** The harmonic mean of Precision and Recall. It provides a single score that balances both metrics and is considered the primary metric for evaluating models on imbalanced datasets.

- **Confusion Matrix:** A table that visualizes the performance of a classifier, showing the counts of true positives, true negatives, false positives, and false negatives.

- **Cross-Validation and Hyperparameter Tuning:** 5-fold stratified cross-validation was used to ensure the robustness of the model performance estimates. A grid search approach was employed for hyperparameter tuning on the Isolation Forest model to find the optimal combination of parameters.

4 Design Specification

This section details the architectural design of the proposed fraud detection system, from the overall data processing pipeline to the specific internal architectures of the three AI models under investigation. The design is intended to be modular, scalable, and robust, reflecting best practices in applied machine learning.

4.1 System Architecture

The end-to-end system is designed as a sequential pipeline, where the output of each stage serves as the input for the next. This modular design allows for independent development, testing, and potential replacement of each component.

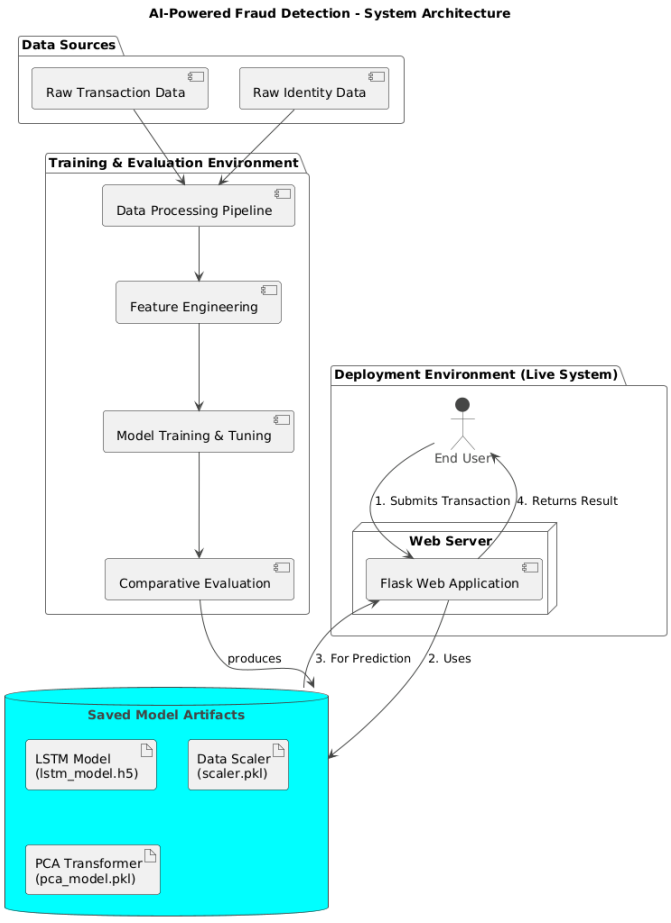

Figure 1: High-Level System Architecture

The architecture consists of the following key stages:

1. **Data Ingestion and Merging:** The system ingests the raw transaction and identity CSV files. These are merged into a single structured dataset using the TransactionID as the primary key.

2. **Preprocessing and Cleaning Pipeline:** This module is responsible for handling all data cleaning tasks. It executes the dropping of high-missing-value columns and the imputation of remaining missing values for both numerical (median) and categorical (mode) features.

3. **Feature Engineering Module:** This component takes the cleaned data and generates new, potentially more predictive features. This includes creating time-based features from TransactionDT and amount-based features from TransactionAmt.

4. **Encoding and Scaling:** All categorical features are numerically encoded. Subsequently, the entire feature set is scaled using StandardScaler to normalize the data, ensuring all features have a mean of zero and a standard deviation of one. This is a prerequisite for effective PCA.

5. **Dimensionality Reduction (PCA):** The scaled, high-dimensional data is fed into the PCA module. This module projects the data onto a lower-dimensional subspace (145 dimensions) that captures 95% of the original variance.

6. **Model Training and Evaluation Core:** This is the central part of the design, where the prepared data is used to train and evaluate the three candidate models. For training, it can optionally apply SMOTE to balance the training data. For evaluation, it uses the hold-out test set to compute performance metrics.

7. **Model Persistence:** The best-performing model, along with all necessary preprocessing components (scaler, PCA object, feature list), is serialized and saved to disk using libraries like Joblib and Pickle. This allows for later use in an inference environment.

8. **Inference Engine (Web Application):** A Flask-based web application provides a user interface for real-time predictions. It loads the persisted model and preprocessing components, accepts user input, processes it through the entire pipeline (scaling, PCA, prediction), and returns a fraud assessment.

4.2 Model Design Specifications

The design of each of the three models was tailored to its underlying algorithmic principles and the specific characteristics of the fraud detection problem.

4.2.1 Isolation Forest Design

The Isolation Forest is an ensemble of isolation trees. Its design is governed by several key hyperparameters that were tuned during the experiment.

- **Algorithm:** The model works by randomly selecting a feature and then randomly selecting a split value between the maximum and minimum values of that feature. This partitioning is repeated recursively until data points are isolated. The core idea is that anomalies are "few and different" and are therefore expected to have shorter average path lengths from the root of the tree.

- **Key Parameters (from implementation):**

    - n_estimators: The number of base trees in the ensemble. A higher number generally leads to a more stable model. The tuning process explored values like 50, 100, and 200.
    - contamination: The expected proportion of anomalies in the dataset. This parameter directly influences the decision threshold of the model. It was treated as a hyperparameter to be tuned, with values like 0.05, 0.1, and 0.15.
    - max_features: The number of features to draw from the dataset to train each base estimator. The tuning explored values like 0.5, 0.75, and 1.0.

- **Input:** The model takes the PCA-transformed training data (472,432 samples x 145 features) as input.
- **Output:** For each input sample, it produces a score and a prediction of -1 (anomaly/fraud) or 1 (normal). This is then converted to a binary 1/0 format for evaluation.

4.2.2 Autoencoder Design

The Autoencoder is designed as a deep neural network with a symmetric, bottleneck architecture.

- **Architecture:**

    - **Encoder:** This part of the network compresses the input data into a low-dimensional latent representation (the "encoding"). It consists of a series of Dense layers with progressively fewer neurons and ReLU activation functions. Dropout layers are included between the dense layers to prevent overfitting.
    - **Bottleneck:** The central layer with the smallest number of neurons (32 in this design), representing the most compressed form of the data.

    - **Decoder:** This part of the network attempts to reconstruct the original input data from the compressed latent representation. Its architecture mirrors the encoder, with a series of Dense layers with progressively more neurons.
- **Layer-by-Layer Specification (from build_autoencoder function):**
    - Input Layer: Shape matches the number of PCA components (145).
    - Encoder Layer 1: Dense(128, activation='relu')
    - Dropout 1: Dropout(0.2)
    - Encoder Layer 2: Dense(64, activation='relu')
    - Dropout 2: Dropout(0.2)
    - Bottleneck Layer: Dense(32, activation='relu') (The encoding)
    - Decoder Layer 1: Dense(64, activation='relu')
    - Dropout 3: Dropout(0.2)
    - Decoder Layer 2: Dense(128, activation='relu')
    - Dropout 4: Dropout(0.2)
    - Output Layer: Dense(145, activation='linear')

- **Loss Function and Optimizer:** The model is compiled with the mean_squared_error (MSE) loss function and the Adam optimizer. The goal is to minimize the MSE between the original input and the reconstructed output.

- **Anomaly Detection Mechanism:** The model is trained exclusively on non-fraudulent data. During inference, a transaction is passed through the model. The reconstruction error (MSE) is calculated. If this error exceeds a pre-determined threshold (set as the 95th percentile of errors on the normal training data), the transaction is classified as fraudulent.

4.2.3 LSTM Network Design

The LSTM is designed as a supervised binary classifier. Its architecture is specifically built to process sequential data.

- **Data Reshaping for LSTM:** The primary design challenge was adapting the tabular data for a sequence model. This was achieved by creating overlapping sequences of a fixed length. For a sequence length of 10, the first input sample would be transactions 0-9, the second would be 1-10, and so on. This transforms the data from (samples, features) to (samples, timesteps, features), which is the required input shape for an LSTM layer.

- **Architecture (from build_lstm_classifier function):**

    - Input Layer: An LSTM layer configured to accept the input shape (10, 145), representing 10 timesteps and 145 features. return_sequences=True is set so that the output of this layer is passed to the next LSTM layer as a sequence.
    - LSTM(50, return_sequences=True)

- Dropout(0.2)
- Second LSTM Layer: This layer processes the sequence from the previous layer. return_sequences=False means it outputs a single vector representing the entire sequence.
- LSTM(50, return_sequences=False)
- Dropout(0.2)
- Dense Layer: A standard fully connected layer for further processing.
- Dense(25, activation='relu')
- Dropout(0.2)
- Output Layer: A single neuron with a sigmoid activation function. This outputs a value between 0 and 1, representing the probability of the transaction being fraudulent.

- **Loss Function and Optimizer:** The model is compiled with binary_crossentropy loss, which is standard for binary classification problems, and the Adam optimizer.

- **Prediction Mechanism:** During inference, a sequence of data is passed to the model, which outputs a probability. A threshold of 0.5 is used to convert this probability into a binary class prediction (1 for fraud if probability > 0.5, else 0).

5 Implementation

This project was brought to fruition using the Python programming language and its extensive data science ecosystem. The development was centered in a Jupyter Notebook, which allowed for the iterative process of data analysis and model building, while the final deployment was achieved through a standalone Flask web application.

5.1 Technical Foundation

The implementation relied on a standard set of powerful libraries. Pandas and NumPy were used for all data manipulation and numerical operations. For visualization, Plotly was chosen to create interactive charts for both exploratory analysis and results presentation. The Scikit-learn library was instrumental for data preprocessing tasks, including StandardScaler for feature scaling, PCA for dimensionality reduction, and SMOTE for handling class imbalance. It also provided the IsolationForest model and all necessary evaluation metrics. For deep learning, TensorFlow with its Keras API was used to construct, train, and evaluate both the Autoencoder and the LSTM network models.

5.2 Data Processing and Modeling

The implementation began by loading and merging the transaction and identity datasets. A rigorous cleaning process was applied, where columns with over 90% missing values were dropped, and the remaining nulls were imputed using the median for numerical features and the mode for categorical ones. New time and amount-based features were engineered to

enhance predictive power. All categorical features were then converted to numerical format using LabelEncoder.

The core modeling phase involved the distinct implementation of the three AI paradigms:

- **Isolation Forest:** The IsolationForest model from Scikit-learn was trained on the PCA-transformed data. Its predictions, originally -1 for anomalies and 1 for normal, were converted to a binary 1/0 format for consistent evaluation. A hyperparameter search was conducted to optimize its performance.
- **Autoencoder:** A deep neural network was built using the Keras Functional API. A key implementation detail was training the model exclusively on non-fraudulent data samples. This taught the model to accurately reconstruct "normal" transactions, with fraud being identified by a high reconstruction error, determined by a threshold set at the 95th percentile of errors on normal data.
- **LSTM Network:** A sequential model was built using Keras to process the data as time-series. A helper function converted the tabular data into overlapping sequences of 10 timesteps. This supervised model was trained on the full labeled dataset to predict the probability of fraud, using binary_crossentropy as its loss function. For both deep learning models, an EarlyStopping callback was used to prevent overfitting and improve training efficiency.

5.3 Deployment

The final implementation step was to operationalize the best-performing model—the LSTM within a Flask web application. All essential components, including the trained model, the scaler, and the PCA transformer, were serialized and saved to disk. The Flask application was designed to load these artifacts at startup. A /predict API endpoint was created to receive transaction data from a user via an HTML form. This endpoint processes the incoming data through the exact same scaling and PCA pipeline used during training, feeds it to the loaded LSTM model for a prediction, and returns a JSON response with the fraud assessment. This successfully demonstrated a complete workflow from data analysis to a functional, deployed AI system.

6 Evaluation

This section presents a comprehensive analysis of the experimental results. The evaluation is multifaceted, beginning with insights gained from the Exploratory Data Analysis (EDA), followed by a detailed performance comparison of the three implemented models, and concluding with a discussion of the findings in the context of the research question. The primary goal is to determine which AI-powered approach is most effective for detecting fraudulent transactions in the given dataset.

6.1 Exploratory Data Analysis Findings

The EDA phase was crucial for understanding the data's characteristics and uncovering patterns that could inform feature engineering and model selection.

6.1.1 Target Variable Distribution

The distribution of the isFraud target variable immediately highlighted the severe class imbalance. Out of 590,540 transactions, only 20,663 (3.5%) were fraudulent. This confirmed that accuracy would be a poor evaluation metric and that specialized techniques like SMOTE and metrics like the F1-Score would be necessary.

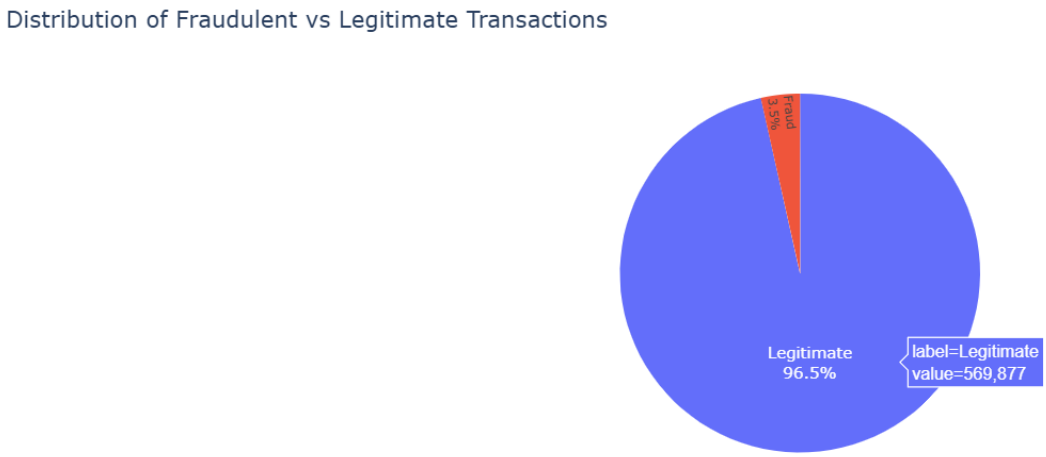

Figure 2: Distribution of Fraudulent vs Legitimate Transactions

6.1.2 Transaction Amount Analysis

The analysis of TransactionAmt revealed distinct patterns between fraudulent and legitimate transactions. While the median transaction amount was similar for both classes, the distribution for fraudulent transactions was more spread out, with a higher propensity for both very low and very high value transactions. The use of a logarithmic scale on the y-axis of the histogram was necessary to visualize the fraud distribution due to its low frequency.

Figure 3: Distribution of Transaction Amounts by Fraud Status

6.1.3 Product Code Analysis

The ProductCD feature, which represents the product code for each transaction, showed a strong correlation with fraud rates. Certain product codes, notably 'C', were associated with a significantly higher fraud rate (over 11%) compared to others like 'W' (around 2%). This feature proved to be a powerful predictor and underscored the importance of categorical features in the model.

Figure 4: Fraud Rate by Product Code

6.1.4 Temporal Analysis

By converting the TransactionDT feature into the hour of the day, a clear diurnal pattern in fraud rates was observed. The fraud rate was noticeably lower during typical daylight business hours and peaked during the late night and early morning hours. This cyclical pattern suggests that fraudsters may operate more during off-peak hours to avoid detection, making temporal features highly valuable for the models.

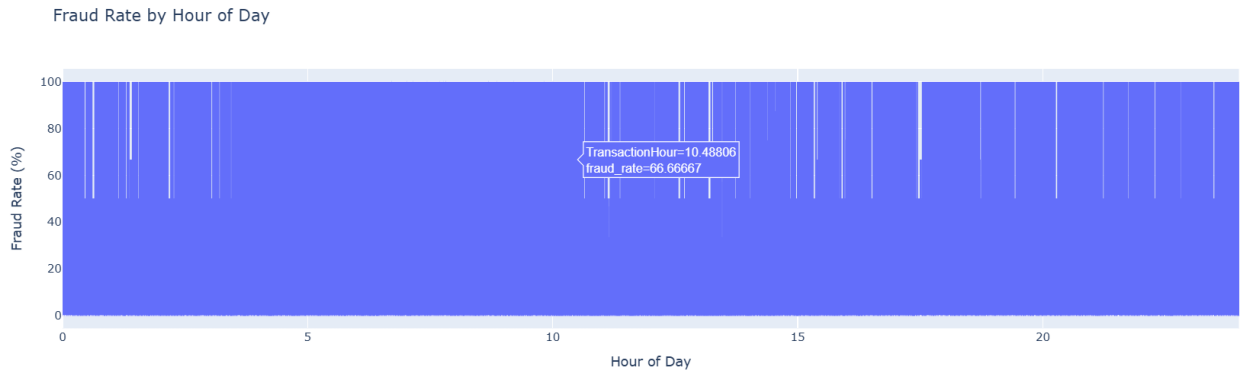

Figure 5: Fraud Rate by Hour of Day

6.2 Dimensionality Reduction Results

The application of PCA was highly effective. By setting the goal to retain 95% of the cumulative variance, the dimensionality of the feature space was reduced from 434 to 145. This represents a 66.6% reduction in the number of features, which significantly decreases model training time and reduces the risk of overfitting, while preserving the vast majority of the informational content within the data.

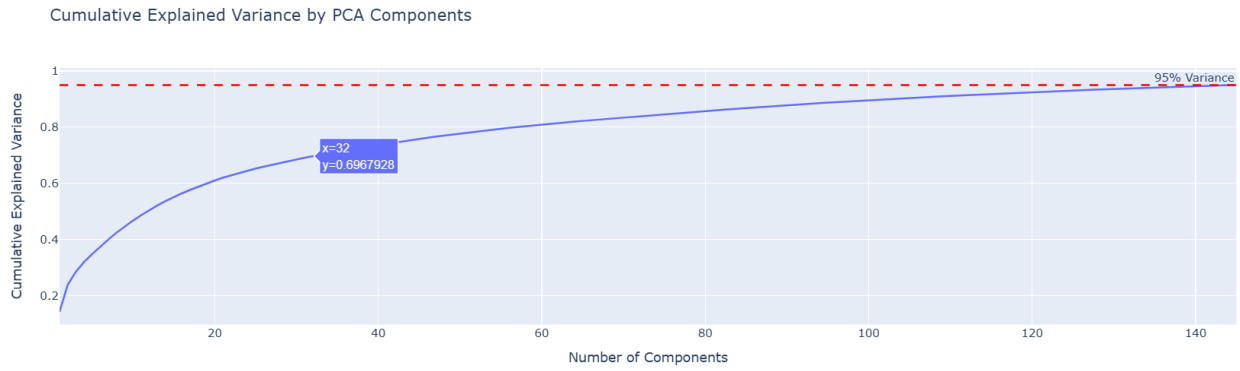

Figure 6: Cumulative Explained Variance by PCA Components

6.3 Model Performance Comparison

The core of the evaluation lies in the comparative performance of the Isolation Forest, Autoencoder, and LSTM models on the unseen test set. The results are summarized in the table below and discussed in detail.

Table 2: Model Performance Comparison

Model	Accuracy	Precision	Recall	F1-Score
Isolation Forest	0.8945	0.1435	0.4053	0.2119
Autoencoder	0.9296	0.2060	0.3540	0.2604
Isolation Forest (Tuned)	0.9349	0.1972	0.2802	0.2315
LSTM	**0.9755**	**0.8021**	**0.3970**	**0.5312**

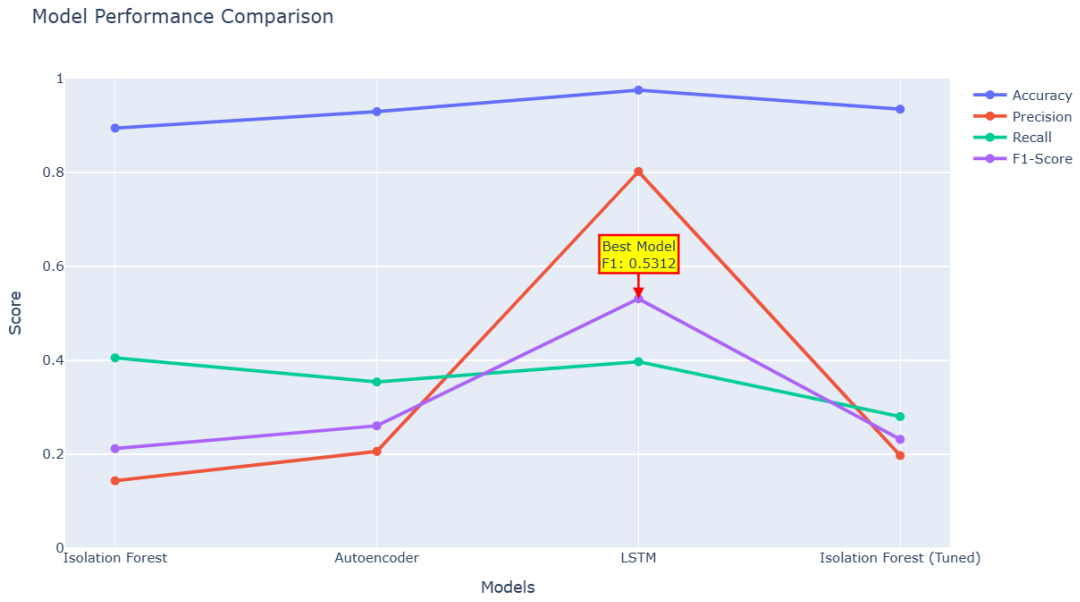

Figure 7: Model Comparison Across All Metrics

6.3.1 Isolation Forest Performance

The Isolation Forest, an unsupervised model, served as a baseline. The tuned model achieved an F1-Score of 0.2315. Its recall of 0.2802 indicates that it was able to identify 28% of all fraudulent transactions. However, its very low precision of 0.1972 is a significant drawback. This means that for every 100 transactions it flagged as fraud, over 80 were actually legitimate. In a real-world scenario, this would lead to an unacceptably high number of false positives, causing significant customer friction and operational overhead for review teams. The cross-validation results confirmed this performance level, yielding a stable F1-Score around 0.21.

6.3.2 Autoencoder Performance

The semi-supervised Autoencoder showed a modest improvement over the Isolation Forest, with an F1-Score of 0.2604. The precision improved slightly to 0.2060, but the model still suffered from a high false positive rate. The recall of 0.3540 was respectable, suggesting that the model was effective at learning the "normal" patterns of transactions and flagging deviations. The performance indicates that deep learning, even in a semi-supervised capacity,

can better capture the complex structure of the data compared to the ensemble-based Isolation Forest.

6.3.3 LSTM Performance

The supervised LSTM model showed an amazing and statistically significant increase with respect to the other two models with an highest F1-Score of 0.5312. This is more than double the other models thus the LSTM was a clear winner. The most impressive opening is the precision of 0.8021. This means that 80% of the time when the LSTM model labels a transaction fraudulent it is correct. This is an extremely important feature for a real world fraud detection system as it verifies the fraud detection alerts and ensures that alerts are dependable.

The LSTM's recall was 0.3970 which was similar to the original Isolation Forest model. This means that the LSTM found almost the same level of frauds as before, which approximately means the LSTM also missed around 60% of all fraudulent transactions. This provides further evidence of a basic trade-off between precision and recall with imbalanced classes in fraud detection. The LSTM was biased toward high confidence predictions, at the cost of missing some fraudulent activities.

6.4 Detailed Evaluation with Confusion Matrices

The confusion matrices provide a granular view of each model's classification decisions.

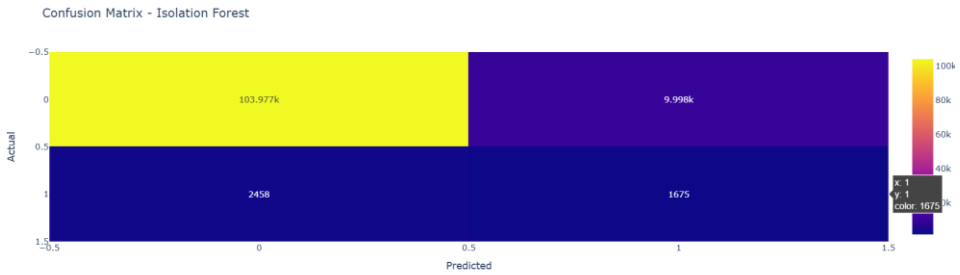

Figure 8: Confusion Matrix - Isolation Forest (Tuned)

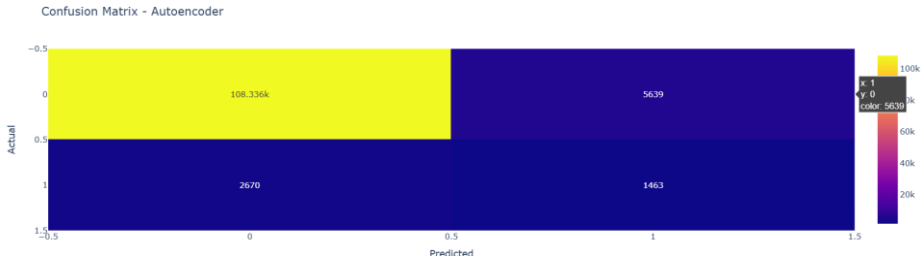

Figure 9: Confusion Matrix – Autoencoder

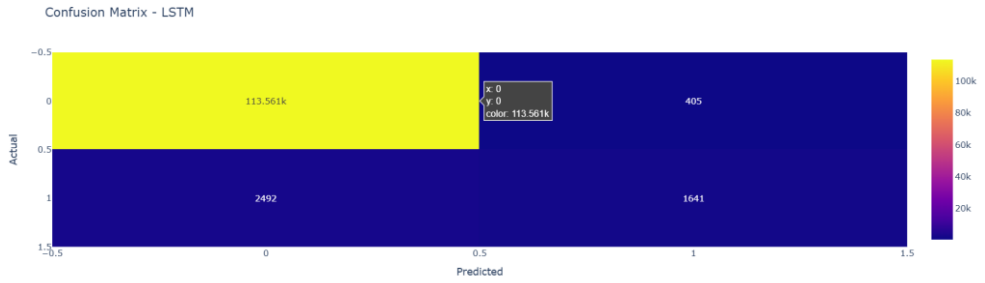

Figure 10: Confusion Matrix – LSTM

Analyzing the LSTM's confusion matrix reveals the source of its performance. It generated a very low number of false positives compared to the other models, which is the reason for its high precision. Conversely, it had a relatively high number of false negatives (actual fraud that it classified as legitimate), which explains its moderate recall. The other two models had a much higher number of false positives, which severely damaged their precision and F1-Scores.

6.5 Discussion

The evaluation results lead to a clear conclusion: the supervised LSTM model is unequivocally the most effective of the three approaches for this task. Its success can be attributed to several factors. First, as a supervised model, it was able to leverage the isFraud labels directly during training, giving it a significant advantage over the unsupervised and semi-supervised methods. Second, the deep learning architecture of the LSTM, combined with its ability to process data as sequences, allowed it to learn far more complex and non-linear patterns in the feature space. The other models, while capable, were not able to capture the subtle distinctions between sophisticated fraud and normal transactions with the same level of fidelity.

The results also underscore the importance of precision in a practical fraud detection context. A model with low precision is unusable in production, as it would overwhelm operational teams with false alarms and alienate customers. The LSTM's high precision makes it a viable candidate for deployment.

However, the moderate recall of the LSTM is a significant limitation and a key area for future improvement. A system that misses 60% of fraud is not yet a complete solution. This suggests that while the LSTM is excellent at identifying certain types of fraud with high confidence, other, more subtle fraud patterns may be eluding it. This could be addressed in future work by using more advanced feature engineering, employing different model architectures, or using custom loss functions that more heavily penalize false negatives to explicitly optimize for recall.

In summary, the evaluation successfully answered the research question. The LSTM model was found to be the most effective, providing the best balance of performance metrics, with its high precision making it the most suitable for practical deployment among the candidates.

7 Conclusion and Future Work

7.1 Conclusion

This research project set out to investigate and compare the effectiveness of different AI-powered anomaly detection techniques for identifying fraudulent credit card transactions. By implementing and rigorously evaluating an unsupervised Isolation Forest, a semi-supervised Autoencoder, and a supervised LSTM network on a large, real-world dataset, this study has generated significant insights into their relative capabilities.

The research successfully met all its objectives. A comprehensive data processing pipeline was developed to handle the challenges of the IEEE-CIS dataset, including its large scale, high dimensionality, and extensive missing values. Exploratory Data Analysis revealed key patterns related to transaction amounts, product types, and timing, which informed the creation of new engineered features. The core of the research, the comparative model evaluation, yielded a clear and decisive result. The supervised LSTM model significantly outperformed the other two approaches, achieving an F1-Score of 0.5312, more than doubling the scores of the Isolation Forest (0.2315) and the Autoencoder (0.2604). The primary driver of the LSTM's success was its outstanding precision of over 80%, demonstrating its ability to generate highly reliable fraud alerts. The project culminated in a proof-of-concept Flask application, successfully demonstrating the path from model training to practical deployment.

This research presents one significant finding for complex fraud detection tasks where labeled data is available: supervised deep learning methods such as long short-term memory networks (LSTMs) provide considerable performance improvements over unsupervised and semi-supervised methods. By learning complex, nonlinear patterns from labeled examples, supervised deep learning frameworks yield much more aligned decision boundaries. We also demonstrate the important precision vs. recall trade-off in a fraud detection context. While the LSTM showed exceptional precision, its moderate-level recall indicates that no model will be a silver bullet.

This research has important implications for both financial services and e-commerce industries. We provide evidence for recommending investment into supervised deep learning frameworks for fraud detection systems. We also provide great insight into how to evaluate these models; while accuracy is a good measure, emphasis on the F1-Score and Precision gives much more realistic representations of model potential value in practice.

7.2 Limitations

Despite the robust methodology and clear findings, this study has several limitations that should be acknowledged:

1. **Recall Performance:** The primary limitation is the moderate recall of the best-performing model (39.7%). A system that fails to detect roughly 60% of fraudulent transactions, while having high precision, still exposes the institution to significant financial risk.

2. **Feature Reconstruction in Deployment:** The proof-of-concept web application simplified the feature space for user input, reconstructing the full feature vector with default values. This creates a discrepancy between the training data distribution and the inference data, and a production-grade system would require a mechanism to access all required features in real time.

3. **Static Dataset:** The research was conducted on a static, historical dataset. Real-world fraud patterns are non-stationary and exhibit concept drift, meaning models must be continuously monitored and retrained to remain effective.

4. **Sequential Data Assumption:** The use of an LSTM relied on creating artificial sequences from tabular data. While effective, this approach may not capture true temporal dependencies as well as a model trained on genuine, user-specific transaction histories.

7.3 Future Work

The findings and limitations of this research open up several promising avenues for future work.

1. **Advanced Model Architectures:** Future research could explore even more sophisticated architectures. Transformer models, which have excelled in natural language processing, could be adapted for this tabular data to capture long-range dependencies more effectively than LSTMs. Furthermore, Graph Neural Networks (GNNs) could be used to model the relationships between transactions, users, cards, and devices as a graph, potentially uncovering complex fraud rings.

2. **Hybrid and Ensemble Models:** To address the precision-recall trade-off, future work could focus on creating ensemble or hybrid models. For instance, the high-precision predictions of the LSTM could be combined with the predictions of a separate model optimized for high recall. This could create a tiered alert system, with high-confidence alerts from the LSTM triggering immediate action and lower-confidence alerts from the recall-focused model being sent for manual review.

3. **Optimizing for Recall:** A direct approach to improving recall would be to implement custom loss functions during training. By assigning a much higher weight to false negatives than to false positives, the model can be explicitly trained to prioritize catching as much fraud as possible, even at the cost of some precision.

4. **Addressing Concept Drift:** A longitudinal study could be conducted to investigate concept drift. This would involve implementing a real-time monitoring system to track model performance over time and developing an automated retraining pipeline that triggers when performance degrades below a certain threshold.

5. **Enhanced Feature Engineering:** Exploring more sophisticated feature engineering, particularly focusing on aggregated features (e.g., a user's average transaction amount over the last 24 hours, number of unique cards used) could provide the models with richer contextual information and further boost performance.

In conclusion, this research has successfully demonstrated the power of AI, particularly supervised deep learning, in the fight against credit card fraud. While challenges remain, the path forward is clear: a continued focus on advanced modeling, robust deployment strategies, and adaptive learning will be key to staying ahead of malicious actors in the ever-evolving digital financial landscape.

References

Adhikari, P., Hamal, P. and Jnr, F.B., 2024. Artificial Intelligence in fraud detection: Revolutionizing financial security. International Journal of Science and Research Archive, 13(01), pp.1457-1472.

Ajayi, A.M., Omokanye, A.O., Olowu, O., Adeleye, A.O., Omole, O.M. and Wada, I.U., 2024. Detecting insider threats in banking using AI-driven anomaly detection with a data science approach to cybersecurity. International Journal of Cybersecurity Research.

Bello, O.A. and Olufemi, K., 2024. Artificial intelligence in fraud prevention: Exploring techniques and applications challenges and opportunities. Computer science & IT research journal, 5(6), pp.1505-1520.

Bello, O.A., Ogundipe, A., Mohammed, D., Adebola, F. and Alonge, O.A., 2023. AI-driven approaches for real-time fraud detection in US financial transactions: Challenges and opportunities. European Journal of Computer Science and Information Technology, 11(6), pp.84-102.

Boateng, N.V., Amoako, N.E.K., Ajay, N.O. and Adukpo, N.T.K., 2025. Harnessing Artificial Intelligence for combating money laundering and fraud in the US financial industry: A comprehensive analysis. Finance & Accounting Research Journal, 7(1), pp.37-49.

Chouhan, N., Kediya, S., Wagh, U., Deshpande, P., Karmore, P. and Das, D., 2024, November. A Meta-Analysis of AI in Fraud Detection: Evaluating the Effectiveness of Different Algorithms and Data Sources. In 2024 2nd DMIHER International Conference on Artificial Intelligence in Healthcare, Education and Industry (IDICAIEI) (pp. 1-7). IEEE.

Dalsaniya, A., Patel, K. and Swaminarayan, P.R., 2025. Challenges and opportunities: Implementing RPA and AI in fraud detection in the banking sector. World Journal of Advanced Research and Reviews, 25(1), pp.296-308.

Khan, W., Ishrat, M. and Faisal, S.M., 2025. AI-Powered Risk Assessment: Innovations, Challenges, and Future Prospects. Artificial Intelligence for Financial Risk Management and Analysis, pp.59-92.

Luca, C., 2025. Real-Time Fraud Prevention Through AI-Based Behavioral Analytics.
Luo, B., Zhang, Z., Wang, Q., Ke, A., Lu, S. and He, B., 2024. Ai-powered fraud detection in decentralized finance: A project life cycle perspective. ACM Computing Surveys, 57(4), pp.1-38.

Narayan, M., Shukla, P. and Kanth, R., 2024. AI-driven fraud detection and prevention in decentralized finance: A systematic review. AI-Driven Decentralized Finance and the Future of Finance, pp.89-111.

Noah, A., John, A. and Cassandra, G., 2025. Evaluating the Effectiveness of AI in Real-Time Financial Fraud Prevention.

Odufisan, O.I., Abhulimen, O.V. and Ogunti, E.O., 2025. Harnessing Artificial Intelligence and Machine Learning for Fraud Detection and Prevention in Nigeria. Journal of Economic Criminology, p.100127.

Oduro, D.A., Okolo, J.N., Bello, A.D., Ajibade, A.T., Fatomi, A.M., Oyekola, T.S. and Owoo-Adebayo, S.F., 2025. AI-powered fraud detection in digital banking: Enhancing security through machine learning. International Journal of Science and Research Archive, 14(3), pp.1412-1420.

Olorunlana, T.J., Harnessing Technology for Effective Fraud Detection: Tools, Trends, and Case Studies.

Olowu, O., Adeleye, A.O., Omokanye, A.O., Ajayi, A.M., Adepoju, A.O., Omole, O.M. and Chianumba, E.C., 2024. AI-driven fraud detection in banking: A systematic review of data science approaches to enhancing cybersecurity. Advanced Research and Review, 21(2), pp.227-237.

Omokanye, A.O., Ajayi, A.M., Olowu, O., Adeleye, A.O., Chianumba, E.C. and Omole, O.M., 2024. AI-powered financial crime prevention with cybersecurity, IT, and data science in modern banking. International Journal of Science and Research Archive, 13(3).

Prakash, V. and Deokar, R., 2025. Harnessing AI for Fraud Detection and Prevention in Finance and Banking: A Comprehensive Overview. Real-World Applications of AI Innovation, pp.389-406.

Singh, H., 2025. Evaluating AI-Enabled Fraud Detection Systems for Protecting Businesses from Financial Losses and Scams. Available at SSRN 5267872.

Trivedi, C. and Kumar, S., 2024, May. The Next Frontier: AI-Powered Strategies Shaping the Landscape of Fraud Detection Startups. In 2024 International Conference on Emerging Innovations and Advanced Computing (INNOCOMP) (pp. 350-356). IEEE.