Customer fraud detection project

Table of Contents

1. Business Context 2
2. Project Objectives 2
3. Executive Summary 2
4. Background 3
    4.1 Dataset 4
5. Findings: Exploratory Data Analysis 5
    5.1 Transaction Trends 5
    5.2 Fraudulent Transactions at Late Hours 6
    5.3 Transaction Frequency and Burst Behaviour 7
    5.4 Higher Spend on Fraudulent Transactions 8
    5.5 Days Between Repeated Frauds 10
    5.6 Fraud by Merchant Category 10
    5.7 Amount Spent and Category-Based Fraud Patterns 12
6. Modelling 13
    6.1 Data Preparation 13
    6.2 Models 14
7. Results 18
    7.1 Results at Optimal F1 18
    7.2 LSTM Overfitting Tests 20
    7.3 Cost-Based Tuning 21
8. Recommendations and Limitations 24
    8.1 Recommendations 24
    8.2 Limitations 26
9. Model Deployment 27
10. Conclusion 28
11. References 28

1. Business context:

Fraud remains a significant threat to financial institutions and their customers worldwide, resulting in millions of fraudulent transactions and substantial financial losses each year. This project aims to generate actionable insights and predictive tools to support the early detection and prevention of fraudulent activity.

When estimating the cost of fraud, both business and bank perspectives are considered. From a business (merchant) standpoint, fraud and false declines directly affect sales revenue, customer trust, and operational expenses. From a banking perspective, the costs extend to reimbursements, regulatory penalties, and reputational damage. Together, these viewpoints provide a holistic framework for understanding the true financial impact of fraud, ensuring that model optimisation is guided not only by statistical accuracy but also by real-world economic consequences.

2. Project Objectives

1. Uncover key patterns and indicators associated with fraudulent transactions through data analysis to inform effective prevention strategies.
2. Develop a machine learning model that can accurately identify potentially fraudulent transactions, enabling proactive fraud mitigation.

3. Executive Summary

Fraud continues to represent a major threat to financial institutions and their customers, leading to substantial monetary and reputational losses each year. This project aimed to develop a data-driven fraud detection system capable of identifying high-risk transactions before they are processed. Using a large synthetic dataset of 1.8 million credit-card transactions, a range of models: logistic regression, XGBoost, and deep learning, were tested to find the most effective solution.

The final model, a Long Short-Term Memory (LSTM) network, achieved the highest overall performance, with an F1 score of 0.97, precision of 0.98, and recall of 0.95. Unlike traditional models that analyse single transactions, the LSTM leverages temporal patterns, learning how transaction behaviour evolves, allowing it to detect subtle anomalies missed by other methods.

To bridge statistical accuracy with real-world business relevance, the model's decision threshold was optimised not for the F1 score but for minimum financial cost, considering both merchant and banking perspectives. False negatives were costly at 3.68× the direct transaction amount (reflecting reimbursement, chargebacks, and reputational losses), while false positives captured lost sales and customer churn costs. This analysis yielded an optimal threshold of 0.011, reducing expected total cost per transaction to $0.17 and maintaining high detection accuracy.

The model was deployed on an AWS EC2 instance using FastAPI and Docker, enabling real-time fraud predictions via a REST API, demonstrating production readiness and scalability potential for live financial systems.

4. Background:

Fraudulent transactions not only siphon money from victims but also erode trust in digital commerce and banking systems. As electronic payments become ubiquitous, the scale of fraud is increasing dramatically. In 2023, Australians spent about $1.1 trillion on cards, yet card fraud grew 32 % to $762 million, with card-not-present (CNP) fraud accounting for more than 90 % of losses. The overall fraud rate climbed to 70.2 cents per $1 000 spent, a 22 % rise from 2022 (*Australian payment fraud 2024 report* 2025). This surge means that institutions must balance customer convenience with security to prevent escalating losses. Personal fraud is also widespread: an Australian Bureau of Statistics survey found that 1 in 10 Australians (≈2.1 million people) experienced card fraud in 2023-24 (*Personal fraud, 2023-24 financial year* 2025). Such prevalence underscores the need for robust detection systems that can quickly identify anomalous transactions without unduly inconveniencing customers.

Beyond direct monetary losses, fraud imposes substantial indirect costs on banks and merchants. Fraudulent transactions trigger chargebacks, legal expenses, manual reviews and customer churn costs that can far exceed the stolen amount. A 2024 LexisNexis True Cost of Fraud Study reported that for every $1 lost to fraud, Australian companies spend about AUD $3.68 on resolution; financial institutions bear even higher costs (≈$4.21 per $1), and 66 % of companies reported rising fraud. Digital channels were responsible for 51 % of fraud losses, reflecting the shift to online scams (*Every dollar lost to fraud in Australia costs firms AUD$3.68 according to LexisNexis true cost of fraud study* 2024).

Machine learning models provide a robust and cost-effective way to flag fraudulent transactions in real-time. As such, their adoption could save millions by preventing fraudulent transactions. Previous research by the MDPI systematic review identified several key indicators of financial fraud that are consistently flagged across machine learning studies. Transaction-based indicators include unusually high or round-number amounts, high-frequency transactions, and activity occurring at odd hours or across distant geolocations within short timeframes. Behavioural red flags involve sudden changes in user habits, such as using new devices, repeated failed login attempts, or accessing accounts from high-risk regions. Profile mismatches, like low-income users making large purchases, new accounts executing large transactions, or multiple accounts sharing the same credentials, also signal potential fraud. In corporate and financial statement fraud, anomalies in financial ratios (e.g., excessive asset turnover or self-financing) and unexplained rapid growth compared to industry norms are common indicators. Additionally, text-based features, such as urgency or pressure-related language in emails and communications, are emerging as subtle but powerful predictors (Ali et al., 2022). These indicators are most effective when analysed in combination, as machine learning models can capture complex patterns across transactions, behaviours, and profiles to detect fraud with high accuracy (Ali et al., 2022).

4.1 Dataset:

Data from: https://www.kaggle.com/datasets/kartik2112/fraud-detection?resource=download

The dataset, sourced from Kaggle, consists of 1,852,394 labelled transactions, of which 9,651 are fraudulent, representing approximately 0.52% of the total. It was synthetically generated, which may limit the generalizability of models and findings to real-world data. The dataset includes the following variables:

Figure 4.1.1. *Dataset overview and variable descriptions — Kaggle Fraud Detection Dataset.*

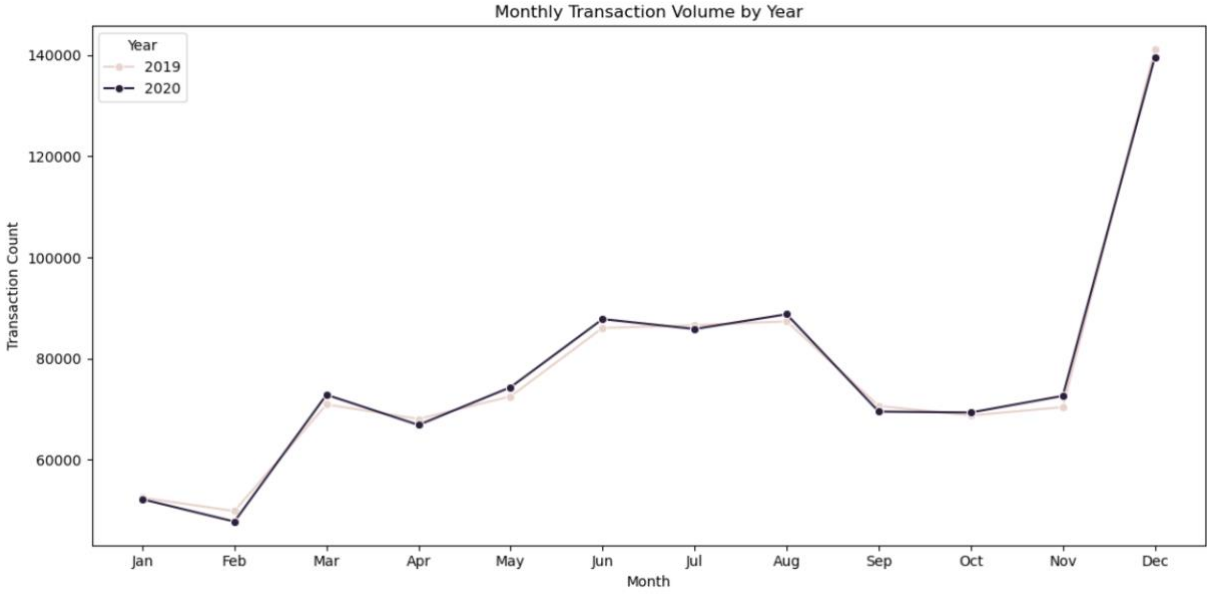
Variable Name	Type	Description
Unnamed: 0	Index	Row index (can be ignored)
trans_date_trans_time	DateTime	Timestamp of the transaction
cc_num	Categorical	Credit card number (anonymized)
merchant	Categorical	Name of the merchant
category	Categorical	Merchant category (e.g., gas, travel)
amt	Numeric	Transaction amount
first	Categorical	First name of cardholder
last	Categorical	Last name of cardholder
gender	Categorical	Gender of cardholder
street	Categorical	Street address of cardholder
city	Categorical	City of cardholder
state	Categorical	State of cardholder
zip	Numeric	ZIP code of cardholder
lat	Numeric	Latitude of cardholder
long	Numeric	Longitude of cardholder
city_pop	Numeric	Population of cardholder's city
job	Categorical	Job title of cardholder
dob	Date	Date of birth
trans_num	ID	Unique transaction ID
unix_time	Numeric	Unix timestamp of transaction
merch_lat	Numeric	Latitude of merchant
merch_long	Numeric	Longitude of merchant
is_fraud	Binary	Fraud label (1 = fraud, 0 = not fraud)

5. Findings: Exploratory Data Analysis

The dataset consists of 9651 fraudulent transactions from 999 unique bank accounts. Of those bank accounts, 976 of the numbers experienced fraudulent transactions. The high number of accounts experiencing fraud is a result of the simulated dataset not being reflective of the real world.

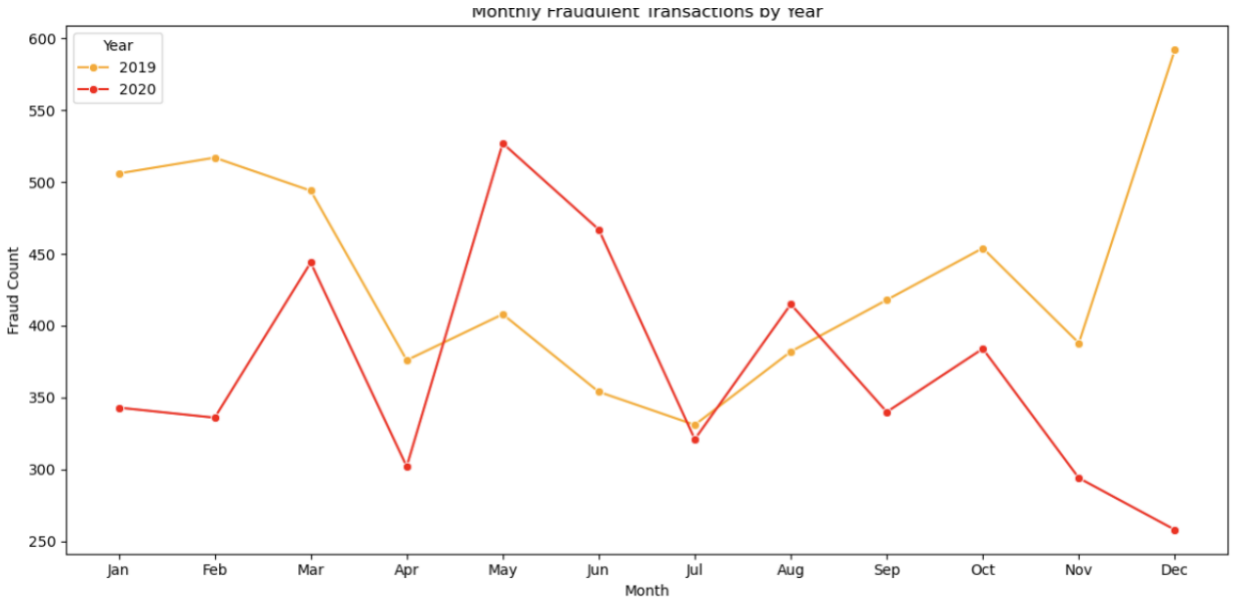
5.1 Transaction trends:

Figure 5.1.1. *Monthly transaction volume by year (2019–2020)*



The dataset spans from 2019 to 2020 and exhibits a distinct seasonal trend in legitimate transaction volumes. Specifically, transaction counts rise steadily throughout the year and spike significantly in December, likely due to holiday-related spending, while they drop around February, a pattern consistent with post-holiday financial behaviour. This trend is visualised in the plot of monthly transaction volumes by year.

Figure 5.2.1. *Monthly fraudulent transaction counts by year (2019–2020)*



In contrast, the analysis of fraudulent transaction counts reveals no clear seasonal trend. The number of fraudulent transactions appears relatively stable month to month, with fluctuations that do not align with the seasonal patterns observed in total transaction volume in 2020, but a spike in November 2019 is consistent with seasonal patterns. This discrepancy suggests that the month of the year alone may not be a strong indicator of fraud; however, it indicates the possibilities of time dependencies within the dataset.

5.2 Fraudulent Transactions at late hours

Figure 5.2.1. *Transaction counts by hour and fraud status — showing overall activity peaks during business hours.*

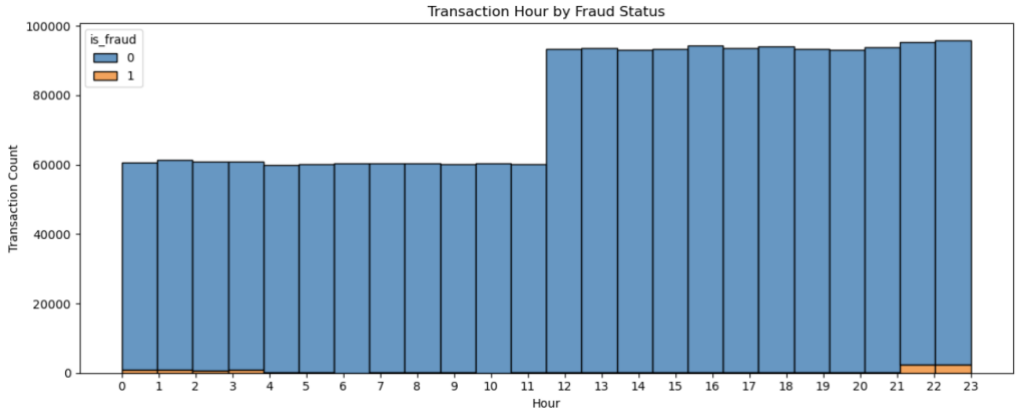
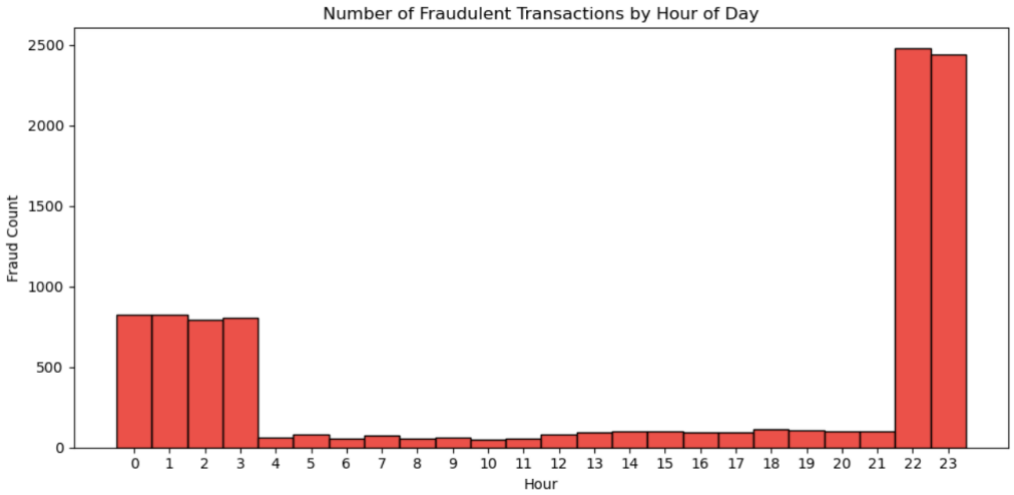


Figure 5.2.2. *Fraudulent transactions by hour — concentrated during late-night and late-evening periods.*

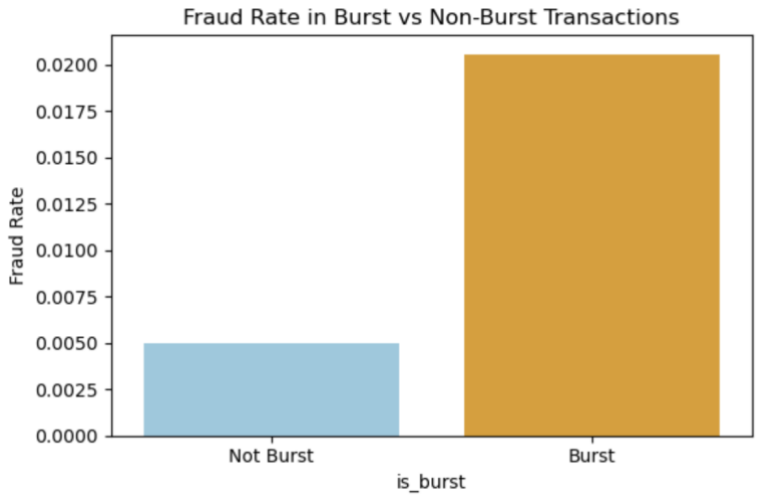


Visual analysis of transaction timing revealed that the hour of the day is a strong indicator of fraudulent behaviour. As shown in the plots, the overall distribution of transactions remains relatively consistent across the day, with higher activity during typical business hours. However, when isolating fraudulent transactions, a disproportionate number occur during atypical hours**,** particularly late at night (e.g., 0:00–3:00) and late evening (21:00–23:00)**.** This sharp contrast suggests that fraudsters may exploit times when legitimate user activity is lower, making the hour of transaction a valuable feature for identifying fraud**.**

5.3 Transaction frequency in burst behaviour

Exploratory analysis began by engineering a feature, time_since_last, measuring the number of seconds between consecutive transactions on each card. Any pair of transactions occurring within 5 minutes (300 seconds) was flagged as belonging to a burst. A bar chart comparing fraud rates in burst versus non-burst transactions (Figure 5.3.1) shows that fraud is about four times more likely in burst transactions (~2 %) than in isolated ones (~0.5 %). This striking difference immediately suggests that rapid succession is a meaningful signal of anomalous behaviour.

Figure 5.3.1. *Fraud rate in burst vs non-burst transactions — showing higher fraud likelihood in rapid successive (burst) activity.*



Time-gap distributions

To gain a deeper understanding of this burst effect, two histograms were plotted. The first histogram (Figure 5.3.2) displays the distribution of time gaps between consecutive transactions for all users. The distribution is positively skewed: the majority of gaps cluster around 10,000 seconds (~2.8 hours), implying that most customers transact only a few times per day. When isolating fraudulent transactions (Figure 5.3.3), the distribution shifts dramatically. There are two notable spikes:

- Very short intervals: A sharp peak at small time gaps confirms the burst hypothesis. Fraudsters often execute several rapid transactions to test a compromised card or extract funds quickly.
- Approximately one-day interval: A second spike near 86,400 seconds (1 day) suggests a recurring fraud pattern. Fraudsters may repeat attempts daily, perhaps to bypass simple velocity checks or exploit known behavioural windows.

The overall shape is far more concentrated at short intervals than the general population, underscoring the importance of incorporating time-based features (such as burst flags or rolling counts) into fraud detection models.

Figure 5.3.2. *Distribution of time gaps between transactions*

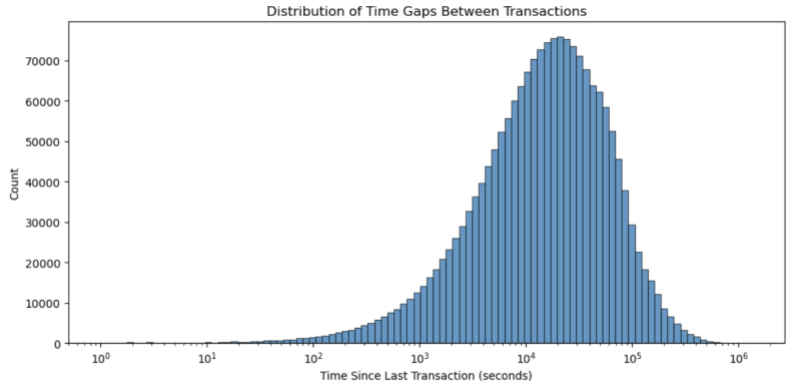
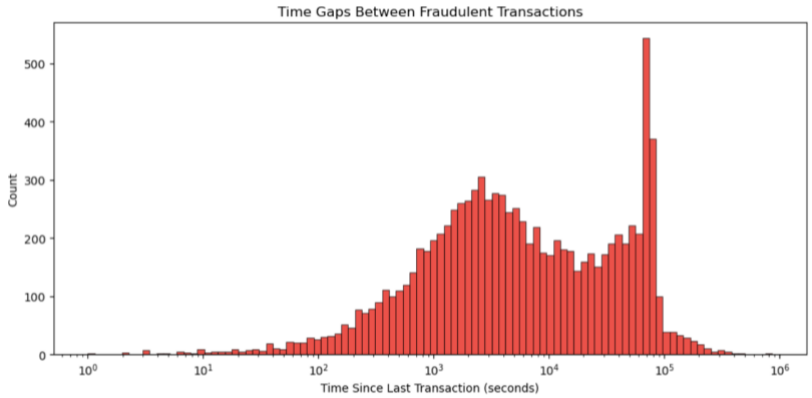


Figure 5.3.3. *Distribution of time gaps between Fraudlent transactions*



5.4 Higher Spend on Fraudulent Transactions

Next, transaction amounts were examined for all transactions versus fraudulent ones. Summary statistics (Figure 5.4.1) reveal that the mean transaction value across all data is $70.06 with a standard deviation of $159.25, and most legitimate transactions fall in the $0–$200 range. The histogram of all transactions (5.4.3) is heavily left-skewed, confirming that small everyday purchases dominate the dataset.

Figure 5.4.1. *Summary statistics of transaction amounts — showing higher mean and variance for fraudulent transactions.*

	All transactions	Fraudulent transactions
Mean	70.06	530.66
std	159.25	391
min	1	1
max	28,948.90	1376.04

Figure 5.4.2. *Transaction amount distribution of fraudulent transactions*

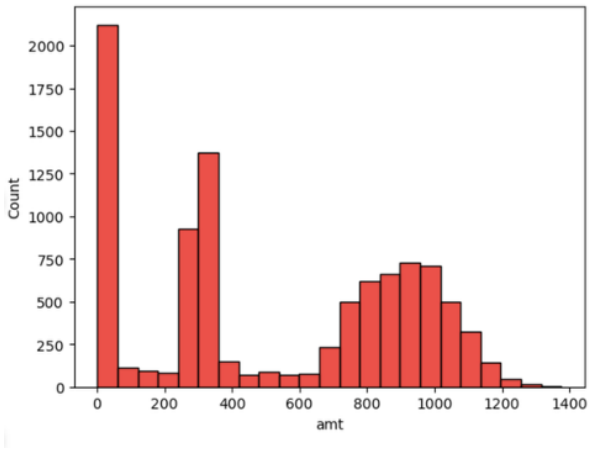
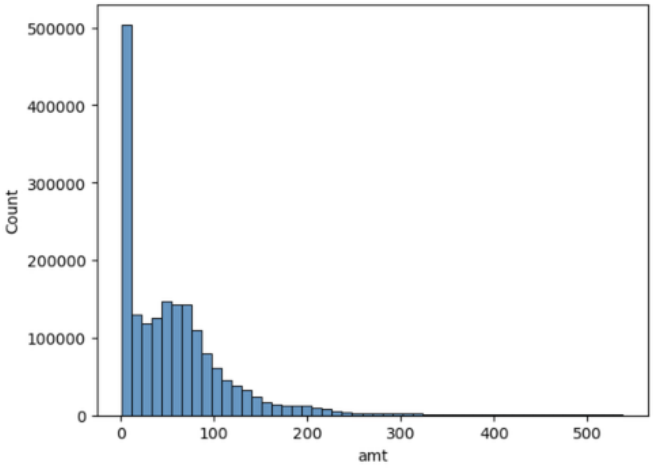


Figure 5.4.3. *Transaction amount distribution of all transactions*



In contrast, fraudulent transactions look very different. The mean amount for fraud is $530.66, over seven times higher than the overall mean. Although the maximum fraudulent amount ($1376.04) is far lower than the maximum of non-fraudulent transactions (nearly $29 000), the histogram of frauds (Figure 5.4.2) shows a pronounced right skew and two apparent peaks:

- A sharp spike near $0, which suggests that many fraudsters begin with low-value “test” transactions to verify card validity without triggering suspicion. Such small payments often correspond to low-risk categories like groceries or transport.
- A second cluster between $600 and $1200, reflecting high-value fraud attempts once the test transaction has succeeded. This bimodal shape indicates at least two distinct strategies: testing versus large-scale exploitation.

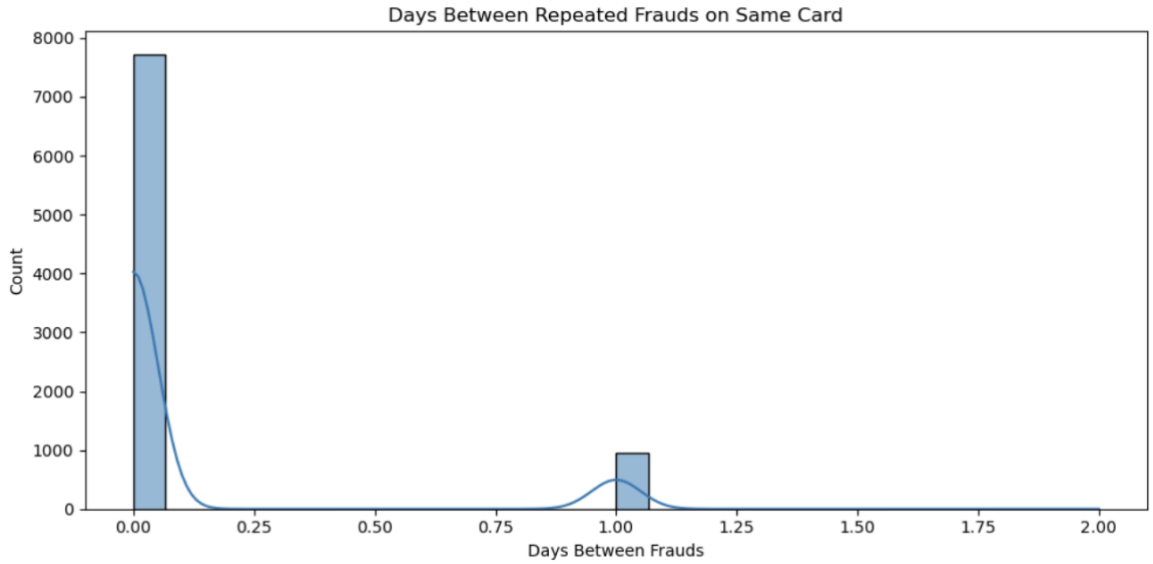
Taken together, the timing and amount analyses reveal a coherent fraud pattern. Fraudsters often start with small, low-value transactions executed in rapid succession, likely to confirm that the card is active and the detection system will not immediately block it. If these tests succeed, they are quickly followed by larger purchases (within minutes or at regular intervals, such as daily) as the fraudster maximises their haul before the cardholder or bank notices. This behaviour explains why burst transactions have a much higher fraud rate and why fraudulent amounts exhibit both a spike near zero and a separate high-value cluster. Incorporating both time-based features (e.g., time_since_last, burst

flags) and amount-based features (e.g., recent average amount, ratios of consecutive amounts) can thus help models capture these sequential patterns and improve fraud detection.

5.5 Days between repeated frauds

For a subset of cards that suffered multiple fraud incidents, the spacing between successive fraudulent transactions was analysed. The resulting distribution (Figure 5.5.1) is highly concentrated near zero days, indicating that most repeated frauds occur within the same day. In other words, once a card is compromised, fraudsters tend to execute several unauthorised transactions in rapid succession rather than waiting long intervals between attempts. This finding reinforces the earlier observation of burst-activity: fraud often arrives in clusters, with multiple hits on the same card within hours.

Figure 5.5.1. *Distribution of days between repeated frauds*



A smaller secondary peak appears at approximately one day (around 1.0 on the x-axis), suggesting a recurring pattern where attackers return about a day after the initial fraud. Such behaviour may reflect fraudsters testing whether the card remains active or trying to circumvent daily monitoring thresholds. Beyond this one-day mark, the distribution tails off quickly; very few cards experience repeat fraud separated by more than a day or two.

These results have practical implications. Detection systems should monitor not only the absolute number of fraudulent transactions but also the timing between them, flagging situations where multiple unauthorised charges occur in quick succession. Additionally, incorporating rolling time-window features (e.g., counts of fraud events in the past 24 hours) can help identify and pre-empt recurring attempts that follow a predictable daily cadence.

5.6 Fraud by merchant category

To determine whether specific merchant categories are disproportionately targeted by fraudsters, the total number of transactions in each category was compared against the number of fraudulent transactions.

Figure 5.6.1. *Number of transactions per merchant category*

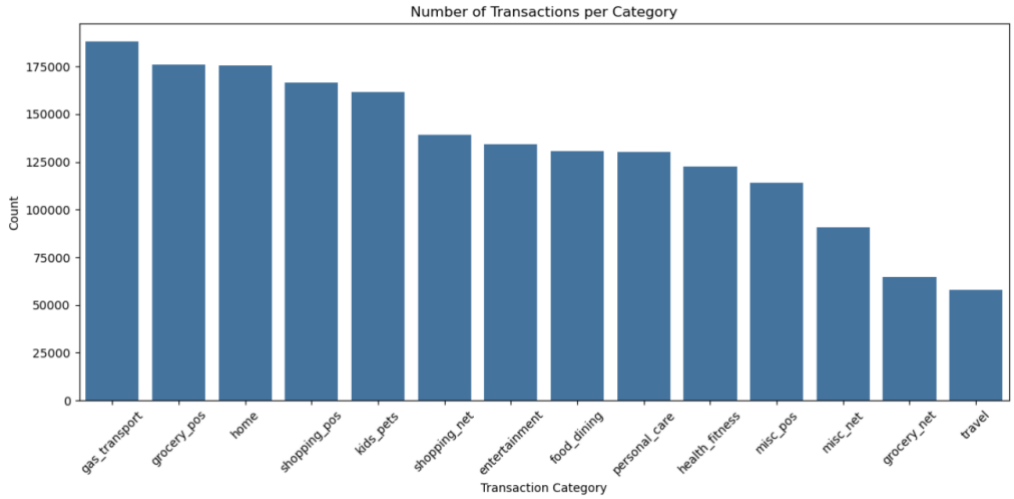


Figure 5.6.2. *Number of fraudulent transactions per merchant category*

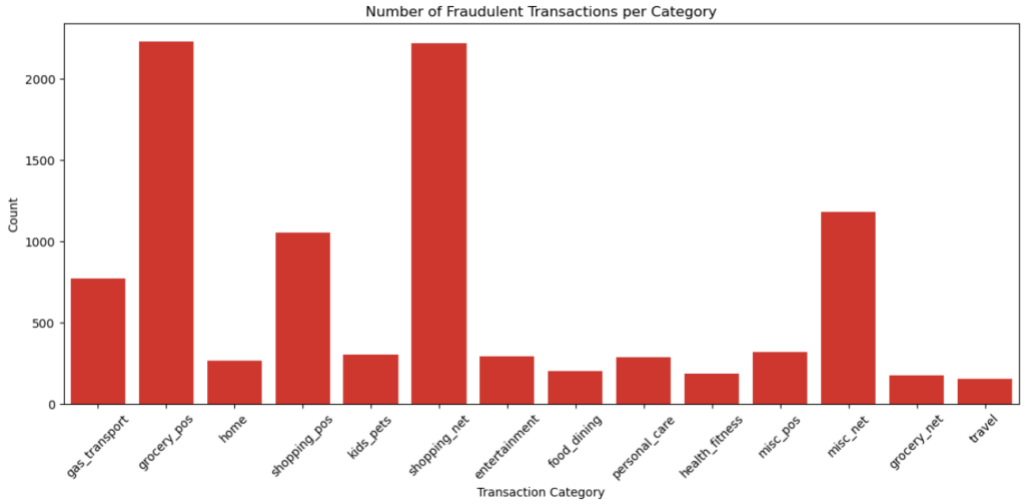


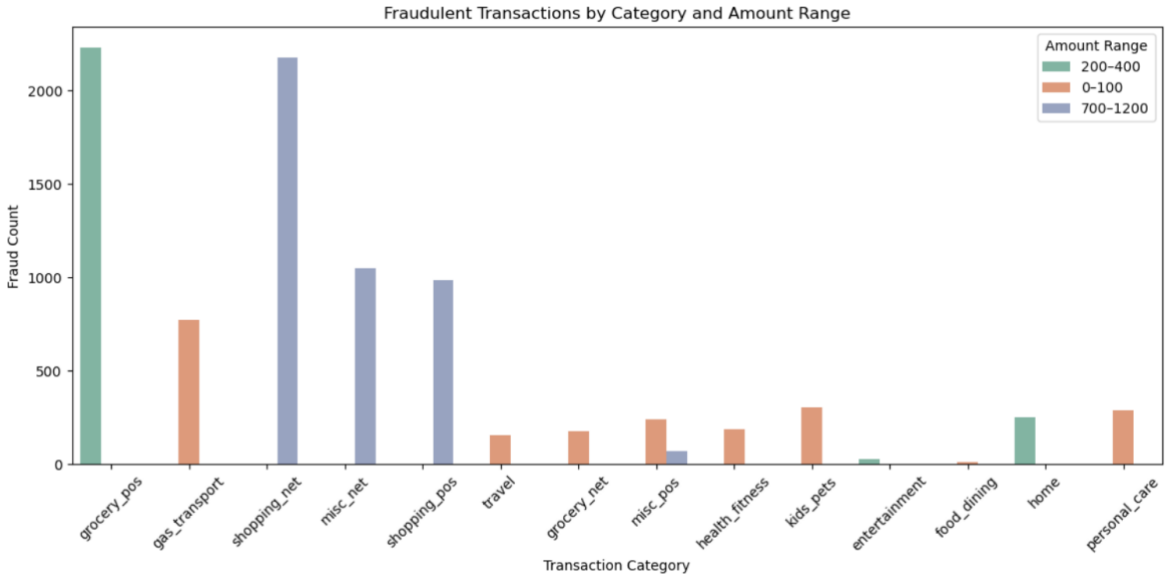
Figure 5.6.1 shows categories such as grocery_POS, gas_transport, home and shopping_POS being the most common destinations for card spending. When isolating fraudulent transactions (see figure 5.6.2), however, a different picture emerges: categories like grocery_POS and shopping_NET exhibit far more fraud incidents than their usage alone would predict. Conversely, categories with high overall volume (e.g., transport, entertainment) do not show correspondingly high fraud counts. This imbalance implies that certain merchant categories are inherently more vulnerable, perhaps due to their online nature, consumer behaviour patterns or merchant-side security practices rather than merely reflecting high transaction volumes.

Given these findings, it is prudent to incorporate merchant category as a predictive feature in fraud-detection models. Additional investigation into why some categories attract more fraud could lead to targeted mitigation strategies (for example, by examining whether these purchases are made online or in-person, or whether particular merchants have lax verification processes).

5.7 Amount spent and category-based fraud patterns

To uncover more nuanced behaviour, fraudulent transactions were further stratified by amount range and merchant category. The distribution reveals that low-value frauds (e.g., $0–$100) are heavily concentrated in categories such as gas_transport and personal_care, whereas high-value frauds (e.g., $700–$1 200) appear predominantly in shopping_NET, shopping_POS and misc_NET categories. This suggests that fraudsters may tailor their tactics to the merchant environment: small charges are often used for low-risk verification (commonly at fuel stations or small retailers), while larger purchases are attempted through online shopping platforms where high-ticket items can be acquired quickly.

Figure 5.7.1. *Fraudulent transactions by category and amount range*



Building on the earlier observation that fraudsters often perform a small “test” purchase followed by a larger transaction, the analysis identified sequences in which a transaction under $10 was followed by a purchase over $100 on the same card within five minutes. A histogram of the time gaps between these small-to-large pairs shows a relatively uniform distribution across the 0–300 second window, indicating that this burst behaviour occurs throughout the five-minute period and is not confined to a single narrow timeframe. To quantify the association between these bursts and fraudulent activity, a Chi-squared test of independence was performed. The test yielded a highly significant result ($\chi^2 = 56.87$, $p < 0.0001$), confirming that fraudulent transactions are disproportionately likely to follow a small transaction in quick succession.

These patterns strengthen the evidence for burst behaviour in fraud, where attackers perform a low-value transaction to test card validity and then rapidly exploit it with a high-value purchase. Recognising this behaviour underscores the value of combining merchant category, transaction amount and time-gap features in fraud-identification models, and highlights the need for targeted investigations into why certain merchant categories are more susceptible to this two-stage attack.

6. Modelling

Fraud costs consumers and businesses millions of dollars every year. Not only is the consumer impacted, but in card-not-present environments, merchants lose an estimated 3.68$ for every dollar lost to fraud. These costs include chargebacks, reputational damage, labour costs related to dealing with fraudulent transactions and more. As such, the business objectives of the modelling phase are to develop a solution to reduce the cost of fraudulent transactions to individuals and businesses. To achieve this, a machine learning model will be trained to detect fraudulent transactions in real time, preventing fraud before it can occur. It is also important to note that in reducing the cost of fraud, we must also take into account the cost of blocking legitimate transactions, as such our model will ideally have accurate recall and precision.

The final model developed is an LSTM network. This network processes a window of raw transaction data (up to 32 transactions for an individual) and, in real-time, can correctly identify fraudulent transactions with 97.5% accuracy.

6.1 Date preparation:

The majority of the data cleaning occurred in the analytical phase of the project, and as the data was synthetically generated, not much cleaning was required. The data was checked for null values and outliers, data typing was fixed, and transactions with the location 'DE' were dropped, as this location had a 100% fraudulent transaction rate compared to the next highest location, which was <0.05.

Feature engineering:

Based on the analysis, there were clear temporal relationships between fraudulent transactions. One of the main findings was that fraudulent transactions often occurred in bursts where one smaller test transaction was quickly followed by a larger fraudulent transaction. Furthermore, most fraudulent transactions occurred late at night. To capture these patterns, the features 'time_since_last', 'last_amt' and 'trans_hour' were created. 'time_since_last' measures the amount of time in seconds between the current transaction and the last transaction for a given card number. The value '-1' was used if there were no previous transactions. 'last_amt' records the monetary amount of the last transaction for a given cardholder, and if there are no previous transactions, the amount is set to '0'. The inclusion of both these features was intended to capture temporal relationships, specifically the presence of 'burst' transactions. Finally, 'trans_hour' was simply extracted from the date of the transaction and represents the hour of the day the transaction occurred at.

Data preprocessing

For all regression and deep learning models, the numerical data was preprocessed in a pipeline using a standard scaler. The categorical features were one-hot encoded in a pipeline, and the first feature was dropped to avoid the dummy variable trap. The dependent variable 'is_fraud' is a binary variable with 0 being a legitimate transaction and 1 being a fraudulent transaction.

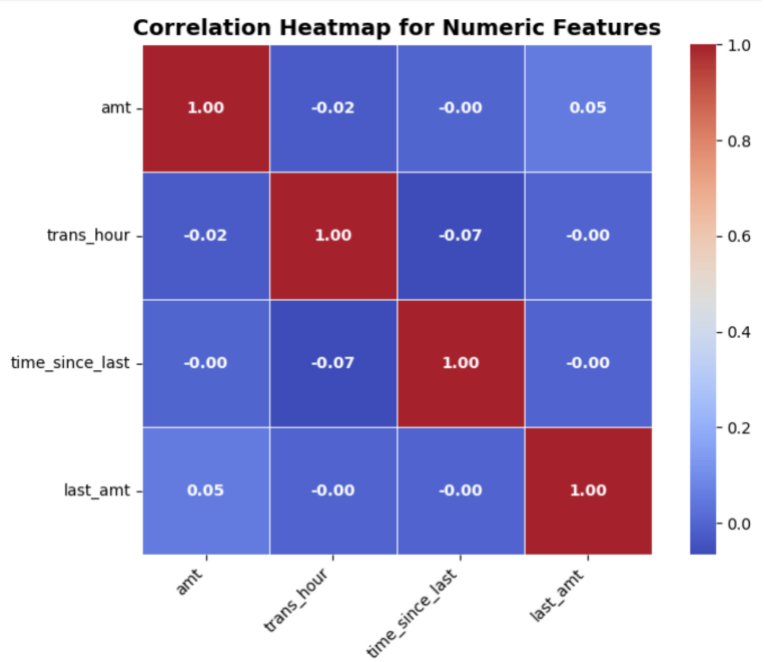
Feature selection

Due to the size of the data set (1.8 million rows) and the high cardinality of many categorical features, such as state and job, the feature selection process needed to balance capturing relevant data and reducing the size of the dataset to fit within computational constraints.

Initial feature selection was done using logistic and LASSO regression models. These models were used due to their simplicity, making them quick to train, their interpretability in identifying relevant features and for their ability to select features through regularisation. Based on findings from initial analysis, the following features were selected as a base: 'category', 'amt' and 'tran_hour'. To ensure the engineered features 'last_amt' and 'time_since_last' were effective, a logistic regression model

including these features and the base features was compared against each other. The inclusion of 'time_since_last' and 'last_amt' improved precision by 162% and recall by 14% and as such w ere included in the final models.
To test the importance of the remaining features, a LASSO regression model was trained using the following features: merchant, category, amt, gender, state, lat, long, city_pop, job, merch_lat, merch_long, trans_hour, time_since_lat, last_amt and age. The regularisation term can select and deselect features, and by analysing the features selected, we can gain some insight into what features hold meaningful data for fraud identification. The selected features included the ones present in our logistic regression model: 'category', 'amt', 'trans_hour', 'last_amt', and 'time_since_last'. Also included were 'merchant' and 'state' variables. This indicates that these features may be relevant for modelling; however, the decision to not include them was made due to the high cardinality of these features 'merchant' having 693 categories and 'state' having 50. By including features with so many categories, the training times of the model would have exponentially increased, making more complicated models, such as a neural network, much more difficult to train. Future modelling could use techniques such as PCA to reduce the cardinality of these features, but our final model also produced strong results; this was not required.

From the feature selection process, the following variables were identified as relevant and will be used in our final models: 'amt', 'trans_hour', 'time_since_last', 'last_amt' and 'category'. Figure 6.1.1: A correlation heatmap confirms no multicollinearity within our features.
Figure 6.1.1. *Correlation heatmap for numeric features*



6.2 Models

Baseline Regression models

All models used the following features: [amt, trans_hour, time_since_last, last_amt, category] and predicted the binary dependent is_fraud. The features were all pre-processed using a pipeline where numerical features were scaled and categorical features were one-hot encoded.

The baseline model, a logistic regression, was trained on an 80/20 train-test split and had moderate success in identifying fraudulent transactions, as evidenced by a recall of 0.86; however, it struggled with misclassifying legitimate transactions as fraudulent, having a very high false positive rate.

Figure 6.2.1. *Confusion matrix for baseline logistic regression*

		Predicted	
		0	1
Actual	0	351174	17954
	1	183	1166

Recall: 0.8643, Precision 0.061

Leading on from the baseline model, a polynomial regression was trained to capture any non-linear trends within the data. The model was trained up to the second degree and included interaction terms. This iteration improved recall by a marginal 2.9% however, it reduced precision by 42%. As the precision of the baseline logistic regression was poor, it suggested a poor fit to the data

XGBoost

One candidate model was the XGBoost model. The move to a gradient boosting model was made because this architecture is robust to class imbalance when properly tuned, and it handles both tabular and classification tasks well. The model was trained on an 80/20 train-test split (note the data was split by transaction time to avoid any data leakage). To address the class imbalance and ensure the model was effective at classifying fraud, the weights were adjusted by a ratio of neg/pos (or the number of negative transactions/ positive transactions). This increased the loss from misclassifying a fraudulent example by said ratio, shifting the model's learning signals to ensure the model was trained primarily to identify fraudulent transactions and not just focus on the majority class. To optimise the hyperparameters, a time series cross-validation grid search was used (the TimeSeriesSplit function was used for the grid search to ensure no data leakage).

Figure 6.2.2. *Confusion matrix for XGBoost model*

		Predicted	
		0	1
Actual	0	368173	955
	1	64	1285

Recall: 0.95, Precision: 0.57

At the 0.5 probability threshold, the XGBoost model was significantly more accurate than our base logistic regression model. The model saw a 9.47% increase in recall and a significant 89.47% increase in precision. This suggests that the XGBoost model is a strong fit for our data, being able to accurately identify fraudulent transactions; however, although a significant improvement to our baseline model, a precision of 0.57 is far from ideal, as almost half of the transactions identified as fraud are legitimate. In an attempt to balance recall and precision better, the decision threshold was optimised (i.e. probability at which a transaction is classified as fraudulent). To do this, the precision, recall and scores were calculated for each threshold value (ranging from 0 to 1) and plotted on a graph. The threshold value that optimised the F1 score was 0.9925, improving precision by 63% (0.93) and reducing recall by 11% (0.84).

Figure 6.2.3. *Precision, recall, and F1 vs threshold - XGBoost model optimised at F1=0.88 (t =0.99), illustrating trade-off between recall and precision.*

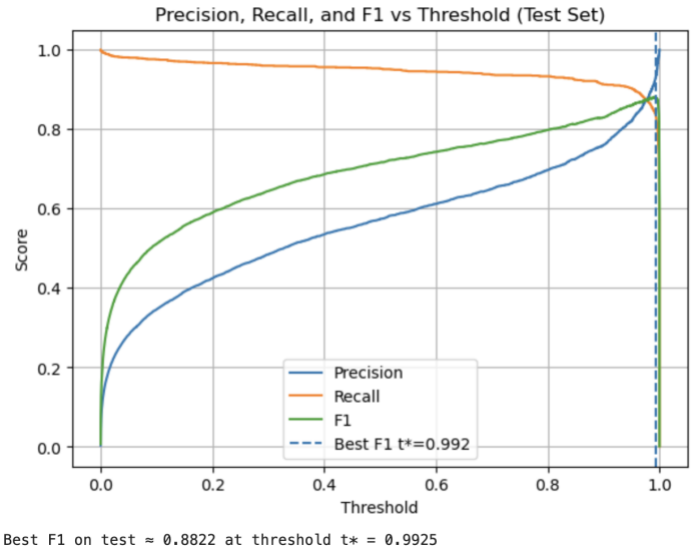


Figure 6.2.3 indicates the main weakness of the XGBoost model and shows that when optimising the model to boost recall, we see a significant decrease in the precision of our model. Due to the business context (i.e. the higher cost of fraud) requiring optimisation to reduce the overall cost of fraud (instead of maximising statistical values), this is a significant disadvantage for the model and when compared to the finalised LSTM model has worse precision and recall.

Despite the pitfalls, the XGBoost model saw significant improvements when compared to our base model and overall can classify fraudulent transactions with high accuracy, and when tuned for the optimal F1 score, it had high precision and recall. However, as will be discussed was outperformed by the LSTM network. However, an XGBoost does have some advantages over the LSTM model. Primarily, in ease of use. This model is faster and easier to train on a larger dataset, and unlike an LSTM model, which requires a window of transactions for the best results can accurately classify based on a single transaction. As such, in environments with little previous user data, this model may be more suitable and could outperform the final LSTM model.

Neural Network:

Deep learning models are highly capable of capturing complex nonlinear patterns. Furthermore, they are especially effective when relationships between features and the target (fraud vs. legitimate transactions) are subtle or involve interactions across time or multiple variables. They can also automatically learn feature representations from the data rather than relying entirely on manual feature engineering.

The first deep learning model implemented was a neural network. Due to the size of the dataset and computational constraints, training and optimising the hyperparameters on the whole dataset would have been impractical. As such, a sampling method was implemented. This sampling method aimed to reduce the size of the dataset for training efficiency as well as to reduce the class imbalance in hopes of improving recall. The first sampling method used was called down-sampling. To implement this method, transactions were binned by time frame (i.e. 1 month). All fraudulent samples were kept; however, only a number of non-fradulent transactions from this time frame were included. The number of legitimate transactions was determined by the keep_ratio parameter, and for instance, it kept 4 legitimate transactions for every fraudulent transaction. The first model was trained using 2 hidden layers with a relu activation and a 70/15/15 train, validation test split. At the 0.5 threshold, this first model did moderately well with a recall of 0.82 and a precision of 0.86, giving a good balance

between accurate fraud detection while mitigating the number of false positives. However, due to the high cost of fraud in regard to the business context, the next iteration increased the weight of fraud by a factor of: $W_c = \frac{Total\ number\ of\ samples}{number\ of\ classes * number\ of\ samples\ in\ class\ c}$, increasing the class weight improved recall to 0.96 but reduced precision to 0.15. This massive drop-off in precision was a large issue, as only 15% of cases classified as fraud were actually fraudulent, and as such, this model was not explored further. Also experimented with was oversampling the minority class, where additional duplicate samples of fraudulent transactions were created to balance the dataset. However, this model saw a decrease in both precision and recall when compared to under-sampling. Finally, a cross-validation grid search was implemented to optimise hyperparameters. Overall, the neural network saw moderate success; however, none of the networks trained were overall better than the Xgboost and, when optimising decision thresholds, saw a sharp decrease in precision for marginal improvements to recall. As such, this model was dropped.

Long-Short-term memory Neural network

An LSTM (Long Short-Term Memory) model was used because it is designed to capture temporal dependencies in sequential data. In fraud detection, transaction patterns over time, such as spending frequency, time gaps, or sudden changes in behaviour, are often more indicative of fraud than any single transaction alone. LSTMs can retain and update information across previous time steps, allowing the model to learn contextual patterns and evolving behaviours of individual users.

The LSTM model was the final model selected and saw strong results in precision and recall. The LSTM model was trained using the same features are the previous model's 'category', 'amt', 'trans_hour', 'time_since_last', 'last_amt', with the dependent remaining 'is_fraud'. The dataset was pre-processed in a pipeline using a standard scaler, and categorical features were one-hot encoded. To train the model, the data set was split into a 70/15/15/ test, validation, and training set (this split was done in time order to avoid any data leakage). The LSTM model was trained on transaction windows. Each window consists of 32 transactions from a given cardholder, sorted by transaction date and time, with the last transaction being labelled as fraud or legitimate. Windows with fewer than 32 transactions are left filled with 0's. Although making predictions worse than having the full 32 transactions, this allows the model to still make predictions based on a single transaction alone. The model was trained with a stride of 2, meaning that for each new training window, the sequence advanced by two transactions rather than one. (Note that reducing the stride to 1 and increasing the lookback or number of transactions in a window may improve model accuracy; however, due to computational constraints, these settings provide a compromise between data quality and training time.)

The LSTM model had the following structure: two hidden LSTM layers of 128 and 64 neurons, a dense hidden layer with a relu activation and 64 neurons and a final sigmoid output layer to output a probability. Finally, the threshold values were tuned against the validation set. This model had the strongest performance of all the models, and although it is difficult to interpret is most likely due to its ability to learn the behaviour of users based on their previous transactions and leverage that to detect anomalies. When maximising the threshold for the f1 score, the model had a recall of 0.95 and a precision of 0.98. Furthermore, when changing the threshold for optimising recall, we saw only a small reduction in precision, unlike the previous model, which had a sharp trade-off when optimising between the two. This is be further discussed in the results section. Also of note was that under-sampling was tested to improve the class imbalance. However, the results seen were worse than our original model, and as such, this sampling method was not explored further.

7. Results

Across all models, the LSTM delivered the strongest results in terms of both precision and recall. When the decision threshold was selected to maximise the F1 score (t=0.28), the LSTM outperformed the next best model, the XGBoost (also being optimised for F1), by 10.23% having a better recall and precision. The high accuracy of the LSTM model indicates strong discrimination under class imbalance, being able to accurately classify both legitimate and fraudulent transactions. When optimising the model from recall, there was also a smaller reduction in precision when compared to every other model. As such the LSTM was the final model selected.

7.1 Results at optimal f1

At the F1-optimised threshold (t = 0.28), the LSTM model achieved exceptional performance, correctly identifying 95% of all fraudulent transactions with a precision of 0.98 and an overall F1 score of 0.97. These results demonstrate that the model effectively captures the temporal dependencies in transaction behaviour, enabling accurate distinction between legitimate and fraudulent activity.

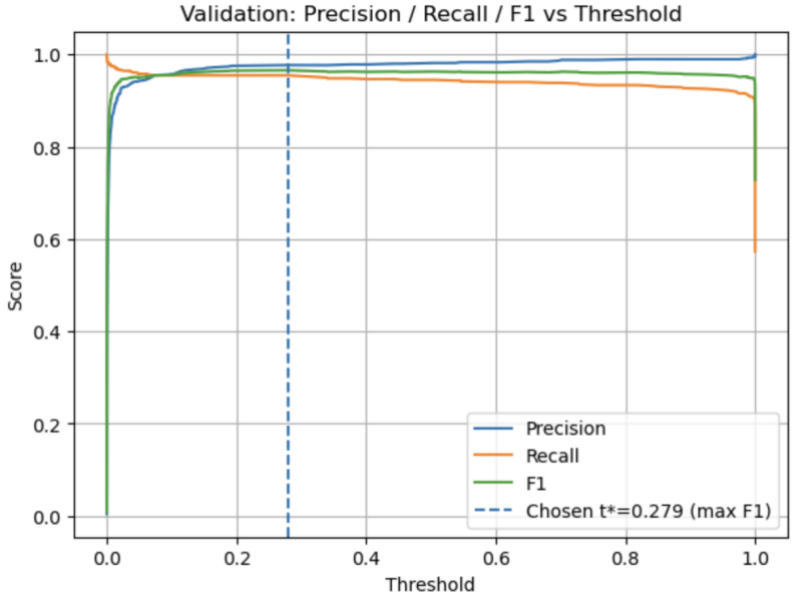
Figure 7.1.1 A confusion matrix highlights the model's strong classification ability, showing a very low number of false positives (14) and false negatives (29) across more than 136,000 legitimate cases. This indicates that the LSTM is not only highly sensitive to fraudulent patterns but also maintains reliability in recognising normal behaviour.

Figure 7.1.1. *Confusion matrix at F1-optimised threshold (t = 0.28) – LSTM Model.*

		Predicted	
		0	1
Actual	0	136082	14
	1	29	612

F1 Score: 0.97, Precision: 0.98, Recall: 0.95

Figure 7.1.2. *Validation precision, recall, and F1 vs threshold — LSTM (max F1 at t ≈ 0.279).*

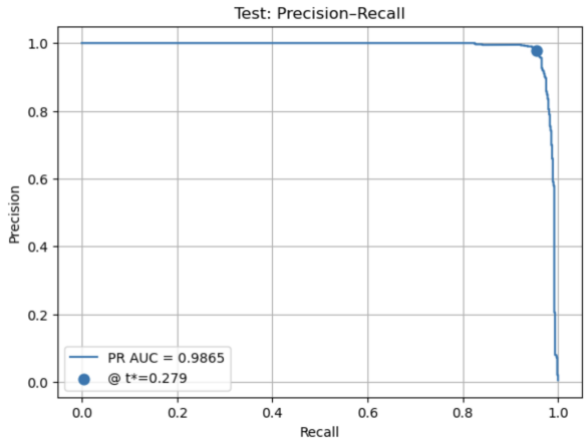



```
[Validation] t* (max-F1) = 0.2792 | F1=0.9655 | Precision=0.9767 | Recall=0.9545
```

The precision-recall-F1 curve (Figure 7.1.2) further illustrates this balance. The LSTM maintains high precision as recall increases, showing only a minor trade-off when optimising for recall compared to XGBoost, which exhibited a much steeper decline in precision. This stability confirms the LSTM's

advantage in handling imbalanced, time-dependent data, making it the most effective model for this fraud-detection task.

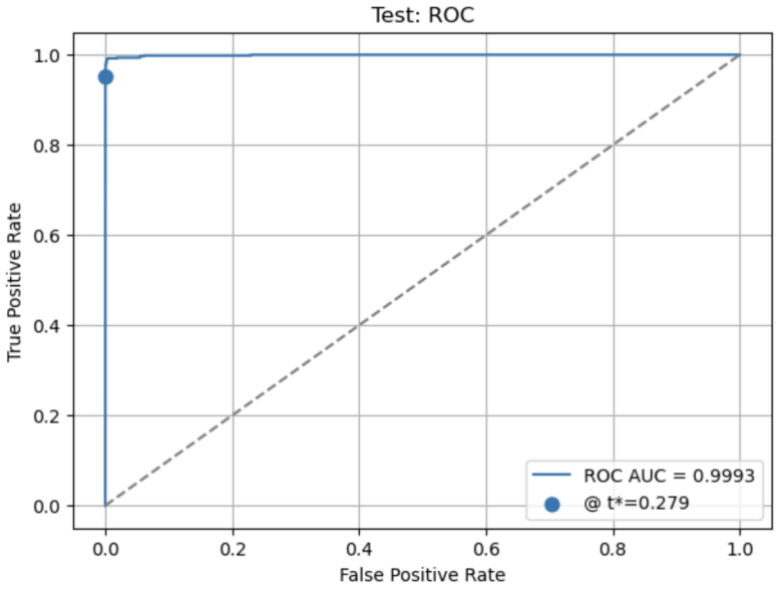
Figure 7.1.3. *Precision–Recall curve (test) - LSTM, PR-AUC = 0.9865 at t ≈ 0.279.*



The Precision–Recall curve provides a more informative evaluation of model performance under severe class imbalance. The LSTM model achieved an exceptionally high PR-AUC of 0.9865, indicating it sustains high precision while maintaining strong recall across thresholds. This demonstrates that the model rarely misclassifies legitimate transactions as fraud, even when sensitivity to rare fraud events is increased.

The slight dip near the highest recall values suggests that when the threshold is lowered to capture almost every fraudulent case, the model begins to introduce a few additional false positives, a reasonable trade-off in a fraud-prevention context. Overall, the PR-AUC confirms that the LSTM achieves an excellent balance between fraud detection coverage and false-alert control.

Figure 7.1.4. *ROC curve (test) - LSTM, ROC-AUC = 0.9993 at t ≈ 0.279*



The ROC curve further supports the LSTM model's discriminative ability, with a ROC-AUC of 0.9993, signifying near-perfect separation between fraudulent and legitimate transactions. The curve hugs the top-left corner, reflecting high true-positive rates with minimal false positives.

While this metric reinforces the model's robustness, it should be interpreted cautiously in this context. Because the dataset is highly imbalanced, even a small number of false positives can appear negligible in the ROC space. Thus, the ROC-AUC may overstate real-world performance, making the PR-AUC a more reliable indicator. Nonetheless, the ROC results confirm that the model effectively

distinguishes fraud from normal activity, validating the strength of the temporal patterns captured by the LSTM

7. 2 LSTM overfitting tests

***Figure** 7.2.1. LSTM learning curves showing stable training and validation loss with minimal divergence, confirming no overfitting*

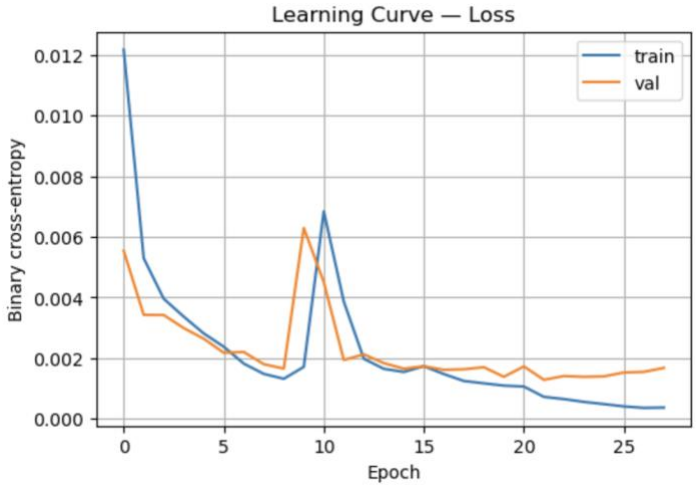
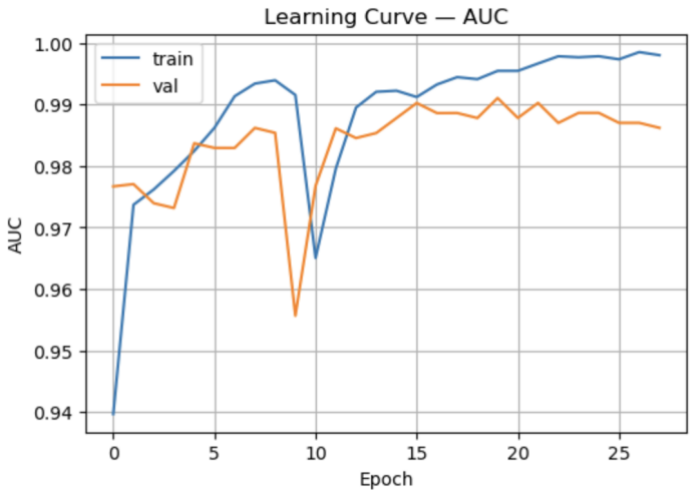


***Figure** 7.2.2. LSTM AUC learning curves validation AUC closely follows training (~0.98–0.99), indicating strong generalisation.*



The loss curves indicate that both training and validation losses decreased steadily over time, converging to low values (~0.001–0.002). The validation loss remained closely aligned with the training loss, indicating that the model generalised well to unseen data without significant variance or divergence, a strong sign of stability.

Similarly, the AUC learning curves demonstrated consistently high performance, with the validation AUC stabilising around 0.98–0.99, closely following the training curve. No major divergence between curves was observed, suggesting that the model did not simply memorise training data but learned patterns that generalise effectively.

Performance metrics further supported the absence of overfitting:

- Validation: PR-AUC = 0.9835, ROC-AUC = 0.9993, LogLoss = 0.0014
- Test: PR-AUC = 0.9869, ROC-AUC = 0.9995, LogLoss = 0.0013

The near-identical validation and test results indicate that the model maintains its predictive power on unseen data, confirming robust generalisation.

Across all diagnostics learning curves, loss values, and validation/test performance, the LSTM model demonstrates no evidence of overfitting. It achieves a strong balance between training accuracy and generalisation, maintaining high AUC and low loss across datasets.

7.3 Cost-based tuning

While traditional statistical metrics such as precision, recall, and AUC provide valuable signals of model performance, they do not necessarily reflect financial impact. In the realm of digital payments, especially for online (card-not-present) transactions, the business consequences of fraud and incorrect declines are asymmetric and substantial. For example, as reported by J.P. Morgan Chase & Co., an airline booking merchant saw false positives (legitimate orders declined) cause revenue loss, chargebacks, and operational inefficiencies, in one case leading to a decline rate of 19% per month and over US $400 K in chargebacks. According to the same source, while actual fraud loss may constitute around 7% of total fraud-related cost, false positives account for approximately 19% (*False positives & fraud prevention tools: J.P. Morgan* 2023).

These findings highlight that the cost of false negatives (missed fraud) and false positives (legitimate transactions incorrectly flagged) goes well beyond simple error counts. The model threshold must therefore be tuned not merely to maximise statistical metrics, but to minimise business cost, by weighing the real-world cost implications of each type of error.

Thus, the expected cost function is defined as:

$$Expected\ Cost(t) = FP(t) \times C_{FP} + FN(t) \times C_{FN}$$

Where:

- $FP(t)$and $FN(t)$are the number of false positives and false negatives at the threshold t.
- $C_{FP}$and C_{FN}represent the respective business costs of each error type.
  By varying t, the optimal threshold (t^*) can be found that minimises this expected cost.

Estimating False Negative Cost (C_{FN})

A false negative represents a fraudulent transaction that goes undetected. Its cost extends beyond the transaction amount, encompassing reimbursement, investigation, and reputational damage. (*Every dollar lost to fraud in Australia costs firms AUD$3.68 according to LexisNexis true cost of fraud study* 2024)

Drawing on LexisNexis Risk Solutions, the external cost multiplier for card-not-present fraud was estimated at 3.68× per dollar lost, meaning that for every $1 of fraud, businesses incur an additional $2.68 in indirect losses. (*Every dollar lost to fraud in Australia costs firms AUD$3.68 according to LexisNexis true cost of fraud study* 2024)

The mean fraudulent transaction amount in the dataset was $530.66, so the estimated cost per false negative was calculated as:

$$C_{FN} = \text{mean\_fraud\_amount} \times \text{secondary multiplier}$$

Sensitivity analysis was performed using a multiplier range from 2.0 to 4.0 to account for uncertainty, giving approximate C_{FN}values from $1,061 to $2,122 per missed fraud.

Estimating False Positive Cost (C_{FP})

A false positive (legitimate transaction incorrectly flagged as fraud) incurs costs to merchants through lost sales, support overhead, and customer churn. Research from Riskified estimates that up to 40% of customers churn after an incorrectly declined transaction, leading to long-term loss of customer lifetime value (CLV) (Hallett, 2025).

The cost per false positive was estimated as:

$$C_{FP} = \text{median\_sales\_amt} + \text{service\_cost} + (\text{CLV} \times \text{retention rate})$$

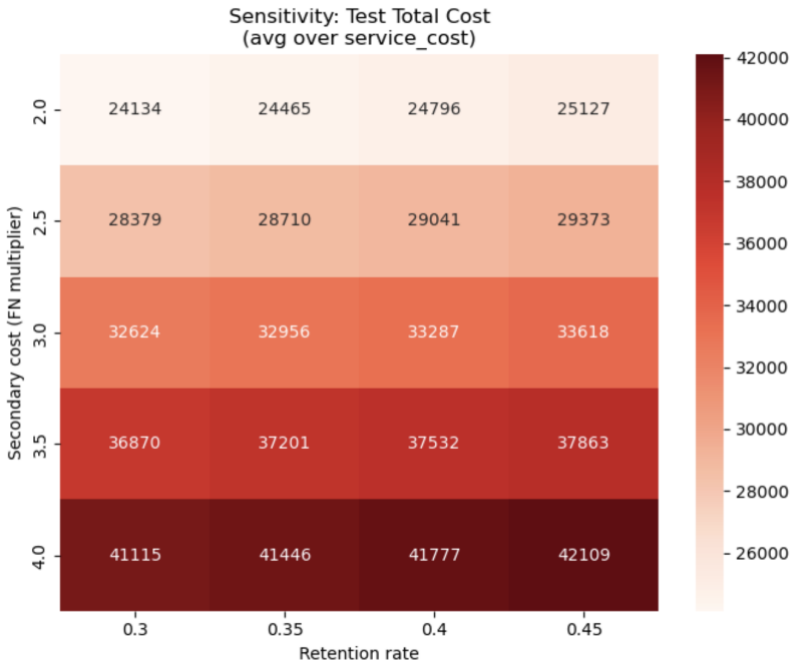
Sensitivity analysis was conducted on the following feature ranges:

- Median sales amount = $47.45 (immediate lost revenue)
- Service cost = 5–10 (representing customer support overhead)
- Average yearly customer spends (CLV proxy) = $70.47
- Retention rate range = 30–45%

Sensitivity analysis

The investigation included sensitivity analysis across ranges of secondary cost multiplier (2.0–4.0×) and retention rate (30–45%).

Figure 7.3.1. *Sensitivity heatmap of test total cost - showing sharp cost increases as false-negative cost multipliers rise.*



A heat-map of test total cost showed that as C_{FN} increases, overall cost rises sharply, underscoring the high impact of undetected fraud.

Figure 7.3.2. *Test total cost vs false-negative multiplier*

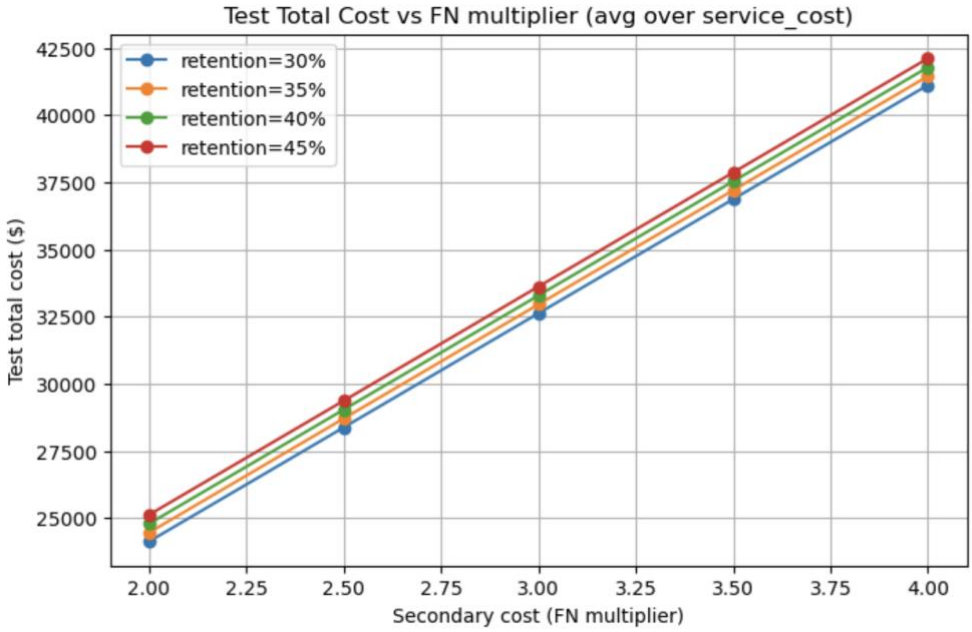


Figure 7.3.2 illustrates how the total business cost varies with changes in the false negative (FN) multiplier and customer retention rate. Each line represents a different retention rate (30%–45%), while the x-axis shows the secondary cost multiplier applied to undetected fraud cases.

The results show a strong linear relationship between the false negative multiplier and total business cost, indicating that as the indirect cost of fraud (e.g., reimbursement, chargebacks, and reputational impact) increases, total costs rise proportionally. The slope of each line is steeper than the gap between retention rates, reinforcing that missed fraud events have a far greater financial impact than marginal variations in customer retention assumptions.

Additionally, higher retention rates (red line) slightly increase overall costs, reflecting that as more customers remain active, the cumulative cost of fraud affecting those customers grows. Nonetheless, across all retention scenarios, the optimal threshold (t*) = 0.011 consistently minimised total cost, confirming that the model is robust and stable under varying business conditions.

Threshold Optimisation for Cost Minimisation

Using the fraud-probability outputs from the LSTM model, we evaluated thresholds from 0.001 to 0.999. For each threshold and each combination of C_{FN} and C_{FP}, we computed the expected cost on the validation set and selected the threshold t^* that minimised cost. This optimal threshold was then applied to the test set to evaluate real-world cost impact.

- Optimal threshold found: $t^* \approx 0.011$
- Test set results at t^*:
  - Total cost: $23,898
  - Average cost per transaction: $0.17
  - False negatives: 16
  - False positives: 94

These figures demonstrate the efficacy of aligning threshold selection to business cost rather than purely statistical criteria.

8. Recommendations and Limitations

8.1 Recommendations

Based on the exploratory data analysis and modelling outcomes, several key behavioural and transactional patterns were identified that inform targeted fraud prevention strategies.

Implement Real-Time Temporal and Velocity Monitoring

The analysis revealed that fraudulent activity often occurs in rapid succession, with fraud being four times more likely in burst sequences of transactions occurring within five minutes, compared to isolated events. Furthermore, repeated frauds typically occur within the same day, suggesting that fraudsters exploit short time windows to extract maximum value before detection. The LSTM model's success in capturing these sequential dependencies (F1 = 0.97, PR-AUC = 0.9865) reinforces the value of incorporating temporal features.

To mitigate this, financial institutions should integrate real-time velocity and time-window monitoring that automatically flags unusual transaction bursts or recurring attempts within defined intervals (e.g., multiple transactions within five minutes or repeat activity within 24 hours). This would allow early intervention before subsequent high-value frauds occur, particularly for compromised accounts exhibiting sudden spikes in frequency.

Apply Tiered Transaction Controls Based on Merchant Category and Amount

EDA findings showed that fraud is disproportionately concentrated in specific merchant categories, notably *shopping_NET* and *grocery_POS*, which exhibit higher fraud ratios than their overall transaction share. Moreover, a bimodal spending pattern was observed: small "test" purchases under $10 were frequently followed by high-value transactions ($600–$1 200) within a short window, with this relationship statistically confirmed ($\chi^2 = 56.87$, $p < 0.0001$).

In response, financial institutions should adopt tiered transaction-control mechanisms that dynamically adjust verification requirements based on merchant category and transaction value. For example:

- Introduce enhanced authentication for high-risk categories such as online shopping or grocery network merchants.
- Implement temporary spending caps or "hold-and-verify" checks when a small purchase is immediately followed by a large one on the same account.
  Such controls directly target the behavioural signatures observed in the data while maintaining a smooth customer experience for legitimate users.

Continuously Optimise Model Thresholds Using Business-Cost Metrics

While traditional metrics such as precision and recall are essential, the cost-based tuning phase demonstrated that aligning model thresholds with financial impact leads to better business outcomes. Optimising the decision threshold (t = 0.011) reduced the average cost per transaction to $0.17, striking a balance between false positives (legitimate declines) and false negatives (missed fraud). Sensitivity analysis confirmed that false negatives drive exponentially higher total costs due to reimbursement, chargebacks and reputational loss, while excessive false positives risk customer churn.

It is therefore recommended that institutions continuously recalibrate fraud-detection thresholds using up-to-date internal cost data rather than static statistical cut-offs. This ensures that model performance remains aligned with evolving fraud dynamics and business priorities, supporting both profitability and customer trust.

8.2 Future Modelling Recommendations

Incorporating High-Cardinality Categorical Variables via Dimensionality Reduction

During feature selection, variables such as *merchant*, *state*, and *job* were excluded due to their extremely high cardinality, which would have significantly increased model complexity and training time. Future work could leverage Principal Component Analysis (PCA) or embedding layers to compress these categorical variables into lower-dimensional representations without losing key variance information. For instance, a reduced-dimensional *job* representation could allow the model to capture occupation-based spending behaviour while maintaining computational efficiency. This approach may also improve the model's ability to generalise across users with similar demographic or occupational profiles.

Hybrid and Ensemble Architectures

Although the LSTM demonstrated excellent temporal learning, ensemble or hybrid approaches may provide incremental gains by combining strengths of different algorithms. For example, integrating an XGBoost model trained on static tabular features with the LSTM network's sequential outputs could yield a hybrid model capable of capturing both temporal dependencies and non-sequential interactions. This fusion could also improve performance for users with limited transaction histories, where temporal data is sparse.

Adaptive Thresholding and Online Learning

Fraud patterns evolve rapidly over time. Implementing adaptive thresholding, where the fraud probability threshold dynamically adjusts based on recent transaction performance metrics, could maintain optimal detection sensitivity as data distributions shift. Similarly, introducing online or incremental learning would allow the LSTM model to continuously retrain on new transactions, reducing degradation in accuracy caused by emerging fraud tactics.

Integration of External and Network-Based Data

Finally, expanding the dataset to include external behavioural or network-level indicators could strengthen model robustness. Examples include merchant reputation scores, IP address risk ratings, or graph-based user–merchant relationships to detect suspicious transaction clusters. Graph neural networks (GNNs) could be explored to model these relational structures, potentially uncovering coordinated fraud rings or shared behavioural signatures across compromised accounts.

8.3 Limitations

While the LSTM fraud detection model achieved strong performance and business alignment, several limitations remain that should be considered when interpreting results and planning future work.

Estimating the True Cost of Fraud
One of the most significant challenges lies in accurately estimating the cost of false positives and false negatives. The analysis relies on external benchmarks (e.g., LexisNexis and Riskified studies) that provide average multipliers and customer retention rates. However, these figures vary widely across industries, transaction types, and customer demographics. As such, the model's cost-optimised threshold may not directly translate to all real-world contexts. A more precise estimate would require institution-specific cost data and continuous recalibration.

Synthetic Data and Generalisability
The dataset used was synthetically generated and may not perfectly reflect the complexity of real transaction behaviour. Fraudulent activity in practice often evolves dynamically, influenced by seasonality, emerging fraud tactics, and user-specific patterns. This means that while the LSTM model performs well on this dataset, its performance could differ when applied to real banking data without additional retraining or adaptive mechanisms.

Cold-Start and New User Limitation
Because the LSTM model learns from windows of previous transactions (32-step sequences), it performs best when sufficient historical data exists for each user. For new customers or recently issued cards, where transaction history is sparse, the model may struggle to detect anomalies effectively, a classic cold-start limitation. Hybrid approaches that combine static user profiles or heuristic rules could help mitigate this weakness in production.

Interpretability and Transparency
Deep learning models like LSTMs are inherently less interpretable than tree-based methods such as XGBoost. While explainability tools (e.g., SHAP or LIME) could provide insights into decision logic, they were not applied in this project due to computational and time constraints. This limits direct visibility into the reasoning behind individual fraud predictions.

Operational Considerations
Real-time fraud detection systems must handle continuous transaction streams at scale. Although deployment on AWS EC2 with FastAPI demonstrated proof-of-concept feasibility, performance may vary under high load conditions. Future work should test container orchestration, auto-scaling, and API latency under production-level traffic.

9. Model deployment

After model development and evaluation, the final LSTM fraud detection model was deployed on an Amazon Web Services (AWS) EC2 instance to demonstrate its practical implementation and real-world applicability. The goal of this deployment was to expose the trained model as an API service that can perform real-time fraud prediction, simulating how it would function in a production environment used by financial institutions. Deploying the model allowed the project to transition from a research-based exercise to an operational proof of concept, highlighting the model's potential for scalable use in live transaction systems.

The deployment process involved building a cloud-hosted API using FastAPI, a lightweight Python web framework designed for high-performance applications. FastAPI was selected because of its speed, ease of integration with machine learning models, and support for asynchronous processing, which is essential for real-time fraud detection. To ensure reproducibility and consistent performance across environments, the entire service was containerised using Docker. This approach bundled the trained model, preprocessing pipeline, and all dependencies into a single image that could be run reliably both locally and on the EC2 instance. The container was configured to expose two primary endpoints: /health, which returned model metadata and confirmed that the service was active, and /predict, which accepted transaction data in JSON format and returned a fraud probability and classification decision.

The container image was built locally and then deployed to a t3.micro EC2 instance running Amazon Linux 2023. Docker was installed on the instance, and the service was launched using the command docker run -d -p 80:8000 fastapi-demo, mapping port 8000 in the container to port 80 on the instance so that it could be accessed publicly through the instance's IP address. The trained model and preprocessing artifacts, including the. keras LSTM file and feature transformation pipeline were mounted from the instance directory into the container to ensure the API had access to the correct files at runtime. Once deployed, the /health endpoint returned confirmation that the model was successfully loaded, along with metadata including the number of timesteps, number of features, and the optimal decision threshold value.

To validate the deployment, a series of test requests was made to the public API endpoint using Python's requests library. Each request contained a sample transaction represented as a JSON payload including features such as transaction amount, time since last transaction, previous transaction amount, and category. The API correctly processed these inputs, applied the preprocessing pipeline to encode the categorical feature, and returned a JSON response containing the predicted probability of fraud, the model's binary decision, and the threshold applied.

Overall, deploying the LSTM fraud detection model as a cloud-based API demonstrated the model's operational readiness and scalability. It also provided practical experience in end-to-end machine learning engineering from model training and evaluation through to containerisation, cloud deployment, and API interaction. The service responded to requests in under one second and maintained consistent results across multiple test cases

10. Conclusion

This project successfully demonstrated how machine learning, and specifically LSTM-based architectures, can be leveraged to detect and prevent fraudulent activity in transactional data. The model effectively captured both behavioural and temporal relationships between transactions, delivering near-perfect discrimination between legitimate and fraudulent cases. By incorporating cost-based optimisation, the project went beyond traditional accuracy metrics to align predictive performance with economic impact, providing a framework that balances security and customer experience.

However, accurately quantifying business cost remains a key challenge, and performance may vary under real-world transaction dynamics. The use of synthetic data, lack of interpretability, and the LSTM's dependency on prior transaction history are notable limitations that could be addressed in future work through real-data retraining, explainability techniques, and hybrid model architectures. Overall, the results demonstrate that integrating data science, domain knowledge, and cost-aware optimisation can significantly enhance fraud detection efficiency. With further validation and refinement, this approach could provide a scalable, data-driven solution for modern financial fraud prevention.

11. References

Ali, A., Abd Razak, S., Othman, S. H., Eisa, T. A. E., Al-Dhaqm, A., Nasser, M., Elhassan, T., Elshafie, H., & Saif, A. (2022). Financial Fraud Detection Based on Machine Learning: A Systematic Literature Review. Applied Sciences, 12(19), 9637. https://doi.org/10.3390/app12199637

Australian payment fraud 2024 report (2025) *Indue*. Available at: https://www.indue.com.au/2025/02/01/australian-payment-fraud-2024-report/ (Accessed: 29 October 2025).

Every dollar lost to fraud in Australia costs firms AUD$3.68 according to LexisNexis true cost of fraud study (2024) *LexisNexis Risk Solutions*. Available at: https://risk.lexisnexis.com/global/en/about-us/press-room/press-release/20240429-tcof-australia#:~:text=SYDNEY%20%E2%80%94%20LexisNexis%C2%AE%20Risk%20Solutions,value%20lost%20in%20fraudulent%20transactions (Accessed: 29 October 2025).

False positives & fraud prevention tools: J.P. Morgan (2023) *False positives & fraud prevention tools | J.P. Morgan*. Available at: https://www.jpmorgan.com/insights/payments/analytics-and-insights/cnp-fraud-prevention-combat-chargebacks?utm_source=chatgpt.com (Accessed: 29 October 2025).

Hallett, J. (2025) *How much does a false decline cost your business?* Available at: https://www.riskified.com/blog/reduce-false-declines/ (Accessed: 29 October 2025).

Personal fraud, 2023-24 financial year (2025) *Australian Bureau of Statistics*. Available at: https://www.abs.gov.au/statistics/people/crime-and-justice/personal-fraud/latest-release#:~:text=Key%20statistics (Accessed: 29 October 2025).