The School of Mathematics

THE UNIVERSITY
of EDINBURGH

Model and Feature Manipulation to Predict Specific Anomalies

by

Aravindh Sankar Ravisankar, S2596860

Dissertation Presented for the Degree of
MSc in Statistics with Data Science

June 2024

Supervised by
Dr. Tim Cannings, Dr. Cecilia Balocchi and Johnny Lee (University of Edinburgh) and
Callum Hodgkinson, Dimitros Ntakoulas and George Deskas (Lloyd's Bank)

Executive Summary

Supervised models have shown limited effectiveness in dynamic environments [Ahmed et al.(2016)[1]], prompting a shift towards more effective unsupervised approaches for anomaly detection [Ackay et al.(2022)[2]]. Our objective was to deploy unsupervised models for risk detection. Following exploratory data analysis, we recognized the necessity for extensive feature engineering and the creation of new features. Subsequently, we implemented and compared two traditional models (Isolation Forest and One-class SVM) alongside a deep learning model (LSTM Autoencoder) employing Principal Component Analysis for feature selection. Our findings indicate that while Isolation Forest and One-class SVM perform well in terms of recall, LSTM AutoEncoder demonstrates superior discrimination overall, establishing it as a robust choice for risk detection within this dataset.

Acknowledgments

I would like to extend my heartfelt thanks to everyone who supported me during my first dissertation project. First and foremost, I would like to express my deep gratitude to my project supervisors: Dr. Tim Cannings, Dr. Cecilia Balocchi, and Johnny Lee from the School of Mathematics at the University of Edinburgh, as well as Callum Hodgkinson, Dimitros Ntakoulas, and George Deskas from Lloyd's Bank, for their invaluable guidance, insightful suggestions, and unwavering support.I would like to extend a special thanks to Lloyd's Bank for providing the problem statement and dataset, and to the School of Mathematics for facilitating this project. I am also grateful to my friends and family for their constant support and encouragement.

Thank you all for your contributions and support.

Yours sincerely,
Aravindh Sankar Ravisankar

University of Edinburgh – Own Work Declaration

This sheet must be filled in, signed and dated - your work will not be marked unless this is done.

Name: Aravindh Sankar Ravisankar
Matriculation Number: S2596860
Title of work: Model and Feature Manipulation to Predict Specific Anomalies

I confirm that all this work is my own except where indicated, and that I have:

- Clearly referenced/listed all sources as appropriate
- Referenced and put in inverted commas all quoted text (from books, web, etc)
- Given the sources of all pictures, data etc. that are not my own
- Not made any use of the report(s) or essay(s) of any other student(s) either past or present
- Not sought or used the help of any external professional academic agencies for the work
- Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)
- Complied with any other plagiarism criteria specified in the Course handbook

I understand that any false claim for this work will be penalised in accordance with the University regulations (`https://teaching.maths.ed.ac.uk/main/msc-students/msc-programmes/statistics/data-science/assessment/academic-misconduct`).

Signature: Aravindh Sankar Ravisankar

Date: 28th June 2024

Contents

List of Tables

List of Figures

1 Introduction

1.1 Background and Motivation

In the realm of data-driven decision-making, identifying and mitigating risk events in spending data has become pivotal. Numerous studies have highlighted the limitations of traditional anomaly detection techniques. Chandola et al.(2009)[5] emphasize that these methods often fail to account for the temporal and multivariate complexities inherent in financial datasets, leading to failed detection of true anomalies.Qin et al.(2011)[14] argues that supervised models for anomaly detection in network traffic are impractical due to the scarcity of labeled data and their inability to detect novel anomalies leading to limited generalization. Pimentel et al.(2014)[12] suggests that obtaining labeled data for anomaly detection is impractical due to the high cost associated with labeling large datasets. Similarly, Ahmed et al.(2016)[1] discuss the challenges of deploying conventional methods, stating that these models are often ill-suited for dynamic, real-time data.

To address these challenges, unsupervised models have emerged as a viable alternative. Leung et al.(2005)[8] laid the foundation for implementing clustering-based approaches that effectively identify outliers in multivariate datasets. Recently, Munir et al.(2019)[9] have shown the efficacy of deep learning models in capturing intricate patterns in time series data, resulting in improved accuracy in anomaly detection. Moreover, Ackay et al.(2022)[2] introduced ANOMALIB, a novel library for unsupervised anomaly detection, which has incorporated state-of-the-art algorithms and shown superior performance across various benchmarks. Recently, Najari et al.(2022)[10] have utilized robust transformer-based models for unsupervised time series anomaly detection through RESIST, demonstrating the effectiveness of advanced unsupervised techniques in this field.

1.2 Problem Statement

The **main objective** of this project is to **develop an unsupervised model** to replace the current human-based system for **identifying risk events** in spending data. Given the longitudinal nature of the data, the ultimate challenge is to enhance the model's capability to distinguish true risk events from anomalies. By deriving new features through advanced feature engineering techniques that could serve as potential risk indicators, the model aims to emulate the decision-making process of human experts. This initiative is driven by the imperative to improve the effectiveness of risk management strategies, necessitating the development of a robust framework for practical risk detection and mitigation.

1.3 Data

The dataset under study consists of longitudinal and multivariate spending data, encompassing 105,277 records across 10 columns, spanning from January 9, 2023, to April 30, 2023, with over 2,185 unique individuals present. Each individual exhibits varying frequencies of events. There are no missing values in the continuous features, which are predominant compared to the categorical feature 'department' encompassing 20 different values. The response variable 'risk_event' indicates whether each event is classified as a risk event, with values of 'True' or 'False'. Other variables of interest include 'spend' **(the amount spent by each individual)**, 'individual_id' **(the unique identifier given to every individual present)**, and 'date' **(date of transaction)**. The dataset's structure necessitates robust feature engineering **(as discussed already in the Problem Statement section)** to derive temporal features that could help us detect risk events effectively. Later in the report, the terms 'spending' and 'expenditure' are used interchangeably, as are 'total' and 'aggregate'.

2 Literature Review

A considerable amount of research has been conducted on unsupervised anomaly detection in time series data, with some studies specifically focusing on the financial domain.

Munir et al.(2019)[9]introduce a CNN-based model designed to predict the next timestamp in a series and detect anomalies based on the deviations from the actual values. Evaluated on various benchmark datasets, the results show that DeepAnT outperforms 15 state-of-the-art methods showing significant improvements over the other paradigms present, highlighting its robustness and applicability in real-world scenarios.

A more enhanced approach was given by Provotar et al.(2019)[13], where he proposed an effective method for detecting anomalies using LSTM-autoencoders. Trained on the DCASE rare sound event dataset, the 8-layer LSTM autoencoder model achieved an accuracy of 87% and correctly localized anomalies in approximately 91.7% of cases. The study emphasizes the superiority of LSTM-autoencoders over other conventional methods.

In recent times, Hilal et al.(2021)[7] have provided a comprehensive analysis of anomaly detection methods in financial fraud detection. The review advocates for a transition away from traditional supervised learning approaches towards more sophisticated semi-supervised and unsupervised models. Given the dynamic nature of fraudulent behavior, Hilal strongly advocates for the adoption of unsupervised learning techniques. In support of his statement, deep learning methodologies such as convolutional neural networks **(CNNs)**, autoencoders **(AEs)**, and generative adversarial networks **(GANs)** have displayed substantial potential to achieve high accuracy in fraud detection.

In a study delving more specifically into credit card fraud detection, Mahdi et al.(2015)[15] conducts a comprehensive evaluation of the effectiveness of unsupervised learning techniques. His research meticulously compares and contrasts the performance of one-class SVM, autoencoders, and robust Mahalanobis distance methods using real-world datasets. The findings reveal that each algorithm performed similarly well, with the one-class SVM achieving the highest precision score of 90%. This study once again re-emphasizes the potential of unsupervised methods in effectively detecting and mitigating risk events.

In alignment with the above research work, we contribute by developing and deploying an unsupervised machine learning model, with a strong emphasis on deep learning techniques. However, our approach diverges from the previous work by extending beyond mere anomaly detection to focus on modeling that predicts actual risk events.

3 Exploratory Data Analysis (EDA)

3.1 Spend wise Analysis

The histogram plot depicted in Figure 1 of the expenditure **(spend)** feature column shows a significant right skew, indicating that most expenditures are relatively low, with very few large expenditures. The highest frequency of expenditures is in the range of 0 to 25, with a peak close to zero, suggesting that small expenditures are very common. As the expenditure amount increases, the frequency drops sharply, leading to a long tail on the right side of the plot, which could indicate potential outliers. This distribution pattern highlights that while small expenditures dominate the data, there are **occasional large expenditures** that could be significant for further analysis, potentially **indicating anomalies or risk events**. A notable point is that all expenditure figures are denoted in thousands of pounds (**£1,000s**).

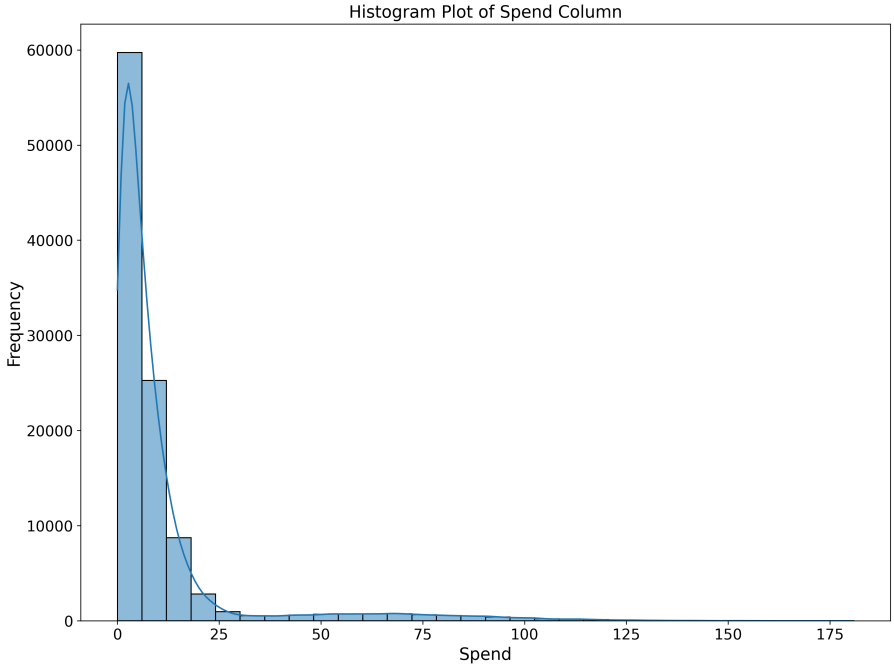

Figure 1: Histogram plot of Spend feature column

3.1.1 Analysis on the Top Performing Individuals

Given the dataset encompassing 2,185 unique individuals, a comprehensive analysis of each individual is impractical. Therefore, Figure 2 presents time series plots of the expenditure pattern for the top three contributing individuals **(213, 1985, and 2182)** based on the total number of transactions they have made. Meanwhile, Figure 3 displays time series plots for the top three individuals **(951, 847, and 1115)** based on their total**(aggregate)** spending over the entire time period. Individuals 213 and 1985 also appear in the second category **(top spenders)**, but since we have already plotted them, we thereby include the next three in line. By examining the expenditure patterns of these top contributing individuals, we can understand, to some extent, the nature of flagged risk events and their correlation with the anomalies present.

As per Figure 2 and Figure 3, both sets of time series plots indicate the presence of **point anomalies** and **contextual anomalies** in the expenditure patterns of individuals. Point anomalies appear as sudden, unexplained spikes or drops in spending, while contextual anomalies are deviations that are significant only within certain contexts, such as unexpected high expenditures during typically low-spend periods. Although **these anomalies highlight unusual expenditure behaviors, they do not always align with the risk events**, as observed from the time series plots. The risk events

represent specific points of concern that may be influenced by factors beyond just the expenditure anomalies, suggesting that risk assessment involves additional considerations beyond mere spending irregularities.

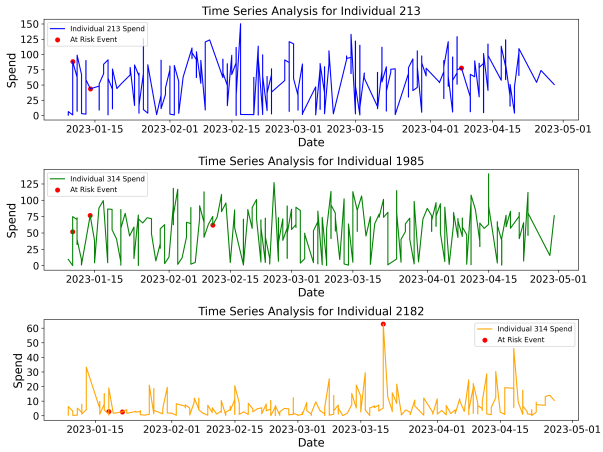

Figure 2: Top Contributors

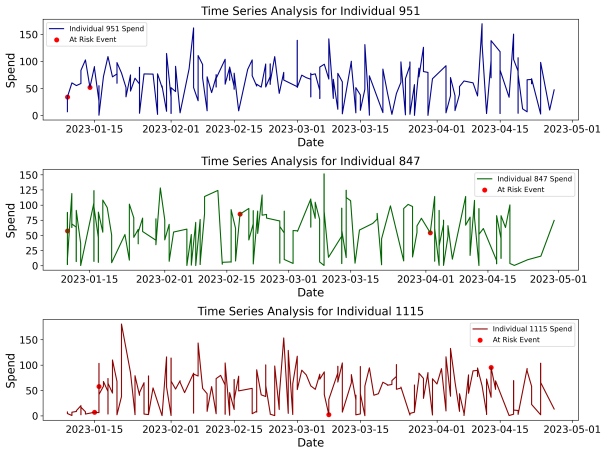

Figure 3: Top Spenders

3.1.2 Understanding the Time Series Aspects

On inferring from Figure 4 and Figure 5, both plots illustrate common insights in the expenditure data when analyzed over different periods **(31 days and 15 days respectively)** through STL[6]. Both plots reveal distinct periodic patterns, indicating recurrent spending cycles. The trend components demonstrate an initial rise in expenditure peaking around mid-February, saturating temporarily, and gradually declining after April 1st. The seasonal components highlight regular cyclic expenditure fluctuations with consistent peaks and troughs. Residual components in both plots represent noise or irregular fluctuations that are not captured by trends or seasonality. In summary, regardless of the period analyzed, there is strong evidence of seasonality and an upward trend characterized by the data, reaffirming the nature of a time series.

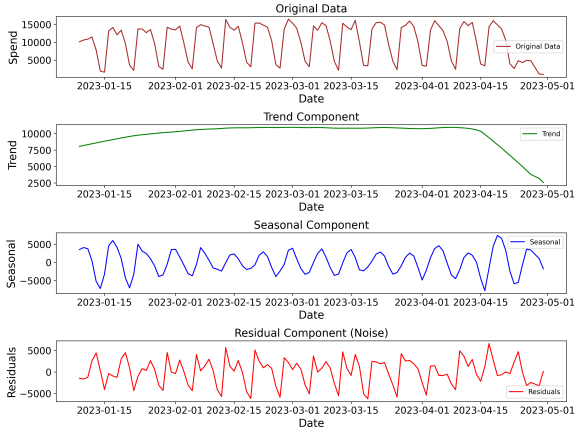

Figure 4: STL Plot on a 15 day period

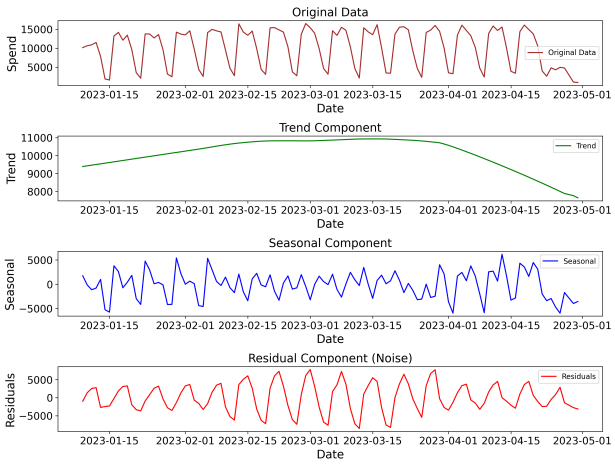

Figure 5: STL Plot on a 31 day period

3.2 Department wise analysis

The histogram plot Figure 6 below illustrates the total number of transactions across different departments. Among them, the **top three** departments in terms of transaction volume are **legal, production, and marketing**, while **product management** has the **fewest transactions**. This visualization offers insights into the varied behaviors observed across departments. Transaction counts

range from a minimum of approximately 800 to a maximum of around 10,000, highlighting the varying levels of activity and engagement within each department over the observed time period.

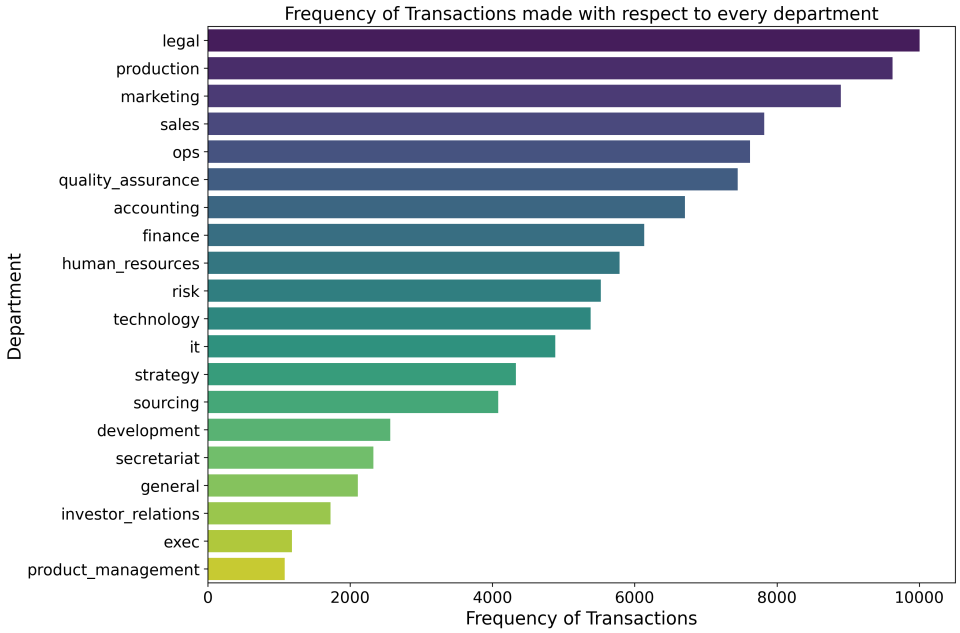

Figure 6: Department Analysis

The graph Figure 7 presents two overlapped time series plots: one depicting aggregated spend and the other showing risk ratios across various departments over the period of time. A key insight is the **lack of consistent correlation** between **risk ratio** and **aggregated spending** across departments. For instance, legal and ops exhibit the highest aggregated spending yet maintain low-risk ratios, suggesting efficient spending with controlled risk. Conversely, marketing and sales show high-risk ratios alongside high spending, whereas in contrast, departments such as development, general, and product management exhibit both low spending and low risk. It all looks so varied that **nothing conclusive** on the **magnitude of risk** could be formulated with the aggregated spend measure where every department's behavior is distinct.

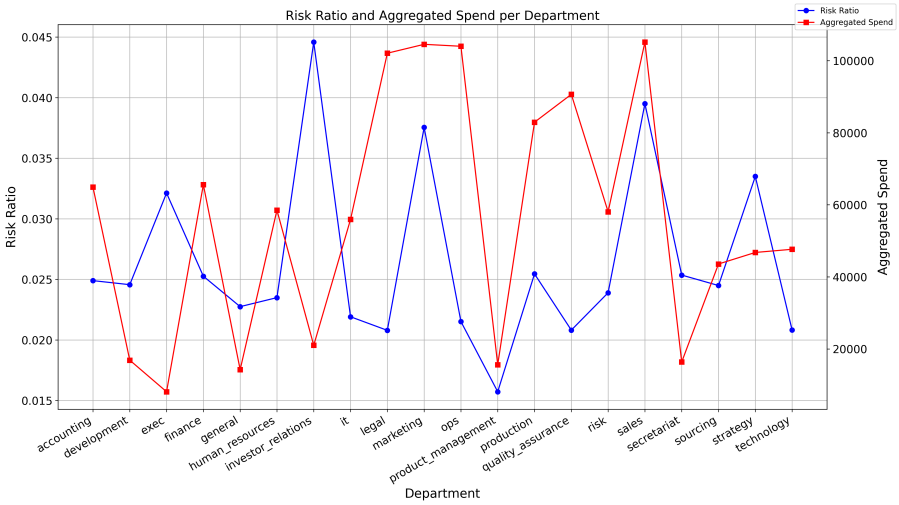

Figure 7: Risk Ratio and Aggregated Spend

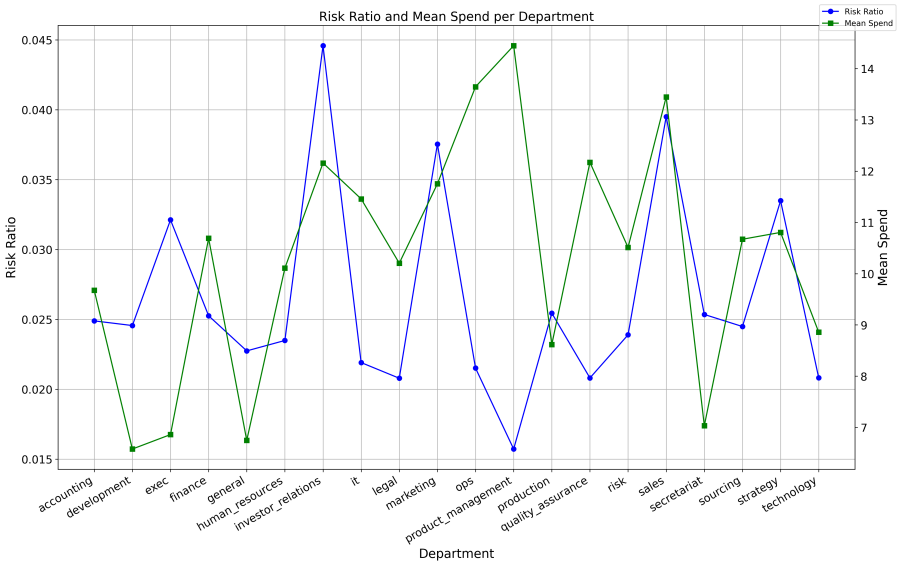

Figure 8: Risk Ratio and Mean Spend

Figure 8 illustrates the relationship between risk ratio and mean spending across departments. Departments such as investor relations, marketing, and sales show high-risk ratios alongside high mean spending, indicating greater susceptibility to risk compared to other departments. In contrast, product management and ops exhibit high mean spending but maintain low-risk ratios. Conversely, departments like development, general, and technology display low spending and low-risk behavior, suggesting minimal exposure to risks. The significant variability observed across departments**(as noted earlier from the previous plot)** highlights the need for targeted risk mitigation strategies.

3.3 Timeline analysis

3.3.1 Daily analysis

The bar plot Figure 9 illustrates the frequency of transactions across seven days of the week. Weekdays, excluding Friday, consistently show a significant number of transactions, aggregating around 21,000 per day over the analyzed period. In contrast, Friday records approximately 14,000 transactions, while both Saturday and Sunday **(weekend)** exhibit a sharp decline with around 3,000 and 2,000 transactions respectively. The lower transaction numbers on weekends can be attributed to these days falling outside the typical working week paradigm. This plot highlights **distinct behavioral patterns** between **weekends and weekdays**, suggesting potential insights for modeling, such as deriving new features based on these observed patterns

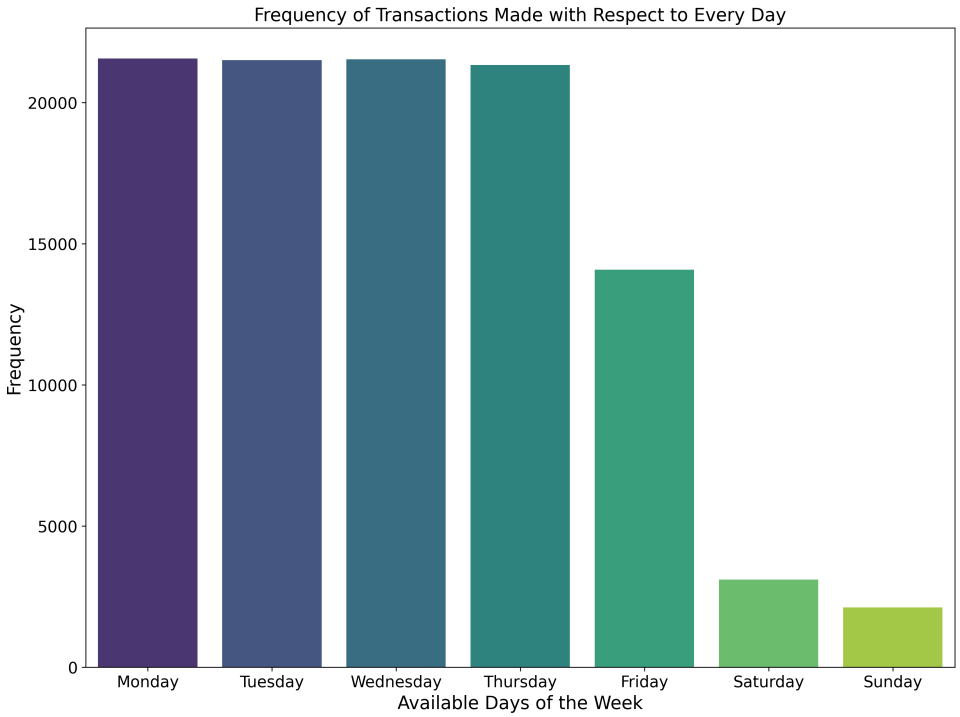

Figure 9: Bar plot indicating the frequency of transactions on a daily basis

In Figure 10, a plot encompassing two line plots comparing the aggregated spending and risk ratios for each weekday over our timeline present. Weekdays **(excluding Fridays)** generally show low-risk ratios despite higher aggregated spending. Conversely, there is a significant increase in risk ratios starting from Friday through the weekend, while aggregated spending tends to decrease. The graph illustrates a clear **negative correlation** between total spending and risk ratio, suggesting that risk events are more likely to occur over the weekend compared to weekdays highlighting the importance of keen monitoring during the weekends.

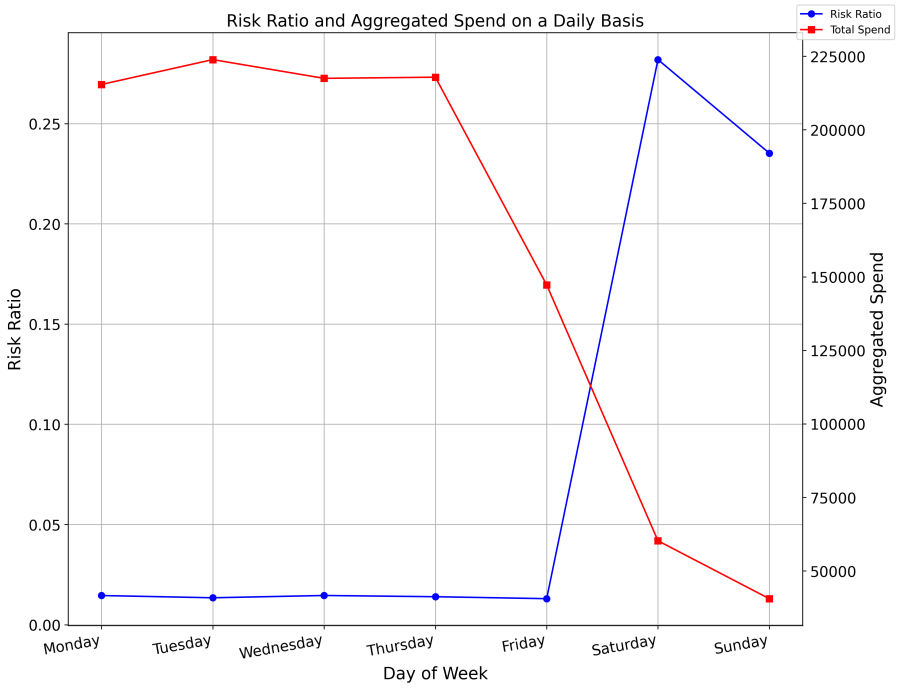

Figure 10: Risk Ratio and Aggregated Spend

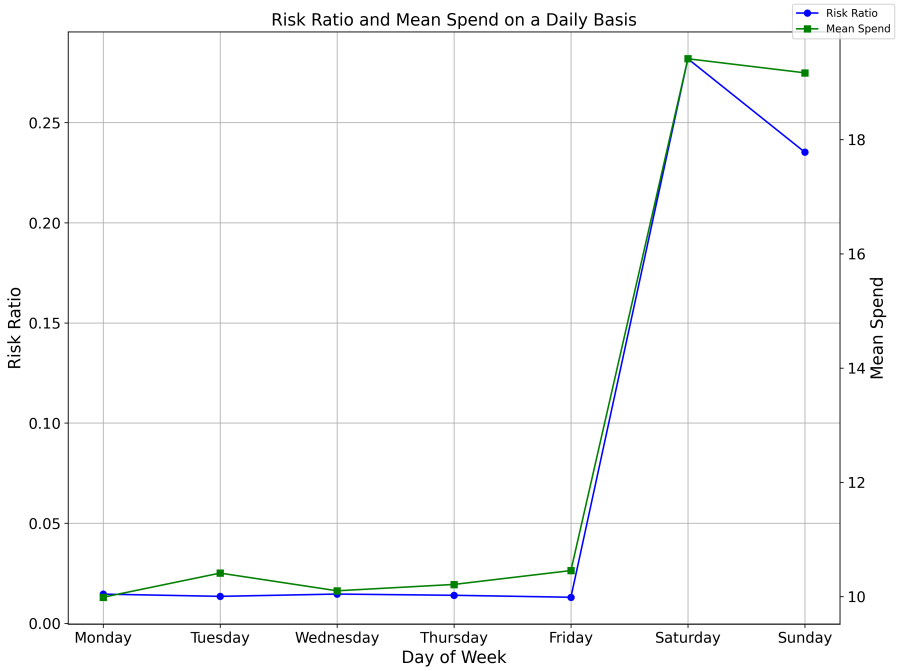

Figure 11: Risk Ratio and Mean Spend

Upon observing Figure 11, the line plots for risk ratio and mean spending reveal a notable spike on Saturday, indicating higher average spending and a higher probability of events being riskier compared to other days of the week. A key observation is that the risk ratio and mean spending align closely almost every day, deviating to some extent on Sunday. This analysis emphasizes the need for more **focused risk mitigation** practices during **weekends** compared to weekdays **(as suggested above)** given the consistently higher levels of both mean spending and risk ratio over the weekend.

3.3.2 Hourly Analysis

The bar plot below Figure 12 illustrates the number of transactions associated with each hour of the day. The hour column, derived from the timestamp feature and rounded off to the nearest whole number for easy plotting, reveals distinct patterns. Transactions show a steady increase from the early hours, peaking noticeably at 8 AM, and gradually climbing until midday **(around the 13th hour)**. Following this peak, there is a gradual decline until the late afternoon **(17th hour)**, with a more pronounced decrease towards the end of the day **(24th hour)**.

These trends highlight peak transaction activity during the typical **working hours**, roughly from **8:00 AM to 6:00 PM** which aligns with our business domain knowledge. There's a significant decline in transaction volumes on either side of the working hours, which is also consistent with the expected pattern of higher business activity during standard working hours compared to out-of-office hours.

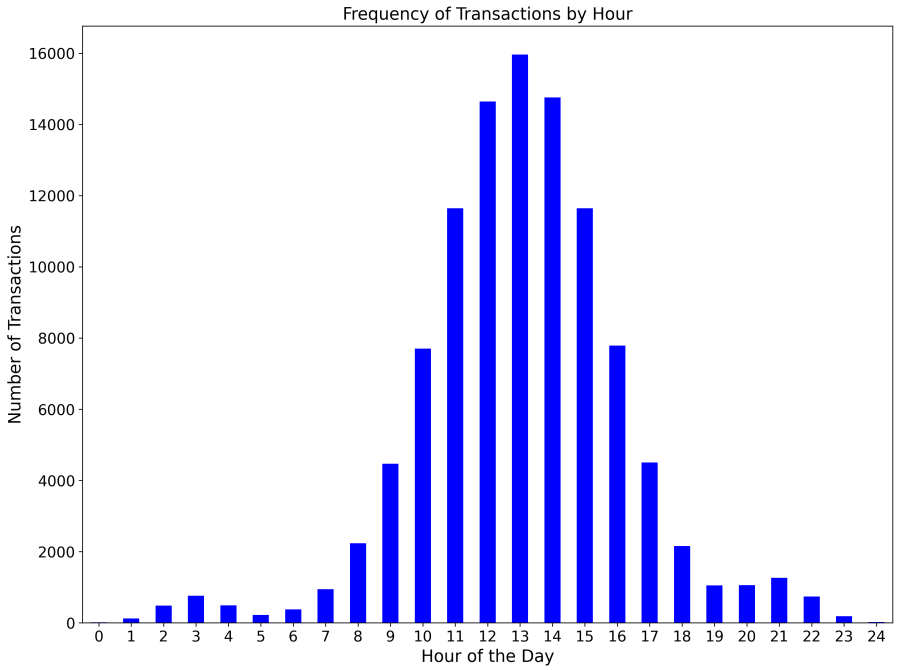

Figure 12: Frequency of transactions aggregated by every hour present

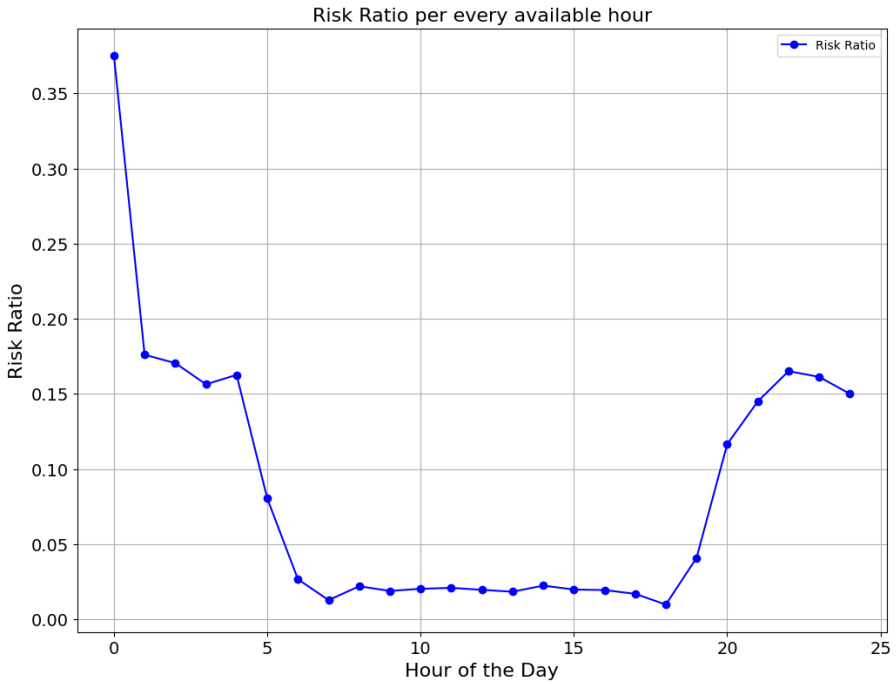

Figure 13: Risk Ratio on every hour present

The graph Figure 13 displays the risk ratio for every hour of the day, highlighting notable patterns in risk fluctuations. The risk ratio is highest around midnight **(12:00 AM)**, gradually declines, and hits a low at around 7:00 AM. It continues to stay low until 6:00 PM **(business working hours)**, after which there is a considerable increase. This pattern suggests that the risk ratio is higher outside typical business working hours **(aligning with our domain knowledge)**, indicating that events are more prone to risk during these times. This finding highlights the need for well-contemplated monitoring and enhanced risk management strategies to more focus on out-of-office hours.

3.4 Correlation Heatmap

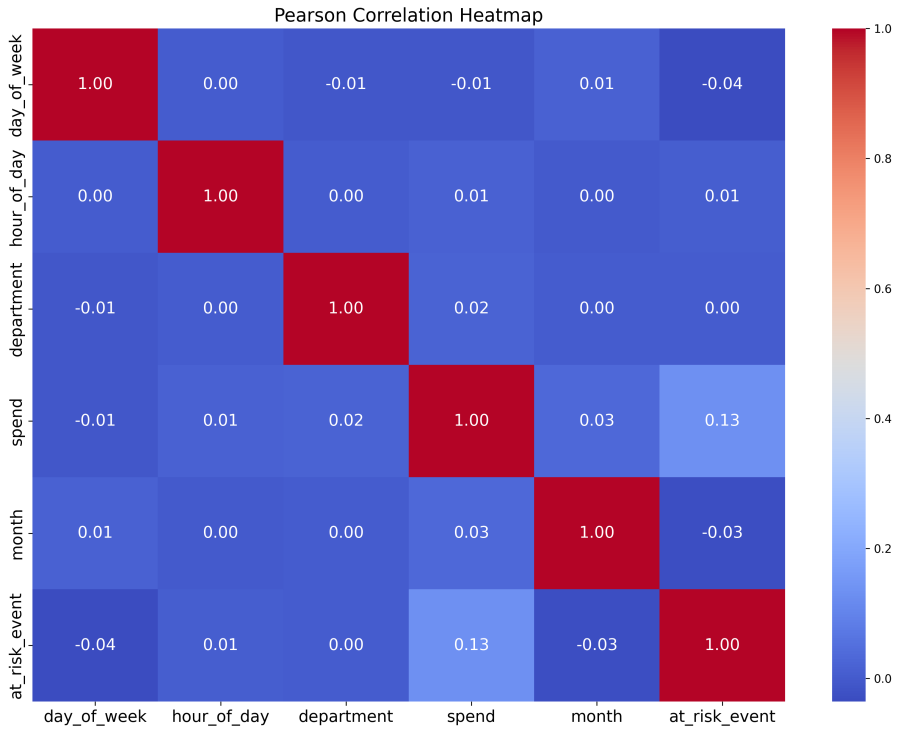

Figure 14: Pearson Correlation Heatmap

In Figure 14, the Pearson Correlation heatmap, reveals correlations among the features and their relationships with the target feature **(at_risk_event)**. The correlations are predominantly low and positive, suggesting minimal mutual influence among features and also with respect to the target feature. The notable exception is **'spend'**, which shows a **moderate positive correlation of 0.13** with the target feature. This underscores the necessity for extensive feature engineering to generate additional features and improve modeling efficacy in capturing risk events effectively.

4 Feature Engineering

Since we lack sufficient base features for the modeling stage, our exploratory data analysis has revealed the potential to derive multiple new features. Given the time series nature of the data, we can create two broad classes of features:

- Aggregate Level features.
- Rolling Window features.

4.1 Aggregate Features

Aggregate-level features are statistical summaries derived from raw data, typically computed over groups or categories to provide insights into comprehensive patterns and trends. These features aggregate and synthesize data at a broader scale, offering a more encompassing view of the underlying behaviors and characteristics within the dataset.

- **Aggregations on an individual level :**

  In this feature engineering process, we have developed several features to capture spending patterns at the individual level. By computing summary statistics such as mean **(mean_amount)**, median **(median_amount)**, sum **(total_amount)**, standard deviation **(std_amount)**, minimum **(min_amount)**, and maximum **(max_amount)** spent for each individual, we gain valuable insights into their expenditure behaviors. These features provide a detailed understanding of typical spending patterns, variability in spending habits, and the range of amounts spent, offering a comprehensive view of each individual's financial activities over time. Given that the 'spend' column alone does not reveal the detailed occurrences within each individual's spending events, these derived features allow us to capture granular patterns that could be effective later in the modeling stage.

- **Aggregation by individual and date :**

  In addition to the individual-level aggregate features, we have also aggregated spending data on a daily basis to capture short-term spending behavior. When there are chances of multiple transactions occurring over the same day for a particular individual, we intend to compute some additional features to capture daily spending patterns and variations. These features include the daily mean **(daily_mean_amount)**, which provides the average spend per day; the daily total **(daily_total_amount)**, which sums up all expenditures on that day; and the daily standard deviation **(daily_std_amount)**, which measures the variability of spending amounts within a day. These daily aggregated features offer insights into how spending fluctuates day-to-day for each individual, enriching our understanding and modeling of spending behaviors over shorter time intervals.

4.2 Rolling Window Features

In this feature engineering approach, we compute several rolling window features to extract nuanced insights from the spending data over the specified time interval. These features are computed over different intervals, such as 5 days and 7 days for each individual, providing a dynamic and comprehensive perspective on short-term spending behaviors. This approach is particularly insightful considering the contrasting expenditure patterns observed between weekdays and weekends**(as seen earlier)**. The **rolling_mean_amount_5d**, **rolling_std_amount_5d**, **rolling_sum_amount_5d**, and **rolling_transaction_count_5d** are some of the features derived **(likewise for the 7-day window)**. These features are instrumental in enhancing the predictive power of our model in detecting actual risk events by capturing temporal changes in spending behaviors.

5 Modelling

5.1 Conventional Technique

5.1.1 Isolation Forest

The Isolation Forest **(IF)**, as suggested by Belay et al. (2023)[4]is a widely used tree-based method for unsupervised anomaly detection. It operates on two key assumptions: anomalies are few and distinct. The algorithm constructs multiple binary trees **(iTree)** by randomly selecting features and splitting values at each node to isolate individual data points. This randomness effectively detects anomalies, as data points traverse down branches based on their feature values until they reach a leaf node, indicating isolation. Anomalies, which require fewer splits, result in shorter path lengths, whereas normal points necessitate more splits.

The anomaly score is calculated as per the given below formula :

$$s(x, n) = 2^{-\frac{E(h(x))}{c(n)}} \tag{5.1}$$

and c(n) is given by :

$$c(n) = 2H(n-1) - \frac{2(n-1)}{n} \tag{5.2}$$

n is the sample size, **h(x)** represents the path length of a particular data point in a given binary tree, **E(h(x))** is the expected value of this path length across all the binary trees, **H(i)** is the harmonic number and **c(n)** is the normalization factor defined as the average depth in an unsuccessful search in a Binary Search Tree **(BST)**. If this score exceeds a threshold determined through domain knowledge, the data point is classified as an anomaly.

5.1.2 One-class SVM

Traditional Support Vector Machines**(SVM)**have several disadvantages when applied to unsupervised anomaly detection, as they primarily require labeled data that encompasses normal and anomalous points, which is often impractical in real-world scenarios. On the other hand, One-Class SVM as suggested by Amer et al.(2013)[3] operates by learning a decision function for the single class **(normal data)** and identifying data points that lie outside this learned boundary as anomalies. One-Class SVM works by mapping the input data into a high-dimensional feature space and finding the maximal margin hyperplane that best separates the data from the origin, effectively distinguishing normal data from potential anomalies, which makes it one of the prominent algorithms in the domain of unsupervised anomaly detection.

The primary objective of the one-class SVM is described in the following equation:

$$\min_{w,\xi,\rho} \quad \frac{1}{2}\|w\|^2 - \rho + \frac{1}{\nu n}\sum_{i=1}^{n}\xi_i$$

subject to:

$$w^T\phi(x_i) \geq \rho - \xi_i, \quad \xi_i \geq 0$$

In here, ξ_i is the slack variable for point i that allows it to lie on the other side of the decision boundary, n is the size of the training dataset, and ν is the regularization parameter.

Furthermore, enhancements to One-Class SVM, such as the Robust One-Class SVM and Eta One-Class SVM, have been developed to reduce sensitivity to outliers, improving the accuracy and robustness of anomaly detection. These enhancements allow One-Class SVM to maintain computational efficiency while effectively handling outliers, making it a superior choice for unsupervised anomaly detection tasks.

5.2 DNN-based Methods

5.2.1 LSTM Autoencoder

One major constraint of Recurrent Neural Networks **(RNNs)** is their inability to retain long-term data, as they tend to forget previous inputs due to the vanishing and exploding gradient problem. Belay et al. (2023)[4] addressed this issue by Long Short Term Memory **(LSTM)** networks. LSTMs, a type of RNN, are designed to effectively handle long-term dependencies.

In LSTM architecture, there exists a cell state and three primary control gates: **the input gate, the forget gate, and the output gate**. The cell state acts as the memory component of the network, responsible for storing, updating, or retrieving information from previous states. The input and forget gates regulate the addition or deletion of long-term memory within the cell state, while the output gate controls the information output from the current hidden state.

The following equations are defined to describe the internal operations on the various gates present:

$$i^{(t)} = \sigma(W_{hi}h^{(t-1)} + W_{xi}x^{(t)} + b_i) \quad \text{(Input Gate)} \tag{5.3}$$

$$f^{(t)} = \sigma(W_{hf}h^{(t-1)} + W_{xf}x^{(t)} + b_f) \quad \text{(Forget Gate)} \tag{5.4}$$

$$o^{(t)} = \sigma(W_{ho}h^{(t-1)} + W_{xo}x^{(t)} + b_o) \quad \text{(Output Gate)} \tag{5.5}$$

$$\tilde{c}^{(t)} = \tanh(W_{hc}h^{(t-1)} + W_{xc}x^{(t)} + b_c) \quad \text{(Candidate Cell State)} \tag{5.6}$$

$$C^{(t)} = f^{(t)} \circ C^{(t-1)} + (1 - f^{(t)}) \circ \tilde{c}^{(t)} \quad \text{(Cell State)} \tag{5.7}$$

$$h^{(t)} = o^{(t)} \circ \tanh(C^{(t)}) \quad \text{(Hidden State)} \tag{5.8}$$

Here $\sigma(\cdot)$ is the sigmoid function, $\circ$ denotes the Hadamard product, W is a weight matrix, and b is the bias vector in each gate.

As Nguyen et al. (2021)[11] suggested, LSTM autoencoders are particularly effective for anomaly detection in multivariate time series data. An autoencoder, comprising of an encoder and decoder based on LSTM networks, learns to compress input data into a lower-dimensional latent space and reconstruct it back to the original dimensions. By minimizing reconstruction error, the autoencoder captures essential data features.

The reconstruction error is given by the following equation:

$$L = \frac{1}{2}\sum_{x} \|x - \hat{x}\|^2 \tag{5.9}$$

where x is the given input and is the output representation which we have got from the decoder present.

The steps involved in detecting anomalies through LSTM autoencoders in our model include defining the autoencoder with an optimal number of sequential layers, training it on the available data, establishing a threshold for the reconstruction error using a suitable metric, and assessing anomalous points in the test dataset. An instance is classified as an anomaly if its reconstruction error exceeds the designated threshold.

6 Results and Inferences

Model Name	No. of Features	Precision	Recall	F1-Score	AUC	AUPR
Isolation Forest	15	0.104	0.743	0.183	0.184	0.016
	20	0.106	0.762	0.186	0.192	0.016
	25	0.115	**0.835**	0.203	0.124	0.015
	31	0.114	0.820	0.200	0.140	0.015
One Class SVM	15	0.145	0.478	0.222	0.176	0.016
	20	0.130	0.459	0.203	0.186	0.016
	25	0.141	0.485	0.218	0.101	0.015
	31	0.123	0.447	0.201	0.122	0.015
LSTM AutoEncoder	15	**0.187**	0.296	**0.230**	**0.833**	**0.173**
	20	0.118	0.215	0.153	0.724	0.088
	25	0.125	0.222	0.160	0.779	0.103
	31	0.085	0.152	0.109	0.767	0.093

Table 1: Various metric measures tabulated for Model comparison

The models are trained on the **training data** which has **84,719 records**, and the **test dataset** has **20,523 records**, split based on their individual_id. Following the 80-20 rule, we have records of **1,742 individuals** selected at **random** for the **training set** and the remaining **443 individuals** for the **test set**. Feature selection is performed using the Principal Component Analysis**(PCA)** algorithm, and four different feature sets of varying lengths are evaluated as mentioned in the above Table 1. The models compared are **Isolation Forest**, **One-Class SVM**, and **LSTM AutoEncoder**.

The **Isolation Forest** model demonstrates a **high recall** measure of **0.835** with 25 features suggesting that it is highly effective in identifying actual anomalies**(risk events)**. However, its precision is generally low across all feature sets indicating a higher number of false positives. The best F1-score achieved is 0.203 with 25 features. The AUC**(Area Under Curve)** and AUPR**(Area under the precision-recall curve)**values are relatively low indicating a moderate ability to distinguish between normal and anomalous data.

The **One-Class SVM** model shows a more balanced performance between precision and recall compared to Isolation Forest. The **highest F1-score** achieved is **0.222** with 15 features, indicating a better balance between precision and recall. The precision measures for One-Class SVM are slightly on the higher end compared to the other two models, whereas recall values are lower than those of Isolation Forest. The AUC and AUPR values are similar to the Isolation Forest, indicating a moderate performance in distinguishing between normal and risk events.

The **LSTM AutoEncoder** model stands out in terms of AUC and AUPR, achieving the **highest AUC** of **0.833** and **AUPR** of **0.173** with 15 features. This indicates a strong overall discrimination capability between normal and anomalous data. The precision measures do not take a spike whereas recall values hit a new low. The **best F1-score** achieved is **0.230** with 15 features which is the greatest among all models present. Despite lower precision and recall, the high AUC and AUPR values suggest that LSTM AutoEncoder is highly effective at distinguishing true risk events from normal data.

All three models show a decrease in performance metrics as the number of features increases beyond 15 except the Isolation Forest suggesting that additional features may introduce noise rather than useful information and lesser computational complexity. The AUPR scores are consistently low across all models, indicating a common challenge in identifying anomalies with high precision. Additionally, none of the models achieve exceptionally high F1 scores, pointing to the difficulty of risk detection.

In conclusion, each model has its strengths and weaknesses in unsupervised anomaly detection.

Isolation Forest excels in recall, making it suitable for applications where identifying as many true anomalies as possible is crucial, even at the expense of false positives. One-class SVM provides a more balanced performance between precision and recall, making it suitable for scenarios where a balance between true positive rate and false positive rate is desired. LSTM AutoEncoder, with its high AUC and AUPR values, is best suited for applications where the overall discrimination capability between normal and anomalous data is paramount, despite its lower precision and recall. Therefore, the choice of model should depend on the specific requirements of the anomaly detection task at hand.

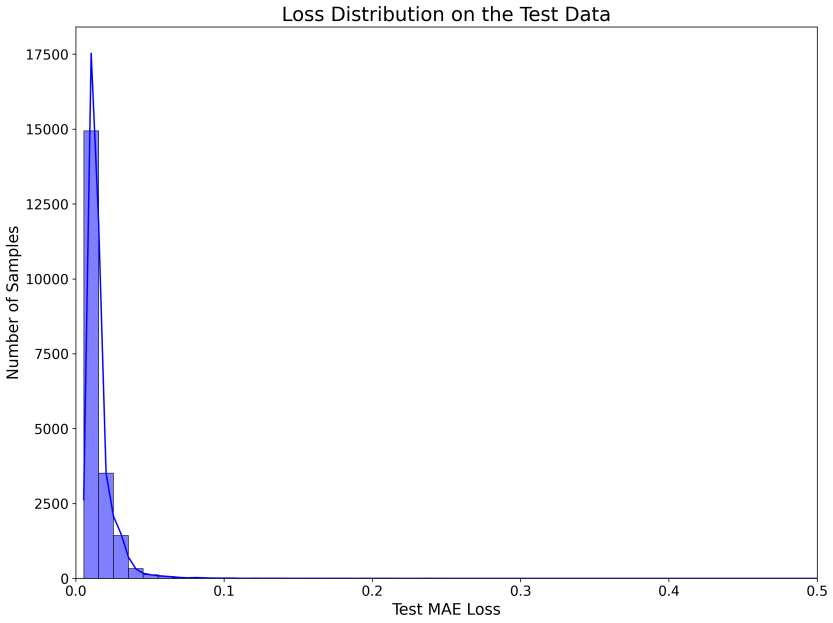

Figure 15: Reconstruction error on the test data

The above figure Figure 15illustrates the reconstruction error on the test dataset for the best LSTM Autoencoder model **(the model with 15 features)**. The plot distribution is heavily skewed towards lower values, indicating that the model reconstructs most test samples with high accuracy, resulting in low reconstruction errors. A large concentration of samples with near-zero MAE loss suggests that these samples exhibit normal behavior similar to the training data, demonstrating the model's good generalization capabilities.

7 Conclusions

This study demonstrates the effectiveness of unsupervised machine learning models in detecting anomalies within complex spending data. After extensive exploratory data analysis and feature engineering, we implemented and compared Isolation Forest, One-Class SVM, and LSTM AutoEncoder models using PCA-selected features. Isolation Forest showed high recall but low precision, indicating many false positives. One-class SVM achieved a balanced performance between precision and recall, whereas the LSTM AutoEncoder excelled in overall discrimination, achieving the highest AUC and AUPR scores, highlighting its robustness in distinguishing normal from anomalous data. These findings suggest that LSTM AutoEncoder is preferable for tasks requiring strong discrimination capabilities, while Isolation Forest and One-Class SVM are effective where recall takes priority. Future work could explore hypernetworks to handle varied transaction counts, balancing techniques for equal weekday and weekend representation, and composite models to enhance accuracy and reliability in anomaly detection.

References

[1] M. Ahmed, A. Naser Mahmood, and J. Hu. A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, 60:19–31, 2016.

[2] S. Akcay, D. Ameln, A. Vaidya, B. Lakshmanan, N. Ahuja, and U. Genc. Anomalib: A deep learning library for anomaly detection. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 1706–1710, 2022.

[3] M. Amer, M. Goldstein, and S. Abdennadher. Enhancing one-class support vector machines for unsupervised anomaly detection. In *Proceedings of the ACM SIGKDD workshop on outlier detection and description*, pages 8–15, 2013.

[4] M. A. Belay, S. S. Blakseth, A. Rasheed, and P. Salvo Rossi. Unsupervised anomaly detection for iot-based multivariate time series: Existing solutions, performance analysis and future directions. *Sensors*, 23(5):2844, 2023.

[5] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3), jul 2009.

[6] R. B. Cleveland, W. S. Cleveland, J. E. McRae, I. Terpenning, et al. Stl: A seasonal-trend decomposition. *J. off. Stat*, 6(1):3–73, 1990.

[7] W. Hilal, S. A. Gadsden, and J. Yawney. Financial fraud: a review of anomaly detection techniques and recent advances. *Expert systems With applications*, 193:116429, 2022.

[8] K. Leung and C. Leckie. Unsupervised anomaly detection in network intrusion detection using clusters. In *Proceedings of the Twenty-eighth Australasian conference on Computer Science-Volume 38*, pages 333–342, 2005.

[9] M. Munir, S. A. Siddiqui, A. Dengel, and S. Ahmed. Deepant: A deep learning approach for unsupervised anomaly detection in time series. *IEEE Access*, 7:1991–2005, 2019.

[10] N. Najari, S. Berlemont, G. Lefebvre, S. Duffner, and C. Garcia. Resist: Robust transformer for unsupervised time series anomaly detection. In *International Workshop on Advanced Analytics and Learning on Temporal Data*, pages 66–82. Springer, 2022.

[11] H. Nguyen, K. Tran, S. Thomassey, and M. Hamad. Forecasting and anomaly detection approaches using lstm and lstm autoencoder techniques with the applications in supply chain management. *International Journal of Information Management*, 57:102282, 2021.

[12] M. A. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko. A review of novelty detection. *Signal Processing*, 99:215–249, 2014.

[13] O. I. Provotar, Y. M. Linder, and M. M. Veres. Unsupervised anomaly detection in time series using lstm-based autoencoders. In *2019 IEEE International Conference on Advanced Trends in Information Theory (ATIT)*, pages 513–517, 2019.

[14] T. Qin, X. Guan, W. Li, P. Wang, and Q. Huang. Monitoring abnormal network traffic based on blind source separation approach. *Journal of Network and Computer Applications*, 34(5):1732–1742, 2011. Dependable Multimedia Communications: Systems, Services, and Applications.

[15] M. Rezapour. Anomaly detection using unsupervised methods: credit card fraud case study. *International Journal of Advanced Computer Science and Applications*, 10(11), 2019.

[16] K. Yamanishi and J.-i. Takeuchi. A unifying framework for detecting outliers and change points from non-stationary time series data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, page 676–681, New York, NY, USA, 2002. Association for Computing Machinery.

Appendix

A Data Preprocessing

The dataset consisted of 105,277 records of time series data, covering entries for approximately 2,185 individuals across 10 different columns. No null values were found in the primary columns, except for the 'at_risk_window' feature column, which was excluded from analysis as it serves as a target variable for another use case. A significant preprocessing step involved identifying and removing 35 records where the 'spend' column had a value of zero. These records were considered irrelevant due to the importance of 'spend' in our ongoing analysis. Subsequently, the dataframe was updated accordingly.

B Other Exploratory Data Analysis

B.1 Monthly Analysis

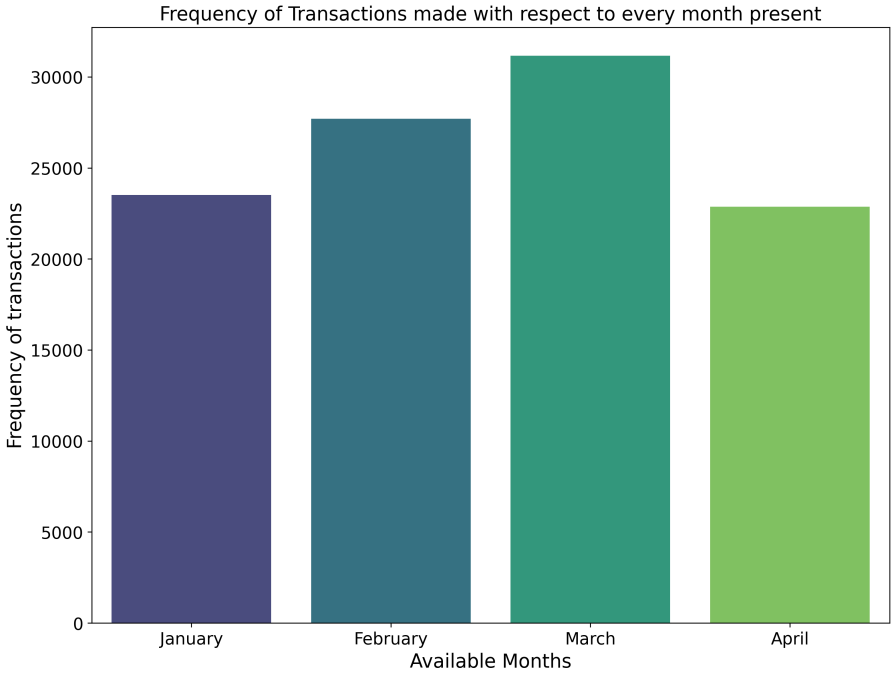

Figure 16: Barplot on the frequency of transactions aggregated over every month

We were given a 'datetime' feature column, from which they have already extracted the hour and day as separate columns. Additionally, we derived a 'Month' column to capture the respective month. After aggregating the entire dataframe across the available months **(January, February, March, and April)**, we obtained the aggregated count of transactions for each month over the entire period as shown in Figure 16. There is a consistent distribution of transactions across the four months, with March slightly surpassing the other three in transaction volume.

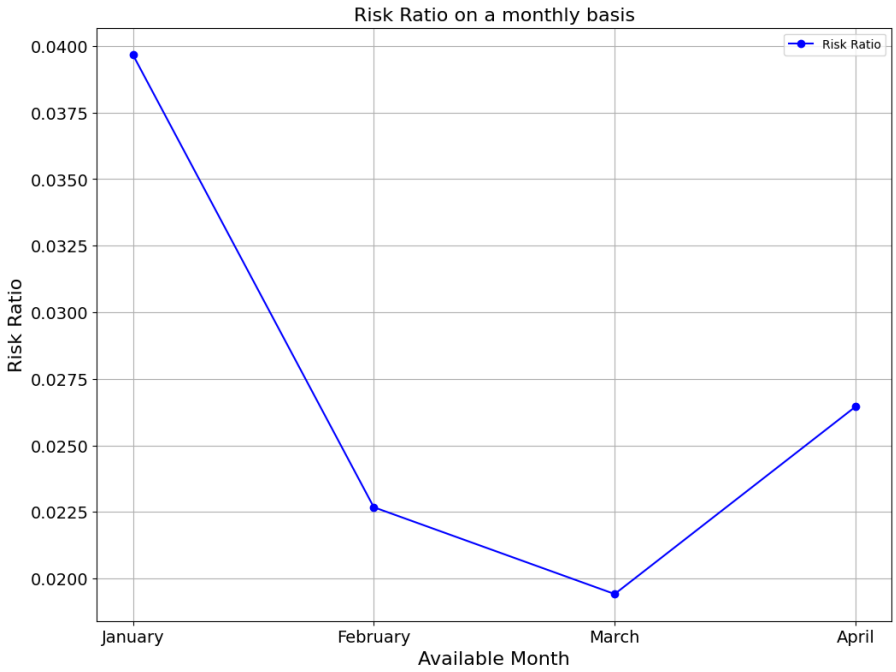

Figure 17: Risk Ratio on every month

Figure 17 provides a visual representation of the risk ratio observed across each available month. The risk ratio appears highest in January, dips in the following two months, and shows a slight increase in April. Further analysis can delve into understanding the reasons behind these varied risky behaviors, possibly by examining the day and time of transactions to gain insights.

C Derived Features

Other than the aggregate and rolling window features (mentioned in the report earlier), we have derived some other key features based on domain knowledge which are as follows:

- **Time Since Last Transaction:** This feature help us understand the difference in time (in hours) since the last transaction for that particular individual.
- **Amount to Mean Ratio:** This feature is calculated as the ratio between the spend amount to the average amount of that particular individual.
- **Is Weekend:** This feature takes in binary values help us understand whether the transaction happens on a weekday or over the weekend.

Other features include finding the average and total sum on the basis of their department **(likewise for day and hour)**.

D Feature Selection

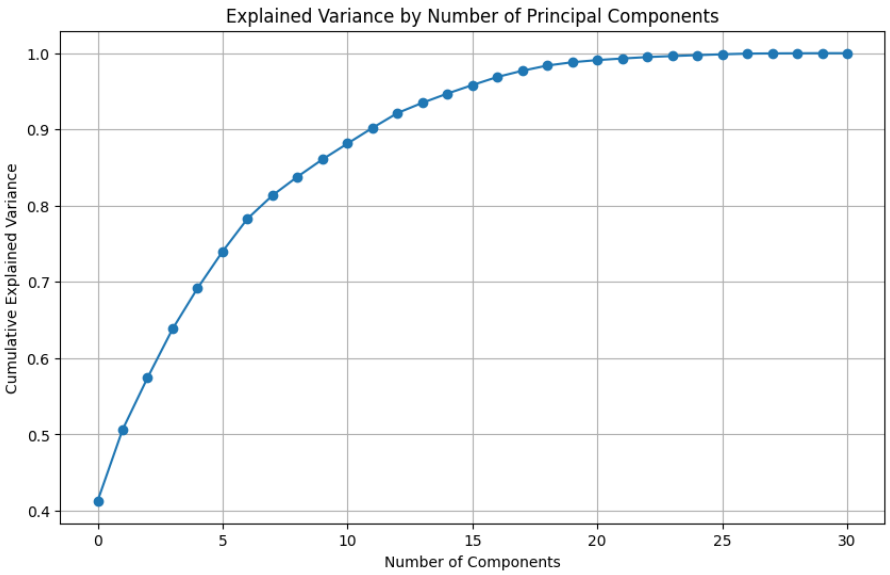

Figure 18: Cumulative Explained Variance Graph

As part of the feature selection process, Principal Component Analysis (PCA) is used in an unsupervised manner. Initially, we determined the number of components required to retain approximately 95% of the variance, as depicted in the variance graph shown in Figure 18. It is evident from the graph that 16 components are sufficient for this purpose, and we initialized the PCA model accordingly. Following this, we fitted the model on our training data and selected our desired number of top features based on the results. In Table 1, the column labeled 'No.of Features' indicates the count, which is nothing but the number of features **(best ones)** we select from this PCA algorithm.