The School of Mathematics

THE UNIVERSITY
of EDINBURGH

Risk Events and Risk Behaviour Windows Detection with Univariate LSTM Model

by

Adlensius Fransiskus Djunaedi

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

July 2024

Supervised by
Dr Tim Cannings and Dr Cecilia Balocchi

Executive Summary

This report aims to develop an unsupervised modelling process/framework to detect risk events and risk behaviour windows in Lloyds Banking Group employees' spending activities. Panel data on 2185 Lloyds Banking Group employees was collected, which included their spending amounts, departments, dates, days of the week, and time stamps of activities. Two target flags, an indicator of whether an activity is a risk event and an indicator of whether an activity comprises a risk behaviour window, were also provided for model evaluation purposes.

Time-series anomaly detection models that can identify risky behaviours were then constructed. The framework includes model selection, threshold computation, and post-processing to prune false positives and address weekend effect. The weekend effect refers to any weekend anomalous spending due to it either being the first weekend spending or that 29 days or longer have passed following its preceding weekend spending. Furthermore, to preserve the sequential nature of the data, rolling window analysis with a fixed window of size 5 was implemented. It is crucial for the model to be able to not only accurately detect risky behaviours to prevent unwanted loss and ensure stability, but also refrain from raising excessive false alarms to curb unnecessar ty costs and maintain the user's trust level in the model. Therefore, the F1 score metric, which captures the balance of both criteria, is employed to measure models' performances.

Overall, results show that the univariate LSTM model with the dynamic threshold is the best model for identifying risk events as it yielded the maximum F1 scores of 31.59% for training data and 31.08% for testing data. Post-processing the predicted anomalies by pruning false positives slightly increased the F1 scores and taking into account the weekend effect raised the F1 scores by almost 2.5 times to 77.40% for the training data and 74.46% for the testing data. This indicates that the model could adequately predict non-weekend effect risk events and addressing the weekend effect during post-processing is crucial for optimal performance.

In detecting risk behaviour windows, the univariate LSTM model with the dynamic threshold was also found to perform the best with F1 scores of 17.41% for the training data and 15.10% for the testing data. Its performance, however, was underwhelming compared to when it was employed in detecting risk events. This result is expected due to the more complicated preemptive nature of a risk behaviour window compared to the more simplistic temporally instantaneous nature of a risk event. Subsequently, false positive pruning was found to be detrimental to the model's performance and thus, it was omitted. Taking into account the weekend effect anomalies offered only slight improvement to the F1 scores, raising them to 17.64% for training data and 18.45% for testing data. This implies that the weekend effect is not as influential in determining risk behaviour window compared to their presence in risk events.

It is thus recommended for Lloyds Banking Group to employ the univariate LSTM model with the dynamic threshold for identifying both risk events and risk behaviour windows. Post-processing for detecting risk events should include false positive pruning and addressing the weekend effect. On the other hand, post-processing for detecting risk behaviour windows should only address the weekend effect to obtain optimal results. Hopefully, the implementation of the proposed frameworks could contribute in helping Lloyds Banking Group achieve a more comprehensive understanding of risky behaviour detection and ensure appropriate measures could be taken promptly.

Nevertheless, the proposed models have some limitations. Firstly, there is no method for determining the optimal length of the rolling window for the LSTM models. Furthermore, a natural limitation of a rolling window analysis is that the forecasts for the first few observations are unattainable. Additionally, an employee should have more spending data than the window length for the model to be applicable. Finally, the proposed LSTM model may not be able to fully capture the temporal irregularity of the dataset. More advanced models, such as conditional RNN, or curve-fitting approaches, including Gaussian process regression, could be explored in future studies to help model more complex temporal dynamics.

Acknowledgments

I would like to thank my supervisors, Dr Tim Cannings and Dr Cecilia Balocchi, for their continuous guidance and support over the course of this project. I also would like to thank PhD student helper Johnny MyungWon Lee for sharing his experience and advice in writing dissertations and reports. I also would like to give special thanks to the Lloyds Banking Group team, especially Callum Hodgkinson, Dimitrios Ntakoulas, and George Deskas, for providing the opportunity and the dataset to work with. Their expertise and knowledge played a huge part in the completion of this report. This project would not have been possible without them. Finally, I would like to thank my friends for sharing their knowledge and experience during discussions.

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: Adlensius Fransiskus Djunaedi
Matriculation Number: S2591760
Title of work: Risk Events and Risk Behaviour Windows Detection with Univariate LSTM Model

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: Adlensius Fransiskus Djunaedi

Date: 27 June 2024

Contents

List of Tables

List of Figures

1 Introduction

1.1 Background and Motivation

As a leading financial services group, Lloyds Banking Group (LBG) provides a multitude of banking and financial services for people and businesses all across the UK [10]. Considering its magnitude and influence, an internal fraudulent act could endanger both LBG's customer well-being and internal stability. Thus, the development of a model that could reliably identify internal risky behaviour is crucial such that early monitoring and appropriate measures can be taken to prevent malicious acts against LBG.

Noting that risky behaviours could be identified by their deviation from normal behaviours, time-series outlier/anomaly detection techniques are employed to detect threatening risks as outliers appear when the data-generating process behaves abnormally [1]. They attempt to identify the time instants of such unexpected anomalous behaviour within a sequence of measurements [1]. Note that although in general all risks are anomalous, not all anomalies are risks. However, the words anomaly and threatening risks will be used interchangeably in this report as detecting anomalies that are not threats is not of interest.

Unlike ordinary anomaly detection techniques, time-series anomaly detection techniques pose their own challenges as they need to take into account the time dependency between data points along with other time series characteristics such as trends and seasonality. In particular, time-series anomalies can be classified into three different categories: point anomalies, contextual anomalies, and collective anomalies [2]. Point anomalies occur when a single observation significantly deviates from the rest of the observations while contextual anomalies only occur when an observation is anomalous within its context of similar observations. Finally, a collective anomaly occurs when a sub-sequence of observations behaves unusually compared to other sub-sequences. Further, this report considers the situation where anomaly labels are not available, a common scenario in many real datasets, and thus, an unsupervised model shall be considered. The provided anomaly labels will only be used for model evaluation instead of training.

A robust detection of local risks would require an algorithm that could address the aforementioned issues. Hence, it is believed that a prediction and reconstruction approach to anomaly detection using Long Short-Term Memory (LSTM) models, a type of Recurrent Neural Network (RNN), are ideal for this model as they are able to process sequential data and capable of learning long-term dependencies. Moreover, threshold computations, false positive pruning, and weekend effect will also be explored to further address seasonal and trend effects in the series.

1.2 Objective

This report aims to build an unsupervised time series modeling process/framework that could effectively identify risk events and risk behaviour windows based on LBG employees' spending data from multiple departments. Assuming that anomalies are rare events in the dataset, the F1 score shall be used as performance metrics instead of accuracy as it measures the balance between recall and precision. Two target variables, risk events and risk behaviour windows, will be used to judge models' performances with the former as the main target. As the dataset contains multiple features, including time-related, department, and spending data, both univariate and multivariate models will be considered. The methods and results presented in this report are expected to provide LBG with a deeper insight into risk events and risk behaviour windows detection methods to ensure prompt reactions to abnormal activities.

The remainder of this report is organised as follows. Section 2 presents a preliminary analysis of the data which includes exploratory data analysis and feature engineering process. Section 3 provides a brief review of the methods including model selection, threshold computations, and post-processing. Section 4 then presents model performances and a discussion of the results. Finally, section 5 concludes the report.

2 Exploratory Data Analysis and Feature Engineering

Panel data of 2185 individuals from 20 different departments of spending amounts and their corresponding date, day of the week, and timestamps in addition to the two target variables, risk events and risk behaviour windows, was provided by LBG. Table 1 presents the descriptions of the features in the dataset and Table 2 shows the descriptions of the target variables.

Features	Description
`individual_id`	a unique identifier of an employee
`timestamp`	minutes and seconds timestamp of the event
`date`	date of the event
`day_of_week`	the day of the week of the event
`hour_of_day`	the hour of the day of the event
`department`	the department the employee sits in
`spend`	the spending amount of an event (in 1000)

Table 1: Features in the Dataset

Target Variables	Description
`at_risk_event`	whether the event posed a risk
`at_risk_behaviour_window`	whether the individual has started to become a risk

Table 2: Target variables in the Dataset

Figure 1 illustrates the risk events and risk behaviour windows for individuals 1, 100, and 1734. It can be observed that there is no clear relationship between risk events and risk windows. A risk window does not necessarily contain a risk event and a risk event is not necessarily contained within a risk window. Thus, we shall consider the two targets separately.

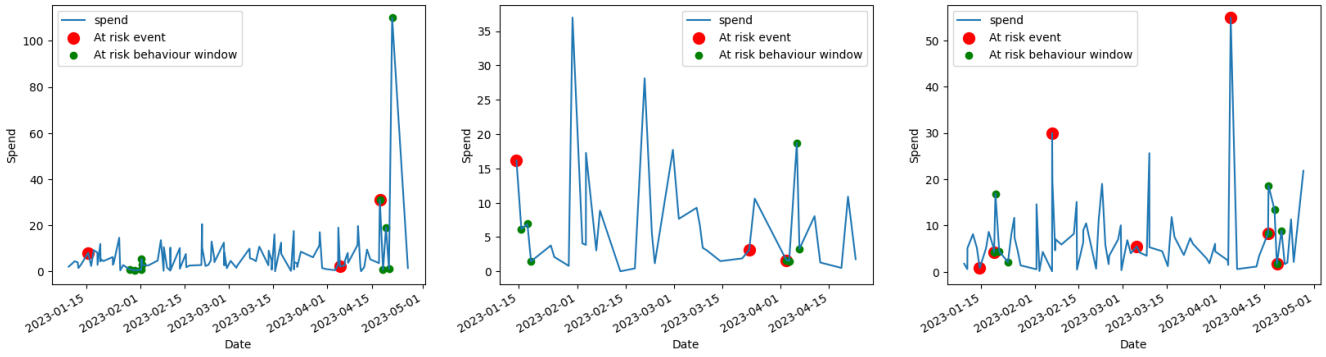

Figure 1: Risk events and risk behaviour windows of individuals 1 (left), 100 (middle), and 1734 (right)

Figure 2 shows the mean anomalous and normal spending for each department. The mean normal spending of some departments, such as secretariat, general, exec, and development, can be observed to be lower than the others. However, while different departments may have different spending behaviours, it is clear that high spending is more likely to be a risk than low spending as the mean of anomalous spending for all departments are strictly higher than the normal spending. A similar inference can be made from the density plots in Figure 3. The densities of normal spending from all departments have similar shapes and are concentrated at a similar point differing only in the degree of spread. The anomalous spending,

however, are much more dispersed than the normal spending with most of them exhibiting heavier right tails. Thus, high spending tends to be associated with risky behaviour.

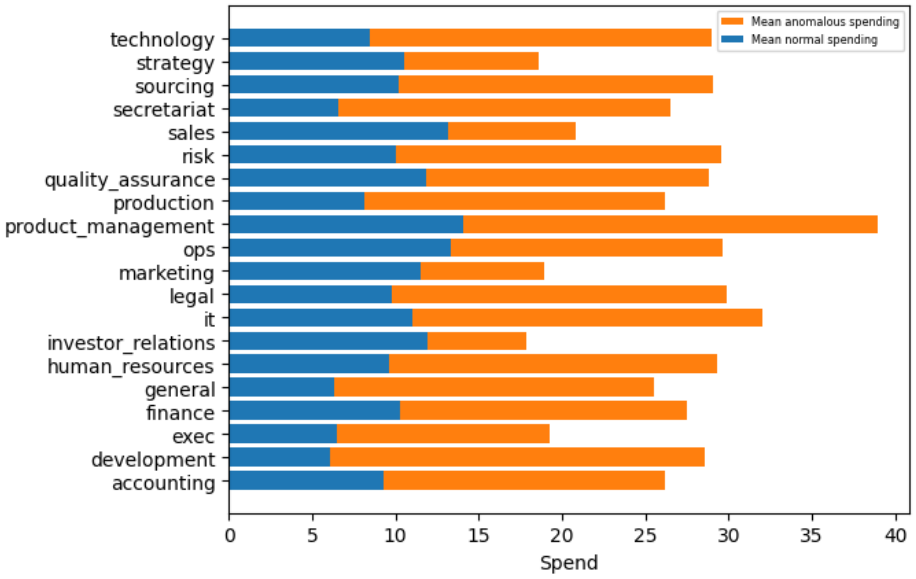

Figure 2: Mean normal (blue) and anomalous (orange) spending by department

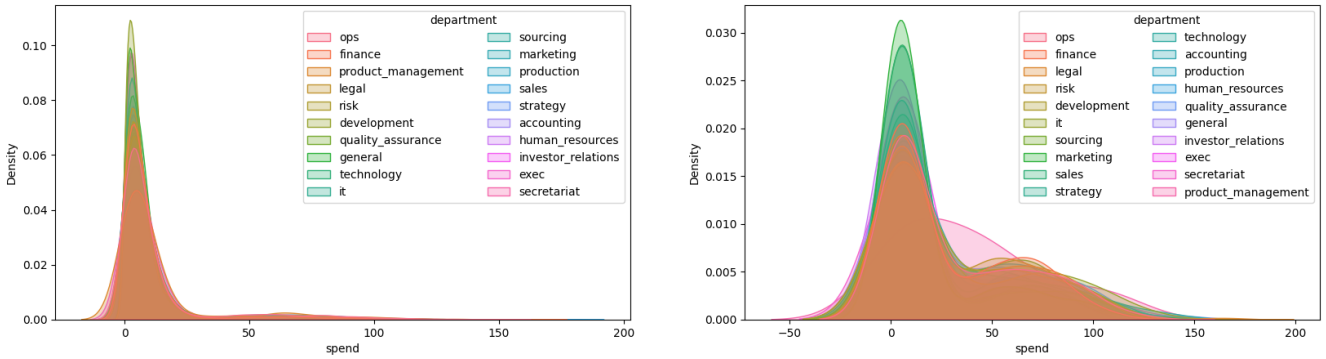

Figure 3: Density plots of normal (left) and anomalous (right) spending by department

Figure 4 illustrates the mean anomalous and normal spending behaviour depending on the hours of the day. In general, a low stable mean normal spending pattern can be observed between 6:00 and 18:00 while relatively high spending occurs between 19:00 to 5:00. On the other hand, the opposite trend can be seen for mean anomalous spending where high spending occurs between 6:00 and 18:00 while low spending occurs between 18:00 and 5:00. Therefore, to address the effect of the hour of the day on spending behaviour, a new binary variable, `work_hour`, is constructed where it takes the value 1 if spending occurs between 6:00 and 18:00 and 0 if spending occurs between 19:00 and 5:00.

Table 3 summarizes the anomalous behaviour on the weekend. It can be seen that 49.55% of the total anomalies happen during the weekend. A little above 26% of weekend spending are anomalies while less than 2% of weekday spending are anomalies. Therefore, weekend activities have a relatively high tendency to be flagged as anomalies. A new indicator variable, `weekend`, is created to capture this pattern.

Furthermore, there is a 100% rate of the first weekend activity of an individual being an anomaly and around 86% of subsequent weekend activities being anomalous given that more than a month has passed after the previous weekend activity. Section 3.3.2 addresses this weekend effect by creating a separate flag

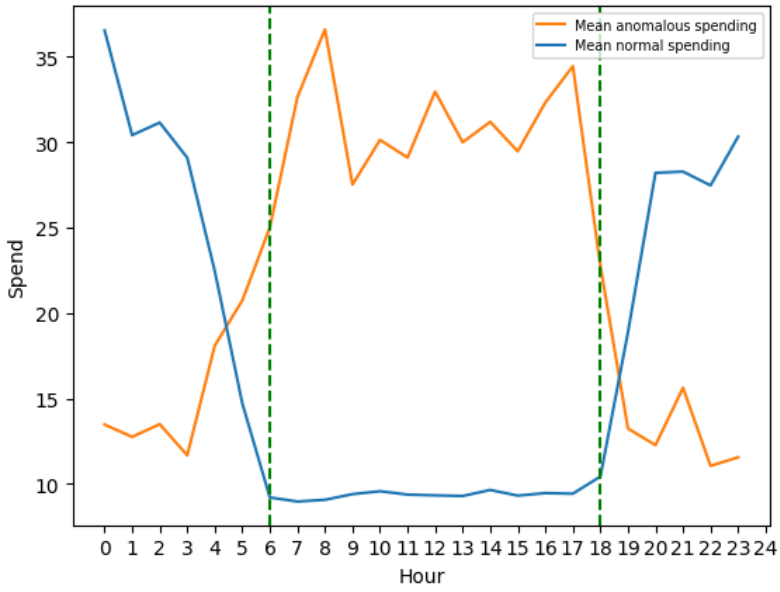

Figure 4: Mean normal (blue) and anomalous (orange) spending by the hour of the day

Weekend/ weekday	**Number of Transactions**	**Number of Anomalies**	**Anomaly Proportion**
Weekend	5222	1373	26.29%
Weekday	100055	1398	1.40%

Table 3: Summary of anomalous weekend spending

for weekend activities.

It is also noted that the dataset is an irregularly spaced time series data. Although it is common practice to treat irregularly spaced time series data as if it were regular, this may introduce bias into predictions as conventional models may fail to model the temporal irregularity [3, 17]. Therefore, a new variable, `time_diff`, which captures the time difference between two consecutive spending activities is constructed to capture the dependence.

Finally, to appropriately test models and methods performance, data is split into training and testing sets. Noting that the dataset is panel data, splitting is done across individuals instead of data entry to preserve the sequential nature of the data. 70% of the individuals are allocated to training data while the remaining 30% of the individuals are allocated to test data.

3 Methods

We divide the anomaly detection process into three stages. First, appropriate time-series anomaly detection models are constructed. Subsequently, three methods of threshold computation, which are simple quantile threshold, department-based threshold, and dynamic threshold, are explored. The best threshold computation method is then chosen to produce the initial forecasts. Finally, predictions are further refined by post-processing through false positive pruning and taking into account the weekend effect. Henceforth anomalous spending will be defined as positive cases while normal spending will be defined as negative cases.

Models' performances are measured by the F1 score, which measures the balance between recall and precision. Recall measures a model's capability in correctly identifying sufficient anomalies while precision

ensures a model is not generating excessive false positives. While detecting anomalies is crucial as failure to do so may cause tremendous loss to LBG, excessive false alarms may incur unnecessary costs and lower user's trust in the anomaly detection model, rendering the model unusable. Thus, it is critical for the model to strike a good balance between precision and recall. The formulas for the metrics are as follows.

$$\begin{aligned} Precision &= \frac{TP}{TP + FP} \\ Recall &= \frac{TP}{TP + FN} \\ F1\ score &= \frac{2 \times Precision \times Recall}{Precision + Recall}, \end{aligned}$$

where TP, FP, FN denote the number of true positives, false positives, and false negatives. All numerical variables are also standardised to improve the models' performances.

3.1 Model Selection: Persistence Model and LSTM

The persistence model is chosen as the baseline model. The persistence model produces naive forecasts where it assumes that the current spending is equal to the spending at the previous time step. In other words, for observed data at time $t-1$, y_{t-1}, the predicted spending at time t is

$$\hat{y}_t = y_{t-1}.$$

To take into account the sequential nature of the data, LSTM models are considered for detecting anomalous events. Three models in total will be considered in this report, which are the univariate LSTM, multivariate LSTM, and LSTM autoencoder.

The LSTM model was first proposed to tackle the vanishing gradient issue in RNN by introducing multiplicative gates that admit constant error to go through internal memory cells [5]. Figure 5 illustrates the architecture of an LSTM memory cell. Its construction of three gates, the forget, input, and output gates, allow LSTM to retain relevant long-term information and remove irrelevant memory [12]. LSTM is thus an ideal model for complex time-series sequences as it can capture long-term dependencies through both forgetting and accumulating past information [6].

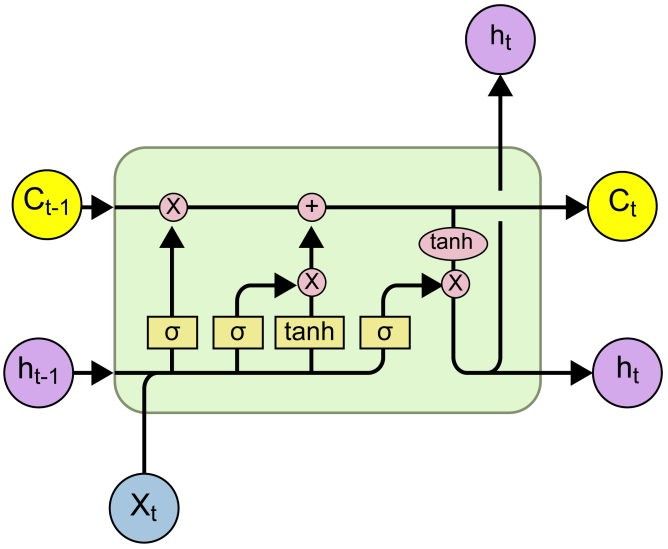

Figure 5: LSTM memory cell architecture [8]

Let f_t be the forget gate, i_t be the input gate, o_t be the output gate, c_t be the cell state, and h_t be

the hidden state. Then, at time step t, each component can be mathematically written as

$$\begin{aligned} f_t &= \sigma(W_f \times x_t + U_f \times h_{t-1} + b_f) \\ i_t &= \sigma(W_i \times x_t + U_i \times h_{t-1} + b_i) \\ o_t &= \sigma(W_o \times x_t + U_o \times h_{t-1} + b_o) \\ c'_t &= tanh(W_c \times x_t + U_c \times h_{t-1} + b_c) \\ c_t = f_t \cdot c_{t-1} + i_t \cdot c'_t, h_t &= o_t \cdot tanh(c_t) \end{aligned}$$

where x_t is the input at time step t, W_k are the weights of the input x_t for the respective gate k, U_k are the weights of the hidden state h_{t-1} for the respective gate k, b_k are the biases for the respective gate k, $\sigma(\cdot)$ is the sigmoid activation function, and $tanh(\cdot)$ is the tanh activation function. The activation function formulas are as follows.

$$\begin{aligned} \sigma(x) &= \frac{1}{1+e^{-x}} \\ tanh(x) &= \frac{e^x - e^{-x}}{e^x + e^{-x}}. \end{aligned}$$

The cell state can be interpreted as long-term memory while the hidden state can be interpreted as the short-term memory. The forget gate f_t measures the proportion of previous long-term memory c_{t-1} to be remembered. Similarly, the input gate i_t measures the proportion of potential long-term memory c'_t to be remembered. The new long-term memory c_t to be passed to the next layer is therefore the summation of the remembered previous long-term memory and potential long-term memory. Finally, the output gate o_t measures the proportion of potential short-term memory $\sigma(c_t)$ to be remembered. Multiplying o_t and $\sigma(c_t)$ thus produces the new short-term memory h_t to be passed to the next layer. In the output layer, the hidden state h_t is directly used as the output of the network.

To predict the anomalies, a prediction-based approach is implemented by employing the LSTM models to forecast a single time step using past values from a rolling window [2]. This means that for a fixed window length l, the first l spending are used to predict the $(l+1)$-th spending. Then, the second spending until the $(l+1)$-th spending data are used to predict the $(l+2)$-th spending. The process continues until the last spending data is forecasted. Significant deviations between predicted values and real values could then indicate an anomalous event. Additionally, as it is impossible to obtain the forecasts of the first l spending, they are assumed to be normal data.

As an extension to the ordinary LSTM model, we also implement an LSTM autoencoder model. An autoencoder first learns the critical characteristics of the input data and encodes it into a lower dimensional latent space. It then decodes the point in the latent space back to the data space to reconstruct the original input data [7]. The difference between the original and reconstructed data is called the reconstruction error. An autoencoder model is commonly used for anomaly detection as anomalous data is expected to produce high reconstruction error [2]. However, a common ground among research on anomaly detection by autoencoder is that the model is usually trained with only the normal data [9, 14, 11]. As it is impossible to separate the dataset of interest into normal and anomalous events, an autoencoder is not expected to perform well in this dataset.

3.2 Threshold Computation

As mentioned in the previous section, determining whether an observation is anomalous depends heavily on determining the threshold of deviation between forecasts and real values. Three methods of threshold selection are introduced, including the simple quantile threshold, the department-based threshold, and the dynamic threshold.

3.2.1 Simple Quantile Threshold

The simple quantile threshold computes the threshold according to the set quantile of the absolute errors between predicted and observed data. Let y be the observed data and $\hat{y}$ be the predicted data. Then, the absolute error for the i-th data point, AE_i, is

$$AE_i = |y_i - \hat{y}_i|, \quad i = 1, \ldots, n,$$

where n is the total number of observations. Let $AE = [AE_1, \ldots, AE_n]$ be the set of all absolute errors. The simple threshold $\tau_{simple}(q)$ is defined as the q-th quantile of the absolute error for $q \in [0, 1]$, that is

$$Pr(AE < \tau_{simple}(q)) \leq q \leq Pr(AE \leq \tau_{simple}(q)).$$

An observation y_i is thus considered an anomaly when its absolute error exceeds the threshold, i.e., $AE_i > \tau_{simple}(q)$. The parameter q is then empirically chosen such that it maximises the F1 score.

3.2.2 Department-based Quantile Threshold

Instead of fixing a single threshold for all observations, a threshold for each department is determined to take into account the variations between them. Denote y_i^k as the i-th observation in department k. Then, the corresponding absolute error is

$$AE_i^k = |y_i^k - \hat{y}_i^k|, \quad i = 1, \ldots, n, \quad k = 1, \ldots, m,$$

where m is the number of departments. Let $AE = [AE_1^1, \ldots, AE_n^m]$ be the set of all absolute errors. Define $AE^k = \{AE_i^j \in AE | j = k\}$ to be the set of all absolute errors for all observations belonging to department k. Thus, the threshold for department k, $\tau_{dep}^k(q_k)$, is the simple quantile threshold per department defined as

$$Pr(AE^k < \tau_{dep}^k(q_k)) \leq q_k \leq Pr(AE^k \leq \tau_{dep}^k(q_k)), \quad k = 1, \ldots, m.$$

An observation y_i^k is considered an anomaly when $AE_i^k > \tau_{dep}^k(q_k)$. The parameter $q_k \in [0, 1]$ is then empirically tuned to maximise the F1 score.

3.2.3 Dynamic Threshold

The final threshold computation method is the dynamic threshold [13]. Instead of setting a fixed threshold, the threshold is computed for each data point. This method allows us to take into account the trend and seasonality effects present in the time-series dataset. This also addresses the variability of behaviours between individuals. Assuming the model output is a vector of length Q, Let $\mathbf{h_L} \in \mathbb{R}^{Q \times 1}$ and $\mathbf{h_U} \in \mathbb{R}^{Q \times 1}$ be column vectors of the lower and upper bound. Then, the i-th observation $\mathbf{y}(i)$ is an outlier if $\mathbf{y}(i) < \mathbf{h_L}(i)$ or $\mathbf{y}(i) > \mathbf{h_U}(i)$ and not an outlier if $\mathbf{h_L}(i) \leq \mathbf{y}(i) \leq \mathbf{h_U}(i)$.

Let the output of a network in the j-th training step be $\mathbf{y}_{train}^{(j)} \in \mathbb{R}^{Q \times 1}, \quad j \in \{0, \ldots, J-1\}$ where J is the total number of training steps. Then,

$$\mathbf{h_U} = \overline{\mathbf{y}}_{train} + \beta \sqrt{var(\mathbf{y}_{train})}$$
$$\mathbf{h_L} = \overline{\mathbf{y}}_{train} - \beta \sqrt{var(\mathbf{y}_{train})},$$

where $\overline{\mathbf{y}}_{train}$ and $var(\mathbf{y}_{train})$ are the mean and variance of $\mathbf{y}_{train}^{(j)}$ with formulas as follows.

$$\overline{\mathbf{y}}_{train} = \frac{1}{J}\sum_{j} \mathbf{y}_{train}^{(j)}$$

$$var(\mathbf{y}_{train}) = \frac{1}{J}\sum_{j} ||\mathbf{y}_{train}^{(j)} - \overline{\mathbf{y}}_{train}||^2.$$

The parameter $\beta > 0$ reflects how many standard deviations away an observed data should be from the mean forecast to be considered an anomaly. A lower β value implies that more observations will be flagged as anomalies which may in turn produce too many false anomalies. In contrast, a higher value of beta implies that fewer observations will be flagged as anomalies, thus making it easier to miss real anomalies. The parameter can be empirically tuned to maximise the F1 score.

3.3 Post-processing

After the best model and threshold computation method have been chosen, predicted anomalous events undergo some post-processing to further refine model results. Reduction of the number of false anomalies is first attempted, followed by addressing the weekend effect.

3.3.1 False Positive Pruning

To reduce the number of false positives, a false positive pruning algorithm is run to differentiate between prediction errors caused by normal noise and real anomalies [16]. It can be reasonably assumed that errors caused by normal noise would tend to cluster around certain values within a certain period. On the other hand, errors caused by anomalies would be significantly larger and occur relatively apart from each other.

Let $E = [e^{(t)}, e^{(t+1)}, \ldots, e^{(Q)}]$ be the set of error values corresponding to forecasts $\mathbf{y}_{pred} = [\hat{y}^{(t)}, \hat{y}^{(t+1)}, \ldots, \hat{y}^{(Q)}]$. Then, let $E_A = \{e^{(t)} \in E | \hat{y}^{(t)} \text{ is an anomaly}\}$. $e^{(t_1)}$ and $e^{(t_2)}$ are both removed from E_A, i.e., both $y^{(t_1)}$ and $y^{(t_2)}$ are no longer considered an anomaly, if $|e^{(t_1)} - e^{(t_2)}| < p$ and $|t_1 - t_2| < t_\Delta$. The intuition behind this is that if two consecutive predicted anomalies have similar errors or occur very closely to each other, then it is probable that they are not anomalies but a new routine of an individual. The two parameters p and t_Δ will be empirically tuned to maximise the F1 score.

One cautionary note is that the pruning algorithm may remove some of the true anomalies despite careful selection of the parameters. Thus, this may cause deterioration in model performance. In such cases, the false positive pruning algorithm may be skipped.

3.3.2 The Weekend Effect

To address the weekend effect noted in section 2, we automatically flag a weekend activity as an anomaly given that it satisfies either one of these two conditions:

1. It is the first weekend activity
2. The duration between the current weekend activity and its preceding weekend activity is longer than 29 days

Condition one is placed because all first weekend activities are consistently flagged as anomalies in the observed dataset. Condition two is established by noting that most weekend activities that have passed a certain duration of time after their previous weekend activities are flagged as risk events. Thus, there exists a threshold such that a weekend activity is no longer considered a routine once the duration between the current and its preceding weekend activity has passed the threshold. The threshold of 29 days ($\approx$ 1 month) is decided based on data observation as this threshold applies for more than 86% of

the data (see Figure 6). Moreover, anomalous weekend activity conducted within less than 29 days from its previous weekend activity seems to be flagged due to reasons other than it simply being a weekend activity. The flags for weekend spending which satisfy the two conditions are stored in a new variable called `weekend_flag`

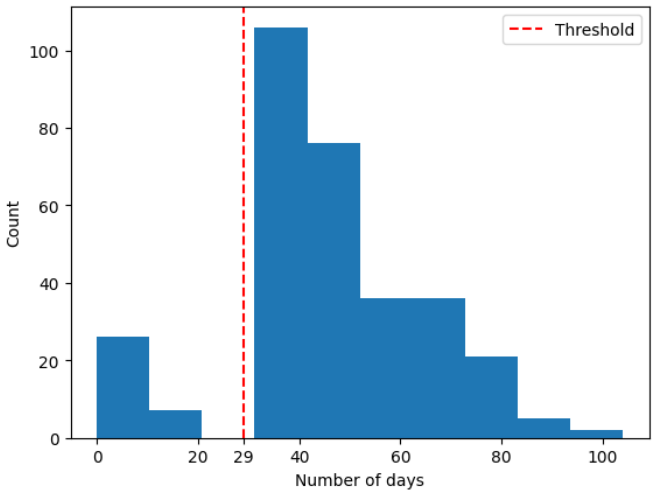

Figure 6: Histogram of the number of days between two consecutive anomalous weekend spendings

4 Results

All neural network models were implemented in Python based on TensorFlow and Keras models. The input is a matrix of shape (batch size, window size, attribute size). Batch size is the number of sequences in one epoch. Window size is the number of past observations used for one-step forecasting. In the LSTM autoencoder model, it is the number of records in one sequence. A fixed window size of 5 is implemented for all models. Finally, attribute size is the number of features in a batch.

For the classic LSTM models, 2 LSTM layers with 5 neurons each and a drop-out layer with a rate of 0.2 between the two LSTM layers to prevent overfitting were constructed. For the multivariate LSTM models, 2 LSTM layers with 25 neurons each and a drop-out layer with a rate of 0.2 between the two LSTM layers were used. Five features were considered for each batch, including spending data (`spend`), whether spending occurs between 6:00 and 18:00 (`work_hour`), whether spending occurs during the weekend (`weekend`), whether a weekend activity is flagged as anomaly based on the two criteria described in section 3.3.2 (`weekend_flag`) and the time difference between two consecutive spendings (`time_diff`). Finally, for the LSTM autoencoder model, 2 LSTM layers in both the encoder in decoder part were applied with 5 neurons and 2 neurons in each layer. A drop-out layer with a rate of 0.2 after every LSTM layer was also included.

4.1 Risk Event Target

Table 4 summarizes the F1 score for the risk event target with the three threshold computation methods, which are the simple quantile, the department-based quantile, and the dynamic threshold. Highlighted cells indicate the highest F1 score to note the best model for each threshold computation method.

Results show that for the risk event target, the univariate LSTM model with dynamic threshold is the best model and threshold computation method. Fixing $\beta = 304$ for the dynamic threshold, it produced the maximum F1 scores of 31.59% for training data and 31.08% for testing data, almost double the F1 scores of the baseline model. Out of the four models, LSTM Autoencoder performed the worst even compared

to the baseline model. As emphasised in Section 3.1, the LSTM autoencoder does not perform well for anomaly detection when it is trained on contaminated data. This is because an autoencoder will assume the training data as normal behaviour. Thus, when anomalies are fed into the model, an autoencoder will instead learn the anomalous events as normal events. Hence, when it tries to reconstruct the original data, it will fail to identify the anomalies. Additionally, the multivariate LSTM also unexpectedly performed worse than the univariate LSTM in most cases despite having more information in the training data. One possible explanation is that the multivariate model was unable to differentiate between normal data and anomalous data during training as it was trained with more complete data.

Out of the three threshold computation methods, the dynamic threshold was shown to be the most superior compared to both the simple quantile and the department-based quantile method. Consistent improvements can be seen from applying the department-based quantile method when compared to the fixed threshold set by the simple quantile method. This reflects the effect of one's department on spending behaviour and thus, each department shall have a different anomaly threshold. Nevertheless, computing the threshold for each forecast by applying the dynamic threshold method better captures the trend and seasonality of each spending series. Thus, it managed to consistently add an additional 10% to the F1 score from the department-based quantile method for all LSTM models.

Model	Threshold Computation Methods					
	Simple Quantile		Department-based Quantile		Dynamic	
	Train	Test	Train	Test	Train	Test
Persistence Model	16.45%	17.86%				
Univariate LSTM	**20.05%**	**19.79%**	21.56%	**20.55%**	**31.59%**	**31.08%**
Multivariate LSTM	19.75%	19.71%	**21.64%**	20.03%	28.78%	28.86%
LSTM Autoencoder	13.33%	14.73%	13.38%	14.40%	24.86%	23.46%

Table 4: F1 scores for risk event target of baseline and LSTM models with three different threshold computation methods

Post-processing was then conducted on anomaly flag predictions from the best threshold computation method, which is the dynamic threshold method. The updated F1 scores for all three LSTM models are shown in Table 5. The highest F1 scores are highlighted in yellow. Implementation of the false positive pruning algorithm seems to offer consistent yet very slight improvement. Empirical tuning showed that for the univariate LSTM model, consecutive predicted anomalies are false anomalies if either the absolute error between them is less than 0.14 or they occurred within less than 8 days. Further addition of the weekend effect flags substantially increased the F1 score by almost 2.5 times, achieving the highest F1 score of 77.40% for the training data and 74.46% for the testing data. The corresponding precision rate of 87.19% and recall rate of 69.56% were also measured. This implies that the univariate LSTM model can sufficiently detect anomalies which are not due to the weekend effect. For the remaining weekend anomalies, the algorithm proposed in Section 3.3.2 was able to identify most of them. The combination of the univariate LSTM model with dynamic threshold and post-processing to reduce false anomalies and address the weekend effect proves to be capable of reliably detecting most of the anomalies while not generating excessive false alarms.

Figure 7 illustrates the detected, falsely predicted, and undetected risk events from the univariate LSTM model with the dynamic threshold and post-processing for individuals 1, 100, and 1734. It can be observed that most anomalies were able to be detected with few false anomalies, reflecting the high F1 scores of the model. However, there is a tendency for peaks to be identified as false anomalies. This may be due to the relatively simple false positive pruning algorithm which was not able to differentiate

Model	Post-processing			
	False Positive Pruning		Weekend Effect	
	Train	Test	Train	Test
Univariate LSTM	31.64%	31.08%	**77.40%**	**74.46%**
Multivariate LSTM	28.84%	28.86%	75.01%	71.37%
LSTM Autoencoder	25.29%	23.54%	74.68%	70.94%

Table 5: F1 scores for risk event target of LSTM models after post-processing

between real and false anomalies for isolated points.

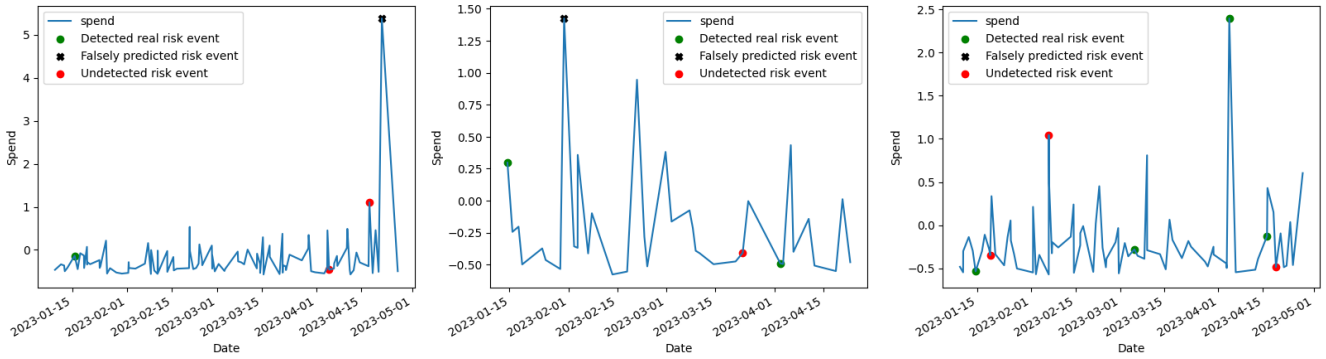

Figure 7: Risk event detection results with univariate LSTM model with dynamic threshold and post-processing for individuals 1 (left), 100 (middle), and 1734 (right)

4.2 Risk Behaviour Window Target

Table 6 presents the F1 scores for the risk behaviour window target with the three threshold computation methods for the baseline model and three LSTM models. Overall, the F1 scores are significantly lower than those for the risk event target. This implies that all of the proposed models were having much more difficulty in identifying risk windows compared to detecting risky spending. This may be due to a risk behaviour window being inherently more complicated than a risk event. While a risk event is a single anomalous spending entry, a risk behaviour window is a sequence of entries with a more preemptive nature rather than instantaneous as it indicates whether an individual has started to exhibit risky behaviour. This implies that even a sequence of spending which may seem normal as a whole may comprise a risk behaviour window for it may exhibit the characteristics of potential risky spending. Thus, conventional LSTM models proposed in this report may be insufficient in predicting risk behaviour window.

Nevertheless, results were similar with the risk event target where the univariate LSTM with the dynamic threshold performed the best in most of the threshold computation methods. LSTM autoencoder still performed the worst for all thresholds while multivariate LSTM was shown to perform the best on the testing set with the simple quantile and the dynamic threshold with F1 scores of 14.03% and 16.87%. In general, the dynamic threshold achieved the highest F1 scores across all three LSTM models and thus, shall be chosen as the best threshold computation method.

To choose the best LSTM model, post-processing, which includes false positive pruning and weekend effect, was conducted on the predictions based on the dynamic threshold. Table 7 reveals the F1 score of all three LSTM models following post-processing. Note that the false positive pruning algorithm worsened the F1 scores of univariate LSTM. This may be caused by the algorithm pruning too many real anomalies. Thus, the false positive pruning step was skipped for the univariate LSTM model. Additionally, the

Model	Threshold Computation Methods					
	Simple Quantile		Department-based Quantile		Dynamic	
	Train	Test	Train	Test	Train	Test
Persistence Model	8.26%	8.39%				
Univariate LSTM	**12.26%**	13.24%	**12.41%**	**14.22%**	**17.41%**	15.10%
Multivariate LSTM	11.46%	**14.03%**	12.34%	13.50%	15.98%	**16.87%**
LSTM Autoencoder	6.98%	8.78%	7.72%	9.16%	11.61%	12.78%

Table 6: F1 scores for risk behaviour window target of baseline and LSTM models with three different threshold computation methods

addition of the weekend effect anomalies only increased the F1 score marginally by approximately 0-5%, unlike the risk event target where it was able to increase the F1 scores by almost 2.5 times. This discovery implies that the weekend effect does not play as huge of a role in determining risk behaviour window as in determining risk events.

Nevertheless, results seem to follow the previous trend where multivariate LSTM performed better in the training dataset and LSTM autoencoder performed the worst. However, while multivariate LSTM performed better in the testing dataset compared to the univariate LSTM with the dynamic threshold, the improvement in the F1 score is minimal of only 0.42% higher. Furthermore, the univariate LSTM model still performed better in the training dataset. Setting $\beta = 368$ for the dynamic threshold, multivariate LSTM achieved an F1 score of 16.76% for the training set while the univariate LSTM managed to achieve an F1 score of 17.64%, 0.88% higher than that of the multivariate LSTM. Therefore, it is believed that the more parsimonious univariate LSTM model with dynamic threshold and post-processing to address the weekend effect is the best model for identifying risk behaviour windows.

Model	Post-processing			
	False Positive Pruning		Weekend Effect	
	Train	Test	Train	Test
Univariate LSTM	15.10%	15.10%	**17.64%**	18.45%
Multivariate LSTM	16.87%	16.87%	16.76%	**18.87%**
LSTM Autoencoder	12.78%	12.78%	15.29%	17.49%

Table 7: F1 scores for risk behaviour window target of LSTM models after post-processing

Figure 8 illustrates the univariate LSTM models and relevant post-processing results in predicting the risk behaviour windows for individuals 1, 100, and 1734. The existence of many falsely predicted risk behaviour windows and undetected risk behaviour windows reflects the relatively low F1 score of the model. Nonetheless, the model still managed to successfully identify some risk behaviour windows as indicated by the green dots.

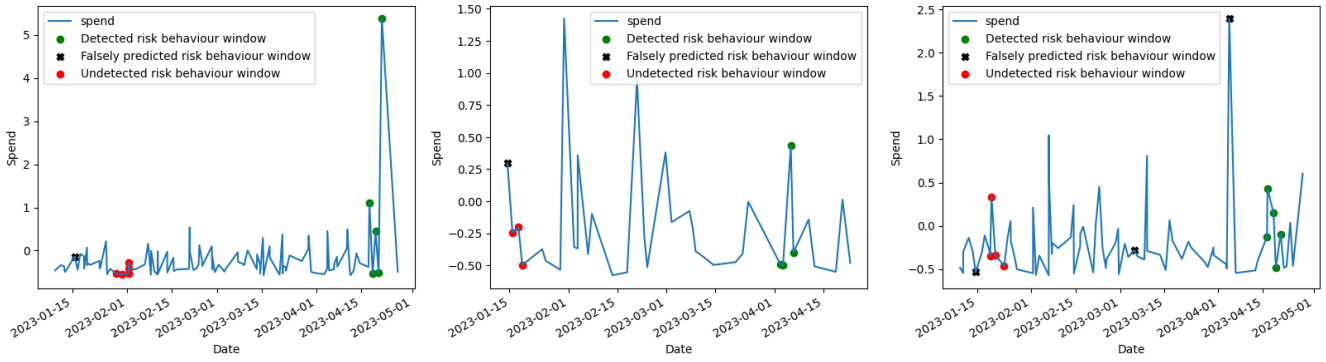

Figure 8: Risk behaviour window detection results with univariate LSTM model with dynamic threshold and weekend effect post-processing for individuals 1 (left), 100 (middle), and 1734 (right)

5 Conclusions

The objective of this report was to develop an anomaly detection model to identify risk events and risk behaviour windows from spending data of LBG employees. Results suggest that LBG is recommended to implement the univariate LSTM model with dynamic threshold and appropriate post-processing for both detecting risk events and risk behaviour windows. For the risk event target, conducting post-processing model predictions by false positive pruning and taking into account the weekend effect produced the highest F1 score of 77.40% for the training dataset and 74.46% for the testing dataset. In particular, the univariate LSTM model with the dynamic threshold was shown to be capable of sufficiently detecting non-weekend effect anomalies while the post-processing algorithm was able to detect most of the weekend effect anomalies as shown by the approximately 2.5 times increase in F1 scores.

For the risk behaviour window target, post-processing was utilized only to address the weekend effect as the false positive pruning algorithm was shown to be detrimental towards model performance. Post-processing offered slight improvement to the model performance which resulted in the F1 scores of 17.64% and 18.45% for the training and testing dataset, respectively. The relatively lower F1 score of the model in detecting risk behaviour window compared to detecting risk events reflects its more complicated nature which is harder to be captured by the model.

There are, however, some limitations to the presented methods. Firstly, there is no analytical way to determine the fixed length of the rolling window. A larger window size provides more data and smaller variance but it is not as sensitive to short-term fluctuations. A smaller window size, on the other hand, is capable of capturing short-term changes but it is more sensitive to noise. Another limitation comes from the nature of rolling window analysis where for a fixed window length l, the first l forecasts of spending data are unattainable. Additionally, individuals are required to have at least $(l+1)$ spending data for the model to be applicable. Naturally, the proposed model will not be able to be employed for detecting risky behaviour of a new employee until there is at least $(l+1)$ spending data. Finally, the proposed model ignores the irregular sampling period of the data. Although the implemented univariate LSTM model can address the sequential nature of the time series, it may not be able to fully capture the temporal irregularity of the data.

Future research could investigate more advanced time-series models than the proposed classical models. For example, conditional LSTM could be explored to deal with static features as an extension of the multivariate LSTM. More advanced false positive pruning algorithms which can tackle isolated points may also be of interest. Additionally, further examination of curve-fitting approaches could develop deeper insights into irregularly spaced time series data. Gaussian process regression, for example, could be explored as it models the correlation between two data points depending on their distance in time with kernel functions that represent trends, periodicity, and seasonality.

References

[1] C. C. Aggarwal and C. C. Aggarwal. *An introduction to outlier analysis*. Springer, 2017.

[2] 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), 2023.

[3] E. Erdogan, S. Ma, A. Beygelzimer, and I. Rish. Statistical models for unequally spaced time series. In *proceedings of the 2005 SIAM International conference on data mining*, pages 626–630. SIAM, 2005.

[4] N. Gröwe-Kuska and W. Römisch. *Stochastic unit commitment in hydro-thermal power production planning*. Preprints aus dem Institut für Mathematik. Humboldt-Universität zu Berlin, Institut für Mathematik, 2001.

[5] S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 11 1997.

[6] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 387–395, 2018.

[7] V. Lempitsky. Autoencoder. *Computer Vision: A Reference Guide*, pages 1–6, 2019.

[8] C. Liu. Long short-term memory (lstm)-based news classification model. *Plos one*, 19(5):e0301835, 2024.

[9] P. Liu, X. Sun, Y. Han, Z. He, W. Zhang, and C. Wu. Arrhythmia classification of lstm autoencoder based on time series anomaly detection. *Biomedical Signal Processing and Control*, 71:103228, 2022.

[10] Lloyds Banking Group. Lloyds banking group - about us. `https://www.lloydsbankinggroup.com/who-we-are/our-strategy.html`, 2024. [Online; accessed 18-June-2024].

[11] S. Maleki, S. Maleki, and N. R. Jennings. Unsupervised anomaly detection with lstm autoencoders using statistical data-filtering. *Applied Soft Computing*, 108:107443, 2021.

[12] P. Malhotra, L. Vig, G. Shroff, P. Agarwal, et al. Long short term memory networks for anomaly detection in time series. In *Esann*, volume 2015, page 89, 2015.

[13] R. Mo, Y. Pei, N. Venkatarayalu, P. Nathaniel, A. B. Premkumar, and S. Sun. An unsupervised tcn-based outlier detection for time series with seasonality and trend. In *2021 IEEE VTS 17th Asia Pacific Wireless Communications Symposium (APWCS)*, pages 1–5. IEEE, 2021.

[14] M. Said Elsayed, N.-A. Le-Khac, S. Dev, and A. D. Jurcut. Network anomaly detection using lstm based autoencoder. In *Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks*, pages 37–45, 2020.

[15] T. Shiina and J. R. Birge. Stochastic unit commitment problem. *International Transactions in Operational Research*, 11(1):19–32, 2004.

[16] Y. Wang, X. Du, Z. Lu, Q. Duan, and J. Wu. Improved lstm-based time-series anomaly detection in rail transit operation environments. *IEEE Transactions on Industrial Informatics*, 18(12):9027–9036, 2022.

[17] P. B. Weerakody, K. W. Wong, G. Wang, and W. Ela. A review of irregular time series data handling with gated recurrent neural networks. *Neurocomputing*, 441:161–178, 2021.

Appendices

A Word count

Word count of the main text: 4868

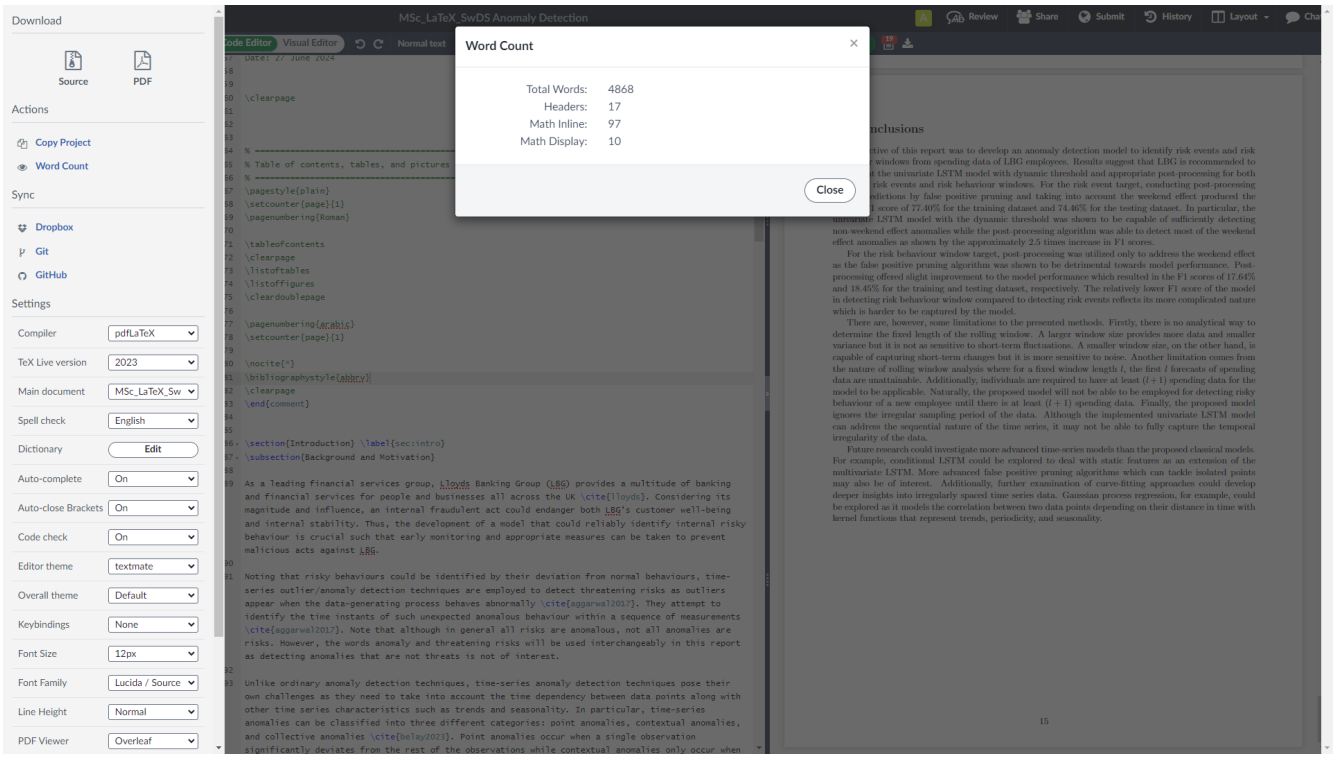

Figure 9: Screenshot of LaTeX word count of the main text