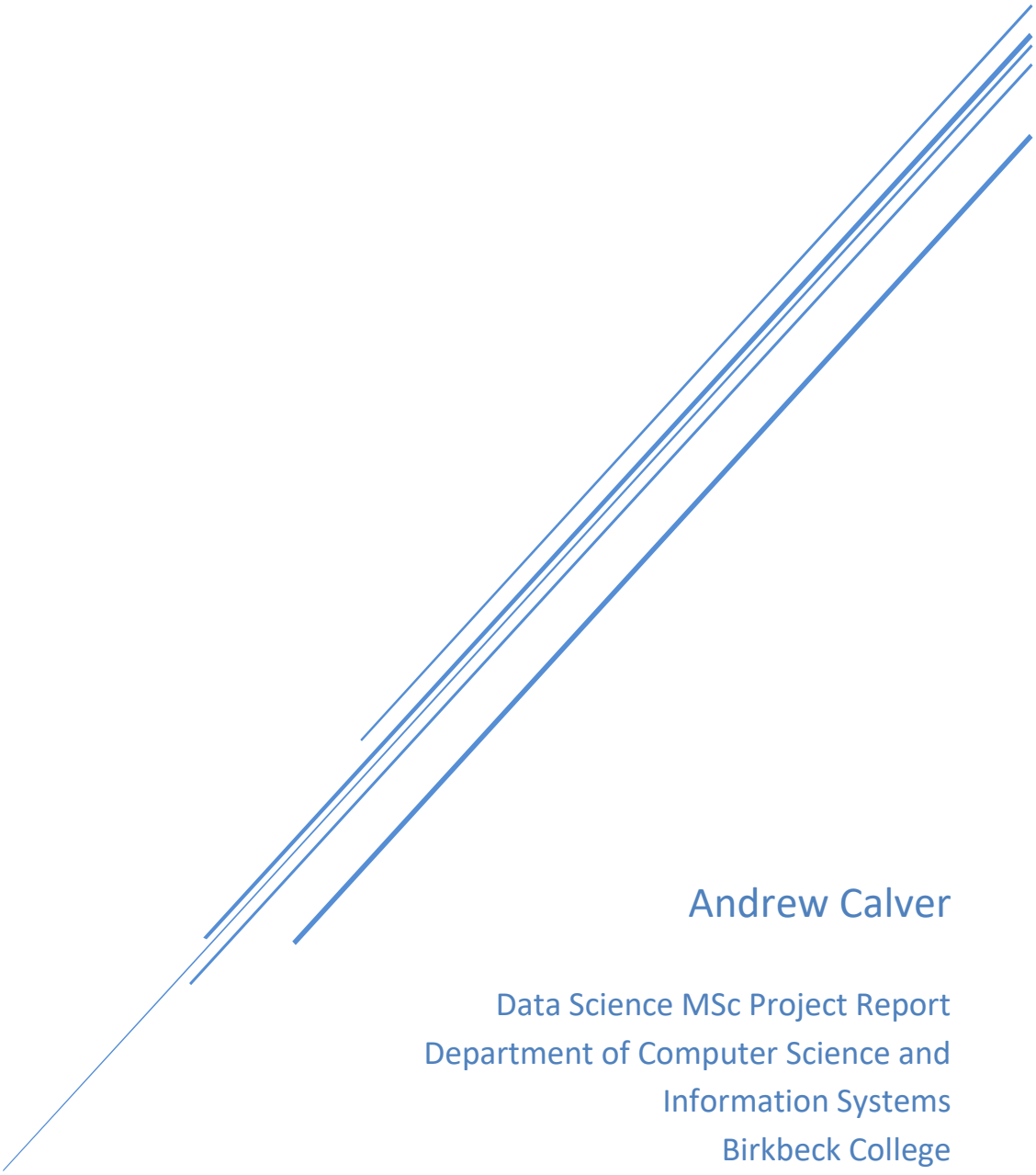
NEEDLE IN A HAYSTACK

Using deep learning to detect outliers in regulatory insurance data



Andrew Calver

Data Science MSc Project Report

Department of Computer Science and Information Systems
Birkbeck College
University of London

2021

Abstract

Solvency II is a regulatory framework covering the insurance industry, with firms required to report data both quarterly and annually. It is necessary that the data submitted by these firms be assessed every reporting period to ensure the quality of data is maintained for analysis, and outliers that are detected queried. This project aimed to apply machine-learning algorithms on time series of Solvency II's 'Balance Sheet' reporting template in order to automate the process of detecting outliers in the most recent reported figures.

An autoencoder was trained on the data to gain a reduced feature set. These features were joined to the original data and a Long Short Term Memory network was trained on each firm. This project explored two variations of this method in the goal of outlier detection. Firstly, an autoencoder was trained on the whole dataset and the LSTM network run on each firm. Secondly, the data was clustered using Dynamic Time Warping and separate autoencoders trained on the clusters before a LSTM was trained on the data.

This project demonstrates that the benefit of using an autoencoder (on the whole dataset or on reduced clusters) to complement Long Short Term Memory networks for the purpose of outlier detection is extremely limited.

Academic declaration

This report is substantially the result of my own work, expressed in my own words, except where explicitly indicated in the text. I have read and understood the sections on plagiarism in the Programme Handbook and the College website. I give my permission for it to be submitted to the JISC Plagiarism Detection Service.

The report may be freely copied and distributed provided the source is explicitly acknowledged.

Table of Contents

Introduction

Solvency II

Solvency II (SII) is a risk-based regulatory framework that covers insurance firms across Europe. Weaknesses and shortcomings were identified in the prior regulatory structure of European supervision of insurance firms as early as 1999, and the required improvements became all the more urgent in the wake of the 2007 financial crisis [1]. SII was adopted by European insurers in 2016 and is based on a three-pillar framework:

I. Quantitative requirements

Firms are required to calculate their Solvency Capital Requirement – the total number of funds the firms are required to hold. This is calculated using a 99.5% value-at-risk over the period of a year i.e. "enough capital must be held to cover the market-consistent losses that may occur over the next year with a confidence level of 99.5%, resulting from changes in market values of assets held by insurers" [2].

II. Governance of the undertaking and supervisory activity

"Insurers are required to identify, measure, monitor, manage and report risks they're exposed to. Insurers must put risk management at the heart of decision-making, and are required to conduct an Own Risk and Solvency Assessment" [3].

III. Supervisory reporting and public disclosure [1]

This specifies the nature of SII supervisory reporting and outlines what needs to be reported publically by insurance firms, financial or otherwise.

Pillar III - Solvency II reporting

As of October 2020, there were 1253 authorised insurers in the UK (this includes both firms registered in the UK and firms registered elsewhere in the EU) [4]. Firms submit their data to the Bank of England, however not all authorised firms are required to report SII. Insurance firms must report Solvency II unless they "are not part of a group and write less than €5 million in premiums per year" [2].

Three main types of insurer are required to report SII:

- Solo firms (includes UK registered insurers and foreign insurers with branches in the UK)
- Group firms (firms that cover subsidiaries of multiple solo insurers – Aviva PLC includes Aviva Life, Aviva Insurance and Aviva Investors in the UK [5])
- Lloyd's syndicates (Lloyd's members joining together to provide capital and accept insurance risks [6])

For the purposes of this project, only Solo insurers and Syndicates will be included as Group firms include data from their subsidiaries. Not including Group firms removed the risk of some duplicate data.

The structure of a SII report submitted by a firm is made up of various templates which contain data from the firm. For example, S.02 'Balance Sheet' contains an overview of the assets and liabilities of

the firm; S.12 and S.17 contain more detailed data on the life and non-life technical provisions (respectively) of the firm.

The data is submitted by firms in XBRL (eXtensible Business Reporting Language) format meaning the data is submitted in a consistent and reliable format. The data is also subject to a collection of taxonomy validations published by EIOPA [7] (European Insurance and Occupational Pensions Authority, the overseer of the SII regulatory framework). The validations ensure the SII reports submitted by firms are consistent and will block the submissions if any key validations fail – for example, do the asset rows on S.02 sum to the total 'Total Assets'. The validations mean confidence can be had in the consistency and quality of the data for each period a firm submits. Although the validations do not cover every aspect in which a report needs to be consistent, it means the largest data quality question is: are movements in the latest period plausible given the historic time series.

Reporting schedule

There are five reporting periods in a year, four quarterly periods and one annual. The quarterly SII reports are a subset of the annual SII report, with some templates specific to only the annual template – see Annex.

If a firm is small enough, it can apply to be exempted from some (or all) quarterly reporting [8] – this is referred to as having a 'waiver'. This means there is variation in the number of time series available for the population of firms. For example, a firm without any waivers would have 25 periods in its time series set – 5 periods for 5 years from 2016. However, a firm with a waiver for all quarterly reporting would only have 5 periods in its time series set – 1 reported period annually for 5 years.

Different financial year ends

Although the majority of firms have their financial year-end on 31st December, there are financial year-ends all through the calendar year from February to November [9]. It is important when analysing SII data this variation is taken into account as the default, alphabetical time series ordering in Python will mean some firms have time series that are not correctly ordered. This has an impact on one of the machine learning algorithms, discussed in more detail below.

Plausibility checking and outlier detection

Ensuring the quality of Solvency II data is extremely important. SII data is used in a variety of ways, and the integrity and quality of that data is important to stakeholders who use the raw data, and for aggregate analysis. The data is used in two main ways:

1. Supervision analysis

Solvency II data is used to supervise insurance firms on an individual firm basis, and on an aggregate basis looking at the situation of the sector as a whole. Time series for individual firms are analysed to ensure the continued financial stability of a firm – if a firm fails to fulfil their Quantitative requirements (Solvency II Pillar I), this could indicate financial stress. The time series of the quantitative requirements gives key information on a specific firm. Additionally, aggregate analysis is performed on groups of firms to detect general trends in sectors and indicate whether there might be concerns in certain sectors. This analysis can only be meaningful if the data is of high quality. Plausibility checking the data is vital to ensure confidence in the analyses can be high.

2. External stakeholders

Solvency II data is not only used by the Bank of England. External stakeholders use the data for further analysis, such as the Office for National Statistics [10] or the Organisation for Economic Co-operation and Development [11]. In order for these outputs to be accurate, the underlying data must be high quality. As with supervision analysis, plausibility checking the received SII data is important to ensure the reliability of outputs from external stakeholders.

Project motivation

Current outlier detection method

The current process of outlier detection involves an analyst manually looking at time series and judging whether a particular data point is potentially an error or not. There are a number of issues with this method:

1. Time intensive

Manually checking time series is a laborious process that takes a significant amount of analysts' time. Using S.02 as an example, there are 80 different data points on that template, therefore there are 80 time series that have to be assessed. Most firms will report 5/6 of the required quarterly templates (most firms either report a life insurance template or a non-life insurance template as most firms write one type of business or the other). Extrapolating S.02, 80 time series across 5 different templates means 300 time series for each firm. These are very rough estimates but demonstrate the volume of data that is required to be checked. The time requirement to check all this data is significant and the current process of manually checking this data is laborious.

2. Very low proportion of outliers

The proportion of data points queried as outliers is less than 1%, which means the vast majority of time is spent assessing time series which do not need to be queried. This is not an efficient use of time.

3. Variability of analysts behaviour

With the total number of SII reports shared across a team of analysts, some of the outlier decisions can be subjective. If a particular data point lies on the border of being an outlier or not, it is up to the discretion of the analyst whether to investigate further.

Purpose of this project

The goal of this project was to resolve the above three issues with the current outlier detection process. This project aimed to create a machine learning pipeline in order to: i) improve the time efficiency of routine Solvency II outlier detection, ii) highlight outliers so only those time series in question need to be assessed by analysts, and iii) create an impartial model that assesses each firm in a methodical, identical way.

Project methodology

For this project, data from SII template S.02 'Balance Sheet' was chosen because it is reported both quarterly and annually. It also gives an overview of the firm's business and is an important template in the context of plausibility checking the data, so was an appropriate choice to test this project on. The training data spanned 25 reporting periods: from 2016 quarter one to 2020 year-end. The test period was 2021 quarter one.

This project looked at two potential methods for a combining an autoencoder with a Long Short Term Memory neural network in order to automate and add impartiality to the outlier detection of Solvency II data.

The first method used all the training data to train an autoencoder, join the generated columns to the original features, deselect generated features that had high correlation with other features (generated or original) and then run a LSTM network for each firm. The predicted values for the test period were assessed against the metrics of the relative importance of a feature and the standard deviation of the past time series to assess if any outliers were present in the time series for a firm.

The second method looked at splitting the population of firms into two clusters using Dynamic Time Warping, on the hypothesis the autoencoder could perform better when trained on closer related data as opposed to the whole dataset. Once the data was split, the pipeline followed the same process as the first method, with a separate autoencoder being trained on each cluster and then the LSTM network run for each firm and outliers assessed using the same method.

Overall results

The results of this project demonstrate neither method constitute a viable outlier detection system. While there is correlation between the number of SII outliers and autoencoder generated feature outliers, there appears to be no distinction between an outlier firm and a non-outlier firm. The model appears to be overfitting the training data, with large differences between the training mean square error and the test mean square error, which could account for the poor performance in distinguishing outlier firms from non-outlier firms.

This complete lack of differentiation between outlier classes and probable model overfitting renders these pipelines of extremely little benefit for outlier detection.

Machine Learning Technologies

Autoencoder

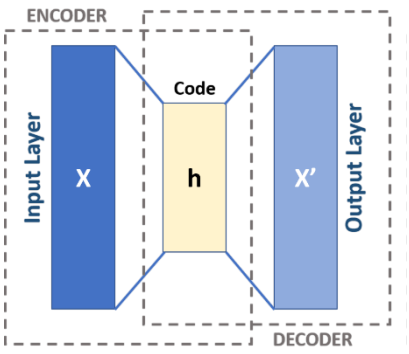


***Figure 1**: basic structure of an autoencoder*

An autoencoder is a type of neural network that aims to compress its input data into a smaller number of features (called latent variables) by extracting key information and removing potential noise within the dataset. It does this by compressing the input data into a smaller representation, and then tries to replicate the original data from the compressed layer (Figure 1[1]). By comparing the constructed data to the input data, the autoencoder can assess how successful its compression was in extracting the relevant information of the data and ignoring noise. The difference between the input and output is measured by loss and the aim of the autoencoder is to minimise the loss over a number of iterations.

There are different possible structures of autoencoder. A stacked autoencoder is an autoencoder with multiple layers, with the compressed layer of the first autoencoder providing the input layer to a second autoencoder. A sparse autoencoder has a certain number of neurons 'deactivated' during the learning process, meaning the active neurons are more conditioned to learn useful features of the data and not overfit the training data. As discussed below, the hyperparameter showed a sparse autoencoder with a single layer performed the best.

An autoencoder was chosen for experimentation in this project because it is able to learn non-linear relationships between the data and extract the key features and remove noise [12]. Within the template of S.02 it is highly likely there could be relationships between some of the data points – for example, Government Bonds could be correlated (positively or negatively) with Corporate Bonds or life technical provisions correlated with health technical provisions[2]. An autoencoder was used in this project to assess the strength of this hypothesis and assess the extent it could shrink the 62 'lowest-level' features on S.02 into a smaller subset, compressing any features that may have a correlation. A final outlier detection model might have higher performance on this subset than on the original features.

The implementation of the autoencoder was performed using the Keras API [13], which allows for flexible and robust functionality.

Autoencoder hyperparameter tuning

In order to achieve the best possible configuration of the autoencoder hyperparameters, the hyperparameters needed to be tuned. Two different comparisons were made: a stacked autoencoder vs a single layered autoencoder, and once this was determined a grid search of hyperparameters.

[1] Image from https://en.wikipedia.org/wiki/Autoencoder
[2] The amount that an insurer requires to fulfil its insurance obligations and settle all expected commitments to policyholders

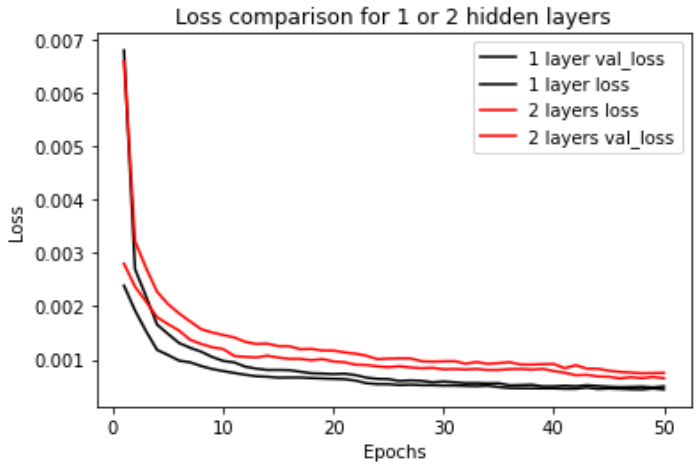


***Figure 2**: loss comparison between a single-layered and stacked autoencoder*

A stacked autoencoder with two hidden layers was compared to a single layer autoencoder, with a single layered autoencoder demonstrating both a lower loss and a lower validation loss function - Figure 2. The single-layered autoencoder showed better performance so was chosen.

The hyperparameters of the single autoencoder were tuned using sklearn's GridSearchCV [14] function which is compatible with the keras API using the wrapper KerasRegressor. The parameters tested were:

- Core size (the reduced number of features into which the input features are compressed)
- Level of dropout (proportion of nodes to randomly deactivate, preventing the model overfitting)
- Activation function (what function to apply to the output)
- Number of epochs (number of training iterations)

The results of the tuning are shown below in Table 1.

Hyperparameter	**Chosen Value**
Core size	40
Dropout	0.1
Activation function	Tanh
Epochs	80

***Table 1**: chosen hyperparameters for the autoencoder*

In practice, although the grid search determined epoch 80 as the epoch which produced the lowest loss value, Figure 2 shows strong convergence by epoch 40. In order to prevent the autoencoder overfitting the data, the autoencoder epoch value was set at 40 for the pipeline.

Long Short Term Memory network

A Long Short Term Memory (LSTM) network is a type of recurrent neural network that specialises in predicting sequential data. It is more suited than other neural networks to time series analysis because of its ability to learn long-term dependencies [15]. LSTM networks can use information from past time steps to better inform predictions at the current time step. They achieve this by having a cell state that runs from neuron to neuron, and this cell state determines what the output is of each neuron. Each LSTM neuron has three gates: input, forget, and output. The input and forget gates (along with the previous cell state) determine what the cell state is for that neuron – figure 3[3]. The input gate adds the data from the current time step and the forget gate decides what information

[3] Image from Towards Data Science: https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

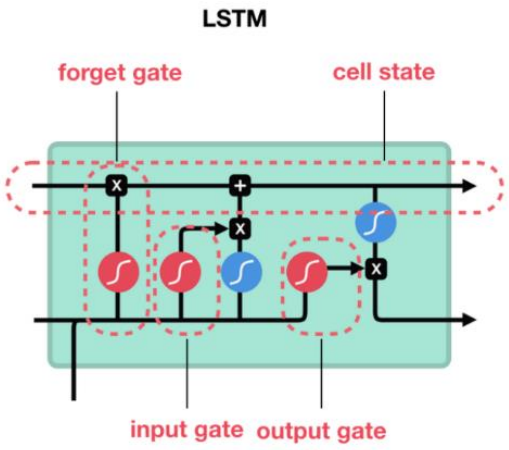


***Figure 3**: structure of a LSTM cell*

can be kept from previous steps, and what can be discarded. The output gate determines the hidden state for the next neuron. It is the forget gate that allows LSTM cells to assess long-term dependencies and recall relevant data from previous steps to better make predictions for current and future steps.

A LSTM network was chosen as the prediction method for this project because of its ability to remember past data in order to influence its predictions for the present, as well as its documented suitability for financial time series [16, 17]. When plausibility checking financial time series data, it is not enough to assess the movement from the previous time period alone. The historical movements of a time series are a vital factor in determining whether the current movement is an outlier. For example, a volatile time series can expect large movements and a large movement in the test period should not be flagged as an outlier. On the other hand, very stable time series should not have large movements in the test period. The history of time series is vital in determining outliers in the present, and the ability of LSTM networks to use past instances to predict the present was the reason it was chosen.

The implementation of the Long Short Term Memory network also utilised the Keras API.

LSTM hyperparameter tuning

The hyperparameter selection was much more challenging than the hyperparameter selection of the autoencoder. The GridSearch method used for the autoencoder was not appropriate for time series data, as the data for a time period is dependent on the data in previous time periods. The cross-validation would have used random time periods as the validation time periods and assessed them out of sequence. Therefore, to find the hyperparameters a 'brute force' method using a number of nested loops was used, with the selected hyperparameters producing the lowest loss values.

Because a fresh LSTM network was constructed for each firm in the model, tuning hyperparameters for the whole population of firms was extremely impractical. To balance tuning hyperparameters with realistic running expectations, 4 sample firms were chosen based on size of firm and business (a life or non-life firm). The tuning was performed on each of the firms with the chosen parameters for each firm being compared to the chosen parameters of the other sample firms. The results of the tuning are shown below in Table 2.

Hyperparameter	Chosen Value	All sample firms agree
Number of nodes	100	No
Activation function	Tanh	Yes
Recurrent activation function	Tanh	Yes
Dropout	0	Yes

***Table 2**: hyperparameter tuning results for the LSTM network*

The sample firms had consistent preferences for three out of the four hyperparameters tested. The hyperparameter without consensus on initial testing was the number of nodes. However, as figure 4

shows, the number of nodes was insignificant in the resulting loss function of the LSTM networks. 100 was chosen as the final number of nodes.

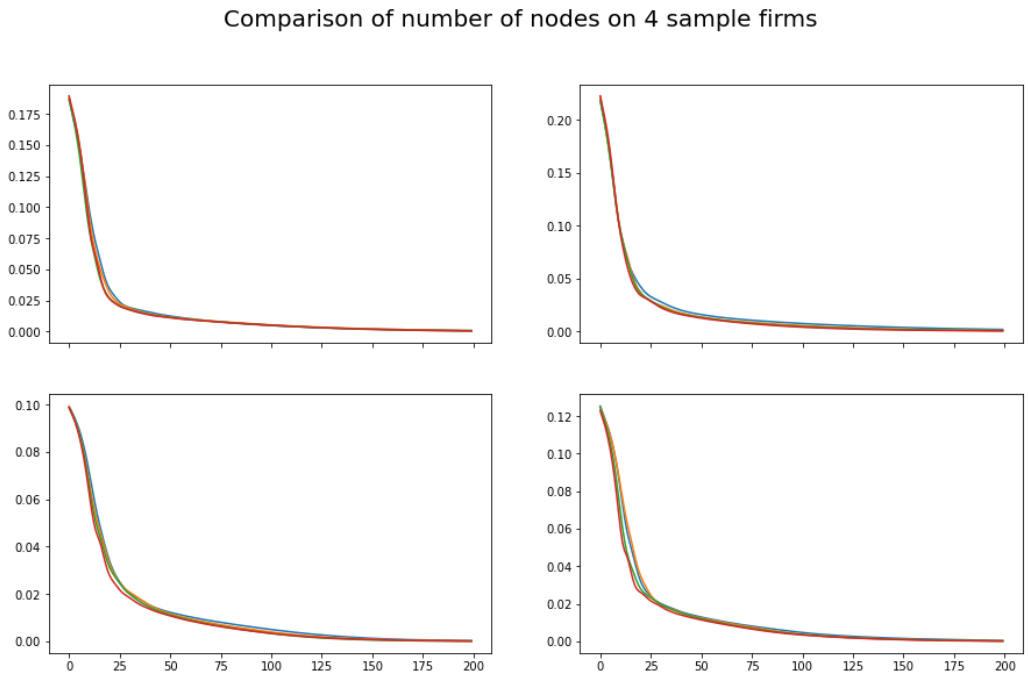


***Figure 4**: variation of loss when training a LSTM network with 90, 100, 110 and 120 nodes over 200 epochs. As demonstrated, the number of nodes chosen is irrelevant, all converge to the same value*

As discussed in the Project Proposal, the input data for this project is confidential since it is financial data of insurance firms. Therefore, it was not possible to use an external data processing server like Google Colab to speed up the data processing. The whole pipeline was run on an internal company machine, which had a large impact on the running time of the algorithms. The hyperparameter tuning of the LSTM model was particularly time inefficient and caused significant running issues, hence the choice of tuning the LSTM network on only 4 sample firms.

Clustering & Dynamic Time Warping

Clustering is the umbrella term for a family of machine learning algorithms that group data based on a measure of similarity. A common measure of similarity of Euclidean distance – the distance between points indicates its similarity / dissimilarity. Based on the distances between the points, points are grouped into clusters that are close together [18]. Time series can also be assessed in this way, with the Euclidean distance being measured on a point-by-point basis and the similarity returned from the whole time series [19]. However, the problem with measuring time series similarity with Euclidean distance is the lack of flexibility. Euclidean distance compares on a point-by-point basis. Time series could realistically have very similar shapes but if the time points didn't match, it could receive an erroneous low similarity score – figure 5. It is not a flexible algorithm for measuring time series that could have the same shape (and therefore be similar), but be different lengths.

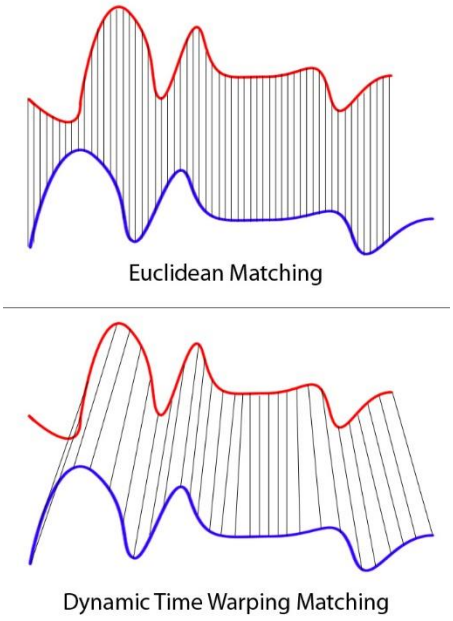


***Figure 5**: comparison of different matching methods*

Dynamic time warping (DTW) is a method of calculating similarity between two time series that is not restricted to measuring point-by-point. The benefit of this method over simple Euclidean distance is the time series do not need their respective time points to match up exactly – figure 5[4]. This allows DTW to be much more flexible and gives a much better measure of similarity when there is variability in time series length [20].

Dynamic time warping was chosen for this project because the DTW algorithm doesn't require the compared time series to be the same length. As described in the Introduction, some firms have waivers so the length of the time series varies from firm to firm. Using DTW as the similarity measuring algorithm allowed firms with similar temporal shapes to have close similarity scores despite potentially having time series of different lengths.

Filtering for firms with a December financial year-end

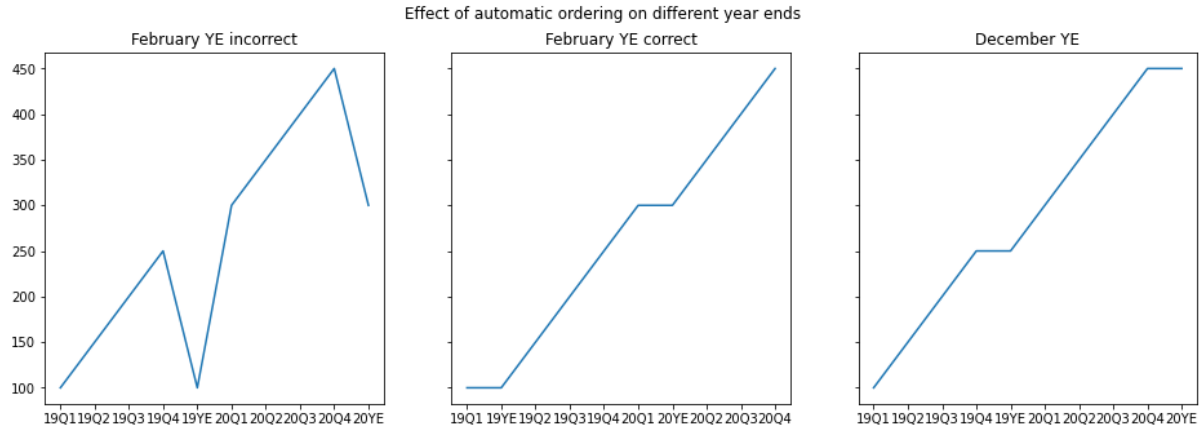


***Figure 6**: shapes of time series with different orderings*

For both the LSTM and DTW algorithms, it was vital that the data was ordered correctly before the algorithms were run. This is because the algorithms are suited for time series data, which is by definition strictly ordered. Incorrectly ordered time series would result in poor performance from the LSTM network, and incorrect clusters from the DTW algorithm. To ensure the correct ordering of the time series, it was necessary to restrict the population of firms to those with 31st December year-ends. The reasoning is intuitive for LSTM networks, however the justification for doing this with the DTW algorithm will be briefly explained.

Figure 6 shows the problem with including all firms in the dynamic time warping calculations. A firm with a February year-end should have their year-end ordered after Q1 (as February is in the 1st quarter of the year), and a firm with a December year-end should have their year-end ordered after

[4] Image taken from https://commons.wikimedia.org/wiki/Category:Dynamic_time_warping#/media/File:Euclidean_vs_DTW.jpg

Q4 – for this example note the assumption that the year-end figure is the same as the corresponding quarter, which is often the case. Python, by default, orders the time periods alphabetically, meaning the year-end periods are always ordered after the quarterly periods in each year. Figure 6 shows how the automatic (and incorrect) alphabetical ordering by Python means the time series shape of the February year-end firm is jagged whereas the correct shape should be smooth, as in the middle chart. Table 3 shows the similarity scores between the February year-end time series and the December year-end time series. DTW aims to minimise the cost, so the smaller the similarity score the more similar the time series. The difference in values demonstrates the effect incorrectly ordered time series would have on the function.

Time Series 1	Time Series 2	Similarity score
December YE	February YE incorrect	200.0
December YE	February YE correct	0.0

***Table 3**: The tslearn documentation for the dtw package gives the property: $\forall x, DTW(x,x)=0$, so 'December YE' and 'February YE correct' are considered identical by the algorithm.[21]*

For this reason, the population used for the LSTM networks and clustering were restricted to firms with a December financial year-end only.

Outlier detection pipelines

Data import and format

The data was loaded via a CSV file and formatted so the features were the cell reference codes on S.02 and each observation was data from one firm in one reporting period. The cell reference code defines a single data point on S.02 (row 30 column 10, row 40 column 10 etc.) as shown in the Annex. For this project, only 'Solvency II' values were included (column 10). The formatted training dataset had 62 features – 62 possible data points from column 10.

The testing period for this project was 2021Q1, so all available reporting periods were extracted up to the period before 2021Q1 i.e. from 2016Q1 to 2020YE. A maximum of 25 periods were available for each firm (firms with waivers had fewer than 25). Any totalled fields were removed, leaving the lowest level of data points in the remaining dataset. This was to ensure the autoencoder could learn relationships within the data that were not immediately obvious – there is little value in the autoencoder learning that 'Bonds' is correlated to 'Government Bonds' and 'Corporate Bonds' - Table 4. Missing values were filled with 0, as a missing value in the template implies the value is 0.

Bonds	R0130
Government Bonds	R0140
Corporate Bonds	R0150

***Table 4**: Bonds on S.02. 'Bonds' is the sum of 'Government Bonds' and' Corporate Bonds' and hence correlated. The summed totals were removed from the data so the autoencoder could discover more 'hidden' relationships*

Whole Dataset Method

Scaling

Once the data was loaded and formatted, it needed to be scaled before running the autoencoder. Three different scaling options were assessed:

The first was to scale the entire dataset as a whole. This is the default method of scaling data in data science. This method does not take into account the difference in firm size.

The second was to scale each firm individually and then join all the scaled firms together into a new dataset. The rationale for this is because the LSTM model was run on each firm individually so the autoencoder would be trained on data scaled in the same way as the input data to the LSTM. Additionally, this method would remove the effect of the different sizes of firms. There is a very large range of the size of firms, with gross premium income ranging from €5m (the minimum requirement SII requirement) to £65bn [22]. Scaling all firms at once could cause the smaller firms to be dwarfed by the larger ones and the autoencoder could end up learning patterns and dependencies disproportionately from the larger firms.

The last method assessed was similar to the second method, but an initial row of zeros was added prior to scaling and after scaling the same row was removed. The reasoning behind this was when plausibility checking, the movements are assessed with zero being the baseline for a movement. For example, if Total Assets moves in the sequence: 100, 200, 100, 200, 100; the movements are judged as doubling and halving, as the movements are relative to zero. Scaled with zero as a reference point, this becomes 0.5, 1, 0.5, 1, 0.5, 1. Without zero, the sequences oscillates between 0 and 1, which is more extreme. This different scaling method was tested to see if data that is scaled to mimic an analysts' plausibility methodology produces a lower loss when training the autoencoder.

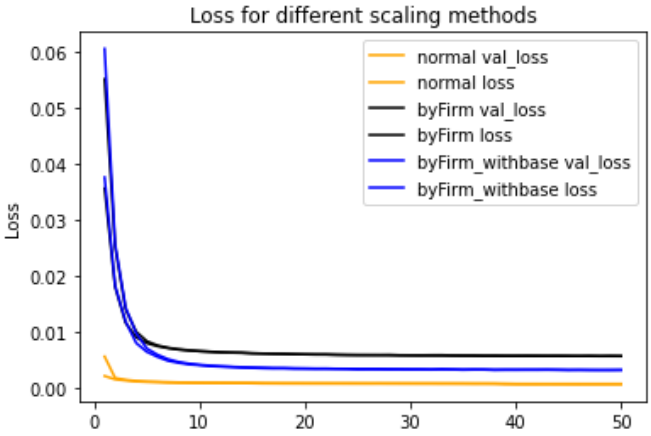


***Figure 7**: loss comparison of different scaling methods*

Figure 7 shows the loss functions of the different scaling methods. All converge to below 0.01, but the first (default) scaling method of scaling the entire dataset as a whole clearly shows the lowest loss. As such, that method was chosen for scaling the data.

Autoencoder

The autoencoder was applied to the training data, with 40 latent variables extracted from the original dataset. These created features were joined with the original data to create a dataset with 102 features.

Feature selection

Machine learning algorithms benefit from being trained on a subset of features rather than the complete dataset. This is to improve time efficiency (fewer features means lower training time) and accuracy (if redundant features are removed, there is less noise in the model and performance will be improved).

Because of the nature of this project, it was necessary that all the original SII features remained in the dataset as all features must be assessed for potential outliers. Using an extreme example, if a firm has consistently reported 0 for one of the features across all time periods in the training data, but reports £1bn in the test time period, this would be a glaring outlier that needs detecting. Running a feature selection algorithm on the training data would likely remove this feature; therefore feature selection was not performed on the SII features.

Due to this requirement, feature selection was performed on the latent variables produced by the autoencoder only. The threshold for removing columns was above 0.7. 4 latent variables were above this threshold and were removed from the dataset.

Latent variable with over 0.7 correlation to other features	7, 21 23, 32

***Table 5**: autoencoder generated variables removed*

Long Short Term Memory network

Once the correlated features were dropped, the data was filtered by firm and a separate LSTM network was trained on each firm separately.

In order for the LSTM to process the data correctly, the input needed to be formatted into 'time windows'. These are the blocks that are fed to the LSTM, for the model to produce a predicted output. For this project, a time window of 5 was chosen i.e. the trend of 5 periods was used to predict the following period. This is because for firms without waivers, there are 5 periods of data reported each year (four quarterly and one annual). The trend of a year's worth of data is an adequate balance between detecting trends and not having an overreliance on historical data. As Figure 8 shows, the vast majority of firms reported every period since the beginning of 2016. Setting the window at 5 meant a handful of firms were excluded from the dataset as their number of reported periods available was less than five, however this was a small proportion of the total population and their exclusion was insignificant.

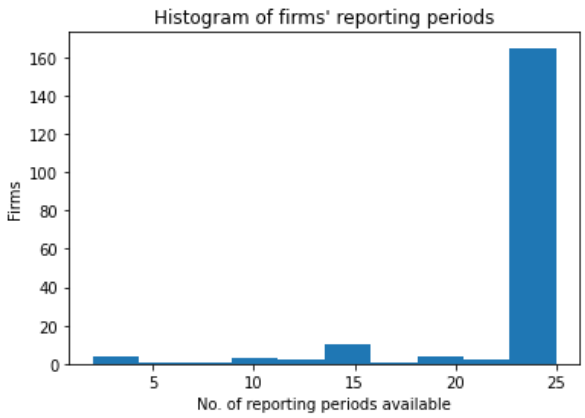


***Figure 8**: showing the proportion of data available from firms. While a handful of firms had waivers, the majority reported every period*

Once the data was split into the required time windows, the LSTM model was run for each firm and the results separated into SII features and autoencoder features for analysis.

The testing methodology will be outlined in the Testing section below.

Cluster Method

There are two main distinct sectors of insurance: life and non-life insurance. Life insurance firms tend to have higher asset values and longer liability exposures, whereas non-life insurance firms have lower assets value and shorter liability exposures. The second method explored in this project looked at whether splitting the original dataset into clusters by exploiting these differences between insurance firms and running an autoencoder on these clusters increased the model's performance over the model run on all the data together. By grouping firms with similar 'behaviour' (in this case firms with similar financial positions are regarded as having similar behaviour), the impact of autoencoder performance trained on data higher in similarity could be assessed. While the most intuitive split of insurance firms into the life/non-life distinction, the makeup of the clusters was not investigated. Therefore, there could be other metrics that influenced the outcome of the clustering algorithm.

Clustering

Two scaling options for similarity measure

In the process of calculating the similarity scores for the firms, it became clear that the scaling would have a large impact on the resulting clusters. The scaling process for the Whole Dataset Method was to scale the data as a whole. However, when the similarity was being calculated, the clusters were extremely unbalanced, with a negligible number in one cluster and the vast majority of other firms in the other cluster – figure 9's left chart (for confidentiality and clarity reasons, the x-axis labels have been removed). The clusters were extremely unbalanced, with the blue cluster not having nearly enough firms for an autoencoder to learn effectively. This result is most likely because there are a handful of very large firms and a high number of much smaller firms.

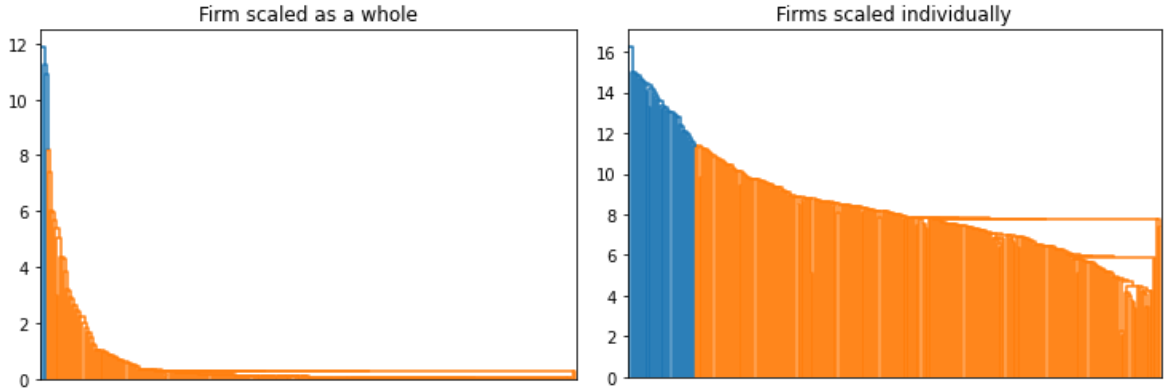


***Figure 9**: results of clustering with different scaling methods*

However, when the scaling was performed on each firm individually and then joined into a new dataset, the similarity calculation produced a more proportional (but still unbalanced) clustering. Life insurance firms tend to be far larger than non-life firms, so by scaling each firm individually the size factor was removed and the clustering could perform with each firm on an equal basis. The resulting clustering is likely to be life/non-life firms has their business is different and values for S.02 different. The unbalanced groupings is intuitive as there are fewer life firms than non-life firms [4]. Figure 9's right chart shows the clustering of firm-scaled data. The gradient is much smoother for all firms, unlike the exponential shape of the scaled as a whole method. As the clusters are more balanced and the differences less extreme, the firm-scaled method was used to calculate the similarity.

Once scaled using the by-firm scaling method, the similarity scores were calculated using the Dynamic Time Warping algorithm and the firms clustered using this score. Then the firms were split into their respective clusters for the outlier analysis.

Running the Cluster Method

Once the method of scaling and clustering was confirmed, the pipeline was run according to the Whole Dataset Method for each cluster – the autoencoder was trained on the cluster, features selected from the generated features, then LSTM model run on each firm in the cluster.

It is worth noting that for the Cluster Method, no columns were flagged to be dropped. The cluster autoencoders generated fewer correlated features (and therefore features with more unique information).

Testing

To determine whether the predicted value from the LSTM model was an outlier, a metric of assessment was needed. This project used two measurements to determine whether a certain predicted feature was an outlier or not.

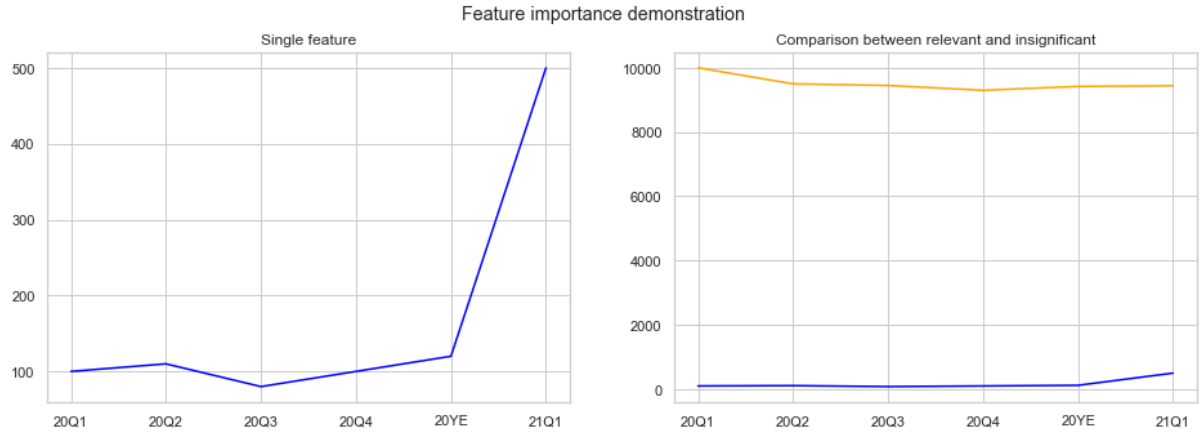


***Figure 10**: the single feature has had a large increase in 2021Q1, however relative to other features, the movement is fairly insignificant*

Firstly, the importance of a feature was calculated in relation to 'Total Assets' (row 500 on S.02). This metric only applied to the SII features, not the generated features. The reason for this was large movements in insignificant features should not be raised as outliers – see Figure 10. When plausibility checking data and looking for potential outliers, a balance needs to be struck between maintaining good data quality and not wasting time resource investigating statistically insignificant variations in the data. Additionally, sending queries to firms over insignificant movements also puts pressure on the firm to investigate and respond. Large movements in insignificant data are of low importance, and the plausibility process needs to be proportional. Therefore, it is important to put features in context of the firm when judging outliers. A threshold of 5% was put on the importance measure of a feature – any feature equal to or greater than 5% of 'Total Assets' was applicable to be judged as an outlier. This also had the benefit of not flagging features which are constant of 0. Some features are rarely reported, and often are 0 across all time periods. The nature of LSTM prediction means an exact predicted value of 0 is extremely rare, and a predicted value as small as 1 might raise a flag, given the deviation from the training data. The measure of importance ensures these small features are not flagged as outliers.

The second calculated metric used to judge outliers was to calculate the standard deviation of the training time series for each feature (this metric was applicable to both SII features and generated features). Predicted values that fell outside one standard deviation of the last training period were flagged as outliers.

Any feature that tested positive against both of these measures was highlighted as an outlier.

As with the hyperparameter selection for the LSTM network, significant challenges were met when running the pipelines. Due to the hardware of the system used, running either pipeline in a single run resulted in the system crashing. Because of this, for the Whole Dataset Method the population of firms was split in half, with the pipeline run for the first half of firms and the results saved. The

laptop had to be restarted and then the pipeline was run for the second half of firms. For the Cluster Method, the clusters were run separately, with the results saved between runs so they could be loaded retrospectively. This is outlined in more detail in the Annex.

Results

The test period was chosen as 2021Q1 as the analysts had manually checked the data for that period already. Therefore, the output of the model could be compared to what the analysts had raised using the existing method. In the below section, an outlier firm is defined as a firm that had at least one data point that was queried as a potential outlier in 2021Q1. A non-outlier firm is a firm with no data points queried.

System specifications

This project was run on a Windows 10 tablet with an i5-7200U processor and 16GB of RAM.

Whole Dataset Method

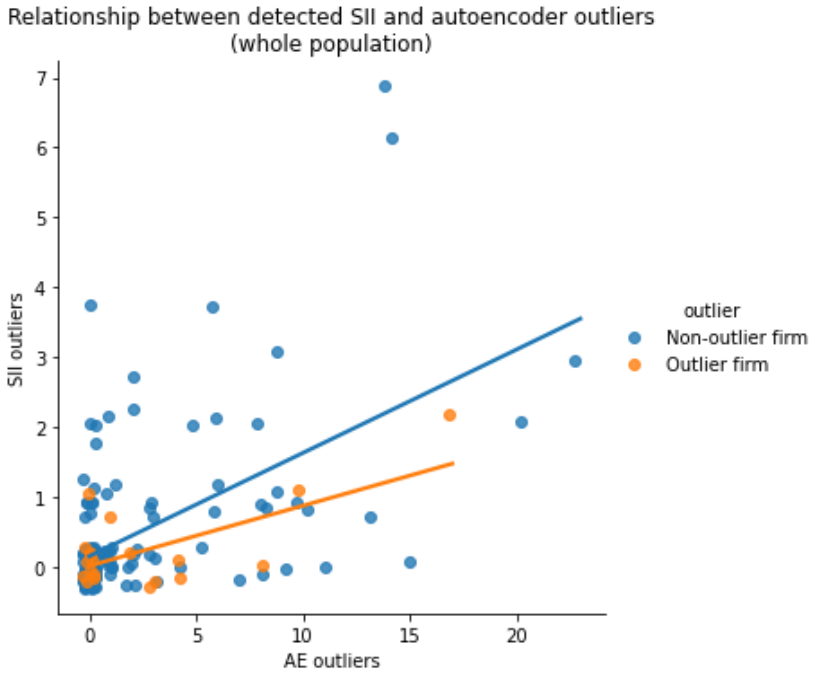


***Figure 11**: comparison of the outliers detected by the model and the firms with outliers in 2021Q1*

Figure 11 shows the relationship of generated autoencoder feature outliers (AE) and the SII feature outliers as judged by the above metrics. As shown, there is a positive correlation between the AE outliers and the SII outliers under the Whole Dataset Method. However, this does not translate into effective outlier detection. A number of non-outlier firms registered a high number of AE outliers and SII outliers, and the majority of test outlier firms registered no SII outliers or AE outliers. The model appears to produce more AE outliers relative to SII outliers for the test outlier firms, however the sample of test outliers is too small for this relationship to be certified with confidence. Research across more time periods needs to be done to explore if this relationship holds.

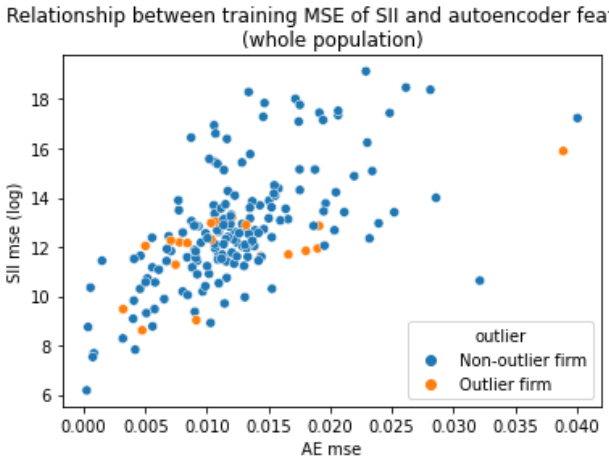


***Figure 12**: training MSE and outlier comparison*

Figure 12 shows the relationship between the training mean squared errors of the AE features and the SII features[5]. As with the outliers, there appears to be a positive correlation between the training MSE of the AE features and SII features. However, there also appears to be no distinction between the MSE values and the outlier firms.

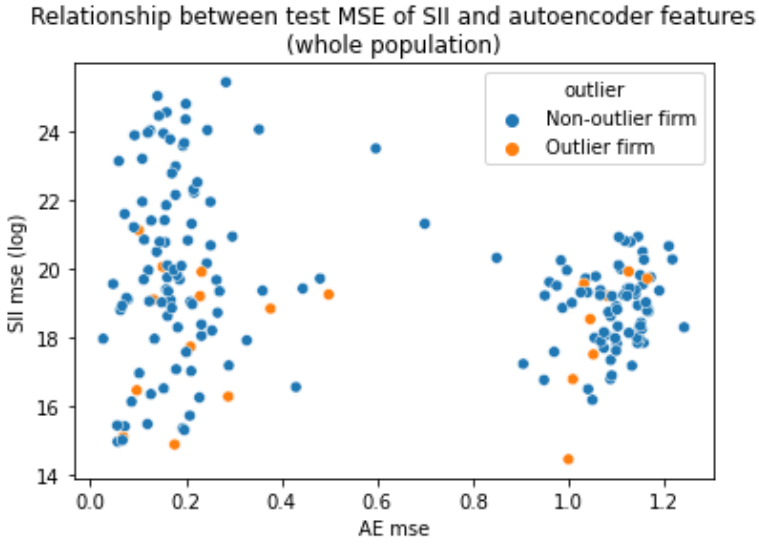


***Figure 13**: test MSE and outlier comparison*

Figure 13 shows the relationship between the test MSE of the SII and AE features. There are two clusters of autoencoder test MSE, on the extremes of the scale. The maximum cluster has a much smaller spread of SII MSE, while the minimum cluster has a much larger spread. The range of SII MSE is much larger for the test values (~15 – 25) than the training values (6 – 18), which indicates the model has overfit the data. As with the training MSE, within these clusters there appears to be no relationship between the firms with outliers and firms with no outliers. Outlier firms appear in both clusters in Figure 13.

The Whole Dataset Method does not appear to have produced meaningful results regarding outlier detection, with little difference being observed between outliers firms and non-outliers firms. The

[5] The SII MSE errors have been logarithmically scaled for readability

model also appears to have overfit the test data, with a large difference between the MSE of training and test data.

Cluster Method

The same analysis was performed on the data produced by the Cluster Method.

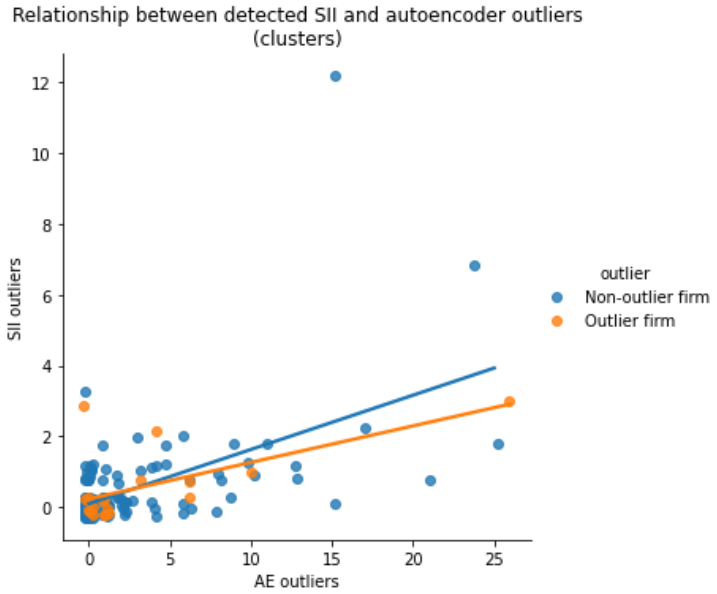


***Figure 14**: comparison of outliers detected by the model and outlier test firms*

Figure 14 shows the outliers detected from the SII features and the AE features. As with the results from the Whole Dataset Method, there was a positive correlation between the outliers detected from the SII features and the AE features, with far more outliers flagged in the AE features then the SII features. There was also very little distinction between the test outlier firms and the non-outlier firms. Again, the majority of outlier firms registered no SII outliers or AE outliers.

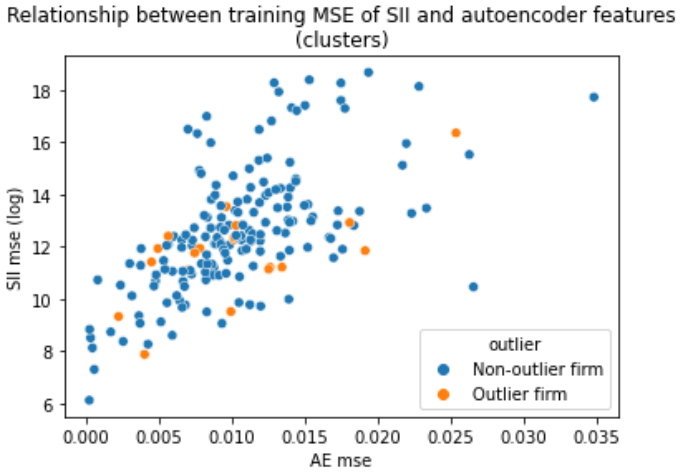


***Figure 15**: comparison of training MSE and test outlier firms*

Figure 15 shows the comparison of the training mean squared errors of the SII features and the AE features. There is a large spread of SII MSE (again, this has been logarithmically scaled), and no discernible separation between the test outlier firms and non-outlier firms.

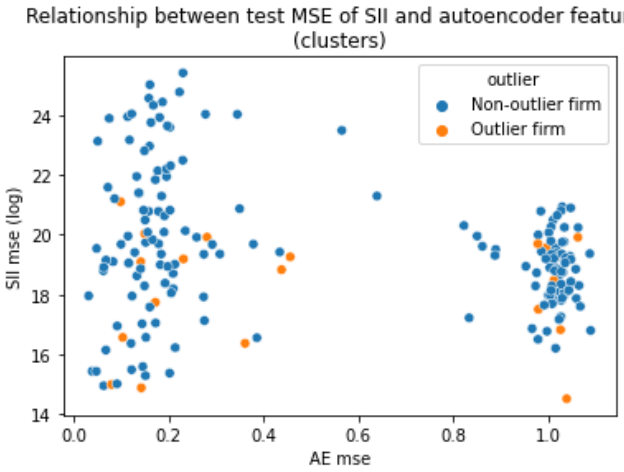


***Figure 16**: comparison of test MSE and outlier test firms*

Figure 16 shows a similar pattern as the test MSE for the Whole Dataset Method. There are 2 groupings on the end of the AE MSE range, a large spread of SII MSE values, much larger test SII MSE values than the training MSE and no pattern between the outlier test firms and non-outlier firms.

The Cluster Method appears to have made next to no improvement on the Whole Dataset Method. The outlier results, training MSE and test MSE all follow the same pattern. It appears using a bespoke autoencoder on clusters of firms has an insignificant impact on improving the performance of a LSTM network on detecting outliers.

Summary / conclusions

The aim of this project was to investigate the contribution an autoencoder could make to an outlier detection system using a Long Short Term Memory neural network to predict the next step of a time series. The first method trained an autoencoder on the whole dataset before running the LSTM network. The second method split the population of firms into two clusters using Dynamic Time Warping and a separate autoencoder was trained on each of the clusters before running the LSTM network.

The results show neither method provides a suitable model for an outlier detection system. The linear trend of outliers detected for outlier firms and non-outlier firms is largely the same and the distinction between outlier firms and non-outlier firms non-existent. The Mean Squared Error is significantly larger on the test data than on the training data, indicating the model has overfit the data and most likely contributing to poor model performance. There are many reasons this could occur and there is potential for more investigation to reduce the level of overfitting. As more data becomes available over time, rerunning these pipelines could produce better results.

Further work

There is potential for further work as a result of this project. Firstly, the LSTM and clustering algorithms were sensitive to the ordering of time series, which meant firms with a financial year-end other than December were excluded. Further work could explore the effect of including these firms by ordering the dates manually, not relying on the alphabetical ordering of Python. This means the autoencoder would have more data to learn dependencies with and the accuracy of the autoencoder could increase. This in turn might lead to better performance with the LSTM network.

Secondly, further investigation is needed to explain if there is a reason for the clustering of firms on Figure 16. It is possible the groupings of the autoencoder test MSE is related to the number of fields a firm has reported – the fewer features reported, the lower the test MSE. However, currently this theory is speculative and requires more research.

References

[1] Solvency II background, EIOPA, accessed 16 September 2021, https://www.eiopa.europa.eu/browse/solvency-ii/solvency-ii-background_en

[2] Solvency II Overview – Frequently asked questions, EIOPA, accessed 16 September 2021, https://ec.europa.eu/commission/presscorner/detail/el/MEMO_15_3120

[3] Solvency II Infographic, Aegon, accessed 16 September 2021, https://www.aegon.com/contentassets/03ec290ec16f4bc0b64feeb95972886e/sovency-ii.pdf

[4] UK Insurance & Long-Term Savings Key Facts, Association of British Insurers, February 2021, accessed 16 September 2021, https://www.abi.org.uk/globalassets/files/publications/public/key-facts/abi_key_facts_2021.pdf

[5] Aviva company details, Aviva, accessed 16 September 2021, https://www.aviva.co.uk/services/about-our-business/about-us/aviva-company-details/

[6] The Lloyd's Market, Society of Lloyd's, accessed 16 September 2021, https://www.lloyds.com/about-lloyds/what-is-lloyds/lloyds-market

[7] Supervisory reporting - DPM and XBRL, EIOPA, accessed 16 September 2021, https://www.eiopa.europa.eu/tools-and-data/supervisory-reporting-dpm-and-xbrl_en

[8] Solvency II: regulatory reporting and limitations', Bank of England, July 2018, accessed 16 September 2021, https://www.bankofengland.co.uk/-/media/boe/files/prudential-regulation/supervisory-statement/2018/ss1115update.pdf?la=en&hash=575507249FF8FF7E20EB824DCD5ADFAE038D7875

[9] Reporting schedules, Regulatory reporting - insurance sector, Bank of England, accessed 16 September 2021, https://www.bankofengland.co.uk/prudential-regulation/regulatory-reporting/regulatory-reporting-insurance-sector

[10] Enhanced financial accounts (UK flow of funds), Office for National Statistics, November 2019, accessed 16 September 2021, https://www.ons.gov.uk/economy/nationalaccounts/uksectoraccounts/articles/enhancedfinancialaccountsukflowoffundsexperimentalfinancialstatisticsfortheukinsurancesectorusingsolvencyiidata/2019-11-26

[11] OECD.Stat, Organisation for Economic Co-operation and Development, accessed 16 September 2021, https://stats.oecd.org/

[12] Y Wang, H Yao, S Zhao, Auto-encoder based dimensionality reduction, Neurocomputing, Volume 184, 2016, Pages 232-242, ISSN 0925-2312, https://doi.org/10.1016/j.neucom.2015.08.104. (https://www.sciencedirect.com/science/article/pii/S0925231215017671)

[13] Keras API reference, Keras, accessed 16 September 2021, https://keras.io/api/

[14] GridSearchCV, scikit learn, accessed 16 September 2021, https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV

[15] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," in *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 15 Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[16] T. Fischer, C. Krauss, Deep learning with long short-term memory networks for financial market predictions, European Journal of Operational Research, Volume 270, Issue 2, 2018, pp. 654-669

[17] K. Chen, Y. Zhou and F. Dai, "A LSTM-based method for stock returns prediction: A case study of China stock market," 2015 IEEE International Conference on Big Data (Big Data), Santa Clara, CA, USA, 2015, pp. 2823-2824

[18] CHEN Xiangyang, YANG Yang, XIANG Yunfei. Measurement of Point Cloud Data Segmentation Based on Euclidean Clustering Algorithm[J]. 测绘通报, 2017, 0(11): 27-31,36.

[19] Berthold, Michael R., and Frank Höppner. "On clustering time series using euclidean distance and pearson correlation." arXiv preprint arXiv:1601.02213 (2016).

[20] Berndt DJ, Clifford J. Using dynamic time warping to find patterns in time series. InKDD workshop 1994 Jul 31 (Vol. 10, No. 16, pp. 359-370).

[21] Dynamic Time Warping, tslearn, accessed 16 September 2021, https://tslearn.readthedocs.io/en/stable/user_guide/dtw.html#dtw

[22] Solvency and Financial Condition Report, Legal and General Group PLC, December 2019, accessed 16 September 2021, https://www.legalandgeneralgroup.com/media/17749/sfcr-legal-general-group-plc.pdf

Annex

S.02 Balance Sheet Template

Balance sheet

		Solvency II value	Statutory accounts value
		C0010	C0020
Assets			
Goodwill	R0010		
Deferred acquisition costs	R0020		
Intangible assets	R0030		
Deferred tax assets	R0040		
Pension benefit surplus	R0050		
Property, plant & equipment held for own use	R0060		
Investments (other than assets held for index-linked and unit-linked contracts)	R0070		
Property (other than for own use)	R0080		
Holdings in related undertakings, including participations	R0090		
Equities	R0100		
Equities - listed	R0110		
Equities - unlisted	R0120		
Bonds	R0130		
Government Bonds	R0140		
Corporate Bonds	R0150		
Structured notes	R0160		
Collateralised securities	R0170		
Collective Investments Undertakings	R0180		
Derivatives	R0190		
Deposits other than cash equivalents	R0200		
Other investments	R0210		
Assets held for index-linked and unit-linked contracts	R0220		
Loans and mortgages	R0230		
Loans on policies	R0240		
Loans and mortgages to individuals	R0250		
Other loans and mortgages	R0260		
Reinsurance recoverables from:	R0270		
Non-life and health similar to non-life	R0280		
Non-life excluding health	R0290		
Health similar to non-life	R0300		
Life and health similar to life, excluding health and index-linked and unit-linked	R0310		
Health similar to life	R0320		
Life excluding health and index-linked and unit-linked	R0330		
Life index-linked and unit-linked	R0340		
Deposits to cedants	R0350		
Insurance and intermediaries receivables	R0360		
Reinsurance receivables	R0370		
Receivables (trade, not insurance)	R0380		
Own shares (held directly)	R0390		
Amounts due in respect of own fund items or initial fund called up but not yet paid in	R0400		
Cash and cash equivalents	R0410		
Any other assets, not elsewhere shown	R0420		
Total assets	**R0500**		

Liabilities			
Technical provisions - non-life	R0510		
Technical provisions - non-life (excluding health)	R0520		
Technical provisions calculated as a whole	R0530		
Best Estimate	R0540		
Risk margin	R0550		
Technical provisions - health (similar to non-life)	R0560		
Technical provisions calculated as a whole	R0570		
Best Estimate	R0580		
Risk margin	R0590		
Technical provisions - life (excluding index-linked and unit-linked)	R0600		
Technical provisions - health (similar to life)	R0610		
Technical provisions calculated as a whole	R0620		
Best Estimate	R0630		
Risk margin	R0640		
Technical provisions - life (excluding health and index-linked and unit-linked)	R0650		
Technical provisions calculated as a whole	R0660		
Best Estimate	R0670		
Risk margin	R0680		
Technical provisions - index-linked and unit-linked	R0690		
Technical provisions calculated as a whole	R0700		
Best Estimate	R0710		
Risk margin	R0720		
Other technical provisions	R0730		
Contingent liabilities	R0740		
Provisions other than technical provisions	R0750		
Pension benefit obligations	R0760		
Deposits from reinsurers	R0770		
Deferred tax liabilities	R0780		
Derivatives	R0790		
Debts owed to credit institutions	R0800		
Financial liabilities other than debts owed to credit institutions	R0810		
Insurance & intermediaries payables	R0820		
Reinsurance payables	R0830		
Payables (trade, not insurance)	R0840		
Subordinated liabilities	R0850		
Subordinated liabilities not in Basic Own Funds	R0860		
Subordinated liabilities in Basic Own Funds	R0870		
Any other liabilities, not elsewhere shown	R0880		
Total liabilities	**R0900**		
Excess of assets over liabilities	**R1000**		

The contents of S.02 'Balance Sheet'. Highlights are EIOPA's from Taxonomy 2.5 annotated templates[6].

S.01 for quarterly and annual periods

Quarterly S.01 Content of submission template

Content of the submission

		C0010
Template Code - Template name		
S.01.02.01 - Basic Information - General	R0010	
S.02.01.02 - Balance sheet	R0030	
S.05.01.02 - Premiums, claims and expenses by line of business	R0110	
S.06.02.01 - List of assets	R0140	
S.06.03.01 - Collective investment undertakings - look-through approach	R0150	
S.08.01.01 - Open derivatives	R0170	
S.08.02.01 - Derivatives Transactions	R0180	
S.12.01.02 - Life and Health SLT Technical Provisions	R0220	
S.17.01.02 - Non-Life Technical Provisions	R0290	
S.23.01.01 - Own funds	R0410	
S.28.01.01 - Minimum Capital Requirement - Only life or only non-life insurance or reinsurance activity	R0580	
S.28.02.01 - Minimum Capital Requirement - Both life and non-life insurance activity	R0590	

[6] https://www.eiopa.europa.eu/tools-and-data/supervisory-reporting-dpm-and-xbrl_en

Annual S.01 Content of submission template

Content of the submission

		C0010
Template Code - Template name		
S.01.02.01 - Basic Information - General	R0010	
S.01.03.01 - Basic Information - RFF and matching adjustment portfolios	R0020	
S.02.01.01 - Balance sheet	R0030	
S.02.02.01 - Assets and liabilities by currency	R0040	
S.03.01.01 - Off-balance sheet items - general	R0060	
S.03.02.01 - Off-balance sheet items - List of unlimited guarantees received by the undertaking	R0070	
S.03.03.01 - Off-balance sheet items - List of unlimited guarantees provided by the undertaking	R0080	
S.04.01.01 - Activity by country	R0090	
S.04.02.01 - Information on class 10 in Part A of Annex I of Solvency II Directive, excluding carrier's liability	R0100	
S.05.01.01 - Premiums, claims and expenses by line of business	R0110	
S.05.02.01 - Premiums, claims and expenses by country	R0120	
S.06.01.01 - Summary of assets	R0130	
S.06.02.01 - List of assets	R0140	
S.06.03.01 - Collective investment undertakings - look-through approach	R0150	
S.07.01.01 - Structured products	R0160	
S.08.01.01 - Open derivatives	R0170	
S.08.02.01 - Derivatives Transactions	R0180	
S.09.01.01 - Income/gains and losses in the period	R0190	
S.10.01.01 - Securities lending and repos	R0200	
S.11.01.01 - Assets held as collateral	R0210	
S.12.01.01 - Life and Health SLT Technical Provisions	R0220	
S.12.02.01 - Life and Health SLT Technical Provisions - by country	R0230	
S.13.01.01 - Projection of future gross cash flows	R0240	
S.14.01.01 - Life obligations analysis	R0250	
S.15.01.01 - Description of the guarantees of variable annuities	R0260	
S.15.02.01 - Hedging of guarantees of variable annuities	R0270	
S.16.01.01 - Information on annuities stemming from Non-Life Insurance obligations	R0280	
S.17.01.01 - Non-Life Technical Provisions	R0290	
S.17.02.01 - Non-Life Technical Provisions - By country	R0300	
S.18.01.01 - Projection of future cash flows (Best Estimate - Non Life)	R0310	
S.19.01.01 - Non-life insurance claims	R0320	
S.20.01.01 - Development of the distribution of the claims incurred	R0330	
S.21.01.01 - Loss distribution risk profile	R0340	
S.21.02.01 - Underwriting risks non-life	R0350	
S.21.03.01 - Non-life distribution of underwriting risks - by sum insured	R0360	
S.22.01.01 - Impact of long term guarantees measures and transitionals	R0370	
S.22.04.01 - Information on the transitional on interest rates calculation	R0380	
S.22.05.01 - Overall calculation of the transitional on technical provisions	R0390	
S.22.06.01 - Best estimate subject to volatility adjustment by country and currency	R0400	
S.23.01.01 - Own funds	R0410	
S.23.02.01 - Detailed information by tiers on own funds	R0420	
S.23.03.01 - Annual movements on own funds	R0430	
S.23.04.01 - List of items on own funds	R0440	
S.24.01.01 - Participations held	R0450	
S.25.01.01 - Solvency Capital Requirement - for undertakings on Standard Formula	R0460	
S.25.02.01 - Solvency Capital Requirement - for undertakings using the standard formula and partial internal model	R0470	
S.25.03.01 - Solvency Capital Requirement - for undertakings on Full Internal Models	R0480	
S.26.01.01 - Solvency Capital Requirement - Market risk	R0500	
S.26.02.01 - Solvency Capital Requirement - Counterparty default risk	R0510	
S.26.03.01 - Solvency Capital Requirement - Life underwriting risk	R0520	
S.26.04.01 - Solvency Capital Requirement - Health underwriting risk	R0530	
S.26.05.01 - Solvency Capital Requirement - Non-Life underwriting risk	R0540	
S.26.06.01 - Solvency Capital Requirement - Operational risk	R0550	
S.26.07.01 - Solvency Capital Requirement - Simplifications	R0560	
S.27.01.01 - Solvency Capital Requirement - Non-life and Health catastrophe risk	R0570	
S.28.01.01 - Minimum Capital Requirement - Only life or only non-life insurance or reinsurance activity	R0580	
S.28.02.01 - Minimum Capital Requirement - Both life and non-life insurance activity	R0590	
S.29.01.01 - Excess of Assets over Liabilities	R0600	
S.29.02.01 - Excess of Assets over Liabilities - explained by investments and financial liabilities	R0610	
S.29.03.01 - Excess of Assets over Liabilities - explained by technical provisions	R0620	
S.29.04.01 - Detailed analysis per period - Technical flows versus Technical provisions	R0630	
S.30.01.01 - Facultative covers for non-life and life business basic data	R0640	
S.30.02.01 - Facultative covers for non-life and life business shares data	R0650	
S.30.03.01 - Outgoing Reinsurance Program basic data	R0660	
S.30.04.01 - Outgoing Reinsurance Program shares data	R0670	
S.31.01.01 - Share of reinsurers (including Finite Reinsurance and SPV's)	R0680	
S.31.02.01 - Special Purpose Vehicles	R0690	
S.36.01.01 - IGT - Equity-type transactions, debt and asset transfer	R0740	
S.36.02.01 - IGT - Derivatives	R0750	
S.36.03.01 - IGT - Internal reinsurance	R0760	
S.36.04.01 - IGT - Cost Sharing, contingent liabilities, off BS and other items	R0770	

As these demonstrate, the annual reports are much larger than the quarterly reports. All the quarterly templates are also reported annually.

Saving results during pipeline runs

Issues were encountered when trying to run both of the pipelines from start to finish due to a lack of system memory. Due to the confidential nature of the input data, it was not possible to find an external system to process the data (such as Google Colab, Amazon Web Services etc.). The pipelines were coded to be run holistically from start to finish on a capable system. However, for the purpose of producing this report using the software available, it was necessary to run the pipeline in halves, save the results, run the second half and join the results together.

The 'lstm saved files.py' script contains the code for saving and loading the results of the pipeline. On adequate software, this script should not be necessary for obtaining results.

For the Whole Dataset Method, to get consistent results it was necessary to save a version of the autoencoder once it had been trained. Then the population of firms was halved, the pipeline run and the results saved. For the second run, the autoencoder was loaded from the saved file and the pipeline run for the second half of firms. These results were saved. The outputs seen in the report are a result of loading these saved files and joining them together.

For the Cluster Method, the clusters were run separately and the results of each cluster saved. As above, to obtain the results in the report, the results of both clusters were loaded and joined together.