A Multi-head LSTM Architecture for Bankruptcy Prediction with Time Series Accounting Data

Gianfranco Lombardo , Mattia Pellegrino , Stefano Cagnoni , George Adosoglou , Panos Pardalos , Agostino Poggi *

Posted Date: 17 January 2024

doi: 10.20944/preprints202401.1290.v1

Keywords: Bankruptcy prediction; Deep learning; Multi-head; Recurrent Neural Networks; Stock market

Article

A Multi-Head LSTM Architecture for Bankruptcy Prediction with Time Series Accounting Data

Mattia Pellegrino [1,†], Gianfranco Lombardo [1,†], George Adosoglou [2,†], Stefano Cagnoni [1,†], Panos M. Pardalos [2] and Agostino Poggi [1,*]

[1] University of Parma - Department of Engineering and Architecture,Parma, Italy; gianfranco.lombardo@unipr.it (G.L.); mattia.pellegrino@unipr.it (M.P); stefano.cagnoni@unipr.it (S.C); agostino.poggi@unipr.it (A.P)

[2] University of Florida - Department of Industrial and Systems Engineering,Gainesville FL, USA; g.adosoglou@ufl.edu (G.A.); pardalos@ufl.edu (PM.P);

* Correspondence: agostino.poggi@unipr.it

† These authors contributed equally to this work.

Abstract: With the recent advances in Machine Learning (ML), several models have been successfully applied to financial and accounting data to predict the likelihood of companies' bankruptcy. However, time series have received little attention in literature with a lack of studies on the application of Deep Learning sequence models as Recurrent Neural Networks (RNN) and the recent Attention-based models in general. In this research work, we investigated the application of Long Short Term Memory (LSTM) networks to exploit time series of accounting data for bankruptcy prediction. The main contributions of our work are the following: a) we have proposed a Multi-head LSTM that models each financial variable in a time window independently and compared it with a single-input LSTM and other traditional ML models. The Multi-head LSTM outperforms all the other models; b) We identify the optimal time series length for bankruptcy prediction to be equal to 4 years of accounting data; c) We made public the dataset we used for the experiments that include data from 8262 different public companies in the American stock market generated in the period between 1999-2018. Furthermore, we prove the efficacy of the Multi-head LSTM models in terms of lower false positives and better division of the two classes.

Keywords: bankruptcy prediction; Deep Learning; Multi-head; Recurrent Neural Networks; stock market

1. Introduction

Predicting corporate bankruptcy is one of the most fundamental tasks in credit risk assessment. Especially after the 2007/2008 financial crisis, it has become a top priority, for most financial institutions, fund managers, and lenders, due to the substantial financial damage that can result from corporate default. Indeed, corporate failure may result in high social costs and further propagate recession, especially when it involves a large number of companies simultaneously and affects the entire economy as a whole [1]. Since the 2008 financial crisis, researchers and practitioners made several efforts to build models that can efficiently assess the likelihood of companies default, especially for public companies in the stock market. Regulators also benefit from accurate bankruptcy forecasting models since they can monitor the financial health of institutions and curb systemic risks [2].

Since Altman presented his bankruptcy forecasting model in 1968 [3], research has shown that accounting-based ratios and stock market data can signal whether a firm is likely to face severe difficulties, such as bankruptcy. Although default prediction models have been studied for decades, we still lack a definite theory of predicting corporate failure [4]. The lack of a theoretical framework led to the adoption of a common development methodology where research is more focused on identifying discriminant features using a trial and error approach [5,6].

The advent of Machine Learning (ML) and its advances offered novel possibilities for bankruptcy prediction in terms of learning models with several attempts with different ML algorithms and

 doi:10.20944/preprints202401.1290.v1

techniques such as Support Vector Machine (SVM) [7], boosting techniques [8], discriminant analysis [9] and Neural Networks. Moreover, different architectures have been evaluated to identify effective decision boundaries for this binary classification problem, such as the least absolute shrinkage and selection operator [10], dynamic slacks-based model [11] and two-stage classification [12].

However, a common element of those models is the punctual application of market-based and accounting-based variables, while time series receive little attention and there is insufficient literature concerning the application of the most recent Deep Learning models for sequence data such as Recurrent Neural Networks and Attention-based models. In this paper we compare two different Recurrent Neural Networks (RNN) architectures based on Long Short Term Memory (LSTM) units to predict Bankruptcy based on time series of accounting data. In general RNN-based models have been seldom investigated in recent literature. In particular, we used a single-input RNN and a Multi-head RNN that models each financial variable within a time window independently exploiting the latent representation learned only in the last stage of the bankruptcy classification process. The idea of building a Multi-head architecture aims to investigate whether an attention method pipeline con outperform a classical RNN setting when learning a latent representation of the company by focusing on each time series independently.

The main contributions of this paper are the following:

- We propose a Multi-head LSTM for bankruptcy prediction on time series data.
- We investigated the optimal time window of financial variables to predict bankruptcy with a comparison among the main state-of-the-art approaches in Machine Learning and Deep Learning. Experiments have been performed on public companies traded in the American stock market with data available between 2000 and 2018.
- We anonymized our dataset and made public for the scientific community for further investigations and to provide a benchmark for future studies on this topic.
- We analyzed our models on the test set, using T-SNE [13] to show the ability of our models to capture patterns. We also performed an in-depth analysis of false positives.

2. Related Works

In traditional methods to forecast bankruptcies, Altman's Z-score is the most prominent, but Kralicek quick test, and Taffler's model also use scoring methodologies to provide ordinal rankings of default risk. [14,15]. Altman, as well as Beaver and William, used discriminant analysis, which has been widely used following their works, while Ohlson was the first to introduce a binary response model using explanatory variables and applying a logistic function [16,17]. The scoring methodologies have also been used to produce a binary response given a pre-set threshold. For example, Altman suggested the use of two thresholds, 1.81 and 2.99. According to this, an Altman's Z-score above the 2.99 threshold means that firms are not predicted to default in the next two years, below 1.81 that they are predicted to default, while between the two thresholds lies a "zone of ignorance" where no clear decision can be taken. However, even though many practitioners use these thresholds, in Altman's view, this is an unfortunate practice since over the past 50 years, credit-worthiness dynamics and trends have changed so dramatically that the original zone cutoffs are no longer relevant [18].

Even though many authors continue to work on traditional bankruptcy models, the exploration of machine learning applications on corporate default has been more prevalent in recent years [5,6,12, 19–22]. Barboza et al. show that, on average, machine learning models exhibit 10% higher accuracy compared to traditional ones. Specifically, in this study, Support Vector Machines (SVM), random forests (RF) as well as bagging and boosting techniques were tested for predicting bankruptcy events and compared with results from discriminant analysis, logistic regression, and neural networks. The authors found that bagging, boosting and RF outperformed all other models [23]. Altman, in his recent book, however, discusses a trade-off between models' performance and explainability when using machine learning models, expressing skepticism whether practitioners would adopt "black-box" methods nevertheless acknowledging the superiority of the models in assessing corporate distress [18].

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 17 January 2024 doi:10.20944/preprints202401.1290.v1

Considering that the results regarding the superiority of models are still inconclusive, new studies exploring different models, contexts, and datasets are relevant. Machine learning techniques like ensembles of classifiers had been firstly explored for default prediction by Nanni et al.[24]. Kim et al. showed that the ensembles greatly outperform standalone classifiers [25]. Wang et al. further analyzed the performance of ensemble models, finding that bagging outperforms boosting in average accuracy for all credit databases they used, as well as type I and type II error [26]. In [27] some evidence is presented about the need of considering time series for the survival probability estimation over the years and also for bankruptcy prediction with some benchmarks that also prove that neural networks, when properly designed, can achieve better results with time-dependent accounting variables.

Barboza et al. also argue that a firm's failure will likely be caused by difficulties over time, not just the year before bankruptcy. In order to incorporate the dynamic behavior of firms, they added new variables reflecting changes in financial metrics such as growth measures and changes [23]. Findings dating back to 1966 show that firms exhibit failure tendencies as much as five years prior to the actual event [16]. On the other hand, in 1998, Mossman et al. pointed out that the models are only capable of predicting bankruptcy two years prior to the event, which improves to three years if used for multiperiod corporate default prediction [28,29]. In most studies, ratios are analyzed backward in time starting with the bankruptcy event and going back until the model becomes unreliable or inaccurate. The time threshold for developing good classification models is two or three years, at most five, while Altman mentions in his book that there are certain characteristics of bonds at birth that can significantly influence their default likelihood over up to ten years after issuance [16,18,28,29].

However, most of the bankruptcy prediction models in the literature do not take advantage of the sequential nature of the financial data. This lack of multi-period models is also emphasized in Kim et al. literature review [30]. One of the few studies that do leverage the sequential nature of accounting data is that of Vochozka et al., who examined the performance of a long short-term memory (LSTM) model for bankruptcy prediction in the Czech manufacturing sector [31]. Kim et al. also used quarterly accounting data for non-financial industry companies and daily market data from January 2007 through December 2019 in both RNN and LSTM models, finding that RNNs made reasonable predictions in most situations, while both LSTMs and RNNs outperformed logistic regression, support vector machine, and random forest methodologies [32]. However, to the best of our knowledge, there are no studies of corporate bankruptcy that examine a similar number of observations or leverage time series data examining different time windows to predict default. Since it is difficult to make a fair comparison with the available literature (most of the datasets are either small and not usable with Deep Learning or, more commonly, they have not been publicly released), we compared our Deep learning model with all the algorithms presented in this section on our dataset. The dataset has been publicly released for further investigations and comparisons and it is avaliable on GitHub [1]. See Section 4 for more details.

3. Recurrent Neural Networks

Recurrent Neural Networks (RNN) are a Deep Learning architecture that aims to process sequences of values in the form $x^{(1)}$,$x^{(2)}$.....,$x^{(t)}$. This ability is due to the networks' parameter sharing across different parts of a model that makes it possible to extend and apply the model to examples of different forms. Moreover, parameter sharing also allows to preserve generalization across the sequence since the same parameters (weights) are used for each value of the time index, while a traditional fully connected feed-forward network would have separate parameters for each input feature. The time index refers to the position in the sequence. A Recurrent Neural network is generally composed of a single unit of processing that produces an output y at each time step and has recurrent connections in general from the hidden units $a^{<j>}$ and optionally from the output. When an RNN has

[1] https://github.com/sowide/bankruptcy_dataset

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 17 January 2024 doi:10.20944/preprints202401.1290.v1

only recurrent connections from the hidden units, it processes an entire sequence and then produces a single output. Figure 1 shows a generic RNN structure that processes a sequence of t elements with recurrent connections only from the hidden units. Input, output, and recurrent hidden states are propagated using different weight matrices whose elements are learned during training using the back-propagation through time algorithm [33]. Equations 1 and 2 describe the internal behavior of an RNN unit. The initial hidden state a^0 is generally equal to zero. In general, for a generic time index j the hidden state is computed as a weighted sum of the previous state a^{j-1} and the current input x^j plus a bias term. After that, an activation function σ is applied to the result as in fully-connected networks. The output at each time step only depends on the current internal state plus a different bias b_y. The two activation functions to estimate the hidden state and the output may differ.

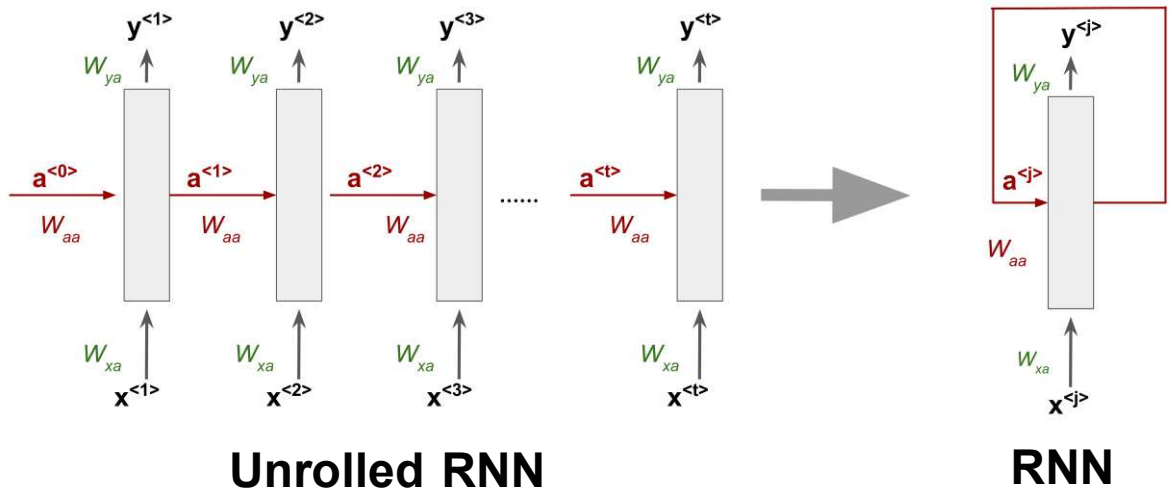

Figure 1. (Left) A generic structure of an unrolled Recurrent Neural Network with recurrent connections from the hidden layer. (Right) The resulting RNN when presented as a single unit with recurrent connections.

$$a^{<j>} = \sigma(W_{aa} \cdot a^{<j-1>} + W_{xa} \cdot x^{<j>} + b_a) \tag{1}$$

$$y^{<j>} = \sigma(W_{ya} \cdot a^{<j>} + b_y) \tag{2}$$

In this way, Recurrent Neural Networks can process entire sequences and are able to use contextual information when mapping inputs into outputs. Unfortunately, for standard RNN architectures, the range of context that can be in practice accessed is quite limited, especially for long sequences and for more than one sequence in the input (matrix input). The major issue is that the influence of a given input on the hidden state, and therefore on the network output, either decays or blows up exponentially as it cycles around the network's recurrent connections. This effect is often referred to in the literature as the vanishing gradient problem [34]. Several approaches have been presented to solve this issue, like the Long-Short Term Memory (LSTM) architecture [35] and the Gated-Recurrent Unit (GRU) [36]. In both architectures, several additional components (called gates) are introduced inside the unit to extend the memory of the network in case of a long sequence so that the first part of each sequence is not forgotten when producing the output (long-term dependency problem) and to prevent the gradient from vanishing. We describe the LSTM since it is the one used in this research work and because the GRU unit can be taken back to a particular case of the LSTM unit. The basic idea is to employ a unit state s_j to retain the information taken from earlier time indexes in the sequence. The unit is composed of three gates:

1. **Forget Gate**: It determines the amount of information that should be retrieved from the previous unit state.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 17 January 2024 doi:10.20944/preprints202401.1290.v1

2. **Input Gate**: It defines the amount of information from the new input $x^{<j>}$ that should be used to update the unit's internal state.
3. **Output Gate**: It defines the output of the unit as a function of its current unit state.

An example of an LSTM unit is presented in Figure 2. Every connection is weighted by a different matrix whose elements are estimated using back-propagation through time as in basic RNNs.

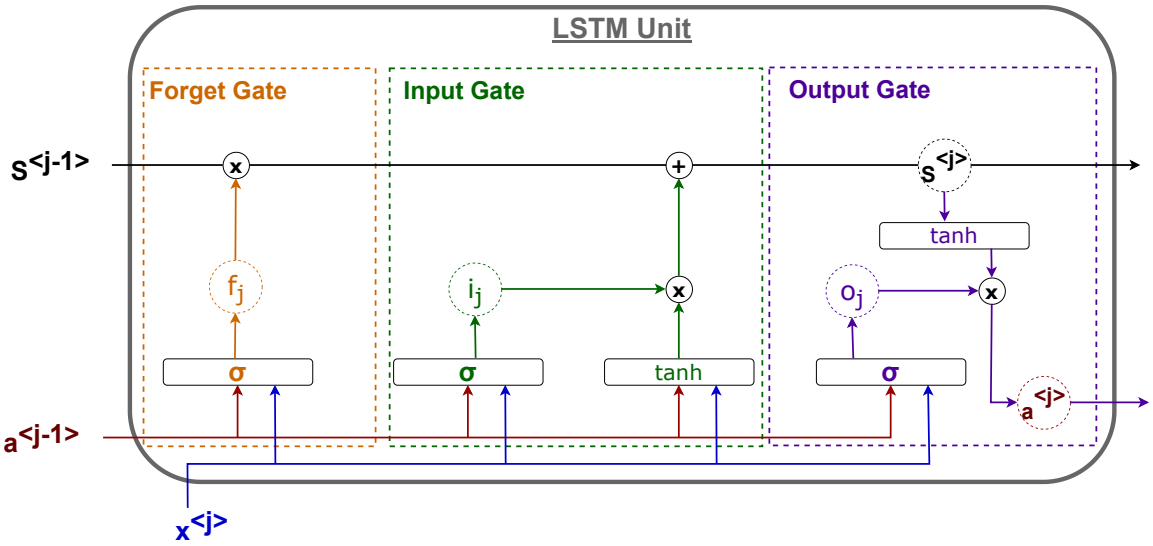

Figure 2. The internal structure of an LSTM unit with Forget, Input and Output gates and the respective activation functions.

4. Dataset

In this section, we present the dataset used in the experimental part and that we make available for the scientific community. The procedure used to build the dataset can be described as follows:

1. We collected data about 8262 different public companies in the American stock market between 1999 and 2018. We selected the same companies used in [37] and [38] since these companies have been considered a good approximation of the American stock market (NYSE and NASDAQ) in those time intervals.
2. For such firms, we collected 18 financial variables, often used for bankruptcy prediction, for each year. In bankruptcy prediction it is common to consider accounting information and up-to-date market information that may reflect the company's liability and profitability. We selected the variables listed in Table 1 as the minimum common set found in the literature [3,10,39] and to have a dataset for twenty years without missing observations.
3. For all the experiments presented in Section 5, we only considered firms with at least 5 years of activity since we aim to first identify the time window that optimizes the bankruptcy prediction accuracy.

Each company has been labeled every year depending on its next year's status: According to the Security Exchange Commission (SEC) a company in the American market is considered bankrupted in two cases:

- If the firm's management files Chapter 11 of the Bankruptcy Code to "reorganize" its business: management continues to run the day-to-day business operations, but all significant business decisions must be approved by a bankruptcy court.
- If the firm's management files Chapter 7 of the Bankruptcy Code: the company stops all operations and goes completely out of business.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 17 January 2024 doi:10.20944/preprints202401.1290.v1

In both cases, we labeled the fiscal year before the chapter filling as "Bankruptcy" (1). Otherwise, the company is considered healthy (0). In light of this, our dataset enables learning how to predict bankruptcy at least one year before it happens.

There is typically a strong imbalance in bankruptcy datasets since the number of firms that declare default each year is usually a small percentage below 1% of the available firms in the market. However, in some periods, bankruptcy rates have been higher than usual, for example during the Dot-com Bubble in the early 2000 and the Great Recession between 2007-2008. Our dataset reflects this condition as shown in Figure 3. The dataset firm distribution by year is presented in Table 2.

Moreover, each company in the dataset is categorized using the Standard Industrial Classification (SIC) system, developed by the U.S. government to classify businesses based on their primary economic activities. The SIC codes not only distinguish firms but also enable more granular categorization by specifying major groups within each firm [40]. Major groups represent specific subcategories that define the business-type activities undertaken by these companies. The inclusion of SIC codes and major groups allows us a deep analysis of bankruptcy trends across a wide range of industries. This classification system helps researchers and analysts gain insights into the economic factors and market dynamics affecting various sectors of the American economy, making it a valuable resource for studying bankruptcy patterns and their implications for different industries and major groups. Additionally, we generated a frequency distribution chart in Figure 4, to represent companies' distribution across different divisions within our dataset. This histogram provides a clear overview of the prevalence of companies in each different division, highlighting which sectors of the economy are more heavily represented among the bankrupt firms. It works for evaluating the overall dataset composition and identifies any potential trends or disparities in bankruptcy occurrences across divisions. In this work, we performed a comprehensive examination of the dataset, with a particular focus on the division distribution.

Finally, the resulting dataset of 78682 firm-year observations is divided into three subsets according to the time period: a training set, a validation set, and a test set. We used data from 1999 to 2011 for training, data from 2012 to 2014 for validation and model comparison, and the remaining years from 2015 to 2018 as a test set to assess the ability of the models to generalize their prediction to unseen data.

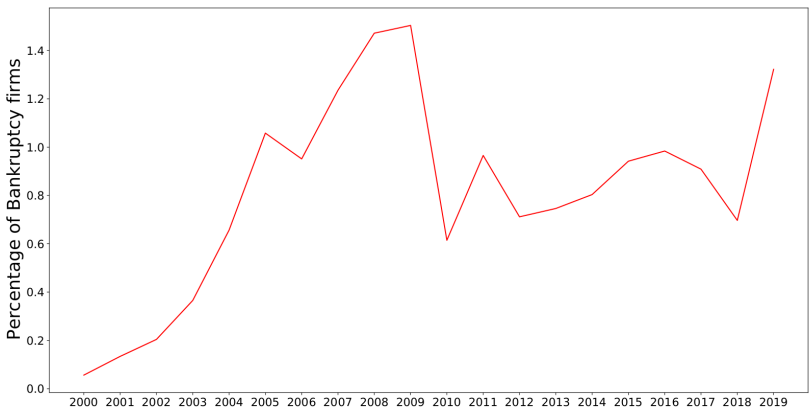

Figure 3. Rate of Bankruptcy in the dataset (2000-2019) with financial variables in the period (1999-2018).

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 17 January 2024 doi:10.20944/preprints202401.1290.v1

Table 1. The 18 numerical bankruptcy features

Variable name	Variable name
Current assets	Total assets
Cost of good sold	Total Long term debt
Depreciation & amortization	EBIT
EBITDA	Gross profit
Inventory	Total Current liabilities
Net income	Retained earnings
Total Receivables	Total Revenue
Market value	Total Liabilities
Net Sales	Total Operating expenses

Table 2. Firm distribution by year in the dataset.

Year	Total Firms	Bankruptcy firms	Year	Total firms	Bankruptcy firms
2000	5308	3	2010	3743	23
2001	5226	7	2011	3625	35
2002	4897	10	2012	3513	25
2003	4651	17	2013	3485	26
2004	4417	29	2014	3484	28
2005	4348	46	2015	3504	33
2006	4205	40	2016	3354	33
2007	4128	51	2017	3191	29
2008	4009	59	2018	3014	21
2009	3857	58	2019	2723	36

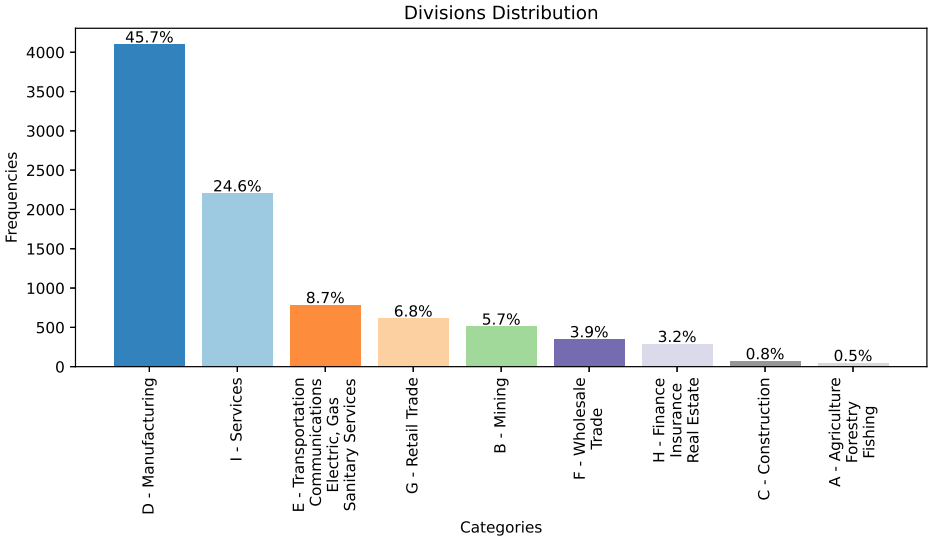

Figure 4. Distribution of Companies by Division

5. Hardware specifications

All the experiments described in this work have been performed using a Linux Ubuntu server with the following hardware specifications:

- **CPU**: Intel i9-10900 @2.80 GHZ
- **GPU**: Nvidia RTX 3090 (24 GB)
- **RAM**: 32 GB DDR4 - 2667MHz
- **Motherboard**: Z490-A PRO (MS-7C75)

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 17 January 2024 doi:10.20944/preprints202401.1290.v1

6. Temporal window selection

Before considering the use of time series with Deep Learning we investigated a key question: How many years should be taken into account to maximize bankruptcy prediction performance? When considering more than one year of accounting variables, different trade-offs should be considered:

- Some firms could only be considered for certain time windows since they have only recently been made public.
- Some firms could be excluded depending on the time window although they existed in the past because of an acquisition or merging operation.
- By extending the training and testing window, the number of companies available for training and testing will inevitably decrease. Moreover, one should consider that a time window above a certain number of years introduces a statistical bias that limits the analysis to only structured companies that have been on the market for several years. At the same time, it leads to ignoring the relatively new companies, which usually have smaller market capitalization and thus are riskier and with a higher probability of default, especially in an overall adverse economic environment.

In order to answer these questions, we experimented with different machine-learning models to identify the most promising time window length. In particular, we used the same ML models that have been considered most effective in literature for bankruptcy prediction [18]: Support Vector Machine (SVM), Logistic Regression (LR), Random Forest (RF), AdaBoost (AB), Gradient Boosting (GB), Extreme Gradient Boosting (XGB) as well as two other tree-based boosting ML models, LightGBM (LGBM) [41] and CatBoost (CB) [42]. Although they have not been used previously for Bankruptcy prediction these models recently achieved outstanding performances in other tasks when compared with AB,GB and XGB.

All the models were trained on the same training set (1999-2011) and compared using the validation set (2012-2014). The training set has been balanced because, otherwise, a bias would occur that would cause the less representative class in the bankruptcy to be wrongly classified and learned. For this reason, every model is evaluated over 100 independent and different runs: For every run, the training set is balanced with exactly all the bankruptcy examples and a random choice of healthy examples from the same period.

We compared all the models using the average Area Under the Curve (AUC) on the 100 runs. AUC is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the Receiver Operating Characteristic (ROC) curve. Every model implements a binary classification task where the positive class (1) represents bankruptcy, and the negative class (0) represents the healthy status.

In order to deal with the constraints previously listed, we evaluated all the companies using a time window of accounting variables spanning 1-5 years. For RF, AB, GB, XGB, LGBM and CB we used the same number of estimators (equal to 500) for a fair comparison, while the other specific parameters were taken equal to the defaults provided in the Scikit-Learn implementations.

In Table 3 we report the average results obtained on the validation set for every model depending on the number of years considered (Window length WL). Figure 5 summarizes the comparison. As expected and according to previous literature, the ensemble models usually reach better results. In particular, we found that for $WL = 4$, the Random Forest with 500 estimators obtains, on average, a greater AUC on the validation set. On the other hand, for the $WL = 5$ case the best model found has been the CatBoost. In both cases, the two ML algorithms achieve better performance if compared with the others baselines. For this reason, we considered both CB and RF for the subsequent analysis.

It is important to note that all the ML models considered are not designed to work on time series data, and thus they consider all the variables as independent features. As expected, increasing the window length led to longer training time on average for all the models (Table 4). However, all the models require just a few seconds of training with our hardware settings. However, ensemble models perform better than SVM and LR with more variables with a larger window length. Moreover, using

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 17 January 2024 doi:10.20944/preprints202401.1290.v1

the 18 accounting variables for 4 or 5 consecutive years yields a better result in terms of AUC in particular with Random Forest (0.759 for $WL = 4$) and CatBoost (0.771 for $WL = 5$). A possible consideration is that although RF and CB show similar overall performance, the computation time required to train CB is almost six time the one required for RF. However, the computation time is not relevant in our experiments since the dataset is not particularly large. In light of this, we selected $WL = 4$ and $WL = 5$ to further study time series data with Recurrent Neural Networks.

Table 3. Average AUC on the validation-set depending on the number of years considered (Window Length-WL). Best results for $WL = 4$ and $WL = 5$ are highlighted in boldface.

	Average AUC				
ML models	**WL=1**	**WL=2**	**WL=3**	**WL=4**	**WL=5**
Support Vector Machine	0.635	0.641	0.594	0.589	0.587
Logistic Regression	0.731	0.648	0.676	0.705	0.702
AdaBoost	0.647	0.655	0.664	0.642	0.719
Random Forest	0.745	0.733	0.731	**0.759**	0.760
Gradient Boosting	0.716	0.702	0.729	0.685	0.742
XGBoost	0.678	0.730	0.695	0.699	0.726
CatBoost	0.724	0.729	0.686	0.749	**0.771**
LightGBM	0.751	0.699	0.671	0.741	0.736

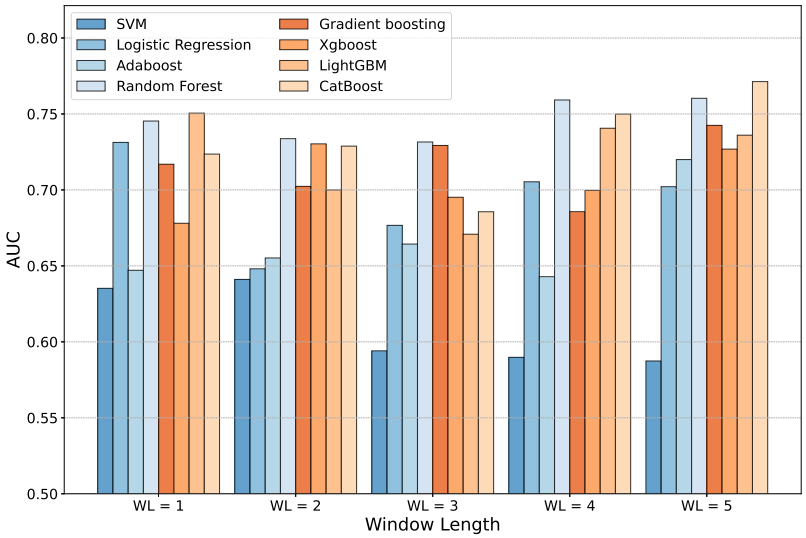

Figure 5. Average AUC on the validation set over 100 runs for each model for different the window lengths in years (WL). The training set is randomly balanced for every run while the validation set is imbalanced.

Table 4. Average training time [seconds] on the validation set depending on the Window Length (WL). Times in seconds refer to the average training time for a single run.

	Average training time[s]				
ML models	**WL=1**	**WL=2**	**WL=3**	**WL=4**	**WL=5**
Support Vector Machine	0.032	0.036	0.036	0.034	0.033
Logistic Regression	0.018	0.022	0.023	0.024	0.026
AdaBoost	0.826	1.246	1.560	1.769	1.994
Random Forest	0.799	0.973	1.033	1.034	1.059
Gradient Boosting	1.022	1.833	2.545	3.097	3.555
XGBoost	0.421	0.422	0.493	0.478	0.483
CatBoost	6.670	7.059	7.088	7.383	7.398
LightGBM	0.187	0.195	0.191	0.184	0.179

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 17 January 2024 doi:10.20944/preprints202401.1290.v1

7. LSTM architectures for Bankruptcy prediction

According to [43], LSTM performs better than GRU when the sequence is short although with a matrix input. For this reason, we chose the LSTM approach for our experiments since we are considering eighteen different time series as input, each with a short sequence length, as determined in the first experiments in Section 6. In order to study the application of RNNs to bankruptcy prediction, we evaluated two different architectures:

- **A single-input LSTM**: This is the most common approach with RNNs. The input is a matrix with 18 rows (number of accounting variables) and a number of columns equal to the time window selected for the experiment. Moreover, the LSTM is composed of a sequence of units as long as the time window. Finally, a dense layer with a Softmax function is used as an output layer for the final prediction.
- **A Multi-head LSTM**: This is one of the main contributions of our research with respect to the current state of the art. In order to deal with a smaller training set due to the temporal window selection and the class imbalance, we developed several smaller LSTMs, one for each accounting variable to be analyzed by the model, named LSTM heads. Each network includes a short sequence of units equal to the input sequence length and contributes to the latent representation of the company learned by utilizing the accounting variables. Indeed, the output of the Multi-head layer is then concatenated and exploited by a two-layer feed-forward network with a Softmax function in the output layer. This architecture aims to test whether an attention method based on a latent representation of the company that focuses on each time series independently can outperform a classical RNN setting.

Figure 6 summarizes the main differences between the two architectures. The source code for the Multi-head LSTM is publicly available on GitHub [2]

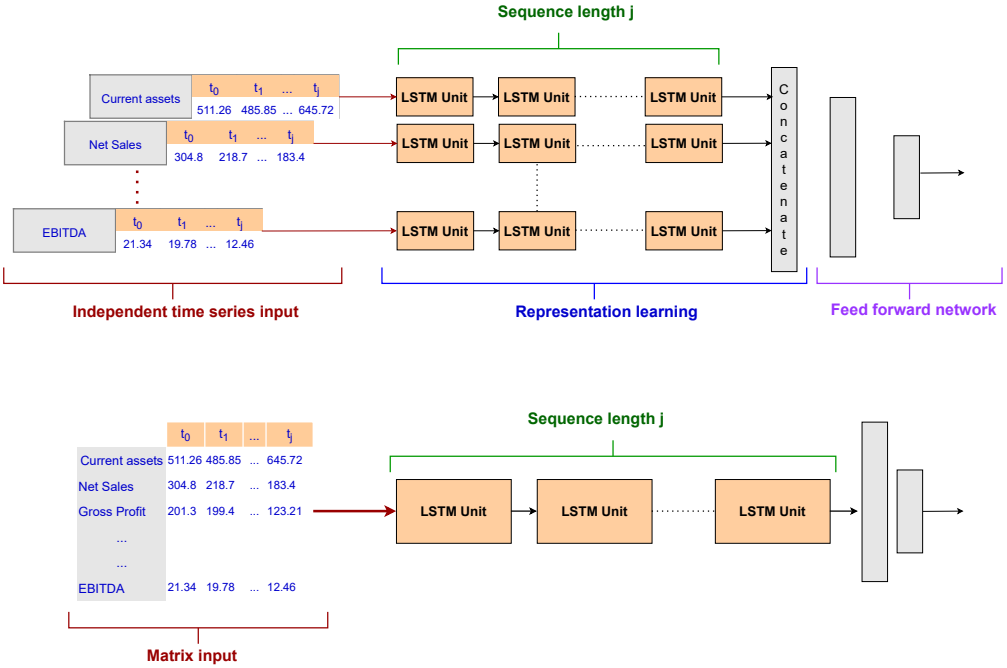

Figure 6. (Top image): The Multi-head LSTM setting where each financial time-series serves as the input to a different shorter and smaller LSTM. The representation learned by each head is then concatenated and exploited as the input to a subsequent feed-forward network. (Bottom image): The classical RNN setting where the input for our bankruptcy task is a single matrix with 18 rows and j columns.

[2] https://github.com/sowide/Multi-head_LSTM_for_bankruptcy-prediction

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 17 January 2024 doi:10.20944/preprints202401.1290.v1

8. Results

In this section, we present the results we achieved on bankruptcy prediction using the two RNN architectures presented in Section 7. Firstly, we compare the RNNs with the best model found in the preliminary experiments presented in Section 5 using the validation set (2012-2014), and, finally, we show the results obtained on the previously unseen test set (2015-2018) to assess the generalization ability of our models.

8.1. Metrics

We implemented the bankruptcy prediction as a binary prediction task where the positive class (1) indicates bankruptcy in the next year while the negative class (0) means that a company has been classified as healthy in the next year. In order to compare our models and prove their effectiveness, we used different metrics that take into account the imbalanced condition of the validation and test sets. Consider the following quantities for the default prediction:

- **True Positive (TP)**: The number of actually defaulted companies that have been correctly predicted as bankrupted
- **False Negative (FN)**: The number of actually defaulted companies that have been wrongly predicted as healthy firms.
- **True Negative (TN)**: The number of actually healthy companies that have been correctly predicted as healthy
- **False Positive (FP)**: The number of actually healthy companies that have been wrongly predicted as bankrupted by the model.

Since the validation and test set are both imbalanced with a prevalence of healthy companies, we did not compare the models in terms of model accuracy. Indeed, the proportion of correct matches would be ineffective in assessing the model performance. Instead, we computed each class's Precision, Recall, and F_1 scores. This is highlighted because, in predicting bankruptcy, an error has a different cost depending on the class that has been incorrectly predicted. The cost of predicting a company going into default as healthy is much higher than the cost of predicting a company that will default as healthy. In light of this, the precision achieved for a class is the accuracy of that class' predictions. The Recall (Sensitivity) is the ratio of the class instances that are correctly detected as such by the classifier. The F_1 score is the harmonic mean of precision and recall: whereas the regular mean treats all values equally, the harmonic one gives much weight to low values. Consequently, we obtain a high F_1 score for a certain class only if its precision and recall are high. Equations 3,4, and 5 report how these quantities are computed for the positive class. The definition for the negative class is exactly the same by inverting positives with negatives.

$$Precision = \frac{TP}{(TP + FP)} \tag{3}$$

$$Recall = \frac{TP}{(TP + FN)} \tag{4}$$

$$F_1 score = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} \tag{5}$$

Moreover, we reported three global metrics for the classifier that have been selected because they enable an overall evaluation of the classifier on both classes without being influenced by the dataset imbalance:

- **The Area Under the Curve (AUC)** measures the ability of a classifier to distinguish between classes and is used as a summary of the Receiver Operating Characteristic (ROC) Curve. The ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 17 January 2024 doi:10.20944/preprints202401.1290.v1

- **The macro F_1 score** is computed as the arithmetic mean of the F_1 score of all the classes.
- **The micro F_1 score** is used to assess the quality of multi-label binary problems. It measures the F_1 score of the aggregated contributions of all classes giving the same importance to each sample.

Finally, we used two other metrics that are often evaluated in Bankruptcy prediction models. Because bankruptcy is a rare event, using the classification accuracy to measure a model's performance can be misleading since it assumes that type I error (Eq 6) and type II error (Eq 7) is equally costly. Actually, the cost of false negatives is much greater than the cost of false positives for a financial institution. In light of this, we explicitly computed and reported type I and type II errors and compared the models focusing in particular on type II and Recall of the positive class.

$$Type \quad I \quad error = \frac{FP}{TN + FP} \tag{6}$$

$$Type \quad II \quad error = \frac{FN}{TP + FN} \tag{7}$$

8.2. LSTMs training and validation

The two LSTM architectures have been trained with exactly the same parameters for a fair comparison. The main hyper-parameters are the following:

- Epochs = 1000
- Learning rate =10^{-4}
- Batch size = 32 (Default Keras value)

Moreover, we used the Early-stopping technique to prevent the network from overfitting, and we used the validation set to select the hyper-parameters. In particular, to deal with the imbalance of the training set, we performed 500 runs for each LSTM by considering a randomly balanced training set every time.

After that, we first compared the single-input LSTM, the Multi-head LSTM, and the previous results achieved with Random Forest and CatBoost on the same validation set. Figure 8 shows the model comparison for the temporal windows 4 and 5. Table 5 summarizes this result in terms of the average AUC achieved in the 500 runs. For each run, the models' weights are randomly initialized. In light of this result, it is clear that, at least in terms of AUC, the recurrent network-based deep-learning models outperform traditional classifiers.

Table 5. Average and max AUC achieved on the validation set (2012-2014) by Random Forest, CatBoost, Single LSTM and the Multi-head LSTM. The average training time for each model is also reported for WL=4 and WL=5.

	Avg AUC	Max AUC	Avg Training time [s]		Avg AUC	Max AUC	Avg Training time [s]
Multi-head LSTM	0.813356	0.837	103.86		0.79	0.828	82.60
LSTM	0.8026	0.8415	51.80		0,7928	0,864	42.86
Random Forest	0.75	0.777	1.035		0.762	0.794	1.062
CatBoost	0.731	0.767	7.388		0.750	0.797	7.432
	WL=4				**WL =5**		

8.3. Statistical analysis

The experiments described in the previous section and presented in Table 5 have been computed as an average over 500 different runs and show that our model achieves much better AUCs with respect to the other models (LSTM, Random Forest and CatBoost). We further analyzed these results to prove our model's performance is statistically significant. According to [44], a common way to test whether the difference between two classifiers' results over different datasets or runs is non-random is to compute a paired t-test which checks whether the difference of their average performance is significantly different from zero.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 17 January 2024 doi:10.20944/preprints202401.1290.v1

However, one of the t-test's requirements is that the differences between the two random variables compared are distributed normally. As we show in Figure 7 all the performance differences between each model and our Multi-head LSTM are not normally distributed, for this reason, according to [45] we used the Wilcoxon Signed-Rank test (WSR)([46]), a non-parametric alternative to the paired t-test which ranks the differences in performances of two classifiers for each run, ignoring the signs, and compares the ranks for the positive and negative differences. We set to 0.05 the p-value under which we reject the null hypothesis (the two distributions have the same median, and thus, the performance difference can be considered random).

In our case, we achieved a $p - value$ equal to 0 for each comparison with the Multi-head LSTM over the same 500 runs where models have been trained using the same balanced training set, and the AUC has been evaluated over the same unbalanced validation set. In light of these results, we can conclude that our model shows better performance in the period between 2012-2014. Finally, since the main goal of this analysis is to build a model that is able to generalize on unseen samples that have never been used during the design phase, we report, in the next section, our final analysis of the performances of the best models on the test-set to better prove the benefit of our approach.

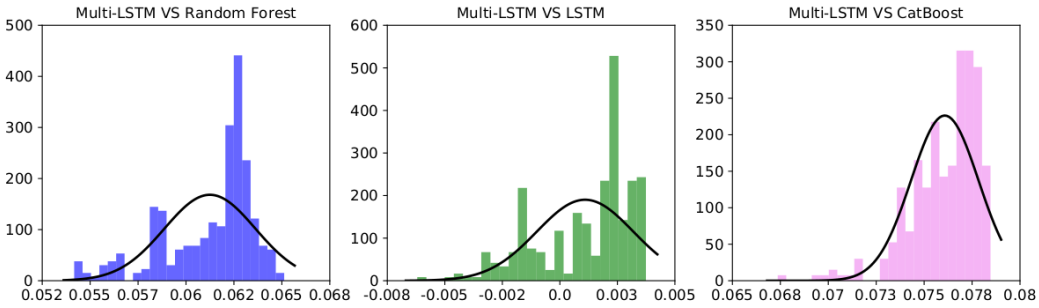

Figure 7. Performance score distributions with respect to our Multi-LSTM model on the 500 runs over the validation set. Since none of the random variables is normally distributed, we decided to use the WSR test (See Section 7.3).

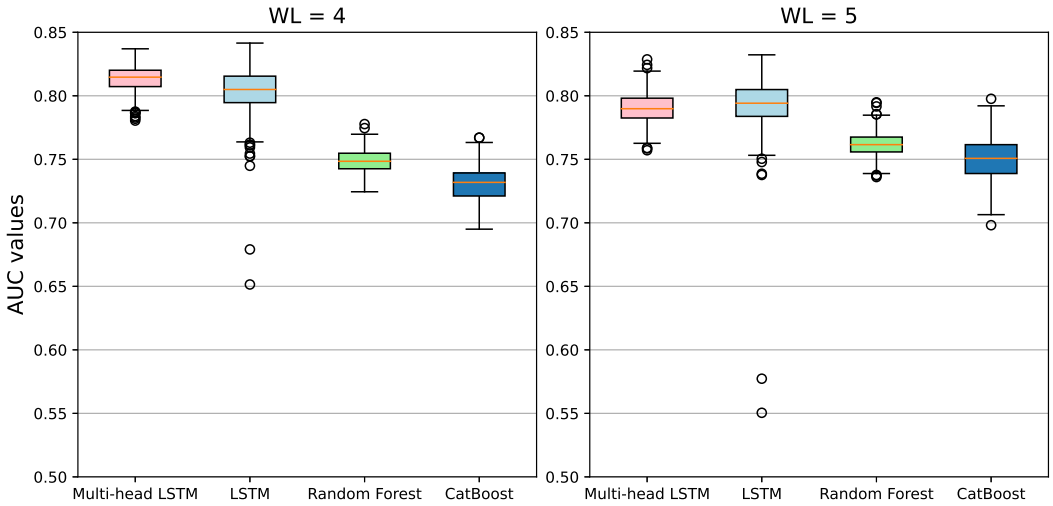

Figure 8. The box-plot shows the locality, spread, and skewness groups of AUC values through their quartiles achieved for each model in 500 different runs with a different balance training set. The lines (whiskers) extending from the boxes indicate the variability outside the upper and lower quartiles. The orange line represents the median between the first and third quartiles. The circular points (Fliers) are those past the end of the whiskers.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 17 January 2024 doi:10.20944/preprints202401.1290.v1

9. Final analysis on the test set

In light of the results achieved on the validation set, we selected the optimal single-input LSTM, the optimal Multi-head LSTM and the optimal Random Forest. We defined the optimal model as the one that offers these three conditions:

- Lowest difference between training and validation loss to ensure that the highest AUC is not achieved as a consequence of overfitting.
- Lowest validation loss
- Highest AUC on the validation set

We experimented with these best models on the previously unseen test set (companies between 2015 - 2018). We again compared the single-input LSTM, the Multi-head, and the Random Forest classifier. Figure 9 shows the models' comparison in terms of AUC. In Table 6, we report detailed results for each model in terms of Recall on the bankruptcy class, type I and II error as well as the micro and macro F_1 scores. As expected from the previous results obtained for the AUC on the validation set, the best model is still the Multi-head LSTM which achieves the best result on the test set. In addition, to gain a deeper insight into our model's performance on the test set, we leveraged the BAC (balanced accuracy) metric, which is the arithmetic mean of sensitivity and specificity.

Since the model shows a really high precision over the healthy class 6, which is also the majority class in the validation and test sets, the slope differences are probably due to the higher number of correct predictions in the validation set during learning. However, it is possible to observe that no overfitting or underfitting phenomena affect our model, as also shown by the good results over the unseen test set (Figure 9).

Therefore, we can conclude that RNNs have an impact on bankruptcy prediction performance. However, the attention model induced by the Multi-head LSTM seems to achieve better results for all the metrics (AUC, micro, and macro F_1 scores) with a temporal window equal to 4 years. On the other hand, considering the minimization of the False Negative rate, the models with WL=5 reach, in general, the lowest Type II error.

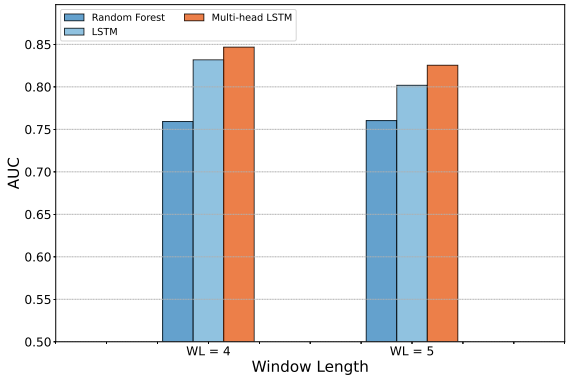

Figure 9. AUC values on the test-set (2015-2018) for Random Forest, Single-input LSTM and the Multi-head LSTM by varying the temporal window (WL).

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 17 January 2024 doi:10.20944/preprints202401.1290.v1

Table 6. Overall results on the test set (2015-2018) with the Single-input LSTM and the Multi-head LSTM. Rec means Recall and Pr means Precision. Best results are achieved on WL=4 with the Multi-head LSTM.

	LSTM	Multi-head LSTM		LSTM	Multi-head LSTM
TP	88	75		89	71
TN	1233	2158		1071	2085
FN	8	21		2	20
FP	1591	666		1573	559
AUC score	0.832	0.847		0.802	0.825
BAC	0.677	0.773		0.692	0.784
micro-f1	0.772	0.797		0.777	0.793
macro-f1	0.53	0.55		0.528	0.542
I Error	21.74	18.45		22.65	20.5
II Error	23.95	28.125		21.97	21.97
RecBankruptcy	0.76	0.71		0.78	0.78
Pr Bankruptcy	0.106	0.117		0.106	0.115
Rec Healthy	0.782	0.815		0.773	0.795
Pr Healthy	0.989	0.988		0.99	0.99
	WL = 4			**WL = 5**	

9.1. Further Analysis on the Test Set

Understanding the distribution and the relationships within the feature space is crucial for gaining insights into the behavior of a model. Among the methodologies employed for representing high-dimensional data, one of the most powerful approaches is the t-distributed stochastic neighbor embedding (T-SNE)[13]. T-SNE preserves the non-linearity structure of data points and tends to maintain the relative distances between neighboring points, which can reveal clusters and patterns. This analysis reported in this subsection aims to study the distribution of both healthy and bankrupt firms in a reduced dimensional space, focusing on the capabilities of Multi and single LSTMs of identifying an optimal decision boundary.

We first leveraged T-SNE on the original input to visualize the data in a two-dimensional space while preserving the pairwise similarities between data points. Each data point represents a firm, allowing us to gain an intuitive understanding of its inherent structures. The results are shown in Figure 10.(a). We performed an analysis using the best window lengths identified on the validation set (WL=4 and WL=5). We can infer that bankruptcy-prone firms usually do not cluster in a specific space region of the feature space of the input. This observation highlights how challenging is to classify bankrupt companies because there is some degree of overlap with healthy ones.

Following the outcomes achieved from the T-SNE analysis of the original test set, we focused on how the LSTM networks represent each firm in their latent space before classifying them, depicted in Figure 10. These snapshots offer some insights into the decision boundaries identified by the two recurrent architectures. Decision boundaries for LSTM are depicted in Figure 10.b and for Multi-head networks in Figure 10.c.

Our analysis reveals that these embeddings effectively distribute firms, with bankrupt firms forming distinct clusters. This demonstrates the models' ability to capture meaningful patterns in the data, particularly concerning bankruptcies. The models consistently demonstrated the ability, whether the window length is equal to 4 or 5, to discriminate between healthy and bankrupt firms in a two-dimensional environment. The models' constant ability to identify significant patterns linked to financial distress shows the models' robustness. By comparing the latent representations achieved with the single LSTM with the ones with the Multi-head, it is evident that the Multi-head achieved a more scattered representation that led to smaller overlaps between healthy and bankrupt firms. This

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 17 January 2024 doi:10.20944/preprints202401.1290.v1

result is also in line with the results presented in Table 6 by checking the number of False positives (FP) achieved by the two models. Thanks to this, the Multi-head LSTM outperforms the single LSTM, in terms of AUC, BAC, and FP for both window lengths.

For this reason, we decided to further study the false positives with an additional analysis in the following section.

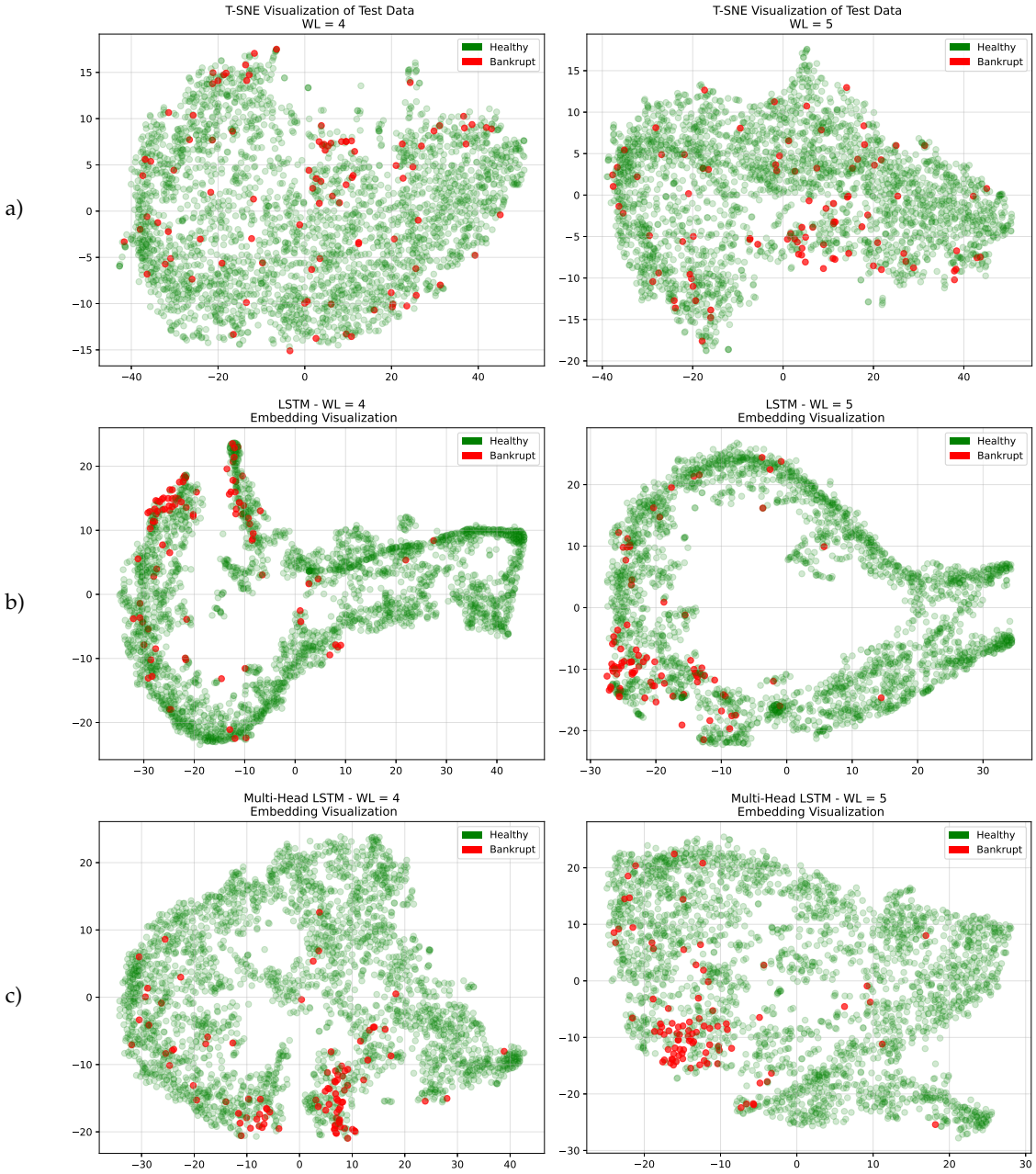

Figure 10. Two-dimensional t-distributed stochastic neighbor embedding (T-SNE) visualization of the test set. Each data point represents a firm. Subfigure (a) represents plain test data for WL 4 e 5, (b) displays embeddings from a single-input LSTM, and (c) shows embeddings from a Multi-head LSTM. These snapshots offer insights into the data's intrinsic structure and the modeling impact on representation.

9.2. False Positive Analysis

In bankruptcy prediction, an error has a different cost depending on the class that has been incorrectly predicted. The cost of predicting a company going into default as healthy (FN) is much

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 17 January 2024 doi:10.20944/preprints202401.1290.v1

higher than the cost of predicting a healthy one as bankrupt (FP). However, both networks achieved a small number of false negatives, but a considerable number of false positives that affect the performance. Moreover, in light of the evidence presented in the previous section about the different levels of overlap among bankrupt and healthy firms achieved by the multi and single LSTMs, we decided to further analyze the latent representation (embedding) of the false positives achieved with both networks, to better prove the benefits introduced by the multi-head LSTM.

For this analysis, we leveraged again the T-SNE dimensionality reduction to display the distribution of false positives. The results are depicted in Figure 11 for window lengths 4 and 5.

Examining the embedding obtained from both models, one may observe how the false positives are close to the bankruptcy clusters presented in the previous section with a considerably smaller overlap ratio with healthy firms in the case of the Multi-head.

Furthermore, to complete our analysis, we considered how the false positives are distributed across to the financial industries since the economic conditions and market dynamics can significantly impact the behavior of businesses within a particular industry. With this aim, we leveraged the SIC codes presented in Section 4 to understand if industry-specific factors can contribute to misclassification.

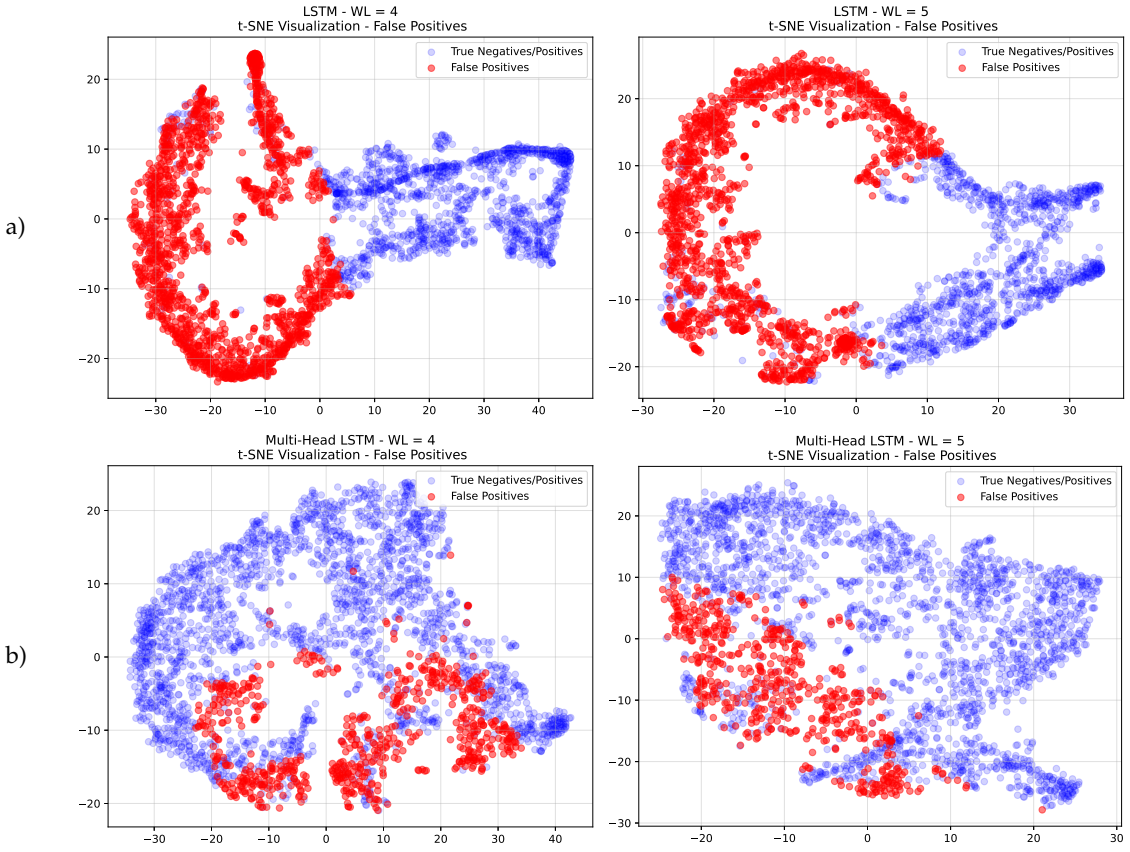

Figure 11. Visualization of False Positives Using T-SNE: false positive distribution in a high-dimensional feature space visualized using T-SNE. Subfigure (a) represents the embeddings from a single-input LSTM, and (b) shows the embeddings from a Multi-head LSTM.

Table 7 presents the false positive, for both models, across different SIC divisions under different test set conditions (WL = 4 and WL = 5), as well as the respective percentage. We can infer the following observations:

- Across all divisions, the Multi-head model outperforms LSTM, for both window lengths, in the number of false positives. This shows how much better the Multi-head LSTM is in classifying and identifying between alive and bankrupt firms.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 17 January 2024 doi:10.20944/preprints202401.1290.v1

- Examining the variance of false positives with an emphasis on window length reveals an interesting pattern. When considering the Multi-head LSTM, the more the window length grows, the lower the percentage of false positives. This can be explained by the Multi-head LSTM's capacity to identify structures and patterns that can be missed when the window is set to 4. On the opposite, since the single LSTM's performance has more variance, no definitive conclusions can be made about it, making it difficult to understand how false positives change across various time windows. This is because there are equal amounts of positive and negative variations.

These observations regarding the sectors and spatial distribution of false positives provide important guidance for improving model parameters, maximizing predictive precision, and eventually improving our models' applicability in bankruptcy prediction tasks.

Table 7. Divisions' Distribution in the Test Set for each Model - This table provides an overview of the distribution of companies across various divisions within the test set, including the total number of companies belonging to each category. However, for LSTM and Multi-head LSTM models, only instances of false positives are reported, along with their distribution among different divisions. Additionally, percentages are presented to express the ratio of false positives in relation to the total data present in the test set.

Division	Test set		LSTM (False Positive)		Multi-head LSTM (False Positive)	
	WL = 4	WL = 5	WL = 4	WL = 5	WL = 4	WL = 5
A	17	15	12 (70%)	9 (60 %)	4 (23 %)	3 (20 %)
B	152	142	84 (55%)	79 (52 %)	34 (22 %)	29 (20 %)
C	30	28	13 (43%)	14 (50 %)	5 (17 %)	6 (21 %)
D	1476	1382	824 (55%)	790 (57 %)	326 (22 %)	284 (21 %)
E	254	248	107 (42 %)	139 (56 %)	77 (30 %)	44 (18 %)
F	106	99	73 (69 %)	71 (72 %)	34 (32 %)	28 (28 %)
G	213	202	134 (62 %)	131 (65 %)	104 (49 %)	95 (47 %)
H	66	63	34 (52 %)	32 (51 %)	12 (6%)	5 (8 %)
I	606	556	310 (51 %)	308 (55 %)	70 (11 %)	65 (12 %)

10. Conclusions

In this paper, we have proposed a Multi-head LSTM neural network to assess corporate bankruptcy. According to the experimental analysis on the test-set, this model outperforms a single-input LSTM with the same hyper-parameters and architecture as well as the other traditional ML models. The better forecasting performance of the Multi-head LSTM also proves that modelling each accounting time-series independently with an attention-head contributes to better-identifying companies that are likely to face default events (Highest Recall and lowest type II error). This is also evident in the analysis of false positives presented in the experimental part. Moreover, we can finally argue that using accounting data for the four most recent fiscal years leads to better performance when predicting the likelihood of corporate distress. Experiments have been conducted on a dataset composed of accounting variables from 8262 different American companies over the period 1999-2018 for a total of 78682 firm-year observations. This dataset has been made public so that it can be used as a benchmark for future studies. Future developments will involve exploiting textual disclosures from financial reports in conjunction with this model. Moreover, it should be possible to predict defaults only in specific sectors by adding macroeconomic variables such as sustainability, interest rates, sovereign risk, and credit spread. Furthermore, our models provide predictions over a single period, not the survival probabilities over time. In future work, multi-period models can be incorporated to also reach that goal.

 doi:10.20944/preprints202401.1290.v1

References

1. Danilov, C.; Konstantin, A. Corporate Bankruptcy: Assessment, Analysis and Prediction of Financial Distress, Insolvency, and Failure. *Analysis and Prediction of Financial Distress, Insolvency, and Failure (May 9, 2014)* **2014**.
2. Ding, A.A.; Tian, S.; Yu, Y.; Guo, H. A class of discrete transformation survival models with application to default probability prediction. *Journal of the American Statistical Association* **2012**, *107*, 990–1003.
3. Altman, E.I. Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *The journal of finance* **1968**, *23*, 589–609.
4. Wang, G.; Ma, J.; Huang, L.; Xu, K. Two credit scoring models based on dual strategy ensemble trees. *Knowledge-Based Systems* **2012**, *26*, 61–68.
5. Wang, G.; Ma, J.; Yang, S. An improved boosting based on feature selection for corporate bankruptcy prediction. *Expert Systems with Applications* **2014**, *41*, 2353–2361.
6. Zhou, L.; Lai, K.K.; Yen, J. Bankruptcy prediction using SVM models with a new approach to combine features selection and parameter optimisation. *International Journal of Systems Science* **2014**, *45*, 241–253.
7. Geng, R.; Bose, I.; Chen, X. Prediction of financial distress: An empirical study of listed Chinese companies using data mining. *European Journal of Operational Research* **2015**, *241*, 236–247.
8. Alfaro, E.; García, N.; Gámez, M.; Elizondo, D. Bankruptcy forecasting: An empirical comparison of AdaBoost and neural networks. *Decision Support Systems* **2008**, *45*, 110–122.
9. Bose, I.; Pal, R. Predicting the survival or failure of click-and-mortar corporations: A knowledge discovery approach. *European Journal of Operational Research* **2006**, *174*, 959–982.
10. Tian, S.; Yu, Y.; Guo, H. Variable selection and corporate bankruptcy forecasts. *Journal of Banking & Finance* **2015**, *52*, 89–100.
11. Wanke, P.; Barros, C.P.; Faria, J.R. Financial distress drivers in Brazilian banks: A dynamic slacks approach. *European Journal of Operational Research* **2015**, *240*, 258–268.
12. du Jardin, P. A two-stage classification technique for bankruptcy prediction. *European Journal of Operational Research* **2016**, *254*, 236–252.
13. Van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *Journal of machine learning research* **2008**, *9*.
14. Taffler, R.J.; Tisshaw, H. Going, going, gone–four factors which predict. *Accountancy* **1977**, *88*, 50–54.
15. Kralicek, P. *Fundamentals of finance: balance sheets, profit and loss accounts, cash flow, calculation bases, financial planning, early warning systems*; Ueberreuter, 1991.
16. Beaver, W.H. Financial ratios as predictors of failure. *Journal of accounting research* **1966**, pp. 71–111.
17. Ohlson, J.A. Financial ratios and the probabilistic prediction of bankruptcy. *Journal of accounting research* **1980**, pp. 109–131.
18. Altman, E.I.; Hotchkiss, E.; Wang, W. *Corporate financial distress, restructuring, and bankruptcy: analyze leveraged finance, distressed debt, and bankruptcy*; John Wiley & Sons, 2019.
19. Schönfeld, J.; Kuděj, M.; Smrčka, L. Financial health of enterprises introducing safeguard procedure based on bankruptcy models. *Journal of Business Economics and Management* **2018**, *19*, 692–705.
20. Moscatelli, M.; Parlapiano, F.; Narizzano, S.; Viggiano, G. Corporate default forecasting with machine learning. *Expert Systems with Applications* **2020**, *161*, 113567.
21. Danenas, P.; Garsva, G. Selection of support vector machines based classifiers for credit risk domain. *Expert systems with applications* **2015**, *42*, 3194–3204.
22. Tsai, C.F.; Hsu, Y.F.; Yen, D.C. A comparative study of classifier ensembles for bankruptcy prediction. *Applied Soft Computing* **2014**, *24*, 977–984.
23. Barboza, F.; Kimura, H.; Altman, E. Machine learning models and bankruptcy prediction. *Expert Systems with Applications* **2017**, *83*, 405–417.
24. Nanni, L.; Lumini, A. An experimental comparison of ensemble of classifiers for bankruptcy prediction and credit scoring. *Expert systems with applications* **2009**, *36*, 3028–3033.
25. Kim, M.J.; Kang, D.K. Ensemble with neural networks for bankruptcy prediction. *Expert systems with apphttps://www.osha.gov/data/sic-manuallications* **2010**, *37*, 3373–3379.
26. Wang, G.; Hao, J.; Ma, J.; Jiang, H. A comparative assessment of ensemble learning for credit scoring. *Expert systems with applications* **2011**, *38*, 223–230.
27. Lombardo, G.; Pellegrino, M.; Adosoglou, G.; Cagnoni, S.; Pardalos, P.M.; Poggi, A. Machine Learning for Bankruptcy Prediction in the American Stock Market: Dataset and Benchmarks. *Future Internet* **2022**, *14*, 244.

28. Mossman, C.E.; Bell, G.G.; Swartz, L.M.; Turtle, H. An empirical comparison of bankruptcy models. *Financial Review* **1998**, *33*, 35–54.
29. Duan, J.C.; Sun, J.; Wang, T. Multiperiod corporate default prediction—A forward intensity approach. *Journal of Econometrics* **2012**, *170*, 191–209.
30. Kim, H.; Cho, H.; Ryu, D. Corporate default predictions using machine learning: Literature review. *Sustainability* **2020**, *12*, 6325.
31. Vochozka, M.; Vrbka, J.; Suler, P. Bankruptcy or success? the effective prediction of a company's financial development using LSTM. *Sustainability* **2020**, *12*, 7529.
32. Kim, H.; Cho, H.; Ryu, D. Corporate bankruptcy prediction using machine learning methodologies with a focus on sequential data. *Computational Economics* **2022**, *59*, 1231–1249.
33. Gruslys, A.; Munos, R.; Danihelka, I.; Lanctot, M.; Graves, A. Memory-efficient backpropagation through time. *Advances in Neural Information Processing Systems* **2016**, *29*.
34. Graves, A. Supervised sequence labelling. In *Supervised sequence labelling with recurrent neural networks*; Springer, 2012; pp. 5–13.
35. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural computation* **1997**, *9*, 1735–1780.
36. Cho, K.; van Merriënboer, B.; Bahdanau, D.; Bengio, Y. On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. *Syntax, Semantics and Structure in Statistical Translation* **2014**, p. 103.
37. Adosoglou, G.; Lombardo, G.; Pardalos, P.M. Neural network embeddings on corporate annual filings for portfolio selection. *Expert Systems with Applications* **2021**, *164*, 114053.
38. Adosoglou, G.; Park, S.; Lombardo, G.; Cagnoni, S.; Pardalos, P.M. Lazy Network: A Word Embedding-Based Temporal Financial Network to Avoid Economic Shocks in Asset Pricing Models. *Complexity* **2022**, *2022*.
39. Campbell, J.Y.; Hilscher, J.; Szilagyi, J. In search of distress risk. *The Journal of Finance* **2008**, *63*, 2899–2939.
40. Structure, S.I.C.S.M.D. https://www.osha.gov/data/sic-manual.
41. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.Y. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems* **2017**, *30*.
42. Dorogush, A.V.; Ershov, V.; Gulin, A. CatBoost: gradient boosting with categorical features support. *arXiv preprint arXiv:1810.11363* **2018**.
43. Yang, S.; Yu, X.; Zhou, Y. Lstm and gru neural network performance comparison study: Taking yelp review dataset as an example. 2020 International workshop on electronic communication and artificial intelligence (IWECAI). IEEE, 2020, pp. 98–101.
44. Demšar, J. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine learning research* **2006**, *7*, 1–30.
45. Benavoli, A.; Corani, G.; Demšar, J.; Zaffalon, M. Time for a change: a tutorial for comparing multiple classifiers through Bayesian analysis. *The Journal of Machine Learning Research* **2017**, *18*, 2653–2688.
46. Rey, D.; Neuhäuser, M. Wilcoxon-signed-rank test. In *International encyclopedia of statistical science*; Springer, 2011; pp. 1658–1659.