DMDP: A Dynamic Multi-source Default Probability Prediction Framework

Yi Zhao¹ · Yanyan Shen¹ · Yong Huang²

Received: 11 December 2018 / Accepted: 15 March 2019 / Published online: 27 March 2019
© The Author(s) 2019

Abstract

In this paper, we propose a dynamic forecasting framework, named DMDP (dynamic multi-source default probability prediction), to predict the default probability of a company. The default probability is a very important factor to assess the credit risk of listed companies on a stock market. Aiming at aiding financial institutions in decision making, our DMDP framework not only analyzes financial data to capture the historical performance of a company, but also utilizes long short-term memory model to dynamically incorporate daily news from social media to take the perceptions of market participants and public opinions into consideration. The study of this paper makes two key contributions. First, we make use of unstructured news crawled from social media to alleviate the impact of financial fraud issue made on default probability prediction. Second, we propose a neural network method to integrate both structured financial factors and unstructured social media data with appropriate time alignment for default probability prediction. Extensive experimental results demonstrate the effectiveness of DMDP in predicting default probability for the listed companies in mainland China, compared with various baselines.

Keywords Default probability prediction · Time series · Multi-source data

1 Introduction

Granting loans to potential borrowers is considered as one of the core business activities for financial institutions. Though loans can help these institutions gain profits, they may also cause huge loss, which is often known as financial risks. For instance, the 2008 financial crises resulted in huge losses globally. Hence, nowadays financial institutions devote more and more attention to evaluating risks before granting loans. In particular, most financial institutions are now cognizant of the need to adopt rigorous credit risk assessment models when determining whether or not to grant loans to specific borrowers.

In the early stage, various classical statistical approaches [3, 7] such as logistic regression [10], multivariate adaptive regression splines [19, 20] and linear discriminate analysis [1, 2] were proposed for credit risk prediction. However, statistical approaches are typically based on certain assumptions, e.g., multivariate normality for independent variables and non-multicollinearity of data, which make the proposed solutions theoretically invalid for finite samples [16]. Fortunately, with the advent of machine learning algorithms, many studies demonstrated that neural network (NN) [19, 31, 38], support vector machine (SVM) [6, 11, 13, 15, 23], decision tree (DT) [4, 32], random forest (RF) [5, 29] and Naive Bayes (NB) [3, 25, 33] can be used to build credit scoring models for measuring default risks with high accuracy. Some practical works [8, 26, 27, 30, 34] have focused on classifier ensembles and demonstrated that ensemble classifiers constantly outperform single classifier in terms of prediction accuracy.

Although machine learning methods can automatically learn hidden and critical factors based on the past observations and do not require specific prior assumption, the performance of these supervised methods greatly relies on the quality of training data. To be more specific, the accuracy of the risk evaluation results is typically affected by

✉ Yanyan Shen
sheny@sjtu.edu.cn

Yi Zhao
zhaoyizhaoyi@sjtu.edu.cn

Yong Huang
yonghuang826@yahoo.com

¹ Shanghai Jiao Tong University, Shanghai, China

² Hong Kong University of Science and Technology,
Clear Water Bay, Hong Kong

the trustworthiness and the comprehensiveness of the available historical data. At corporation level, the data for default probability prediction involve basic financial indicators such as industry section, geographical area and financial statements. One of the major problems encountered in adopting financial indicators for credit risk assessment is that the companies might commit accounting fraud in order to artificially improve the appearance of the financial reports, which impedes the effectiveness of the learned prediction models. Furthermore, almost all the prior works focus on static models that leverage the most recent indicator values for prediction, but do not consider the temporal trend of indicators that is valuable to reflect the long-term financial status of a company.

In practice, for credit risk assessment, most lenders take advantage of the information from social media networks, such as Twitter and Facebook, to decide whether their potential borrowers are creditworthy. However, probably due to the difficulty in grabbing daily financial news, we do not find any study that leverages social media analysis to improve the prediction performance in terms of the default probabilities for companies. It is important to notice that besides financial statements, social media data contain subjective appraisals of the firm's prospects which are discriminate indicators to assess the default probability of a publicly traded company.

In this paper, we propose a dynamic multi-source default probability predication framework named DMDP, to predict the default probability of the listed companies. In order to relieve the exertion of potentially flawed financial data and enhance the accuracy of machine learning-based default probability prediction method, we make use of social media data to trace the latest development of a company. Through mining the public opinions (randomly updated) as well as the financial indicators (periodically updated), we can make a comprehensive evaluation of the observed companies in terms of their default probabilities. More importantly, aiming at prior evaluation for group loans, our framework is designed to handle the evolving data and continuously produce default probability prediction results based on the up-to-date company statuses, thus allowing financial institutions to make quick response when the borrowers experience a drastic market decline.

A preliminary version of this work was published in [37]. In this paper, we have made four important enhancements as follows:

- We have changed the frequency of news text updates, from quarterly updates to weekly updates. More frequent updates allow us to be aware of fine-grained changes of public opinions on the target company. Apparently, it is easier to identify subtle fluctuations of the public opinions within a shorter time window. It is important to note that more frequent updates require zooming into weekly

news updates will result in longer sequences, which typically calls for a more complicated model to learn the complex temporal dependencies.

- We have improved the way to represent the news text. In the preliminary work, we first concatenated all the related news text concerning the target company in a quarter. Then, we extracted the top 50 keywords to and calculated the average word embeddings of the keywords to represent the public opinion inherent in the news text of the target company during the quarter. In this work, we use the average word embeddings of the title of a news text to represent it, which we dub it as the title embedding. Then, we calculate the average title embedding out of all the news text concerning the target company during a week. This change is mainly due to the observation that the titles themselves are good abstractions of the news text produced by professional editors.
- We have improved the dataset splitting mechanism. In the preliminary work, we randomly sample 70%, 15% and 15% of the dataset as training, validation and test sets, respectively. In this work, we treat all data points before Quarter 3, 2016 as training data. Data points ranging from Quarter 3, 2016 to Quarter 4, 2016 are treated as validation data. Data from Quarter 1, 2017 on are treated as testing set. By splitting the data according to the time order, we can avoid the problem of future information leaking.
- To better model the asynchronous nature of news series, we have replaced the LSTM in the original paper with the Phased LSTM [22], which extends the LSTM unit by adding a new time gate and can process asynchronous time series data. We also adopted a recently proposed CNN-based Wavenet [28] to encode the news text time series. The results have shown that Wavenet has greatly improved AUC on our default behavior dataset.

The remainder of this paper is organized as follows. Section 2 presents preliminaries. Section 3 introduces our proposed framework with two major components: dynamic multi-source data alignment and neural network-based prediction. Section 4 provides the experimental results. We review the related works in Sect. 5 and conclude this paper in Sect. 6.

2 Preliminaries

In general, the objective of this study is to effectively distinguish “bad” corporations from “good” ones, which can be considered as a classification problem where a company is categorized to class “1” if it is predicted to be default, i.e., receive delisting risk warning (*ST). Otherwise, it will be labeled by “0.” In this work, we use sequences of historical

financial indicators and unstructured news data in the previous time periods to predict whether the target company will be default or not in the future.

2.1 Definitions

We first introduce the definitions used in this paper. Because the financial indicators and the target value are updated at quarterly frequency, we denote the financial indicators and the target value by quarter. During each time period (quarter) t , the information of a company c during that time period (quarter) t can be recorded in a tuple of $(FIN_t^c, TEXT_t^c, y_t^c)$, where FIN_t^c contains the values of a set of financial indicators collected at the end of period t . Because there can be multiple financial indicators observed, FIN_t^c could be a multi-dimensional vector. Similarly, because the news text information is updated at more granular frequency (by week), $TEXT_t^c$ could be a sequence with each one being a representation of the news text during the more granular observation period (week). y_t^c is the binary response variable to indicate whether the company is labeled (*ST) in the period.

2.1.1 Financial Indicators (FIN)

Financial indicators are commonly included in the financial statements, and they reflect important characteristics of an company. For instance, the indicator “cash flow to liabilities ratio” directly reflects a company’s ability to cover its liability within a time window and hence is critical when predicting the default probability of the company in near future. By similar logic, we extract several key financial indicators from financial statements of a company at the end of a financial period. Suppose we extract P indicators in each time period (quarter) t for a company c , we denote them by $FIN_t^c = \{X_{1t}^c, X_{2t}^c, \dots, X_{Pt}^c\}$.

2.1.2 News Data Representation ($TEXT$)

$TEXT$ stands for the embedding-based representation of news data that are relevant to the company during a period. The news is crawled from social media. Because the news is classified into each company directly, we can easily match them to corresponding companies. As described earlier, in a specific time period (quarter) t for company c , we can denote the more granular news text sequence as $TEXT_t^c = \{TEXT_{tk}^c\} = \{TEXT_{t1}^c, TEXT_{t2}^c, \dots, TEXT_{tK_t}^c\}$, where $k \in \{1, 2, \dots, K_t\}$ is the k th week during quarter t . $TEXT_{tk}^c$ is the representation of all the news texts related to the company c during the k th week in quarter t and is calculated by the average of all the title embeddings of news in that week.

2.1.3 Class Label (y)

y stands for the class label of the target company in a financial period. That is, $y = 1$ if the company receives delisting risk warning (*ST) during the period. Otherwise, $y = 0$. In this work, we do not take the sequence of the previously observed label values as the input for our prediction model as the discriminative power of the previous *ST values is quite limited.

2.2 Problem Statement

We now formally define the problem studied in this paper as follows.

Definition 1 (Problem Statement) Given a target company, an observation financial period (quarter) t , sequences of financial indicators FIN and news $TEXT$ before time period t (t is included) in chronological order, we want to develop a framework to predict the default probability of the company during time period $t + 1$. The predicted value could be 0 or 1, representing relatively low default probability or high default probability, respectively.

3 Methodology

Our DMDP framework consists of two major components: (1) dynamic multi-source data alignment and (2) neural network-based prediction model. Figure 1 provides the illustration of our DMDP framework.

3.1 Dynamic Multi-source Data Alignment

To predict whether a company c will be labeled (*ST) or not at time period $T + 1$, we extract the sequences of financial indicators $\{FIN_1^c, \dots, FIN_T^c\}$ and unstructured news representations $\{TEXT_1^c, \dots, TEXT_T^c\}$, where FIN_1^c and $TEXT_1^c$ are the inputs in the first financial period after company c gets listed. Because a listed company is required to publish its financial statements for each financial period according to regulation, the sequence of the historical financial indicators is complete. We denote the sequence of financial indicators for a company c before time period T by $\{FIN_t^c\} = \{X_{it}^c\}$, where $i \in \{1, \dots, P\}$ and $t \in \{1, \dots, T\}$. For the sequence of news representations, we perform preprocessing and alignment over the raw news data to deal with the problems of data missing and irregularity. On the one hand, the number of news within a specific financial period varies from one company to another. On the other hand, even for the same company, the number of relevant news released in different financial periods varies greatly, from zero to several dozens. For instance, a company may receive multiple news within

Fig. 1 DMDP framework

one day, while it might take weeks to get one relevant news for certain time period.

To handle the missing and irregularity problems of the news sequence, we apply the following method to preprocess the raw news data and align them properly. First, given each news text for a company c during week k of quarter t , we obtain the embedding of the title by averaging the pre-trained Word2Vec embeddings of the words in the title. We denote the embedding of n th news title as $title_{tkn}^c$ to represent news n , out of N_{tk}^c news for company c during week k of quarter t . Then, we calculate the average of the title embeddings for company c in that week as $TEXT_{tk}^c = \frac{1}{N_{tk}^c} \sum_{n=1}^{N_{tk}^c} title_{tkn}^c$. Formally, we denote the sequence of news text representation for a company c before time period T by

$$\{TEXT_t^c\} = \{TEXT_{tk}^c\} = \left\{ \frac{1}{N_{tk}^c} \sum_{n=1}^{N_{tk}^c} title_{tkn}^c \right\} \quad (1)$$

where $t \in \{1, 2, \dots, T\}$ is the quarter starting from the company get listed up to quarter T . $k \in \{1, 2, \dots, K_t\}$ is the week number during quarter t . N_{tk}^c is the number of news for company c during week k of quarter t . $n \in 1, 2, \dots, N_{tk}^c$ is the n th news text for company c during week k of quarter t . $title_{tkn}^c$ is the average of word embeddings of the words in title of n th article for company c during week k of quarter t and is d -dimensional given the d -dimension of the pre-trained word embeddings.

Note that not every company has related news in each week, and hence, we have to decide how to impute or align

the data for news text sequence. We have tried three ways to align such data. One is to impute zero vectors into the missing weekly representations of news for each company. To be specific, if a company c has news representation until week k in quarter t , we impute zero vector into all missing weeks before week tk . We call this alignment method **ZeroInputAlign**. The second way is to serialize the week number tk into a sequence and feed them into the model too. Since Phased LSTM can take time step as direct input into the model, we feed it in this way. We dub this alignment method as **TimeInputAlign**. The third way is to squeeze the text news sequence directly without considering the potential sparsity of the input. For instance, for a company c , which has only three records $\{TEXT_{11}^c, TEXT_{35}^c, TEXT_{T4}^c\}$, we directly input the sequence as $\{TEXT_1^c, TEXT_2^c, TEXT_3^c\}$ into the model. In this way, the time only determines the position in the sequence but does not serve as input to the model. We refer to this alignment method as **SqueezeInputAlign**. We would compare the performance of the three methods in our experiments.

Finally, we put the financial variables and news text time series into different parts of the model and concatenate them to predict. After aligning multi-source data, we obtain the final input to our prediction model $\mathbf{X}_T^c = (\{FIN_t^c\}, \{TEXT_{t_{\text{text}}}^c\})$, where $\{t_{\text{text}}\}$ is determined by the aligning method we use. We summarize the pseudocode for the multi-source data alignment in Algorithm 1.

Fig. 2 The architecture of the proposed neural prediction model

Algorithm 1: AlignMultiSourceData

Input : Sequence of financial indicators $\{FIN_1^c, \dots, FIN_T^c\}$; Sequence of news for a company.

Output: Aligned results for the two sequences, $\mathbf{X}_T^c = (\{FIN_t^c\}, \{TEXT_{t_{text}}^c\})$.

- 1 Calculate the average word embeddings for title of n -th news for company c during week k of quarter t from pre-trained word embeddings of all the words in the title as $title_{tkn}^c$.
- 2 Calculate the average of the title embeddings in each week as $\frac{1}{N_{tk}^c} \sum_{n=1}^{N_{tk}^c} title_{tkn}^c$, where N_{tk}^c is the number of news for company c during week k of quarter t .
- 3 Align the representation of financial indicators and that of the news representation. We denote it by $\mathbf{X}_T^c = (\{FIN_t^c\}, \{TEXT_{t_{text}}^c\})$, where t_{text} is determined by a specific align method.
- 4 **return** $\mathbf{X}_T^c = (\{FIN_t^c\}, \{TEXT_{t_{text}}^c\})$.

3.2 Neural Network-Based Default Probability Prediction Model

The architecture of our neural network-based model is illustrated in Fig. 2. In this section, we describe each layer of the model in details.

3.2.1 Input Layer

The first layer is the input layer, which contains the aligned sequences of financial indicators and news representations, during time periods $1, \dots, T$. Formally, the input layer is defined as $\mathbf{X}_T^c = (\{FIN_t^c\}, \{TEXT_{t_{text}}^c\})$.

3.2.2 Encode Layer

The second layer is to encode the financial variables time series and the news text time series, respectively. Because the news representations are updated at a weekly frequency while the financial variables are updated quarterly, the length of the former is much longer than the latter and it is much

harder to encode the news time series. To resolve this problem, we introduce Phased LSTM and Wavenet in addition to LSTM into our model. Phased LSTM [22] introduces a time gate to take time as an input directly and can thus deal with asynchronous time series. Wavenet is a CNN-based model with causal dilations [28], which has shown powerful expressive power in encoding audio, text. Next we will briefly describe the three models and compare their results in the experiments part.

Long short-term memory (LSTM) was proposed in [14]. It is a variant of vanilla RNN, which includes cell states with the gating mechanism. With different types of gates, LSTM is capable of controlling how much information flow through the cell state and thus is capable of learning long-term dependencies. In this work, we follow the implementation of LSTM used in [35]:

$$\mathbf{f}_t = \sigma(\mathbf{W}_f[\mathbf{h}_{t-1}; \mathbf{x}_t] + \mathbf{b}_f) \quad (2)$$

$$\mathbf{i}_t = \sigma(\mathbf{W}_i[\mathbf{h}_{t-1}; \mathbf{x}_t] + \mathbf{b}_i) \quad (3)$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o[\mathbf{h}_{t-1}; \mathbf{x}_t] + \mathbf{b}_o) \quad (4)$$

$$\mathbf{s}_t = \mathbf{f}_t \odot \mathbf{s}_{t-1} + \mathbf{i}_t \odot \tanh(\mathbf{W}_s[\mathbf{h}_{t-1}; \mathbf{x}_t] + \mathbf{b}_s) \quad (5)$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{s}_t) \quad (6)$$

where $\mathbf{f}_t, \mathbf{i}_t, \mathbf{o}_t, \mathbf{s}_t, \mathbf{h}_t$ refer to the forget gate, input gate, output gate, memory cell state and hidden state, respectively; x_t refers to the input to the LSTM unit; σ represents the standard sigmoid function; $\odot$ denotes the Hadamard product; $\mathbf{W}$ and $\mathbf{b}$ terms are the weight matrices and bias terms to be learned, respectively.

The Phased LSTM [22] model extends the LSTM model by adding a time gate k_t and uses the gate to mediate the cell updates. The time gate is calculated as follows:

$$\phi_t = \frac{(t - s) \bmod \tau}{\tau} \quad (7)$$

$$k_t = \begin{cases} \frac{2\phi_t}{r_{on}}, & \text{if } \phi_t < \frac{1}{2}r_{on} \\ \frac{2-2\phi_t}{r_{on}}, & \text{if } \frac{1}{2}r_{on} < \phi_t < r_{on} \\ \alpha\phi_t, & \text{otherwise} \end{cases} \quad (8)$$

Then, the equations for cell updates in LSTM are as follows:

$$\tilde{c}_t = f_t \odot c_{t-1} + i_t \odot \sigma_c(x_t W_{xc} + h_{t-1} W_{hc} + b_c) \quad (9)$$

$$c_t = k_t \odot \tilde{c}_t + (1 - k_t) \odot c_{t-1} \quad (10)$$

$$\tilde{h}_t = o_t \odot \sigma_h(\tilde{c}_t) \quad (11)$$

$$h_t = k_t \odot \tilde{h}_t + (1 - k_t) \odot h_{t-1} \quad (12)$$

Wavenet [28] uses a stack of dilated causal convolutional layers to have very large receptive fields. In addition, residual connection is also used in the network to speed up convergence and enable training of deeper models. Specifically, given a time series $\mathbf{x}$, we can encode it with Wavenet as follows:

$$\mathbf{z} = \tanh(W_{f,k} * \mathbf{x}) \odot \sigma(W_{g,k} * \mathbf{x}) \quad (13)$$

where $*$ denotes a convolution operator, k is the layer index, f and g denote filter and gate, respectively, and W is a learnable convolution filter.

After encoding the news time series using one of these three models, we then extract the representation of the final time step and combine it with the hidden states from the LSTM on financial variable time series and pass the concatenated vector to the next layer to predict the default probability in the next financial period. In the experiments, we evaluate the performance of the three models on a real dataset.

3.2.3 Prediction Layer

At last, we feed the output from the LSTM layer to a fully connected layer. The output of the fully connected layer is

the predicted response value indicating whether the company c will be default in time period $T + 1$. Formally, we have:

$$\hat{y}_{T+1}^c = \mathbf{w} \mathbf{h}_T^c + b \quad (14)$$

where $\mathbf{w}$ is the weight vector and b is the bias to be learned.

3.3 Learning and Optimization

Our prediction problem is essentially a binary classification problem. Hence, we choose to use cross-entropy as the loss function, which is defined as follows.

$$\mathbf{L} = \sum_{\substack{c \in \{1, \dots, C\}, \\ T \in \mathbf{T}_c}} -y_{T+1}^c \log(\sigma(\hat{y}_{T+1}^c)) - (1 - y_{T+1}^c)(1 - \log(\sigma(\hat{y}_{T+1}^c))) \quad (15)$$

where y_{T+1}^c is the actual class label of company c at time period $T + 1$ and $\mathbf{T}_c$ is the collection of financial periods for company c after it gets listed. We use stochastic gradient descent and Adam optimizer [18] in our training process. The Adam optimizer dynamically adapts the learning rate and makes the convergence faster via adaptive estimates of the lower-order moments. It is computationally efficient with little memory requirements. Besides, it is invariant to diagonal rescaling of the gradients and is well suited for problems that involve data or parameters in large size.

4 Experiments

4.1 Experimental Settings

4.1.1 Datasets

We obtained the financial indicators for the listed companies using **tushare**¹ package. The dataset covers 30 financial indicators for 3125 listed companies in mainland China, from January 2014 to September 2017. It contains 32,420 records of financial indicators updated in each quarter, and thus, we have 29,295 records which has at least one historical quarterly record. The news data are crawled from Sina Finance² and contain more than 700,000 pieces of news, describing the historical financial performance and the public opinions of the observed companies. The news data are categorized by the companies and can be integrated with the financial variables by stock id. Among the 3125 companies, the vast

¹ <http://tushare.org>.

² <http://finance.sina.com.cn>.

Table 1 The numerical financial indicators used in our experiments

Category	Financial indicators
Cash flow ability	Cash flow to sales ratio
	Cash flow to net profit margin ratio
	Cash flow to liabilities ratio
	Cash flow ratio
Operation ability	Account receivable turnover
	Account receivable turnover days
	Inventory turnover
	Inventory turnover days
	Current asset turnover
	Current asset turnover days
Profitability	Return of equity(roe), EPS
	Net profit ratio, net profits
	Gross profit rate, business income
	Business income per share
Solvency	Current ratio, cash ratio
	Quick ratio
	Interest coverage ratio
	Shareholders equity ratio
Growth	Main business rate of growth
	Net profit rate of growth
	Net asset
	Total asset rate of growth
	EPS rate of growth
	Shareholders equity rate of growth

majority are in the “normal” class and only 209 companies bearing 238 delisting risk warnings are categorized as “relative high default risk.” To even up the extremely imbalanced classes in our dataset, we adopt random over-sampling before model training procedure. The financial variables are updated every 3 months according to the quarterly financial statements of the companies. We chose 30 critical financial indicators reflecting the statuses of the listed companies. The indicators are summarized in Table 1.

4.1.2 Preprocessing News Text

In the preliminary paper [37], we concatenated the news in one season and extracted top 50 keywords using TF-IDF for each company. In this paper, however, we make two changes. First, instead of concatenating news in a quarter into single text and extracting the keywords, we use the titles instead of the body of the news. This is based on the observation that the titles are mostly good abstractions of the news written by the professional editors. Second, we further change the granularity from quarterly base to weekly base. That is, for each company, we calculate the average of all the news titles embeddings to represent the news for the company in the week. By doing this, we expect that the updates of news

text could be more timely and we can better model the fluctuations of the public opinions for the stock. When calculating the average word embeddings in a title, we first adopt Jieba³ for Chinese word segmentation. Next we remove the stop words and look up the word embeddings from the pre-trained Word2Vec model provided by Facebook’s FastText module [17]. We represent the news text in week with a 300-dimensional embedding vector, which is the average of the all the news concerning the company in the week. After that, we remove the null value from the financial dataset and standardize the numerical data into range (0,1) with Max-Min Scalar. We align the two sequences of financial indicators and news representation using Algorithm 1.

4.1.3 Implementation Details

In the preliminary paper [37], we randomly partitioned the datasets into the training, validation and test sets. In this work, we split the data according to the time order to avoid using future information. Specifically, we treat all records ranging from Quarter 1 of 2014 to Quarter 2 of 2016 as training set. And data records ranging from Quarter 3 of 2016 to Quarter 4 of 2016 are labeled as validation set. And records since 2017 are treated as testing set. Such splitting mechanism yields 17, 509 records for training, 4, 781 for validation and 7, 005 records for testing set. To minimize the influence of the variability of the training set, model training process is repeated 10 times for each setting. We use Tensorflow to implement and train our prediction model. Generally, we test different learning rates ranging from 0.1 to 0.001. We also test different batch sizes. Besides, we evaluate the effects of different numbers of LSTM hidden units, i.e., {16, 32, 64, 128}. In our experiments, we try to lessen the influence of class imbalance using random over-sampling. We control the parameter “resampling ratio” in our experimental results, which stands for the percentage of positive samples in a training batch. We summarize our parameters in Table 2.

4.1.4 Compared Methods

In this work, we compare our DMDP framework with the baseline GAM model. Also we run our prediction model on *FIN* sequence or *TEXT* sequence only. The compared methods are described as follows:

- (1) *GAM* (generalized additive model) [24] is treated as the baseline in our study. In general, GAM is used to deal with time series data with a fixed window size. Enabling the discovery of a nonlinear fit between a variable and the response, the model makes use of the

³ <https://github.com/fxsjy/jieba>.

Table 2 Parameter ranges

Parameter	Description	Range
Learning rate	learning rate	{0.1, 0.01, 0.001}
Hidden units in LSTM (FIN)	Number of hidden units	{16, 32, 64, 128}
Hidden units in LSTM/Phased LSTM (TEXT)	Number of hidden units	{16, 32, 64, 128}
Resample ratio	The ratio of positive cases in a batch	{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}
Dilation in Wavenet	Number of dilation layers in Wavenet	{2, 4, 8, 16}

Table 3 Comparison of different methods. Note that each method was trained 5 times and we report their average performance and standard deviations (in brackets) for comparison

Framework	Model	Data	Accuracy	AUC
–	GAM	FIN	0.651(0.000)	0.550(0.000)
–	LSTM	FIN	0.891(0.070)	0.662(0.033)
–	LSTM	TEXT	0.991(0.010)	0.628(0.061)
–	Phased LSTM	TEXT	0.963(0.032)	0.583(0.060)
–	Wavenet	TEXT	0.979(0.016)	0.757(0.018)
DMDP	LSTM (FIN) + LSTM (TEXT)	FIN+TEXT	0.981(0.009)	0.682(0.042)
DMDP	LSTM (FIN) + Phased LSTM (TEXT)	FIN+TEXT	0.996(0.003)	0.607(0.133)
DMDP	LSTM (FIN) + Wavenet (TEXT)	FIN+TEXT	0.982(0.009)	0.761 (0.013)

Table 4 The results of different alignment methods

Alignment method	Model	Data	Accuracy	AUC
ZeroInputAlign	LSTM (FIN) + Wavenet (TEXT)	FIN+TEXT	0.998(0.000)	0.672(0.027)
TimeInputAlign	LSTM (FIN) + Phased LSTM (TEXT)	FIN+TEXT	0.996(0.003)	0.607(0.133)
SqueezeInputAlign	LSTM (FIN) + Wavenet (TEXT)	FIN+TEXT	0.982(0.009)	0.761 (0.013)

idea that time series could be decomposed as a plenty of individual trends, denoted by a sum of smooth functions [12].

- (2) *LSTM* on *FIN* data only. The LSTM here is an ordinary LSTM with variable-length inputs. The inputs to this model is the 30-dimensional financial indicators sequence.
- (3) *LSTM* on *TEXT* data only. The LSTM here is also an ordinary LSTM with variable-length inputs. The inputs to this model is the 300-dimensional embedding representation of text sequence.
- (4) *Phased LSTM* on *TEXT* only. The inputs to this model are the 300-dimensional embedding representation of text sequence.
- (5) *Wavenet* on *TEXT* only. The inputs to this model are the 300-dimensional embedding representation of text sequence.

4.1.5 Metrics

We adopt area under receiver operating characteristic curve (AUC) as our evaluation criterion. Commonly used in selecting the optimal classifier that predicts the classes best, AUC weights errors on the two classes separately and tells a more truthful story when working with the imbalanced dataset. The random predictor will produce the AUC value with 0.5, the more powerful the classifier is, the larger AUC value will be. We also report the prediction accuracy of different methods on the test set.

4.2 Comparison Results

To verify the efficiency of our DMDP framework, we compare the results of DMDP with several baseline methods. The average accuracy and AUC are summarized in Table 3.

From the table, we find that DMDP with LSTM on financial variables and Wavenet on news time series achieves the highest AUC of 0.761, higher than all the baselines. Note that all the neural network-based methods outperform the

Table 5 The results of different learning rates in DMDP with LSTM (FIN) + Wavenet (TEXT)

Learning rate	Accuracy	AUC
0.1	0.850(0.104)	0.498(0.016)
0.01	0.720(0.059)	0.579(0.054)
0.001	0.982(0.009)	0.761 (0.013)

Here the number of hidden units in the LSTM on FIN is 16, and the resampling ratio is 0.4

Table 6 The results of different numbers of hidden units of LSTM (FIN) in DMDP with LSTM (FIN) + Wavenet (TEXT)

#Hidden units	Accuracy	AUC
16	0.982(0.009)	0.761 (0.013)
32	0.986(0.016)	0.715(0.018)
64	0.987(0.014)	0.742(0.040)
128	0.988(0.000)	0.760(0.014)

Here the learning rate is set to be 0.001, and the resampling ratio is 0.4

Table 7 The results of different resampling ratios in DMDP with LSTM (FIN) + Wavenet (TEXT)

Resampling ratio	Accuracy	AUC
0.1	0.997(0.001)	0.749(0.019)
0.2	0.987(0.005)	0.729(0.022)
0.3	0.994(0.005)	0.757(0.044)
0.4	0.982(0.009)	0.761 (0.013)
0.5	0.991(0.003)	0.723(0.019)
0.6	0.968(0.013)	0.715(0.033)
0.7	0.993(0.001)	0.696(0.012)
0.8	0.974(0.020)	0.669(0.016)
0.9	0.975(0.021)	0.698(0.045)

Here the learning rate is set to be 0.001, and the number of hidden units is 16

GAM baseline. Also note that there are huge differences when applying different models to extract information from news text data. The results show that Wavenet outperforms LSTM and Phased LSTM by a large margin in extracting news text information. Besides, when combining financial variables with text information, the models usually improve compared to only financial variables or news text alone. For Wavenet, however, the improvement of introducing financial variables is not large, which indicates the importance of news text information.

Table 8 The results of different numbers of dilation layers in Wavenet (TEXT) in DMDP with LSTM (FIN) + Wavenet (TEXT)

#Dilation layers	Accuracy	AUC
2	0.792(0.155)	0.566(0.090)
4	0.737(0.113)	0.598(0.041)
8	0.982(0.009)	0.761 (0.013)
16	0.973(0.017)	0.703(0.004)

Here the learning rate is set to be 0.001, and the resampling ratio is 0.4

4.3 Parameter Tuning

Alignment method We run experiments with three different alignment methods on the real-word dataset. The results can be seen in Table 4. The results show that **SqueezeInputAlign** yields the best performance, while **TimeInputAlign** may suffer due to the selection of models because it can only be applied to Phased LSTM.

Learning rate We test different values of learning rate from {0.1, 0.01, 0.001} in our models. The results are presented in Table 5. We can see that our model configured with learning rate as 0.001 outperforms the other settings.

Hidden units We also evaluate the effects of different numbers of hidden units on model performance. The results are presented in Table 6. We can see that the models with 16 hidden units give the best performance. The results indicate that model with more hidden units tends to overfit and yields worse results.

Resampling ratio The data, which have no more than 1% of records as *ST samples, are highly imbalanced. To alleviate this problem, we randomly over-sample the positive cases during the training process. We manually set the percentage of positive samples in the mini-batch as a parameter and tune this parameter to achieve the best performance. We call this parameter “the resampling ratio.” In our experiments, we tested the resampling ratio ranging from 0.1 to 0.9. The results could be seen in Table 7. The results show that we achieve the best result when the resampling ratio is equal to 0.4.

Dilation layers in wavenet We also evaluate the effects of different numbers of dilation layers in Wavenet on model performance. The results are presented in Table 8. We can see that the models with 8 dilation layers give the best performance. Following the practice in [28], we also tune the parameters including the skip channels, residual channels and the number of filters. The setting which achieves the best performance on validation set is obtained via grid search.

Since the tuning is similar with dilation layers, we only list results of different dilation layers here.

5 Related Work

Recently, RNN variants such as LSTM [14] have been very successful in modeling the long-range sequential dependencies. And they have been applied to many time series forecasting or classification tasks. In [9], the authors used LSTM to predict whether a stock (6 typical stocks) would increase 0–1% (class 1), above 1% (class 2) or not increasing (class 3) within next three hours with the highest accuracy at 59.5%. In [36], the authors proposed a novel SFM model, which incorporates discrete Fourier transform (DFT) into LSTM, to predict values in the future series. They argued that by decomposing the hidden states of LSTM into multi-frequency components, they could capture different latent patterns behind the original time series. In [21], the authors tried to combine LSTM and CNN into a single framework called TreNet as the author argued that CNNs extract salient features from local raw data, while LSTM captures long-term dependency. The results demonstrated that the combined network outperforms both CNN and LSTM as well as various kernel-based models in predicting the trend in time series. However, to the best of our knowledge, no prior work has studied the problem of default probability prediction which is a critical task to perform risk assessment for listed companies. Moreover, none of the existing time series prediction methods leverages the informative social media data to enhance prediction accuracy.

6 Conclusion

This paper has developed a default probability prediction framework DMDP, which leverages both structured financial factors and unstructured news from social media, to capture default risk states of the observed corporations. DMDP involves a data alignment component to absorb multi-source data with different timestamps. We further adopt LSTM, Phased LSTM and Wavenet for financial variables time series and news text time series, respectively, to effectively extract the latent information. In the experiments, we considered over 30 financial indicators including the profitability, solvency, operation ability, cash flow ability and potential growth ability of records for over 3000 listed corporations in mainland China. The results show that compared to the existing risk assessment approach that only considers financial factors, our neural method with additional indicators from social media news improves the accuracy of the default probability prediction results. As future work, we will investigate the following research directions: (1) the effects of public opinions among

affiliated companies on a company's default value; (2) the importance of different features on default probability prediction performance.

Acknowledgements This work is supported by the National Key Research and Development Program of China (No. 2018YFC0831604). Yanyan Shen is also supported by NSFC (No. 61602297).

Author Contributions YS proposed the overall framework for default probability prediction and proposed the initial model structure. YZ finalized the two neural network-based methods and conducted all the experiments. YH reviewed the related works and participated in collecting and cleaning multi-source data.

Compliance with Ethical Standards

Ethics approval and consent to participate Yes.

Consent for publication Yes.

Availability of data and material Yes.

Conflict of interest The author declares that they have no conflict of interest.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Altman EI (1968) Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *J Finance* 23(4):589–609
2. Altman EI, Saunders A (1997) Credit risk measurement: Developments over the last 20 years. *J Bank Finance* 21(11–12):1721–1742
3. Baesens B, Van Gestel T, Viaene S, Stepanova M, Suykens J, Vanthienen J (2003) Benchmarking state-of-the-art classification algorithms for credit scoring. *J Oper Res Soc* 54(6):627–635
4. Bijak K, Thomas LC (2012) Does segmentation always improve model performance in credit scoring? *Expert Syst Appl* 39(3):2433–2442
5. Brown I, Mues C (2012) An experimental comparison of classification algorithms for imbalanced credit scoring data sets. *Expert Syst Appl* 39(3):3446–3453
6. Chen W, Ma C, Ma L (2009) Mining the customer credit using hybrid support vector machine technique. *Expert Syst Appl* 36(4):7611–7616
7. Desai VS, Crook JN, Overstreet GA (1996) A comparison of neural networks and linear scoring models in the credit union environment. *Eur J Oper Res* 95(1):24–37
8. Doumpos M, Zopounidis C (2007) Model combination for credit risk assessment: a stacked generalization approach. *Ann Oper Res* 151(1):289–306
9. Gao Q (2016) Stock market forecasting using recurrent neural network. Ph.D. thesis, University of Missouri–Columbia
10. Hand DJ, Henley WE (1997) Statistical classification methods in consumer credit scoring: a review. *J R Stat Soc Ser A Stat Soc* 160(3):523–541

11. Harris T (2015) Credit scoring using the clustered support vector machine. *Expert Syst Appl* 42(2):741–750
12. Hastie T, Tibshirani R (1990) Generalized additive models. Wiley Online Library, Hoboken
13. Hens AB, Tiwari MK (2012) Computational time reduction for credit scoring: an integrated approach based on support vector machine and stratified sampling method. *Expert Syst Appl* 39(8):6774–6781
14. Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural Comput* 9(8):1735–1780
15. Huang CL, Chen MC, Wang CJ (2007) Credit scoring with a data mining approach based on support vector machines. *Expert Syst Appl* 33(4):847–856
16. Huang Z, Chen H, Hsu CJ, Chen WH, Wu S (2004) Credit rating analysis with support vector machines and neural networks: a market comparative study. *Decis Support Syst* 37(4):543–558
17. Joulin A, Grave E, Bojanowski P, Douze M, Jégou H, Mikolov T (2016) Fasttext.zip: compressing text classification models. *arXiv preprint [arXiv:1612.03651](https://arxiv.org/abs/1612.03651)*
18. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. *arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980)*
19. Lee TS, Chen IF (2005) A two-stage hybrid credit scoring model using artificial neural networks and multivariate adaptive regression splines. *Expert Syst Appl* 28(4):743–752
20. Lee TS, Chiu CC, Chou YC, Lu CJ (2006) Mining the customer credit using classification and regression tree and multivariate adaptive regression splines. *Comput Stat Data Anal* 50(4):1113–1130
21. Lin T, Guo T, Aberer K (2017) Hybrid neural networks for learning the trend in time series. In: *Proceedings of the twenty-sixth international joint conference on artificial intelligence, IJCAI-17*, pp 2273–2279
22. Neil D, Pfeiffer M, Liu SC (2016) Phased lstm: accelerating recurrent network training for long or event-based sequences. In: *Advances in neural information processing systems*, pp 3882–3890
23. Schebesch KB, Stecking R (2005) Support vector machines for classifying and describing credit applicants: detecting typical and critical regions. *J Oper Res Soc* 56(9):1082–1088
24. Sousa MR, Gama J, Brandão E (2016) A new dynamic modeling framework for credit risk assessment. *Expert Syst Appl* 45:341–351
25. Tsai CF, Chen ML (2010) Credit rating by hybrid machine learning techniques. *Appl Soft Comput* 10(2):374–380
26. Tsai CF, Wu JW (2008) Using neural network ensembles for bankruptcy prediction and credit scoring. *Expert Syst Appl* 34(4):2639–2649
27. Twala B (2010) Multiple classifier application to credit risk assessment. *Expert Syst Appl* 37(4):3326–3336
28. Van Den Oord A, Dieleman S, Zen H, Simonyan K, Vinyals O, Graves A, Kalchbrenner N, Senior AW, Kavukcuoglu K (2016) Wavenet: a generative model for raw audio. In: *SSW*, p 125
29. Verikas A, Gelzinis A, Bacauskiene M (2011) Mining data with random forests: A survey and results of new tests. *Pattern Recognition* 44(2):330–349
30. Wang G, Hao J, Ma J, Jiang H (2011) A comparative assessment of ensemble learning for credit scoring. *Expert Syst Appl* 38(1):223–230
31. West D (2000) Neural network credit scoring models. *Comput Oper Res* 27(11):1131–1152
32. Yap BW, Ong SH, Husain NHM (2011) Using data mining to improve assessment of credit worthiness via credit scoring models. *Expert Syst Appl* 38(10):13274–13283
33. Yeh IC, Lien Ch (2009) The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Syst Appl* 36(2):2473–2480
34. Yu L, Wang S, Lai KK (2008) Credit risk assessment with a multistage neural network ensemble learning approach. *Expert Syst Appl* 34(2):1434–1444
35. Zaremba W, Sutskever I, Vinyals O (2014) Recurrent neural network regularization. *arXiv preprint [arXiv:1409.2329](https://arxiv.org/abs/1409.2329)*
36. Zhang L, Aggarwal C, Qi GJ (2017) Stock price prediction via discovering multi-frequency trading patterns. In: *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pp 2141–2149. ACM
37. Zhao Y, Huang Y, Shen Y (2018) *dmdp²*: A dynamic multi-source based default probability prediction framework. In: *Asia-Pacific Web (APWeb) and Web-age information management (WAIM) Joint international conference on web and big data*, pp 312–326. Springer
38. Zhao Z, Xu S, Kang BH, Kabir MMJ, Liu Y, Wasinger R (2015) Investigation and improvement of multi-layer perceptron neural networks for credit scoring. *Expert Syst Appl* 42(7):3508–3516