Forecasting Corporate Financial Time Series using Multi-phase Attention Recurrent Neural Networks

Shuhei Yoshimi
Kobe University
Kobe, Hyogo, Japan
shuhei@cs25.scitec.kobe-u.ac.jp

Koji Eguchi
Hiroshima University
Higashi-Hiroshima, Hiroshima, Japan
eguchi@acm.org

ABSTRACT

These days, attention-based Recurrent Neural Networks (RNNs) have been widely used for learning the hidden temporal structure of raw time series data. More recently, attention-based RNNs have been further enhanced to represent multivariate temporal or *spatio-temporal* structure underlying multivariate time series. This latest study achieved more effective prediction by employing attention structure that simultaneously capture the *spatial* relationships among multiple different time series and the temporal structure of those time series. That method assumes single time-series samples of multi- or uni-variate explanatory variables, and thus, no prediction method was designed for multiple time-series samples of multivariate explanatory variables. Moreover, such previous studies have not explored on financial time series incorporating macroeconomic time series, such as Gross Domestic Product (GDP) and stock market indexes, to our knowledge. Also, no neural network structure has been designed for focusing a specific industry. We aim in this paper to achieve effective forecasting of corporate financial time series from multiple time-series samples of multivariate explanatory variables. We propose a new industry specific model that appropriately captures corporate financial time series, incorporating the industry trends and macroeconomic time series as side information. We demonstrate the performance of our model through experiments with Japanese corporate financial time series in the task of predicting the return on assets (ROA) for each company.

1 INTRODUCTION

In recent years, a huge amount of information is accumulated day by day with the developing information technology. One such information is corporate financial time series data in the economic and financial fields. Many economic experts have interest in gaining new insights from these data. Corporate financial time series are particularly complex since they are often affected by various information, such as business conditions of each company, trends in the industry, and business sentiment in society. Traditional time series analysis and modern deep learning technology have addressed the problem of time-series prediction (or forecasting); however, there is plenty of room for new research on complex multivariate time series, such as corporate financial time series. Among the widely recognized time series analysis methods, the Autoregressive Integrated Moving Average (ARIMA) model and the kernel methods can capture one aspect of spatio-temporal patterns; however, it is not easy to achieve accurate forecasting of multivariate time series[1, 10]. Moreover, Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) can take into account time dependencies and have been well accepted lately even in the fields of financial time series analysis, such as stock price forecasting [8]. However, it may not be easy to achieve accurate long-term prediction for multivariate time series, since a part of multivariate explanatory variables may not contribute to the prediction and even do harm the prediction accuracy. It can be considered that, when some explanatory variables have relatively small contributions to the prediction, those variables may result in noises. In another line of research, a time-series prediction model was proposed so that it uses the attention-based RNN to learn the attention weights of raw time series and further enhances the ability to represent spatio-temporal features [2]. Qin et al. [12] and Yuxuan et al. [9] combined attention mechanisms with encoder-decoder models to achieve better performance in predicting one or several steps ahead. Liu et al. [11] developed Dual-Stage Two-Phase (DSTP) attention-based RNN model, by capturing correlations among multivariate explanatory variables and embedding past observations of target time series via multiple levels of attention mechanism. However, no prediction was made for multiple time-series samples with multivariate explanatory variables, in those previous studies. Moreover, no previous studies have explored on deep learning models for financial time series incorporating macroeconomic time series, such as Gross Domestic Product (GDP) and stock market indexes, to our knowledge. Also, no structure of deep learning models has been designed for focusing a specific industry, even the industry trend can be influencing. This paper aims to establish a useful method for forecasting corporate financial time series by appropriately learning from multiple time-series samples with multivariate explanatory variables. We propose a new industry specific model that appropriately captures business and industry trends, as well as macroeconomic time series, in an extension of attention-based RNN. Through experiments with Japanese corporate financial time series, we demonstrate our proposed model focusing on the wholesale industry works effectively in the task of predicting the return on assets (ROA) for each company.

2 RELATED WORK

This paper attempts to predict one step ahead of corporate financial indicators by deep learning. This section consists of four topics. The first topic is about RNNs, which have been one of the most popular deep learning methods for predicting time series data. The second topic is about LSTMs, which have been extended from RNNs to capture long and short time dependencies. The third topic is on attention mechanisms, which have attracted much attention recently due to the promising prediction performance. Those topics provide basic techniques for time series analysis based on deep learning. As the last topic, we briefly review state-of-the-art related work on deep learning-based financial time series prediction.

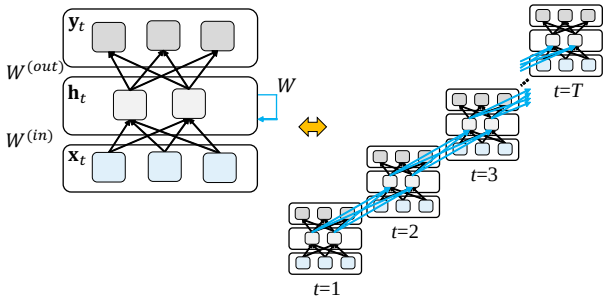

Figure 1: Structure of RNN.

2.1 Recurrent Neural Networks

Sequential data refers to any kind of data where the order of the samples is important. Especially, sequential data is also called as time series data in case that the order is based on time. It is known that prediction performance can be improved by considering the dependencies between the current samples and the past samples. One most popular method is RNN, which is an extension of feed-forward neural networks for handling sequential data [3]. Now, we suppose that the RNN receives input sequence $\mathbf{x}_t$ for each time t and also returns one output sequence $\mathbf{y}_t$ at the time t. At time $t = n$, we can assume that output $\mathbf{y}_n$ is successfully produced from input sequence $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \cdots, \mathbf{x}_n$. This is because the RNN is based on a neural network with directed closed loop called 'return path'. This structure makes it possible to store temporal information and change behaviors. Figure 1 shows the structure of the RNN and its structure expanded in time dimension.

Now, we describe the calculation process in RNNs. First, suppose $\mathbf{x}_t$ is input to the network; $\mathbf{h}_t$ is output from the middle layer; $\mathbf{y}_t$ is output from the output layer; $W^{(in)}$ is input weight matrix that represents connections from the input layer to the middle layer; and $W^{(out)}$ is output weight matrix that represents connections from the middle layer to the output layer.

In the return path, the RNN returns the output of the middle layer to its own input. The RNN assumes the connection between the middle layer at time $t-1$ to that at time t. Therefore, a weight w is assigned to each recurrent connection from an arbitrary unit of the middle layer at time $t-1$ to an arbitrary unit of the middle layer at time t. We use the notation W as a recurrent weight matrix consisting of each recurrent connection's w. Figure 2 shows one unit of middle layer in RNN.

Hidden variables in the middle layer at time t, $\mathbf{h}_t$, can be obtained by $\mathbf{x}_t$, $W^{(in)}$, $\mathbf{h}_{t-1}$, W, activation function f and bias b, as follows:

$$\mathbf{h}_t = f(W^{(in)}\mathbf{x}_t + W\mathbf{h}_{t-1} + b) \tag{1}$$

Output y_t are then obtained by f, $W^{(in)}$, and $\mathbf{h}_t$ from each unit in the middle layer:

$$\mathbf{y}_t = f(W^{(out)}\mathbf{h}_t) \tag{2}$$

In this paper, we assume that activation function f in Eq. (1) is the hyperbolic tangent function ($tanh$) and f in Eq. (2), also called as loss function, is the squared error.

2.2 Long Short-Term Memory

RNNs can capture the context of sequential data. In this case, it is important to understand the length of past sequence that should be captured in the model, in other words, how long past inputs from the current time should be reflected to predict the output.

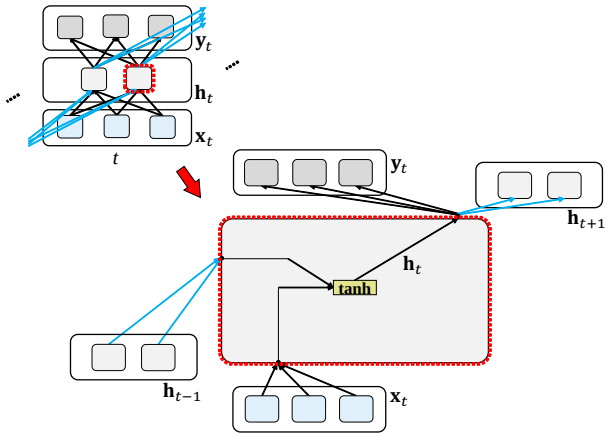

Figure 2: A unit of middle layer in RNN.

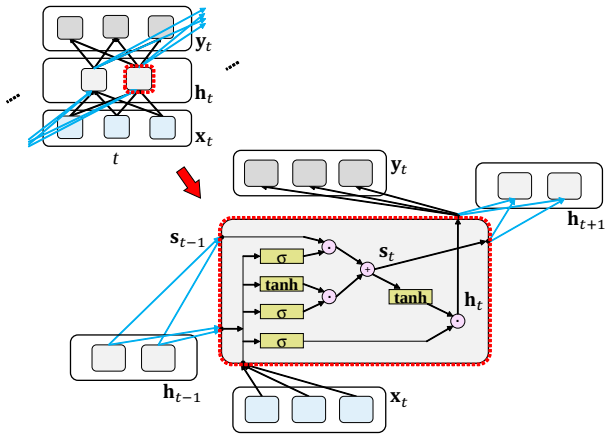

Figure 3: A unit of middle layer in LSTM.

However, the gradient usually vanishes or explodes after a certain number of iterations during learning in case of RNNs [4, 5]. This limit is caused by the so-called *gradient vanishing problem* —when calculating weights, the value of the gradient explosively decreases or increases as it is propagated backward through the RNN network. To address this problem, LSTMs [6] were proposed to achieve long- and short-term memory. Compared to the RNN, each unit in the middle layer of the LSTM has a memory cell and three gates: an input gate, an output gate, and a forget gate, while the other structure is basically the same as that of the RNN. Figure 3 shows one unit of middle layer in the LSTM.

Now, let W_i, W_o, W_f, and W_s be input weight matrices, where the subscript indicates the input gate i, output gate o, forget gate f, or memory cell s in the LSTM network. Also, let U_i, U_o, U_f, and U_s be recurrent weight matrices, and b_i, b_o, b_f, and b_s be biases. Suppose σ is the sigmoid function, and $tanh$ is the hyperbolic tangent function. At time t, we suppose that $\mathbf{i}_t$ is the output of the input gate; $\hat{\mathbf{s}}_t$ is a new candidate state of the memory cell; $\mathbf{o}_t$ is the output of the output gate; $\mathbf{f}_t$ is the output of the forget gate; $\mathbf{s}_t$ is the state of the memory cell; and $\mathbf{h}_t$ is the output of the memory cell. We obtain these variables, as follows:

$$\mathbf{i}_t = \sigma(W_i\mathbf{x}_i + U_i\mathbf{h}_{t-1} + b_i) \tag{3}$$

$$\hat{\mathbf{s}}_t = tanh(W_s\mathbf{x}_t + U_s\mathbf{h}_{t-1} + b_s) \tag{4}$$

$$\mathbf{f}_t = \sigma(W_f\mathbf{x}_t + U_f\mathbf{h}_{t-1} + b_f) \tag{5}$$

$$\mathbf{s}_t = \mathbf{i}_t \odot \hat{\mathbf{s}}_t + \mathbf{f}_t \odot \mathbf{s}_{t-1} \tag{6}$$

$$\mathbf{o}_t = \sigma(W_o \mathbf{x}_t + U_o \mathbf{h}_{t-1} + b_o) \tag{7}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot tanh(\mathbf{s}_t) \tag{8}$$

where $\odot$ indicates the Hadamard (or element-wise) product.

Now, we will discuss how the LSTM's units work in the four steps below.

- Step 1 —Update the output of the forget gate $\mathbf{f}_t$:
  First, the model needs to determine what to forget from the cell state, as shown in Eq. (5). $\mathbf{f}_t$ is obtained by the output of the previous step $\mathbf{h}_{t-1}$ and the input $\mathbf{x}_t$. σ is used to give output values between 0 and 1. For instance, when the value is 1, the current cell state is stored completely. When the value is 0, it is forgotten completely.
- Step 2 —Update the output of input gate $\mathbf{i}_t$ and new candidate state of memory cell $\hat{\mathbf{s}}_t$:
  Second, the model needs to determine what information is going to be added to the cell state, as shown in Eqs. (3) and (4). In this step again, the input is obtained by $\mathbf{h}_{t-1}$ and $\mathbf{x}_t$. Input gate $\mathbf{i}_t$ first applies σ to determine which previous cell state will be updated. *tanh* is then used to obtain a new candidate value $\hat{\mathbf{s}}_t$. In the next step, these two will be combined to update the cell state $\mathbf{s}_{t-1}$.
- Step 3 —Update the state of the memory cell $\mathbf{s}_t$:
  Third, the model updates the cell state, as shown in Eq. (6). Now, the old cell state $\mathbf{s}_{t-1}$ is multiplied by $\mathbf{f}_t$ to forget unnecessary information. Then, the product of $\mathbf{i}_t$ by $\mathbf{s}_{t-1}$ is added to the cell state memory.
- Step 4 —Update the output of the memory cell $\mathbf{h}_t$ and the output of output gate $\mathbf{o}_t$:
  Finally, the model determines the output unit, as shown in Eqs. (7) and (8). $\mathbf{h}_t$ is based on the cell state but in a filtered version. First, σ is applied to the previous memory output $\mathbf{h}_{t-1}$ and input $\mathbf{x}_t$, to determine output gate $\mathbf{o}_t$. This value indicates which cell state is going to be output in the range from 0 to 1. Then, cell state $\mathbf{s}_t$ is transformed by *tanh* in the range from -1 to 1. Next, this transformed cell state is multiplied by output gate $\mathbf{o}_t$, resulting in $\mathbf{h}_t$. This output will be forwarded to the next step in the network.

These structures improve the limitation of RNNs in which limited-term memory can only be captured, achieving a more accurate estimation that captures a longer-term memory.

2.3 Attention mechanisms

Attention mechanisms were successfully used for LSTM [11, 12]. In time series analysis with LSTM, the attention mechanisms can simultaneously capture the *spatial* relationships among multiple different time series and the temporal structure of those time series. In the rest of this subsection, we will briefly review the attention mechanism developed by Liu et al. [11].

2.3.1 Spatial-attention LSTM. The purpose of the spatial attention mechanism is to obtain the spatial correlations among multivariate input time series. Given the time series (of length T) of k-th explanatory attribute at time t, $\mathbf{x}_t^k = (x_{t-T+1}^k, \cdots, x_t^k) \in \mathbb{R}^T$, the following attention mechanism can be used:

$$a_t^k = \mathbf{v}_a^{\mathrm{T}} \tanh(W_a[\mathbf{h}_{t-1}; \mathbf{s}_{t-1}] + U_a \mathbf{x}_t^k + \mathbf{b}_a) \tag{9}$$

$$\alpha_t^k = \frac{\exp(a_t^k)}{\sum_{j=1}^n \exp(a_t^j)} \tag{10}$$

where $[*;*]$ is a concatenation operation, and $\mathbf{v}_a, \mathbf{b}_a \in \mathbb{R}^T$, $W_a \in \mathbb{R}^{T \times 2m}$, $U_a \in \mathbb{R}^{T \times T}$ are parameters to learn, $\mathbf{h}_{t-1} \in \mathbb{R}^m$ and $\mathbf{s}_{t-1} \in \mathbb{R}^m$ are respectively the hidden state and cell state vectors of time $t-1$, and m is the number of hidden states of this attention module. Spatial attention weight at time t, $(\alpha_t^1, \cdots, \alpha_t^n)$, is determined by the hidden states and cell states at time $t-1$, and inputs of explanatory attributes at time t. It represents the effect of each explanatory attribute on the forecast of the target. Using the attention weight associated with each explanatory attribute, the multivariate input at time t, $\mathbf{x}_t = (x_t^1, \cdots, x_t^n)$, is weighted as follows:

$$\tilde{\mathbf{x}}_t = (\alpha_t^1 x_t^1, \alpha_t^2 x_t^2, \cdots, \alpha_t^n x_t^n)^{\mathrm{T}} \tag{11}$$

Let $f_{spatial}$ be the LSTM with the spatial attention mechanism mentioned previously. Then we can get the following equation.

$$(\mathbf{h}_t, \mathbf{s}_t) = f_{spatial}(\mathbf{h}_{t-1}, \mathbf{s}_{t-1}, \tilde{\mathbf{x}}_t) \tag{12}$$

2.3.2 Temporal-attention LSTM. The purpose of the temporal attention mechanism is to maintain the temporal relationships of the spatial attention. The spatio-temporal relationships in a fixed-size window is extracted using the spatial relationship among multivariate time series in the time window of length T, which was mentioned previously. It is not sufficient to understand the temporal relationships in a fixed-size window, so an attention mechanism for selecting hidden states is promising. The hidden state most relevant to the target (or objective) variable is selected. For each i-th hidden state, the attention mechanism gives temporal attention weightes $(\beta_t^1, \cdots, \beta_t^T)$, as follows:

$$b_t^i = \mathbf{v}_b^{\mathrm{T}} \tanh(W_b[\mathbf{d}_{t-1}; \mathbf{s}'_{t-1}] + U_b \mathbf{h}_i + \mathbf{b}_b) \tag{13}$$

$$\beta_t^i = \frac{\exp(b_t^i)}{\sum_{j=1}^T \exp(b_t^j)} \tag{14}$$

where $\mathbf{h}_i$ represents the i-th hidden state vector obtained in the spatial attention module mentioned previously. $\mathbf{d}_{t-1} \in \mathbb{R}^p$ and $\mathbf{s}'_{t-1} \in \mathbb{R}^p$ are respectively the hidden state and cell state vectors of time $t-1$, and p is the number of hidden states of this attention module. $\mathbf{v}_b, \mathbf{b}_b \in \mathbb{R}^p$, $W_b \in \mathbb{R}^{p \times 2p}$, and $U_b \in \mathbb{R}^{p \times m}$ are the parameters to learn. Next, context vector $\mathbf{c}_t$ are defined as follows:

$$\mathbf{c}_t = \sum_{j=1}^T \beta_t^j \mathbf{h}_j \tag{15}$$

Context vector $\mathbf{c}_t$ represents the information of all the hidden states, representing the temporal relationships within a time window. This context vector $\mathbf{c}_t$ is then aligned with target variable y_t, as follows:

$$\tilde{y}_t = \tilde{\mathbf{w}}^{\mathrm{T}}[y_t; \mathbf{c}_t] + \tilde{b} \tag{16}$$

where $\tilde{\mathbf{w}} \in \mathbb{R}^{m+1}$ and $\tilde{b} \in \mathbb{R}$ are parameters that map the concatenation to the target variable. Aligning the target time series with the context vector makes it easier to maintain temporal relationships and makes use of the results to update the hidden state and cell state. Let $f_{temporal}$ be the LSTM with the temporal attention mechanism mentioned previously. Then we can get the following equation.

$$(\mathbf{d}_t, \mathbf{s}'_t) = f_{temporal}(\mathbf{d}_{t-1}, \mathbf{s}'_{t-1}, \tilde{y}_{t-1}) \tag{17}$$

Given T-length multivariate explanatory time series $X_T = (\mathbf{x}_1, \cdots, \mathbf{x}_t, \cdots, \mathbf{x}_T)$, where $\mathbf{x}_t = (x_t^1, \cdots, x_t^n)$, and target time series $\mathbf{y}_T = (y_1, \cdots, y_t, \cdots, y_T)$, context vector $\mathbf{c}_T$ and hidden

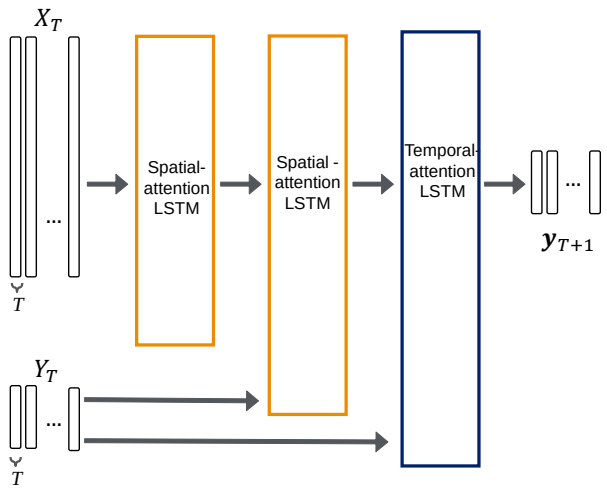

Figure 4: Structure of the DSTP-based model.

state vector $\mathbf{h}_T$ are concatenated to make the final prediction of one step ahead of target variable y_{T+1}, as follows:

$$\hat{y}_{T+1} = F(X_T, \mathbf{y}_T) \quad (18)$$

$$= \mathbf{v}_y^{\mathrm{T}}(W_y[\mathbf{d}_T; \mathbf{c}_T] + \mathbf{b}_y) + b'_y \quad (19)$$

where F denotes the predictor. $W_y \in \mathbb{R}^{p\times(p+m)}$ and $\mathbf{b}_y \in \mathbb{R}^p$ map the concatenation $[\mathbf{h}_T; \mathbf{c}_T] \in \mathbb{R}^{p+m}$ to p-dimensional latent space. A linear function with weights $\mathbf{v}_y \in \mathbb{R}^p$ and bias $b'_y \in \mathbb{R}$ produces final prediction $\hat{y}_{T+1}$.

2.4 State-of-the-art financial time series prediction with LSTM

Quite recently, Liu et al. [11] focused on spatial correlations and incorporated target time series to develop Dual-Stage Two-Phase (DSTP) attention-based LSTM model. Here, we briefly introduce the structure of their model in Figure 4. Let X_t be multivariate explanatory time series and Y_t be target time series. At the second phase of Spatial-attention LSTM, x^k in Eq. (9) is replaced with $[\tilde{x}^k; y^k]$, where the (observed) target time series is concatenated with the corresponding explanatory time series.

They enhanced the attention mechanisms to incorporate both spacial correlations and temporal relationships. However, we have three concerns when we employ the DSTP-based model to achieve our goal. First, the dataset they used was NASDAQ 100 stock data[1], which involves multiple time series samples with a univariate explanatory variable, not with multivariate explanatory variables. Second, their model did not incorporate macroeconomic time series. Third, their model did not consider industry-wide trends.

3 DATA

In this study, we use "Surveys for the Financial Statements Statistics of Corporations by Industry"[2] collected by the Ministry of Finance Japan. The surveys are based on sampling in which target commercial corporations are general partnership companies, limited partnership companies, limited liability companies, and stock companies, all of whose head offices are located in Japan. We excluded the companies in 'finance and insurance' industry for our study. The collection period is from the first quarter of 2003 to the fourth quarter of 2016. The surveys consist of annual survey and quarterly survey. The total number of companies are 57, 775 in the quarterly survey and 60, 516 in the annual survey. These surveyed items include the financial indexes shown in Table 1.

We use financial statements in the quarterly survey, and calculate various financial ratios as explanatory and target variables in our analysis. We need to perform preprocessing for the use of time-series analysis. First, the survey dataset contains both long-term and short-term companies; however, it is not easy to include short-lived companies for time-series analysis and thus we excluded short-term companies for our analysis. Secondly, the survey dataset contains a number of missing values, so we need to take care of them before calculating financial ratios. In summary, we employed the following three steps, in this order, for preprocessing of the data: (1) extraction of survey items for long-term companies (*exclusion processing*), (2) imputation of missing values in survey items (*imputation processing*), (3) calculation of financial ratios using the survey items (*calculation processing*). We will describe the details in the following subsections.

3.1 Exclusion processing

Before imputation processing, we do firstly exclusion processing. This is because some of company data greatly diverge from true data even if imputation processing is performed at this step. We performed exclusion processing in the following cases:

- Case 1: companies that do not have all of 56 time steps.
- Case 2: companies of which financial statements do have no data in the entire time steps.

After the exclusion, the number of companies in the dataset for the experiments became 2296.

3.2 Imputation processing

Before calculation processing, we do secondly imputation processing. This is because some of company data have missing values, and if so, we cannot calculate financial ratios. We show the details on the imputation processing in the appendix.

3.3 Calculation processing

In this study, a number of financial ratios are used as explanatory variables. Each financial ratio is based on formula of the financial sales ratio in corporate enterprise statistics, as defined in Table 2.

3.4 External data

In addition to financial ratios, we use two macroeconomic time series as external data: one is Nikkei Average closing price[3] (N) and the other is Japanese GDP[4] (G). The Nikkei Average closing price (N) is extracted from January in 2003 to December in 2016 on monthly basis. The Japanese GDP (G) is extracted from the 1st quarter of 2003 to the 4th quarter of 2016 on quarterly basis.

4 PROPOSED MODEL

In this paper, we propose a model for predicting one step ahead of the corporate financial time series in a target industry. Figure 5 shows the model's flow from input time series to output time series. The 1st phase aims to extract the spatial correlations

[1] https://cseweb.ucsd.edu/ yaq007/NASDAQ100_stock_data.html
[2] https://www.mof.go.jp/english/pri/reference/ssc/outline.htm
[3] https://indexes.nikkei.co.jp/nkave//index/profile?idx=nk225
[4] https://www.esri.cao.go.jp/jp/sna/data/data_list/sokuhou/files/2019/toukei_2019.html

Table 1: Financial statements. '†' and '‡' indicate that the designated item is supposed to be recorded as of the beginning of every term (fiscal year or quarter) and as of the end of every term, respectively.

Classifications	Quarterly surveys	Annual surveys
Liabilities	Notes, accounts payable, and trade†,‡	Notes†,‡
		Accounts payable and trade†,‡
Fixed assets	Land †,‡	Land †,‡
	Construction in progress †,‡	Construction in progress †,‡
	Other tangible assets †,‡	Others †,‡
	Intangible assets †,‡	Excluded software †,‡
		Software †,‡
	Total liabilities and net assets †,‡	Total assets †,‡
Personnel	Number of employees ‡	Number of employees ‡
Profit and loss	Depreciation and amortization ‡	Depreciation and amortization ‡
		Extraordinary depreciation and amortization ‡
	Sales ‡	Sales ‡
	Cost of sales ‡	Cost of sales ‡
	Operating profit ‡	Operating profit ‡
	Ordinary profit ‡	Ordinary profit ‡

Table 2: Financial ratios. '$*$' indicates that the designated item is obtained by averaging that as of the beginning of each quarter and that as of the end of the quarter. '$**$' indicates that the designated item is obtained as the amount of increase (or decrease) from that as of the beginning of each quarter to that as of the end of the quarter.

X_0	Operating return on assets	$\frac{\text{(Operating profit)}}{\text{(Total liabilities and net assets}^*\text{)}}$
X_1	Ordinary return on assets	$\frac{\text{(Ordinary profit)}}{\text{(Total liabilities and net assets}^*\text{)}}$
X_2	Operating profit ratio	$\frac{\text{(Operating profit)}}{\text{Sales}}$
X_3	Ordinary profit ratio	$\frac{\text{(Ordinary profit)}}{\text{Sales}}$
X_4	Total asset turnover ratio	$\frac{\text{Sales}}{\text{(Total liabilities and net assets}^*\text{)}}$
X_5	Tangible fixed assets turnover ratio	$\frac{\text{Sales}}{\text{(Notes, accounts payable, and trade}^*\text{)}}$
X_6	Accounts payable turnover ratio	$\frac{\text{Sales}}{\text{Land}^*\text{+(Other tangible assets}^*\text{)}}$
X_7	Depreciation and amortization ratio	$\frac{\text{(Depreciation and amortization)}}{\text{(Other tangible assets)+(Intangible assets)+(Depreciation and amortization)}}$
X_8	Capital equipment	$\frac{\text{Land}^*\text{+(Other tangible assets}^*\text{)}}{\text{(Number of employees)}}$
X_9	Cash flow ratio	$\frac{\text{(Ordinary profit)+(Depreciation and amortization)}}{\text{(Total liabilities and net assets}^*\text{)}}$
X_{10}	Capital Investment ratio	$\frac{\text{(Construction in progress}^{**}\text{)+(Other tangible assets}^{**}\text{)+(Intangible assets}^{**}\text{)+(Depreciation and amortization)}}{\text{(Total liabilities and net assets}^*\text{)}}$
X_{11}	Gross profit ratio	$\frac{\text{Sales}-\text{(Cost of sales)}}{\text{Sales}}$

among multivariate time series. The 2nd and 3rd phases aim to make the prediction of a target variable at time $T+1$, given the past explanatory and target time series of window size T. More specifically, the 2nd phase aims to extract the spatial correlations between the target time series and the multivariate explanatory time series, while the 3rd phase aims to extract temporal relationships.

1st phase. This phase captures spatial correlations in all of the multivariate time series (including both explanatory and target variables observed as time series) of the entire length. By applying the attention mechanisms to the entire time series before splitting the time-series data, the correlations between time-series samples can be captured well. For multiple time-series samples with multivariate variables, such that each time-series sample corresponds to each company, it is not easy to well capture the spatial correlations after the data splitting, which does not distinguish which time-series slice belongs to which company.

First, Spatial Attention-based LSTM, denoted as $F_{Spatial}$, is applied to all company time series (A) to obtain the spatial correlations among the multivariate time series for every company, as in Eqs. (9), (10), and (11).

$$\acute{A} = F_{spatial}(A) \tag{20}$$

We then use a linear function to aggregate all samples. This liner function summarizes all companies' features to an aggregated feature space. Here, let W be weights and b be biases in the liner function.

$$\tilde{A} = W\acute{A} + b \tag{21}$$

Eqs. (20) and (21) allow the model to incorporate all financial statements. Here, $\acute{A}$ with shape (J, L, K) —$J \times L \times K$ third-order tensor— is aggregated to $\tilde{A}$ with shape $(1, L, K)$, where J, L, and K indicate the number of time-series samples (or companies), the length of training data, and the number of dimensions of multivariate variables (including both explanatory and target variables), respectively. Since the closing price of Nikkei Average (N) is on monthly basis and its length is $\acute{L} = L \times 3$, the attention mechanism is applied to aggregate it on quarterly basis. Here, N with shape $(1, \acute{L}, 1)$ is aggregated to $\tilde{N}$ with shape $(1, L, 1)$. More specifically, the following formula is applied when $i \in \{1, 2, 3\}$ indicates either of the first month to the third month for each quarter t:

$$\gamma_t^i = \frac{\exp(N_t^i)}{\sum_{j=1}^{3} \exp(N_t^j)} \tag{22}$$

$$\tilde{N}_t = (\gamma_t^1 N_t^1, \gamma_t^2 N_t^2, \gamma_t^3 N_t^3)^{\mathrm{T}} \tag{23}$$

Then, the quarterly representations of Nikkei Average closing price, $\tilde{N}$, are concatenated with the aggregated company data, $\tilde{A}$, as well as Japanese GDP, G, and the multivariate explanatory

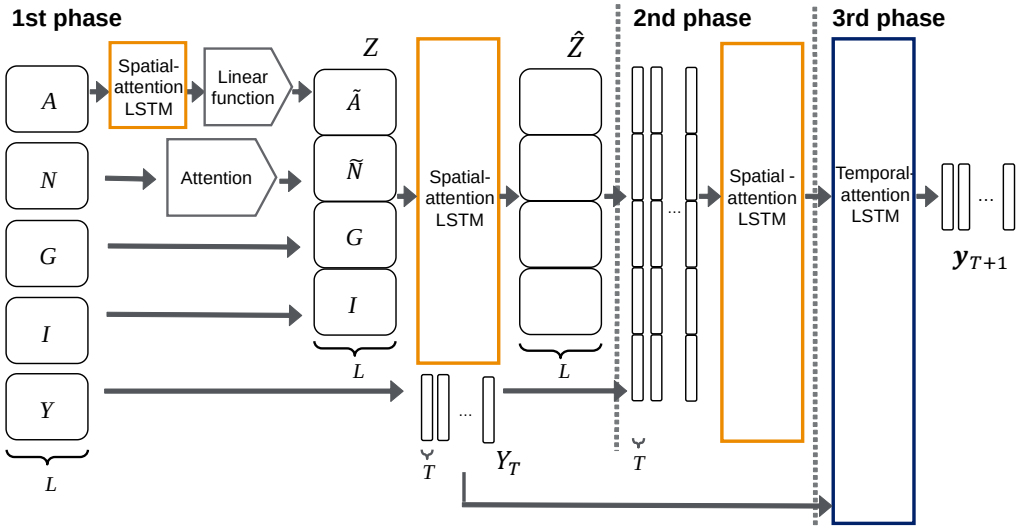

Figure 5: Structure of the proposed model.

time series in the target industry, I. Here, I indicates all the financial ratio time series other than the target time series in its industry.

$$Z = [\tilde{A}; \tilde{N}; G; I] \tag{24}$$

The shape of G is $(1, L, 1)$ and the shape of I is $(B, L, K-1)$, where B indicates the mini-batch size for I for learning the model. Therefore, the shape of the concatenated samples Z is $(B, L, K \times 2 + 1)$. By applying Spatial-attention LSTM to this, we can obtain the spatial correlations between the time series.

$$\hat{Z} = F_{spatial}(Z) \tag{25}$$

In this paper, $F_{spatial}$ in Eqs. (20) and (25) are obtained independently from A and Z, respectively. As the final stage of the 1st phase, $\hat{Z}$ is sliced by shifting, one by one, the time steps of length T. Therefore, the number of time series samples is $B \times (L - T)$, and the shape is $(B \times (L - T), T, K \times 2 + 1)$.

2nd phase. First, $\hat{Z}$ of length T and the target time series Y_T of the same length are concatenated at each time $t \in \{1, \cdots, T\}$. Here, Y_T is obatained by slicing in the same manner of Z mentioned previously.

$$\dot{Z} = [\hat{Z}; Y_T] \tag{26}$$

After the concatenation, the shape of the time series samples $\dot{Z}$ is $(B \times (L - T), T, K \times 2 + 2)$. By applying Spatial-attention LSTM to this, we can obtain the spatial correlations between the target time series and the other time series.

$$\hat{\dot{Z}} = F_{spatial}(\dot{Z}) \tag{27}$$

3rd phase. At this phase, we apply Temporal-attention LSTM (denoted as $F_{temporal}$) to capture the temporal relationships in the spatial attentions, as in Eq. (18). In other words, it captures the spatio-temporal relationships of multiple time series starting at different times.

$$\hat{\mathbf{y}}_{T+1} = F_{temporal}(\hat{\dot{Z}}, Y_T) \tag{28}$$

The generated $\hat{\mathbf{y}}_{T+1}$ is the final prediction. Here, all the models in this paper use a back-propagation algorithm to train the models. During the training process, the mean squared error (MSE) between the predicted target vector $\hat{\mathbf{y}}_{T+1}$ and the ground-truth vector $\mathbf{y}_{T+1}$ is minimized using the Adam optimizer [7].

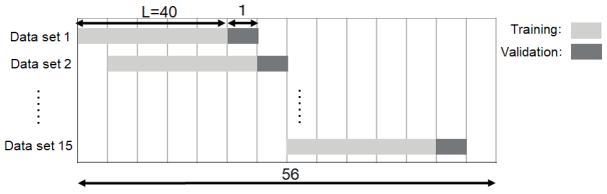

Figure 6: Data splitting for validation step.

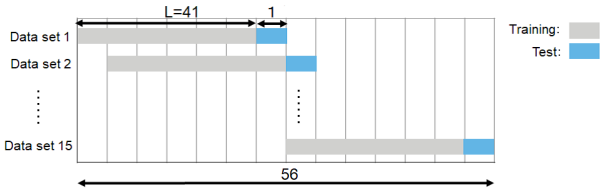

Figure 7: Data splitting for test step.

5 EXPERIMENTS

5.1 Settings

For experiments, we carry out validation step and test step, as below. In both steps, we slide data one step forward, resulting in 15 datasets.

- *Validation step*: We set the length of training data L to 40 steps, as shown in Figure 6. Training is carried out to search the best hyperparameter of the number of LSTM units. We assume the search range to be $U \in \{16, 32, 64, 128\}$.
- *Test step*: We set the length of training data L to 41 steps, as shown in Figure 7. For training, we use the best hyperparameter selected in the validation step.

Moreover, we set the other hyperparameters, as follows. Window length $T = 12$, the number of epochs is 500, learning rate is 0.001, and the mini-batch size $B = 64$. These are determined empirically.

In the setting above, we assume 'Whole sale and trade' as a target industry. We also assume Return on assets (ROA), denoted

as X_1 in Table 2, as target variable Y. As for multivariate explanatory variables, we use $X = \{X_0, X_2, X_3, \cdots, X_{11}\}$. Therefore, the number of the variables in time series is $K = 12$, including both explanatory and target variables.

In order to demonstrate the effectiveness of the proposed model, we compare the prediction performance of the proposed model to that of the DSTP-based model and simple LSTM model. The Nikkei Average closing price (N) was converted to quarterly by averaging every three months for the DSTP-based model and simple LSTM model. We assumed that the number of LSTM layers is two.

For evaluation, we use Mean Squared Error (MSE) with the sample standard deviation. We further confirm whether the improvement of the proposed model was statistically significant via Wilcoxon signed rank testing at 0.05 level, compared with the baselines.

5.2 Results and discussions

We show in Table 3 the evaluation results in terms of mean squared errors (MSE) and the sample standard deviations (SD) using the proposed model, DSTP-based model, and simple LSTM model in various settings. Here, I indicates a target industry data and $I = [X; Y] \in A$. Also, '# of units' indicates the number of the LSTM units for each model, which was determined in the validation step.

The following are discussions to clarify the contributions from the three points of view:

- *Using macroeconomic time series*: Comparing the proposed models with macroeconomic time series (Model 1) and those without macroeconomic time series (Model 3) in Table 3, Model 1 works more effectively than Model 3, on average, by successfully capturing properties of the macroeconomic time series. We confirmed that the improvement brought by Model 1 was statistically significant via Wilcoxon signed rank test at 0.05 level, compared to Model 3.
- *Focusing a specific industry*: When we compare the proposed model with all the time series (Model 1) and that without considering spatial correlations among all companies' time series (Model 2), Model 1 works moderately more effectively than Model 2, on average. We confirmed that this improvement was statistically significant in the same manner mentioned previously.
- *Using multiple time-series samples with multivariate explanatory variables*: Given all the time series, Model 1 was more effective on average (with the statistical significance) than Models 4 and 7; however, this is not the case in the other situations. It can be said that our proposed models work modestly more effective than or comparable to the DSTP-based models, depending on the situations, and that our models work greatly more effective than the simple LSTM models. More detailed evaluation is left for the future work.

6 CONCLUSIONS

In order to establish a useful method for forecasting corporate financial time series data, we aimed in this paper to appropriately forecast one step ahead using multiple time-series samples of multivariate explanatory variables. For this objective, we proposed an industry specific model that simultaneously captures corporate financial time series and the industry trends. We further proposed a new model structure that appropriately captures macroeconomic time series. In particular, we showed the effectiveness of the proposing model, especially focusing on the following three points. First, the model can capture macroeconomic time series, such as GDPs, more appropriately than the DSTP- and LSTM-based models. Second, the model can be focused to a specific industry. Third, the model is designed to be learned from multiple time-series samples of multivariate explanatory variables, producing modestly more effective or comparable performance of the prediction compared to the DSTP-based model. We developed a model for predicting corporate ROA in the wholesale trade industry. Our model may be effective in forecasting other target variables in other industries; however, this extension is left for the future work.

ACKNOWLEDGMENTS

We thank Takuji Kinkyo and Shigeyuki Hamori for valuable discussions and comments. This work was supported in part by the Grant-in-Aid for Scientific Research (#15H02703) from JSPS, Japan.

REFERENCES

[1] M. Hadi Amini, Amin Kargarian, and Orkun Karabasoglu. 2016. ARIMA-based decoupled time series forecasting of electric vehicle charging demand for stochastic power system operation. *Electric Power Systems Research* 140 (2016), 378–390.
[2] Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F. Stewart, and Jimeng Sun. 2016. RETAIN: Interpretable predictive model in healthcare using reverse time attention mechanism. *Advances in Neural Information Processing Systems (NIPS 2016)* 29 (2016), 3504–3512.
[3] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press.
[4] Sepp Hochreiter. 1991. Untersuchungen zu dynamischen neuronalen Netzen. Diploma thesis, Institut für Informatik, Technische Universiät München.
[5] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, and Jürgen Schmidhuber. 2001. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In *A Field Guide to Dynamical Recurrent Neural Networks*, S. C. Kremer and J. F. Kolen (Eds.). IEEE Press.
[6] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (1997), 1735–1780.
[7] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*.
[8] Hao Li, Yanyan Shen, and Yanmin Zhu. 2018. Stock Price Prediction Using Attention-based Multi-Input LSTM. *Proceedings of Machine Learning Research* 95 (2018), 454–469.
[9] Yuxuan Liang, Songyu Ke, Junbo Zhang, Xiuwen Yi, and Yu Zheng. 2018. GeoMAN: Multi-level attention networks for geo-sensory time series prediction. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI-18)*. 3428–3434.
[10] Jie Liu and Enrico Zio. 2017. SVM hyperparameters tuning for recursive multi-step-ahead prediction. *Neural Computing & Applications* 28, 12 (2017), 3749–3763.
[11] Yeqi Liu, Chuanyang Gong, Ling Yang, and Yingyi Chen. 2019. DSTP-RNN: A dual-stage two-phase attention-based recurrent neural network for long-term and multivariate time series prediction. *Expert Systems with Applications* 143 (2019).
[12] Yao Qin, Dongjin Song, Haifeng Cheng, Wei Cheng, Guofei Jiang, and Garrison W. Cottrell. 2017. A dual-stage attention-based recurrent neural network for time series prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI-17)*. 2627–2633.

A IMPUTATION PROCESSING

Now, we briefly describe imputation processing to handle time series with missing values. We suppose time series g in the quarterly survey dataset and time series G in the annual survey dataset with regard to each financial index. We carry out imputation processing for each financial index in the two steps below. Here, let q_j^y be a value as of quarter j of year y in a specific financial index time series, as shown in Table 4.

- Step 1 —using annual data:

Table 3: Mean squared errors (MSE) and the sample standard deviations (SD) using the proposed model, DSTP-based model, and simple LSTM model in various settings.

Model IDs	Models	Data used	# of units	MSE	SD
Model 1	Proposed model	(I), A, N, G	16	2.08×10^{-4}	8.48×10^{-5}
Model 2	Proposed model (w/o 'A')	I, N, G	32	2.11×10^{-4}	7.43×10^{-5}
Model 3	Proposed model (w/o 'N' and 'G')	(I), A	128	2.28×10^{-4}	7.64×10^{-5}
Model 4	DSTP-based model	A, N, G	16	2.12×10^{-4}	6.47×10^{-5}
Model 5	DSTP-based model	I, N, G	16	2.26×10^{-4}	9.40×10^{-5}
Model 6	DSTP-based model	A	16	2.11×10^{-4}	7.43×10^{-5}
Model 7	Simple LSTM model	A, N, G	16	3.86×10^{-4}	6.07×10^{-4}
Model 8	Simple LSTM model	A	16	3.10×10^{-4}	3.72×10^{-4}
Model 9	Simple LSTM model	I	16	2.95×10^{-4}	3.43×10^{-4}

Table 4: Notations of years and terms.

Year	Quarters	Quarterly surveys	Annual surveys
y	1	q_1^y	G_y
	2	q_2^y	
	3	q_3^y	
	4	q_4^y	

For the imputation, we make use of the annual survey dataset. When the value q_j^y is missing, we attempt to look up G_y in the annual survey dataset. If G_y is also missing, this step for q_j^y is skipped. This imputation processing consists of the following six cases, depending on financial indexes.

– Case 1: q_1^y is equal to G_y.

$$q_1^y = G_y \tag{29}$$

– Case 2: q_4^y is equal to G_y.

$$q_4^y = G_y \tag{30}$$

– Case 3: The sum of q_1^y, q_2^y, q_3^y and q_4^y is equal to G_y. The imputation in this case depends on the number of missing values in $\{q_1^y, q_2^y, q_3^y, q_4^y\}$. We assume Cases 3-1, 3-2, 3-3, and 3-4, as below, where $a, b, c, d \in \{1, 2, 3, 4\}$:

∗ Case 3-1 —when the number of missing values in $\{q_a^y, q_b^y, q_c^y, q_d^y\}$ is one:
Suppose q_a^y is missing.

$$q_a^y = G_y - q_b^y - q_c^y - q_d^y \tag{31}$$

∗ Case 3-2 —when the number of missing values in $\{q_a^y, q_b^y, q_c^y, q_d^y\}$ is two:
Suppose q_a^y and q_b^y are missing.

$$q_a^y = q_b^y = \frac{G_y - q_c^y - q_d^y}{2} \tag{32}$$

∗ Case 3-3 —when the number of missing values in $\{q_a^y, q_b^y, q_c^y, q_d^y\}$ is three:
Suppose q_a^y, q_b^y and q_c^y are missing.

$$q_a^y = q_b^y = q_c^y = \frac{G_y - q_d^y}{3} \tag{33}$$

∗ Case 3-4 —when the number of missing values in $\{q_a^y, q_b^y, q_c^y, q_d^y\}$ is four:
Suppose q_a^y, q_b^y, q_c^y and q_d^y are missing.

$$q_a^y = q_b^y = q_c^y = q_d^y = \frac{G_y}{4} \tag{34}$$

– Case 4: q_1^y is equal to the sum of G_y and the other financial index G'_y.

$$q_1^y = G_y + G'_y \tag{35}$$

– Case 5: q_4^y is equal to the sum of G_y and the other financial index G'_y.

$$q_4^y = G_y + G'_y \tag{36}$$

– Case 6: The sum of q_1^y, q_2^y, q_3^y and q_4^y is equal to the sum of G_y and the other financial index G'_y.
The imputation in this case depends on the number of missing values in $\{q_1^y, q_2^y, q_3^y, q_4^y\}$. We assume Cases 6-1, 6-2, 6-3 and 6-4, as below, where $a, b, c, d \in \{1, 2, 3, 4\}$:

∗ Case 6-1 —when the number of missing values in $\{q_a^y, q_b^y, q_c^y, q_d^y\}$ is one:
Suppose q_a^y is missing.

$$q_a^y = G_y + G'_y - q_b^y - q_c^y - q_d^y \tag{37}$$

∗ Case 6-2 —when the number of missing values in $\{q_a^y, q_b^y, q_c^y, q_d^y\}$ is two:
Suppose q_a^y and q_b^y are missing.

$$q_a^y = q_b^y = \frac{G_y + G'_y - q_c^y - q_d^y}{2} \tag{38}$$

∗ Case 6-3 —when the number of missing values in $\{q_a^y, q_b^y, q_c^y, q_d^y\}$ is three:
Suppose q_a^y, q_b^y and q_c^y are missing.

$$q_a^y = q_b^y = q_c^y = \frac{G_y + G'_y - q_d^y}{3} \tag{39}$$

∗ Case 6-4 —when the number of missing values in $\{q_a^y, q_b^y, q_c^y, q_d^y\}$ is four:
Suppose q_a^y, q_b^y, q_c^y and q_d^y are missing.

$$q_a^y = q_b^y = q_c^y = q_d^y = \frac{G_y + G'_y}{4} \tag{40}$$

- Step 2 — linear interpolation and extrapolation:
We perform commonly-used linear interpolation and extrapolation for missing values in each financial index time series in the quarterly survey dataset.