Transfer Learning for Financial Time Series Forecasting

Chapter · August 2019

DOI: 10.1007/978-3-030-29911-8_3

3 authors:

Qiqiao He
University of Macau

Patrick Cheong-Iao Pang
Macao Polytechnic Institute

Yain Whar Si
University of Macau

Transfer Learning for Financial Time Series Forecasting

Qi-Qiao He[1], Patrick Cheong-Iao Pang[2], and Yain-Whar Si[3]

[1] Department of Computer and Information Science, University of Macau, Macau
[2] School of Computing and Information Systems, The University of Melbourne, Parkville, Australia
[3] Department of Computer and Information Science, University of Macau, Macau
fstasp@umac.mo

Abstract. Time-series are widely used for representing non-stationary data such as weather information, health related data, economic and stock market indexes. Many statistical methods and traditional machine learning techniques are commonly used for forecasting time series. With the development of deep learning in artificial intelligence, many researchers have adopted new models from artificial neural networks for forecasting time series. However, poor performance of applying deep learning models in short time series hinders the accuracy in time series forecasting. In this paper, we propose a novel approach to alleviate this problem based on transfer learning. Existing work on transfer learning uses extracted features from a source dataset for prediction task in a target dataset. In this paper, we propose a new training strategy for time-series transfer learning with two source datasets that outperform existing approaches. The effectiveness of our approach is evaluated on financial time series extracted from stock markets. Experiment results show that transfer learning based on 2 data sets is superior than other base-line methods.

Keywords: transfer learning · financial time series · forecasting · artificial neural networks.

1 Introduction

Time-series forecasting is one of the challenging tasks in data analytics and artificial intelligence area. Time-series prediction plays a crucial role in plethora of applications such as forecasting sales, marketing, finance, and production planning etc. Traditional statistical methods in time-series forecasting include autoregressive integrated moving average (ARIMA) [16] for non-stationary data, simple exponential smoothing (SES) [9] for predicting time-series, and Holt and Damped exponential smoothing [10]. Besides, Conventional machine learning techniques have been used in time series forecasting, such as support vector regression (SVR) [19] and various hybrid methods [12]. However, time-series forecasting is a challenging task when limited data is available for training the machine learning models.

Recently, the success of deep learning models in image and Natural Language Processing (NLP) applications becomes a driving force behind the adoption of deep learning model for time-series forecasting. The information obtained during the learning processes is used for interpretation of data such as text, images and sound. The learning process also allows the computer to automatically extract the pattern features. It also integrates the feature learning into the process of modeling and therefore reduces the incompleteness caused by artificial design features. Three techniques, namely, a large number of hidden units, better learning algorithms, and parameter initialization techniques, have contributed to the success of deep learning approach [6]. However, when the dataset does not have sufficient data, deep learning approach could result poor forecasting performance [26].

In order to alleviate insufficient data problem, transfer learning is commonly used in majority of deep learning model [29]. Transfer learning is shown to be effective in computer vision [3] and NLP [23]. Despite its promising results in computer vision and related applications, transfer learning has been rarely used in deep learning models for time series data. Recently, Fawaz et al. [8] investigate how to transfer deep convolutional neural networks (CNNs) for Time Series Classification (TSC) tasks. Ye et al. [27] propose a novel transfer learning framework for time series forecasting. Most of these studies were designed to transfer features from a source dataset to a target dataset. However, in certain cases it may be necessary for the model to learn features or patterns from different source datasets when the target dataset is insufficient. This research problem has been widely considered in developing multilingual speech technologies. Most of these works focus on transferring features between languages because of the limited resources available for the target language [15]. In the time series forecasting, Hu et al. [14] combine wind speed information from multi-sources to build a deep neural network (DNN). In their model, the hidden layers are shared across many farms while the output layers are designed to be farm dependent. In their approach, the shared hidden layers can be considered as a universal feature transformation.

Motivated by these recent findings, in this paper, we investigate whether or not transfer learning can be effectively used for financial time series forecasting. Specifically, we evaluate the effectiveness of transfer learning with more than one source dataset for stock price prediction. In a more detailed case study, we also evaluate the effect of choosing different source datasets based on a similarity measure. Besides, a new training strategy for transfer learning with two source datasets is also proposed in this paper. To the best of authors' knowledge, our work makes the first attempt to investigate transfer learning for financial time series forecasting problem. Our contributions are two-fold:

- We propose a new training strategy for transfer learning with two source datasets.
- We propose a similarity based approach for selecting source datasets for training the deep learning models with transfer learning for financial time series forecasting.

The rest of the paper is structured as follows. In section II, we describe some background knowledge and review existing work on transfer learning for time series forecasting. In section III, we describe our proposed training strategy used in this paper. In section IV, we present our experiment setups and discuses the result. Finally, in section V, we conclude the paper with future work.

2 BACKGROUND AND RELATED WORK

2.1 Financial Time Series Forecasting

Financial time series forecasting is a challenging task due to the noise and volatile features of the underlying market situations [24]. Several technical indicators used for time-series prediction include auto-regression (AR), moving-average(MA), ARIMA, Holt-Winters Exponential Smoothing (HWES) and so on. However, with the development of deep learning models, DNN, recurrent neural network (RNN) and CNN have been extensively researched in time series forecasting area. Deep learning can successfully learn complex real-world data by extracting robust features that capture most useful information [13]. Ding et al. [7] combine the neural tensor network and deep CNN to predict the short–term and long–term influences of events on stock price movements. Yoshihara et al. [28] also adopted deep belief networks in financial market forcasting. However, Makridakis et al. [18] conclude that when the available data is insufficient for training, the performance of deep learning can be poorer than simple statistical methods.

2.2 Transfer Learning in Time Series Forecasting

Transfer learning aims to extract knowledge from one or more source tasks and applies the knowledge to a target task [20]. The study of Transfer learning is motivated by the fact that people can intelligently apply knowledge learned previously to solve new problems faster or with better solutions. Karl et al. [25] have defined the transfer learning as follows: "*Transfer learning for deep neural networks, is the process of first training a base network on a source dataset and task, and then transfer the learned features (the network's weights) to a second network to be trained on a target dataset and task.*" Transfer learning has been used in computer version and nature language processing. These cases include application of transfer learning in visual application by Amaral et al. [3] and initialization of the language model's weights with pre-trained weights by Prajit et al. [21]. Recently, transfer learning was also adopted for time series analysis. For example, Ye et al. [27] propose a novel transfer learning framework for time series forecasting. To calculate the similarity as the guideline of selecting source datasets, Fawaz et al. proposed a dynamic time warping (DTW) based algorithm in [8].

In [14], Hu et al. proposed an approach for the prediction of the wind speed for the new farm by transferring information from several old farms. In [14], the authors use the time series data of several old farms to pre-train a two-layers DNN model. The parameters of the model are then shared with all wind farms. The model can be considered as a universal feature transformation. In contrast to the approach proposed by Hu et al. [14], in our training strategy, a source dataset is used for training the first layer of the DNN only and the second layer is trained by using both source datasets. In addition, the parameters of these layers are not shared among the layers. Therefore, the model from our strategy not only have the universal features but also the specific features. In addition, Hu et al. [14] did not compare the performance of the model based on one source dataset with model which is built from multi-source datasets.

3 Method

In this paper, we aim to answer several questions related to transfer learning in time series prediction:

- Is transferring features from two source datasets better than transferring from one source dataset?
- What is the effect of transfer learning when the ratio for training is increased while the size for the testing is fixed in the target dataset?
- Whether or not the similarity measure of two data sets can be used as an indicator for selecting source datasets?

3.1 Training Strategy

In the context of multi-domain transfer learning, Xavier et al. [11] proposed a strategy in which a learned shared model from a set of domains is adapted for each individual target domain. The shared hidden layers can be considered as a universal feature transformation that works well for many domains. However, all source domains may not have the same influence on the performance of target domain. In this paper, we extend shared-parameters strategy proposed in [11]. In our approach, the parameters of the hidden layers not only contain the universal features, but also maintain specific features of the source domains.

The architectures of the proposed strategy is shown in Figure 1. Assume that D_s^1 and D_s^2 are two different source datasets and D_s^1 is more similar to the target dataset D_t than D_s^2. In our approach, the first layer of the proposed architecture will learn the features of D_s^1 and the second layer will learn the features of both D_s^1 and D_s^2. The proposed strategy has three steps:

1. First, we use the first source dataset to pre-train the deep learning model.
2. Next, we freeze the first layer of the model. Then we use the second source dataset to train the model.
3. Finally, we use the target dataset to fine-tune the whole deep learning model. The process of training model is similar to stack auto-encoder [4].

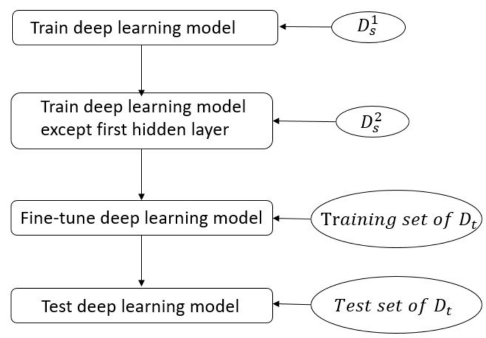

Fig. 1. Our Proposed Training Strategy.

3.2 Network Architecture

The deep learning model used in our approach comprises of an input layer, two hidden layers, and an output layer. The model is designed to be a generic model meaning that it can be replaced with a DNN or a Long Short-Term Memory (LSTM) model. Note that our training strategy proposed in the previous section is independent of the chosen network architecture.

Hu et al. [14] used two hidden layers and each number of each hidden layer contains 100 nodes. In [14], Sigmoid function is used as the activation function. Similar to their approach, in DNN model, we also used two hidden layers and increased the number of nodes in each layer to 256. Besides, we choose Tanh as activation function because it gives better training performance for multi-layer neural networks [17]. The output layer contains one unit with Linear activation function. The network is shown in Figure 2. The LSTM model used in our approach consists of two LSTM cells with Tanh as activation function and one output layer of a neuron with Linear activation function. Each LSTM layer has a 256-dimensional state vector. LSTM solves the gradient explosion and vanishing problem of RNN. The LSTM network used in this paper is shown in Figure 3.

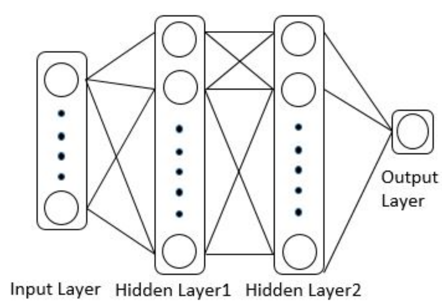

Fig. 2. Two-layers DNN Network.

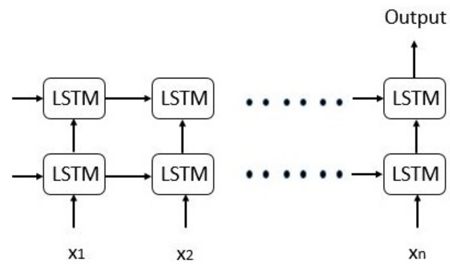

Fig. 3. Two-layers LSTM Network.

4 Experiments

4.1 Experiment Setup

During the experiment, we used DNN and LSTM models for forecasting. We implemented our network using open source deep learning library Keras [1] with the Tensorflow [2] back-end. We run our experiments on Octal Core Intel(R) Core(TM) i7-6700 CPU @ 3.40 GHz. In this paper, we have compared five strategies for testing the effectiveness of different transfer learnings in financial time series forecasting. These strategies are numbered from M1 to M5.

- M1: Training the neural network model without transfer learning.
- M2: Transfer learning from source dataset D_s^1 to target dataset D_t.
- M3: Transfer learning from source dataset D_s^2 to target dataset D_t.
- M4: Transferring learning from source datasets D_s^1 and D_s^2 to target dataset D_t with shared parameter strategy.
- M5: Transferring features from source datasets D_s^1 and D_s^2 to target dataset D_t with our proposed strategy.

The first strategy is to train the model with the target dataset without transfer learning. We denote this strategy as the baseline approach. In the second and third strategies, the model is pre-trained by only using one of the source datasets and subsequently fine-tuned by using the target dataset. In the fourth strategy, the model is pre-trained by using two source datasets with shared parameters [14] and subsequently fine-tuned by using target dataset. In the last strategy, the model is pre-trained by using two source datasets with our proposed training strategy from section 3.1. In the experiments, we adopt similar Hyper-parameters from [8] except that the epochs and batch size are set to 100 and 200 respectively. Besides, the learning rate of fine-tune step is also set to 0.00001. The Hyper-parameters used for the models are listed in Table 1.

Hyperparameter	Baseline	Pre-train	Fine-tune
Epochs	100	100	100
Batch size	200	200	200
Optimizer	Adam	Adam	Adam
Learning rate	0.001	0.001	0.00001
First moment	0.9	0.9	0.9
Second moment	0.999	0.999	0.999
Loss function	Cross-entropy	Cross-entropy	Cross-entropy

Table 1. The hyperparameters used for the experiments.

4.2 Evaluation

After the learning process, the output of the model are inverse-normalized before computing the indicators. In this paper, we choose three classical indicators ($MAPE$, $RMSE$ and R^2) to measure the predictive accuracy of each model. The definitions of these indicators are as follows:

$$MAPE = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \tag{1}$$

$$RMSE = \sqrt{\frac{\sum (y_i - \hat{y}_i)^2}{n}} \tag{2}$$

$$R^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \overline{y}_i)^2} \tag{3}$$

In these equations, y_i is the actual value and $\hat{y}_i$ is the predicted value. n represents the prediction period. $MAPE$ measures the size of the error. $RMSE$ is the mean of the square root of the error between the predicted value and the true value. R^2 is used for evaluating the fitting situation of the prediction model. The lower the $MAPE$ and $RMSE$, the better the model in forecasting. In contrast, higher the R^2, better the trained model.

4.3 Datasets

In the experiments, we use the stock market data from Yahoo Finance (https://finance.yahoo.com/). We choose three different groups of source and target datasets. The first group of datasets include Hang Seng Commerce & Industry (HSNC, 1000 time points) and Hang Seng Properties Index (HSNP, 1000 time points) as source datasets and Hang Seng Finance Index (HSNF, 1000 time points) as target dataset. The second datasets contain Dax Performance-Index (DAX, 5000 time points) and CAC 40 (CAC, 5000 time points) as source datasets and FTSE 100 (FTSE, 4000 time points) as the target dataset. Third datasets include S&P 500 (GSPC, 10000 time points) and Nasdaq (IXIC, 10000 time points) as source datasets, and Dow 30 (DJI, 8000 time points) as target dataset. These groups are different in size and we label them as small, mid-size, and large datasets.

Time series data are preprocessed before they are used for deep learning model. First, we drop missing and abnormal values. We also transform the time series into acceptable data format by the DNN and LSTM. For a given time-stamp t (day), the input vector x consists of 90-day historical stock price: $\mathbf{x} = [p_{(t)}, \ldots, p_{(t-89)}]$ and the output vector y consists of 1-day stock price from time t: $\mathbf{y} = p_{t+1}$. We trained the model using 90 days for the lookback and 1 day for the forecast horizon. The value of time series are min-max scaled to [-1, 1] interval. During the experiments, we used the 70% and 30% of the target dataset for training and testing. The test size of target dataset of HSNF, FTSE, DJI are 273, 1173, and 2373 respectively.

4.4 Results and Discussion

In the experiments, we compare the performance of our strategy with the shared parameter strategy for multi-source datasets. We also examine the effect of transfer learning when the ratio for training is increased while the size for the testing is fixed in the target dataset. The experiment results are listed in Table 2. From these results, we can observe that the model with transfer learning have significant impact on time series forecasting. Besides, transfer learning with two source datasets (M4 and M5) is better than transfer learning with one source dataset (M2 and M3). Our proposed strategy (M5) achieve good results in majority of the cases.

$\mathbf{D}_s^1$	$\mathbf{D}_s^2$	$\mathbf{D}_t$	Strategy	DNN			LSTM		
				MAPE	***RMSE***	$\boldsymbol{R^2}$	***MAPE***	***RMSE***	$\boldsymbol{R^2}$
HSNC	**HSNP**	**HSNF**	**M1**	1.2445	622.0425	0.9569	1.5868	801.2603	0.9291
			M2	1.0356	522.7943	0.9698	1.0088	511.6846	0.9711
			M3	1.0297	518.9767	0.9702	0.9899	507.3583	0.9716
			M4	0.9944	505.0655	0.9718	**0.9618**	494.5743	0.9730
			M5	**0.9851**	**499.9882**	**0.9724**	0.97348	**492.6246**	**0.9732**
DAX	**CAC**	**FTSE**	**M1**	0.9380	79.9719	0.9718	0.7254	63.7207	0.9835
			M2	0.7020	62.6288	0.9841	**0.6533**	**58.6790**	**0.9861**
			M3	0.6832	61.3956	0.9847	0.6817	60.3091	0.9853
			M4	**0.6708**	**60.4791**	0.9852	0.6750	59.8770	0.9855
			M5	0.6752	60.5211	**0.9852**	0.6675	59.4312	0.9857
GSPC	**IXIC**	**DJI**	**M1**	0.9287	212.3614	0.9978	1.2121	289.9345	0.9959
			M2	0.7428	172.5451	0.9985	0.7162	166.7077	0.9986
			M3	0.7023	168.2734	0.9986	0.7469	174.1719	0.9985
			M4	0.9489	210.1081	0.9979	0.6807	159.6134	0.9987
			M5	**0.6858**	**158.5243**	**0.9988**	**0.6787**	**158.7363**	**0.9988**

Table 2. Experiment results for different transfer learning strategies.

The experiment results for the effect of transfer learning when the ratio for training is increased while the size for the testing is fixed in the target dataset are shown in Figure 4, 5, and 6. The experiment settings for testing with different ratio of target training dataset for fine-tuning is similar to previous experiments except that we use 20%, 40%, 60%, 80%, 100% of target training dataset for fine-tuning. In these experiments, the size of test datasets for target dataset of HSNF, FTSE, DJI are kept constant at 273, 1173, and 2373 respectively. From 4, 5, and 6, we can observe that DNN model with transfer learning is better than the model without transfer learning regardless of the size of training dataset. We also found that M5 (LSTM) gained good result in majority of the cases.

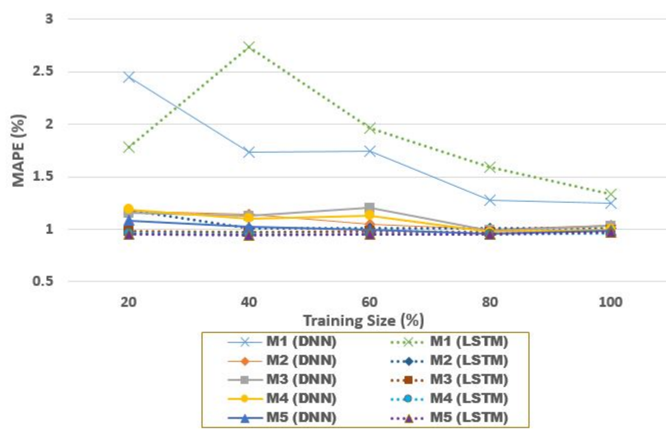

Fig. 4. The MAPE of HSNF dataset.

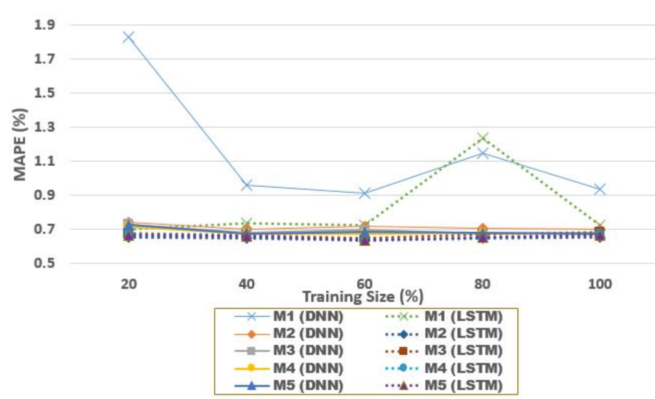

Fig. 5. The MAPE of FTSE dataset.

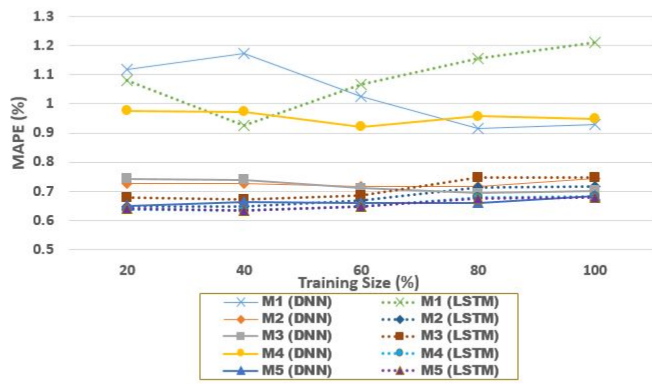

Fig. 6. The MAPE of DJI dataset.

4.5 Additional Experiment with Similarity Measure for Source Datasets

Rosenstein et al. [22] empirically showed that if two datasets are dissimilar, then brute-force transfer may negatively effect the performance of the target dataset. Such effect is also labeled as negative transfer by [22]. In this section, we further examine the effect of similarity between the source and target datasets on the overall results of transfer learning strategies. In this paper, we adopt Dynamic Time Warping (DTW) [5] algorithm to calculate the similarity of two time series (two datasets). DTW does not require that the two time series to be in the same length. DTW also permits time shifting between the two time series. Therefore, in the following four experiments, we use the mean value of two DTW distances as the main criteria to find potential source datasets for a given target dataset. The equation of $mDTW$ (mean DTW distance) is defined as follow:

$$mDTW = \frac{DTW(D_s^1, D_t) + DTW(D_s^2, D_t)}{2} \tag{4}$$

The smaller the mDTW, the more similar between the source and target dataset. In the four experiments, time series of Bank of China (BOC) is used as the target datasets for all transfer learning strategies. Source datasets for the experiments include China Construction Bank Corporation (CCB) and Industrial and Commercial Bank of China Limited (ICBC), Alibaba (BABA) and Lenovo Group (Leveno), Hang Seng Bank Limited (HSB) and Bank of America (BOA), Tencent and CLP Group (CLP). In experiment 1 and 3, the source datasets are selected from the same industry (i.e. Finance). In experiment 2 and 4, the source datasets are selected from the different industries. In Experiment 1 and 2, we evaluate the effect of small distance ($mDTW = 0.6$) between source and target datasets and Experiment 3 and 4 are designed to evaluate the effect of larger distance ($mDTW > 0.15$). The results of the experiments are listed in Table 3.

From these results, we can observe that selecting the source and target datasets from the same industry produces the best results. We can also observe that for the case of same industry with small $mDTW$, the results of M4 (DNN) and M5 (LSTM) are superior. The results of M5 (LSTM) is also superior for the case of selecting source and target datasets from the same industry with large $mDTW$ value.

Exp	$\mathbf{D}_s^1$	$\mathbf{D}_s^2$	$\mathbf{D}_t$	**mDTW**	**Strategy**	**DNN**			**LSTM**		
						$\boldsymbol{MAPE}$	$\boldsymbol{RMSE}$	$\boldsymbol{R^2}$	$\boldsymbol{MAPE}$	$\boldsymbol{RMSE}$	$\boldsymbol{R^2}$
					M1	1.3095	0.0672	0.9735	1.3271	0.07107	0.9705
Exp1	**CCB**	**ICBC**	**BOC**	**0.06**	**M2**	1.1568	0.0609	0.9783	1.0011	0.0558	0.9819
					M3	1.1419	0.0592	0.9795	0.9765	0.0543	0.9828
					M4	**1.0397**	**0.0551**	**0.9823**	0.9817	0.0551	0.9823
					M5	1.0631	0.0560	0.9817	**0.9780**	0.0547	0.9825
Exp2	**BABA**	**Lenovo**	**BOC**	**0.06**	**M2**	1.1499	0.0601	0.9789	1.0264	0.0567	0.9812
					M3	1.2798	0.0656	0.9749	1.4130	0.0768	0.9656
					M4	1.0769	0.0585	0.9800	1.0060	0.0561	0.9816
					M5	1.0927	0.0586	0.9800	1.0070	0.0552	0.9822
Exp3	**HSB**	**BOA**	**BOC**	**0.15**	**M2**	1.1989	0.0634	0.9766	1.1064	0.0601	0.9790
					M3	1.2582	0.0648	0.9755	1.0459	0.0563	0.9815
					M4	1.1565	0.0613	0.9781	0.9861	0.0548	0.9825
					M5	1.1287	0.0597	0.9792	0.9828	**0.0542**	**0.9829**
Exp4	**Tencent**	**CLP**	**BOC**	**0.17**	**M2**	1.2217	0.06404	0.9761	1.4027	0.0734	0.9685
					M3	1.1988	0.0613	0.9781	1.0334	0.0560	0.9817
					M4	1.1434	0.0599	0.9790	0.9899	0.0556	0.9819
					M5	1.1119	0.0585	0.9800	1.0019	0.0563	0.9815

Table 3. Transfer learning strategies for different DTW distance and industry.

5 Conclusion

In this paper, we propose a new training strategy for transfer learning with two source datasets. The experiment results reveal that that the model with transfer learning has positive impact on financial time series forecasting. The experiment results also reveal that transfer learning with more source datasets is superior than using a single source dataset. In addition, the proposed training strategy (M5) achieve good results in majority of the cases. Although the proposed strategy is tested with only 2 source datasets, it can be extended for training more datasets after additional hidden layers are added to the network architecture. A similarity based approach based on DTW for selecting source and target datasets for training the deep learning models with transfer learning for financial time series forecasting is also proposed in this paper. Experiment results show that the transfer learning with similar (i.e. smaller $mDTW$) source datasets from the same industry is superior than selecting source datasets from different industries. In the paper, we use the DTW distance to calculate the similarity of two time series. As for the future work, we are planning to investigate the effect of other distance functions on the transfer learning strategies. In addition, we are also planning to test the effect of more than two source datasets on the training outcome. We are also conducting experiments for the deep learning neural networks with varying number of hidden layers.

6 Acknowledgement

The research was funded by the Research Committee of University of Macau, Grant MYRG2018-00246-FST.

References

1. Keras. `https://keras.io/` (2015)
2. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al.: Tensorflow: A system for large-scale machine learning. In: 12^{th} USENIX Symposium on Operating Systems Design and Implementation (OSDI 16). pp. 265–283 (2016)
3. Amaral, T., Silva, L.M., Alexandre, L.A., Kandaswamy, C., de Sá, J.M., Santos, J.M.: Transfer learning using rotated image data to improve deep neural network performance. In: International Conference Image Analysis and Recognition. pp. 290–300. Springer (2014)
4. Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H.: Greedy layer-wise training of deep networks. In: Advances in Neural Information Processing Systems. pp. 153–160 (2007)
5. Berndt, D., Clifford, J.: Using dynamic time warping to find patterns in time series. In: KDD Workshop. vol. 10/16, pp. 359–370 (1994)
6. Deng, L., Yu, D.: Deep learning for signal and information processing. Foundations and Trends in Signal Processing **2-3**, 197–387 (2013)
7. Ding, X., Zhang, Y., Liu, T., Duan, J.: Deep learning for event-driven stock prediction. In: Proceedings of the 24^{th} International Conference on Artificial Intelligence. pp. 2327–2333. AAAI Press (2015)
8. Fawaz, H.I., Forestier, G., Weber, J., Idoumghar, L., Muller, P.A.: Transfer learning for time series classification. In: 2018 IEEE International Conference on Big Data (Big Data). pp. 1367–1376. IEEE (2018)
9. Gardner Jr, E.S.: Exponential smoothing: The state of the art. International Journal of Forecasting **4**(1), 1–28 (1985)
10. Gardner Jr, E.S.: Exponential smoothing: The state of the art—part ii. International Journal of Forecasting **22**(4), 637–666 (2006)
11. Glorot, X., Bordes, A., Bengio, Y.: Domain adaptation for large-scale sentiment classification: A deep learning approach. In: Proceedings of the 28^{th} International Conference on Machine Learning (ICML-11). pp. 513–520 (2011)
12. Haque, A.U., Nehrir, M.H., Mandal, P.: A hybrid intelligent model for deterministic and quantile regression approach for probabilistic wind power forecasting. IEEE Transactions on Power Systems **29**(4), 1663–1672 (2014)

13. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science **313**(5786), 504–507 (2006)
14. Hu, Q., Zhang, R., Zhou, Y.: Transfer learning for short-term wind speed prediction with deep neural networks. Renewable Energy **85**, 83–95 (2016)
15. Huang, J.T., Li, J., Yu, D., Deng, L., Gong, Y.: Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. pp. 7304–7308. IEEE (2013)
16. Hyndman, R., Khandakar, Y.: Automatic time series forecasting: The forecast package for R. Journal of Statistical Software **27**(3), 1–22 (2008)
17. Karlik, B., Olgac, A.V.: Performance analysis of various activation functions in generalized mlp architectures of neural networks. International Journal of Artificial Intelligence and Expert Systems **1**(4), 111–122 (2011)
18. Makridakis, S., Spiliotis, E., Assimakopoulos, V.: Statistical and machine learning forecasting methods: Concerns and ways forward. PLOS ONE **13**(3), 1–26 (2018)
19. Ortiz-García, E.G., Salcedo-Sanz, S., Pérez-Bellido, Á.M., Gascón-Moreno, J., Portilla-Figueras, J.A., Prieto, L.: Short-term wind speed prediction in wind farms based on banks of support vector machines. Wind Energy **14**(2), 193–207 (2011)
20. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering **22**(10), 1345–1359 (2009)
21. Ramachandran, P., Liu, P., Le, Q.: Unsupervised pretraining for sequence to sequence learning. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. pp. 383–391. Association for Computational Linguistics (2017)
22. Rosenstein, M.T., Marx, Z., Kaelbling, L.P., Dietterich, T.G.: To transfer or not to transfer. In: NIPS 2005 Workshop on Transfer Learning. vol. 898, pp. 1–4 (2005)
23. Vu, N.T., Imseng, D., Povey, D., Motlicek, P., Schultz, T., Bourlard, H.: Multilingual deep neural network based acoustic modeling for rapid language adaptation. In: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 7639–7643. IEEE (2014)
24. Wang, B., Huang, H., Wang, X.: A novel text mining approach to financial time series forecasting. Neurocomputing **83**, 136–145 (2012)
25. Weiss, K., Khoshgoftaar, T.M., Wang, D.: A survey of transfer learning. Journal of Big Data **3**(1), 9 (2016), `https://doi.org/10.1186/s40537-016-0043-6`
26. Wu, D.D., Olson, D.L.: Financial risk forecast using machine learning and sentiment analysis. In: Enterprise Risk Management in Finance, pp. 32–48. Springer (2015)
27. Ye, R., Dai, Q.: A novel transfer learning framework for time series forecasting. Knowledge-Based Systems **156**, 74–99 (2018)
28. Yoshihara, A., Fujikawa, K., Seki, K., Uehara, K.: Predicting stock market trends by recurrent deep neural networks. In: Pacific Rim International Conference on Artificial Intelligence. pp. 759–769. Springer (2014)
29. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In: Advances in Neural Information Processing Systems. pp. 3320–3328 (2014)