IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 8, 2021, accepted April 26, 2021, date of publication May 3, 2021, date of current version May 11, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3077004

Stock Price Prediction Based on Morphological Similarity Clustering and Hierarchical Temporal Memory

XINGQI WANG, KAI YANG, AND TAILIAN LIU
Science and Information College, Qingdao Agricultural University, Qingdao 266109, China
Corresponding author: Tailian Liu (tailian12@163.com)

This work was supported in part by the Shandong Provincial Agricultural Science and Technology Fund under Grant 2019YQ015, and in part by the Talents of High Level Scientific Research Foundation of Qingdao Agricultural University under Grant 663/1115004.

ABSTRACT Predicting stock prices through historical data is a hot research topic. Stock price data is considered to be a typical time series. Recurrent neural network (RNN), long short-term memory (LSTM), and gated recurrent units (GRU) have been commonly employed to handle this type of data. However, most studies focus on the analysis of individual stocks, thus ignoring the correlation between similar stocks in the entire stock market. This paper proposes a clustering method for mining similar stocks, which combines morphological similarity distance (MSD) and kmeans clustering. Subsequently, Hierarchical Temporal Memory (HTM), an online learning model, is used to learn patterns from similar stocks and make predictions at last, denoted as C-HTM. The experiments on the price prediction show that 1) compared with HTM which has not learned similar stock patterns, C-HTM has better prediction accuracy, 2) in terms of short-term prediction, the performance of C-HTM is better than all baseline models.

INDEX TERMS Machine learning, kmeans, morphological similarity distance, hierarchical temporal memory, stock prediction.

I. INTRODUCTION

The non-random walk hypothesis and the efficient market hypothesis suggest that historical stock data are of great commercial value and that the study of past prices can be used to predict future prices [1]–[3]. Moreover, an promising stock prediction model has been proven to bring considerable benefits to investors and companies. Although the stock price prediction task is attractive to researchers, it is still considered to be a challenging problem because the stock data is real-time, high-noise and nonlinear. Therefore, many scholars try to use various methods to achieve a better accuracy [4]–[7]. Machine learning models have shown more promising prospects than traditional statistical models such as Autoregressive Integrated Moving Average (ARIMA), a time-series prediction model which utilizes differences. Nabipour *et al.* [8] evaluated the performance of artificial neural networks (ANNs), recurrent neural network (RNN), long short-term memory (LSTM) and six tree-based models (decision tree, bagging, random forest, adaboost, gradient boosting, and xgboost). The results show that LSTM was the top performer in comparison with other techniques. Chen and Zhou [9] employed a genetic algorithm (GA) for feature selection, in order to improve perfomance of LSTM. However, to the best of our knowledge, there is little research work that deals with the issue of correlation between multiple stocks.

Another issue of some previous works is that their models are prone to overfitting or underfitting, which means that the parameters need to be adjusted frequently. Hence, the following features are necessary for the model: 1) Due to the real-time nature of stock data, the model can continue to learn and does not require excessive parameter adjustments. 2) The model is sensitive to input, thus it can learn the potential patterns. 3) The model has high robustness and fault tolerance mechanism to adapt to the high-noise data environment.

To address those two issues, the method proposed in this paper is that employing a clustering method based on kmeans to find similar stocks in the stock market, which uses morphological similarity distance (MSD) as a measure of similarity,

The associate editor coordinating the review of this manuscript and approving it for publication was K. Kotecha.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

denoted as K-MSD. The MSD has been proven to be more suitable for evaluating the similarity of time series [10]. In addition, we use Hierarchical Temporal Memory (HTM) model, a biologically-constrained theory of intelligence originally described in [11], to learn patterns among similar stocks, because its data structure of sparse distributed representations (SDRs) ensures the robustness and sensitivity. The experiment results show that compared with HTM which has not learned similar stock patterns, the HTM after clustering, that we called C-HTM, has better prediction accuracy. Furthermore, in terms of short-term prediction, the performance of C-HTM is better than other three baseline models.

The main contributions of this work are summarized as follows:

- This work is the first to implement KMSD clustering algorithm, in order to mining similar stocks in the entire stock market.
- We apply the HTM model, an online learning model, to learn potential stock patterns on best clusters, which achieves promising performance on short-term stock price prediction tasks.

The rest of this paper is organized as follows: In Section II, recent researches on stock price prediction are presented. The methods used in this paper are introduced in detail in Section III. In Section IV, experiment and results are reported followed by conclusion and future work directions in Section V.

II. RELATED WORKS

There are a lot of researches on stock price prediction recently. We only broadly introduce part of the previous works.

Based on recent research, clustering algorithms have been widely used in stock price prediction task [12]–[14]. Xu *et al.* [12] proposed a hybrid two-stage stock forecasting method based on clustering and ensemble learning. In this method, kmeans is employed to cluster different technique factors which affect the stock price. Li and Wu [13] utilize hierarchical clustering algorithms to cluster the stock time series windows into different categories, in order to improve stock price predictions with the help of market styles. Nakagawa *et al.* [14] proposed a k-medoids clustering with Indexing dynamic time warping (IDTW) to grasp price fluctuation patterns useful for prediction. In [15], Kumari *et al.* used the CUDA parallel computing framework to accelerate clustering operations.

In addition, many models specifically dealing with stock data have been proposed. Hoseinzade and Haratizadeh [16] trained a convolutional neural network (CNN) model which takes a 3-dimensional tensor to aggregate and align a diverse set of variables as input. An elman neural network (ENN) model optimized by grey wolf optimization (GWO) algorithm achieves promising stock predictive performance on the task of predicting the closing price for one day in advance, proposed by Chandar *et al.* in [17]. Pang *et al.* [18] use the embedded layer and the automatic encoder, respectively, to vectorize the data for LSTM. The experimental results show that the LSTM with embedded layer is better.

Currently, few studies focus on analyzing the impact of stock similarity on the model. The model learning the pattern of similar stocks means high robustness and high generalization, because the stock price is affected by unpredictable factors. Therefore, this paper trains the HTM model on similar stocks which are clustered by KMSD to improve the prediction accuracy.

III. METHODOLOGY

In this section, we introduce the method used in this paper, including KMSD clustering, HTM, baseline models and evaluation measures for clustering and prediction.

A. KMEANS WITH MORPHOLOGICAL SIMILARITY DISTANCE

Kmeans [19] is a common clustering algorithm in machine learning tasks. Given n samples of time series $(x_1, x, \ldots, x_n)$, where each sample is a d-dimensional real vector, kmeans clustering aims to partition the n samples into $k(\leq N)$ sets $S = (S_1, S_2, \ldots, S_i)$. In other words, its goal is to find the cluster S_i that satisfies the following formula:

$$\arg\min_{S} \sum_{i=1}^{k} \sum_{x \in S_i} f(x, \mu_i) \tag{1}$$

where μ_i is the mean value of all series in S_i, the function f is the similarity between sample x and μ_i. We use MSD as the similarity measure, so f can be expressed as follows:

$$\begin{aligned} f(x, \mu_i) &= ED \times \left(2 - \frac{ASD}{SAD}\right) \\ &= \sqrt{\sum_{j=1}^{d} (x_j - \mu_{ij})^2} \times \left(2 - \frac{|\sum_{j=1}^{d} (x - \mu_i)|}{\sum_{j=1}^{d} |x - \mu_i|}\right) \end{aligned} \tag{2}$$

where ED is Euclidean distance, ASD is the absolute sum of the difference, and SAD is Manhattan distance.

B. HIERARCHICAL TEMPORAL MEMORY

In this section, we only make a brief introduction to HTM. For readers who are not familiar with HTM, please refer to literature [20], [21].

1) HTM STRUCTURE

HTM is an unsupervised machine learning technology that simulates the working principle of the neocortex of the mammalian (mainly human) brain, proposed by [11] originally. The structure of an abstract HTM model with two level is shown in Figure 1. A typical HTM model is a tree-like hierarchical structure. Each level is composed of smaller elements called regions, while single level in a hierarchical structure may contain multiple regions. Generally, the higher a level is in the model, the less regions it contains. Furthermore, each region is composed of columns of multiple neurons.

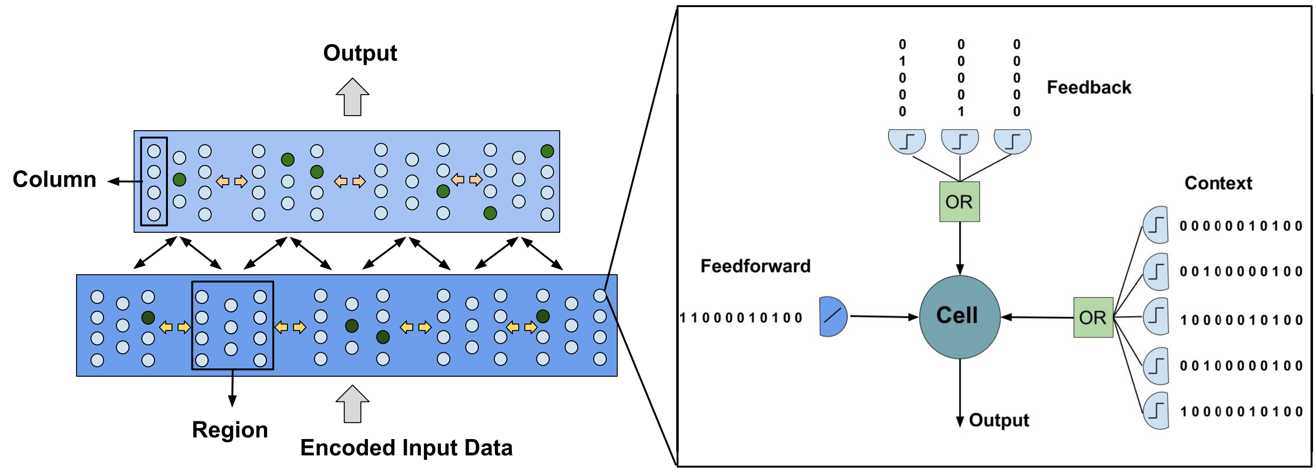

FIGURE 1. A two-level HTM model.

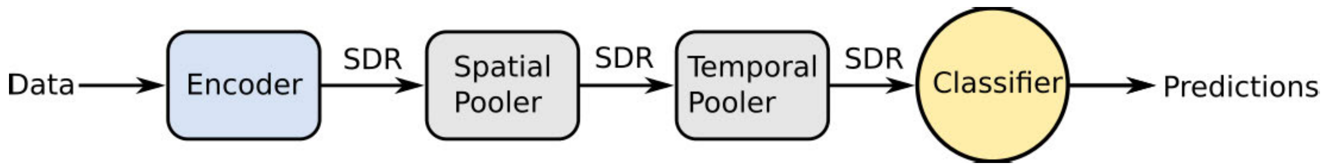

FIGURE 2. Workflow of HTM model.

The design of neuron is like the pyramidal cells in the brain. It is a multipolar neuron that includes three types of dendrites: proximal dendrites, which receive feedforward input information; terminal dendrites, which receive contextual information; apical dendrites, which accept feedback.

2) HTM WORKFLOW

As shown in Figure 2, the HTM workflow mainly includes four parts: encoding, spatial pooling, temporal pooling and classifier prediction. Firstly, the encoder [22] transforms the input into sparse distributed representations (SDRs), data structure composed of binary, in which the number of 1 is much smaller than 0. SDRs is the basis for the robustness of the model. Then, the spatial pooling algorithm [23] and the temporal pooling algorithm integrate similar patterns and divide time groups, respectively. Finally, the prediction result of the model is given by the classifier.

3) SPARSE DISTRIBUTED REPRESENTATION

As far as nerual network is concerned, sparsity terms has been proven to improve prediction accuracy in [24]–[26]. As sparse data representation in HTM, SDRs ensures the robustness of noise and the sensitivity to input. An SDR consists of thousands of binary bits in which 1 represents a relatively active neuron and a 0 represents a relatively inactive neuron. However, the number of 1 is much smaller than 0, which is why we think SDRs is sparse. In general, SDR is considered to be the core concept of HTM.

C. BASELINE MODELS

In this paper, we use RNN [27], LSTM [28] and GRU [29] as baseline models. As a class of neural network, RNN is widely used in processing variable-length time series because of its internal memory. In addition, LSTM and GRU are variants of RNN, which avoid the gradient vanishing problem [30] and prove to be more suitable for processing long time series.

In the experiment, we will use the common sliding window method to train the baseline models. Figure 3 is an example indicating how to use previous 7 days to predict the future 1 days.

D. EVALUATION MEASURES

The method in this paper can be divided into two stages: clustering and prediction. Learning and prediction are performed in a better cluster, so the evaluation of clustering and prediction need to be given separately.

1) CLUSTERING METRIC

Silhouette is an evaluation method of clustering effect, proposed by [31]. Its value is a measure of how similar an sample is to its own cluster (cohesion) compared to other clusters (separation), which ranges from -1 to $+1$. A high value indicates that the sample is well matched to its own cluster and poorly matched to neighboring clusters. The calculation method of the Silhouette of sample i is as follows:

$$s(i) = \begin{cases} 1 - a(i)/b(i), & \text{if } a(i) < b(i) \\ 0, & \text{if } a(i) = b(i) \\ b(i)/a(i) - 1, & \text{if } a(i) > b(i) \end{cases} \tag{3}$$

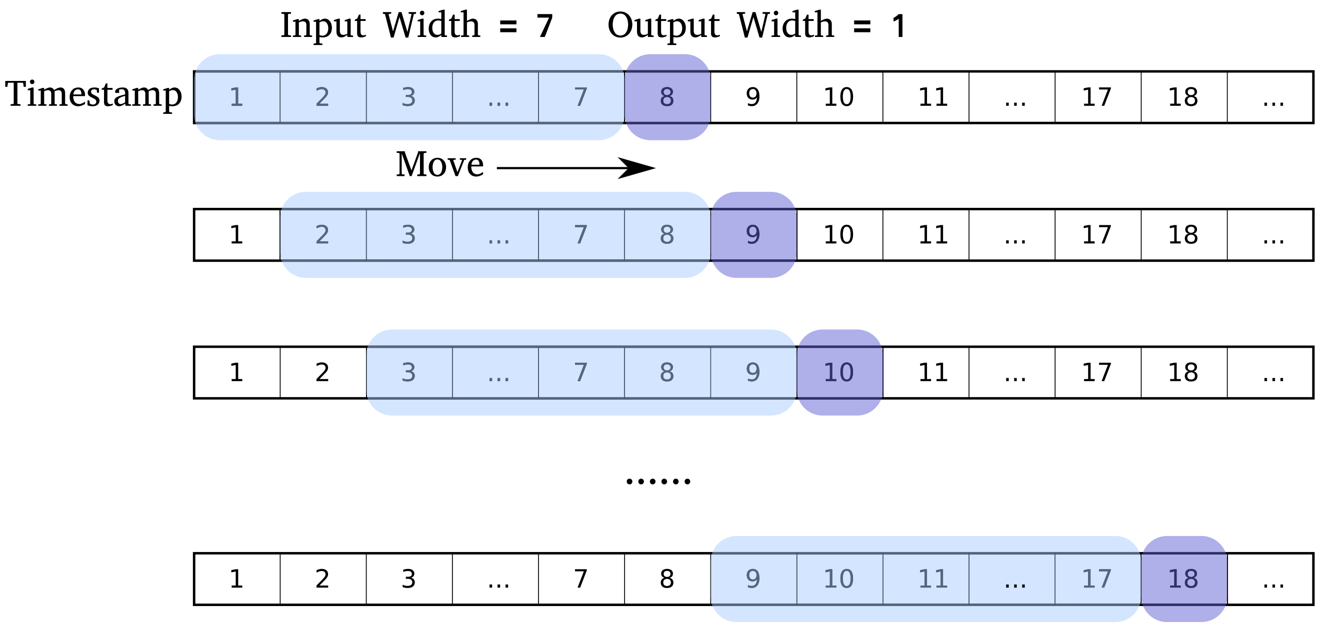

FIGURE 3. An example of silding window.

where $a(i)$ is the mean distance between i and all other samples in the same cluster, $b(i)$ is the smallest mean distance of i to all samples in any other cluster.

Assume all samples have been clustered via KMSD clustering into k clusters. If the average Silhouette of a cluster is greater than or equal to the average Silhouette of all samples, the cluster can be regarded as a better cluster.

2) PREDICTION METRIC

Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE) and coefficient of determination (R^2) are used in this paper to evaluate the performance of prediction models. Their formulas are as follows:

$$MAE = \frac{\sum_{i=1}^{n} \left|\hat{y}_i - y_i\right|}{n} \tag{4}$$

$$MSE = \frac{\sum_{i=1}^{n} (\hat{y}_i - y_i)^2}{n} \tag{5}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (\hat{y}_i - y_i)^2}{n}} \tag{6}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (\hat{y}_i - y_i)^2}{\sum_{i=1}^{n} (\bar{y} - y_i)^2} \tag{7}$$

where y_i is the actual value, $\hat{y}_i$ represents the predicted value, $\bar{y}$ is the mean value of actual value. In the best case, the predicted values exactly match the actual values, which results in $MAE = 0$, $MSE = 0$, $RMSE = 0$ and $R^2 = 1$.

IV. EXPERIMENT AND RESULTS

A. OVERVIEW

In this experiment, we select opening price as our forecasting target. The overview of the experiment in this paper is shown in Figure 4. Firstly, the input of KMSD clustering is the opening price of the all samples in the training set. Then, the average Silhouette are employed to select the better clusters. Finally, in order to verify the applicability of KMSD and HTM, the model will be trained in the following ways:

- Comparing models which are trained in the better cluster and models without clustering to determine whether KMSD can improve the prediction accuracy.
- Comparing univariate and multivariate inputs to determine whether the model can learn more knowledge through multiple features.
- Comparing the performance of HTM and baseline models to determine whether HTM is more suitable to stock data.

In addition, the input data need to be normalized before clustering and training so that the values of variables are between [0,1]. The normalization formula is as follows:

$$X_{normailization} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{8}$$

At last, the prediction result need to be denormalized as follows:

$$X = X_{normailization} \times (X_{max} - X_{min}) + X_{min} \tag{9}$$

where X_{max} is the maximum value of data, X_{min} is the minimum value of data.

B. DATASET

The original dataset used in the experiment is the historical data of China A-shares since listing to 2021-02-26, with a total of 4112 samples. Each stock sample contains 13 features: closing price, high price, low price, opening price,

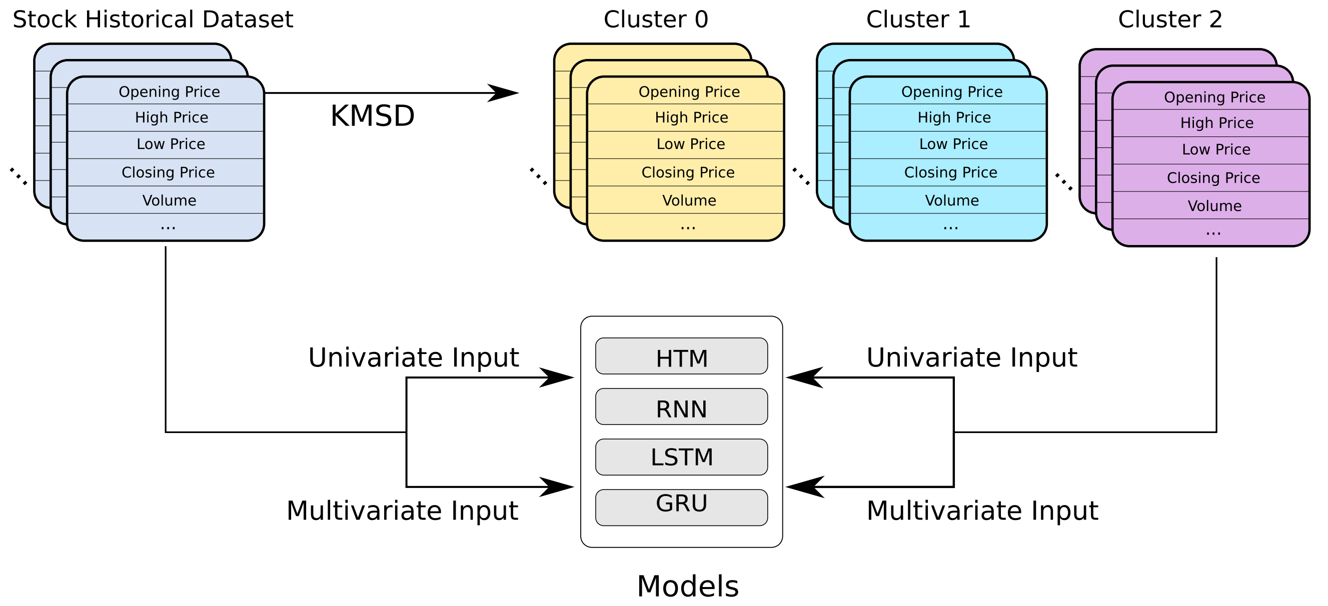

FIGURE 4. **Overview of the experiment process.**

TABLE 1. **HTM parameters.**

Parameter	Value(s)
Number of columns N	2048
Number of cells per column M	32
Dendritic segment activation threshold θ	14
Initial synaptic permanence	0.21
Connection threshold for synaptic permanence	0.5
Synaptic permanence increment p^{+}	0.05
Synaptic permanence decrement p^{-}	0.05
Synaptic permanence decrement for predicted inactive segments p^{-}	0.01
Maximum number of segments per cell	128
Maximum number of synapses per segments	128
Maximum number new synapses added at each step	32
Multistep inferences (N-days)	1, 7 ,15 ,30

TABLE 2. **Baseline models parameters.**

Parameter	Value(s)
Hidden Layers	2
Inputs	7, 15, 30, 50
Outputs (N-days)	1, 7, 15, 30
Activation Function	tanh
Loss function	Mean squared error
Optimizer	RMSprop
Learning Rate	0.001
Epochs	100

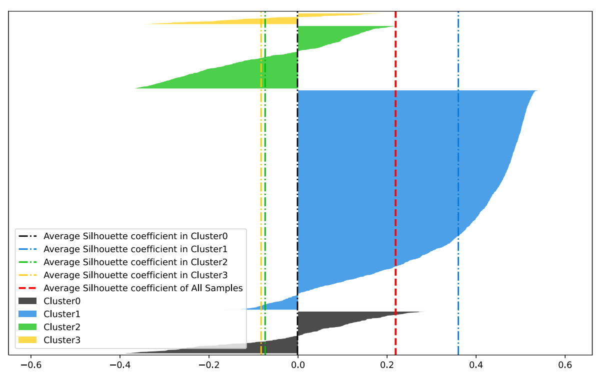

FIGURE 5. **The silhouette values of cluster with K = 4.**

previous closing, up/down amount, up/down range, turnover rate, volume, turnover amount, total market capitalization, market capitalization outstanding and number of transactions. Then, stock samples with data from 2011-01-04 to 2021-02-26 are screened out. So a total of 1957 samples are used as our actual experimental data. The date length of each sample is 2467. Finally, the first 2100 days of each sample are used as the test set, and the last 367 days are used as the test set to verify our method.

C. MODEL PARAMETERS

Table 1 presents the parameters of the HTM model referred from [32]. The parameter N-days represents the number of days to predict in the future. These prediction results will be given in an HTM model, which we call simultaneous multi-step prediction.

TABLE 3. Silhouette values.

K	Cluster 0	1	2	3	4	5	6	Average
3	-0.0401	0.2265	-0.0287					0.1385
4	-0.0013	**0.3601**	-0.0739	-0.0830				**0.2192**
5	0.3623	-0.1022	-0.0031	-0.0781	-0.0819			0.1920
6	0.0000	0.3434	-0.0735	-0.1182	-0.0345	0.0129		0.1690
7	0.0514	-0.0190	0.0000	-0.0095	-0.0795	0.3247	-0.0192	0.1503

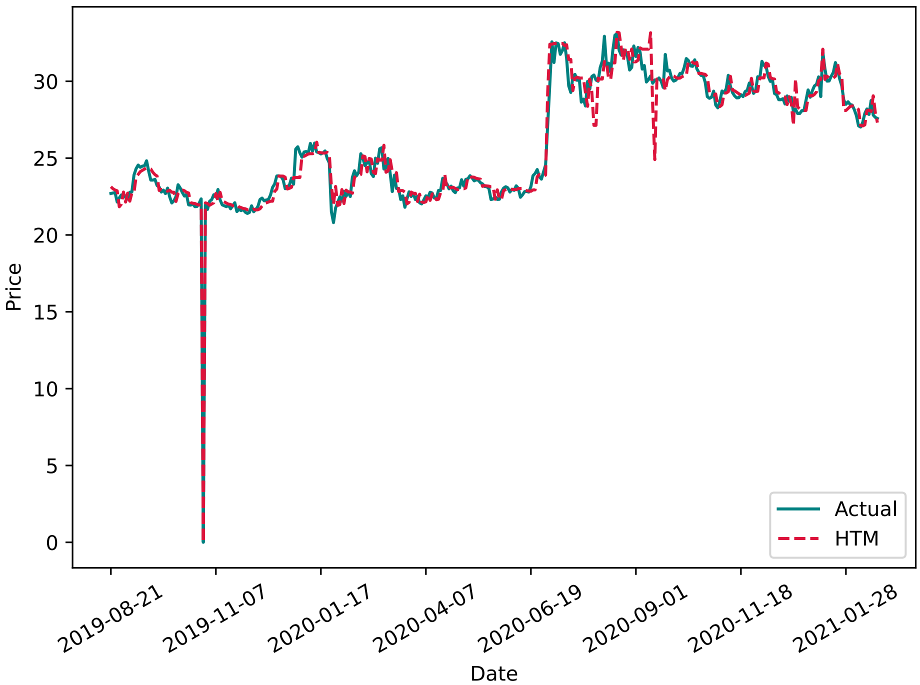

FIGURE 6. The result of HTM aftering clustering with 1-day ahead.

To evaluate our method, the parameters of the baselines model (RNN, LSTM and GRU) are shown in Table 2. We set the length of the input window as 7, 15, 30 and 50 respectively to correspond to output window which is same as the output of HTM.

D. CLUSTERING RESULT

Table 3 presents the results obtained from KMSD clustering setting K from 3 to 7. It is apparent that all samples tend to be divided into 4 clusters, because the average Silhouette is 0.2192 which is closer to 1 compered to other K values. Then, the result of the clustering are more intuitively shown in Figure 5. What stands out in the figure is that the cluster 1 is the best cluster, so models will be trained on it.

TABLE 4. Results of univariate input without clustering.

N-days	Prediction Models	Error Measures MAE	R2	MSE	RMSE
1	RNN	0.6923	0.8226	2.5888	1.6090
	LSTM	2.7342	0.3731	9.1482	3.0246
	GRU	2.0819	0.5737	6.2213	2.4943
	HTM	1.3408	0.2839	10.4244	3.2287
7	RNN	0.7752	0.8223	2.6250	1.6202
	LSTM	1.9504	0.6221	5.5814	2.3625
	GRU	1.6843	0.6787	4.7447	2.1782
	HTM	2.0831	-0.3243	19.2773	4.3906
15	RNN	1.7602	0.6511	5.2422	2.2896
	LSTM	1.5457	0.7116	4.3334	2.0817
	GRU	1.1624	0.7794	3.3139	1.8204
	HTM	2.3056	-0.4076	20.4899	4.5266
30	RNN	2.8318	0.2888	10.5756	3.2520
	LSTM	3.2688	0.1501	12.6377	3.5549
	GRU	1.6932	0.6256	5.5677	2.3596
	HTM	3.3498	-1.3291	33.9023	5.8226

E. PREDICTION RESULTS

In this section, we show the performance of all models on stock SHA:600030 which is in the better cluster. Prediction results are shown in Table 4, 5, 6 and 7. From these tables, we can draw the following points.

Firstly, whether univariable or multivariable input for models, the performance of HTM without clustering is almost the worst in Table 4 and 6. However, it is worth emphasizing that from Table 5 and 7, learning similar patterns from the better cluster obtained from KMSD clustering can only improve the prediction accuracy of HTM. There are several possible reasons to explain this phenomenon: 1) Because of high robustness and sensitivity of HTM, it is tend to learn

TABLE 5. Results of univariate input after clustering.

N-days	Prediction Models	Error Measures			
		MAE	R2	MSE	RMSE
1	C-RNN	4.3327	-1.2110	32.2627	5.6800
	C-LSTM	1.3354	0.5755	6.1943	2.4888
	C-GRU	2.0076	0.4931	7.3964	2.7196
	C-HTM	0.5125	0.9602	0.5799	0.7615
7	C-RNN	4.5489	-0.5708	23.1982	4.8164
	C-LSTM	3.1986	-0.2274	18.1269	4.2576
	C-GRU	2.5447	0.1224	12.9609	3.6001
	C-HTM	0.8072	0.8542	2.1224	1.4568
15	C-RNN	10.4123	-6.6293	114.6170	10.7059
	C-LSTM	2.5042	0.2467	11.3171	3.3641
	C-GRU	1.7592	0.5068	7.4088	2.7219
	C-HTM	1.4427	0.2215	11.3327	3.3664
30	C-RNN	14.2054	-13.7233	218.9290	14.7962
	C-LSTM	3.2494	-0.8710	27.8205	5.2745
	C-GRU	3.5780	-0.1984	17.8194	4.2213
	C-HTM	3.2959	-3.0833	59.4375	7.7096

TABLE 6. Results of multivariable input without clustering.

N-days	Prediction Models	Error Measures			
		MAE	R2	MSE	RMSE
1	RNN	1.6856	0.7053	4.3007	2.0738
	LSTM	1.2931	0.7503	3.6439	1.9089
	GRU	1.9505	0.6218	5.5184	2.3491
	HTM	2.0546	-0.0024	14.5910	3.8198
7	RNN	1.8255	0.6546	5.1012	2.2586
	LSTM	0.6835	0.8609	2.0548	1.4334
	GRU	0.5201	0.8736	1.8672	1.3665
	HTM	1.9043	0.2381	11.0906	3.3303
15	RNN	1.3434	0.7614	3.5849	1.8934
	LSTM	1.4569	0.7126	4.3174	2.0778
	GRU	0.9108	0.8348	2.4813	1.5752
	HTM	2.3522	-0.3880	20.2040	4.4949
30	RNN	0.7031	0.8699	1.9349	1.3910
	LSTM	0.8642	0.8288	2.5457	1.5955
	GRU	0.5696	0.8700	1.9337	1.3906
	HTM	2.8981	-0.5789	22.9821	4.7940

TABLE 7. Results of multivariable input after clustering.

N-days	Prediction Models	Error Measures			
		MAE	R2	MSE	RMSE
1	C-RNN	14.4677	-14.3097	223.4028	14.9467
	C-LSTM	8.6086	-5.9725	101.7438	10.0868
	C-GRU	8.1229	-6.1027	103.6442	10.1806
	C-HTM	0.6498	0.8582	2.0647	1.4369
7	C-RNN	1.6790	0.6013	5.8883	2.4266
	C-LSTM	7.8955	-4.2013	76.8153	8.7644
	C-GRU	21.4680	-34.1692	519.3969	22.7903
	C-HTM	1.1661	0.6230	5.4876	2.3426
15	C-RNN	6.2855	-2.6135	54.2869	7.3680
	C-LSTM	8.7908	-6.0445	105.8320	10.2875
	C-GRU	15.2214	-23.8316	373.0538	19.3146
	C-HTM	2.0019	-0.1084	16.1340	4.0167
30	C-RNN	9.2203	-5.4112	95.3324	9.7638
	C-LSTM	25.7792	-83.8319	1261.4146	35.5164
	C-GRU	3.5341	-0.4184	21.0908	4.5925
	C-HTM	2.0857	-0.1602	16.8880	4.1095

patterns that are useful for prediction in similar stocks. 2) The baseline models without parameter tuning is underfitting or overfitting. In other words, as a online learning model, there are no hyperparameters to adjust for HTM. Such features can be regarded as the advantages of HTM in stock prediction task.

Secondly, from Table 4 and 6 we can find that multivariable input is vaild for LSTM and GRU obviously. However, the effectiveness on RNN and LSTM is little or even counterproductive. And in Table 5 and 7, it is only effective for RNN. This demonstration shows that RNN, LSTM, and GRU are better at accepting multiple inputs. From another perspective, it means that HTM may not need other variables as input to achieve satisfactory results.

Finally, comparing Table 4, 5, 6 and 7, the result of HTM prediction after clustering in Table 5 with 1-days ahead have the best score, which is shown in Figure 6.

In general, the experimental results show that 1) KMSD can largely improve the prediction accuracy of HTM, and 2) HTM is suitable for short-term stock price prediction. Moreover, it is not necessary to focus on feature selection because multivariate input has little benefit for HTM.

V. CONCLUSION AND FUTURE RESEARCH

Predicting stock prices based on historical data is considered an attractive and challenging task. In this paper, we first employed KMSD clustering to find similar stocks. Moreover, HTM is applied to learn from the similar stock dataset. The experiment results show that KMSD clustering can improve the accuracy of HTM significantly and HTM after clustering is suitable for show-term stock price prediction. In the future, we plan to further improve the prediction accuracy and generalization of the model by developing algorithms that can consider more variables to find similar stocks.

ACKNOWLEDGMENT

(Xingqi Wang and Kai Yang are co-first authors.)

REFERENCES

[1] E. F. Fama, "Efficient capital markets: A review of theory and empirical work," *J. Finance*, vol. 25, no. 2, pp. 383–417, May 1970.

[2] A. W. Lo and A. C. MacKinlay, *A Non-Random Walk Down Wall Street*. Princeton, NJ, USA: Princeton Univ. Press, 2011.

[3] H. Fromlet, "Behavioral finance-theory and practical application: Systematic analysis of departures from the homo oeconomicus paradigm are essential for realistic financial research and analysis," *Bus. Econ.*, vol. 36, pp. 63–69, Jul. 2001.

[4] A. Thakkar and K. Chaudhari, "Fusion in stock market prediction: A decade survey on the necessity, recent developments, and potential future directions," *Inf. Fusion*, vol. 65, pp. 95–107, Jan. 2021.

[5] F. G. D. C. Ferreira, A. H. Gandomi, and R. T. N. Cardoso, "Artificial intelligence applied to stock market trading: A review," *IEEE Access*, vol. 9, pp. 30898–30917, 2021.

[6] Z. Hu, Y. Zhao, and M. Khushi, "A survey of forex and stock price prediction using deep learning," *Appl. Syst. Innov.*, vol. 4, no. 1, p. 9, Feb. 2021.

[7] D. Shah, H. Isah, and F. Zulkernine, "Stock market analysis: A review and taxonomy of prediction techniques," *Int. J. Financial Stud.*, vol. 7, no. 2, p. 26, May 2019.

[8] M. Nabipour, P. Nayyeri, H. Jabani, A. Mosavi, and E. Salwana, "Deep learning for stock market prediction," *Entropy*, vol. 22, no. 8, p. 840, Jul. 2020.

[9] S. Chen and C. Zhou, "Stock prediction based on genetic algorithm feature selection and long short-term memory neural network," *IEEE Access*, vol. 9, pp. 9066–9072, 2021.

[10] Z. Li and J. Yuan, "An estimation similarity measure method based on the characteristic of vector difference," *Inf. Int. Interdiscipl. J.*, vol. 14, no. 3, pp. 1067–1074, 2011.

[11] J. Hawkins and S. Blakeslee, *On Intelligence*. New York, NY, USA: Macmillan, 2004.

[12] Y. Xu, C. Yang, S. Peng, and Y. Nojima, ''A hybrid two-stage financial stock forecasting algorithm based on clustering and ensemble learning,'' *Int. J. Speech Technol.*, vol. 50, no. 11, pp. 3852–3867, Nov. 2020.
[13] X. Li and P. Wu, ''Stock price prediction incorporating market style clustering,'' *Cognit. Comput.*, vol. 13, pp. 1–18, Jan. 2021.
[14] K. Nakagawa, M. Imamura, and K. Yoshida, ''Stock price prediction using K-medoids clustering with indexing dynamic time warping,'' *Electron. Commun. Jpn.*, vol. 102, no. 2, pp. 3–8, Feb. 2019.
[15] S. Kumari, N. Patil, P. Nankar, and M. Kulkarni, ''CUDA parallel computing framework for stock market prediction using K-means clustering,'' in *Proc. Int. Conf. Smart Electron. Commun. (ICOSEC)*, Sep. 2020, pp. 467–473.
[16] E. Hoseinzade and S. Haratizadeh, ''CNNpred: CNN-based stock market prediction using a diverse set of variables,'' *Expert Syst. Appl.*, vol. 129, pp. 273–285, Sep. 2019.
[17] S. K. Chandar, ''Grey wolf optimization-elman neural network model for stock price prediction,'' *Soft Comput.*, vol. 25, no. 1, pp. 649–658, Jan. 2021.
[18] X. Pang, Y. Zhou, P. Wang, W. Lin, and V. Chang, ''An innovative neural network approach for stock market prediction,'' *J. Supercomput.*, vol. 76, no. 3, pp. 2098–2118, Mar. 2020.
[19] J. MacQueen, ''Some methods for classification and analysis of multivariate observations,'' in *Proc. 5th Berkeley Symp. Math. Statist. Probab.*, vol. 1, 1967, pp. 281–297.
[20] J. Hawkins, S. Ahmad, S. Purdy, and A. Lavin. (2016). *Biological and Machine Intelligence (BAMI)*. Initial Online Release 0.4. [Online]. Available: https://numenta.com/resources/biological-and-machine-intelligence/
[21] Y. Dauletkhanuly, O. Krestinskaya, and A. P. James, ''Htm theory,'' in *Deep Learning Classifiers With Memristive Networks* (Modeling and Optimization in Science and Technologies), vol. 14. Springer, 2020, pp. 169–180, doi: 10.1007/978-3-030-14524-8_13.
[22] S. Purdy, ''Encoding data for HTM systems,'' 2016, *arXiv:1602.05925*. [Online]. Available: http://arxiv.org/abs/1602.05925
[23] Y. Cui, S. Ahmad, and J. Hawkins, ''The HTM spatial pooler—A neocortical algorithm for online sparse distributed coding,'' *Frontiers Comput. Neurosci.*, vol. 11, p. 111, Nov. 2017.
[24] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, ''Dropout: A simple way to prevent neural networks from overfitting,'' *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
[25] D. Molchanov, A. Ashukha, and D. Vetrov, ''Variational dropout sparsifies deep neural networks,'' in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2498–2507.
[26] S. Ahmad and L. Scheinkman, ''How can we be so dense? The benefits of using highly sparse representations,'' 2019, *arXiv:1903.11257*. [Online]. Available: http://arxiv.org/abs/1903.11257
[27] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, ''Learning representations by back-propagating errors,'' *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986.
[28] S. Hochreiter and J. Schmidhuber, ''Long short-term memory,'' *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[29] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, ''Learning phrase representations using RNN encoder-decoder for statistical machine translation,'' 2014, *arXiv:1406.1078*. [Online]. Available: http://arxiv.org/abs/1406.1078
[30] S. Hochreiter, ''Untersuchungen zu dynamischen neuronalen netzen,'' Diploma, Technische Universität München, Munich, Germany, Tech. Rep., Apr. 1991, vol. 91, no. 1.
[31] P. J. Rousseeuw, ''Silhouettes: A graphical aid to the interpretation and validation of cluster analysis,'' *J. Comput. Appl. Math.*, vol. 20, pp. 53–65, Nov. 1987.
[32] Y. Cui, S. Ahmad, and J. Hawkins, ''Continuous online sequence learning with an unsupervised neural network model,'' *Neural Comput.*, vol. 28, no. 11, pp. 2474–2504, Nov. 2016.

XINGQI WANG was born in 1999. He is currently pursuing the bachelor's degree with the College of Science and Information, Qingdao Agricultural University, China. His research interest includes machine learning.

KAI YANG was born in 1999. He is currently pursuing the bachelor's degree with the College of Science and Information, Qingdao Agricultural University, China. His research interests include time series prediction and deep learning.

TAILIAN LIU was born in 1983. He received the Ph.D. degree from the Institute of Modern Physics, University of Chinese Academy of Sciences, in 2012. He is currently a Lecturer with the College of Science and Information, Qingdao Agricultural University, China. His research interests include communication protocol and the Internet of Things.