Stock Market Trend Prediction using Supervised Learning

Asad Masood Khattak[†]
College of Technological Innovation
Zayed University, UAE
asad.khattak@zu.ac.ae

Habib Ullah
Institute of Computing and Information Technology (ICIT), Gomal University, D.I.Khan, Pakistan
habibkhan12233@gmail.com

Hassan Ali Khalid
Institute of Computing and Information Technology (ICIT), Gomal University, D.I.Khan, Pakistan
cliccme@gmail.com

Ammara Habib
Institute of Computing and Information Technology (ICIT), Gomal University, D.I.Khan, Pakistan
ammarahabib10@gmail.com

Muhammad Zubair Asghar
Institute of Computing and Information Technology (ICIT), Gomal University, D.I.Khan, Pakistan
zubair@gu.edu.pk

Fazal Masud Kundi
Institute of Computing and Information Technology (ICIT), Gomal University, D.I.Khan, Pakistan
fmkundi@gmail.com

ABSTRACT

The stock trend prediction has received considerable attention of researchers in recent times. It is an important application in machine learning domain. In this work, we propose a machine learning based stock trend prediction system with a focus on minimizing data sparseness in the acquired datasets. We perform outlier detection on the acquired dataset for dimensionality reduction and employ K-nearest neighbor classifier for predicting stock trend. Results obtained show the effectiveness of the proposed system, when compared with baseline studies.

CCS CONCEPTS

• **Information systems → Data mining**; Nearest-neighbor search
• **Computing methodologies → Machine learning**; Supervised learning

KEYWORDS

Trend Prediction, Machine Learning, Supervised Learning

1 INTRODUCTION

The stock exchange plays a central part in a country's economic strength. The financial state of a country is considered stable, if their stock exchange is optimistic [1]. Forecasting of stock market trends has remained an active area of research in computational sciences due to its possible monetary profit [2]. Because of the non-stationery and noisy nature of stock data, the prediction of trend in the financial exchange is viewed as a challenging task [3].

1.1 Problem statement

The stock exchanges play a pivotal role for measuring and monetary the financial development of a nation. The stock trend prediction is an important application of artificial intelligence. The earlier studies [2-5] on stock trend prediction have shown poor performance due to data sparseness in the acquired datasets, applied on different machine learning classifiers. Therefore, it is required to minimize data sparsity in the stock-related dataset for efficient stock trend prediction. To address this problem, a trend prediction framework for stock market, with emphasis on minimizing data sparsity, is proposed, to effectively predict the stock trend.

1.2 Objectives

In this work, we intend to achieve following objectives: (i) Applying outlier detection to reduce data sparsity in the dataset for efficient prediction of stock trend using machine learning technique, namely K-nearest neighbour, (ii) To evaluate performance of proposed model with respect to state-of-the-art method [2].

To achieve our specific objectives, we need to develop an enhanced prediction model that uses existing stock market data to forecast the market trend by minimizing data sparseness.

Our work is significant along following dimensions. (i) predicting stock trend by applying K-nearest neighbor machine learning classifier on different stock-related datasets; (ii) data sparsity in the acquired dataset will be reduced by applying outlier detection for efficient stock trend prediction; (iii) comparing performance of the proposed system with respect to other machine learning classifiers; and the base-line studies to assess the efficacy of the proposed model.

SoICT '19, December 4–6, 2019, Hanoi - Ha Long Bay, Viet Nam
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-7245-9/19/12...$15.00
https://doi.org/10.1145/3368926.3369680

2 REVIEW OF WORK

This section presents a review of stock-trend prediction related works. A review of related studies are as follows;

In recent years, with the advancement of ***machine learning*** *techniques* the researchers applied different ML classifiers for stock-trend prediction. For instance, Ghazanfar et al. [2] proposed techniques for machine learning to forecast the future stock market. Various machine-learning algorithms, such as SVM, KNN, neural networks, Bayesian network, and Ada boosts, are applied to predict stock trade volume. The result obtained are promising, yielding low error (MAE =0.0904 and RMSE =0.2402). The limitation of study includes: small size of dataset and sparseness. However, the work can be extended to achieve results that are more efficient by conducting experiment on larger dataset. Another baseline work, proposed by Khan et al. [3] exploits different machine learning classifiers, such as Support Vector Machine (SVM), (KNN) and Naïve Bayes to forecast stock trend. Among others, KNN gives the better prediction accuracy with lowest calculated (MAE =0.15 and RMSE =0.25). The limitation of this study is that only stock historical data is taken into account instead of social media data. The work can be enhanced by the use of more loss function and other classifiers. To determine movement in stock market on day to day basis, Khedr et al. [6] applied Naïve Bayes and K-NN classifier, the prediction accuracy was 89.80%. The limitation of study includes not using the textual financial information. The work can be enhanced by including some technical analysis indicators. Similar to Khedr et al.'s work, Ritesh et al. [7] proposed a model that determines movement in stock market on daily basis. Data is extracted using news article and trained using SVM, receiving 75% accuracy. However, sophisticated sentiment analysis techniques can be applied for further enhancement.

In today's data science world, ***Deep learning*** is an emerging technique in many fields which achieved remarkable results. Zhang et al. [8] suggested a replica to analyze the impact of internet and social media on financial market. Dataset was acquired from the network news through the web crawler. LSTM model is used to train the data. The prediction results are not very much satisfactory, ranging 40.3% to 50.6%. The limitation of study is that the time series process is not fine-tuned and it is required to improve emotion extraction algorithm by incorporating more useful feature to train the model. Another study conducted by Akita et al. [9] applied different deep learning techniques like paragraph vector and long short-Term memory (LSTM). The result obtained from various experiment are promising. The limitation of study is that only news titles are considered. More technical metrics can be used for further enhancement, such as Moving Average (MA) and Moving Average Convergence Divergence (MACD).

Some of the ***other*** techniques have also been applied for stock market prediction, such as Pagolu et al. [10] suggested a stock forecasting paradigm, based on public opinion expressed on Twitter. The data were gathered using Word2vec and N-gram features of the Twitter API to analyze public feelings in Tweets. The result obtained are satisfactory with an accuracy of 69.10%. Limited size of the dataset can affect prediction results, which need further extension by adding stock wits, news and taking large number of tweets for further enhancement. Li et al. [11] used 200 million tweets extracted from the Twitter and correlated it with DJIA SMeDA-SA models. The system received a 70% prediction accuracy. Long textual data and other outlets, such as Facebook, can be used to improve prediction accuracy. Dong et al. [12] acquired the dataset from the Weibo website (China) and applied through the time series dataset to a hybrid Back Propagation Neural network (BPNN) and, receiving a prediction precision of 92.99%. The limitation of the work includes using a time series dataset rather than the features of online public opinions. However, for obtaining more efficient results, improvement in the model is required.

3 METHODOLOGY

This section deals with description of proposed methodology for stock trend prediction using supervised machine learning technique, namely K-Nearest Neighbor (KNN) classifier (Figure. 1).

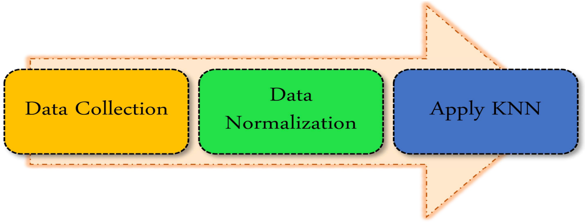

Figure 1: Proposed System

3.1 Data Collection

By using the past day's stock transaction, the future stock trend can be predicted. So, required stock data is collected from financial website of Karachi Stock Exchange over the duration of 8 years (Table 1). Each stock is represented by a stock symbol like KSE Stand for Karachi stock exchange having the different attributes like stock opening, stock closing, high, low and volume.

Table 1: Summary of dataset

Dataset Title	Description	URL
KSE 100	This dataset contains Karachi stock exchange 100 company stock indices	www.ksestocks.com/QuotationData

The detail of KSE 100 stock quotes is shown in Table 2. The first "symbol" column denotes the specific company or stock equity, such as KSE stand for Karachi stock exchange. "Open", "High" and "Low" columns indicate the starting value, peak value and lowest value of an equity, and close represents the final worth of an equity at the time of closing stock market. Similarly, "Volume" column shows the number of stock traded on a specific day.

Table 2: Sample listening of the acquired dataset

Symbol	Open	High	Low	Close	Volume
KSE-100	9411.75	9447.29	9375.89	9437.85	94598763
KSE-100	9441.18	9664.32	9441.18	9657.38	1.93E+08
KSE-100	9688.53	9810.19	9688.53	9727.36	2.28E+08
KSE-100	9746.42	9812.97	9712.16	9737.47	2.36E+08
KSE-100	9752.95	9823.64	9736.6	9776.21	2.27E+08
KSE-100	9814.47	9865.85	9783.06	9797	1.78E+08
KSE-100	9822.38	9833.26	9751.06	9778.36	1.12E+08

As the dataset is based on daily stock prices, so it is time series dataset for stock prediction.

The dataset used by the proposed work is a benchmark dataset used by Asghar et al. (2019) [1] for stock market trend predictions using machine learning technique. So, in the proposed work, we built a machine learning-based KNN approach, applied on the benchmark dataset and achieved the promising results with respect to the baseline studies.

3.2 Data normalization

This stage involves different data derivation steps applied on the different attributes of the dataset for efficient processing by the prediction model [1], because sparsity may ultimately degrade the prediction result. To remove the data sparseness, and to obtain a quality prediction result, we normalize the dataset to detect the outliers by applying normal distribution. The data normalization phase is comprised of following sub-tasks (i) computation of stock return, (ii) computation of z-score, (iii) applying Z-score for outlier detection, and (iv) implementation of outlier removal using pandas code. Detail of these steps is mentioned as under.

3.2.1 Computation of stock return. Calculating returns on stocks is a necessary step to define returns on stocks or income. The return on stock is essentially the distinction between the present stock price and the prior stock price. It can be calculated by subtracting the preclosure costs of the stock from the present closing days and dividing them by the closing day before.

$$\begin{aligned} &stock\_return \\ &= (current\_day\_closing\_price \\ &- \ closing\_price\_of\_previous\_day) \\ &/previous\_day\_closing\_price\_of\_the\_stock \end{aligned} \quad (1)$$

A sample computation of stock return is demonstrated using Eq. 1 as follows:

$$stock\_return = (321.67 \ - \ 310.6)/310.6 \ \ = 0.035641$$

3.2.2 Computation of z score. There are different techniques for performing outlier detection, such as Numeric outlier, DBSCAN, Isolation forest, and Z-score. Inspired from the early work conducted by [1] on outlier detection, we apply Z-score technique for outlier detection. The Z-score is computed as follows:

$$Z - score = Open - \frac{Mean\ of\ Open}{Stdv\ of\ Open} \quad (2)$$

For example, if the stock's open value is 9441.18 and its mean value is 27257.94, and its standard deviation is 12927.85, then we apply Eq. 2 and calculate the following Z-score.

$$Z - score = 9441.18 - \frac{27257.94}{12927.85} = 9439.071$$

3.2.3 Applying Z-score for outlier detection. Any value that is significantly lower or higher than the other values in given a set of numerical data. So, let's say we have the following figures: 1,2, 3,19. 19 It's our outlier. Why? Because, it's obvious. 19 The other figures are distinct from each other. An outlier is an element in a collection of information that stands out clearly from the majority of the information. An outlier is the information set's largest or lowest value. This may alter the outcome, resulting in incorrect calculation. For example, to compute students' average age in a given set of values {22, 26, 27, 29, 31, 32, 47, 33}, 47 is an outlier, because it is out of range in the given set of values. Due to this outlier, we can't get the exact value of average age.

Therefore, removal of such outliers is essential for getting the exact outcome. It is computed as follows:

$$\begin{aligned} &\boldsymbol{Outlier} \\ &= \boldsymbol{IF}(\boldsymbol{OR}(\boldsymbol{M2} > \mathbf{1}, \boldsymbol{M2} < -\mathbf{1}), \text{"Outlier"}, \boldsymbol{C2}) \end{aligned} \quad (3)$$

Where M is Z-score and C is that specific column of which the outlier detected. For example, if M2 = -1.370433599 and C2 = 9441.18, we compute the outlier as follows:

$$\begin{aligned} \boldsymbol{Outlier} = \boldsymbol{IF}(\boldsymbol{OR}(-\mathbf{1.370433599} > \mathbf{1}, -\mathbf{1.370433599} \\ < -\mathbf{1}), \text{Outlier}, \mathbf{9441.18}) \end{aligned}$$

3.2.4 Implementation of outlier removal using pandas code. To eliminate the detected outlier from the dataset, we perform implementation through panda's code. The computation steps of z-score for outlier detection and removal, are listed in the Table 3.

Table 3: Outlier detection and removal steps

(1) Calculating Z-score

$Z = V - X^{-}/S$

Where V represent the current stock value.

X^{-} and S represent Mean and Standard deviation respectively.

(2) Deleting Outlier

(3) Removing outliers from the dataset

3.3 Applying K-Nearest Neighbours Classifier

In this work, we employ a K-Nearest Neighbor classifier to effectively predict stock trend, as the said model is best suited to evaluate financial terms and less prone to data overfitting [1]. A non-parametric technique K-Nearest Neighbor algorithm (K-NN) is used for categorization and relapse. The information includes the K-Nearest preparing precedents in the component space in both cases. If the value of k = 1, then object is allocated to the adjacent class. We can make predictions based on the KNN examples in the wake of choosing the estimation of k. For regression, KNN forecasting is the average of the k-nearest outcome of the neighbor. It is computed as follows.

$$y = \frac{1}{K}\sum_{i=1}^{k} y_i \qquad (4)$$

In the above Eq. 4, yi is the *i*th case and *y* is the query point prediction (outcome). Unlike regression, KNN forecasts depend on a casting a ballot conspires in which the champ is utilized to mark the inquiry in arrangement issues. To avoid ties, odd value of y=1, 3, 5 are used for binary classification tasks i.e. two class names accomplish a similar score. K is the number of training data points in the area of the test information point that we will use to predict the class. After choosing the value of K, we can make prediction based on the KNN. For regression, KNN prediction is the average of the k-nearest neighbor outcome. In this case, Stock return is the target variable and open, high, low, close and volume are dependent variables. In Eq. 4, y is the outcome (RMSE), K is the distance that we select and yi is the ith case of the example and i is the iteration #, performed. To compute the accuracy of continuous variables, two most common metrices are used namely Mean Absolute Error (MAE) and Root mean squared error (RMSE).

Mean Absolute Error (MAE): It is the average over the test instances of the absolute differences among actual and prediction observation where every individual difference has equivalent weight. It is formulated as follows in Eq. 5;

$$MAE = {}^{1}/_{i}\sum_{k=1}^{i}|x_k - \hat{x}_k| \qquad (5)$$

Root mean squared error (RMSE): RMSE computes the average magnitude of the error and it is also the quadric scoring rule. The square root of the average of squared differences among the prediction and actual observation is defined as root mean squared error. It is formulated as follows in Eq 6;

$$RMSE = \sqrt{{}^{1}/_{i}\sum_{k=1}^{i}(x_k - \hat{x}_k)^2} \qquad (6)$$

4 RESULTS AND DISCUSSION

To address the first objective, we applied K-Nearest neighbor classifier to predict stock trend (see Section 3). The parameter setting of the propose KNN model for stock trend prediction is presented in Table 4.

Table 4: Parameter setting for KNN

Parameter	Description
Algorithm = 'auto'	Will attempt to decide the most appropriate algorithm based on the values passed to fit method.
Leaf size = 30 (default = 30)	This can affect the speed of the construction and query, as well as the memory required to store the tree. The optimal value depends on the nature of the problem.
Metric = 'minkowski'	The distance metric to be used for the tree. The default metric is minkowski.
Metric prams = None	Additional keyword arguments for the metric function.
n-Jobs = 1	The number of parallel jobs to run for neighbors' search.
n-neighbors = n	Number of neighbors to be used by default for kneighbors queries.
p = 2	For arbitrary p, minkowski_distance (l_p) is used.
weight = 'uniform'	All points in each neighborhood are weighted equally.

We conducted different experiments to evaluate the prediction efficiency of the proposed model on un-normalized and normalized data.

Proposed work experimental environment: The experimental environment used by the proposed work is Anaconda-based jupyter notebook[1] with python language.

4.1 Experiment #1. Prediction on unnormalized (Raw) dataset.

We applied KNN Regressor on the raw (un-normalized) dataset. A partial listening of the dataset is presented in Table 5.

[1] https://www.anaconda.com/

Table 5: A partial listening of raw dataset

Symbol	Open	High	Low	Close	Volume
KSE-100	9411.75	9447.29	9375.89	9437.85	94598763
KSE-100	9441.18	9664.32	9441.18	9657.38	1.93E+08
KSE-100	9688.53	9810.19	9688.53	9727.36	2.28E+08
KSE-100	9746.42	9812.97	9712.16	9737.47	2.36E+08
KSE-100	9752.95	9823.64	9736.6	9776.21	2.27E+08
KSE-100	9814.47	9865.85	9783.06	9797	1.78E+08
KSE-100	9822.38	9833.26	9751.06	9778.36	1.12E+08
KSE-100	9798.77	9875	9781.08	9784.85	1.27E+08
KSE-100	9814.69	9820.05	9742	9802.45	1.63E+08
KSE-100	9860.04	9968.55	9860.04	9923.14	2.05E+08

In Table 6, we present a partial listening of prediction results obtained by applying KNN regressor on the raw dataset. The first column shows the iteration no. while the second, third and fourth column show the prediction results, quantified using RMSE, MAE, and Logcosh.

Table 6: Prediction results on raw dataset

Iteration	RMSE	MAE	Logcosh
5	202955735.8	11976.81202	Inf
11	188881104.5	11746.46341	Inf
17	180075597.2	11570.63793	Inf
23	174864623.6	11430.98562	Inf
29	173025251.2	11378.50613	Inf
35	171312709.5	11335.03967	Inf
41	169612906.4	11275.85507	Inf

The result presented in Table 4 show that the value of RMSE, MAE and logcosh are high enough, resulting in poor prediction [5].

4.2 Experiment#2. To Perform prediction on normalized dataset.

We applied KNN Regressor on normalized dataset (see Section 3). A partial listening of such dataset is presented in Table 7. The data normalization process is already mentioned in Section 3.

Table 7: A partial listening of prediction results on normalized dataset

Symbol	Open	High	Low	Close	Return
KSE-100	9411.75	9447.29	9375.89	9437.85	
KSE-100	9441.18	9664.32	9441.18	9657.38	0.023261
KSE-100	9688.53	9810.19	9688.53	9727.36	0.007246
KSE-100	9746.42	9812.97	9712.16	9737.47	0.001039
KSE-100	9752.95	9823.64	9736.6	9776.21	0.003978
KSE-100	9814.47	9865.85	9783.06	9797	0.002127
KSE-100	9822.38	9833.26	9751.06	9778.36	-0.0019
KSE-100	9798.77	9875	9781.08	9784.85	0.000664
KSE-100	9814.69	9820.05	9742	9802.45	0.001799
KSE-100	9860.04	9968.55	9860.04	9923.14	0.012312

In Table 8, we present a partial listening of prediction results obtained by applying KNN regressor on the normalized dataset. The first column shows the iteration no., while the second third and fourth column show the prediction results quantified using RMSE, MAE, and Logcosh.

Table 8: A sample listing of normalized dataset result

Iteration	RMSE	MAE	Logcosh
5	4.42E-05	0.004892914	0.003716503
11	6.07E-05	0.005722075	0.005097995
17	6.31E-05	0.005944986	0.005303328
23	6.64E-05	0.006160232	0.005573119
29	6.71E-05	0.006216439	0.005634309
35	6.94E-05	0.006315764	0.005827321
41	7.05E-05	0.006379043	0.005918586

The results presented in Table 6 show that the value of RMSE, MAE and logcosh are low enough, resulting in better prediction [5].

4.3 Experiment#3. Comparison with other classifiers and the state-of-the art methods.

We put dataset in our proposed model (KNN) train it and check the results. After training the proposed Model (KNN) and checking the result on each iteration (Table 9). On iteration 1 we get very low RMSE, MAE and Logcosh which are promising result as compared to the other iterations.

Table 9: Performance Evaluation of proposed Model (KNN) at different iterations

Iteration	RMSE	MAE	Logcosh
5	4.42E-05	0.004892914	0.003716503
11	6.07E-05	0.005722075	0.005097995
17	6.31E-05	0.005944986	0.005303328
23	6.64E-05	0.006160232	0.005573119
29	6.71E-05	0.006216439	0.005634309
35	6.94E-05	0.006315764	0.005827321
41	7.05E-05	0.006379043	0.005918586

4.3.1 Comparison with other classifiers. After normalizing the data, we conduct experiment on different classifiers and evaluate their prediction performance (RMSE, MAE, Logcosh). We split the dataset into 20% for test data and 80% for training data, then we apply the different classifiers listed in a Table 10.

Firstly, we perform experiment with Decision Tree classifier and check the results. Decision Tree yields low gives low RMSE, MAE and Logcosh, these are not up to the mark. In next experiment, we use the same dataset, which we used for decision tree, and apply the SVM Regressor and receive low values for RMSE and MAE but the results are not so promising. Likewise, we conduct experiment on ANN Model taking activation function (Logistic, Tanh) and Hidden layer size (60,70,80,90,10), the result obtained are better but not promising. Finally, we conduct experiment on KNN Model, and check its performance. KNN gives very low RMSE, MAE and Logcosh, when compared to SVM, ANN and Decision Tree.

Table 10: Comparison with other classifiers

Classifier	RMSE	MAE	Logcosh
Decision Tree	0.000064953	0.005623272736	0.005168238520
SVM	0.000074455	0.00657043367	0.006253967401
ANN	0.0000691	0.006358753	0.005806286
KNN(proposed)	0.000044246	0.004892914	0.003716503

4.3.2 Comparison with baseline methods. In this section, we compare the performance evaluation results of our proposed model with other baseline studies as shown in Table 11.

In Ghazanfar et al. [2] work, authors took the dataset of Karachi stock exchange (KSE-100 index) for a period of 6 months. On this dataset various algorithm are applied, such as SVM, KNN, Neural Network, Bayesian network, and Ada Boost to predict stock trade volume. The results get are good with low error (MSE = 0.2402 and MAE = 0.0904)

In Khan et al. [3] work, data is collected from Karachi stock exchange of 5 years. The said dataset is trained on various machine learning classifiers before and after applying principal component analysis (PCA) and calculated RMSE and MAE against each classifiers. They observed that KNN has lowest MAE (0.38), which is best classifier as compared to SVM, Naïve Bayes.

In our proposed model, we took historical dataset of 8 years from Karachi stock exchange. Our dataset is sparse in nature which affects the prediction result. To remove data sparseness from our dataset and to obtain a high prediction results, we normalize our dataset by detecting the outlier by using normal distribution technique. After normalization of data, we trained the data on various machine learning classifiers and calculated RMSE, MAE and Logcosh. From our experimental result, the proposed model gives much better results from the previous similar works conducted by Ghazanfar et al. [2] and Khan et al. [3]. Our proposed model(KNN) classifier gives much better result of very low error (MSE = 0.0000631, MAE = 0.05911 and Logcosh = 0.005300) as compared to previous works.

Table 11: Comparison of proposed system (KNN) with baseline methods

Study	Technique	Dataset	Result		
			MSE	MAE	Logcosh
Ghazanfar et al(2017)	Machine learning SVM KNN ANN	KSE	0.2402	0.0904	-
Khan et al (2016)	Machine learning KNN SVM Naïve bayes	KSE, London and New York exchange dataset	0.40	0.38	-
Proposed	Machine learning KNN SVM ANN DT	KSE	0.0000442	0.004892	0.003716

In Figure. 2, KNN Model graph shows the comparison of Root Mean square error, Mean Absolute error and logcosh. The y-axis shows the error and the X-axis shows the K-value, which starts from 5 with a difference of 5. A secondary y-axis is added to the graph to present all three errors in the same graph. As both errors values are not identical but the difference is very small and cannot be differentiated by the primary axis. At K-value=5, we get very low Root Mean square error=0.00004, Mean Absolute error=0.005 and logcosh=0.003, as compared to the other K-value. As the K-value increases, the Root Mean square also increases, and at k-value=40, the error we get, is high.

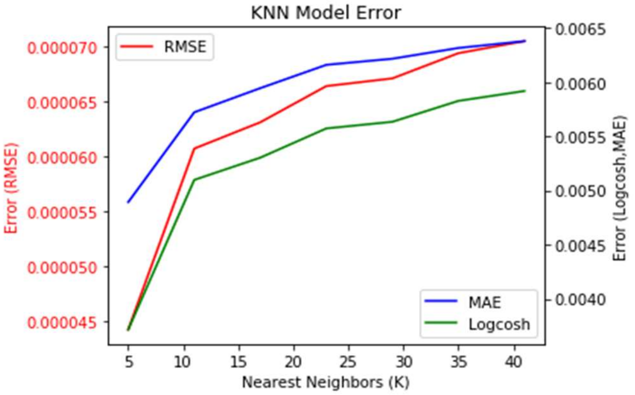

Figure 2: KNN regression graph

5 CONCLUSIONS

In this work, we presented a stock trend prediction system using supervised machine learning technique, namely K-nearest neighbor classifier. Our focus was on minimizing data sparseness in the acquired datasets by performing outlier detection on the acquired dataset for dimensionality reduction. Experimental results are encouraging, showing the effectiveness of the proposed technique. However, it is required perform experimentation with more machine learning classifiers and as future work, we recommend deep learning techniques, such as LSTM, RNN, and CNN on multiple datasets.

ACKNOWLEDGEMENT

This research work was supported by Zayed University Provost Research Fellowship award # R18114.

REFERENCES

[1] Asghar, M. Z., Rahman, F., Kundi, F. M., & Ahmad, S. (2019). Development of stock market trend prediction system using multiple regression. *Computational and Mathematical Organization Theory*, 1-31.

[2] Ghazanfar, M. A., Alahmari, S. A., Aldhafiri, Y. F., Mustaqeem, A., Maqsood, M., & Azam, M. A. (2017). Using machine learning classifiers to predict stock exchange index. International Journal of Machine Learning and Computing, 7(2), 24-29.

[3] Khan, W., Ghazanfar, M. A., Asam, M., Iqbal, A., Ahmad, S., & Khan, J. A. (2016). Predicting trend in stock market exchange using machine learning classifiers. *Science International,* 28(2).

[4] Joseph, A., Larrain, M., & Turner, C. (2017). Daily Stock Returns Characteristics and Forecastability. *Procedia computer science*, *114*, 481-490.

[5] Yetis, Y., Kaplan, H., & Jamshidi, M. (2014). Stock market prediction by using artificial neural network. IEEE, 718-722

[6] Khedr, A. E., & Yaseen, N. (2017). Predicting stock market behavior using data mining technique and news sentiment analysis. *International Journal of Intelligent Systems and Applications*, *9*(7), 22.

[7] Ritesh, B. R., R. Chethan, and Harsh S. Jani. (2017). Stock Movement Prediction Using Machine Learning on News Articles. International Journal on Computer science and Engineering, 9(8).

[8] Zhang, X., Shi, J., Wang, D., & Fang, B. (2018). Exploiting investors social network for stock prediction in China's market. *Journal of computational science*, *28*, 294-303.

[9] Akita, R., Yoshihara, A., Matsubara, T., & Uehara, K. (2016). Deep learning for stock prediction using numerical and textual information. In *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS),* 1- 6.

[10] Pagolu, V. S., Reddy, K. N., Panda, G., & Majhi, B. (2016, October). Sentiment analysis of Twitter data for predicting stock market movements. In *2016 international conference on signal processing, communication, power and embedded system (SCOPES)*,1345-1350.

[11] Li, B., Chan, K. C., Ou, C., & Ruifeng, S. (2017). Discovering public sentiment in social media for predicting stock movement of publicly listed companies. *Information Systems*, *69*, 81-92.

[12] Dong, X., Lian, Y., & Liu, Y. (2018). Small and multi-peak nonlinear time series forecasting using a hybrid back propagation neural network. *Information Sciences*, *424*, 39-54.