Investigating the Effect of Noise Elimination on LSTM Models for Financial Markets Prediction Using Kalman Filter and Wavelet Transform

AMIN KARIMI DASTGERDI, PAOLO MERCORELLI*
Institute of Product and Process Innovation, Leuphana University of Luneburg,
Luneburg, GERMANY

Abstract:- Predicting financial markets is of particular importance for investors who intend to make the most profit. Analysing reasonable and precise strategies for predicting financial markets has a long history. Deep learning techniques include analyses and predictions that can assist scientists in discovering unknown patterns of data. In this project, application of noise elimination techniques such as Wavelet transform and Kalman filter in combination of deep learning methods were discussed for predicting financial time series. The results show employing noise elimination techniques such as Wavelet transform and Kalman filter, have considerable effect on performance of LSTM neural network in extracting hidden patterns in the financial time series and can precisely predict future actions in these markets.

Key-Words: - Deep Learning, Time-series Forecasting, Financial Markets, LSTM, Wavelet Transform, Kalman Filter

Received: June 2, 2021. Revised: December 4, 2021. Accepted: January 17, 2022. Published: January 18, 2022.

1 Introduction

Among time series predictions, Stock market prediction is considered as one of the most challenging problems, due to its noisy and unstable features [1]. Since the early 70s, extensive efforts have been commenced to predict stock prices by using new mathematical methods, time series, and more advanced tools including artificial intelligence and many tests on price information and stock indexes in countries such as United States, United Kingdom, Canada, Germany, and Japan to show the presence or absence of a specific structure in stock price information [2]. To analyze this issue, the use of deep learning has attracted a great deal of attention in recent years and has been mentioned much in the communities related to artificial intelligence and machine learning. Deep learning implies the set of machine learning-based algorithms and methods that tries to discover complex patterns in data and model the high-level abstractions. Neural networks were first used in 1977 to predict exchange markets [3].

Between 1977 and 1995, a total of 213 academic activities in the field of artificial neural networks in the field of commerce. Of these, 54 were in the financial sector and two were in forecasting and analyzing other time series [4]. Most of the researches in this field were conducted to look for the ability of neural networks to identify non-linear patterns in time series and unknown price movements. Deep learning characteristics provided significant success for these types of approaches in machine learning.

Current prediction models suffer from deterministic and stochastic noise which exists in financial time series. Therefore, an appropriate algorithm which could eliminate the noise from financial time series without influencing the real values seems to be necessary to enhance the performance of these models. Wavelet transform and Kalman filter are two technique which are commonly used to eliminate the noise feature of financial time series. These techniques are considered for filtering and mining single-dimensional signals [5]. These techniques have been used in this project to eliminate the input financial time series and then feed them into the LSTM neural network.

Accordingly, this project report can be defined in several sections. The second section includes literature review and problem statement and its analysis to enhance the structure of neural networks. In the third section, the framework of the model which used in this project is discussed by showing its flowchart and describing different parts of it. The fourth section is dedicated to explanations about the project implementation and in the last part result of the project are discussed and analyzed.

2 Literature Review

The statistical methods based on analyzing the past market changes are widely accepted and employed in the methods that predict the financial markets. These approaches use different linear and non-linear methods to predict the market. Two views on linear and non-linear methods are presented [6], which in [7], the non-linear models have been considered to be better than linear models; however, linear models may work better than or as good as non-linear methods [8]. Prediction can be divided into three categories: short-term prediction, mid-term prediction, and long-term prediction, each of which works in different time ranges. In this regard, the available methods for predicting the stock price can be divided into three categories: fundamental analysis, technical analysis, time series analysis. These types of assessments can be described in three sections: fundamental, technical, time series.

In this respect, fundamental analysis is based on the assumption that securities (generally financial markets) possess intrinsic values that can be estimated by investors. Technical analysis in financial markets is a way of predicting the probable behavior of a chart through past data, such as price and its changes, trading volume, etc. In time series analysis, the time series can be defined as a regular sequence of observations for a selected variable in many prediction problems. In financial markets, the stock price is considered as the selected variable.

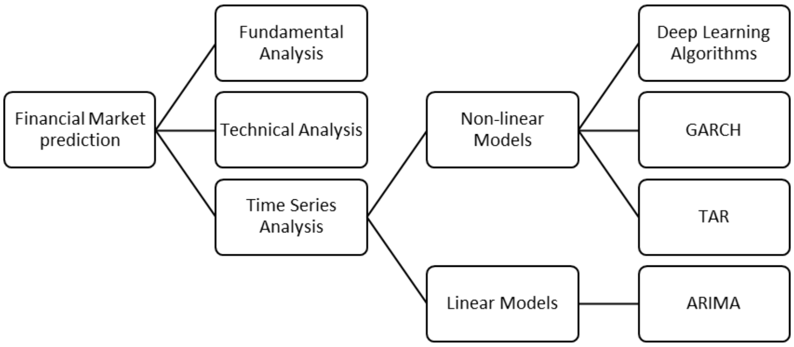

Fig. 1: Overview of variety of prediction methods for financial markets

So far, many researchers have tried to analyse non-linear time series and find a technique to separate noise of time series. One effective and powerful algorithm to separate the true noise component of original time series which has been proposed by Broomhead and King [9] is singular spectrum analysis (SSA). SSA tries to extract a series of singular values that contains the independent information of the original time series through singular spectral decomposition (SVD) and then analyse the features of the original time series over different time scales by reconstructing these singular values[10]. To explore the existing noise which lies in financial markets, Hill et. al [11] performed blockwise white noise tests. They realized that the white noise hypothesis can be accepted for financial markets of China and Japan which shows that those markets are weak form efficient while the white noise hypothesis can not be accepted for financial markets of Unites States and United Kingdom. Xiang Zhang [12] proposed a SVR-ARMA model for stock price prediction based on soft thereshold wavelet denoising which sym wavelet was selected as the wavelet basis function and three vanishing moments and three decomposition layers was considered. The maximum-minimum criterion has been employed to determine the threshold. Paisit Khanarsa et al. [13] proposed a self-identification ResNet-ARIMA order model to solve the issue of identifying ARIMA order and automatically learn the ARIMA order from known ARIMA time series data via sample autocorrelation function, the sample partial autocorrelation function and differencing time series images. In another study, Brogaard et al. [14] proposed a return variance decomposition model to separate the role of different types of information and noise in stock price movements. They came to the conclusion that 31% of the return variance is from noise. Xiaodan Liang et al. [15] proposed a new multioptimal combination wavelet transform (MOCWT) method which makes benefit of threshold-denoising function to reduce the degree of distortion in signal reconstruction to predict S&P 500 prices. Zargar et al. [16] proposed a model called “opening noise trading model” in which the opening price of the stock market was assumed to contain a component of noise which is orthogonal to the true price change caused by the arrival of new information among all the Nifty stocks. They have also estimated the share of noise in the opening price.

2.1 Problem Statement

In this research, Wavelet Transform and Kalman Filter were used to eliminate the noise from financial markets data which includes closing price, volume, technical indicators and other factors affecting the stock price. Also, a fully connected dense LSTM was used for prediction. The actions can be performed, considering the following objectives:

- Improving the prediction ability of future behaviors of financial markets using deep learning techniques.
- Investigating the effect of noise elimination on the accuracy of financial markets prediction.

3 Project Methology

As can be seen in Fig. 2. In this project, financial time series which were the indices of five different stock markets were used as inputs. The noise of these financial time series was eliminated using discrete wavelet transformation as well as Kalman filter algorithms. Then denoised financial time series were used as inputs of LSTM neural network. An LSTM neural network was trained based on training data and was used for prediction of the unseen data. Finally, the accuracy of the LSTM model was evaluated with comparison of predicted data and real values.

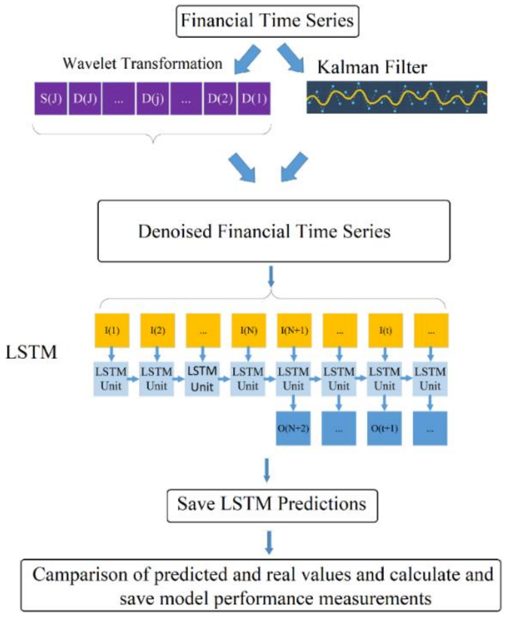

Fig. 2: The flowchart of the proposed model

3.1 Wavelet Transform

Wavelet transform is used in this project because of its ability to noise elimination of financial time series data. In this project, the Haar function considered as the basic wavelet function because this function in addition to breaking down financial time series into time and frequency ranges, has the advantage of significantly reducing processing time [5]. Time complexity of Wavelet transform with the Haar function as the basic function is O(n) where n is the size of the time series [17]. For continuous Wavelet transform, the wallet function is defined as follows:

$$\varphi_{a,\tau}(t) = \frac{1}{\sqrt{a}}\varphi\left(\frac{t-\tau}{a}\right) \tag{1}$$

Where a and τ are the scaling and translation factors, respectively. $\varphi$is basic Wavelet which follows a rule called the Wavelet Acceptance Condition [18]:

$$c_{\varphi} = \int_{0}^{\infty}\frac{\varphi(\omega)}{(\omega)}d(\omega) < \infty \tag{2}$$

Where φ (w) is a function of the frequency w as well as Fourier φtransient. x (t) denotes a complete square integral function. Continuous Wavelet Transform to wavelet is defined as follows:

$$CWT_x(a,\tau) = \frac{1}{\sqrt{a}}\int_{-\infty}^{+\infty} x(t)\varphi(\frac{t-\tau}{a})dt \tag{3}$$

Where $\overline{(\varphi(t))}$ represents conjugate function. the inverse of continuous wavelet transform is as follows:

$$X(t) = \frac{1}{c_{\varphi}}\int_{0}^{+\infty}\int_{-\infty}^{+\infty} CWT_x(a,\tau)\varphi_{a,\tau)}(t)dt \tag{4}$$

3.2 Kalman Filter

Kalman filter enables inference of unmeasured variables from indirect and noisy measurement [19]. As a result, it is arguably one of the most important discoveries in the field of mathematical engineering and has been used to solve various engineering problems in the area of monitoring and control of complex dynamic systems such as manufacturing processes, aircraft navigation, ships, and spacecrafts. While the Kalman filter is used to estimate of a state vector in a linear model in a dynamical system, the extended Kalman filter (EKF) is used in order to estimate a non-linear model [20].

The Kalman filter estimates the state of a dynamic system having certain types of random behavior [21]. The system must be described in a state space form:

$$\begin{aligned} X_{k+1} &= \varphi_k X_k + W_k \\ Z_k &= H_k + V_k \end{aligned} \tag{5}$$

x (n x 1) is called the state vector. It is composed of any set of variables sufficient to completely describe the unforced motion of a dynamic system. z (l x 1) is called the observation vector. It concerns data that can be known through measurements. w_k and v_k are the state and measurement white noise with known covariance matrices Q_k and R_k, respectively. They are mutually not correlated.   is the state transition matrix, H_k the observation matrix. The Kalman filter is based on a recursive algorithm [21]. At time t_k, the optimum combination of measured and estimated results is given by:

$$\hat{X}_k = \hat{X}_k^- + K_k.(Z_k - H_k.\hat{X}_k^-) \tag{6}$$

where   denotes the *a priori* estimate [21]. The "hat" denotes an estimate and the superscript minus indicates that this is the best estimate prior to assimilating the measurement at t_k. The Kalman filter gain K_k can be written as:

$$K_k = P_k.H_k^T.(H_k.P_k^-.H_k^T + R_k)^{-1} \tag{7}$$

where the error covariance matrix P_k associated with the optimal estimate is obtained from:

$$P_k = (I - K_k.H_k).P_k^- \tag{8}$$

To recursively compute the Kalman filter gain for the next step, the predictions for the state estimate and covariance at the next step are given by:

$$\hat{X}_{k+1}^- = \varphi_k.P_k.\varphi_k^T + Q_k \tag{9}$$

The derivation of the *extended* Kalman filter allows the estimation of a non-linear system state. To separate noise from signal, Kalman filter uses state vector. The state vector consists of all the features describing the models of the random processes. Additional state variables are appended to account for either non-white state or measurement noise.   and Q_k are straightforwardly deduced from the models. The observation vector is just the measured signal that is a mix of signal and noise. Therefore, the observation matrix is usually a combination of unit matrices and zeros matrices according to the state vector. The algorithm of the discrete Kalman filter will be used to estimate the signal, Once the system is properly designed [21].

3.3 LSTM

LSTM networks are a type of recurrent neural networks, with the advantage of having feedback links attached to some layers of the network which enables them to learn long-term dependencies. LSTM networks can easily remember information for long periods of without any learning struggle. They were introduced by Hochreiter & Schmidhuber [22], and were developed and refined by many researchers and have been applied to solve many problems such as robot control, time series prediction, speech recognition, rhythm learning, music composition, grammar learning, handwriting recognition, text-based language translation [23] and so on.

All recurrent neural networks have the form of a chain of repeating modules of neural network. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer. Unlike recurrent neural networks, LSTM is well-designed to learn from experience to predict time series when there are time steps with no certain size. In addition, thanks to having the memory unit, it can solve the problem of the vanishing gradient by having the memory unit keep the time related information for an uncertain time [24]. The change in the memory unit has made the LSTM network to be capable of remembering long-term dependencies. A memory cell consists of four components: input gate, target gate, forget gate, and one recurrent neuron. The gates control the interactions between the memory cell and its neighboring memory cells. The input gate determines whether an input signal can alter the status of a memory cell. On the other hand, the target gate determines whether the status of one memory cell can change the status of another memory cell. The target gate also selects to remember or forget the previous status [25].

Cell state is the most important part of LSTM. It is shown in Fig. 3. (The horizontal line running through the top). It runs straight down the whole chain, with only some minor linear interactions. Information can easily flow along it without being changed. The LSTM can remove or add information to the cell state using gates, which are carefully regulated structures. Gates are a way to optionally let information through. They are composed out of a sigmoid neural net layer and a pointwise multiplication operation. The sigmoid layer outputs numbers between zero and one, describing how much of each component should be let through. A value of zero means "let nothing through," while a value of one means "let everything through!".

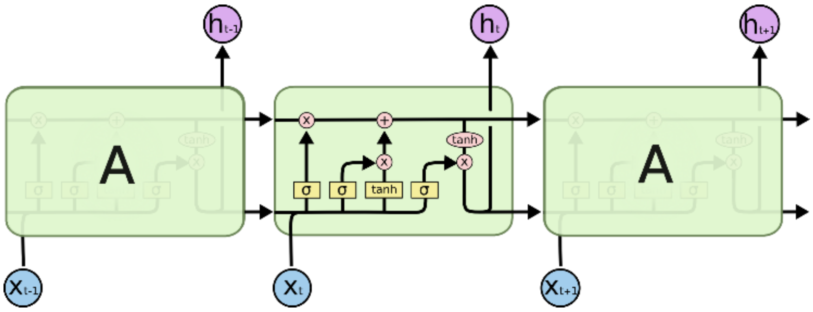

Fig. 3: Structure of a LSTM unit

The first step in LSTM is to decide what information is going to be thrown away from the cell state. This decision is made by a sigmoid layer called the “forget gate layer.” It looks at h and xt, and outputs a number between zero and one for each number in the cell state C. A one represents “completely keep this” while a zero represents “completely get rid of this.”

$$f_t = \sigma(W_f[h_{t-1}.x_t]+b_f) \quad (10)$$

The next step is to decide what new information should be stored in the cell state. This has two parts. First, a sigmoid layer called the “input gate layer” decides which values should be updated. Next, a tangent hyperbolic layer creates a vector of new candidate values, , that could be added to the state. In the next step, these two will be combined to create an update to the state [26].

$$i_t = \sigma(W_i[h_{t-1}.x_t] + b_i \quad (11)$$

$$\tilde{C}_t = \tanh(W_C[h_{t-1}.x_t] + b_C \quad (12)$$

Then the old cell state, C, will be updated into the new cell state . The old state will be multiplied by , forgetting the things we decided to forget earlier. Then $i_t *$ will be added. This is the new candidate values, scaled by how much it has been decided to update each state value.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (13)$$

Finally, the output will be based on the cell state, although a filtered version. First, a sigmoid layer will be run which decides what parts of the cell state is going to be the output. Then, in order to scale the values between -1 and 1 the cell state is put through tanh and then multiply it by the output of the sigmoid gate, so that only output the parts it has been decided.

$$o_t = \sigma(W_o[h_{t-1}.x_t] + b_o) \quad (14)$$

4 Implementation

All data which has been used in this project is from WIND database that developed by Shanghai Wind Observation Company as well as the CSMAR database that developed by GTA Technical and Education Organization Shen Jen and another financial database are located at (http://www.investing.com). data are from July 1, 2008 to September 30, 2016.

4.1 Software

In this project we propose to implement and evaluate the model based on Jupiter Notebook version 6.2.0 and Python programming language as well as Python interpreter version 3.8. All tests were implemented on a computer with 8-core processor at 2.21 GHz and 6 GB main memory by Windows operating system 10, 64-bits.

Some Important Python libraries which were used in this project are: numpy, pandas, sikit-learn, pywt, pykalman, matplotlib and plotly.

4.2 Data Description

The five indexes of the selected exchange include financial markets of China (CSI300), Hong Kong market (Hang Seng), Tokyo market (Nikkei 225), New York exchange markets (S&P500 and DJIA). As a rule, in financial markets around the world, the market situation may affect the credibility of the neural network; hence, the datasets of some markets with different conditions were used to overcome this issue. The data of the S&P500 and DJIA index are known as the most advanced financial markets in the world. On the other hand, CSI300 is considered as a rookie market and has been used as a developing market. In addition to the aforementioned markets, the Hang Seng index and the Nikkei 225 index were used as relatively-developed markets. Therefore, these markets provide a natural condition for evaluating the robustness and efficiency of models in markets with different conditions.

Three sets of variables were used as input. first set of variables which are considered as basic are opening price, highest price, lowest price, last price and volume of transactions. Second set of input variables is included twelve commonly used indicators are technical indicators for each indicator.

the last set of input variables is included macroeconomic variables. macroeconomic conditions with no doubt have a great impact on the performance of the stock markets between different regions [27]. So macroeconomic variables can transfer some information to the predictive neural network. In this project two types of macroeconomic indices of exchange rate and interest rate are used. Dollar index was used as a conversion rate. In addition, the interbank interest rate in each country was used as the interest rate. In table 1 definitions of all the variables which were used in this project are given.

Table 1. Definitions of input variables

Variable name	Definition
First set of input variables (basic information)	
Open/Close Price	**First price and last traded price of a share in day**
High/Low Price	**The highest and lowest quoted price in single day**
Trading Volume	**volume of shares traded in day**
Second set of input variables (technical indicators)	
MACD	**It means convergence and divergence of the moving average. It is used in technical analysis to gain ability, direction and acceleration in a process**
CCI	**The acceleration power of the oscillation in a share trend is well represented. It helps us find the beginning and the end of process.**
ATR	**It shows the fluctuations in the market well in the share**
BOLL	**Provides a relative definition of the highest and lowest prices that help us identify precise patterns.**
EMA 20	**Exponential moving average is a type of moving average that gives more weight to the latest data.**
MA5/MA10	**Moving Average 5 /10 days**
MTM6/ MTM12	**Monthly rate of movement. Helps us find the end or the downtrend or uptrend.**
ROC	**Price change rates. Shows the rate of change in the price of share.**
SMI	**Random motion index. Indicates the proximity of the end price to the midpoint in that range.**
WVAD	**Measures the amount of pressure to buy or sell in share.**
Third set of input variables (macroeconomic variables)	
Exchange Rate	US dollar index
Interest Rate	Interbank interest rates

4.3 Model Training

In this project, dataset is divided into three parts: training data, validation data and test data and the back propagation algorithm is used to train the LSTM network. learning rate, batch size and frequency of training set was set to 0.05, 32, and 100, respectively. Convergence rate is controlled by learning rate which is a descending function of time. When convergence is achieved but the combinations of parameters are varied It can be concluded that the experimental result are stable. In Fig. 4. Training loss and validation loss of LSTM model for different markets are depicted. As can be seen from different plots training loss and validation loss converge after 100 epochs which means the prediction error of the LSTM model is relatively small.

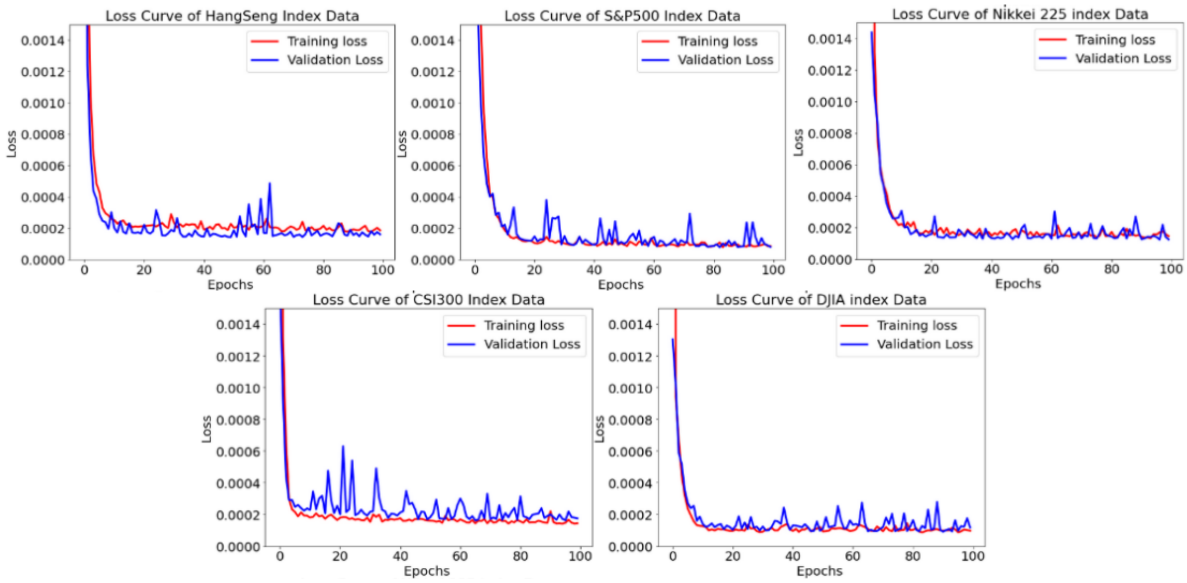

Fig. 4: Training loss and validation loss of LSTM model for different markets

4.4 Model Evaluation

4.4.1 MAPE

MAPE measures the accuracy of predicting a method. For instance, in the process of predicting trends, this measurement is used as an error function for regression problems in machine learning. MAPE indicates the accuracy in percentage and is obtained from the Eq. (10).

$$MAPE = \frac{\sum_{t=1}^{N}\left|\frac{y_t - y_t^*}{y_t}\right|}{N} \qquad (15)$$

4.4.2 Theil U

Theil U is a relative measurement of differences between two variables. Theil U reduces deviations to the power of two to give more weight and importance to more significant errors. Theil U is derived from the Eq. (11).

$$Theil\ U = \frac{\sqrt{\frac{1}{N}\sum_{t=1}^{N}(y_t - y_t^*)^2}}{\sqrt{\frac{1}{N}\sum_{t=1}^{N}(y_t)^2} + \sqrt{\frac{1}{N}\sum_{t=1}^{N}(y_t^*)^2}} \qquad (16)$$

4.4.3 RMSE

RMSE is the difference between the actual value and the value predicted by the model or statistical estimator. RMSE is obtained from the Eq. (12).

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_t - y_t^*)^2} \qquad (17)$$

4.4.4 R: in investing, R is generally interpreted as the percentage of a fund or security's movements that can be explained by movements in a benchmark index. For example, an R-squared for a fixed-income security versus a bond index identifies the security's proportion of price movement that is predictable based on a price movement of the index. R is derived from the Eq. (13).

$$R = \sqrt{\frac{\sum_{i=1}^{N}(y_t^* - \bar{y})^2}{\sum_{i=1}^{N}(y_t - \bar{y})^2}} \qquad (18)$$

In the equations above, y_t represents the actual value of data at time *t*, the variable y_t^* implies to the value predicted by the model at time *t*, and *n* represents the amount of data.

5 Results

In this section results are given. First Fig. 5. presents the plot depicting the performance of Wavelet transform and Kalman filter for S&P500 index market.

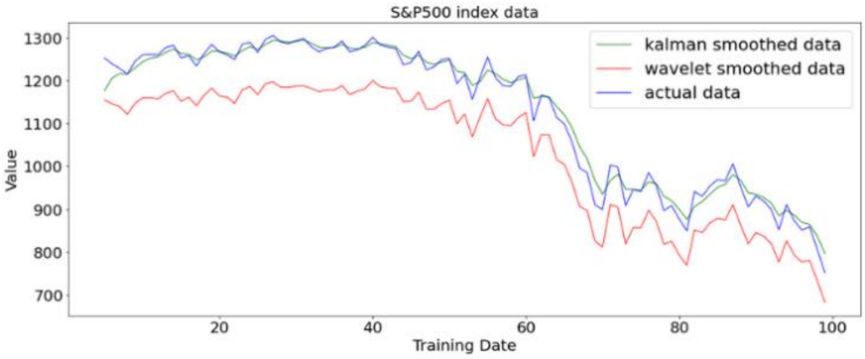

Fig. 5: Results of Wavelet transform and Kalman filter for S&P500 index

In order to evaluate each model, four key error metrics were considered which are root mean squared error (RMSE), mean absolute percentage error (MAPE), R and Theil U. Tables 2 to 5 contains performance of LSTM model which has used data that has been smoothed by both Wavelet transform and Kalman filter as well as raw data. “LSTM” refers to the model which used the raw data, “WLSTM” refers to the model which used the data that has been denoised by Wavelet transform and “KLSTM” refers to the model which used data that has been smoothed by Kalman filter.

Table 2. Predictive accuracy based on MAPE metric

	CSI	**Hang**	**Nikkei**	**S&P 500**	**DJIA**
LSTM	0.019	0.011	0.025	0.007	**0.007**
WLSTM	0.019	0.009	0.018	0.006	**0.005**
KLSTM	0.014	0.007	0.015	0.008	**0.009**

Table 3. Predictive accuracy based on Theil U metric

	CSI	**Hang**	**Nikkei**	**S&P 500**	**DJIA**
LSTM	0.019	0.007	0.017	0.005	**0.004**
WLSTM	0.018	0.006	0.013	0.003	**0.003**
KLSTM	0.014	0.005	0.011	0.004	**0.005**

Table 4. Predictive accuracy based on R metric

	CSI	**Hang**	**Nikkei**	**S&P 500**	**DJIA**
LSTM	0.970	0.972	0.891	0.934	**0.933**
WLSTM	0.972	0.982	0.930	0.963	**0.961**
KLSTM	0.982	0.987	0.949	0.947	**0.906**

Table 5. Predictive accuracy based on RMSE metric

	CSI	**Hang**	**Nikkei**	**S&P 500**	**DJIA**
LSTM	126	354	598	21	**172**
WLSTM	122	280	478	15	**130**
KLSTM	96	240	407	18	**203**

It is clear from tables 2 to 5 that almost in all cases LSTM models which used noised eliminated data have better performance and lower error in predicting the actual value. It can also be interpreted from the results that the performance of LSTM neural networks in the datasets of developed markets is better than its performance in the datasets of developing and relatively-developed markets.

The reason of this issue is related to efficient market hypothesis (EMH) [28] which states that the efficiency of a market affects the predictability of its asset. In other words, the more advanced markets in which more traders participate and therefore more transactions are recorded, the patterns created in the stock price chart are more reliable and the stock trend and movements become more predictable.

6 Conclusion

Stock price prediction is considered to be difficult since stock data is a random walk time series which contains high level of noise. Hence, employing a noise reduction technique to process the underlying noise in stock data is inevitable. However, not all denoising techniques are beneficial for all stock markets. In other words, if we use an inappropriate technique to preprocess and denoise the data it will destroy the integrity and authenticity of the data which leads to miss some usefulness of the price information. Therefore, choosing the right denoising technique is an important task that should be considered with caution.

In order to achieve a more superior prediction results for the LSTM neural network, In this project, authors used two different algorithms of noise elimination which were Wavelet transform and Kalman filter. These algorithms have been used in order to investigate the effect of noise elimination from noisy financial time series on enhancing the LSTM model performance in comparison with Wavelet transform. With a glimpse on results, it is obviously clear that using a noise elimination technique such as Kalman filter or Wavelet transform can enhance the results of predicting financial markets and employing a noise elimination technique is absolutely necessary for anyone who is trying to predict the financial markets.

It can be concluded from the results that since Kalman Filter marginally outperforms Wavelet transform in denoising developing and relatively developed stock indexes, it is going to be a better choice for denoising systems that are less mature and have higher volatility. On the other hand, Wavelet Transform presents a better performance than Kalman filter in developed stock indexes which implies that it is more powerful in processing noise data and could be beneficial in denoising more advanced markets. These findings imply that, to better preserve the original information, processing noisy time series should be done with careful consideration of the stock market's level of maturity.

Employing a noise eliminator algorithm such as Wavelet Transform or Kalman Filter is an effective factor to enhance the performance of a LSTM model, which allows the neural network to better identify the available patterns in the price chart and make more precise predictions. According to the results of this project, deep learning approaches are capable of identifying and extracting hidden patterns in the financial time series and can accurately predict the future behavior of such markets.

One of the most demanding and time-consuming tasks in training deep learning models is to find an optimal value for hyperparameters which are commonly set with try and error such as number of LSTM units, regularization parameters, dropout percentage and so on. Future works could be the study of these hyperparameters and proposing methods which are capable of effectively finding the optimal values and enhance the performance of these networks. Also, the change in structure and the number of layers of LSTM neural networks can be studied to improve its performance by using professional noise elimination algorithms to eliminate noise from input data. Furthermore, profitability can be modeled by incorporating trading commissions into input variables.

References:

[1] Wang, B., H. Huang, and X. Wang, A novel text mining approach to financial time series forecasting. Neurocomputing, 2012. 83: p. 136-145.

[2] Pinches, G.E., The random walk hypothesis and technical analysis. Financial Analysts Journal, 1970. 26(2): p. 104-110.

[3] Schwartz, R.A. and D.K. Whitcomb, Evidence on the presence and causes of serial correlation in market model residuals. Journal of financial and quantitative analysis, 1977. 12(2): p. 291-313.

[4] Wong, B.K., T.A. Bodnovich, and Y. Selvi, Neural network applications in business: A review and analysis of the literature (1988–1995). Decision Support Systems, 1997. 19(4): p. 301-320.

[5] Hsieh, T.-J., H.-F. Hsiao, and W.-C. Yeh, Forecasting stock markets using wavelet transforms and recurrent neural networks: An integrated system based on artificial bee colony algorithm. Applied soft computing, 2011. 11(2): p. 2510-2525.

[6] Lee, C., Y. Sehwan, and J. Jongdae, Neural network model versus SARIMA model in

forecasting Korean stock price index (KOSPI). Issues in Information System, 2007. 8(2): p. 372-378.

[7] Rather, A.M., A. Agarwal, and V. Sastry, Recurrent neural network and a hybrid model for prediction of stock returns. Expert Systems with Applications, 2015. 42(6): p. 3234-3241.

[8] Adebiyi, A.A., A.O. Adewumi, and C.K. Ayo, Comparison of ARIMA and artificial neural networks models for stock price prediction. Journal of Applied Mathematics, 2014. 2014.

[9] Broomhead, D., et al., Singular system analysis with application to dynamical systems, in Chaos, noise and fractals. 2020, CRC Press. p. 15-27.

[10] Xiao, J., et al., A new approach for stock price analysis and prediction based on SSA and SVM. International Journal of Information Technology & Decision Making, 2019. 18(01): p. 287-310.

[11] Hill, J.B. and K. Motegi, Testing the white noise hypothesis of stock returns. Economic Modelling, 2019. 76: p. 231-242.

[12] Zhang, X. SVR-ARMA Stock Price Prediction Model Based on Wavelet Noise Reduction. in The Sixth International Conference on Information Management and Technology. 2021.

[13] Paisit Khanarsa, Arthorn Luangsodsai, and K. Sinapiromsaran, Self-Identification ResNet-ARIMA Forecasting Model. WSEAS Transactions on Systems and Control, 2020.

[14] Brogaard, J., et al. What moves stock prices? the role of news, noise, and information. in AFA Annual Meeting. 2020.

[15] Liang, X., et al., LSTM with wavelet transform based data preprocessing for stock price prediction. Mathematical Problems in Engineering, 2019. 2019.

[16] Zargar, F.N. and D. Kumar, Opening noise in the indian stock market: Analysis at individual stock level. Theoretical Economics Letters, 2019. 9(01): p. 21.

[17] Abramovich, F., P. Besbeas, and T. Sapatinas, Empirical Bayes approach to block wavelet function estimation. Computational Statistics & Data Analysis, 2002. 39(4): p. 435-451.

[18] Chen, Y., et al., Deep learning-based classification of hyperspectral data. IEEE Journal of Selected topics in applied earth observations and remote sensing, 2014. 7(6): p. 2094-2107.

[19] Grewal, M.S. and A.P. Andrews, Kalman filtering: Theory and Practice with MATLAB. 2014: John Wiley & Sons.

[20] Haykin, S., Kalman filtering and neural networks. Vol. 47. 2004: John Wiley & Sons.

[21] Zarchan, P., Progress in astronautics and aeronautics: fundamentals of Kalman filtering: a practical approach. Vol. 208. 2005: Aiaa.

[22] Hochreiter, S. and J. Schmidhuber, Long short-term memory. Neural computation, 1997. 9(8): p. 1735-1780.

[23] Preetham Ganesh, et al., POS-Tagging based Neural Machine Translation System for European Languages using Transformers. WSEAS Transactions on Information Science and Applications 18, 2021.

[24] How, D.N.T., C.K. Loo, and K.S.M. Sahari, Behavior recognition for humanoid robots using long short-term memory. International journal of advanced robotic systems, 2016. 13(6): p. 1729881416663369.

[25] Yu, W., X. Li, and J. Gonzalez. Fast training of deep lstm networks. in International Symposium on Neural Networks. 2019. Springer.

[26] Cao, X., et al. Anomaly Detection for Screw Tightening Timing Data with LSTM Recurrent Neural Network. in 2019 15th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN). 2019. IEEE.

[27] Zhao, H., Dynamic relationship between exchange rate and stock price: Evidence from China. Research in International Business and Finance, 2010. 24(2): p. 103-112.

[28] Malkiel, B.G., Efficient market hypothesis, in Finance. 1989, Springer. p. 127-134.