

## Determining the impact of window length on time series forecasting using deep learning

Ammar Azlan<sup>1\*</sup>, Yuhanis Yusof<sup>2</sup> and Mohamad Farhan Mohamad Mohsin<sup>3</sup>

Research Scholar, School of Computing, Universiti Utara Malaysia, Sintok, Kedah, Malaysia<sup>1</sup>

Associate Professor, School of Computing, Universiti Utara Malaysia, Sintok, Kedah, Malaysia<sup>2</sup>

Senior Lecturer, School of Computing, Universiti Utara Malaysia, Sintok, Kedah, Malaysia<sup>3</sup>

Received: 15-September-2018; Revised: 25-November-2018; Accepted: 30-January-2019

©2019 Ammar Azlan et al. This is an open access article distributed under the Creative Commons Attribution (CC BY) License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

### Abstract

*Time series forecasting is a method of predicting the future based on previous observations. It depends on the values of the same variable, but at different time periods. To date, various models have been used in stock market time series forecasting, in particular using deep learning models. However, existing implementations of the models did not determine the suitable number of previous observations, that is the window length. Hence, this study investigates the impact of window length of long short-term memory model in forecasting stock market price. The forecasting is performed on S&P500 daily closing price data set. A different window length of 25-day, 50-day, and 100-day were tested on the same model and data set. The result of the experiment shows that different window length produced different forecasting accuracy. In the employed dataset, it is best to utilize 100 as the window length in forecasting the stock market price. Such a finding indicates the importance of determining the suitable window length for the problem in-hand as there is no One-Size-Fits-All model in time series forecasting.*

### Keywords

*Deep learning, Long short-term memory (LSTM), Time series forecasting, Window length.*

### 1. Introduction

The time series is a series of observations recorded with respect to time. Usually, the time is set to equal interval, such as daily, monthly, or annually and is best visualized using a line chart. Time series forecasting is a method of predicting future observations based on previous observations. Unlike regression where the predicted value depends on other variables, time series depends on the same variable, but at different times [1]. An example of a problem that can be represented as time series is stock price prediction. According to the random walk hypothesis, stock market prices change in random order, hence making the stock market movement unpredictable [2]. Even though it is consistent with the efficient-market hypothesis, evidence from numerous studies indicates that it is predictable at some level [3–5]. It is also proven by the implementation of technical analysis, which is a class of technique that relies solely on the technical indicators of historical data and chart pattern [6], by most traders.

Historical stock price data, which is a time series data, have a natural temporal order where the current observation is more related to the nearer past observations in time compared to ones that are further [7]. The common practice among traders is to consider historical data of stock price from the previous 6 months. In time series problem, this duration is called window length or time lags. It represents the number of time steps in the time series to be inserted as input for the predictive model.

In this study, we investigated the impact of window length on stock price prediction using deep learning model. We employ the vanilla long short-term memory (LSTM) model for stock price forecasting with different window length as the input. The focus of this study is to empirically investigate the impact of different window length in time series forecasting. Therefore, the LSTM model and its parameters remain constant for all experiments to ensure that the difference in the result is caused by the window setting.

\*Author for correspondence

## 2.Related work

The time series is a sequential problem where the sequence of observations with respect to time needs to be preserved. Examples of time series problem are natural language processing [8], speech recognition [9], weather forecasting [10] and financial market prediction [11]. These problems require insights from previous input to make sense of the current input. This unique characteristic of time series problem that set it apart from other problems like regression and classification has attracted the interest of many researchers. Unfortunately, there is one major problem in time series forecasting which is to determine the window length.

Shynkevich et al. [12] defined window length as a time frame parameter required to be set when calculating many technical indicators. The window length was divided into two categories which are the past window length called input window length and future window length called forecast horizon. They studied with the effect of the combination of these types of window length on the performance of a predictive system. They used machine learning algorithms with technical indicators as input features to forecast the movement of stock price. The experiment was on daily price of 50 stocks for 10 years while the prediction accuracy, winning rate, return per trade, and Sharpe ratio was used as the performance metrics. Their observation indicates that a combination of input window length with approximate equal to the forecast horizon produced the highest predicted. This finding is consistent with the findings by Zhang et al. [13] which states that learning the correlations between different time steps is crucial in achieving artificial intelligence.

A branch of artificial intelligence called artificial neural network (ANN) progressively improves performance of general tasks using given examples without prior knowledge of the tasks. The ability of ANN to learn and generalize from previous data is well suited to problem domain such as time series forecasting. Moreover, the ANN is able to adapt to the data pattern and the relationship between the input and output, resulting in better prediction accuracy than the traditional method [14]. However, the performance of ANN highly depends on the number of training which is limited by the processing power and the number of data available at that time. Fortunately, the processing power and available data have exponentially increased these days, which led to the introduction of deep learning in 2015 [15].

Deep learning is a form of machine learning and is inspired by ANN. Similar to ANN, it also uses existing data to train a model in order to make predictions from new data [16]. However, unlike ANN, deep learning consists of many hidden layers. Although it is not clear on the number of hidden layers that separate shallow from deep learning, it is evidence that more hidden layers result to more abstract processes. The increase in the number of layers in a network requires high computational processing cost, which is why deep learning only gain popularity recently with the introduction of graphics processing unit (GPU). Moreover, digitization of processes which increase data availability for training also contributes to the rising popularity of deep learning since its performance relies heavily on the amount of training data. One of the rising deep learning architecture is called recurrent neural network (RNN).

RNN is a deep learning architecture that loops through time. This characteristic enables RNN to receive input from previous time step hence making it highly suitable for sequence problem like time series and lists. RNN has been proven to excel in numerous sequence problems like speech recognition [17], language modelling [18], translation [19], image captioning [20] and even image generation [21]. Despite its performance, RNN suffers from a long-term dependency problem which limits its ability to preserve past information with a big time gap as mentioned by Bengio et al. [22]. Fortunately, a solution was introduced by Hochreiter and Schmidhuber [23] to overcome the challenges faced by RNN and the relevant model is known as LSTM.

### 2.1Long short-term memory

LSTM is a deep learning model that prevents the vanishing and exploding gradient problem faced by RNN [23]. Instead of vanishing or exploding, errors can flow backwards through unlimited number of virtual layers unfolded in time. This is made possible with the implementation of LSTM cell which comprises of cell state, input gate, forget gate and output gate as illustrated in [24]. Cell state is an element that runs across the entire iteration with minimal linear interactions. Meanwhile, input, forget, and output gate are the elements that control cell state by deciding which information to be written, read, or erased.

The implementation of LSTM cell results in it being able to remember long-term dependencies. That is, LSTM can learn tasks that require memories of

events that happened thousands or even millions of discrete time step earlier [25]. LSTM also works well even if it is given long delays between significant events. In addition, it can also handle signals that mix low and high frequency components. As a result, LSTM performs very well in handwriting recognition [26] and speech recognition [27].

In addition, the application of LSTM has gained high popularity nowadays due to the attention it received from major companies like Google, Apple, Microsoft, and Amazon. These companies have adopted machine learning and neural network technologies to develop new products as well as to improve the performance of their existing products i.e. Google with its Google Assistant [28], Apple with its Siri [29], Microsoft with its Cortana [30], and Amazon with its Amazon Echo [31]. All these products are digital personal assistant, which requires the ability to process natural language, which is what LSTM is great for with record results [32].

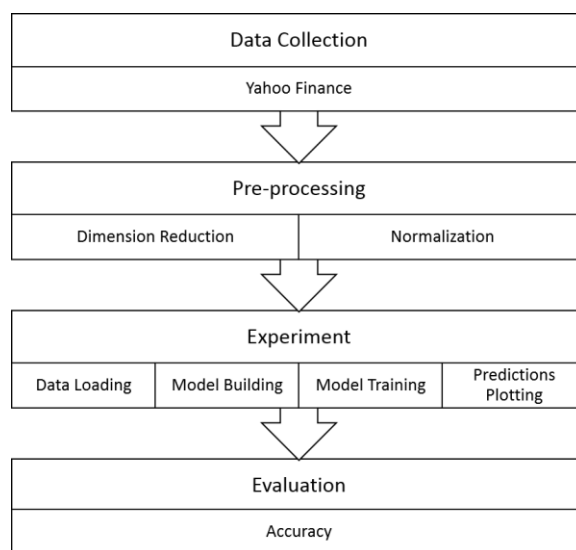
Besides natural language processing, LSTM also achieved state-of-the-art results in gesture recognition in video. Pigou et al. [33] proposed a deep architecture that incorporates bidirectional recurrence which is crucial for tasks with temporal feature and temporal convolutions which led to significant improvements. Their evaluation of different approaches to the Montalbano gesture recognition data set produced remarkable results. Another state-of-the-art research on the performance of LSTM is by Al-Smadi et al. [34] on sentiment analysis task. By employing LSTM on opinion target expressions extraction and sentiment polarity classification of Arabic Hotels’ reviews data set, their proposed approach outperformed baseline research by 39% and 6% respectively.

Among all the tasks that LSTM perform, we are interested in time series forecasting. Kim and Won [35] proposed a new model that integrates LSTM with generalized autoregressive conditional heteroskedasticity (GARCH) type models to forecast KOSPI 200 index price. They discovered that by integrating LSTM with multiple instead of single GARCH-type model, the prediction performance is enhanced significantly. This discovery is consistent with the findings by Fallah et al. [36]. Based on the analysis of more than 50 research papers, they concluded that hybrid models dominated over single model. On the other hand, Pang et al. [37] proposed a different approach where instead of coming up with a

hybrid model, they demonstrated the concept of stock vector. Unlike Kim and Won [35] who utilized multiple models, Pang et al. [37] utilized multi-stock higher-dimensional historical data through embedded layers. They discovered that embedded layers are better. Hence, by comparing the ability of LSTM with the characteristics of financial time series analysis, especially stock market prediction, the performance of LSTM on stock market prediction looks promising.

### 3.Method

In order to undertake the study, a total of four phases was required; data collection, data pre-processing, experiment, and evaluation. Details of the phases are illustrated in *Figure 1*.



**Figure 1** Research design

#### 3.1Data collection

To train our model, we used the daily closing price of S&P500 index from 3 January 2000 to 1 August 2016 which can be retrieved from Yahoo Finance for free. The data set is in CSV file format that consists of seven variables which are labelled as Date for transaction date, Open for opening price of the date, High for the highest price on the date, Low for the lowest price of the date, Close for the closing price of the date, Adj Close for the adjusted closing price of the date, and Volume for the transaction volume of the date. It also has 4171 observations for each record on the working day of a week within the selected period. An example of the data is shown in *Table 1*.

**Table 1** Data retrieved from yahoo finance

Date	Open	High	Low	Close	Adj Close	Volume
3/1/2000	1,469.25	1,478.00	1,438.36	1,455.22	1,455.22	931,800,000
4/1/2000	1,455.22	1,455.22	1,397.43	1,399.42	1,399.42	1,009,000,000
5/1/2000	1,399.42	1,413.27	1,377.68	1,402.11	1,402.11	1,085,500,000
...	...	...	...	...	...	...
29/7/2016	2,168.83	2,177.09	2,163.49	2,173.60	2,173.60	4,038,840,000
1/8/2016	2,173.15	2,178.29	2,166.21	2,170.84	2,170.84	3,505,990,000

### 3.2 Data pre-processing

The acquired data set originally consists of seven variables, namely Date, Open, High, Low, Close, Adj Close, and Volume as shown in *Table 1*. Since we are doing a univariate time series forecasting which involves only a single variable, the dimension of the data set was reduced where we kept variable Close and omitted other variables. The final data set that was fed into the program is shown in *Table 2*.

**Table 2** Final data set fed into the program

1455.22
1399.42
1402.11
...
2173.60
2170.84

The data were also normalized to improve convergence. The normalization was performed using Equation 1 which reflects percentage changes from the starting point. We divided each price  $p_i$  with initial price  $p_0$  and subtracted with 1.

$$n_i = \left(\frac{p_i}{p_0}\right) - 1 \quad (1)$$

After our model made the prediction, the data was denormalized using Equation 2 to get the real-world number. Equation 2 is the reverse of Equation 1. where the predicted normalized value  $n_i$  is added with 1 and multiplied with initial price  $p_0$ .

$$p_i = p_0(n_i + 1) \quad (2)$$

*Table 3* shows the original values labeled price and the normalized values labeled normalized price. The first value of Normalized Price is 0 because since it is the starting point of the time series, therefore there is no percentage change from the starting point. The values also range from -1 to 1 which is suitable with our model.

**Table 3** Normalized closing price

Price	Normalized price
1455.22	0
1399.42	-0.038345
1402.11	0.001922
...	...
2173.60	0.001631
2170.84	-0.001270

### 3.3 Experiment

The LSTM forecasting model for the experiment was developed in the Python programming language on Jupiter Notebook IDE using Keras with Google TensorFlow back-end. The experiments were executed on a desktop computer with Intel® Core™ i7-4770 CPU @ 3.40GHz processor, 8.00GB RAM, Windows 7 Professional 64-bit operating system, and HDD storage. The experiments consist of four steps which includes data loading, model building, model training, and predictions plotting. We run three experiments with different windows i.e. 25, 50, and 100 days to determine the impact of windows on our model.

### 3.4 Evaluation

The model was evaluated based on the accuracy of the prediction made by the model. The accuracy is calculated by dividing the number of correct predictions over the total number of predictions made as shown in Equation 3.

$$Accuracy = \frac{\text{Number of correct predictions}}{\text{Total number of predictions made}} \quad (3)$$

### 4. Results

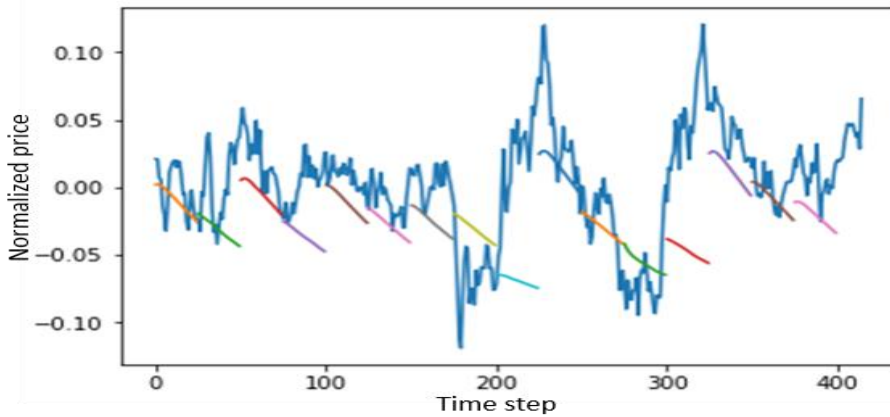
The first experiment is based on a 25-day window which means that the model uses input data from the past 25 days to predict the movement for the next 25 days. *Figure 2* illustrates the obtained prediction where the blue line represents the actual data while lines of other colors represent the predicted values at each point.

Based on the graph, the prediction accuracy of the model with 25-day window is 81.25% since 13 out of 16 points were predicted correctly. Points with incorrect prediction, i.e. point 75 (purple), 200 (light blue), and 300 (red) show that the graph moves downward while the graph of actual values moves upward.

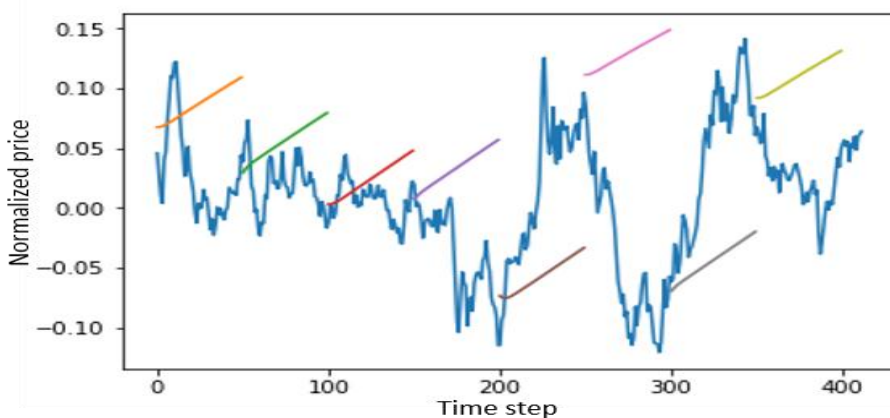
The second experiment is based on a 50-day window which means the model uses input data from the past 50 days to predict the movement for the next 50 days. The result is plotted on a graph as shown in *Figure 3*. The blue line is the actual data while other colors represent the predicted values. Based on the graph, the prediction of the model has been 62.5% accurate since five out of eight points were correctly predicted namely at point 0, 50, 100, 200, and 300. Points with incorrect prediction, i.e. point 150 (purple), 250 (pink), and 350 (yellow) show that the graph moves upward while the graph of actual values moves downward.

The third experiment involves the 100-day window length, which means the model employs data from the past 100 days to predict the movement for the next 100 days. The result is plotted on a graph as shown in *Figure 4*. Similarly, the blue line represents the actual data while other colors are the prediction data. Based on the graph, the prediction of the model has 100% accuracy since all points were correctly predicted.

Data in *Table 4* depicts the result summary. Even though experiment with 100-day window produced a perfect prediction with 100% accuracy, generalization that larger window size window contributes to a better performance could not be made. This is because the experiment with 50-day window has a lower accuracy compared to the 25-day window.

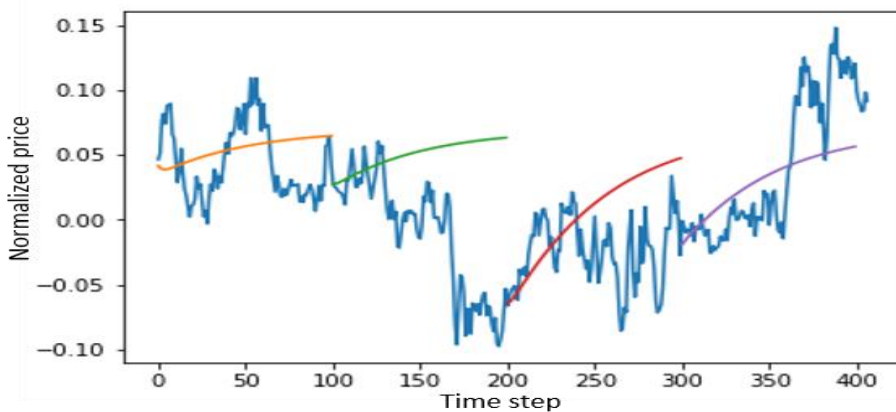


**Figure 2** Experiment result using 25-day window



**Figure 3** Experiment result using 50-day window





**Figure 4** Experiment result using 100-day window

**Table 4** Forecasting results using different window length

Window (DAY)	Accuracy (%)
25	81.25
50	62.5
100	100

## 5. Conclusion

This study suggests that window length is important in LSTM stock market time series forecasting. It shows that despite the general assumption that a deep learning model would perform better with more data, it is also important to identify which data contributes to its performance. Too little data would cripple the model performance while too many data would slow down its learning process or worse, reduce the performance. The data in this study are referred to window length, which had been proven to have an impact on time series forecasting using deep learning. Therefore, further investigation on automatic identification of optimized window length is required in order for a complete and the standalone forecasting application to be realized. Such an application will then facilitate investors in making decisions prior to investing in a particular stock.

## Acknowledgment

This study is funded by the Ministry of Higher Education, Malaysia under the Fundamental Research Grant Scheme (UUM S/O Code: 13820).

## Conflicts of interest

The authors have no conflicts of interest to declare.

## References

- [1] Ullah MI. Time series analysis and forecasting. 2013. <http://itfeature.com/time-series-analysis-and-forecasting/time-series-analysis-forecasting>. Accessed 17-July-2018.
- [2] Malkiel BG, McCue K. A random walk down wall street. New York: Norton; 1985.
- [3] Khare K, Darekar O, Gupta P, Attar VZ. Short term stock price prediction using deep learning. In international conference on recent trends in electronics, information & communication technology 2017 (pp. 482-6). IEEE.
- [4] Sezer OB, Ozbayoglu AM, Dogdu E. An artificial neural network-based stock trading system using technical analysis and big data framework. In proceedings of the southeast conference 2017 (pp. 223-6). ACM.
- [5] Zhang J, Cui S, Xu Y, Li Q, Li T. A novel data-driven stock price trend prediction system. Expert Systems with Applications. 2018; 97:60-9.
- [6] Murphy JJ. Study guide for technical analysis of the futures markets: a self-training manual. New York Inst. of Finance; 1987.
- [7] Heydt M. Learning pandas. Packt Publishing Ltd; 2017.
- [8] Collobert R, Weston J. A unified architecture for natural language processing: deep neural networks with multitask learning. In proceedings of the international conference on machine learning 2008 (pp. 160-7). ACM.
- [9] Hinton G, Deng L, Yu D, Dahl G, Mohamed AR, Jaitly N, et al. Deep neural networks for acoustic modeling in speech recognition. Signal Processing Magazine. 2012; 29: 82-97.
- [10] Grover A, Kapoor A, Horvitz E. A deep hybrid model for weather forecasting. In proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining 2015 (pp. 379-86). ACM.
- [11] Fischer T, Krauss C. Deep learning with long short-term memory networks for financial market

- predictions. *European Journal of Operational Research*. 2018; 270(2):654-69.
- [12] Shynkevich Y, McGinnity TM, Coleman SA, Belatreche A, Li Y. Forecasting price movements using technical indicators: investigating the impact of varying input window length. *Neurocomputing*. 2017; 264:71-88.
- [13] Zhang J, Liu J, Luo Y, Fu Q, Bi J, Qiu S, et al. Chemical substance classification using long short-term memory recurrent neural network. In *IEEE international conference on communication technology 2017* (pp. 1994-7). IEEE.
- [14] Vui CS, Soon GK, On CK, Alfred R, Anthony P. A review of stock market prediction with artificial neural network (ANN). In *IEEE international conference on control system, computing and engineering 2013* (pp. 477-82). IEEE.
- [15] LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature*. 2015; 521:436-44.
- [16] Heaton JB, Polson NG, Witte JH. Deep learning for finance: deep portfolios. *Applied Stochastic Models in Business and Industry*. 2017; 33(1):3-12.
- [17] Graves A, Mohamed AR, Hinton G. Speech recognition with deep recurrent neural networks. In *international conference on acoustics, speech and signal processing 2013* (pp. 6645-9). IEEE.
- [18] Mikolov T, Karafiát M, Burget L, Černocký J, Khudanpur S. Recurrent neural network-based language model. In *conference of the international speech communication association 2010* (pp.1045-8).
- [19] Chaudhary JR, Patel AC. Bilingual machine translation using RNN based deep learning. *International Journal of Scientific Research in Science, Engineering and Technology*. 2018; 4(4):1480-4.
- [20] Jindal V. Generating image captions in Arabic using root-word based recurrent neural networks and deep neural networks. In *thirty-second AAAI conference on artificial intelligence 2018* (pp.8093-4).
- [21] Gregor K, Danihelka I, Graves A, Rezende DJ, Wierstra D. Draw: a recurrent neural network for image generation. *International conference on machine learning 2015*.
- [22] Bengio Y, Simard P, Frasconi P. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*. 1994; 5(2):157-66.
- [23] Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Computation*. 1997; 9(8):1735-80.
- [24] Wu YX, Wu QB, Zhu JQ. Improved EEMD-based crude oil price forecasting using LSTM networks. *Physica A: Statistical Mechanics and its Applications*. 2019; 516:114-24.
- [25] Schmidhuber J. Deep learning in neural networks: an overview. *Neural Networks*. 2015; 61:85-117.
- [26] Chammas E, Mokbel C, Likforman-Sulem L. Handwriting recognition of historical documents with few labeled data. In *IAPR international workshop on document analysis systems 2018* (pp. 43-8). IEEE.
- [27] Tüske Z, Schlüter R, Ney H. Investigation on LSTM recurrent n-gram language models for speech recognition. *Interspeech 2018* (pp. 3358-62).
- [28] Huffman S. The Google assistant, powering our new family of hardware. 2017. <https://www.blog.google/products/assistant/google-assistant-powering-our-new-family-hardware/>. Accessed 13-September-2018.
- [29] Leswing K. Apple completely changed how Siri works and almost nobody noticed. 2016. [https://www.businessinsider.my/apples-siri-using-neural-networks-2016-8/?r=US&IR=T.\(checked\)](https://www.businessinsider.my/apples-siri-using-neural-networks-2016-8/?r=US&IR=T.(checked)). Accessed 13-September-2018.
- [30] <https://venturebeat.com/2018/09/14/gamesbeat-decides-96-tom-nook-goes-to-work/>. Accessed 13-September-2018.
- [31] <http://www.zdnet.com/article/amazon-echo-the-four-hard-problems-amazon-had-to-solve-to-make-it-work>. Accessed 13-September-2018.
- [32] <http://mattmahoney.net/dc/text.html>. Accessed 13-September-2018.
- [33] Pigou L, Van Den Oord A, Dieleman S, Van Herreweghe M, Dambre J. Beyond temporal pooling: recurrence and temporal convolutions for gesture recognition in video. *International Journal of Computer Vision*. 2018; 126(2-4):430-9.
- [34] Al-Smadi M, Talafha B, Al-Ayyoub M, Jararweh Y. Using long short-term memory deep neural networks for aspect-based sentiment analysis of Arabic reviews. *International Journal of Machine Learning and Cybernetics*. 2019; 10(8):2163-75.
- [35] Kim HY, Won CH. Forecasting the volatility of stock price index: a hybrid model integrating LSTM with multiple GARCH-type models. *Expert Systems with Applications*. 2018; 103:25-37.
- [36] Fallah S, Deo R, Shojafar M, Conti M, Shamshirband S. Computational intelligence approaches for energy load forecasting in smart energy management grids: state of the art, future challenges, and research directions. *Energies*. 2018; 11(3):1-31.
- [37] Pang X, Zhou Y, Wang P, Lin W, Chang V. An innovative neural network approach for stock market prediction. *The Journal of Supercomputing*. 2018:1-21.



**Ammar Azlan** is a PhD student in Computer Science at School of Computing, Universiti Utara Malaysia, Malaysia. He received his MSc in Information Technology from Universiti Utara Malaysia, Malaysia and Bachelor of Information Technology from Universiti Teknologi MARA, Malaysia. His research interests are Artificial Intelligence, Machine Learning, Deep Learning, and Time Series Forecasting. Currently, he is involved in projects on species distribution modeling as a graduate research assistant and zakat collection prediction with Lembaga Zakat Negeri Kedah Darul Aman.  
Email: ammar\_azlan@ahgs.uum.edu.my



**Yuhanis Yusof** is an Associate Professor in the School of Computing, Universiti Utara Malaysia, Malaysia. She obtained her PhD in Computer Science from Cardiff University, UK. She also holds an MSc degree in Computer Science from Universiti Sains Malaysia and Bachelor of Information Technology from Universiti Utara Malaysia. Her research interest is broadly in Data Analytics and Management for Large Scale Computing. This includes Data Mining (discovering patterns of interest from data), Information Retrieval, Optimization and Decision Support System. Currently, she is involved in several projects relating to Machine Learning and Swarm Intelligence. Her research work is realized in several domains, including Finance, Education, Business, Environment, Cyber Security and Transport Management.  
Email: yuhanis@uum.edu.my



**Mohamad Farhan Mohamad Mohsin** is a senior lecturer and researcher at School of Computing, Universiti Utara Malaysia, Malaysia. He obtained his PhD in Data Mining and Optimization and MSc degree in Information Technology from Universiti Kebangsaan Malaysia, Malaysia. He also holds a Bachelor of Information Technology from Universiti Utara Malaysia, Malaysia. What he has done is generally to discover the secret of machine learning principle, algorithm, intelligent system, and its capability to infer human cognition and decision-making process. In deep, his main research interest is in Machine Learning and Data Mining projects that seek hidden information from huge, complex data set and finally generate models to ease the human decision-making process. At the moment, most of his research is in Anomaly Detection and Predictive Modeling mainly in climate change studies and outbreak. Now he is engaged with the Artificial Immune System, several Bio-Inspired Approaches, and Text Mining based research.  
Email: farhan@uum.edu.my