Day Ahead Forecasting of FAANG Stocks Using ARIMA, LSTM Networks and Wavelets.

Tom Skehin, Martin Crane, Marija Bezbradica

School of Computing, Dublin City University, Ireland
{tom.skehin2, martin.crane, marija.bezbradica}@dcu.ie

Abstract. Facebook Inc., Apple Inc., Amazon.com Inc., Netflix Inc. and Alphabet Inc., known collectively as FAANG, are a group of the best performing tech stocks in recent years. In this study, we present linear and non-linear methods for predicting the closing price of each stock on the following day. We decompose each time series into component series using wavelet methods and develop an novel ensemble approach to improve forecast accuracy.

Keywords: Machine Learning, Time Series Analysis, Neural Networks, Signal Processing.

1 Introduction

Facebook Inc. (FB), Apple Inc. (AAPL), Amazon.com Inc (AMZN), Netflix Inc. (NFLX) and Alphabet Inc. (GOOG), are a group of multinational companies traded publicly on the NASDAQ. As of 9$^{\text{th}}$ August 2018 they had a combined market capitalisation of $3.48 Trillion [3]. They make 1.5% of the total market capitalisation the S&P 500 and are known collectively as FAANG, a term coined by Jim Cramer, a former fund manager at Goldman Sachs.

In this paper we forecast multiple univariate time series of the daily closing prices of FAANG stocks. We fit the linear Autoregressive Integrated Moving Average (ARIMA) model and the non-linear Long Short Term Memory (LSTM) network to each series to produce next day predictions. Wavelet methods decompose a series into approximation and detail components to better explain behaviour over time. We combine these techniques in a novel ensemble model in an attempt to increase forecast accuracy.

In Section 2 we review the foundation literature of the methods used and discuss related research. In 3 we provide the background theory behind the models. Section 4 details the implementation of our models, while 5 and 6 explain the experiments and how the forecast accuracy is measured, followed with results and discussion in Sections 7 and 8.

2 Related Research

Linear modelling methods such as the Autoregressive model (AR) and Moving Average (MA) models have been prominent in time series forecasting for decades, particularly due to their intuitive nature and robustness in producing short-term forecasts over a wide range of subject matters.

In their seminal textbook [9], Box and Jenkins, described their method for applying an ARIMA model and finding a best fit for a time series. Since then ARIMA models have been employed in modelling diverse series such as inflation in the Irish economy [17] and failures in repairable systems [23].

LSTM Networks date from 1997 [13] as an improvement the previous RNN algorithms, which took a prohibitive amount of time to learn long term time lags. They are widely used in natural language processing (NLP) [22] and speech recognition tasks [12].

Wavelet decomposition, the theory of using small wave like functions or *wavelets* to deconstruct a time series into its approximation and detail components, is described in [18]. Wavelet methods are prominent in signal processing [19] and time series analysis [6] and have been used effectively in multi-resolution image processing [14].

Ensemble approaches combining wavelet decomposition with forecasting methods have been extensively researched with two notable examples:

Yousefi et al [24] use wavelets to forecast crude oil prices over different time horizons using monthly WTI[1] spot prices. These forecasts were compared with NYMEX[2] oil futures prices and are shown to better predict the future spot prices, challenging the perception the futures market is efficiently priced. The authors decompose the series into wavelet component series and extend them using a sinusoidal model, before reconstructing. They suggest expanding the research to other time series methods for extending the component series such as ARIMA or GARCH.

Liu et al [16] combine wavelet decomposition and principal components analysis (PCA) to create input features for multiple models. The performance of these models is compared when training and testing on two classical datasets: the artificial Mackey-Glass equation and the real-world, mean daily flow of the Oldman River near Brocket.

The five implemented models were as follows: Cascade correlation and back propagation neural networks (CCNN and BPNN) on the underlying the differenced, stationary time-series. CCNN's on the wavelet decomposed time series, in one instance with all the decomposed series fed into one network, the another with each sub-series as an input to it's own individual network (WCCNN and WCCNN multi-models). The final network introduces PCA as a preprocessing step on the decomposed series and the first k principal components that account for 85% of the information are selected as inputs (PCA-WCCNN).

As a result of this work, the researchers suggest PCA-WCCNN as more accurate than WCCNN and quicker to train than WCCNN multi-models.

[1] West Texas Intermediate
[2] New York Metal Exchange

3 Methodology

In this section we describe the techniques utilised in our proposed models by describing the mathematical concepts behind ARIMA models and LSTM networks, used in forecasting future values of our selected time series. Finally, we discuss wavelet methods for transforming our series into component sub-series, which we can consider individually or use as a multivariate input to our models.

We define a **univariate time series** X as a time ordered finite set of t observations $X = \{x_1, x_2, x_3, \ldots, x_t\}$ commonly recorded at evenly spaced intervals, with each $x_i \in X$ representing the value of the series at that time.

3.1 Autoregressive Integrated Moving Average (ARIMA) Model

An ARIMA process is a linear combination of p Autoregressive (AR) terms and q Moving Average (MA) terms, modelling a d order differencing of a time series X [9]:

$$\left(1 - \sum_{i=1}^{p} \alpha_i L^i\right)(1-L)^d x_t = \left(1 + \sum_{i=1}^{q} \beta_i L^i\right)\epsilon_t + c$$

where c is a constant and L is a *lag operator*, such $Lx_t = x_{t-1}$, $L^2 x_t = x_{t-2}$ etc.

The selection of appropriate values for the parameters p, d and q is crucial to fitting the most accurate model to the time series. The Auto-Correlation Function (ACF) and Partial Auto-Correlation Function (PACF) can be used to provide a reasonable estimate for p [9].

The Akaike Information Criterion (AIC) is useful for determining the relative performance of a model to another with different parameters [20].

3.2 Long Short-Term Memory (LSTM) Networks

LSTM Networks are a subclass of Recurrent Neural Network (RNN) that are capable of learning long term dependencies or "long term memory" in time series data [10] as opposed to RNNs which place more emphasis on recent inputs or "short term memory". We provide a brief description of each but start with a general definition of an artificial neural network.

At its core, an artificial neural network (ANN) is a collection of nodes or *neurons*, connected by weighted edges or *synapses*, mimicking the biological neurons in the human brain.Each neuron can receive multiple inputs, update it's current state and emit an output based on an internal activation function, for example a sigmoid or *tanh* function [10]. The weights of each synapse determine the influence the output of neuron, with a low weighted synapse meaning that the output of the preceding node is effectively ignored.

One limitation of ANN's is that due to their acyclic nature, all inputs and outputs are independent of each other, that is, the predicted value, $\hat{x}_t$ does not depend on $\hat{x}_{t-1}$ or $\hat{x}_{t-2}$ etc. This is not ideal for time series modelling with a high level of autocorrelation between previous terms.

RNNs address this issue by allowing *feedback loops* within the hidden layers, where a neuron can store information from current step to make predictions in the following steps. In practice, however, RNN's are only suitable for storing a few previous steps and cannot remember long sequences of data which may better inform the prediction. [21]

Long term memory is improved in LSTM's with the introduction of gate functions within the neurons [13] for transforming the state of current step, to be used in the following steps. Three types of gate exist within each node:

- **Filter Gate:** A sigmoid function which determines how much of the previous state should persist in the current state, a value of 1 implies everything should be remembered, while 0 completely disregards the previous state.
- **Memory Gate:** Determines how much of the new data should be added to the current state of the neuron.
- **Output Gate:** Combines the the new data and the current state of the node to produce an output.

As the LSTM is trained it should learn which values in the series to remember or disregard over a longer term than the RNN alone.

3.3 Maximal Overlap Discrete Wavelet Transform (MODWT)

A wavelet transform breaks a time series into components of different frequencies [7] and is better than other decomposition methods as it provides information on the different frequencies present in a series while preserving the positions of these frequencies in the time domain. In this section we discuss the fundamentals of wavelet transformation theory and the application of *Discrete Wavelet Transform (DWT)* and *Maximal Overlap Discrete Wavelet Transform (MODWT)* to a time series.

There are two main functions involved in wavelet analysis, the wavelet function (*mother wavelet* ψ) and the scaling function (*father wavelet* ϕ). In this paper we consider the Haar and Daubechies of order 4 (*db4*) [7] wavelet functions, with the Haar father and mother wavelets as follows:

$$\phi(t) = \begin{cases} 1, \text{ if } 0 \le t < 1 \\ 0, \text{ otherwise} \end{cases} \quad \text{and} \quad \psi(t) = \begin{cases} 1, & 0 \le t < \frac{1}{2} \\ -1, & \frac{1}{2} \le t < 1 \\ 0, & \text{otherwise} \end{cases}$$

In general, the mother and father wavelets have the property

$$\int \phi(t)dt = 1 \quad \text{and} \quad \int \psi(t)dt = 0$$

In a discrete wavelet transform (DWT) the time series is decomposed using the father and mother *wavelet functions*:

$$\phi_{j,k}(t) = 2^{\frac{j}{2}}\phi(2^j t - k) \quad \text{and} \quad \psi_{j,k}(t) = 2^{\frac{j}{2}}\psi(2^j t - k)$$

Where $j \in [1, J]$ in a J-level decomposition is the *scale* and k is the *shift* parameter [7]. The set of these functions provide an orthonormal basis [7] and as such we can represent our original time series $x(t)$ as

$$x(t) = \sum_k A_{J,k}\phi_{J,k}(t) + \sum_k D_{J,k}\psi_{J,k}(t) + \ldots + \sum_k D_{1,k}\psi_{1,k}(t)$$

The set of coefficients of the wavelet functions in the decomposition, $\{A_J, D_J, D_{J-1}, \ldots, D_1\}$, is known is a *decomposition crystal*, with each element $D_j = \{d_{j,1}, d_{j,2}, \ldots, d_{j,k}\}$ a series in its own right.

The A_J series reflect the approximation component or trend of the series, the D_i components represent the detail or deviation from this trend.

The *Maximal Overlap Discrete Wavelet Transform (MODWT)* is a non orthogonal redundant transform which repeats information in neighbouring coefficients by overlapping time series values [7]. This allows the decomposed coefficient sets to have the same cardinality as the original series. Similarly, the *Inverse Maximal Overlap Discrete Wavelet Transform (IMODWT)* takes the decomposition crystal from the MODWT and reconstructs the original series.

4 Proposed Models

In this section we discuss the implementation of the models used in our experiments and the data acquired for training them. The implementation logic for ARIMA and LSTM have been adapted from [4] and [5] respectively.

4.1 Data Model

The time series we focus on the daily closing prices of the shares in FB, AAPL, AMZN, NFLX and GOOG over the period 1$^{\text{st}}$ January 2010 to 1$^{\text{st}}$ January 2017. The number of observations vary between each stock due a number of corporate actions that occurred during the interval: Facebook Inc. was not listed on the NASDAQ for the entire window, meaning there are only closing prices subsequent to their IPO in May 2012. Alphabet Inc. (aka Google) undertook a corporate restructure in March 2014 and issued a new class of share. It is this new class that is the subject of our analysis. All of these time series were obtained from Alpha Vantage Inc. [1] and are made available for replication.

These series have been adjusted to account for stock-splits and buybacks [2] so the transformation required is for the target feature. We are predicting log returns based on the assumption the prices are log normally distributed [8]. For each stock we have a time series of the form $X = \{x_1, \ldots, x_t\}$ where $x_i = \log\left(\frac{p_i}{p_{i-1}}\right)$ and p_i, p_{i-1} are the current and previous closing prices respectfully. The models described in the following sections will be used to produce 100 one-day forecasts for the final 100 terms in each series.

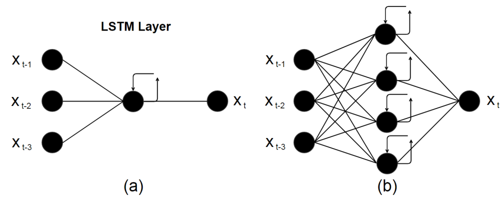

Fig. 1. LSTM topologies: (a) 1 LSTM hidden layer, (a) 4 LSTM hidden layer

4.2 ***PREVCLOSE***

This is our base line model for measuring the relative prediction improvements in subsequence models. Here, the predicted value for x_t is the previous day's closing price, denoted $\hat{x}_t = x_{t-1}$.

4.3 ARIMA

We use following steps to produce day ahead forecasts using ARIMA:

- **Step 1:** The training set $\mathbb{X}$, for the model is extracted from the series X, $\mathbb{X} = \{x_1, \ldots, x_{t-100}\}$.
- **Step 2:** For all possible combinations of (p, d, q) with $p, q \in \{1, 2, 3\}$ and $d \in \{0, 1\}$ an ARIMA(p, d, q) is fitted on $\mathbb{X}$ using maximum likelihood estimation to determine the model coefficients [11].
- **Step 3:** The model with the least AIC is selected and forecast $\hat{x}_{t-99}$ is generated.
- **Step 4:** The observed value x_{t-99} is added to the training set $\mathbb{X}$ and **Steps 2-3** are repeated to produce a forecast for x_{t-98}.

The iterative forecasting process continues until we have a complete set of forecast values $\{\hat{x}_{t-99}, \ldots, \hat{x}_t\}$.

4.4 LSTM

Two different LSTM network topologies were implemented, a 1 node and a 4 node LSTM hidden layer with both having an input layer of 3 nodes. Both are shown in Figure 1.

In order to fit the network, the time series X, needs to be transformed into feature and target spaces $\tilde{X}$ and Y respectively. In the 3 node input layer, the previous 3 observations are used to predict the current:

$$\tilde{X} = \{(x_1, x_2, x_3), (x_2, x_3, x_4), \ldots, (x_{t-3}, x_{t-2}, x_{t-1})\} \quad Y = \{x_4, x_5, \ldots, x_t\}$$

Forecasts were produced with the following steps.

- **Step 1:** The training feature and target sets are defined as $\mathbb{X} = \{\tilde{x}_1, \ldots, \tilde{x}_{t-100}\}$ and $\mathbb{Y} = \{y_1, \ldots, y_{t-100}\}$.

- **Step 2:** The elements of $\mathbb{X}$ and $\mathbb{Y}$ are standardised such that all values lie in the interval $[-1, 1]$. This is to improve convergence performance and avoid local minima [15].
- **Step 3:** The network is trained over 1,000 epochs using the standardised training set $\mathbb{X}'$. The internal memory of the network is reset after each pass to ensure that future observations do not inform the parameter fitting.
- **Step 4:** The set $\mathbb{X}'$ is input to the fitted network so that the internal memory reflects the time $t - 100$. This memory will be available for use in forecasts
- **Step 5:** The standardisation function for $\mathbb{X}$ is applied to the test tuples $\{\tilde{x}_{t-99}, \ldots, \tilde{x}_{t\}}$ which are sequently input into the model to output the forecasts $\{\hat{y}_{t-99}, \ldots, \hat{y}_t\}$.
- **Step 6:** The inverse standardisation function for $\mathbb{Y}$ is applied to $\{\hat{y}_{t-99}, \ldots, \hat{y}_t\}$ to produce a complete set of forecast values $\{\hat{x}_{t-99}, \ldots, \hat{x}_t\}$.

4.5 WAV-ARIMA

An extension to the ARIMA model, this method applies the MODWT to the series before applying the ARIMA model as above. The steps are as follows:

- **Step 1:** The training set $\mathbb{X}$, for the model is extracted from the series X.
- **Step 2:** The J-level MODWT is applied to $\mathbb{X}$ to produce the component series $A_J, D_J, D_{J-1}, \ldots, D_1$
- **Step 3:** The ARIMA process above is applied to each component series $D_j = \{d_{j,1}, \ldots, d_{j,t-100}\}$ to produce the forecast $\hat{d}_{j,t-99}$
- **Step 3:** The forecasts are appended to each component series, the IMODWT is applied and the final term in the constructed series is the forecast $\hat{x}_{t-99}$.
- **Step 4:** The observed value x_{t-99} is added to the training set $\mathbb{X}$ and **Steps 2-3** are repeated to produce a forecast for x_{t-98}.

The iterative forecasting process continues until we have a complete set of forecast values $\{\hat{x}_{t-99}, \ldots, \hat{x}_t\}$.

4.6 WAV-LSTM

The network topologies implemented were the same as those in LSTM, as are most of the steps for producing forecasts. The main difference lies in the transformation of the series into the feature and target spaces, where the real values $x_i \in \mathbb{R}$, are replaced with vectors $\langle a_{J,i}, d_{J,i}, \ldots, d_{1,i} \rangle \in \mathbb{R}^{J+1}$. Out target and feature sets are now:

$$\tilde{X} = \left\{ \begin{array}{c} (\langle a_{J,1}, \ldots, d_{1,1} \rangle, \langle a_{J,2}, \ldots, d_{1,2} \rangle, \langle a_{J,3}, \ldots, d_{1,3} \rangle), \\ (\langle a_{J,2}, \ldots, d_{1,2} \rangle, \langle a_{J,3}, \ldots, d_{1,3} \rangle, \langle a_{J,4}, \ldots, d_{1,4} \rangle), \\ \vdots \\ (\langle a_{J,t-3}, \ldots, d_{1,t-3} \rangle, \ldots, \langle a_{J,t-1}, \ldots, d_{1,t-1} \rangle) \end{array} \right\} \quad Y = \{x_4, x_5, \ldots, x_t\}$$

The **Steps 1-6** are repeated from the LSTM process to produce a complete set of forecast values $\{\hat{x}_{t-99}, \ldots, \hat{x}_t\}$.

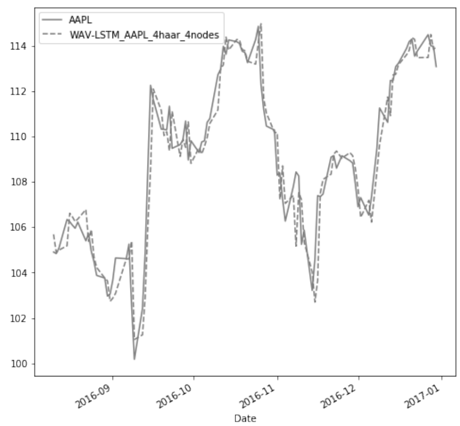

Fig. 2. Day ahead forecast of AAPL with 4-Level Haar MODWT in a four node LSTM

5 Experiments

The model implementations from the previous section were used to perform several pieces of analysis. By way of determining a baseline accuracy of forecasting, we applied PREVCLOSE to the five series FB, AAPL, AMZN, NFLX and GOOG. These baseline forecasts were used to calculate a comparative uplift in accuracy between the models for each series.

Primarily, we were interested in assessing the impact of the MODWT on forecast accuracy by comparing performance the wavelet and non-wavelet models. We first applied the ARIMA and the two LSTM network topologies in Figure 1 to each series. For WAV-ARIMA we applied 4-level and 7-level Haar MODWTs and for the WAV-LSTM, both Haar and *db4* 4-level and 7-level MODWT's. In total this was 13 models for each of the 5 series.

6 Evaluating Performance

With each of the 100-step log return forecasts $\{\hat{x}_{t-99}, \ldots, \hat{x}_t\}$, we were able to produce a set of predicted closing prices $\{\hat{p}_{t-99}, \ldots, \hat{p}_t\}$ with $\hat{p}_i = \hat{p}_{i-1}e^{\hat{x}_i}$ and $\hat{p}_{t-100} = p_{t-100}$. Figure 2 displays the day ahead forecast of AAPL using WAV-LSTM_4haar_4nodes. With these forecasts we calculated the *root mean squared error (RMSE)* as a measure of a model's accuracy. The RMSE metric has the benefit of having the same units as the closing prices we are predicting.

7 Results

In this section we display the model forecast accuracies for the each stock, a results summary can be seen in Table 1. For each stock, we present PREVCLOSE

Table 1. Summary of RMSE's of Each Model. Best Performing MODWT displayed.

Model	PREVCLOSE RMSE	RMSE	% Uplift	Wavelet	MODWT RMSE	% MODWT Uplift	Difference
FB ARIMA	2.07053	1.53865	25.69%	7haar	1.54491	25.39%	-0.30%
FB LSTM_1nodes	2.07053	1.54293	25.48%	7db4	1.53856	25.69%	0.21%
FB LSTM_4nodes	2.07053	1.55138	25.07%	4db4	1.60988	22.25%	-2.83%
AAPL ARIMA	1.52926	1.20641	21.11%	7haar	1.22980	19.58%	-1.53%
AAPL LSTM_1nodes	1.52926	1.18124	22.76%	4db4	1.17908	22.90%	0.14%
AAPL LSTM_4nodes	1.52926	1.18096	22.78%	4haar	1.13724	25.63%	2.86%
AMZN ARIMA	15.01143	10.67527	28.89%	4haar	10.81004	27.99%	-0.90%
AMZN LSTM_1nodes	15.01143	10.91169	27.31%	4db4	10.77066	28.25%	0.94%
AMZN LSTM_4nodes	15.01143	10.96300	26.97%	7haar	11.59515	22.76%	-4.21%
NFLX ARIMA	3.81128	2.59786	31.84%	4haar	2.66228	30.15%	-1.69%
NFLX LSTM_1nodes	3.81128	2.58489	32.18%	4haar	2.59204	31.99%	-0.19%
NFLX LSTM_4nodes	3.81128	2.60693	31.60%	4db4	2.60830	31.56%	-0.04%
GOOG ARIMA	10.44286	7.53559	27.84%	7haar	7.67317	26.52%	-1.32%
GOOG LSTM_1nodes	10.44286	7.64011	26.84%	7haar	7.52944	27.90%	1.06%
GOOG LSTM_4nodes	10.44286	8.21606	21.32%	4db4	8.39511	19.61%	-1.71%

Table 2. Complete result set for all models, topologies and MODWT's

Stock	PREVCLOSE
FB	2.070532462
AAPL	1.529257509
AMZN	15.01142692
NFLX	3.811278731
GOOG	10.44286479

Stock	ARIMA	WAV-ARIMA	
		Level 4 Haar	Level 7 Haar
FB	1.53865366	1.54520178	1.544907855
AAPL	1.206405729	1.230079949	1.229804921
AMZN	10.67527361	10.81561318	10.8100369
NFLX	2.597855905	2.662275598	2.664253074
GOOG	7.535593313	7.682333036	7.673170282

Stock	No. Nodes	LSTM	WAV-LSTM			
			Level 4 Haar	Level 4 d4	Level 7 Haar	Level 7 db4
FB	1 node	1.54292574	1.54683345	1.55667579	1.56150331	1.53856127
	4 node	1.55137747	1.71497296	1.60988027	1.76003729	1.69619960
AAPL	1 node	1.18123917	1.19427418	1.17908097	1.19860824	1.20337696
	4 node	1.18095573	1.13724068	1.25800922	1.26328940	1.29806305
AMZN	1 node	10.91168764	10.93125569	10.77066223	11.00314703	11.58069993
	4 node	10.96300010	11.72580050	12.13373019	11.59514859	11.83471550
NFLX	1 node	2.58489304	2.59204445	2.60772126	2.78392026	4.87489893
	4 node	2.60692975	2.64937129	2.60829730	2.99247356	2.71020707
GOOG	1 node	7.64010799	7.53330948	7.59589637	7.52944289	7.57958487
	4 node	8.21605979	8.93254555	8.39510627	10.56156750	10.74900998

RMSE as a baseline to compare with the model RMSE. The accuracy uplift is the percentage improvement in RMSE.

The best performing MODWT is selected with the MODWT RMSE and MODWT Uplifts displayed. The difference between the two uplifts is a measure of the improvement in accuracy resulting from the MODWT. The full set of results for all MODWT can be seen in Table 2.

8 Discussion

Both the ARIMA and LSTM significantly outperform the PREVCLOSE forecasts in all implementations, providing around a 20 - 30% lift in forecast accuracy. ARIMA outperforms both the LSTM topologies across the majority for stocks, with AAPL the only instance where there is a notable improvement ($> 1\%$ increase in uplift) when using the LSTM.

We suspect this is because of the difference in forecasting methods between the two approaches. With ARIMA, a new model is trained with each step of the 100 day forecast to incorporate each new current day observation. With LSTM, the model is fit on the initial training data only and while the node states will update as the test data is played through the network, the synapse weights are fixed. We suggest there may an increase in LSTM accuracy if the network is retrained after each forecast to include the observed value for the current day.

The 1 node LSTM hidden layer continually outperformed the 4 node layer in both RMSE and MODWT RMSE, with AAPL the only exception, demonstrating that an increase in model complexity does not always translate to an improvement in model performance.

All of the ARIMA models outperformed their WAV-ARIMA counterparts, suggesting that the component series are less suitable to this type of modelling than the original series. The majority of the differences in uplifts between the LSTM and WAV-LSTM models were negligible ($< 1\%$ difference in uplifts), with the exception of the APPL 4 node WAV-LSTM with a 4-level Haar decomposition which had a 2.86% increase in accuracy uplift and the AMZN 4 node WAV-LSTM 7-level Haar with a 4.21% decrease.

In summary, the results show no significant gains in forecast accuracy resulting from the MODWT in either the ARIMA or LSTM models. This contrary to the research in [24] and [16] and would suggest there are further refinements to be made to our approach.

Our selection of the 4-level and 7-level decomposition were arbitrary and further work in our method could make this a more informed decision by incorporating the PCA as seen in [16]. Similarly our choice of the number of input nodes in LSTM, in this instance a 3-lag input, could better inferred by making using of the Partial Autocorrelation Function (PACF) in the ARIMA model.

A major assumption of the ARIMA model is constant variance or volatility. This seems unlikely in a real world scenario given the trading volumes that surround company specific or geo-political events like earnings reports or the USA Presidential election. An ARIMA with addition explanatory variables (ARIMAX) or the General Autoregressive Conditional Heteroskedasticity (GARCH) model as suggested in [24] could better describe this behaviour.

9 Conclusion

In this paper we deployed the ARIMA and LSTM models for forecasting future values in a univariate time series. We demonstrated the MOWDT, a technique for decomposing a time series into approximation and detail components, while preserving the temporal information of the original series. We combined these techniques and developed an ensemble approach in order to produce 100 1-day ahead forecasts of the daily closing price of FAANG stocks on the NASDAQ exchange. We presented our results and offered possible suggestions for the outcomes. Finally we recommended further avenues of investigation in order to expanded and improve upon our work.

References

1. Alpha Vantage. `https://www.alphavantage.co` (2018), [Online; accessed 10-Aug-2018]

2. Investopedia - What are Corporate Actions? https://www.investopedia.com/articles/03/081303.asp (2018), [Online; accessed 10-Aug-2018]
3. Yahoo! Finance. https://finance.yahoo.com (2018), [Online; accessed 10-Aug-2018]
4. Auquan, T.: Time Series Analysis for Financial Data V - ARIMA Models. https://medium.com/auquan/time-series-analysis-for-finance-arima-models-acb5e39999df (2017), [Online; accessed 10-Aug-2018]
5. Brownlee, J.: Time Series Forecasting with the Long Short-Term Memory Network in Python. https://machinelearningmastery.com/time-series-forecasting-long-short-term-memory-network-python/ (2017), [Online; accessed 10-Aug-2018]
6. Conlon, T., Crane, M., Ruskin, H.J.: Wavelet multiscale analysis for hedge funds: Scaling and strategies. Physica A: Statistical Mechanics and its Applications 387(21), 5197–5204 (2008)
7. Crowley, P.M.: A guide to wavelets for economists. Journal of Economic Surveys 21(2), 207–267 (2007)
8. Egan, W.J.: The distribution of s&p 500 index returns (2007)
9. George E. P. Box, G.M.J.: Time Series Analysis: Forecasting and Control. Holden Day (1970)
10. Graves, A.: Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850 (2013)
11. Hannan, E., McDougall, A.: Regression procedures for arma estimation. Journal of the American Statistical Association 83(402), 490–498 (1988)
12. Hasim Sak, A.W.S., Beaufays, F.: Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. CoRR abs/1402.1128 (2014)
13. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation 9(8), 1735–1780 (1997)
14. Kingsbury, N.: Image processing with complex wavelets. Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences 357(1760), 2543–2560 (1999)
15. Laurent, C., Pereyra, G., Brakel, P., Zhang, Y., Bengio, Y.: Batch normalized recurrent neural networks. arXiv preprint arXiv:1510.01378 (2015)
16. Liu, T., Wei, H., Zhang, C., Zhang, K.: Time series forecasting based on wavelet decomposition and feature extraction. Neural Computing and Applications 28(1), 183–195 (2017)
17. Meyler, A., Kenny, G., Quinn, T.: Forecasting irish inflation using arima models. Central Bank and Financial Services Authority of Ireland Technical Paper Series 1998(3/RT/98), 1 – 48 (1998)
18. Percival, D.B., Walden, A.T.: Wavelet Methods for Time Series Analysis. Cambridge University Press (2000)
19. Rioul, O., Vetterli, M.: Wavelets and signal processing. IEEE Signal Processing Magazine 8(4), 14–38 (1991)
20. Ripley, B.: Statistical theories of model fitting. INATO ASI SERIES F COMPUTER AND SYSTEMS SCIENCES 168, 2–26 (1998)
21. Siami-Namini, S., Namin, A.S.: Forecasting economics and financial time series: Arima vs. lstm. arXiv preprint arXiv:1803.06386 (2018)
22. Sundermeyer, M., Schlüter, R., Ney, H.: Lstm neural networks for language modeling. In: Thirteenth annual conference of the international speech communication association (2012)

23. Walls, L., Bendell, A.: Time series methods in reliability. Reliability Engineering 18(4), 239 – 265 (1987)
24. Yousefi, S., Weinreich, I., Reinarz, D.: Wavelet-based prediction of oil prices. Chaos, Solitons & Fractals 25(2), 265–275 (2005)