Deep Learning and Wavelets for High-Frequency Price Forecasting

Andrés Arévalo[1], Jaime Nino[1], Diego León[2], German Hernandez[1], and Javier Sandoval[1]

[1] Universidad Nacional de Colombia
{ararevalom,jhninop,gjhernandezp}@unal.edu.co
[2] Universidad Externado de Colombia
{javier.sandoval,diego.leon}@uexternado.edu.co

Abstract. This paper presents improvements in financial time series prediction using a Deep Neural Network (DNN) in conjunction with a Discrete Wavelet Transform (DWT). When comparing our model to other three alternatives, including ARIMA and other deep learning topologies, ours has a better performance. All of the experiments were conducted on High-Frequency Data (HFD). Given the fact that DWT decomposes signals in terms of frequency and time, we expect this transformation will make a better representation of the sequential behavior of high-frequency data. The input data for every experiment consists of 27 variables: The last 3 one-minute pseudo-log-returns and last 3 one-minute compressed tick-by-tick wavelet vectors, each vector is a product of compressing the tick-by-tick transactions inside a particular minute using a DWT with length 8. Furthermore, the DNN predicts the next one-minute pseudo-log-return that can be transformed into the next predicted one-minute average price. For testing purposes, we use tick-by-tick data of 19 companies in the Dow Jones Industrial Average Index (DJIA), from January 2015 to July 2017. The proposed DNN's Directional Accuracy (DA) presents a remarkable forecasting performance ranging from 64% to 72%.

Keywords: Short-term Forecasting; High-frequency Forecasting; Computational Finance; Deep Neural Networks; Discrete Wavelet Transform

1 Introduction

Modeling and forecasting financial time series continue to be a very difficult task [17,16,21,18,6]. Techniques to address this problem can be split into two categories: analytical models and Machine Learning. On one hand, analytical techniques include statistical and stochastic models, such as Linear Regression (LR), Multiple Linear Regression (MLR), Autoregressive Integrated Moving Average (ARIMA), Generalized Autoregressive Conditional Heteroskedasticity (GARCH/N-GARCH), Brownian Motion (BM), Diffusion, Poisson and Levy Processes. On the other hand, Machine Learning Techniques (MLT) include Decisions Trees, Artificial Neural Networks (ANN), Fuzzy Systems, Kernel

Methods, Support Vector Machines (SVM) and recently, Deep Neural Networks (DNN), an extension of Artificial Neural Networks [6,16,20].

We selected Deep Learning because MLT are good while dealing with non-linearities and complexities, such as those present in financial data, in addition to its capabilities to handle large amounts of data, such as those present on HFD. Moreover, we decided to use wavelets as a feature generator because the sequential behavior of HFD, there are many transactions at the same price and changes (price jumps), under normal market conditions should occur with not high variance. Moreover, Deep Learning has been successfully applied in many different fields including price forecasting, as a result, we think that this kind of representation can improve previous results achieved in [1].

The paper is organized as follows: Section 2 presents a theoretical overview of Time Series, ARIMA, Artificial Neural Networks, and Wavelets. Section 3 describes the dataset used. Section 4 presents the proposed model to forecast one-minute pseudo-log-returns. Section 5 presents some baseline models which are used to compare performance against the proposed method. Section 6 presents final results and their analysis. Finally, section 7 presents conclusions and recommendations for future research.

2 Theory overview

2.1 Financial time series

A time series is a sequence of successive data points with a time order. A Financial Time Series (FTS) is a sequence of financial data points, like price, volume (quantity of financial asset), or any transformation of the previous ones. A FST is a non-stationary process. Formally, a FST X is denoted: $X = X_t : t \in T$

Where T is an index set, where each element is labeled by a date time stamp, and it is associated only with one data point for a specific financial asset.

Forecasting models seek to predict aptly the next value of the series X without the knowledge of future. F, the predicted time series, is the sequence of predicted values. In order to assess model's performance of F, it is necessary to determine the similarity between X and F. Popular similarity measures for time series include [14]:

− Mean Squared Error (MSE): A scale dependent measure.

$$MSE = \frac{1}{n} \sum_{t=1}^{n} (X_t - F_t)^2 \tag{1}$$

− Directional Accuracy (DA): A scale independent measure. Percent of predicted directions that matches the original time series. DA is widely used in finance [22].

$$DA = \frac{100\%}{n-1} \sum_{t=2}^{n} 1_{sign(X_t - X_{t-1}) = sign(F_t - F_{t-1})} \tag{2}$$

Where 1 is an indicator function: $1_A = f(x) = \begin{cases} 1, & A = \text{True} \\ 0, & A = \text{False} \end{cases}$

A common transfromation to make a FST stationary consists of getting the Log-return or pseudo-log-return series.

Log-return. Let p_t be the current closing price and p_{t-1} the previous closing price.

$$R = \ln \frac{p_t}{p_{t-1}} \cdot 100\% = (\ln p_t - \ln p_{t-1}) \cdot 100\% \tag{3}$$

From a log-return R, the price p_t can be reconstructed as follows: $p_t = p_{t-1} \cdot e^{\frac{R}{100\%}}$

Pseudo-log-return. It is a logarithmic difference (log of quotient) between average prices on consecutive minutes. . Let $\overline{p_t}$ be the current one-minute average price and $\overline{p_{t-1}}$ the previous one-minute average price.

$$\hat{R} = \ln \frac{\overline{p_t}}{\overline{p_{t-1}}} \cdot 100\% = (\ln \overline{p_t} - \ln \overline{p_{t-1}}) \cdot 100\% \tag{4}$$

Pseudo-returns can be reconstructed just as log-returns: $\overline{p_t} = \overline{p_{t-1}} \cdot e^{\frac{\hat{R}}{100\%}}$

2.2 ARIMA

Traditionally, econometric models dominate the forecasting arena, where statistical linear methods such as ARIMA. ARIMA(p, d, q) and (Auto-Regressive Integrated Moving Average with orders p, d, q), are the most frequently used. In general, these models are a set of discrete time linear equations with noise, of the form:

$$\left(1 - \sum_{k=1}^{p} \alpha_k L^k\right)(1 - L)^d X_t = \left(1 - \sum_{k=1}^{q} \beta_k L^k\right)\epsilon_t. \tag{5}$$

Particularly, ARIMA forecasting equation is a linear regression-type equation, in which the predictors consist of lags of a dependent variable and lags of predicted errors, where p is the number of autoregressive terms, d is the number of nonseasonal differences needed for stationarity, and q is the number of lagged forecast errors in the prediction equation.

Despite the relative success of these models, they have low capacity to capture market movements, given complexities and non-linear relationships[6] exhibit in financial markets. For these reasons machine learning methods have emerged as an important alternative to handle this kind of problem, since they are able to recognize complex patterns, and they have the ability to process large amounts of data.

2.3 Artificial Neural Networks

The first class of ANN was the Feed-forward Neural Network (FNN), which has multiple neurons connected to each other, but there are no cycles or loops in the network. Therefore, the information always moves forward from input to output

nodes. A Multilayer Perceptron (MLP), which is a FNN subtype, has an input, multiple hidden layers, and one output layer. Each layer has a finite number of neurons that are fully connected to all neurons in the next layer [11].

Since the late 1980s, techniques using ANNs have been widely used to forecast financial time series, due to its ability to extract essential features and to learn complex information patterns in high dimensional spaces [8,16,21,23].

Deep learning (DL) models have demonstrated a greater effectiveness in both classification and prediction tasks, in different domains such as video analysis, audio recognition, text analysis and image processing. Models based on DL attracts the interest of general public because they are able to learn useful representations from raw data, avoiding the local minimum issue of ANNs, by learning in a layered way using a combination of supervised and unsupervised learning to adjust weights W. Nevertheless its advantages, DL applications in computational finance are limited [3,4,7,24,28,25].

Within DL there is a wide variety of architectures, the most simple one uses MLP. However, for the purpose of this paper, we will be using more complex ones such as Recurrent Neural Networks (RNN), Gate Recurrent Units (GRU) and Long Short Term Memory (LSTM). A RNN is an ANN that has connections from output units to input ones, such that a directed cycle is formed. Under this architecture, the network is feedback by the output data; this allows modeling temporal behavior dynamics when storing previous inputs or outputs in an internal memory [11,13,19]. The first known application of a RNN in finance "Stock price pattern recognition-a recurrent neural network approach" was published in [15].

Historically, RNN are better at learning time series, because they are designed to identify patterns through time. But they include a greater complexity than the MLP, and therefore they are more difficult to train. In 1997, the Long Short-Term Memory (LSTM), a kind of RNN, was proposed in [13]. It solved some issues of recurrent networks related to learning too much time dependencies; LSTMs are capable of learning in a balanced way both long and short-term dependencies [10]. Recently, [5] proposed Gated Recurrent Unit (GRU), a variation of LSTM. It combines several internal components of the LSTM; making it simpler than LSTM because it has fewer parameters to be fitted during the training phase.

2.4 Wavelets

A wavelet is a wave function with an average value of zero. One key difference is duration; wavelets, unlike sinusoids, have a finite duration, that is, they have a beginning and an end [9]. Wavelets are a useful way to extract key features from signals in order to reproduce them without having to save or keep the complete signal. Moreover, wavelets possess additional advantages that help to overcome non-stationarity associated with financial time series. In order to get a theoretical background on wavelets, it is important to start with Multi-resolution Analysis (MRA).

Definition 1: A MRA on R is a sequence of subspaces V_j, $j \in Z$ on functions L^2 on R that satisfies the following properties [26,12]:

- $\forall j \in Z, V_j \subset V_{j+1}$
- if f(x) is C_c^0 on R, then f(x) $\in \overline{span} \quad V_j, j \in Z$. Given $\epsilon > 0, \exists \quad j \in Z$ and a function g(x) $\in V_j$, such that $\|f - g\|_2 < \epsilon$
- $\bigcap_{j \in Z} V_j = 0$
- A function $f(x) \in V_0$ if and only if $f(2x) \in V_{j+1} \forall j \in Z$
- $\exists$ a function $\varphi(x), L^2$ on R, called the scaling function, such that $\{\tau_n \varphi(x)\}$ is an orthonormal system of translates and $V_0 = \overline{span}\{\tau_n \varphi(x)\}$

MRA allows an exact calculation of the wavelet coefficients for an L^2 function. Let $\{V_j\}$ an MRA with scaling function $\varphi(x)$, therefore [26]:

- $\varphi(x) = \sum_n h(n)\varphi_{1,n}(x)$, is the scaling function.
- $\psi(x) = \sum_n g(n)\varphi_{1,n}(x)$, is the corresponding wavelet.

Where $g(n) = (-1)^n \overline{h(1-n)}$, is the wavelet filter.

Wavelets transforms could be either continuous or discrete. Since financial time series are discrete, Discrete Wavelet Transform (DWT) is more suitable to filter the data. DWT is defined as follows [26]:

Having a signal $c_o(n)$, its DWT is a sequence collection: $\{d_j(k) : 1 \leq j \leq J; k \in Z\} \cup \{c_j(k) : k \in Z\}$, where

$$
\begin{aligned}
c_{j+1}(n) &= (Hc_j)(n) \\
d_{j+1}(n) &= (Gc_j)(n) \\
c_j(n) &= (H^*c_{j+1})(n) + (G^*d_{j+1})(n) \\
\\
(Hc)(k) &= \sum_n c(n)\overline{h(n-2k)} \\
(Gc)(k) &= \sum_n c(n)\overline{g(n-2k)}
\end{aligned}
\tag{6}
$$

Wavelet analysis offers the following advantages [12]:

- It does not require a strong assumption about the data generation process.
- It provides information in both time and frequency domains.
- It has the ability to locate discontinuities in the data.

However, it has some disadvantages [12]: It requires that the length of the data be 2^j. It is not shift invariant. Finally, it may shift data peaks, causing wrong approximations when compared to the original data.

Because in a previous work [1], a simple DNN achieved good predictability on high frequency data for two different NYSE equities, we will use DWT as a feature selector, to test the DNN's ability to improve forecasting capabilities when using as inputs features extracted from wavelet analysis of price sequences.

Table 1. Dataset description

Company	Exchange	Symbol	# Ticks	File size
Apple	NASDAQ	AAPL	146.98 M	7.1 GB
American Express	NYSE	AXP	4.15 M	197 MB
Boeing	NYSE	BA	3.55 M	168 MB
Caterpillar	NYSE	CAT	4.59 M	218 MB
Cisco Systems	NASDAQ	CSCO	58.11 M	2.8 GB
Chevron	NYSE	CVX	7.1 M	340 MB
DuPont Corporation	NYSE	DD	3.66 M	170 MB
Walt Disney	NYSE	DIS	6.34 M	304 MB
General Electric	NYSE	GE	4.76 M	224 MB
Goldman Sachs	NYSE	GS	3.2 M	151 MB
The Home Depot	NYSE	HD	4.35 M	206 MB
IBM	NYSE	IBM	3.59 M	173 MB
Intel	NASDAQ	INTC	70.47 M	3.4 GB
Johnson & Johnson	NYSE	JNJ	5.77 M	278 MB
JPMorgan Chase	NYSE	JPM	7.9 M	375 MB
Coca-Cola	NYSE	KO	3.72 M	174 MB
McDonald's	NYSE	MCD	0.57 M	28 MB
3M	NYSE	MMM	2.63 M	127 MB
ExxonMobil	NYSE	XOM	8.13 M	386 MB
Total			349.56 M	17 GB

3   Proposed Method

3.1   Dataset Description

Data from the Dow Jones Industrial Average (Dow30) is used. Dow30 is the second-oldest U.S. market index and consists of 30 major US companies. Tick-by-tick data from 19 randomly chosen companies are used. Data were downloaded from January 1st, 2015 to July 31th, 2017. Table 1 shows some dataset descriptors.

3.2   Preprocessing

HFD means high transaction frequency (microseconds to seconds), our approach includes representing a typical one minute for this kind of data. We know that prices can vary dramatically in financial markets, however, given the liquidity of $\boldsymbol{DOW30's}$ instruments, under normal market conditions, they should exhibit the following characteristics:

- There are many transactions, occurring within short intervals (usually milliseconds/microseconds)
- Prices should not have much volatility between ticks.
- There should not be many different prices.
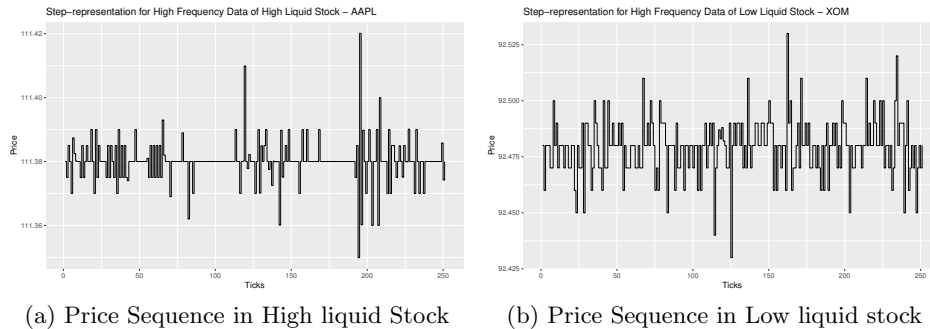- Price jumps should not be very wide.

(a) Price Sequence in High liquid Stock (b) Price Sequence in Low liquid stock

Fig. 1. Stock sequences

This means that for a typical one minute period, any instrument should exhibit similar behavior. The average number of price jumps (measured in cents) is lower as the price jump is larger, and usually, the less liquid stocks exhibit bigger jumps between ticks. There is a strong correlation between an average number of ticks per minute and the number of streaks or sequences occurring at the same price. That is $P_t = P_{t-1}$. Moreover, we can see that nearly 50% of the ticks have the same price in most of the stocks, except for the more liquids (AAPL, CSCO, INTC), where this measure is near 72% on average. We infer that price discovery is more efficiently performed. As a result, there is a larger proportion of ticks at the same price. In contrast, less liquid stocks exhibit a lower percentage of ticks at the same price, and wider price jumps between ticks. Graphically, this behavior is seen in figures 1a and 1b; The charts have a step style that makes them very similar. This kind of behavior is suitable for wavelet decomposition since the signal exhibits stepped changes between points. This is the reason to use DWT, in order to choose Wavelets coefficients as inputs, because on HFD traded prices can be interpreted as price intervals. Next, we will illustrate in detail the steps to select inputs for the DNN.

All data were summarized at a one-minute level. For each minute, the one-minute pseudo-log-return is calculated. Average prices are calculated from all of the ticks for the particular minute. Moreover, the tick-by-tick series is compressed to 8 values using the DWT.

A log transformation is applied to all tick-by-tick prices on a minute, then iterated differences are performed. The resulting time series is decomposed until the last possible level using the DWT with a Haar filter. Finally, Wavelet and Scaling coefficients from the last two levels are selected. The final result is a vector of 8 values: 4 wavelet coefficients and 4 scaling coefficients.

The last level is composed of only 2 coefficients (1 wavelet coefficient and 1 scaling coefficient). But the one before could be composed of only 4 or 6 coefficients depending on the number of transactions that were made in a particular minute. When this is the case, there are 8 coefficients on the last two levels. When the second level is composed of only 8 coefficients, 2 zeros are appended

Table 2. Compressed tick-by-tick series using DWT: Case I

Level	Wavelet Coefficients						Scaling Coefficients					
3	5.65						24.04					
2	2		2		2		5		13		21	
1	0.70	0.70	0.70	0.70	0.70	0.70	2.12	4.94	7.77	10.60	13.43	16.26

Time series:	1	2	3	4	5	6	7	8	9	10	11	12
Final vector:	5.65	2	2	2	24.04	5	13	21				

(a) Case I

Level	Wavelet Coefficients		Scaling Coefficients	
3	5.65		24.04	
2	2	2	5	13
1	0.70 0.70 0.70 0.70		2.12 4.94 7.77 10.60	

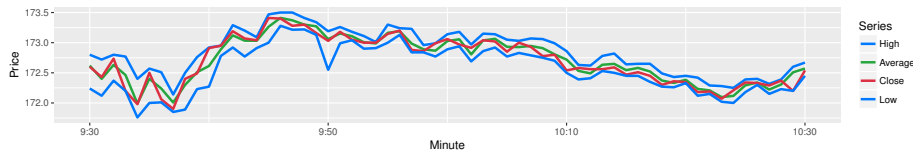| Time series:	1	2	3	4	5	6	7	8
Final vector:	5.65	2	2	0	12.72	5	13	0

(b) Case II

Fig. 2. Example of 60 one-minute prices.

to the vector in order to obtain 8 values. Table 3a shows decomposition of numbers ranging from 1 to 12. Table 3b shows number decomposition ranging from 1 to 8.

3.3 Modeling

Figure 2 shows the average price, a good descriptor of market behavior. The highest or lowest prices are usually found within a confidence range of average price, therefore the next highest and lowest prices can be estimated from a predicted average price. The closing price is the last price within a minute, it can be the highest/lowest price inclusive. Unlike the average price, the highest, lowest and close are exposed largely to market dynamics.

Since this work's objective is to build the best possible forecaster, average prices could be more suitable for this purpose than closing prices. With a good average price forecast, it is known that traded prices will match at some point in time the predicted average during the next minute.

Selected inputs consist of 27 values: The last 3 one-minute pseudo-log-returns and the last 3 compressed tick-by-tick time series (8x3). The selected output is the next predicted one-minute pseudo-log-returns.
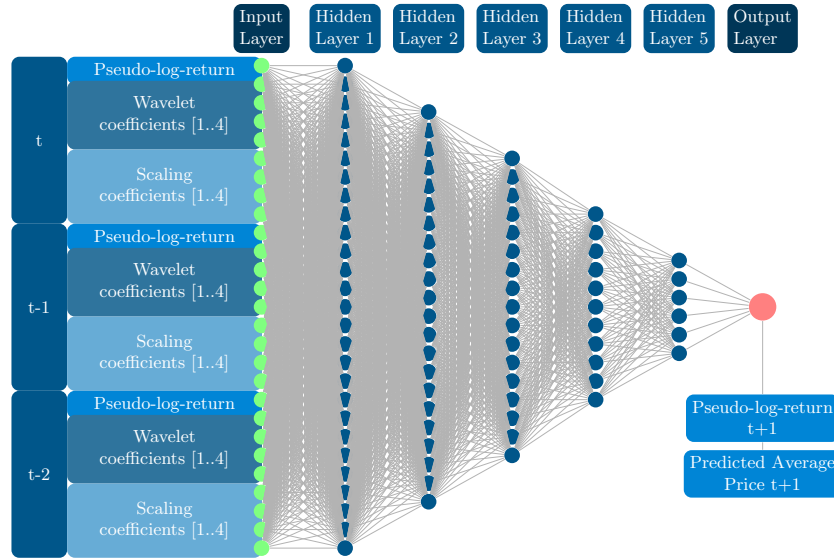
Fig. 3. DNN's Architecture

The network architecture consists of one input layer, 5 hidden layers, and one output layer. For each hidden layer, the number of neurons decreases with a constant factor $\frac{1}{5}$, (27, 22, 17, 11 and 6 neurons respectively). The output layer has one neuron. All neurons in hidden layers use a ***Tanh*** activation function, whereas the output neuron uses a ***Linear*** activation function. Figure 3 shows the architecture overview.

4   Experiments

The proposed method will be compared against the following methods or models, in order to verify its performance.

Table 3. Experiment summary

Method	Architecture	Inputs	Reason to use
ARIMA	Analytical formula	Logarithmic rescale	Traditional Econometric Forecasting Model
DNN	5 layers	4 categories, described in [1,2]	Baseline work
GRU	2 layers	Same as proposed method	More suitable DL topologies for time series
LSTM	2 layers	Same as proposed method	More suitable DL topologies for time series

ARIMA: Although it has good performance over in-sample datasets, it has bad performance over out-sample datasets. The time series were rescaled to a logarithmic scale to help to stabilize strong growth trends. And then, many ARIMA models were fitted with the Augmented Dickey-Fuller Test, the Auto
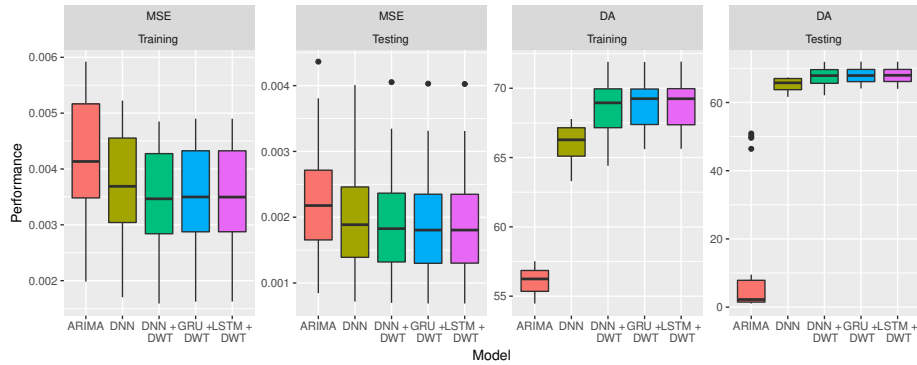
Fig. 4. Model's Performance during the training and testing phases

and Cross-Covariance and Correlation Function Estimation (ACF) and the Partial Auto and Cross-Covariance and Correlation Function Estimation (PACF). Finally, for each dataset, the ARIMA model with the lowest AIC was selected.

DNN: [1,2] proposed a DNN which also forecasts the next one-minute pseudo-log-return. DNN's inputs are composed of four groups: the current time (hour and minute), the last n one-minute pseudo-log-returns, the last n one-minute standard deviations of prices and the last n one-minute trend indicators, where n is the window size. The trend indicator is a statistical measure computed as the linear model's slope ($price = at + b$) fitted on transaction prices for a particular minute. LSTM and GRU architectures only have two layers, whereas our base work exhibits a five layers network. The main reason to change the number of layers was training times. Given the memory effect of GRU and LSTM, complexity of these topologies is greater, therefore reducing the number of layers decreases training times.

5 Results and analysis

The dataset was split up into two parts: an in-sample dataset (first 85%) and an out-sample dataset (last 15%). For each symbol and machine learning model, ten artificial networks were trained and then the average error was calculated and reported. Overall, all networks had homogeneous and stable results.

Figure 4 shows the Model Performances during the training and testing phases. Figure 5a shows the Model Performances per Symbol during the training and testing phases, whereas Figure 5b shows the DA performance on a wider scale.

Given the analysis of models performance over out-sample datasets, machine learning techniques had a much better performance compared to ARIMA. Meanwhile, networks using DWT are slightly better than the one without DWT; On average, DNN without DWT, DNN with DWT, GRU with DWT and LSTM with DWT achieved a MSE of 0.002026, 0.001963, 0.001941 and 0.001939, and a
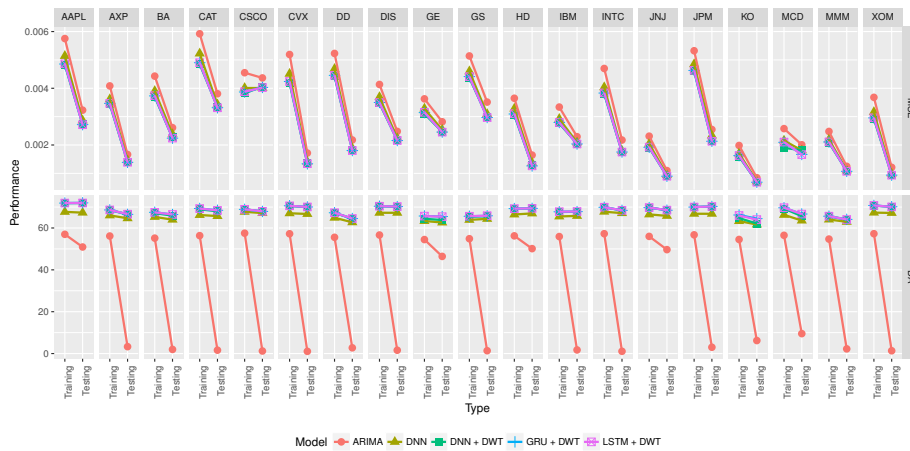
(a) All models
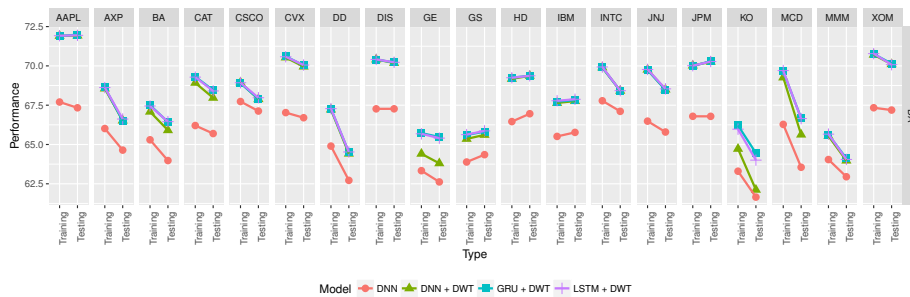
(b) All models except ARIMA

Fig. 5. Model's Performance per Symbol during the training and testing phases

DA of 65.27%, 67.38%, 67.74% and 67.72% respectively during the testing phase. The best model was GRU, though its performance is almost equal to LSTM.

On the other hand, GRU is definitely better than LSTM, because although it has the same performance as LSTM, GRU has less complexity and fewer parameters than LSTM, therefore, the training time is reduced and this method is more suitable for use in a real environment.

Overall, DNNs can learn market dynamics with reasonable precision and accuracy over out-sample datasets. But DA performance may be correlated to liquidity. For instance, symbols AAPL, CSCO and INTC, which exhibit the most liquidity, got much better DA performance, close to 72%, 68%, and 68% respectively. Otherwise, other symbols that are less liquid, like MMM (10 ticks/minute) had a lower DA performance, close to 64%. It is important to mention that more liquidity means more data, the raw material for Machine Learning Techniques.
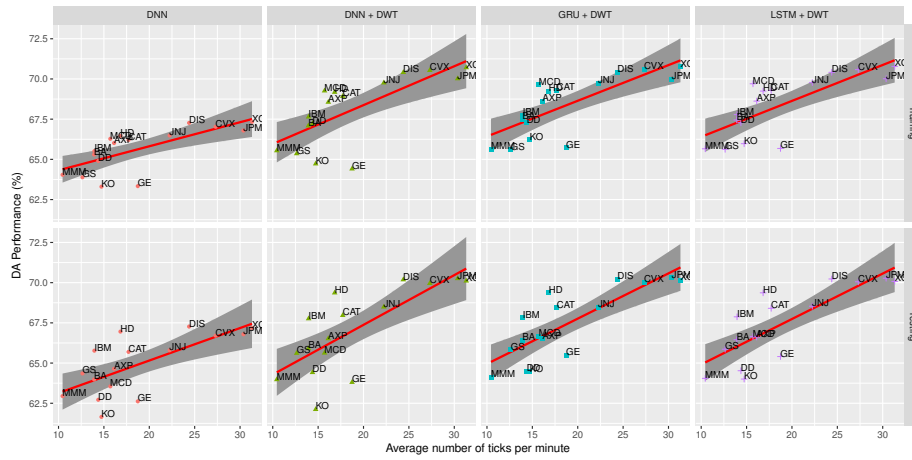
Fig. 6. Average number of ticks per minute vs. DA Performance

A Pearson Correlation Test and a Spearman Correlation Test were performed in order to verify the dependency between the average number of ticks per minute and DA performance archived by each machine learning model on all out-sample datasets. The Pearson correlation archived by DNN without DWT, DNN with DWT, GRU with DWT and LSTM with DWT on all symbols were 0.4407459, 0.4524886, 0.4695877 and 0.4648353 respectively. Whereas, the Spearman correlation achieved by DNN without DWT, DNN with DWT, GRU with DWT and LSTM with DWT over all symbols were 0.7684211, 0.7368421, 0.7649123 and 0.7701754 respectively.

The Spearman correlation suggests that there is a non-linear correlation between liquidity and DA Performance. Hence, as liquidity is higher, the proposed model has greater effectiveness. It is important to clarify that the model effectiveness will be stuck at some unknown point, in other words, the proposed models would never reach 100% precision despite the liquidity of a particular instrument.

Figure 6 shows the relationship between the average number of ticks per minute and DA Performance for each machine learning model. Stocks with higher liquidity, (AAPL, CSCO, and INTC), were excluded for better visualization. As we can see, the most traded the stock the best model predicts. Also, it draws a fitted Generalized Additive Model (GAM).

6   Conclusions

Traders collectively repeat the behavior of the traders that preceded them [27]. Those patterns can be learned by a DNN. The proposed strategy replicates the concept of predicting prices for short-time periods.

Feature selection is a very important step while building a machine learning model. Moreover, frequency domain with temporal resolution, produced by the DWT, allows the network to identify more complex patterns in the time series.

In liquid stocks there are many ticks in a minute, therefore the mean is more stable than another less liquid stocks. As a result, the final average price time series has less noise, therefore, a machine learning technique can perform better for these kinds of stocks.

Within the deep learning arena, there are more models than the ones depicted here. As a result, a possible research opportunity could be to evaluate model's performance against other DL models such as Deep Belief Networks, Convolutional Networks, Deep Coding Networks, among others. Overall, Deep Learning techniques can learn market dynamics with a reasonable precision and accuracy, for instance, using recurrent networks increased model's performance.

Another opportunity would be to explore higher resolution levels, other wavelet filters families such as Daubechies, Coiflets, Symlets, Discrete Meyer, Biorthogonal, among others, as well as other discrete transformations.

References

1. Arévalo, A., Niño, J., Hernández, G., Sandoval, J.: High-Frequency Trading Strategy Based on Deep Neural Networks pp. 424–436 (2016), `http://link.springer.com/10.1007/978-3-319-42297-8_40`
2. Arévalo Murillo, A.R.: Short-Term Forecasting of Financial Time Series with Deep Neural Networks. bdigital.unal.edu.co (2016), `http://www.bdigital.unal.edu.co/54538/`
3. Arnold, L., Rebecchi, S., Chevallier, S., Paugam-Moisy, H.: An Introduction to Deep Learning. ESANN (2011), `https://www.elen.ucl.ac.be/Proceedings/esann/esannpdf/es2011-4.pdf`
4. Chao, J., Shen, F., Zhao, J.: Forecasting exchange rate with deep belief networks. In: The 2011 International Joint Conference on Neural Networks. pp. 1259–1266. IEEE (7 2011), `http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=6033368http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6033368`
5. Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) pp. 1724–1734 (6 2014), `http://arxiv.org/abs/1406.1078`
6. De Gooijer, J.G., Hyndman, R.J.: 25 years of time series forecasting. International Journal of Forecasting 22(3), 443–473 (1 2006), `https://doi.org/10.1016/j.ijforecast.2006.01.001`
7. Ding, X., Zhang, Y., Liu, T., Duan, J.: Deep learning for event-driven stock prediction. Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (ICJAI) (2015), `http://ijcai.org/papers15/Papers/IJCAI15-329.pdf`
8. Gallo, C., Letizia, C., Stasio, G.: Artificial Neural Networks in Financial Modelling (2006), `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.114.2740&rep=rep1&type=pdf`

9. Gençay, R., Selçuk, F., Whitcher, B.: An introduction to wavelets and other filtering methods in finance and economics. Academic Press (2002)
10. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning (2016), `http://www.deeplearningbook.org`
11. Haykin, S.: Neural networks and learning machines (2009)
12. He, T.X., Nguyen, T.: Wavelet Analysis and Applications in Economics and Finance. Research & Reviews: Journal of Statistics and Mathematical Sciences 1(1) (10 2015)
13. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation 9(8), 1735–80 (1997), `http://www.mitpressjournals.org/doi/10.1162/neco.1997.9.8.1735`
14. Hyndman, R.J., Koehler, A.B.: and Business Statistics Another Look at Measures of Forecast Accuracy Another look at measures of forecast accuracy. International journal of forecasting 22(November), 679–688 (2005), `http://www.sciencedirect.com/science/article/pii/S0169207006000239%5Cnhttp://core.ac.uk/download/pdf/6340761.pdf`
15. Kamijo, K., Tanigawa, T.: Stock price pattern recognition-a recurrent neural network approach. International Joint Conference on Neural Networks pp. 215–221 (1990), `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5726532`
16. Krollner, B., Vanstone, B., Finnie, G.: Financial time series forecasting with machine learning techniques: A survey (2010), `http://works.bepress.com/bruce_vanstone/17/`
17. Li, X., Huang, X., Deng, X., Zhu, S.: Enhancing Quantitative Intra-day Stock Return Prediction by Integrating Both Market News and Stock Prices Information. Neurocomput. 142, 228–238 (2014), `http://dx.doi.org/10.1016/j.neucom.2014.04.043`
18. Marszałek, A., Burczyński, T.: Modeling and forecasting financial time series with ordered fuzzy candlesticks. Information Sciences 273, 144–155 (7 2014), `https://doi.org/10.1016/j.ins.2014.03.026`
19. Medsker, L., Jain, L.C.: Recurrent Neural Networks: Design and Applications. International Series on Computational Intelligence, CRC Press (1999)
20. Mills, T.C., Markellos, R.N.: The Econometric Modelling of Financial Time Series. Cambridge University Press, Cambridge (2008), `https://doi.org/10.1017/CBO9780511817380`
21. Preethi, G., Santhi, B.: Stock Market Forecasting Techniques: A Survey. Journal of Theoretical & Applied Information Technology (2012)
22. Schnader, M.H., Stekler, H.O.: Evaluating predictions of change. Journal of Business pp. 99–107 (1990)
23. Sureshkumar, K., Elango, N.: Performance analysis of stock price prediction using artificial neural network. Global journal of computer science and Technology 12 (2012), `http://computerresearch.org/index.php/computer/article/view/426`
24. Takeuchi, L., Lee, Y.: Applying Deep Learning to Enhance Momentum Trading Strategies in Stocks (2013)
25. Tsay, R.S.: Analysis of financial time series, vol. 543. John Wiley & Sons (2005)
26. Walnut, D.F.: An introduction to wavelet analysis. Birkhäuser (2002)
27. Wilder, J.W.: New Concepts in Technical Trading Systems. Trend Research (1978)
28. Yeh, S., Wang, C., Tsai, M.: Corporate Default Prediction via Deep Learning (2014), `http://teacher.utaipei.edu.tw/~cjwang/slides/ISF2014.pdf`