2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

Superiority of Neural Networks for Trading Volume Forecasts of Stocks and Cryptocurrencies

Sulalitha Bowala
Department of Statistics
University of Manitoba
Winnipeg, Canada
bowalams@myumanitoba.ca

Aerambamoorthy Thavaneswaran
Department of Statistics
University of Manitoba
Winnipeg, Canada
Aerambamoorthy.Thavaneswaran@umanitoba.ca

Ruppa Thulasiram
Department of Computer Science
University of Manitoba
Winnipeg, Canada
Tulsi.Thulasiram@umanitoba.ca

Md Erfanul Hoque
Department of Mathematics and Statistics
Thompson Rivers University
Kamloops, Canada
mhoque@tru.ca

Alex Paseka
Department of Accounting and Finance
University of Manitoba
Winnipeg, Canada
alex.paseka@umanitoba.ca

***Abstract*—Trading volume is an important variable to successfully capture market risks along with asset price/returns. Recently, there has been a growing interest in deep learning methods to forecast the trading volume of stocks using historical volatility as a feature. Unlike the existing work, a novel data-driven log volatility forecast is proposed in this paper as an extra feature to improve trading volume forecasts. Recently, neural networks for volatility and neural nets for electricity demand forecasting, constructed with `nnetar` function, have shown to be superior. The novelty of this paper is to demonstrate the neural network based on the `nnetar` function from the forecast package in R for trading volume forecast shows superiority over the other neural network.**

***Index Terms*—Cryptocurrencies, Neural Networks, Trading Volume**

I. Introduction

The total number of shares of a stock or a financial asset bought and sold within a specified period is known as trading volume. Trading volume helps to capture the overall volume and liquidity in the market for a given stock. Elevated trading volume frequently signifies a notable degree of interest and engagement among investors, whereas limited trading volume might imply reduced market interest or participation. For investors and traders, trading volume is an essential metric as it measures the significant impact of price movements on the stock/market. Accurate trading volume forecasts provide valuable insights into market dynamics, aiding investors, traders, regulators, and other stakeholders in making informed decisions and ensuring the smooth functioning of financial markets. Thus, forecasting trading volume in different markets has become a fascinating research topic among scholars.

Many researchers have incorporated the trading volume of stocks in different studies with different scopes. In most cases, the trading volume is considered an explanatory variable to obtain predictions/forecasts of risks/volatility and stock prices. [2] demonstrates that the daily trading volume substantially influences the variability in daily returns, suggesting a robust connection between trading volume and return volatility. [3] investigates trading volume and downside trading volume of the stock spot market and futures markets, and it can be used to predict the downside risk. Moreover, studies such as [4]–[6] inquire relation between trading volume and volatility/volatility forecasts using different models. These studies confirm there is a strong relation between the two variables, volatility and trading volume, and thus, in regression-type models, trading volume is an important variable to predict trading volatility and vice versa. It is important to note that there are fluctuations in the variance of asset (stock) returns as time progresses. [7] proposes that utilizing GARCH coefficient BS models is a suitable approach for capturing changing variances in data over time. Within the realm of academic writing, predictions regarding conditional volatility are derived by extracting the square root of the forecasted conditional variance. [8] highlights that this estimation's asymptotic variance is greater, rendering it an inefficient approach for acquiring volatility estimates. Within the context of this article, we adopt recently introduced data-driven exponentially weighted moving average (DDEWMA) volatility forecast models to directly generate forecasts for volatility (as opposed to variance) and use them as a new feature to obtain trading volume forecasts.

Neural networks started to get popular during the latter part of the 1980s. There was a lot of excitement about this new approach, but some of the excitement was a bit exaggerated. Researchers from fields like machine learning, mathematics, and statistics study the characteristics of neural networks, leading to enhancements in algorithms and the establishment of a more refined methodology. Support vector machines and boosting are two examples of the ways machine learns. However, neural networks have been identified as a better alternative because they could work more automatically. After 2010, neural networks came back with a new name,"deep learning," and new designs for how they operate (see [13] for more details).

The use of neural networks in trading volume is popular among scholars. In [1], authors designed a backpropagation (BP) NN to forecast monthly futures trading volume for the Winnipeg commodity exchange. Trading volume is predicted based on several independent variables, and the authors have concluded that neural networks can produce better forecasts against the naive model using the Theil U statistic and even outperform the autoregressive integrated moving average (ARIMA) model. [10] establish the BP NN model to predict the carbon trading price and carbon trading volume, and authors have shown that the model is effective. The authors of [11] created a combined prediction system that relies on artificial neural networks to estimate the daily trading volume of Bitcoin. They utilized two distinct types of artificial neural networks (radial basis function neural networks and generalized regression neural networks) to predict the Bitcoin trading volume. Through this combined predictive approach, the proposed system managed to decrease forecasting errors significantly. In a recent study conducted by [12], the focus was on determining the predictability of trading volume using its historical lags and establishing the level of model complexity required to provide precise predictions.

Different choices are available for setting up a neural network, and here in this study, we consider two popular approaches for time series data. The first approach uses the `keras` package which connects with the `tensorflow` packages to fit a neural network model, and we refer to this as the neural network (NN) in this study. The NN interfaces with optimized Python code to build a recurrent neural network (RNN). The second approach uses the `nnetar` function from the `forecast` package [14] in R to fit a neural network model, and we refer to this as nnetar for convenience. The nnetar network fits a neural network dynamic regression model $(p, P, k)_m$ model, where p and P are the autoregressive (AR) orders of the non-seasonal and seasonal parts, and k is the number of nodes in the hidden layer. This model has been used in many applications, such as electricity demand [16] and [17]. Using NN and nnetar network, we predict/forecast daily log trading volumes of four technological stocks: Apple (AAPL), Microsoft (MSFT), NVIDIA (NVDA), and Intel (INTC), and four cryptocurrencies: Bitcoin (BTC-USD), Ethereum (ETH-USD), Tether USDt (USDT-USD), and Binance Coin (BNB-USD) based on asset daily log returns and daily log volatility.

The rest of the paper is structured as outlined below. In Section II, we present the theories behind the DDEWMA volatility forecast and describe the neural network's architecture. Section III presents the results of our experiments. Lastly, we conclude with our final remarks in Section IV.

II. Methodology

A. Data-Driven EWMA volatility forecast

In Finance, the stock prices (price P_t, at time t) are modeled as a geometric Brownian motion, and log returns (r_t) of the stocks can be calculated using $log\,P_t - log\,P_{t-1}$. Research has revealed that log returns often deviate from the normal distribution, with the majority exhibiting a t distribution characterized by heavy tails in most instances (see [8] and [15] for more details). If the data are t-distributed, it is important to determine the degrees of freedom to make inferences, and [8] proposed a technique to determine appropriate degrees of freedom using the sign correlation.

Let X be a random variable and X follows a student's t distribution. The corresponding degrees of freedom (d.f.) ν can be computed by solving,

$$2\sqrt{\nu - 2} = (\nu - 1)\rho_X \mathrm{Beta}\left[\frac{\nu}{2}, \frac{1}{2}\right], \tag{1}$$

where ρ_X is sign correlation. The sign correlation of the random variable X with mean μ is defined as

$$\rho_X = \mathrm{Corr}(X - \mu, \mathrm{sign}(X - \mu)). \tag{2}$$

The data-driven algorithmic volatility estimator, in terms of log returns $r_1, \cdots, r_n$, is given as

$$\hat{\sigma}_r = \frac{1}{n}\sum_{t=1}^{n}\frac{|r_t - \bar{r}|}{\hat{\rho}_r}, \tag{3}$$

where $\hat{\rho}_r$ is the sample sign correlation of r_t which can be calculated using equation (2).

In this study, we obtain daily DDEWMA volatility forecasts using log-returns of the past three months for each asset. Thus, for stocks, 63-day rolling forecasts, and for cryptocurrencies, 90-day rolling forecasts are considered. An algorithm to obtain the DDEWMA volatility forecast is given in Algorithm 1.

Algorithm 1 Data-Driven EWMA volatility forecasts

Require: Data: adjusted closing price of stocks / cryptocurrencies P_t, $t = 1, \dots, n$

$r_t \leftarrow log\,P_t - log\,P_{t-1}$, $t = 1, \dots, n$
$\hat{\rho} = Corr(r - \bar{r}, sign(r - \bar{r}))$
$Z_t \leftarrow \frac{|r_t - \bar{r}|}{\hat{\rho}}$
$S_0 \leftarrow \bar{Z}$
α (smoothing paramter) $\leftarrow (0, 1)$
$S_t \leftarrow \alpha Z_t + (1 - \alpha)S_{t-1}$, $t = 1, \dots, n$
$\alpha_{opt} \leftarrow \min\ \sum_{t=k+1}^{n}(Z_t - S_{t-1})^2$
for $t \leftarrow 1, \dots, n$ **do**
    $S_t = \alpha_{opt} Z_t + (1 - \alpha_{opt})S_{t-1}$
return S_n

B. Neural Networks

A neural network is a powerful tool to predict any nonlinear real function on a bounded domain with high accuracy. The fundamental form of a neural network is referred to as a feed-forward neural network. It is composed of an input unit responsible for processing input variables, succeeded by numerous interconnected hidden layers and building up to an output layer. The transition from one layer to the next is characterized by nonlinear functions (e.g., Rectified Linear Unit (ReLU), Sigmoid, and Hyperbolic Tangent (tanh)).

Neural networks exhibit distinctions from conventional time series forecasting models employed in finance. They do not require extensive parameter tuning, and achieving a universal approximate solution in a neural network does not mandate the optimization of all parameters. In an autoregression model, lagged values are used as inputs. Similarly, lagged values of time series can be used as inputs in a time series neural network model. Also, if the lagged values of the target variable $(y_{t-1}, y_{t-2}, \ldots, y_{t-p})$ depend on the lagged values of some features $(x_{1,t}, \ldots, x_{L,t})$ they also can be used as inputs in a neural network model. In this study, the target variable is the log trading volume. The neural network is trained with lag values of log trading volume, log returns, and log volatility. Once the model is trained, log returns and log volatility forecasts (DDEWMA volatility forecasts) from the testing data are used to obtain log trading volume forecasts.

A general representation of a neural network with p number of inputs, one hidden layer, and one output is given in Figure 1. However, more complicated neural networks can be formed with multiple hidden layers with many neurons (hidden layer nodes) and several outputs.

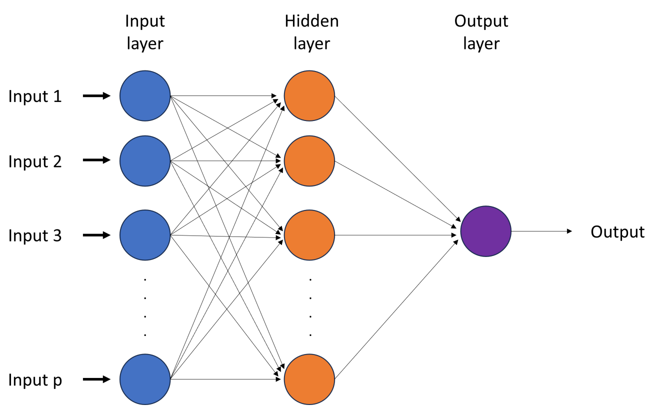

Fig. 1. Illustration of a feed-forward neural network

Similar to how it works in other multilayer feed-forward networks, in the case of the `nnetar` function, inputs are received from the preceding layers. The outputs generated by the nodes within a specific layer serve as the inputs for the subsequent layer. Each node's inputs are integrated through a weighted linear combination, which is subsequently transformed by a nonlinear function before being produced as output. The inputs into each hidden neuron are combined linearly to give

$$z_j = b_j + \sum_{i=1}^{p} w_{i,j} x_i.$$

Within the hidden layer, this is subsequently altered by applying a nonlinear function, like a sigmoid,

$$s(z) = \frac{1}{1 + e^{-z}},$$

to provide input for the subsequent layer, which helps mitigate the impact of exceptionally high or low input values, consequently enhancing the network's resilience against outliers.

The values of the parameters $b_1, b_2, \ldots, b_q$, (q number of neurons in the hidden layer) and $w_{1,1}, \ldots, w_{p,q}$ are determined through a process of learning or estimation based on the data. Typically, measures are applied to limit the magnitude of the weight values to prevent them from becoming excessively large. This controlling parameter for the weights is referred to as the "decay parameter," and it is frequently set to a value of 0.1.

Initially, the weights are assigned random values, which are subsequently adjusted based on the available data. As a result, neural networks introduce an element of unpredictability into their predictions. To address this inherent randomness, the network is typically trained multiple times, starting from different random initial values for the weights, and the outcomes are then averaged. Furthermore, it is essential to predefine the number of hidden layers and the number of nodes within each hidden layer before training the network.

III. Experimental Results

In this section, we investigate the performance of the NN and nnetar networks. Trading volumes of the four stocks (Apple, Microsoft, NVIDIA, and Intel) and four cryptocurrencies (Bitcoin, Ethereum, Tether, and Binance Coin) are obtained along with their adjusted closing prices, and daily volatility forecasts are obtained from DDEWMA volatility forecast using the Algorithm 1. The study period for this work is from 2022-01-01 to 2022-12-31, and all the data are collected from Yahoo! Finance. Stocks are chosen based on their popularity during the study period, while cryptocurrencies are selected according to their market capitalization as per Coinmarketcap. The downloadable data encompass various attributes such as opening, high, low, closing prices, adjusted prices, and daily trading volumes for both stocks and cryptocurrencies. In this research, we specifically employ the daily adjusted price, which is a modified version of the daily asset price, to compute logarithmic returns. The networks are trained with 75% of the observations, and lag values of daily log trading volume, daily log returns, and daily log volatility are used as inputs of the networks. The remaining 25% of the observations are used to evaluate model performances, and two performance evaluation metrics are considered in this study. The mean square error (MSE) and mean absolute deviation (MAD) of the daily log trading volume forecasts/predictions during the testing period from two networks are computed, and the model with the lowest MSE and MAD is considered the superior model.

When using the NN, the network needs to feed with lag values of the variables decided by the user. However, the nnetar network does not need to provide lag values from the user, and the function itself is capable of deciding how many lag values of the target variable need to be considered. It is essential to emphasize that the nnetar network generates a feed-forward neural network with one hidden layer and past input data for predicting univariate time series. In contrast, we have the opportunity to construct more complicated networks with multiple hidden layers and several outputs with NN.

First, the networks are constructed using the NN for all the stocks and cryptocurrencies. All the networks have one hidden layer with twelve neurons (twelve hidden layer nodes).

However, different lag values are used for each stock and cryptocurrency in the input layer. Lag values for each stock and cryptocurrency are decided using ACF (Autocorrelation Function) plots of daily log trading volumes for the study period, and corresponding ACF plots for stocks and cryptocurrencies are given in Figure 2 and Figure 3, respectively. Furthermore, in all the networks, 10% dropout is applied to the input data, and 10% dropout is applied to the recurrent connections within the RNN layer for regularization. Also, when training the models, the `RMSprop` optimization algorithm (see [18] for more details) is used to adjust the learning rate and mean square error chosen as the loss function.

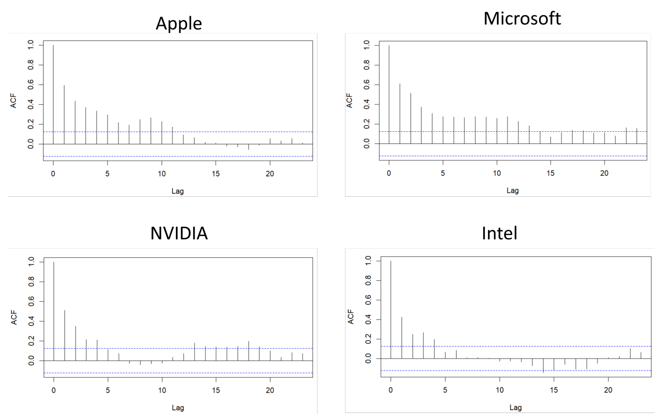

Fig. 2. Autocorrelation Function plot of Trading Volume - Stocks

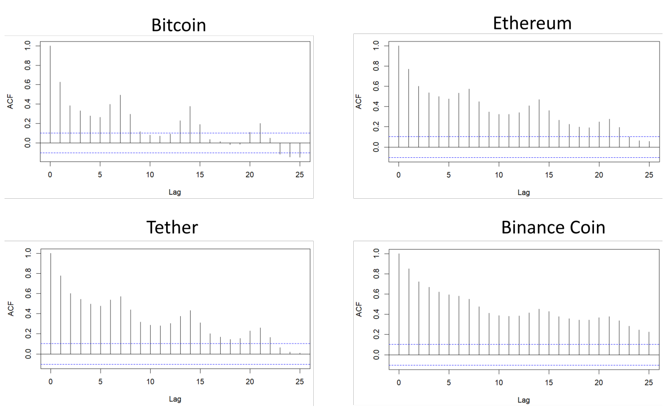

Fig. 3. Autocorrelation Function plot of Trading Volume - Cryptocurrencies

Second, we construct networks from `nnetar`. The target variable or the target time series in the networks is daily log trading volume, and daily log return and volatility are used as exogenous variables. The parameter specifies the regularization strength for the model is set to auto, and thus, the function will automatically determine an appropriate value for the regularization parameter. The rate at which the weights of the neural network model decay is 10% and the summary of the best-fitted model for training data of each asset is provided in Table I. Observe that the model suggests different lag values to be considered with different neurons in the hidden layer for stocks. However, for cryptocurrencies, the model suggests nine neurons in the hidden layer with the last fifteen observations of the target variable (fifteen lag values of daily log trading volume) $y_{t-1}, \dots, y_{t-15}$ to forecast the target y_t (daily log trading volume at time t) for all the cryptocurrencies.

TABLE I
SUMMARY OF BEST-FITTED MODELS USING `nnetar`

Asset	lag values of target variable (p) $(y_{t-1}, \dots, y_{t-p})$	Number of neurons in the hidden layer	$\hat{\sigma}^2$
Apple	4	4	4.20×10^{-07}
MSFT	2	2	7.96×10^{-07}
NVIDIA	1	2	5.97×10^{-07}
Intel	3	3	9.75×10^{-07}
Bitcoin	15	9	3.96×10^{-08}
Ethereum	15	9	3.31×10^{-03}
Tether	15	9	3.35×10^{-08}
Binance Coin	15	9	8.59×10^{-08}

Once the networks are trained, daily log trading volume forecasts for the testing period can be obtained (Figure 4 and Figure 5). Then, using actual daily log trading volumes and forecasts of daily log trading volumes, MSE and MAD are calculated. Results using NN and nnetar network are summarized in Table II. It can be seen from the table that for all the stocks and cryptocurrencies, MSE and MAD using nnetar network are lower than MSE and MAD computed using the NN. This indicates networks constructed with the nnetar network lead to better predictions/forecasts of daily log trading volume.

TABLE II
MSE AND MAD OF DAILY LOG TRADING VOLUMES FORECASTS USING NN WITH ONE LAYER AND `nnetar`

	NN		nnetar	
Asset	MSE	MAD	MSE	MAD
Apple	0.8992	0.7515	0.2064	0.4087
Microsoft	0.9212	0.7002	0.1444	0.3012
NVIDIA	1.0939	0.8031	0.0826	0.2160
Intel	1.2504	0.7736	0.1286	0.2719
Bitcoin	0.6970	0.6401	0.1962	0.3483
Ethereum	1.2455	0.9289	0.6110	0.6705
Tether	0.8159	0.7407	0.5046	0.6143
Binance Coin	0.5758	0.6196	0.2664	0.3862

There is no strict rule for determining the exact number of hidden layers and neurons in a neural network that will work optimally for all tasks. The optimal architecture depends on various factors, including the complexity of the problem, the available data, and the computational resources. Increasing the number of hidden layers and neurons in a neural network can potentially improve its performance and accuracy. Thus, a complex neural network (a network with several hidden layers, and each layer has more neurons) may help to improve the forecasting ability of trading volumes. However, it is important to remember that complex networks do not always guarantee better results. Adding too many layers or neurons can lead to overfitting, and as the network memorizes the training data, it may fail to generalize well to new, unseen data. Also, it increases complexity and reduces interpretability while

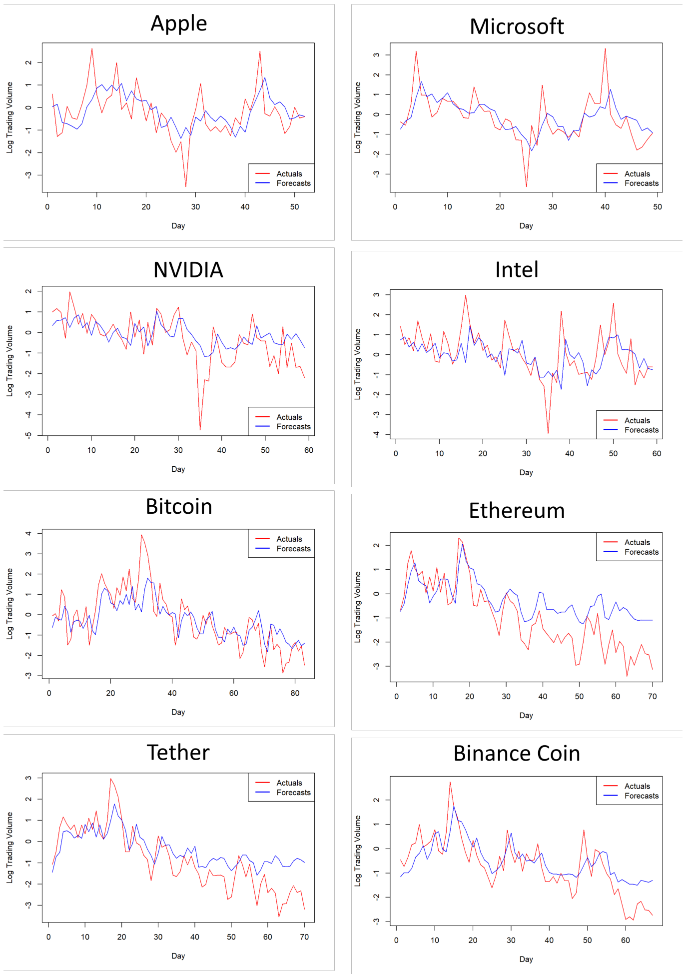

Fig. 4. Log Trading Volume Forecasts using NN with One Layer

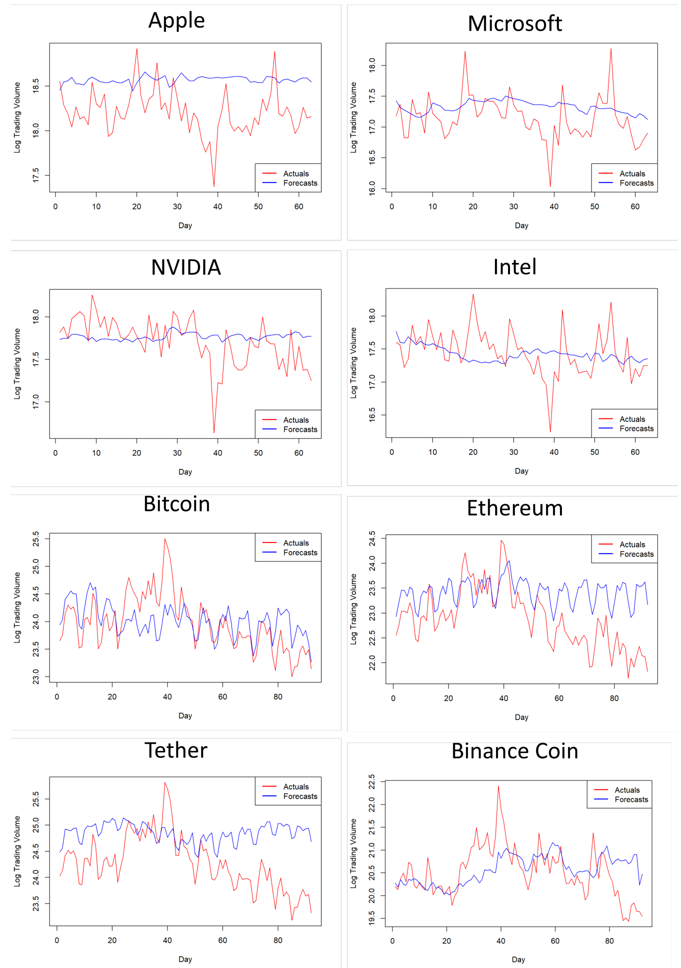

Fig. 5. Log Trading Volume Forecasts using `nnetar`

demanding more resources to implement and run the model. Nonetheless, we introduce another layer and twice as many neurons to the neural network constructed using NN. The new model is trained with the same training data, and daily log trading volume forecasts are obtained for the same testing data (Figure 6). The MSE and MAD are summarized in Table III. Observe that both MSE and MAD values have increased for all the stocks. However, among the cryptocurrencies, Ethereum and Tether show lower MSE and MAD, and For Bitcoin and Binance Coin, both MSE and MAD have increased. This indicates some improvements can be achieved with complex NNs for selected cases when forecasting trading volumes.

TABLE III
MSE AND MAD OF DAILY LOG TRADING VOLUMES FORECASTS USING NN WITH TWO LAYERS

Asset	MSE	MAD
Apple	0.9545	0.8218
Microsoft	1.0282	0.7483
NVIDIA	1.1291	0.8155
Intel	1.3951	0.8647
Bitcoin	0.8366	0.7110
Ethereum	0.9197	0.7958
Tether	0.6964	0.6886
Binance Coin	0.7281	0.6700

IV. CONCLUSIONS

Recently superiority of the `nnetar` neural network dynamic regression models for electricity demand forecasting has been demonstrated. The trading volume forecasts play a crucial role in measuring the substantial influence of price movements on stocks and cryptocurrencies. In this paper, the driving idea, unlike the existing work, is demonstrating the superiority of the trading volume forecasts using `nnetar` function form `forecast` package in R over the neural networks using `keras` and `tensorflow` packages. Moreover, this paper considers the extra features such as data-driven log volatility forecasts and log returns to improve trading volume forecasts. The experimental results show that trading volume forecasts using the nnetar network are superior to the `keras` neural network forecasts.

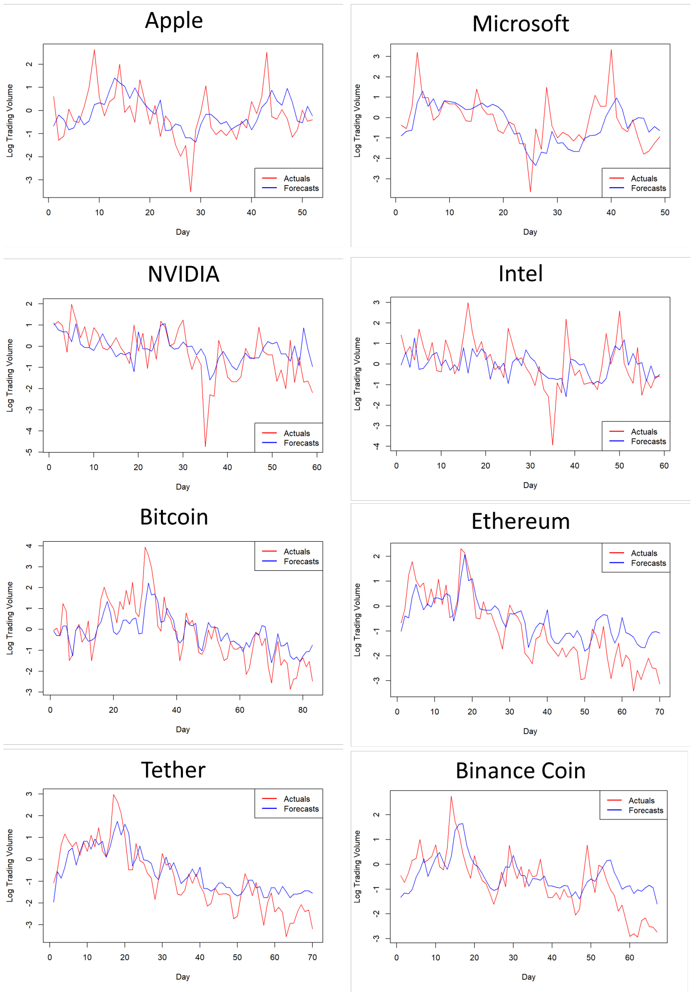

Fig. 6. Log Trading Volume Forecasts using NN with two Layers

Acknowledgment

The first author acknowledges the University of Manitoba Graduate Fellowship (UMGF). The second and third authors acknowledge the Natural Sciences and Engineering Research Council (NSERC) of Canada. The authors are thankful to the three reviewers for their comments and suggestions which have resulted in an improved version of this manuscript.

References

[1] Kaastra, I., and Boyd, M. S. (1995). Forecasting futures trading volume using neural networks. The Journal of Futures Markets (1986-1998), 15(18), 953.

[2] Lamoureux, C. G., and Lastrapes, W. D. (1990). Heteroskedasticity in stock return data: Volume versus GARCH effects. The journal of finance, 45(1), 221-229.

[3] He, Z., Huang, C., Gong, X., Yang, X., and Wen, F. (2017). Do trading volume and downside trading volume help forecast the downside risk?. EURASIA Journal of Mathematics, Science and Technology Education, 13(12), 8367-8382.

[4] Le, V., and Zurbruegg, R. (2010). The role of trading volume in volatility forecasting. Journal of International Financial Markets, Institutions and Money, 20(5), 533-555.

[5] Chiang, T. C., Qiao, Z., and Wong, W. K. (2010). New evidence on the relation between return volatility and trading volume. Journal of Forecasting, 29(5), 502-515.

[6] Wang, X., Wu, C., and Xu, W. (2015). Volatility forecasting: The role of lunch-break returns, overnight returns, trading volume and leverage effects. International Journal of Forecasting, 31(3), 609-619.

[7] Hao, J., and Zhang, J. E. (2013). GARCH option pricing models, the CBOE VIX, and variance risk premium. Journal of Financial Econometrics, 11(3), 556-580.

[8] Thavaneswaran, A., Paseka, A., and Frank, J. (2020). Generalized value at risk forecasting. Communications in Statistics-Theory and Methods, 49(20), 4988-4995.

[9] Thavaneswaran, A., Thulasiram, R. K., Zhu, Z., Hoque, M. E., and Ravishanker, N. (2019, July). Fuzzy value-at-risk forecasts using a novel data-driven neuro volatility predictive model. In 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC) (Vol. 2, pp. 221-226). IEEE.

[10] Zhiyuan, L., and Zongdi, S. (2017). The carbon trading price and trading volume forecast in Shanghai city by BP neural network. International Journal of Economics and Management Engineering, 11(3), 628-634.

[11] Lahmiri, S., Saade, R. G., Morin, D., and Nebebe, F. (2020, November). An artificial neural networks based ensemble system to forecast bitcoin daily trading volume. In 2020 5th International Conference on Cloud Computing and Artificial Intelligence: Technologies and Applications (CloudTech) (pp. 1-4). IEEE.

[12] Xu, X., and Zhang, Y. (2023). A high-frequency trading volume prediction model using neural networks. Decision Analytics Journal, 7, 100235.

[13] James, G., Witten, D., Hastie, T., and Tibshirani, R. (2021). An introduction to statistical learning with Applications in R (Vol. 112, p. 18). New York: springer.

[14] Hyndman, R. J., Athanasopoulos, G., Bergmeir, C., Caceres, G., Chhay, L., O'Hara-Wild, M., ... and Wang, E. (2020). Package 'forecast'. Online] https://cran. r-project. org/web/packages/forecast/forecast. pdf.

[15] Bowala, S., Singh, J., Thavaneswaran, A., Thulasiram, R., and Mandal, S. (2022, May). Comparison of Fuzzy Risk Forecast Intervals for Cryptocurrencies. In 2022 IEEE Symposium on Computational Intelligence for Financial Engineering and Economics (CIFEr) (pp. 1-8). IEEE.

[16] Liang, Y., and Thavaneswaran, A. (2022, June). Long Term Interval Forecasts of Demand using Data-Driven Dynamic Regression Models. In 2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC) (pp. 250-259). IEEE.

[17] Bowala, S., Makhan, M., Liang, Y., Thavaneswaran, A., and Appadoo, S. S. (2022, September). Superiority of the Neural Network Dynamic Regression Models for Ontario Electricity Demand Forecasting. In 2022 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE) (pp. 182-187). IEEE.

[18] Hinton, G., Srivastava, N., and Swersky, K. (2012). COURSERA: Neural networks for machine learning–lecture 6a.