REALIZED VOLATILITY FORECASTING
USING HYBRID NEURAL NETWORKS:
AN APPLICATION FOR THE ISTANBUL STOCK EXCHANGE

MEHMET EKİN GÜLTEKİN

BOĞAZİÇİ UNIVERSITY

2023

REALIZED VOLATILITY FORECASTING
USING HYBRID NEURAL NETWORKS:
AN APPLICATION FOR THE ISTANBUL STOCK EXCHANGE

Thesis submitted to the

Institute for Graduate Studies in Social Sciences

in partial fulfillment of the requirement for the degree of

Master of Arts

In

Business Information Systems

by

Mehmet Ekin Gültekin

Boğaziçi University

2023

DECLARATION OF ORIGINALITY

I, Mehmet Ekin Gültekin, certify that

- I am the sole author of this thesis and that I have fully acknowledged and documented in my thesis all sources of ideas and words, including digital resources, which have been produced or published by another person or institution;
- this thesis contains no material that has been submitted or accepted for a degree or diploma in any other educational institution;
- this is a true copy of the thesis approved by my advisor and thesis committee at Boğaziçi University, including final revisions required by them.

Signature ______________________________

Date ______________________________

ABSTRACT

Realized Volatility Forecasting Using Hybrid Neural Networks:

An Application for The Istanbul Stock Exchange

Volatility forecasting in the financial markets is important in the areas of risk management and asset pricing, among others. In this study, BIST 100's 1-day, 5-day, and 10-day-ahead return volatilities are examined. Two types of hybrid models are utilized to improve individual GARCH-family models' predictions. For the first hybrid model, a group of GARCH-family models is constructed to produce volatility estimates which were then fed into neural network to increase the predictive power. The second hybrid model received GARCH-family models' specifications instead of volatility estimates as inputs for ANN to conduct the learning process. Hybrid neural networks were also fed a set of exogenous, endogenous, and dummy variables. One of the main conclusions is that both hybrid models increased the forecasting precision of individual GARCH-family models while the second hybrid model provided better volatility forecasts for all error measures used in this study. Equal forecast accuracy test also showed that the hybrid models' out-of-sample predictions were significantly better than GARCH-family methods. All model performances deteriorated as forecast horizon was extended, although the steepest decline happened for hybrid models rather than the GARCH-family. Lastly, as the complexity of the neural network architecture was increased, the loss measures for the out-of-sample forecasts improved except on the last case where the network overfitted using the highest number of neurons per hidden layer among the searched hyperparameter grid.

ÖZET

Hibrit Sinir Ağı Modelleri Kullanarak Volatilite Tahmini: Borsa İstanbul İçin Bir Uygulama

Finansal piyasalardaki volatilite tahmini, risk yönetimi ve varlık fiyatlandırma gibi alanlarda büyük önem taşır. Bu çalışmada, BIST 100'ün 1 gün, 5 gün ve 10 gün sonrası getiri volatilitesi incelenmiştir. GARCH ailesi modellerinin tahminlerini iyileştirmek için iki tür hibrit model kullanılmıştır. İlk hibrit model, GARCH ailesi modellerinin volatilite tahminlerini girdi olarak kullanırken ikinci hibrit model ise sinir ağlarının öğrenme sürecini yürütmesi için girdi olarak volatilite tahminleri yerine GARCH ailesi modellerinin spesifikasyonlarını almıştır. Hibrit sinir ağları ayrıca bir dizi değişkenle beslenmiştir. Bu değişkenler yabancı fiyat endeksleri, teknik analiz ve kukla değişkenlerden oluşmuştur. Ana sonuçlardan biri, her iki hibrit modelin de GARCH ailesi modellerinin tahmin kesinliğini arttırdığı, ikinci hibrit modelin ise bu çalışmada kullanılan tüm hata ölçütleri için daha iyi volatilite tahminleri sağladığıdır. Eşit tahmin doğruluk testi ayrıca hibrit modellerin örneklem dışı tahminlerinin GARCH ailesi yöntemlerinden istatistiksel olarak anlamlı ölçüde daha iyi olduğunu göstermiştir. Tahmin ufku genişledikçe tüm model performansları kötüleşirken, en keskin düşüş GARCH ailesinden ziyade hibrit modellerde yaşanmıştır. Son olarak, sinir ağı mimarisinin karmaşıklığı arttıkça hibrit modellerin sonuçları genel olarak iyileşirken, gizli katman başına kullanılan nöron sayısı en yükseğe çıktığında modellerin örneklem içi gözlemleri ezberlemeye başladığı görülmüştür.

TABLE OF CONTENTS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Theoretical and empirical research in academia has long focused on forecasting and modelling stock market volatility. Volatility can be conceptualized as a measure of variation in the price of a financial security relative to its expected value. Financial market investment decisions are also heavily influenced by asset volatility and forecasts of expected returns. Volatility serves as one of the key input parameters for a variety of financial applications such as option pricing, portfolio allocation and risk management. Beginning with Black and Scholes (1973), modern option pricing theory places volatility at the center of calculating the fair value of an option or any derivative instrument having option features. Return volatility of the underlying asset in the option pricing formula is the only parameter that cannot be observed directly which boosts its significance. The other parameters in the option pricing formula, namely the interest rate, strike price, time to expiration and underlying stock price are all known or easily obtainable from the market, however volatility needs to be forecasted.

ARCH and GARCH models were introduced by Engle (1982) and Bollerslev (1986), respectively to model for situations with changing variance. Such times series models with heteroscedastic errors are applicable to modeling highly volatile financial market data even though, many financial time series have non-linear dependency structure. Since GARCH-family models assume a linear correlation structure among the time series data, they may not capture such nonlinear patterns. Nonparametric models estimated by various methods such as ANNs (Artificial Neural Networks), can be fit on a data set much better than linear models. Therefore, first objective of this study is to examine whether this volatility can be captured using the different GARCH-family models. Subsequently, the best model forecasts are used as inputs, together with a set of exogenous and endogenous variables in feed-forward neural networks, specifically MLP (multilayer perceptron), which seeks to improve upon previous results. Another hybridization technique used in this study is to extract variables from different GARCH-family models and feed these series to the network instead of the conditional volatility estimates. Thus, these two approaches of hybridizing GARCH-family models and neural networks will be compared. The main objective is to determine whether there is an improvement in volatility predictions based on the returns of BIST 100, using hybrid ANN models against GARCH-family models. The results of this study are crucial to formulate investment strategies, asset pricing modelling, asset allocation, and option pricing.

There are a total of 6 sections in this study. Section 2 presents an overview of existing literature on modelling and forecasting return volatility. Section 3 gives specifications of individual GARCH-family and hybrid models proposed to produce out-of-sample predictions. Section 4 focuses on the dataset used and characteristics of individual input variables. Section 5 presents the results obtained and Section 6 concludes the study and provides insight on how the findings of this study can be built upon.

CHAPTER 2

LITERATURE REVIEW

The motivation of this study is to improve upon the conditional volatility estimates of GARCH-family models to forecast Istanbul stock exchange's return volatility by constructing hybrid models using artificial neural networks. Additional robustness checks are done based on employing different forecast horizons and different neural network architectures. At the end, models will be compared to select the ones that perform better in a formal statistical sense by employing the Diebold-Mariano test.

Volatility forecasting has crucial applications in financial markets. Mainly, investors are interested in the risk of their portfolios' return. In addition, volatility is a crucial input to asset pricing models. As a result, researchers have used many models to forecast financial volatility however the most used models in the literature are ARCH (Autoregressive Conditional Heteroskedasticity) proposed by Engle in 1982 and the generalized ARCH by Bollerslev in 1986. Volatility is proven to show three characteristics which are volatility clustering first observed by Mandelbrot (1963), asymmetric reaction to past return shocks that is first documented by Black (1976) and Christie (1982), and nonlinearity presented by Engle and Lee (1993). Studies have shown that the GARCH model fails to consider this asymmetric, negative correlation between future returns and volatility. In other words, a negative effect, meaning a falling market against investor expectations, has a bigger influence on future return volatility than a same-magnitude positive effect. In the literature, this asymmetric shock is often called the leverage effect. Following this finding, Nelson has introduced exponential GARCH in 1991 and Glosten, Jagannathan and Runkle have designed sign GARCH, often referred as GJR-GARCH, in 1993 that aim to capture asymmetric volatility effects. Since then, these approaches pioneered the significant improvements in forecasting accuracy achieved by researchers in the modelling of time series.

Researchers have proved that improved forecast performance can be obtained by the integration of Artificial Neural Networks through hybrid models. Hybrid models are constructed in two different ways in the literature. The first way is to feed the conditional volatility predictions of the GARCH-family model to the neural network through the input layer. The second approach is to let the neural network learn the conditional volatility pattern by feeding the GARCH-family model specifications of into the network.

Donaldson and Kamstra (1997) showed that ANN can capture volatility effects that are overlooked by different variations of GARCH models by exploring its forecasting ability for major stock exchanges in London, New York, Tokyo, and Toronto. They modelled the mean return as an AR(1) process while GARCH, EGARCH and GJR-GARCH were used as benchmark models. They chose the optimal number of lagged terms according to BIC. They produce recursively generated one-step-ahead predictions and highlight that GJR-GARCH is the only GARCH-family model which removes all evidence of asymmetricity for every index and the asymmetry parameter is significantly positive for three of the four indices which signals that negative errors lead to more volatility than positive errors. Moreover, they report that the hybrid model outperforms all benchmark models, and it avoids overfitting the data since its excess kurtosis of standardized returns is the lowest for three of the four stock indices. Forecast encompassing tests also signal that the hybrid model performs better than traditional GARCH-family models.

Bildirici and Ersin (2009) enhanced ANN with ARCH-family models to evaluate the volatility of daily returns from the Istanbul Stock Exchange. The study includes several ARCH-family models such as GARCH, EGARCH, GJR-GARCH, TGARCH, NGARCH, SAGARCH, PGARCH, NPGARCH hybridized with ANN models. They use early stopping to avoid overfitting the hybrid model. The greatest improvement is reported with ANN-TGARCH hybrid model compared to TGARCH-only model. They also compared the percentage increase in RMSE value at different forecast horizons. All hybrid models reported lower RMSE values than their GARCH counterparts at every forecast horizon. The highest increase in RMSE is realized at $t + 6$ days-ahead forecasts while the accuracy of forecasts follows a stable path until $t + 161$ days.

Hajizadeh, Seifi, Zarandi and Turksen (2012) propose two hybrid models that are based on GARCH and ANN to forecast the return volatility of the S&P 500 index. Their set of explanatory variables consisted of 9 variables which are chosen among 14 total endogenous and exogenous variables by examining their correlations with estimated volatility based on their benchmark GARCH model. To construct the hybrid models, they first evaluated different lags for each GARCH model by using AIC and BIC. They found that the best forecasting result is provided by EGARCH(3, 3). For the first hybrid model, GARCH model's conditional volatility estimates are re-learnt in ANN models along with the selected set of explanatory variables. For the second hybrid model, they utilized several simulated EGARCH series instead of just one to train the ANN model. They report that the reasoning behind feeding several synthetic series to ANN is to introduce the autocorrelation structure of data to the network. Their results show that both hybrid models outperform benchmark EGARCH model, with second hybrid model performing the best due to inclusion of additional simulated series as extra inputs.

Kristjanpoller, Fadic and Minutolo (2014) performed realized volatility forecasting for Brazil, Chile, and Mexico's stock exchange indices. Future realized volatility is forecasted through a GARCH(1, 1) model with an AR(1) mean model hybridized using an artificial neural network. The input variables for the hybrid model were volatility at time t-22, GARCH model's forecasts, squared index log return which is included under the assumption that the GARCH model does not capture all the information from this variable, and the return of the index which is included as a possible proxy for volatility estimation. Authors also performed robustness checks in terms of altering the size of the training set for the neural network and the network structure. They report that increasing the number of layers had a negative impact for the neural network. Furthermore, best volatility forecasts were achieved through using 70% and 80% training set size. However, the highly volatile Brazilian stock index showed the best forecasts at 60% - 70%. They state that increasing training set size has a limit when dealing with highly volatile markets. It is concluded that ANN models can improve the forecast accuracy of GARCH models.

Kristjanpoller and Minutolo (2016) examined whether oil price volatility can be better forecasted through a hybrid neural network-GARCH model while incorporating additional explanatory variables. Their benchmark models to compare the hybrid model were selected to be ARFIMA and GARCH(1, 1). For the benchmark GARCH model, they utilized a moving window length of 252 days back which corresponds to approximately one year of transactions. They also used an AR(1) mean model to produce the residuals needed estimate GARCH parameters. The explanatory variables for the hybrid model were GARCH forecast estimates, the squared oil price return, the financial time series of the daily variations for Euro/US Dollar and US Dollar/Yen exchange rates as well as the stock market returns of DIJA (Dow Jones Industrial Average) and FTSE (Financial Times Stock Exchange). They used HMSE and HMAE as loss functions and conclude that hybrid model increased the volatility forecast precision by at least 30% over benchmark models as measured by HMSE.

Kristjanpoller and Hernández (2017) analyzed volatility forecasts of gold, silver, and copper. Different GARCH-family models were used to forecast volatility by including explanatory variables like the US Dollar/Euro and US Dollar/Yen exchange rates, the oil price, and the Chinese, Indian, British, and American stock market indices. They also included all explanatory variables for the mean return series of GARCH models. These conditional volatility estimates were then fed to neural networks to create a hybrid model. Models were adjusted at each step using a 252-day rolling window. Best model specifications for each GARCH-family model were selected through AIC. Once model specifications were determined, 14-, 21- and 28-day forecasts were produced. They conclude that the hybrid model improved the out-of-sample forecasting power.

Monfared and Enke (2014) forecasted return volatility of Nasdaq Composite Stock Index by using Neural Network and GJR-GARCH models. Their explanatory dataset included daily returns and variances of ten NASDAQ indices which were chosen to cover the entire economy and import as much information as possible into the hybrid model. They split the entire period into 4 sections to test hybrid model's performance under different economic situations. The hybrid network is trained and tested based on forecast estimates of GJR-GARCH model. A total of 12 models were trained which consisted of three different networks for four sub-sections of the dataset. The different networks were feed-forward, generalized regression, and radial basis function. These models are tested by forecasting the variance of Nasdaq Composite Index over the next

44 trading days which approximately represents two calendar months. Their conclusion was that hybrid models were better at forecasting extreme events as the volatility structure becomes increasingly complex. Traditional econometric models failed to capture such structural changes. Moreover, Monfared and Enke (2014) adds that neural networks need not be used in low volatility periods for forecasting purposes since they are unnecessarily complex. A possible explanation is that the hybrid model trains on post-crisis, high-volatility conditions so that high volatility is memorized and projected onto consequent periods which causes an overestimation of volatility.

Hyup Roh (2007) demonstrates the usefulness of hybrid models to forecast the volatility of KOSPI (Korea Composite Stock Price Index) 200 in terms of its deviation and direction. Hyup Roh (2007) constructs hybrid models by learning the conditional volatility pattern in ANN through inputting the variables obtained from traditional models. As for traditional models, EWMA (Exponential Weighted Moving Average), GARCH and EGARCH models are used. The relative contribution factors of input variables are calculated based on their coefficients. It is showcased that variables extracted from GARCH-family models contributed the most. Macroeconomic variables such as 3-month and 1-year government bond prices and yields, contributed more than endogenous variables like KOSPI 200’s yield, squared yield, price at previous period and volume. Hyup Roh (2007) found that ANN model hybridized using EGARCH and GARCH performed the better than sole ANN for deviation in terms of MAE (Mean Absolute Error). Surprisingly, sole ANN performed better than ANN hybridized with EWMA. A direction comparison is done through hit ratio which resulted in hybrid ANN models with GARCH and EGARCH increasing the hit ratio compared to sole ANN.

Kim and Won (2018) combined an LSTM model with multiple GARCH-family models to predict KOSPI 200's return volatility. As for their explanatory variables, they use the index price, log difference return rate and its lagged version, the 3-year KTB (Korean Treasury Bills) interest rate, the interest rate for the 3-year AA-grade corporate bond, the price of crude oil and the price of gold. The benchmark models to compare hybrid LSTM/GARCH models are GARCH, EGARCH, EWMA (Exponentially Weighted Moving Average). Kim and Won (2018) constructs hybrid models by feeding specifications of GARCH-family models as inputs to the network rather than feeding the estimates. In that way, they also hybridize multiple GARCH-family models with LSTM under the same model. They train the models by feeding the explanatory variables starting from 22 trading days prior to produce one-day-ahead predictions. They also performed Diebold-Mariano test to verify the equivalence of forecast accuracy for two competing models. Their results show that neural network models combining two or more GARCH-family models produced significantly improved predictions over hybrid networks with just one GARCH-family model. The lower out-of-sample prediction error was achieved by the LSTM model hybridizing GARCH, EGARCH and EWMA. They also checked prediction accuracies for different lengths of training windows and forecast horizons. They found that the error increases as training window length shortens and/or forecast horizons widens. They also report that hybrid models based with LSTM has significantly lower prediction errors than feed-forward based hybrid models.

ANNs allow approximating arbitrarily well a wide class of linear and nonlinear functions without knowing the data generating process. Furthermore, ANNs are found to be particularly useful to forecast volatile financial variables exhibiting nonlinear dependence, such as stock prices, exchange rates and realized volatility. Both feed-forward and recurrent neural networks have been utilized in the literature. While recurrent networks have the advantage of preserving long memory of financial time series over feed-forward networks, both are prone to overfitting. Thus, different regularization techniques and strands of recurrent networks are used to overcome this problem.

Bucci (2019) compares different architectures of neural networks in predicting realized volatility of the S&P 500 index return. Neural network architectures employed in this study are feed-forward, Elman, Jordan, LSTM (Long short-term memory) and NAR (Nonlinear Autoregressive) neural networks. Bucci (2019) employs macroeconomic and financial variables to be fed to neural networks. Bucci (2019) states that feed-forward networks fail to capture the temporal dependence in the information set. As for recurrent networks, Elman networks has additional input neurons which are fed back from the hidden layer while Jordan networks has feedback from the output layer. These networks suffer from the so-called "vanishing gradient" problem. To overcome this, LSTMs and NAR neural networks were utilized. To reduce the number of weights to be trained by these networks, a subset of explanatory variables is selected based on LASSO (Least Absolute Shrinkage and Selection Operator) regression. The optimum number of neurons in the hidden layer were found through evaluating the performance of each network with an increasing the number of neurons and checking for lowest training MSE. Out-of-sample accuracy was assessed through MSE and QLIKE (Quasi-likelihood) and the predictive performances of different models were assessed using MCS (Model Confidence Set) and DM (Diebold-Mariano) test. One-step-ahead out-of-sample forecasts were generated from a rolling window scheme while re-estimating the parameters at each iteration. 5-step-ahead forecasts were also iteratively produced while updating the information with the prediction of the previous step. Results show that all neural network models outperformed traditional models like ARFIMA (Autoregressive fractionally integrated moving average model) and LSTAR (Logistic smooth transition autoregressive model). Moreover, including additional explanatory variables improved the forecasting accuracy. Best results were retrieved by LSTM and NAR neural networks with inclusion of explanatory variables. Lastly, recurrent networks also outperformed all others in multi-step ahead forecasts. Both MCS and DM tests highlighted the superiority of long-term memory detecting models LSTM and NAR over both the traditional models and competing neural networks.

Arnerić, Poklepović and Aljinović (2014) compared a Jordan neural network against the standard GARCH model in predicting conditional variance based on past return innovations. Authors developed a recurrent neural network that has a feedback connection from output to input layer through including an additional neuron in the input layer. Economic interpretation of this feedback connection corresponds to the lagged error term. Feeding back the data to the network is similar to GARCH in the sense that GARCH models are influenced by previous variance to produce forecasts. Arnerić et al., (2014) have trained 25 different JNN models by changing the train-test size ratio and the weight of the context unit representing the memory of the network while MSE is found to be lowest when training set size is at 70% of the sample. For weight of the context unit, lowest MSE was achieved using 0.9 which indicates the long-term memory of the neural network. To estimate using the optimal JNN model, in-sample log returns from the Zagreb Stock Exchange's daily closing prices are used for the calculation of squared innovations. Squared innovations are then used as a target for the network and it is trained using squared innovations with one time lag. Their results highlight that selected JNN model has better forecast performance over the benchmark GARCH(1, 1) model.

Christensen, Siggaard and Veliyev (2022) showed that one-day ahead forecasts of realized variance can be improved via machine learning approaches like regularization, tree-based algorithms, and neural networks. The benchmark model that they compare to is HAR (Heterogeneous Autoregressive Model) model. The input variables are daily, weekly, and monthly lags of volatility which captures the long-term dependency structure. The study also includes extended versions of models which consists of exogenous variables in addition to the lagged volatility terms. To avoid overfitting and increase out-of-sample forecasting performance of neural networks, the study utilized learning rate shrinkage, drop-out, early stopping and ensembles. For all models except neural networks, a rolling window scheme is utilized. Christensen et al. (2022) also study the effect of additional input variables by forming three datasets. The first one consists of daily, weekly, and monthly lags of realized variance to ensure direct comparison of HAR and machine learning algorithms. The second dataset includes various financial and macroeconomic variables. To overcome overfitting and multicollinearity concerns for the second dataset, the third dataset is constructed by selecting a subset of variables using least squares regression for each possible combination of feature space. The single best model is then selected using BIC. Christensen et al. (2022) report that machine learning methods can handle nonlinear structure of financial markets and increase in performance as feature space is widened. The random forest and neural network models are stated to perform the best relative to the benchmark HAR model.

Yao, Zhai, Cao, Ding, Liu, and Luo (2017) proposed a two-component volatility model where the long- and short-term components of volatility are modelled separately. They decompose volatility into its long- and short-term components through a low-pass Hodrick-Prescott filter. The long-term component is modelled using an autoregressive neural network and the short-term using AR(1) model. The number of lags to be included in the neural network as well as the number of neurons in the hidden layer are selected by training different networks with various specifications and selecting the one with the lowest measured error. They found 4 lags to yield the best results for the neural network. Total forecasting result is found by summing the short- and long-term forecast outputs. Yao et al., (2017) conducts this experiment to forecast volatilities of EUR/USD, GBP/EUR, GBP/JPY, and GBP/USD exchange rates. They use 1-hour and 1-day realized volatilities, computed from 10 millisecond log returns. Benchmark models are selected as GARCH, EGARCH and neural network model with 4 lags that model realized volatility directly. Their results show that the decomposed model structure provides significantly improved forecasts over the benchmark models, regardless of the forecast horizon. Moreover, they state that the performances of GARCH and EGARCH models deteriorate quickly when moved from 1-hour to 1-day forecast horizon. They also note that neural network that models realized volatility directly performs the worst among all models. This result conforms to that of Nelson et al. (1999) and Zhang and Qi (2005), which concluded that neural network is not able to model volatility directly, but neural networks built with de-seasonalized data could produce significantly more accurate forecasting than with non-deseasonalized data.

The main agreement in the literature is that hybrid models utilizing artificial neural networks improve both in-sample and out-of-sample forecasting accuracy of traditional GARCH-family models. It is also studied extensively that including additional explanatory variables improve the performance of volatility estimates, especially for multi-step ahead forecasts. Among the traditional GARCH-family models, EGARCH and GJR-GARCH stand out as the most implemented and found to yield the best results. When it comes to ANN architectures, recurrent networks are discovered to perform the best, especially when overfitting is considered for producing out-of-sample forecasts.

CHAPTER 3

MODEL BUILDING AND FORECASTING METHODOLOGY

This study investigates the conditional volatility by constructing hybrid models that bring together various GARCH-family models and ANN. Robustness checks will be conducted for all selected models in terms of forecast horizon and the information set. In addition to one-day-ahead forecasts, 5-day and 10-day-ahead estimates will be produced that corresponds to one and two weeks of trading, respectively. For the information set, all hybrid models will first be formed using just outputs or the lagged terms in GARCH-family models. To test the effect of adding explanatory variables, an explanatory information set consisting of 12 variables will be utilized. Chapter 3.5 will expand on how these robustness checks are conducted.

3.1 GARCH-family models

ARCH model proposed by Engle (1982) conveys that the time series in question has a time-varying variance (heteroskedasticity) that depends on (conditional) lagged effects (autoregressive). ARCH-family models have two equations, including the mean and the variance equation. ARCH models first fit the mean equation to estimate the residuals, from which the conditional variance is measured.

$$r_t = \mu_t + \varepsilon_t \tag{1}$$

$$\varepsilon_t \sim (0, \sigma_t^2) \tag{2}$$

$$\sigma_t^2 = \alpha_0 + \sum_i^p \alpha_i \varepsilon_{t-i}^2 \quad (3)$$

Equations (1) through (3) constitute an ARCH model. The term μ_t can be zero, constant, or modelled as a function of past values and past errors. ε_t represents the residual values left after estimating the coefficients. Residuals is a serially uncorrelated sequence with zero mean and the conditional variance of σ_t^2 which may be nonstationary. ARCH model has been extended in several directions based on the empirical evidence that the volatility process is non-linear, asymmetry, and has a long memory. Such extensions that are popular in the literature can be referred to as GARCH—Bollerslev (1986), EGARCH—Nelson (1991) and GJR-GARCH—Glosten et al. (1993).

GARCH model by Bollerslev (1986) introduces a moving average component in addition to the autoregressive component from the ARCH model. It includes lagged variance terms together with lagged residual error terms from a mean process. The intuition behind comes from observations of volatility clustering where one can notice patterns of consecutive periods of low and high volatility in the returns. Equation (4) represents a GARCH model.

$$\sigma_t^2 = \alpha_0 + \sum_i^p \alpha_i \varepsilon_{t-i}^2 + \sum_j^q \beta_j \sigma_{t-j}^2 \quad (4)$$

GARCH model predicts volatility by taking the weighted sum of the variance predicted from the past and volatilities observed from the past. Moreover, it treats positive and negative return shocks symmetrically and thus, ARCH and GARCH models fail to capture asymmetric behavior of returns.

EGARCH and GJR-GARCH were introduced to account for leverage effects of price change on conditional variance. This means large price declines can have a bigger impact on volatility than large price increases. Equation (5) represents an EGARCH model.

$$\ln \sigma_t^2 = \alpha_0 + \sum_{i}^{p} \beta_i \ln \sigma_{t-i}^2 + \sum_{j}^{q} \gamma_j \left(\left| (\varepsilon/\sigma)_{t-j} - \sqrt{2/\pi} \right| \right) + \sum_{k}^{r} \delta_k (\varepsilon/\sigma)_{t-k} \qquad (5)$$

EGARCH model does not require every coefficient to be nonnegative unlike ARCH and GARCH models. The model uses a logarithmic specification and guarantees that the conditional variance is positive even if the estimated parameters are negative.

GJR is another augmentation of GARCH that allows past negative return shocks to affect volatility differently than positive return shocks. Equations (6) and (7) represent a GJR-GARCH model.

$$\sigma_t^2 = \alpha_0 + \sum_{i}^{p} \alpha_i \varepsilon_{t-i}^2 + \sum_{j}^{q} \beta_j \sigma_{t-j}^2 + \sum_{k}^{r} \emptyset_k D_{t-k} \varepsilon_{t-k}^2 \qquad (6)$$

$$D_{t-k} = \begin{cases} 1 & if \quad \varepsilon_{t-k} < 0, \\ 0 & if \quad \varepsilon_{t-k} \geq 0 \end{cases} \qquad (7)$$

Equation (7) is a multiplicative dummy variable to check whether there is statistically significant difference when return shocks are negative. When the return shock is negative, D_{t-k} gets a value of 1 and 0 otherwise. This means “good news” in the market has an impact α_i while “bad news” has an impact of $\alpha_i + \emptyset_k$. The term $\emptyset_k$ is called the asymmetry term. If it is statistically significantly positive, it means the return series has an asymmetric response to positive and negative shocks. If it is zero, the model collapses to a standard GARCH.

3.2 Neural networks

Standard time series models assume a linear pattern in the data while there exist highly non-linear patterns such that they cannot be captured by GARCH models. On the other hand, neural networks can approximate any continuous function by modifying the structure of the network. They are highly flexible and non-linear so that they can handle increasingly complex problems. Furthermore, one does not have to form assumptions about the functional relationship among the variables. For these reasons, neural networks have been employed to forecast time series in various areas, including forecasting extremely volatile financial variables that are hard to predict otherwise, using standard statistical methodologies.

Neural networks are designed to mimic the structure of the human brain. A set of processing neurons are interconnected through layers. These layers are called input, hidden, and output layers. Each layer in the network can have varying number of neurons. The optimal structure of the network in terms of number of neurons and number of hidden layers depends on the problem and is determined through hyperparameter tuning. The connections between neurons from different layers are called weights and they are activated by reaching a threshold, managed by an activation function. The weights between neurons in different layers are learnt using a variety of ways. Most popular technique is called back propagation. It is based on gradient descent rule and updates the weight at each iteration until it yields no improvement in the error function. Neural networks can be divided into two types: feed-forward and recurrent neural networks.

Feed-forward networks receives information from the input layer and forwards it to the output layer through hidden layers where the data is transformed using an activation function. The information moves in only one direction, from input neurons to the output neurons so that there are no cycles or loops in the network, unlike recurrent networks. Figure (1) and Equation (8) represent a sample FNN (feed-forward neural network) structure with a single hidden layer.

$$\widehat{y_t} = f\left(\emptyset_{co} + \sum_{h=1}^{q} \emptyset_{ho} g\left(\emptyset_{ch} + \sum_{i=1}^{p} \emptyset_{ih} x_i\right)\right) \tag{8}$$

The terms $f(.)$ and $g(.)$ are the activation functions for the output and hidden layers, respectively. $\emptyset_{co}$ and $\emptyset_{ch}$ are the bias terms for the output and hidden layers. $\emptyset_{ih}$ represents the weights connecting input to hidden layer and $\emptyset_{ho}$ represents hidden to output layer. Finally, x_i denotes the input variables that are fed to the network.

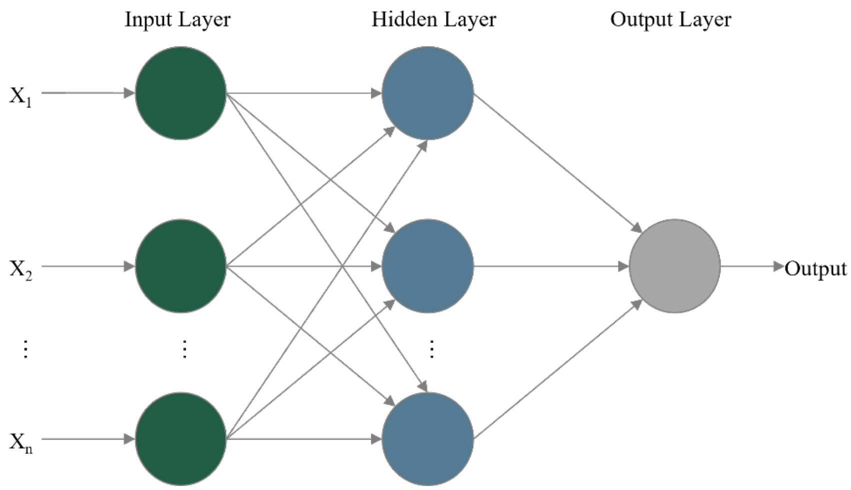


Figure 1. FNN with a single hidden layer

One of the most common problems of ANNs is overfitting and vanishing gradient. To overcome overfitting, there exists a large literature that proposes various regularization methodologies to find optimal weights. One of the most used ones stands out as early stopping and thus, will be used in this study. Early stopping means training is stopped when there is no improvement in the validation error. First, dataset is divided up into three subsets, namely the training set, validation set and test set. The training set is utilized to compute the weights of the network. The error on the validation set is screened through the training process. If the validation error increases over a user-defined threshold, the training is stopped. Another technique to prevent overfitting is called dropout. It refers to dropping out nodes in the hidden layer where all connections, both forward and backward, are temporarily removed from the network. The dropped nodes are determined using a pre-defined dropout probability. How it solves the overfitting problem is that during training, a unit may change in a way that fixes the mistakes of other units. This may lead to complex co-adaptations on the training set which will fail to generalize on unseen data. Using dropout prevents this co-adaptation procedure and thus reduces the overfitting problem. Both techniques will be utilized to construct neural networks in this study.

The other problem, vanishing gradient, arises from the fact that the network weights are updated through gradient descent which leads to exponentially smaller gradient magnitudes at each iteration. As a result, learning becomes slow and steps very small. A possible cure to this problem is the choice of the activation function. A logistic function that reduces all input values to be between 0 and 1 or a sigmoid function produces a small change in the output, in turn vanishes the gradient rapidly. Instead, ReLU (Rectified Linear Units) or leaky ReLU could be used because derivative of ReLU function is defined to be 1 for inputs greater than zero. Another method is to initialize weights to the network so that they do not vanish during back propagation. This study uses ReLU activation function for the hidden layers of the network.

3.3 Proposed hybrid models

This study proposes two types of hybrid models using GARCH-family models (GARCH, EGARCH, GJR-GARCH) and ANN (Feed-forward and recurrent neural networks) to forecast the return volatility of BIST (Istanbul Stock Exchange) 100. For the first hybrid model, GARCH-family models' conditional volatility series will be produced after finding out the optimal number of lagged terms for each model type by checking a pre-defined error measure on the validation set. The GARCH-family model with the most accurate out-of-sample predictions will be chosen and its conditional volatility series will be fed to neural networks along with additional explanatory variables to form the hybrid model.

For the second type of hybrid model, instead of feeding the conditional volatility series produced by a GARCH-family model, its model specifications will be produced and fed to the networks. Hence, the conditional volatility pattern will only be learnt by the neural networks. Using this method, multiple GARCH-family model specifications can be hybridized at the same time. The GARCH model is useful for capturing volatility clustering and leptokurtosis information, while the EGARCH and GJR-GARCH models are useful for leverage effect modeling. Hence, each GARCH-family model has advantages and disadvantages in its volatility prediction. Therefore, combining multiple GARCH-family models to reflect various characteristics, rather than hybridizing a single GARCH-family model with neural networks is expected enhance volatility predictions. Following the approach of Hyup Roh (2007), the extracted variables from GARCH are ε_{t-p}^2 and σ_{t-q}^2 while EGARCH contributes $\ln \sigma_{t-p}^2$, $\left(\left|(\varepsilon/\sigma)_{t-q} - \sqrt{2/\pi}\right|\right)$ and $(\varepsilon/\sigma)_{t-r}$. Lastly, GJR-GARCH brings in ε_{t-p}^2, σ_{t-q}^2, and $D_{t-r}\varepsilon_{t-r}^2$, all with their corresponding coefficients estimated by the respective GARCH-family model.

3.4 Forecasting methodology

To construct individual GARCH-family models, zero, constant and AR(1) specifications for returns are tested and the number of lagged terms for all GARCH-family models are found using the approach of Donaldson and Kamstra (1997), that select optimal number of lagged terms for each model by searching a grid of specifications. GARCH models will be examined on the grid p, q ∈ [1, 3] while EGARCH and GJR-GARCH models on the grid p, q, r ∈ [1, 3]. To select the most reasonable specifications, models that produced negative variance forecasts are discarded from selection. Moreover, Ljung-

Box test is used to test for the absence of serial autocorrelation up to 24 lags in the squared standardized residuals. Models that produced such forecasts that lead to in-sample rejection of Ljung-Box test are also discarded. Since it is important to compare out-of-sample forecasting performance of proposed models, dataset is divided to training, validation, and test sets. Model specifications and optimal GARCH-family models are found by training the models using the training set, selecting optimal specifications by checking the error measure obtained from the validation set while out-of-sample forecasts are produced using the test set. Estimations are done recursively, re-estimating the model parameters at each step.

For hybrid models, a four-layered network structure is utilized. First and last layers are the input and output layers, respectively in addition to two hidden layers that contain 8 units each. Both the input and output series are standardized using the in-sample mean and variance to improve optimization procedure of neural networks. The dataset is divided into training and test sets. The recursive estimation technique that re-estimates parameters at each step is not used for neural networks since the model is already computationally heavy. Instead, network weights are found on the training set and used to produce out-of-sample forecasts, unlike GARCH-family model estimation. To test the effect of employing different network architectures, different numbers of hidden layers and neurons are used and their effect on out-of-sample results are reported. In addition to 1-day ahead, 5- and 10-day-ahead forecasts are produced to test whether model accuracy deteriorates when forecast horizon is extended and if so, how fast is the speed of decay.

CHAPTER 4

DATASET CHARACTERISTICS

This study uses on daily adjusted close prices of BIST 100 over the period of January 5th, 2010, to December 30th, 2014, which consists of 1255 observations. BIST 100 data was gathered from the Central Bank of Turkey website. The target value for the supervised learning process is realized volatility which is obtained by observing how much the stock price has changed during a certain period. Equation (9) represents the calculation of realized volatility for a certain period.

$$RV_t = \sqrt{\frac{1}{\rho_t}\sum_{t=1}^{\rho_t}(s_t - \bar{s})^2} \tag{9}$$

Given period is represented by ρ_t which is the number of trading days after time t. s_t is the log return rate of BIST 100 index at time t which is calculated by taking the first difference of log prices and $\bar{s}$ is the average log return rate during ρ_t. When daily realized volatility is considered, $\bar{s}$ boils down to next day's log return rate since ρ_t only includes the following day after t. The dataset is split into training, validation, and test sets with percentages of 80, 10 and 10%, respectively. The latter consists of 125 days of observations and reserved for producing out-of-sample volatility estimates.

Empirical research suggests that there might some uncaptured information left from historical prices that affect future realized volatility. For example, Christiansen et al., (2012) found that the inclusion of macro-finance predictors can enhance the forecast performance relative to simple autoregressive benchmarks. Hence, 4 endogenous and 5 exogenous variables in addition to few dummy variables are produced and included in the information set.

Following the approach of Hamid and Iqbal (2004), Hajizadeh et al., (2012), and Kim and Won (2018) exogenous variables consist of log returns of S&P 500, FTSE 100 (Financial Times Stock Exchange), US Dollar/Turkish Lira exchange rate, Gold, and Crude Oil for the same period as BIST 100, using again the adjusted close prices of each respective timeseries. For the preprocessing of time series data, each time series are first joined using the days that BIST 100 is open. It is noted that S&P 500 data is shifted by one period before joining to account for the fact that financial markets in the US opens towards the end of day in Turkey. So, the information of day t in the US is mostly reflected on day $t + 1$ in Turkey. Joining the time series led to missing data since there are some days that other exchanges are closed when BIST 100 is open. The number of missing days for other exchanges corresponded to approximately 3% of the whole dataset. None of these missing days were consecutive except the two days falling on Christmas in 2010, 2011, 2012 and 2013 for FTSE 100 in addition to two days starting on 4th of June in 2012 due to anniversary of the accession of Queen Elizabeth II . Missing values were smoothed using ARIMA. ARIMA models consist of an (AR) part where a weighted sum of lagged values is used, an (MA) part where a weighted sum of lagged residuals is used and lastly an (I) part indicating the differencing level of the time series. Optimal ARIMA specifications were found using autocorrelation and partial autocorrelation plots after logging and differencing the time series to achieve stationarity. As a result, ARIMA(1, 1, 1) is used to smooth the missing values for each time series variable. For observations where there are insufficient degrees of freedom to estimate, simple moving average with a lookback period of three days is used. On the last case where there are less than three historical observations, missing value is imputed by carrying back the next available observation in the series.

Table (1) shows common descriptive statistics such as mean, standard deviation, skewness, and kurtosis in addition to results of ADF (Augmented Dicker-Fuller) test which tests for stationarity, Jarque-Bera test which tests for normality, and ARCH test which tests for existence of ARCH effects, of logarithmic returns from each time series data used in this study.

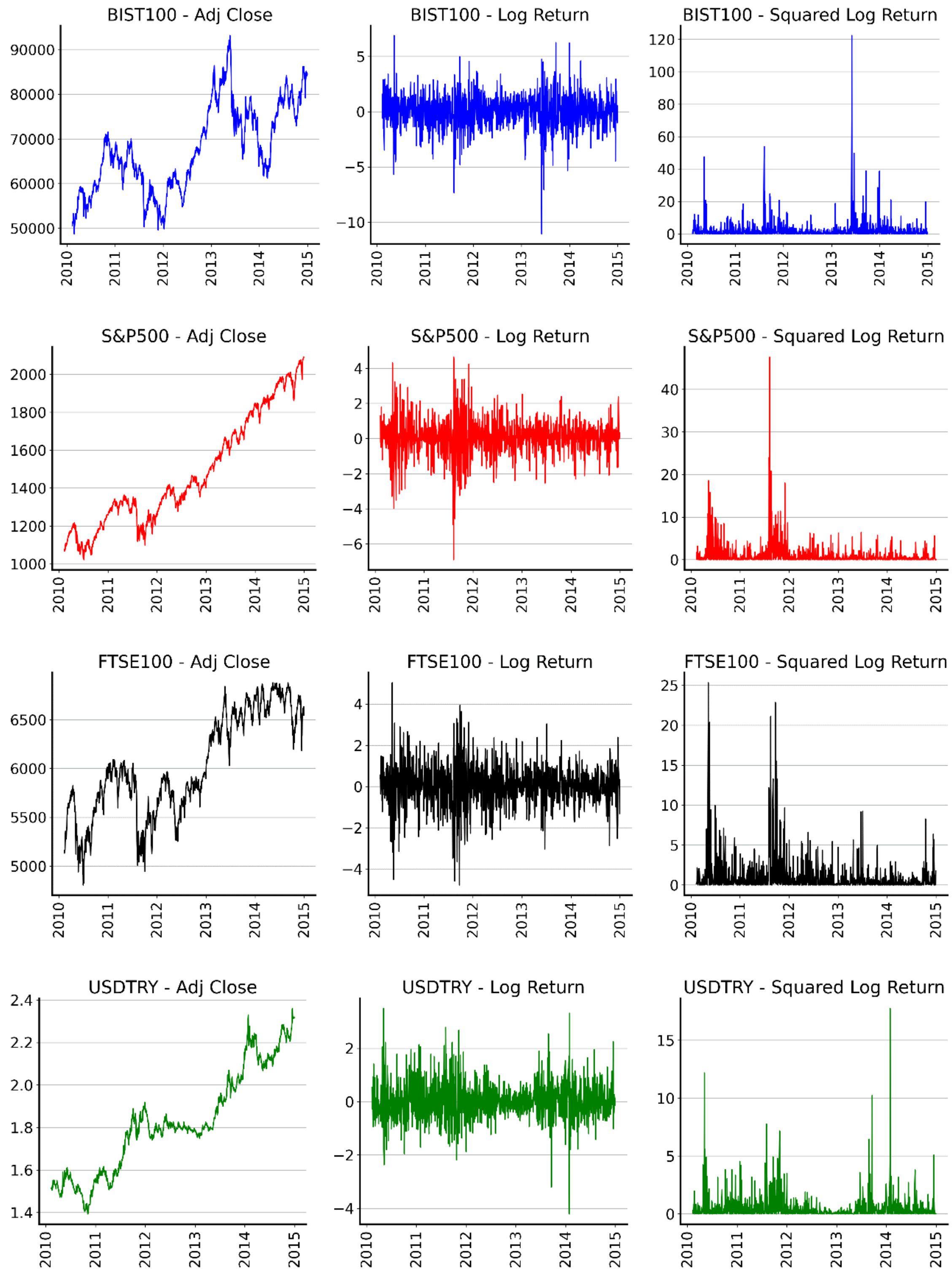
Table 1. Summary Statistics of Log Returns of Each Time Series Variable

Statistics	BIST 100	S&P 500	FTSE 100	USD - TRY	Gold	Crude Oil
Mean	0.04	0.06	0.02	0.03	0.01	-0.03
Standard deviation	1.51	1.00	0.99	0.70	1.15	1.68
Skewness	-0.59	-0.44	-0.19	0.23	-0.95	-0.32
Kurtosis	4.30	4.91	2.81	2.87	7.30	3.76
Jarque-Bera test	1018.08^{*}	1274.08^{*}	410.88^{*}	432.15^{*}	2914.80^{*}	745.02^{*}
ADF test	-36.40^{*}	-12.93^{*}	-34.85^{*}	-36.90^{*}	-36.04^{*}	-36.35^{*}
ARCH-LM test (lags = 5)	66.31^{*}	268.42^{*}	132.00^{*}	115.67^{*}	26.72^{*}	39.70^{*}

Notes: $*$ denotes a rejection of the null hypothesis at the 1% significance level.

Skewness values for BIST 100, S&P 500 and Gold are moderately left skewed. FTSE 100 stands out as the time series that most resembles a symmetric normal distribution. USD-TRY exchange rate and Crude Oil's return series stand out as platykurtic while others suggest leptokurtic distributions. Gold, S&P 500, and BIST 100 appear to have longer and fatter tails which is a common characteristic of returns. When the main time series, BIST 100, is analyzed from Table (1), the Jarque-Bera test confirms the fact that the normality of return distribution is rejected, and it can be concluded that daily returns exhibit significantly heavier tails than a normal distribution. Moreover, the ADF test is significant at 1% level which allows the conclusion that the return series is stationary. Lastly, the ARCH-LM test is also significant at 1% level which indicates the existence of ARCH effects in the BIST 100 return series and the presence of a fat-tailed distribution and time-varying volatility. All independent variables yield similar test results as the BIST 100 return series.

Figure (2) shows the evolution of the adjusted close price, daily logarithmic return, and squared logarithmic return of each time series. All adjusted close price series show an upward trend except Gold and Crude Oil. The sudden drop in Crude Oil in 2014 is due to demand-supply mismatch, created by the Crude Oil production growth in the US which resulted in much lower dependency to foreign oil, increased production levels by OPEC (Organization of Petroleum Producing Countries), and weakening demand in the light of global economic slowdown. One of the reasons behind the gold price drop in 2013 is the announcement of the Federal Reserve which stated that the stimulus program starting with the 2008 financial crisis was finally being wrapped up. This announcement combined with low inflation rates rendered gold's role as a hedge against inflation obsolete. The bullish stock exchanges, as seen from Figure (2) also pushed investors to pursue increasing returns.



BIST100 - Adj Close
BIST100 - Log Return
BIST100 - Squared Log Return
S&P500 - Adj Close
S&P500 - Log Return
S&P500 - Squared Log Return
FTSE100 - Adj Close
FTSE100 - Log Return
FTSE100 - Squared Log Return
USDTRY - Adj Close
USDTRY - Log Return
USDTRY - Squared Log Return
2010
2011
2012
2013
2014
2015

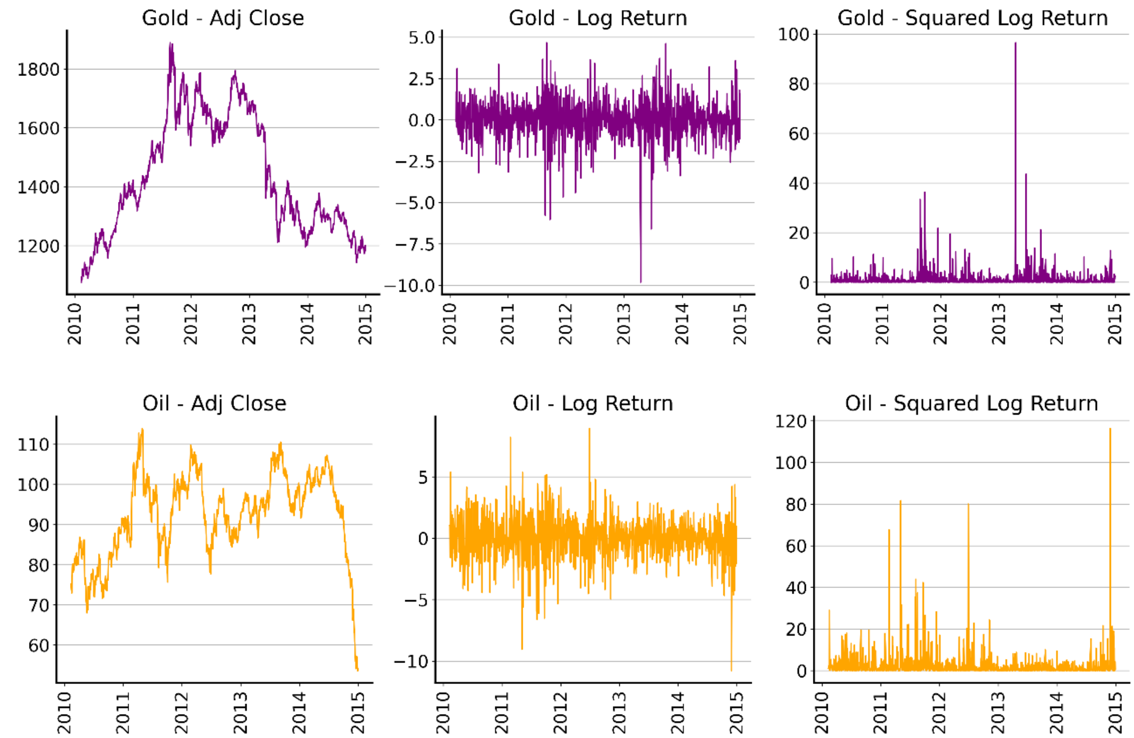


Figure 2. Time series plots of daily adjusted close price, log return, and squared log return for BIST 100, S&P 500, FTSE 100, USD/TRY exchange rate, gold, and crude oil

The included endogenous variables are created following Luong and Dokuchaev (2018) who investigated a number of technical indicators using variable importance ranking from random forests and found several indicators that were best for volatility prediction. The indicators included in this study are listed as follows.

i) Average True Range (ATR) measures market volatility by decomposing the entire range of an asset price for a given period. It can be calculated on intraday, daily, weekly, or monthly basis. Equation (10) presents the computation method for the true range of an asset

$$TR_\delta = max\{H_\delta - L_\delta, |H_\delta - C_{\delta-1}|, |L_\delta - C_{\delta-1}|\}, \tag{10}$$

where H_δ, L_δ, and $C_{\delta-1}$ are the current highest price, the current lowest price, and the previous last close price of a selected period, respectively. From here, the average true range within n days is represented by Equation (11).

$$ATR_{\delta-n,\delta} = \frac{(n-1)ATR_{\delta-n,\delta} + TR_\delta}{n} \tag{11}$$

ii) Close Relative to Daily Range (CRTDR) is an indicator of the location of last return and it stands out as a powerful predictor for next returns. It is calculated as seen in Equation (12)

$$CRTDR_\delta = \frac{C_\delta - L_\delta}{H_\delta - L_\delta}, \tag{12}$$

where H_δ, L_δ, and C_δ are the high, low, and current close prices at time δ for a given period.

iii) Exponential Moving Average of Realized Volatility (EMARV) places greater weight on most recent data points. The weighting applied to recent data points depend on the number of periods and the smoothing factor. Equation (13) represents the formula for EMARV

$$EMARV_\delta = \left(RV_\delta \times \left(\frac{Smoothing}{1+\ \delta} \right) \right) + EMARV_{\delta-1} \times \left(1 - \left(\frac{Smoothing}{1+\ \delta} \right) \right), \quad (13)$$

where smoothing factor is taken to be 2 for this study and RV_δ represents average of log returns in the last δ days. Since there is no $\delta - 1$ value at the first step, RV_δ is taken to be EMARV value for the first step.

iv) Moving Average Convergence / Divergence Oscillator (MACD) is a trend-following momentum indicator that shows the relationship between two exponential moving averages. It is found by subtracting the longer moving average from the shorter one. It is best used with daily periods and the periods for two exponential moving averages are usually taken to be 26 and 12 days, respectively.

In addition to endogenous and exogenous variables, two types of dummy variables are computed and added to the dataset, namely DOW (Day of Week) and IBS (Is BIST Closed). The first type is binary dummy columns for days of the week. Four columns are added to the dataset by dropping one of the weekdays to avoid multicollinearity. It also aims to capture the volatility information in different weekdays, for example, Mondays are expected to be more volatile since they also include the accumulated information from the weekend when the markets are closed. The second type is a single binary column that covers consecutive periods of times when BIST is closed in a given time range except on the weekends. These closed periods are also expected to produce high volatility days in the upcoming period since all new information during that period will be reflected on the next open day in the market.

Following the approaches of Hajizadeh et al., (2012), Kim and Won, (2018), Kristjanpoller and Minutolo, (2016) and the conclusion of Christensen et al., (2022) who found that neural networks can extract information about future realized volatility in an increased feature space which linear models usually fail to do, this study uses a dataset consisting of endogenous, exogenous, and dummy variables to study the effects of including explanatory variables in addition to output / specifications coming in from GARCH-family models. A full list of explanatory variables is shown on Table (2), with their corresponding acronyms. All variables except the last two dummy variables are lagged by one day before being fed to neural networks.

Table 2. List of Explanatory Variables

No	Acronym	Explanatory Variable
1	BIST	Daily log returns of BIST 100
2	S&P	Daily log returns of S&P 500
3	FTSE	Daily log returns of FTSE 100
4	USD/TRY	Daily log returns of USD/TRY exchange rate
5	Gold	Daily log returns of Gold
6	Oil	Daily log returns of Crude Oil
7	ATR	Average True Range
8	CRTDR	Close Relative to Daily Range
9	EMARV	Exponential Moving Average of Realized Volatility
10	MACD	Moving Average Convergence / Divergence Oscillator
11	DOW	Day of Week (Dummy Variable)
12	IBC	Is BIST Closed (Dummy Variable)

In order to evaluate and compare the results of different models, this study follows the approaches of Hajizadeh et al., (2012), Kim and Won (2018), and Kristjanpoller and Minutolo (2016). Hence, Mean Squared Error (MSE), Mean Absolute Error (MAE) are used in addition to their non-linear versions, namely Heteroskedasticity-adjusted Mean Squared Error (HMSE), and Heteroskedasticity-adjusted Mean Absolute Error (HMAE) since typical loss functions apply to linear models. These four loss functions are represented with Equations (14) through (17)

$$MAE = \frac{1}{T}\sum|\widehat{v_t} - RV_t|, \tag{14}$$

$$MSE = \frac{1}{T}\sum(\widehat{v_t} - RV_t)^2, \tag{15}$$

$$HMAE = \frac{1}{T}\sum|1 - \widehat{v_t}/RV_t|, \tag{16}$$

$$HMSE = \frac{1}{T}\sum(1 - \widehat{v_t}/RV_t)^2, \tag{17}$$

where $\widehat{v_t}$ is the volatility estimate at time t and RV_t denotes realized volatility at time t while T is the total number of predictions.

Diebold-Mariano (DM) test is conducted to verify the equivalence of out-of-sample forecast accuracy or show statistically significant difference. The obvious approach of favoring the forecast with smaller error measure does not necessarily mean the difference between the models are significant. The null hypothesis of Diebold-Mariano test is that the two forecasts have the same accuracy while the alternative is that they have different levels of accuracy. First, forecast errors are defined as $e_{it} = \widehat{y_{it}} - y_t$ for $i = 1, 2$ where y_t are the actual values in the series and $\widehat{y_{it}}$ is the forecast. The loss associated with forecast i is assumed to be a function of e_{it} which can be denoted by $g(e_{it})$. The loss function $g(.)$ is typically taken as MSE or MAE. The loss differential between the two forecasts is defined as $d_t = g(e_{1t}) - g(e_{2t})$. The two forecasts have equal accuracy if and only if the loss differential d_t has an expectation of zero for all t, $E[d_t] = 0$. The DM statistics are obtained through Equation (18).

$$DM = \frac{\bar{d}}{\sqrt{2\pi \hat{f}_d(0)/T}} \tag{18}$$

$\bar{d} = \frac{1}{T}\sum_{t=1}^{T}\left(g(e_{1t}) - g(e_{2t})\right)$ and $\hat{f}_d(0)$ is a consistent estimate of $f_d(0)$, which stands for the spectral density of the loss differential at frequency 0.

CHAPTER 5

RESULTS

The first results obtained are the conditional volatility forecasts produced by GARCH-family models. Table (3) reports the chosen GARCH-family model specifications while Table (4) shows their out-of-sample forecast accuracies, establishing these models as benchmarks to be compared when ANNs are incorporated to construct hybrid models. All GARCH-family specifications stand out as of low order. This is expected due to the recursive nature in which past conditional variances are computed. The equation for the past conditional variance $\sigma_{t-1}^2 = \alpha_0 + \alpha_1 \varepsilon_{t-2}^2$ already includes the past squared residuals. Also, including past variances are rendered obsolete since all the information of the conditional variance from two steps ago is already contained in the conditional variance of previous step. It is worthwhile to note that the performances of GARCH(1, 1) and GARCH(1, 2) on the validation set are extremely close. The best GARCH-family model in terms of HMSE emerges as the EGARCH(1, 1, 1) with a constant mean model even though it is the worst-performing model on all other metrics. On the other hand, GARCH(1, 1) stands out as the best for all metrics except HMSE. None of the GARCH-family models' out-of-sample results are distinguishably different from one another.

Table 3. Selected GARCH-family Model Specifications

Model	Mean model	p	q	r
GARCH	Zero	1	2	Not applicable
EGARCH	Constant	1	1	1
GJR-GARCH	Constant	1	1	1

Table 4. Out-of-sample Forecast Results of Selected GARCH-family Models

Model	HMSE	HMAE	MSE	MAE
GARCH	0.59	0.58	1.10	0.76
EGARCH	0.57	0.59	1.12	0.79
GJR-GARCH	0.58	0.59	1.11	0.78

Table (5) contains the second set out-of-sample forecast results from hybrid models. For the first type of hybrid model, there are three variations for each GARCH-family model while the second hybrid model has seven different models by combining single or multiple GARCH-family models with neural networks. Out-of-sample results of first set of hybrid models are comparable to each other while FNN combined with the conditional volatility estimates of EGARCH shows a slight improvement in terms of HMSE over others.

Table 5. Out-of-sample Forecast Results of Hybrid Models

Model	HMSE	HMAE	MSE	MAE
Hybrid model I				
FNN-G	0.43	0.50	0.94	0.73
FNN-E	0.42	0.51	0.97	0.77
FNN-J	0.44	0.51	1.00	0.78
Hybrid model II				
FNN-G	0.39	0.51	0.95	0.77
FNN-E	0.36	0.49	0.95	0.78
FNN-J	0.38	0.50	0.88	0.74
FNN-GE	0.38	0.51	0.90	0.76
FNN-GJ	0.38	0.50	0.90	0.76
FNN-EJ	0.36	0.50	0.95	0.79
FNN-GEJ	0.35	0.49	0.96	0.78

Note: G, E, and J represent GARCH, EGARCH, and GJR-GARCH specifications, respectively.

All seven hybrid models that were fed GARCH-family specifications instead of conditional volatility estimates improved out-of-sample forecast accuracies of both the

GARCH-family models and the first type of hybrid models. The best performing model in terms of out-of-sample HMSE loss measure is the hybrid model that contains all three GARCH-family models. This hybrid model achieved an HMSE reduction of approximately 38.6% and 16.7% over the best performing GARCH-family model [EGARCH(1, 1, 1)] and first type of hybrid model (FNN-EGARCH), respectively.

Table (6) represents the change in HMSE for the best hybrid models from each type when a different neural network architecture is used over the baseline architecture with two hidden layers containing 8 neurons each. First part of the table shows architectures with differing number of hidden layers when number of hidden units is kept constant at 8 while the second part displays differing number of hidden units per layer when the number of hidden layers is kept constant at 2. Only the best models from each type of hybrid models are used, namely FNN-EGARCH for hybrid model I and FNN-GEJ for hybrid model II that combines all three GARCH-family models.

Table 6. Percentage Change in OOS HMSE Over Different Network Architectures

Models	Number of hidden layers (8 hidden neurons per layer)		
	1	3	4
Hybrid model I			
FNN-E	-6.66%	-7.14%	-11.90%
Hybrid model II			
FNN-GEJ	11.43%	2.86%	-0.29%
	Number of hidden neurons (2 hidden layers)		
	16	32	64
Hybrid model I			
FNN-E	-2.38%	-2.67%	26.19%
Hybrid model II			
FNN-GEJ	6.57%	8.57%	28.57%

It can be seen from Table (6) that as the number of hidden layers increase, networks' performance increase gradually. As for number of hidden neurons, at 64 neurons per layer, networks start to perform worse on out-of-sample observations. Best performances were achieved with 4 hidden layers (Hidden neurons per layer kept constant at 8) for the number of hidden layers and 32 neurons per hidden layer (Hidden layers kept constant at 2) for the number of hidden units.

Table 7. Percentage Change in OOS Forecast Results Over Different Forecast Horizons

Models	5-day-ahead HMSE	HMAE	MSE	MAE
GARCH-family				
EGARCH	19.30%	3.39%	13.39%	5.06%
Hybrid model I				
FNN-E	2.38%	5.88%	28.87%	14.29%
Hybrid model II				
FNN-GEJ	14.29%	8.16%	42.08%	21.21%

	10-day-ahead HMSE	HMAE	MSE	MAE
GARCH-family				
EGARCH	15.79%	1.69%	17.86%	8.86%
Hybrid model I				
FNN-E	11.90%	6.77%	36.08%	18.18%
Hybrid model II				
FNN-GEJ	31.43%	12.24%	45.83%	21.79%

Table (7) represents the percentage changes in loss measures for the best models from all classes, namely the GARCH-family, hybrid model I, and hybrid model II, when the forecast horizon is extended to 5-day and 10-day-ahead. Hybrid models still outperform GARCH-family models, with hybrid model II performing the best in terms of nominal loss measures. All model types display lower forecast accuracy scores as forecast horizon is extended. For EGARCH, forecast accuracy have not become considerably worse going from 5 to 10-day-ahead forecasts however, the decay in accuracy continued for hybrid models as forecast horizon is extended from 5 to 10 days. The steepest decline in forecast accuracy is observed for hybrid model II, FNN-GEJ.

Table 8. Diebold-Mariano Equal Forecast Accuracy Test Results

Models	EGARCH	FNN-E	FNN-GEJ
EGARCH			
FNN-E	0.00**		
FNN-GEJ	0.00**	0.39	

Note: ** denotes a rejection of the null hypothesis at 1%, respectively.

The results of the Diebold-Mariano equal forecast accuracy tests for best GARCH-family and hybrid models are shown in Table (8). One-step-ahead out-of-sample forecasts were compared using MSE as the error criterion for the DM test. The null hypothesis of the DM test is that the two predictive models have the same level of accuracy. Thus, if the p-value is less than 0.05, the null hypothesis is rejected, that is, the predictive accuracy of the two competing models is significantly different. As shown in Table (8) , the null hypothesis can be rejected in both cases where the hybrid models are benchmarked against EGARCH. Thus, it can be concluded that the comparisons of the out-of-sample forecasts of the hybrid models against GARCH-family models in this study are statistically significant.

CHAPTER 6

CONCLUSION

In this study, two most utilized types of hybrid models are constructed to model the return volatility of BIST 100 and these hybrid models are compared with GARCH-family models. The first type of hybrid model received the conditional volatility estimates of GARCH-family models while the second type of network was fed with GARCH-family model specifications, namely the lagged residuals and lagged volatility terms. In addition, networks received a set of 12 variables, consisting of technical indicators, exogenous variables, and dummy variables. The first finding is that both types of hybrid models improved out-of-sample forecasts of GARCH-family methods and the equal forecast accuracy test proved that the improvement was statistically significant. When different types of hybrid models are compared, the second type again emerged as the better performing model since it had the capability to combine multiple GARCH-family specifications. In fact, the second type of hybrid model that combined all three GARCH-family models used in this study had the lowest error measures out of all models. Hybrid models with increasing network architecture complexity also showed better results but started overfitting the training set when 64 hidden neurons were used per hidden layer. Lastly, loss measures for hybrid models when forecast horizon is extended to 5 and 10-day-ahead were better however the hybrid model predictions deteriorated more rapidly while moving on from 5 to 10-day-ahead horizon, than the benchmark GARCH-family model.

The findings of this study can be improved upon with a variety of ways. First way would be to include additional GARCH-family models. Another way to improve the GARCH-family models would be to include exogenous variables in modelling the conditional volatility although it is not a common practice in the literature. The model for index returns to produce residuals could also be improved by including exogenous regressors and/or trying out other models such as HAR (Heterogenous Autoregression) or LS (Least Squared). Second way would be to utilize recurrent neural networks to construct hybrid models, especially LSTMs which are heavily used in the literature. Another way would be to conduct hyperparameter optimization in a more encompassing manner to improve out-of-sample forecast errors. Different number of hidden layers and hidden neurons could be tried in conjunction with each other if the computational bandwidth allows such tests. In addition, more hyperparameters could be tested such as activation functions, batch sizes, number of epochs, and weight initialization. Moreover, effects of different explanatory variables could be studied to be included in the hybrid models. While hybrid models were able to model the volatility jumps better than GARCH-family models, it still stands out as a major limitation of realized volatility forecasting models. Different financial, macroeconomic variables and technical indicators could be studied to capture information on expected volatility shocks in the market.

Business managers can use the findings of this study to inform various decisions related to investments, risk management, and resource allocation. The volatility estimates can help managers make informed decisions about when to buy or sell stocks, based on expected levels of volatility. For example, if the algorithm predicts high volatility, managers may choose to reduce their exposure to the stock market by selling stocks or investing in low-risk assets. Another way is that managers can adjust their risk management strategies to protect their investments by anticipating periods of high volatility. For example, they may choose to hedge their portfolios by purchasing options or other derivatives that can mitigate the impact of market fluctuations. Lastly, by understanding the level of risk associated with different investments, managers can allocate resources more effectively. For instance, they may choose to allocate more resources to lower-risk investments during periods of high market volatility. It is important to note that stock market predictions are not always accurate, and that managers should consider multiple sources of information when making investment decisions. The use of algorithmic predictions should be just one component of a broader investment strategy that considers various factors, such as market trends, economic conditions, and company-specific data.

REFERENCES

Arnerić, J., Poklepović, T., & Aljinović, Z. (2014). GARCH based artificial neural networks in forecasting conditional variance of stock returns. *Croatian Operational Research Review*, *5*(2), 329–343. https://doi.org/10.17535/crorr.2014.0017

Bildirici, M., & Ersin, Ö. Ö. (2009). Improving forecasts of GARCH family models with the artificial neural networks: An application to the daily returns in Istanbul Stock Exchange. *Expert Systems with Applications*, *36*(4), 7355–7362. https://doi.org/10.1016/j.eswa.2008.09.051

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, *31*(3), 307–327. https://doi.org/10.1016/0304-4076(86)90063-1

Bucci, A. (2019). Realized volatility forecasting with neural networks. *Munich Personal RePEc Archive*. https://mpra.ub.uni-muenchen.de/95443/

Christensen, K., Siggaard, M., & Veliyev, B. (2022). A machine learning approach to volatility forecasting. *Journal of Financial Econometrics*, nbac020. https://doi.org/10.1093/jjfinec/nbac020

Christiansen, C., Schmeling, M., & Schrimpf, A. (2012). A comprehensive look at financial volatility prediction by economic variables: Financial volatility prediction by economic variables. *Journal of Applied Econometrics*, *27*(6), 956–977. https://doi.org/10.1002/jae.2298

Donaldson, R. G., & Kamstra, M. (1997). An artificial neural network-GARCH model for international stock return volatility. *Journal of Empirical Finance*, *4*(1), 17–46. https://doi.org/10.1016/S0927-5398(96)00011-4

Engle, R. F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica*, *50*(4), 987. https://doi.org/10.2307/1912773

Hajizadeh, E., Seifi, A., Fazel Zarandi, M. H., & Turksen, I. B. (2012). A hybrid modeling approach for forecasting the volatility of S&P 500 index return. *Expert Systems with Applications*, *39*(1), 431–436. https://doi.org/10.1016/j.eswa.2011.07.033

Hamid, S. A., & Iqbal, Z. (2004). Using neural networks for forecasting volatility of S&P 500 Index futures prices. *Journal of Business Research*, *57*(10), 1116–1125. https://doi.org/10.1016/S0148-2963(03)00043-2

Hyup Roh, T. (2007). Forecasting the volatility of stock price index. *Expert Systems with Applications*, *33*(4), 916–922. https://doi.org/10.1016/j.eswa.2006.08.001

Kim, H. Y., & Won, C. H. (2018). Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models. *Expert Systems with Applications*, *103*, 25–37. https://doi.org/10.1016/j.eswa.2018.03.002

Kristjanpoller R, W., & Hernández P, E. (2017). Volatility of main metals forecasted by a hybrid ANN-GARCH model with regressors. *Expert Systems with Applications*, *84*, 290–300. https://doi.org/10.1016/j.eswa.2017.05.024

Kristjanpoller, W., Fadic, A., & Minutolo, M. C. (2014). Volatility forecast using hybrid Neural Network models. *Expert Systems with Applications*, *41*(5), 2437–2442. https://doi.org/10.1016/j.eswa.2013.09.043

Kristjanpoller, W., & Minutolo, M. C. (2016). Forecasting volatility of oil price using an artificial neural network-GARCH model. *Expert Systems with Applications*, *65*, 233–241. https://doi.org/10.1016/j.eswa.2016.08.045

Luong, C., & Dokuchaev, N. (2018). Forecasting of realised volatility with the random forests algorithm. *Journal of Risk and Financial Management*, *11*(4), 61. https://doi.org/10.3390/jrfm11040061

Monfared, S. A., & Enke, D. (2014). Volatility forecasting using a hybrid GJR-GARCH neural network model. *Procedia Computer Science*, *36*, 246–253. https://doi.org/10.1016/j.procs.2014.09.087

Yao, Y., Zhai, J., Cao, Y., Ding, X., Liu, J., & Luo, Y. (2017). Data analytics enhanced component volatility model. *Expert Systems with Applications*, *84*, 232–241. https://doi.org/10.1016/j.eswa.2017.05.025