**Stock Market Volatility Prediction using Long Short-Term Memory Neural Networks**

**Emanuel-Vasile Puțura**

Abstract

This experimental research report aims to propose an artificial intelligence-based model to predict the realized volatility of a stock in the market. Firstly, we give an overview over the main theoretical concepts needed for our research, and we examine the state-of-the-art in the field. We briefly present the implied and realized volatility concepts, the GARCH financial model, and the LSTM neural networks, since our solution uses these concepts. We use a publicly available dataset provided and collected by a large trading and market making company (Optiver). Additionally, we enhance the dataset based on our observations, using several data analytics and feature engineering techniques, including the Isolation Forest outlier detection algorithm. We formulate our concrete approach to this problem, using a LSTM-GARCH hybrid architecture, and present the experiments and measurements we want to use, and the expected results of our approach.

1. Introduction
   1. Stock market overview

The stock market is a centralized marketplace, or platform, where buyers and sellers trade shares of publicly listed companies. Such financial activities are performed using formal exchanges, or via over-the-counter marketplaces that operate under a defined set of regulations and are traded without being listed on public exchanges. The stock market is a crucial component of the global financial system, providing a way for companies to issue stocks and raise capital, and for investors to buy and sell these stocks, enabling them to participate in the ownership and potential profits of these companies. Electronic trading (i.e., the buying and selling of financial instruments online) first started in the 1970s, but significant development occurred during the 1990s and again in the 2000s with the spread of the Internet.

Stocks, also known as equities or shares, represent ownership stakes in a company. There are many types of different financial instruments that could be traded, each with its own particularities and potential advantages (e.g., bonds, options, futures, exchange-traded funds, mutual funds, stock market indices, etc.). The stock market operates through stock exchanges where buyers and sellers interact. Stock prices are determined by various factors, including company performance, economic conditions, industry trends, and investor sentiment.

* 1. Stock market prediction problem

The stock market represents a big opportunity both for individuals and companies to build and accumulate wealth. Naturally, this motivated investors and researchers to attempt to develop and test models aimed at predicting stock market behavior. Obviously, this is an extremely challenging problem, since the markets are dynamic, non-stationary, non-linear, noisy, and sometimes even chaotic. As it is well known, the markets are affected by many apparently unrelated domains, such as economic, psychological, political, and company specific factors [1].

Artificial intelligence has emerged as a powerful tool in the realm of stock market predictions. Machine learning algorithms, a subset of artificial intelligence, can analyze vast amounts of historical and real-time data to identify patterns and trends that may be challenging for human analysts to discern. These algorithms can process information at a speed and scale that surpasses traditional methods, enabling more sophisticated predictions. One important mention to make is that stock market prediction does not refer solely to predicting the future price of a stock. Instead, it can also refer to volatility prediction, volume prediction, sentiment analysis, or liquidity prediction. These advanced technologies enable the analysis of vast datasets, identification of complex patterns, and adaptation to changing market conditions. By addressing various aspects of market behavior, stock market prediction becomes a comprehensive discipline that empowers investors and financial professionals to make informed decisions across different dimensions of market activity.

* 1. This paper

This paper is an experimental research report aiming to propose a solution for predicting the stock market realized volatility, using artificial intelligence and data analytics algorithms. Realized volatility prediction involves forecasting the actual level of price fluctuation that a financial asset experiences over a given period. Unlike implied volatility, which is derived from options prices and reflects market expectations for future volatility, realized volatility is based on historical price movements. Firstly, we give an overview of the stock market, the problem of predicting various stock market metrics and indices, and the structure and goals of this paper.

The second section consists of a brief overview over the main theoretical concepts needed for our experimental research. The section goes over what implied and realized volatility mean in the context of the stock market, and presents the GARCH econometrics model and the long short-term memory networks in artificial intelligence. In the next section, we briefly describe the state-of-the-art in this topic. The forth section consists of the solution setup, including our own understanding of the artificial intelligence-based modeling of the problem. We present an in-depth critical analysis of the whole problem and theoretically formulate the problem definition for our experimental research report. Also, we present the relevance of the research method used, the dataset we train our models on, the data analytics algorithms we use for handling and enhancing this dataset, a concrete solution proposal for this problem, what results we expect to obtain and what metrics we use for our experiments. Furthermore, the experimental results section describes the experiments we perform and a qualitative and quantitative analysis of the experimental results we expect to obtain. We base these estimations on the results obtained in papers proposing similar methods and on our analysis of the topic. Finally, the conclusions section presents some final remarks on this topic and several future work proposals.

1. Theoretical background

This section is intended to present a brief overview of the main theoretical concepts needed for our experimental research. This consists of describing what implied and realized volatility are in the context of the stock market, and briefly presenting the GARCH model in econometrics [4] and the long short-term memory networks in artificial intelligence [5]. We present these topics since, as we will see in the next sections, our approach to this problem uses all of these concepts.

* 1. Implied and realized volatility in the stock market

Implied and realized volatility are critical concepts in the stock market, offering distinct insights into market dynamics from different perspectives. Implied volatility is a forward-looking metric derived from option prices, encapsulating the market's expectations for future price swings. The Black-Scholes option pricing model [3] is commonly used to estimate implied volatility, with the formula involving complex mathematical computations. It considers factors such as the current stock price, option strike price, risk-free interest rate, time to expiration, and the cumulative distribution functions of the standard normal distribution. Since we will not use the implied volatility for our proposed model, we will not go more in-depth into the calculations involved in the computation of the implied volatility.

On the other hand, realized volatility provides a retrospective view, measuring the actual price fluctuations experienced by an asset over a specific historical period. Calculating realized volatility involves determining the standard deviation of logarithmic returns, offering a statistical representation of past price movements. This metric is essential for understanding how much an asset's price has deviated from its average, aiding investors and traders in assessing historical market behavior. The simplest way to calculate realized volatility is by computing the standard deviation of the logarithmic returns. The formula for realized volatility at time is:

,

where represents how many days are still remained after time , is the logarithmic return of the stock at time , and is the average logarithmic return of the stock over period after time , i.e., over period , to be more precise [2].

Both implied and realized volatility play crucial roles in the financial markets. Implied volatility is often used in option pricing models and trading strategies, guiding investors on the perceived risk and uncertainty in the market. Realized volatility, on the other hand, is instrumental in risk management, helping market participants gauge historical price movements and make informed decisions based on past market behavior. As already mentioned in the previous paragraphs, for this experimental research report, we will focus on predicting the realized volatility in the stock market.

* 1. The GARCH model

The GARCH (Generalized Autoregressive Conditional Heteroskedasticity) model, introduced by Bollerslev in 1986 [4], is a statistical model used in econometrics and finance to analyze and model the volatility of financial time series data. First, let’s describe the autoregressive (AR) component of the model, that assumes that the conditional variance of the time series is a linear function of past squared observations, creating an autoregressive structure. This is expressed as:

,

where is the conditional variance at time , is the residual (i.e., the difference between the observed and expected values) at time , is a constant, and , , ..., are coefficients. Now, GARCH also includes a moving average component, which accounts for past conditional variances. This is expressed as:

,

where ,, ..., are coefficients and is the order of the moving average component. Then, the conditional variance at time , as given by the GARCH model, is defined as follows:

.

The total order of the model is given by and , where is the order of the autoregressive component and is the order of the moving average component. The parameters of the GARCH model are typically estimated using maximum likelihood estimation (MLE). The likelihood function is constructed based on the assumption that the residuals follow a normal distribution.

Once the model is estimated, it can be used to make predictions about future volatility. The predicted conditional variance at time is given by the model's equations. GARCH models are widely used in finance to model and forecast volatility in financial markets.

* 1. Long short-term memory networks (LSTMs)

Introduced in 1997 by Hochreiter and Schmidhuber [5], long short-term memory represents a recurrent neural network (RNN) architecture that has garnered significant popularity in the realm of machine learning. Its primary advantage over traditional RNNs lies in its capability to effectively handle long time-series data, whereas RNNs are more suitable for processing short-term information. This distinction arises from the vanishing gradient problem encountered by RNNs when dealing with lengthy sequences, a challenge that LSTM addresses successfully during training.

1. State-of-the-art

The exploration of models that integrate a neural network with an econometric model has been conducted for a long time, alongside investigations into models that predict volatility using a singular neural network. A hybrid model, incorporating a neural network with a financial time-series model like ARCH or GARCH, has demonstrated superior performance compared to using a standalone GARCH model or a solitary neural network.

In their 2014 study [6], Kristjanpoller, Fadic, and Minutolo examined a hybrid model combining neural networks and GARCH for predicting volatility in three Latin American markets. Their findings indicated that the hybrid neural network model outperformed the GARCH model by reducing the mean absolute percentage error. Furthermore, Kristjanpoller and Minutolo proposed another paper on the same topic in 2016 [7], that studies again the problem of volatility prediction in the stock market, this time specifically for oil prices. Their approach consists of using a hybrid of artificial neural networks and GARCH models to predict the volatility. This time, they not only incorporated additional input variables, such as the index and the exchange rate linked to oil prices, but also identified the optimal architecture for accommodating various time windows of volatility. This study showcased the potential of a ANN-GARCH model architecture, their solution improving the prediction of a single GARCH model by 30.6%. Moreover, to further support these observations and results, Hernandez conducted in 2017 a comparison of GARCH-type models and hybrid neural network models, demonstrating that, for forecasting the volatility of main metals, hybrid neural networks proved to be more suitable than GARCH models [8].

Besides GARCH, there are other financial time-series models that represent variations of it, such as EGARCH, for example. These models leverage characteristics such as leverage effects, excess kurtosis, and volatility clustering from financial time-series data. While these models offer advantages, they also come with their own set of disadvantages in the realm of predicting stock market volatility. Thus, instead of using a combination of a single financial time-series model and a single neural network as in previous studies, Kim and Won propose in their paper from 2018 [2] to instead combine a neural network model with various information obtained from multiple econometrics models. Their hypothesis is that such an approach would be more successful in forecasting financial market volatility. In essence, the study introduces a hybrid LSTM model incorporating multiple GARCH-type models. This volatility prediction model takes parameters from two or more GARCH-type models as inputs for the LSTM model. Unlike previous studies that combined an econometric model and a neural network model, they propose that integrating diverse information from multiple econometric models with a neural network model can enhance the effectiveness of predicting financial market volatility. The proposed model is applied to predict the realized volatility of the KOSPI (Korea Composite Stock Price Index) 200 index. The KOSPI 200 index comprises 200 stocks with significant market capitalization and substantial trading volume.

1. Solution setup

This section is intended to be an in-depth critical analysis of the problem and solution proposed in this experimental research report, i.e., using an artificial intelligence and data analytics based approach in order to predict the stock market realized volatility for various stocks. We give a theoretical statement and specification of the problem we aim to solve and we describe the research methods we want to use for this goal. Furthermore, we analyze the dataset used by us in this paper and the data analytics algorithms used for preprocessing and enhancing the data, making it suitable for our algorithms. Finally, we propose a concrete solution for this problem and state what metrics we use for our experiments and what results we expect to obtain, based on the similar research existing for this problem and on our own estimations and measurements.

As already mentioned, our experimental research report aims to predict the realized volatility for various stocks in the market. The dataset we use for this is Optiver’s realized volatility prediction dataset, and our approach consists of a hybrid LSTM model incorporating also a GARCH-based financial time series model. The idea is to use a similar model to the one proposed by Kim and Won propose in their paper from 2018 [2], but to train it on Optiver’s dataset which consists of more data, collected directly from the market data coming from stock market exchanges Optiver operates on. Moreover, we further enhance this data by using some additional data analytics algorithms we will present later in this section. Our goal and motivation is that we expect that by using better and more realistic data as the one provided in this dataset, we will manage to obtain a better performing volatility prediction model. In the next subsections, we will got into more details regarding all these aspects we mentioned above.

* 1. Dataset

For our experimental research report, we utilize the Optiver realized volatility prediction dataset, accessible on Kaggle. This dataset encompasses stock market data crucial for the practical execution of trades in financial markets, incorporating order book snapshots and executed trades. Notably, it offers a detailed examination of the micro-structure of contemporary financial markets, providing one-second resolution for a uniquely fine-grained analysis. The dataset includes 112 stocks across different market sectors. As already mentioned, it consists of order book and trade data of the stocks for multiple time buckets. The goal is to predict the realized volatility for every single time bucket of the stocks, in the dataset. The time buckets structure across the set of stocks is as follows: there are 107 stocks each having 3830 different time buckets, 3 stocks with 3829 time buckets, and two stocks one having 3820 and one 3815 time buckets corresponding to it, in the training dataset. In conclusion, the dataset consists of exactly 428.932 entries. In the main set (i.e., not the one containing the order book and trade data), there are only 3 columns: the stock ID, the time ID of the bucket, and the target value (i.e., the realized volatility of the next 10 minute window under the same stock ID and time ID).

* + 1. Order book data

An order book is a list of buy and sell orders for a particular financial instrument, such as stocks, displaying the prices and quantities of these orders, organized by price level. This data helps our model get a better understanding of the overall structure of the market at the moment of prediction. Live updates of the exchange order book are normally available live to all market participants, ensuring the transparency of the market. The order book files in this dataset are partitioned by the stock ID column. There are ten columns in the order book data files, including temporal, price, and volume-related data.

An important mention is that these values in the order book are the last snapshots of each second. The order book serves as a valuable tool for traders, aiding in the formulation of well-informed trading decisions by revealing order imbalances that can offer insights into the short-term direction of a stock. A significant imbalance with more buy orders than sell orders may suggest an upward movement in the stock, indicating buying pressure, and vice versa. Traders can also leverage the order book to identify potential support and resistance levels for a stock. The presence of a cluster of substantial buy orders at a specific price may signal a level of support, while an abundance of sell orders at or near a particular price could indicate a resistance zone. Realized volatilities tend to increase as such directional moves become more frequent.

In conclusion, the order book plays a crucial role by providing insights into buy and sell orders at highly competitive levels. Additionally, the model target (i.e., the realized volatility) for stocks is often derived from their order books, making features extracted from them exceptionally valuable. Features that capture imbalances in price and size could prove beneficial for models aimed at predicting the realized volatilities of the upcoming 10-minute window.

* + 1. Trading data

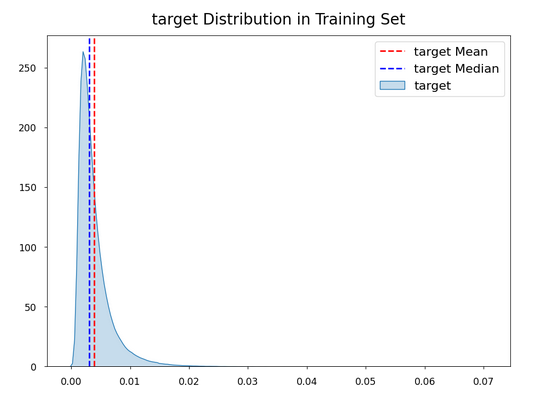
Trading data comprises all individually executed orders for specific stocks and time intervals. The size is the cumulative sum of the size in each individual order, while the price is aggregated as a weighted average of all trades. The order count represents the number of unique trade orders executed. Again, trading data is also partitioned by the stock ID. There are five distinct columns present in these files, including temporal, price, and volume-related data.

A key observation is that a lot of the values in the second column (i.e., • the number of seconds passed since the start of the current time bucket) are missing, since the trading data is more sparse than the order book. This makes sense, since obviously completed trades are much more rare compared to market updates and orders yet unfilled, but present in the order book.

Fluctuations in trade price, size, and order count could potentially serve as indicators of the realized volatility in the upcoming 10-minute window. Another crucial factor to take into account is sparsity. A lack of activity, as indicated by sparsity, suggests a not so active market environment with no trades occurring, which could strongly imply low realized volatility. However, it may not necessarily predict the realized volatility of the subsequent 10-minute windows.

* 1. Data analytics methods and feature engineering

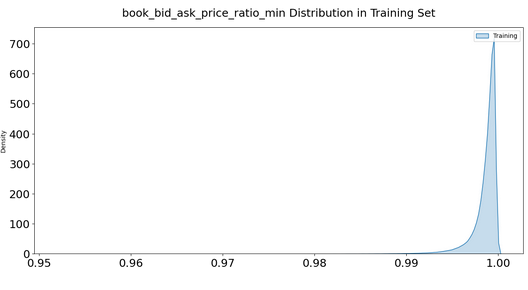
One initial characteristic of the dataset that we noticed is that the distribution of the target value is highly skewed and there are some extreme outliers since some of the stocks present in the dataset are extremely volatile. These extreme outliers can be seen in Fig. 1 from below, where we plotted the distribution of the volatility of the stocks in the dataset.



**Fig. 1:** The distribution of the target value (i.e., the realized volatility) in the dataset

This distribution of the volatility of the stocks leads to the idea of using an outlier detection data analytics algorithm to address the challenges posed usually by the presence of skewed distributions and outliers when dealing with predicting stock market volatility using machine learning. Our choice for this is the Isolation Forest algorithm, since it is particularly suitable at identifying anomalies within datasets that exhibit skewed distributions, making it a fitting choice for datasets with extremely volatile stocks. We train this algorithm on our dataset, focusing on the volatility-related features, i.e., mainly on the realized volatility. Upon identifying the outliers, they can be flagged or removed from the dataset, thereby mitigating their potential impact on the subsequent training of machine learning models. Additionally, the generated outlier scores from the Isolation Forest can be incorporated as an additional feature during the feature engineering stage, providing valuable information to the machine learning model about the anomaly level associated with each observation.

In terms of feature engineering, we incorporate a new feature called bid-ask price ration, which is created when iterating over the order books in the dataset. As the name suggests, it represents the ratio of the bid and ask prices of a stock in the current time window (e.g., we might choose the maximum or minimum of these prices in the window). Multiple aggregations on that simple feature might yield valuable information about realized volatility. Even if this feature is again extremely skewed, as we have plotted in Fig. 2, we still expect it to provide important information to our models.



**Fig. 2:** Order book minimum bid-ask price ratio distribution

Additionally, simple aggregations like mean, standard deviation, minimum and maximum of stock prices or volumes per window can be also used.

* 1. Evaluation metrics

For evaluating our predictions of the realized volatility, we will use the root mean squared percentage error (RMSPE) metric, used also by Optiver when evaluating submissions of models for their dataset. In conclusion, we will be able to compare our results with the publicly available leaderboard for this dataset. The formula for computing RMSPE is the following one:

,

where is the true prediction and is the predicted value of our model. The process involves squaring errors before averaging, heightening the sensitivity of this metric to larger errors. Consequently, the domain does not tolerate substantial errors. A potential drawback of RMSPE is the possibility of triggering a zero division error when a singular data point in the actual values equals zero. Despite the absence of zero target values in the training set, a simple resolution to this issue involves adding a small constant to the actual values. The introduction of a negligible constant has minimal impact on the overall RMSPE and effectively serves as a preventive measure against this potential issue.

Furthermore, an important mention is that when computing the realized volatilities with the formula presented several section above, we use the weighted average price (WAP) of every stock in the order books. The formula that gives the WAP is:

.

Realized volatilities are computed by utilizing the most competitive buy and sell levels. However, the same formula can be applied to the second most competitive buy and sell levels, or alternatively, to other prices and sizes. We evaluate our results on real market-data collected from live exchange feed, provided by Optiver for testing of the models trained on their dataset.

* 1. Problem specification and proposed solution

Given the snapshots of the order book from current time and previous time buckets, and similarly the snapshot of the trading data, in addition to the extra-features we added when preprocessing the dataset, the goal is to predict as accurately as possible the realized volatility for each stock in the dataset, using the formulas and definitions from section 2.1 and section 4.3. We evaluate our results using the RMSPE formula from section 4.3, using real market-data and exchange flow collected and provided by Optiver. Quantitative and experimental research methods are employed for our research, so that we can leverage the strengths of quantitative analysis for modeling market dynamics and experimental methods for testing specific hypotheses or capturing dynamic influences that may not be fully captured by historical data alone.

In our approach, we combine LSTM neural networks with a GARCH model to capture both short-term dependencies and long-term patterns in financial time series data. The parameters of the GARCH model are estimated using maximum likelihood estimation (MLE). We train the LSTM model on historical data to predict the volatility and compute the residuals by comparing the LSTM-predicted values with the actual values. Then, we feed the residuals into the GARCH model to capture the conditional heteroskedasticity in the time series of residuals and combine the LSTM-predicted volatility with the GARCH-modeled conditional volatility to get the final volatility prediction. Additionally, another potential experiment we plan to have is to train the LSTM model on GARCH-modeled volatility. This combination allows the LSTM to capture complex patterns in the data, while the GARCH model helps account for conditional volatility dynamics that may not be fully captured by the neural network. The integration of these models can provide a more comprehensive approach to stock market volatility prediction.

Our approach thus involves using a similar model and LSTM architecture to the one proposed by Kim and Won in 2018 [2], but to train it on Optiver’s dataset which provides a more accurate data and a lot more important features for this problem. As we have already seen, we also enhance this data using data analytics and feature engineering algorithms. We expect that utilizing superior and more representative data, as provided in this dataset, will result in the development of a more effective volatility prediction model.

1. Experimental results analysis

By using an architecture inspired by the state-of-the-art model proposed by Kim and Won [2], and training it on such a complete and accurate dataset like Optiver’s one, we expect our results to approach or even reach the current state-of-the-art for this dataset. The current best performing models on Optiver’s dataset achieve a RMSPE score of around 0.19548. Out of the few thousands models present in the leaderboard, the top one hundred models achieve a score of at least 0.21. We expect our model to score around these levels. For instance, one very well performing hybrid LSTM-based architecture currently present in Optiver’s leaderboard, proposed by three students from the University of Beijing, managed to obtain a RMSPE score of 0.21656, being the 36th model on the leaderboard. On the other hand, from our observations, single ARCH or GARCH-based models usually do not manage to obtain higher scores than 0.30-0.35 on this dataset. From the study of Kim and Won [2], similar LSTM and GARCH hybrid architectures usually tend to perform around 30% better compared to simple GARCH-based architecture, which further motivates our expectations for our results.

From a quantitative analysis point of view, we already mentioned the usage of performance metrics such as the RMSPE score. In terms of qualitative analysis, we examine the distribution and patterns of residuals to gain qualitative insights into the model's behavior, while also identifying areas where the model may struggle or exhibit systematic errors. Additionally, we qualitatively compare the performance of our LSTM-GARCH model with baseline models or alternative approaches, like simple GARCH models, to better understand the unique strengths and weaknesses of the proposed model.

The incorporation of additional data analytics algorithms, as mentioned earlier, is expected to enhance the overall predictive capabilities of the model. In summary, our expectations are centered around achieving a more accurate, robust, and adaptive stock market volatility prediction model through the integration of LSTM and GARCH, trained on Optiver's Realized Volatility dataset. The combination of short-term and long-term modeling capabilities, coupled with the utilization of a comprehensive dataset, positions our model to deliver insightful and reliable volatility forecasts for diverse market conditions.

1. Conclusions and future work

In conclusion, in this experimental research report we proposed using a LSTM-GARCH based architecture for solving the stock market realized volatility problem, using the realized volatility dataset provided by Optiver. This approach has proved itself to be very performant on other less complete datasets, and now, in conjunction with small refinements on the architecture and data analytics algorithms applied, we use and train it on Optiver’s dataset, expecting to see an enhancement in the model’s performance.

Future work based on our findings might include the combination of additional ARCH-based econometrics models with our GARCH-LSTM architecture, as other researchers have also tried in the past. Additionally, we might explore the concept of model stacking, where predictions from multiple ARCH-based models and the LSTM + GARCH model are combined using a meta-model. Weighted averaging of the predictions based on the historical performance of each model could lead to an ensemble model that outperforms individual components in diverse market scenarios. Moreover, training models for specific markets and adapting them to the characteristics of each market might also be a promising idea in the future.

1. Bibliography

[1] Shah, D., Isah, H., Zulkernine, F., 2019. Stock market analysis: A review and taxonomy of prediction techniques. International Journal of Financial Studies 7, 1–22.

[2] 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 Syst. Appl. 103, 25–37.

[3] Black, F., and Scholes, M., 1973. The pricing of options and corporate liabilities. Journal of political economy, 81(3), 637–654.

[4] Bollerslev, Tim, 1986. Generalized Autoregressive Conditional Heteroskedasticity. Journal of Econometrics, 31 (3), 307–327.

[5] Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9, 1735–1780.

[6] Kristjanpoller, W., Fadic, A., Minutolo, M.C., 2014. Volatility forecast using hybrid neural network models. Expert Systems with Applications 41, 2437–2442.

[7] 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.

[8] Kristjanpoller, W. and Hernández, E., 2017. Volatility of main metals forecasted by a hybrid ANN-GARCH model with regressors. Expert Systems with Applications 84, 290–300.