T. C.

BAHCESEHIR UNIVERSITY

NON-THESIS MASTERS PROGRAM

TURBULANCE OF MARKETS

BIG DATA ANALYTICS AND MANAGEMENT

UFUK ALTAN

ISTANBUL, 2022

T. C.

BAHCESEHIR UNIVERSITY

NON-THESIS MASTERS PROGRAM

TURBULANCE OF MARKETS

BIG DATA ANALYTICS AND MANAGEMENT

UFUK ALTAN

Project Advisor: Dr. Rüştü Murat Demirer

ISTANBUL, JULY 2022

SUMMARY

World's markets are considered to have random movements or affected by variables like wars, crisis, pandemics that can't be calculated for certain. It is very controversial whether fundamental analysis or technical analysis should be utilized on the markets. There are many studies on using machine learning methodologies for financial analysis and stock market price prediction. However, it is usually considered non-beneficial since stock or crypto market does not have a specific pattern. Therefore, rather than utilizing pattern recognition it is possible to build a trading strategy that could determine the lows and highs of the market. To analyze different markets including indices, stocks, crypto, commodity, metal and currency, Yahoo Finance and Binance APIs will be used as data sources. To analyze market movements, causality analysis will be performed using Granger Causality Test and Spearman Correlation Coefficient. Results of causality will be plotted to a causality network graph indicating the possible relationships between markets. Later, VARMA algorithm will be applied using the information that is acquired from causality tests. Simulations will be performed for technical analysis, combining Correlation Analysis and Stochastic RSI to see how it performs in different crypto coins. In the end, it would be possible to have an idea on which techniques are more advantageous on understanding the turbulence of markets and succeeding in the investment world.

ABSTRACT

Even though financial markets are considered random and unpredictable, it is possible to recognize some patterns that repeats when historical data is examined. In this project, two different investment perspectives are performed and tested. First, variety of market data is aggregated, Granger Causality Test is applied, and VARMA algorithm is used to predict the future price while utilizing the data of different markets. Second, a momentum indicator stochastic RSI and Spearman Correlation Coefficient are used to build a trading strategy called Stochastic Swing and simulated on the crypto market to measure the results.

Keywords: Machine learning, finance, technical analysis, causality

Table of Contents

Table of Tables…..……….……….……….……….……….……….……….……………..**5**

SYMBOLS / ABBREVIATIONS

AI: Artificial Intelligence
ML: Machine Learning
DL: Deep Learning
ARIMA: Auto Regressive Integrated Moving Average
SARIMA: Seasonal Auto Regressive Integrated Moving Average
VARMA: Vector Auto Regressive Moving Average
NARMAX: Non-linear Auto Regressive Moving Average Exogenous
ANN: Artificial Neural Networks
LSTM: Long Short-Term Memory
GRU: Gated Recurrent Unit
RSI: Relative Strength Index
ADX: Average Directional Index
STD: Standard Deviation
MDN: Modular Neural Networks
BP: Back Propagation
GRNN: Generalized Regression Neural Network
MLP: Multilayer Perceptron
FFBPNN: Feed Forward Back Propagation Neural Network
RBFN: Radial Basis Function Network
TD3: Twin Delayed Deterministic Policy Gradient

1. INTRODUCTION

There has been many projects, tasks, and theories on trading in different markets like stock market and crypto currencies using statistical techniques. Some methods may provide profitable results in the long run. It is possible to simulate a specific trading strategy on the past data, however, all markets may vary in their patterns on different years thereof simulation may not be reliable if tested on a single market.

In finance, technical analysis techniques are applied to understand the market movement. Financial analysts assume historical patterns tend to repeat themselves and reflect the movement on the market price. There are variety of technical analysis techniques like momentum, volatility, trend, and volume.

In data science and machine learning, time series analysis and forecasting techniques are applied whether to predict possible future values of the data, hence the market price. Time series is a well-studied area of statistics and machine learning. Algorithms may include general forecasting techniques like ARIMA, SARIMA or VARMA which exclusively recognize patterns on stationary data. Furthermore, deep learning techniques like LSTM or GRU networks are considered very powerful and useful for accurate predictions.

Algorithms of technical analysis and machine learning can be combined in theory. Furthermore, utilizing the data of different markets for causality between their market movements may, in theory, establish new perspectives on the subject and increase accuracy. This project aims to create a new perspective and a potentially beneficial research subject that tests combination of different statistical techniques including technical analysis and machine learning in the concept of investment strategies.

1.1 Theoretical Frame

Stocks, currency, crypto and many other markets are highly influenced by expectation of investors. Factors like economic growth, technological advancements and political events could possibly cause different expectations for the market. It is also possible to state that contents on social media, especially Twitter, may build perspectives or curiosities like metaverse or blockchain.

As an example, USDT, the tether coin, is intended to has the same value as the American Dollar. However, as the crypto market falls dramatically, it is possible to state that USDT increases unusually compared to USD. As a result of a high expectation that crypto market would crush, investors may intend to hold their earnings on a more stable asset like USDT causing a possible imbalance between USD and USDT.

Would it be possible to understand the market movements based on expectations as these expectations reflect themselves on the prices of other markets? If so, can retrieving data of different markets and applying statistical tests to understand the relationship between these markets be assisting on achieving higher accuracy?

1.2 Problem State

This project will utilize Binance and Yahoo Finance APIs of Python programming language to retrieve different market data.

In the machine learning perspective, Granger Causality Test will be applied and possible causations between different assets will be placed into a causality network. This network could reveal several independent and dependent variables that would be utilized in a VARMA model. This model would reveal if it is possible to understand turbulence of markets when utilizing machine learning techniques.

In the technical analysis perspective, a momentum indicator called Stochastic RSI is used to understand lows and highs of the market in a specified interval. Furthermore, this indicator is combined with Spearman Correlation analysis to create a probabilistic strategy that could be used as a trading strategy which is named as Stochastic Swing.

First, Stochastic RSI signal is produced using the bounds of 20 & 70. Second, correlation is calculated between K value and the price for the several time points and a correlation threshold is determined. If value K crosses above value D when these values are lower bound of 20 while correlation values are above the specified threshold a buy signal is generated. On the other hand, when value D crosses above the value K, a sell signal is generated.

1.3 Purpose of the Study

It is a renowned challenge to understand market movements and build a profitable investment strategy. There are many studies on supervised regression models, reinforcement learning, time series analysis and financial technical analysis methods for trading in different markets.

One question is, is it possible that these markets have a considerable effect on one and another since market movements are highly affected by investors' expectations? As an expectation rises, transactions may increase among different assets. Therefore, creating a turbulence in markets.

Another question would be how effective would it be to involve different market data and applying machine learning techniques rather than using popular technical analysis methods? There are three possible decisions in investing into markets. Buy, sell, or hold. In such case, investment strategies are presented as a classification problem. In order to classify accurately, time series forecasting techniques will be applied on aggregated market data.

1.4 Hypothesis / Research Questions

Data science could respond with solutions in a variety of different fields including medicine, psychology, marketing, etc. As long as important variables and causations are understood and analyzed with proper methods, accuracy and, therefore, solutions can be acquired. However, when it comes to random data like market data, it becomes challenging to generate correct decisions.

There are different variables affecting market movements. Some might be accountable some may not be. Political events, unexpected pandemic or specific problems in the business environment like alterations on the company staff are examples of non-countable variables. On the other hand, investors may decide to switch their interests on different assets therefore creating a turbulence on markets. This might form new hypothesis to test and questions to research for.

- Is it possible that markets have a considerable effect on one and another since market movements are highly affected by investors' expectations?
- How effective would it be to involve different market data and applying machine learning techniques rather than using popular technical analysis methods?
- What statistical analysis methods could possible present accurate resuls on time series data for a classification task?

1.5 Importance of the Study

There are considerable number of studies and projects on the subject including simulations, automated trading bots and time series forecasting. Some of these sources may not be considered academical like medium articles or Udemy courses. On the other hand, there are number of academic studies investigating effects of supervised classification algorithms on different indices of different countries and time series forecasting algorithms on stock data. Furthermore, besides the academic articles, personal hypothesizes is developed and tested utilizing different perspectives.

Considering various strategies and hypothesis developed in the subject, it can not be said that there are fully reliable strategies on trading. Risk is inevitable on the face of the unpredictable random movements of markets. This study may develop possible investment strategies and also produce creative perspectives on the subject for further academical research.

1.6 Definitions

Artificial Intelligence: Development of intelligent systems that can implement human tasks and learn through experience.

Machine Learning: Sub-field of artificial intelligence. It is a field utilizing mathematical and statistical techniques to process data in order to establish intelligent computers that can learn without being explicitly programmed.

Deep Learning: Deep learning, which is simply a neural network with three or more layers, is a subset of machine learning. These neural networks make an effort to mimic how the human brain functions, however they fall far short of being able to match it, enabling it to "learn" from vast volumes of data.

Data Analytics: Analyzing data collections to identify trends and make judgments about the information they contain is known as data analytics (DA).

Fundamental Analysis: The analysis of market data for long term investment, utilizing information of macro economy, political events and busines understanding.

Technical Analysis: The analysis of market data for short or middle term investing, utilizing technical indicators that are derived from original price.

Time Series Forecasting: Forecasting or predicting the future value over a period is known as time series forecasting. It involves creating models based on historical data and using them to draw conclusions and direct future tactical decisions.

Python: Python is a well-liked general-purpose programming language that has a broad range of uses. High-level data structures, backend development, data science, and many other capabilities make it equally helpful for scripting or "glue programming" that ties components together as they do for developing advanced applications.

Numpy: A library for the Python programming language, that supports multi-dimensional arrays and matrices along with a substantial number of high-level mathematical operations that may be performed on these arrays.

Pandas: The most often used open-source Python library for data science, data analysis and data manipulation. It is constructed on top of the Numpy library.

Statsmodels: Statsmodels is a Python module that provides classes and functions for estimating various statistical models, performing statistical tests, and examining statistical data. It is considered qualified for time series analysis.

Seasonality: It is a continuous upward and downward trend that repeat itself after a fixed interval of time.

Stationarity: Stationarity implies that taking consecutive samples of data with the same size should have identical covariances regardless of the starting point.

Augmented Dickey-Fuller Test: In statistics and economics, it is a test for a unit root in a time series sample. It is used in Python for measuring stationarity.

Causality: Causality is the process by which one event, process, state, or object influences the development of another event, process, condition, or object, where the cause and effect are both somewhat influenced by one another.

Correlation: It is a statistical test that is utilized to measure the degree of the relationship between two numerical variables. Different correlation test could be utilized whether it is a parametric or non-parametric test depending on the probability distribution of the data.

Granger Causality Test: A statistical test for assessing whether one time series may be used to predict another.

ANN: Artificial neural networks, more commonly known as neural networks or even just neural nets, are computer architectures that draw inspiration from the biological neural networks that make up human brains.

ARIMA: A statistical analysis model called as autoregressive integrated moving average, uses time series data to either better comprehend the data set or forecast future trends.

Momentum: Momentum and rate of change are two straightforward indicators used in financial technical analysis that show the difference between today's closing price and the close N days ago. These technical indicators may provide information on overbought and oversold time points on the financial data

RSI: A momentum indicator used in the research of financial markets. Based on the closing prices of a recent trading session, it aims to chart the present and past strength or weakness of a stock or market.

Stochastic Oscillator: It is a momentum indicator, evaluates a security's closing price in relation to a range of prices over a predetermined period of time.

Stochastic RSI: The Stochastic RSI, a technical analysis indicator with a range of zero to one, is created when the Stochastic oscillator formula is applied to a collection of relative strength index (RSI) values rather than to conventional price data (or zero and 100 on some charting systems).

Volatility: Volatility in finance refers to how much a trading price series varies over time. Pace of price movement increases with volatility as price decreases or increases. However, volatility does not provide information related to direction or trend of the market.

ADX: It serves as a volatility indicator for a collection of prices of a financial instrument.

STD: It allows you to determine the range of variation or distance between a set of data.

2. LITERATURE

There are many different studies, analysis, and tests on understanding financial markets. Some common methodologies are using time series forecasting on the past data, utilizing technical analysis techniques to understand the movement of the market and also developing reinforcement learning algorithms to forecasting market movement with trial and error.

Using artificial intelligence techniques to anticipate stocks has been the subject of numerous empirical investigations. A stock on the Taiwanese market may be predicted with an accuracy of 77 utilizing neural network and decision tree models. Stock market can accurately be forecasted for the following day by 74.4 percent for the NASDAQ, 77.6 percent for the S&P, and 77.6 percent for the DJIA (Tsai and Wang, 2009).

Comparing various machine learning algorithms for financial time series forecasts, neural network approach is the most successful. The absence of studies using XU100 stock data was discovered during the study process (Krollner, Vanstone, and Finnie, 2010).

An artificial neural network (ANN) is employed to forecast the general stock index on the Santiago de Chile Stock Exchange. The ANN was trained using the index's daily values, the total number of transactions, and time series. Results from combined ANN outperformed those from ANN with a simple architecture (Bengoechea, 1996).

Various neural network categorization topologies to represent the Standard and Poors 500 Index in this study. The slope, decline, or stability of the index is predicted in this study using a probabilistic neural network and a multilayer perceptron architecture. The trading outcomes are then compared to the highest potential performance and the performance of the index using the network's recommendations. The outcomes demonstrated that both networks can outperform the array, with the probabilistic neural network multilayer perceptron performing somewhat better (Schierholt & Dal, 1996, p. 72).

The market price data of Tokyo Stock Exchange, TOPIX, was forecasted using past data and renowned technical indicators. 33 months of index data is included from 1987 to 1989. Utilizing the method, a decision system was developed to aid investors. The method included Modular Neural Networks (MDN) and multi-variable regression model, and comparison of two models. ANN model presented almost two times more accurate results compared to the multi-variable regression analysis (Kimoto, T, 1990, p 1-6).

In 1991, using ANN and discriminant analysis stock markets are classified and compared. Discriminant analysis achieved a success rate of 65% while ANN achieved 77.5%. With the model that is built, stocks and, therefore companies, were able to be classified whether successful or not. The interesting part of the study was its independent variables that are utilized for the algorithm. Qualitative variables like trust and optimism were key indicators (Yoon, Y., & Swales, 1991, p. 156).

Techniques like conjugate gradient method, back propagation method and random optimization method are used to forecast the stock market price. As independent variables indices data and past data of the stock are utilized. A hybrid model would result in a better performance (Baba, N., & Kozaki, 1992, p. 371).

Various technical analysis techniques and ANN is used to forecast market movements of S&P 500, FTSE 100 and Nikkei 225 indices. ANN is performed better than other models. As a result of the analysis, authors decided that using deep learning algorithms are more successful at forecasting the direction of the market. However, degree of the movement can not be forecasted successfully (Leung, Daouk, 2000, p. 173).

Research is made on the Canadian stock market companies to classify and forecast possible earnings. A dataset is used that starts from 1976 and ends at 1993. Furthermore, 61 financial ratios are included, and forecast is made for the possible earnings on one year mark for different companies. Using same information over 2352 different companies that are from different indices and profit they might offer are forecasted. As algorithm, least squares method, logistics regression and artificial neural networks are tested and compared. ANN performed more accurate as usual. Considering the variations in different company data average classification result is 58.32 % and profit on the index was 16.41 % (Olson, D., & Mossman, 2003, p. 453).

Generalized Regression Neural Network (GRNN) and Multilayer Perceptron (MLP) is used to forecast market movements on the Kuwait stock market. For optimization, methods like Quick Propagition, Conjugate Gradient Descent and Quasi-Newton algorithm are tested. As a result of the research, Quasi-Newton algorithm and Multilayer perceptron had the lowest error in forecasts (Mostafa, 2010, p. 6302).

Three different neural network architectures are tested with past data alongside its technical indicator data as a hybrid model. Simple Feed Forward Back Propagation Neural Network (FFBPNN), Radial Basis Function Network (RBFN) and Elman Recurrent Network are tested and compared. FFBPNN had an accuracy of 92.33& on the stock market and achieved a far greater result than other models (Shah, M., Prabhu, N., & Rao, 2014, p. 8347).

Utilizing macro-economic factors as independent variables in regression algorithms including artificial neural networks monthly closing prices of stock indices are forecasted and tested using mean squared error. Similar to other research artificial neural networks are selected as the best performing model in conclusion of the study (Karaatlı, M., Güngör, İ., Demir, Y., & Kalaycı, 2005, p. 22).

Feed Forward Artificial Neural Networks are performed on the XU100 index. In order to achieve the task macro variables are differentiated to generate three different models. Best model resulted with 96% accuracy. When this model is compared with simple moving average, artificial neural network model presented higher accuracy results (Kutlu, B., & Badur, 2009, p. 25).

Three years of data between 1997 and 2000 of XU100 index is predicted in weekly prices. Using ANN feed forward networks and back propagation a standard error of 6% is achieved in predictions (Ulusoy, T, 2010, p. 21).

Utilizing different technical indicators as independent variables and artificial neural networks as machine learning algorithm, direction of XU100 index was predicted for the next day with a 60.81% accuracy (Diler, 2003, p. 66).

The direction of the index was attempted to be approximated on the XU100 index using the linear regression model and ANN. With prediction success rates of 57.8 percent daily, 67.1 percent weekly, and 78.3 percent monthly, ANN outperformed linear regression in terms of performance. (Altay and Satman, 2005, p. 18)

When feasibility of using ANFIS is analyzed to precisely predict the stock market index. The sample area, XU100, was selected, and an attempt was made to estimate the monthly index change. The study focuses on the period from January 1990 to December 2008. The root mean square was utilized by the authors as a performance indicator, and they used macroeconomic data as independent variables. The XU100 index was therefore computed using ANFIS at a rate of 98.3%, and the ANFIS approach was presented as a potential method for developing nations like Turkey (Acar Boyacıoğlu, M., & Avcı, 2010, p. 7908).

Based on fundamental firm data like earnings per share, price-earnings ratio, dividends, sales, and profit margin, new inputs and criteria are employed in the estimation. The primary drivers of a stock's price are these measures and indicators, particularly those that relate to earnings. It has been noted that the early findings are really encouraging. The outcomes of the neural network are always superior, it is concluded (Atiya & Talaat, 1997, p. 2112).

A study is conducted that compares statistical models, notably those used in time series analysis, regression-based forecasting, and decision making, with artificial neural networks. In this study, the potential of artificial neural networks for prediction and decision-making models will be evaluated fairly. Most of the research we analyzed used back propagation for neural network prediction. Alternative neural network models and the back propagation technique have also recently been offered for improvements. The same estimation and discretion in administrative chores will be needed for these enhancements and alternatives (Hill, 1999, p. 5).

Fuzzy neural networks and the conventional methods are compared in stock market crises between 1987 and 1998. While using traditional neural networks does not improve performance, it is discovered that the rules generate a more consistent prediction quality. The traditional neural network demonstrates that it outperforms the market for 1998 model sets and does well in informed (bull) markets. It has been discovered that fuzzy neural networks are more stable and thus less sensitive to changes in the structure of the global market (Rast, 1999, p. 418).

Multilayer neural networks were successfully used to forecast time series data. To address these issues, the conjugate gradient learning algorithm with the restart mechanism is presented in this paper. Using neural networks, the daily trading data of the companies traded on the Shanghai Stock Exchange is gathered for technical analysis. We compare two learning processes and two weight initializations. Results, weight, and learning algorithm. He discovered that, independent of initialization, neural networks successfully model time series (Chung, 2000, p. 61).

An application of artificial neural networks for the forecasting of stock index growth is created. The network model was put to the test using information from five significant stock market indices (DAX, DJIA, FTSE-100, HSI, and NASDAQ). The computational findings from five different financial markets revealed that the confidence region-based neural network model performed better than other neural network models in comparison to the outcomes. More specifically, it demonstrated that our model had an average success rate of more than 60% across the five exchanges in predicting the sign of index rises (Phua, 2003, p. 260).

Different neural network research approaches are attempted to predict stock returns 45 publications. Additionally collated and discussed are modeling methods and suggestions found in the literature. The findings demonstrate the potential of neural networks as a cutting-edge computational technique for upcoming studies (Thawornwong, 2003, p. 47).

In this paper, a fuzzy neural network—a type of adaptive networks that is functionally identical to a fuzzy inference system—is developed to address the drawbacks of neural networks. The experimental findings, which were based on the Shanghai Stock Exchange's comprehensive index, demonstrated that the suggested fuzzy neural network can be a useful tool for estimating financial time series. An empirical analysis has been conducted for the illustration to help make this point more evident. The proposed fuzzy neural network can be a useful method for estimating financial time series, according to the experimental results (Li Rong & Xiong, 2005, p. 3475)

Using daily returns in the IBEX-35 stock market index from December 30, 1989 to February 10, 2000, out-of-sample prediction performance of smooth transition autoregressive (STAR) models and artificial neural networks (ANNs) are examined. The outcomes demonstrate that different neural network specifications can predict more accurately than AR models and smooth transition non-linear models under statistical criteria. (Rodriguez & Torra, 2005, p. 490)

Various data preparation techniques to forecast market returns are applied to predict the following day. Over various time frames, forecast performance is evaluated. Additionally, the test set is used to evaluate the network's capacity to forecast changes in pricing (Zhora, 2005, p. 2549).

The goal was to demonstrate that the feed forward artificial neural networks approach can forecast the XU100 index. It was shown that the XU100 index value may also be successfully modelled with feed forward artificial neural networks through experiments using the data obtained between 2001 and 2006 from the websites of the Central Bank of the Republic of Turkey and other stock markets (Kutlu and Badur, 2009, p. 25)

A forecasting model is created for stock price prediction utilizing a fuzzy-neural model or method to show the opinions of experts using technical and fundamental indices. The empirical findings support the suggested model's superior performance in accurately predicting stock price. By assisting in making stock forecasts more realistic, this model seeks to improve the caliber of stock market investors' decisions (Adebiyi & Ayo, 2011, p. 3).

Two effective models are created to forecast the Istanbul Stock Exchange's XU100 Index and assess how well they performed in foretelling the movement of the index. The models are based on support vector machines (SVM) and artificial neural networks (ANN), two classification techniques (SVM). The experimental findings demonstrated that the ANN model's average performance (75.74 percent) outperformed the SVM model (71.52 percent). As a result, they demonstrated that ANN and SVM are both effective forecasting techniques for this subject (Kara, 2011, p. 5311).

Fuzzy logic was utilized to forecast future events in the US and Brazilian markets between January 2000 and October 2011. As a result, it has been discovered that the evolving models have excellent predictive power and that, when applied properly, fuzzy logic outperforms conventional models (Maciel, 2012, p. 20).

3. METHOD

Goal of the project is to determine possible relationships between different markets, mentioned as turbulence of markets. Relationships between variables can efficiently be analyzed using different plots or correlation tests in a supervised machine learning model. On the other hand, for time series data analysis relationships must be analyzed in terms of lagged values.

First, for correlation tests, various values could be specified for lag parameter which would present information related to different time periods. Hyperparameters tuning is a common practice in machine learning. Different lag values can be tested between different market data and correlation coefficients can be determined for these different values. As a result, relationships between different markets be presented mathematically.

Second, it is possible to indicate causality for different time series datasets using the Granger Causality Test. Similar practice to a correlation analysis, different lagged values can be tested and placed into a causality matrix. Since many markets will be analyzed and tested, a matrix would become a challenge to interpret. A more efficient practice would be to utilize plotting libraries of Python.

Third, using the results of the causation matrix, a causality network will be created. This network will consist of all markets that are accounted in causality matrix, however, only the markets with highest causality values will be plotted on lag value of 15. After a threshold is determined, highest values will remain, and a causality network will be plotted.

Finally, this causality network consists the information of several different markets that might have relationships in short period of time. Among these markets, one with the most causation will be selected for applying VARMA analysis. The question is, can a market price be predicted when utilizing the data of different X markets?

3.2 Research Model

In this project, there will be two different perspectives on understanding financial market movements and trading in short term.

First perspective is to utilize technical indicators on crypto data and building a simulation to generate buy and sell signals on specific dates. Later, prices on these dates will be utilized to calculate win and loss percentages and ratios. Simulation will start with a budget of $1000 and calculate its way through wins and losses to get a result in one year time interval.

In this simulation stochastic RSI and Spearman Correlation Coefficient are used to establish a trading strategy called Stochastic Swing. Results will be calculated using a custom created Python class and presented on a pandas data frame.

Second perspective would be to use machine learning models for time series forecasting purposes. VARMA algorithm from ARIMA family, is performed to calculate the relationship between two or more time series data to generate predictions on a single time series data.

In this perspective, a new task would be to determine markets that might influence each other. Granger Causality Test is a popular test to calculate causality between two time series data in a specified lag value. Using this test sum of squares chi square values will be calculated and first placed in a causality matrix and later a causality network.

3.2 Data Aggregation

a) Data Aggregation Tools: Market data could be drawn from variety of sources. A specific API or web crawling can be used for aggregating investment data. In this project, crypto data is collected from Binance API and other market data like commodities, stocks and indices are drawn from the Yahoo Finance API of Python.

b) Data Aggregation Process: To retrieve data from Binance API, a personal Binance account is required. Within this account, an api key and a secret key has to be generated These keys could be utilized read data, allow trades or even transfer assets between different accounts inside Python.

   On the other hand, retrieving data from Yahoo Finance API just requires an installation with pip command. yfinance is the Python library that includes options of daily, weekly, and monthly data requests for different markets.

c) Data Analysis: After data aggregation is completed, it is possible to acknowledge that different type of markets are open to trade in different time intervals. For example, crypto and currency markets are open 24 hours a day for 365 days while stock market is open only 252 days of the year. Furthermore, stock markets of Nasdaq and XU100 may be open or closed as a result of different holidays. In such obstacle, common dates are matched among all markets.

   In order to create a Granger Causality Matrix data has to be stationary. Therefore, a custom function has been created that apply differencing in time series data transforming it into a stationary pandas series. Later, Granger Causality Test is applied in a specified lag parameter. Finally, using the information acquired from the matrix a Causality Network is created. Using the network, the asset with the most connections is selected and processed for the VARMA analysis.

d) Validity and Reliability: The fact about financial markets is they are arbitrary. There could be random patterns occurring in different seasons and it is possible to find causalities between markets in specified lag values. However, it is crucial to remember that these calculations are mostly based on probabilities rather than certainties.

3.3 Limitations

There are many studies proving the effectiveness of building machine learning models. In terms of time series forecasting, concepts like stationarity, causality, seasonality are critical for acquiring accurate predictions. In the business environment, it is possible to observe seasonality especially in sales, production, or inventory fields.

On the other hand, in a stock data seasonality is not present. Thereof, an algorithm like SARIMA can not resolve a pattern for data analytics purposes. In the same way, other financial markets like commodities, crypto and currencies have random movements.

Some studies explain that deep learning algorithms like ANN and LSTM bring highly accurate predictions. However, when these algorithms are tested in a different time interval of the same market or in a completely different market, they might provide results with low accuracy. Therefore, albeit these algorithms ensure an accurate result, it may not be realistic in the long run.

4. FINDINGS

4.1 Data Aggregation

First, market data must be aggregated, analyzed, and preprocessed before applying statistical tests. When different stock markets of different countries are considered, there are null data being presented since markets are open or closed in different dates. This might be caused by time zone differences or vacation dates. In the same way, markets like crypto, currency and commodities are open 24 hours a day. In order to interpret the relationship between different markets datetime values must be matched and presented inside a single pandas data frame.

It is relatively not challenging to connect Yahoo Finance API and Binance API using Python. To connect Yahoo Finance API yfinance library must be installed and stock ticker must be configured. On the other hand, to connect Binance API and request crypto data a secret key and a api key are required. Getting these keys requires a Binance account. After getting the keys, they must be configured inside the Client class of binance.client library.

```
from binance.client import Client

api_key = ''
secret_key = ''
client = Client(api_key, secret_key)
```

There might be scenarios where markets are analyzed inside a single data frame or separately in different data frames to test and understand volatility or momentum of the market. Furthermore, technical analysis simulations can easily be done using a single market data where indicator data is prepared, and calculations are done simultaneously.

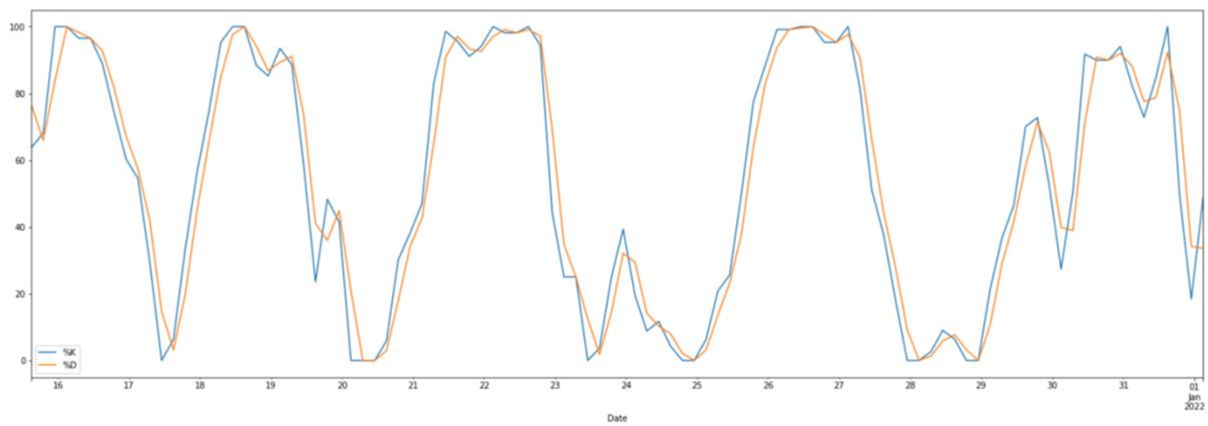

Figure 1 Stochastic RSI plot

To retrieve and process data in a more productive manner, a custom Request class is created. It receives tickers for retrieving data from Yahoo Finance API, coins to retrieve crypto data from Binance API, start and end dates of market data that is needed and finally, api and secret keys to access Binance API.

```python
class Request():

    def __init__(self, start, end, tickers, coins, api_key, secret_key):
        self.crypto_start, self.crypto_end = self.process_dates(start, end, source = 'binance')
        self.ticker_start, self.ticker_end = self.process_dates(start, end, source = 'yahoo')

        self.tickers = tickers
        self.coins = coins

        self.client = Client(api_key, secret_key)

    def process_dates(self, start, end, source = 'binance'):
        if source == 'binance':
            start = datetime(start[0], start[1], start[2]).strftime("%d %b, %Y")
            end = datetime(end[0], end[1], end[2]).strftime("%d %b, %Y")
        elif source == 'yahoo':
            start = datetime(start[0], start[1], start[2])
            end = datetime(end[0], end[1], end[2])
        return start, end

    def single_coin_data(self, coin):
        data = self.client.get_historical_klines(coin, Client.KLINE_INTERVAL_4HOUR, self.crypto_start, self.crypto_end)
        values = [[datetime.fromtimestamp(d[0] / 1000), float(d[4]), float(d[2]), float(d[3])] for d in data]
        coin_data = pd.DataFrame(values, columns = ["Date", 'Close', 'High', 'Low'])
        coin_data = coin_data.set_index('Date')
        return coin_data

    def single_ticker_data(self, ticker):
        stock = yf.Ticker(ticker)
        dataset = stock.history(start=self.ticker_start, end = self.ticker_end, period="1y")
        return dataset

    def request_coin_data(self):
        coin_data = {}

        for coin in self.coins:
            # Request data & preprocess
            data = self.client.get_historical_klines(coin, Client.KLINE_INTERVAL_1DAY, self.crypto_start, self.crypto_end)
            values = [[datetime.fromtimestamp(d[0] / 1000), float(d[4])] for d in data] # Date, Close
            dataset = pd.DataFrame(values, columns=["Date", 'Close'])
            # Fix index
            dataset = dataset.set_index('Date')
            dataset = dataset[dataset.index.day_of_week < 5]
            dataset.index = dataset.index.date
            # Create dictionary
            coin_data[coin] = dataset
        return coin_data

    def request_ticker_data(self):
        ticker_data = {}

        for ticker in self.tickers:
            # Request data & preprocess
            stock = yf.Ticker(ticker)
            dataset = stock.history(start=self.ticker_start, end = self.ticker_end, period="1y")
            dataset = dataset[['Close']]
            # Create dictionary
            ticker_data[ticker] = dataset
        return ticker_data
```

	BTCUSDT	AAPL	THYAO.IS	XU100.IS	^GSPC
2020-01-01	7200,85				
2020-01-02	6965,71	73,78592	14,83	115932,1	3257,85
2020-01-03	7344,96	73,06855	14,27	113684	3234,85
2020-01-06	7758	73,65079	13,57	111412,7	3246,28
2020-01-07	8145,28	73,30442	13,54	112599,9	3237,18
2020-01-08	8055,98	74,48362	13,5	112876,1	3253,05

Table 1: Example data

After data is retrieved, it is possible to observe empty rows. It is controversial to whether utilized median, mean or some other technique to fill in the missing values. However, since this is a financial time series data, best approach would dropping the null values and not complicating the market movements.

For this project 36 different financial market data are retrieved. Tickers are as follows. DOGEUSDT, ETHUSDT, BTCUSDT and MATICUSDT from Binance API, AEFES.IS, CLEBI.IS, ARCLK.IS, ASELS.IS, THYAO.IS as Turkish stock markets, AAPL, MSFT, GOOG, AMZN, FB and TSLA from Nasdaq Index, EURUSD, TRYUSD, GBPUSD, JPYUSD, CADUSD and USDSEK as currency markets, XU100.IS, XU030.IS, ^NYA, ^GSPC, ^DJI, ^IXIC, ^XAX, ^BUK100P, ^RUT, ^VIX, ^FTSE, ^N225 as indices from different countries and finally RIO, NUE, WPM as metal market tickers.

4.2 Data Analysis

Analysis part includes understanding the volatility, momentum, and causality of different market data. In this part, volatility is calculated through standard deviation and average directional index, momentum is calculated and plotted with stochastic RSI components K and D, finally causality is calculated using Spearman Correlation Coefficient and Granger Causality Test.

4.2.1 Volatility

Volatility can be calculated and analyzed through different technical indicators. Most popular tools might be bollinger bands, standard deviation, average true range (ATR) and average directional index.

Standard deviation can be calculated, easily with .std() function in pandas data frames. On the other hand, to calculate average directional index high and low columns of the price data. Therefore, it is useful to create a function retrieve single financial data instead of an assembled one. Volatility can be calculated and interpreted in different time intervals. As an example, MATICUSDT coin will be analyzed in time intervals of 5, 9, 14 and 21.

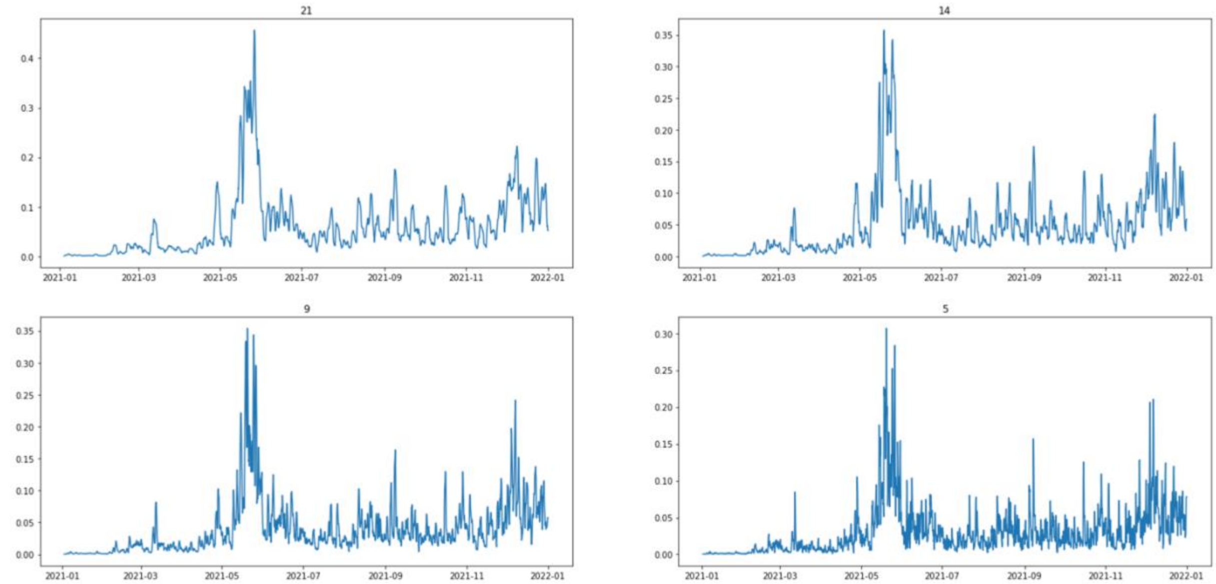

Figure 2: Standard Deviation in different periods

Plot above presents the standard deviation between January 2021 and January 2022. It is possible to state that similar to the price itself volatility is also varying over time. Considering the values generated by the closing price, it is not reasonable to stress a relationship between price and its volatility. Volatility does not present or provide information related to the direction of the market; however, it presents an increase or a decrease on movement of the market.

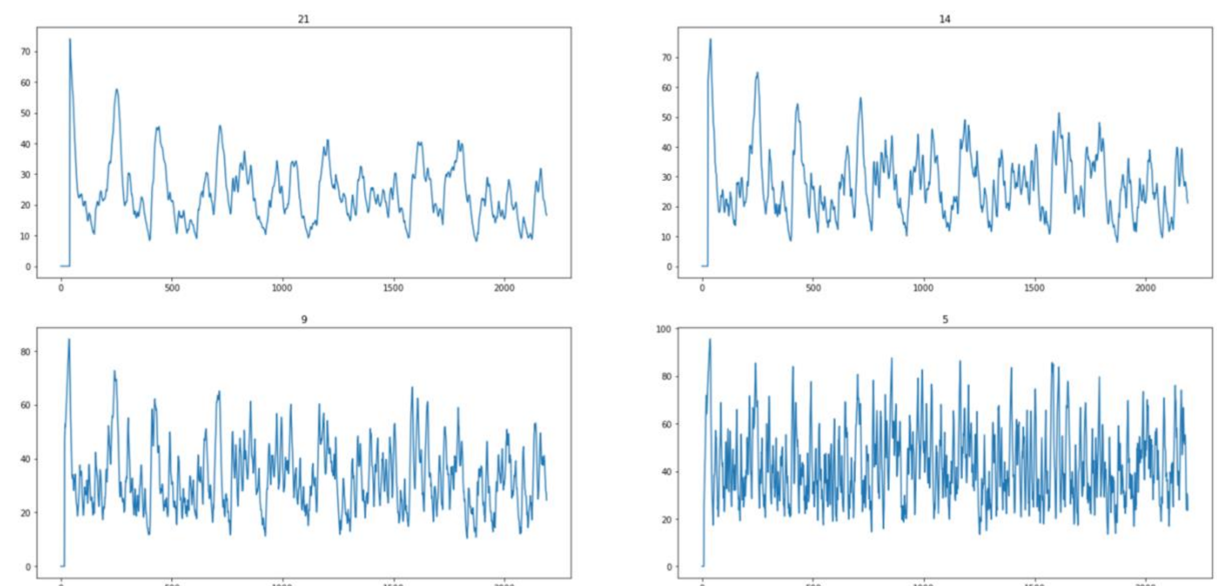

Figure 3: Average Directional Index in different periods

Average directional index, on the other hand, could be stated as more stationary and balanced compared to the standard deviation. Same 5-, 9-, 14- and 21-time intervals are used, and plotted on a line chart. However, price data of the MATICUSDT also does not present a reliable relationship with the ADX data. Volatility data may not be used as a predictor for machine learning algorithms but may establish a trading strategy in different perspectives in the near future.

4.2.2 Momentum

Momentum indicators could provide buy or sell signals in the specified time intervals. It is probabilistically assisting to decide whether price has decreased or increased in the short term. Popular momentum indicators are relative strength index, stochastic oscillator, Connors RSI and stochastic RSI. In this project, stochastic RSI components K and D are used to understand market movements.

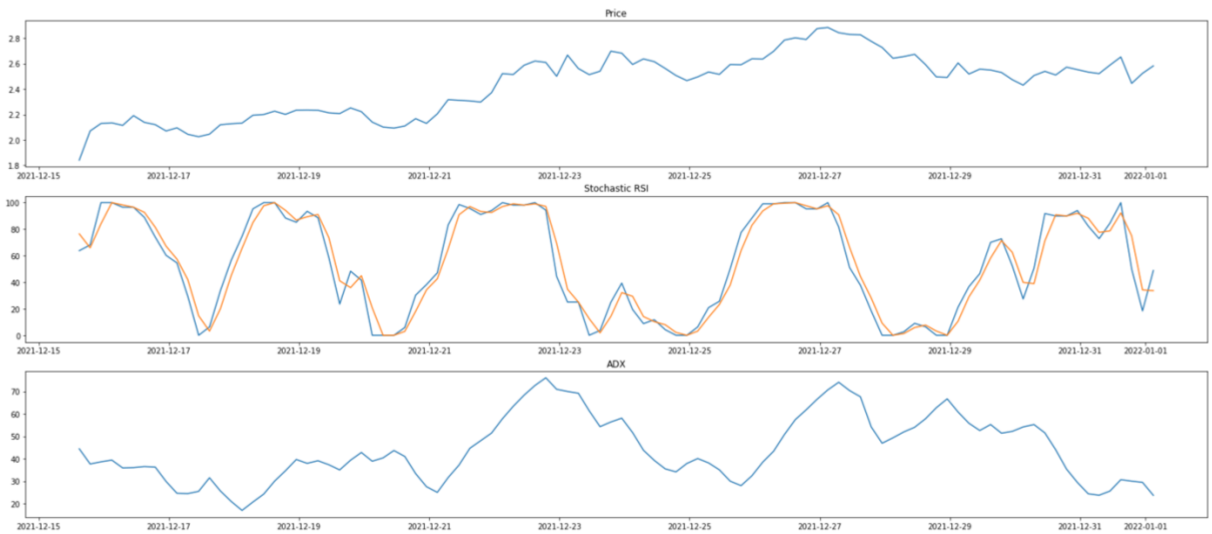

Figure 4: Closing Price, Stochastic RSI and ADX

The plot above shows closing price of MATICUSDT, its stochastic RSI parameters and average directional index data. There is not a linear relationship between indicators and the closing price.

However, looking at the K and D parameters of stochastic RSI, as blue line crosses above the orange one price tend to increase in a short period of time. There is no indication on the degree of the increase or specific time for market direction to alter. Therefore, there is not a high correlation coefficient between stochastic RSI and the closing price.

In a different perspective, as the price decreases and gets closer to possible swing moment, there could be a possible relationship between the parameter K and the closing price. To measure the relationship between the two in a strictly short time interval, Spearman Correlation Coefficient is used. To clarify, correlation coefficient between the last few days of closing price data and the parameter K can be calculated and utilized as a threshold to generate buy or sell signals.

In such method a trading strategy is established which is named as the Stochastic Swing. It uses stochastic RSI K parameter and Spearman Correlation Coefficient to provide trading decisions.

4.3 Simulation

It is popular to create back tests before creating automated trading bots that utilizes technical indicators. In this simulation, a trading strategy named Stochastic Swing has been performed. In simulate() function, data is requested, stochastic RSI components are calculated, correlation coefficient is calculated and trading decisions are generated in specific dates and

```
class StochasticSwing(Request, Momentum, SRSI, Strategy, Arrange, Evaluation):

    def __init__(self, start, end, tickers, coins, api_key, secret_key):
        super().__init__(start, end, tickers, coins, api_key, secret_key)
        self.coins = coins

    def simulate(self, params = params):
        result, progress, orders, tables, signals  = {}, {}, {}, {}, {}
        for coin in self.coins:
            print(coin)
            # Request
            coin_data = self.single_coin_data(coin)
            # Momentum
            coin_data['%K'], coin_data['%D'] = self.srsi(coin_data['Close'], period = params['period'], smoothK = 
params['smoothK'], smoothD = params['smoothD'])
            # SRSI
            coin_data['S-RSISignal-1'] = self.srsi_signal(coin_data, bottom = params['bottom'], top = params['top'])
            # Strategy
            signal = self.stochastic_swing(coin_data, interval = params['interval'], timestamp = params['timestamp'], 
threshold = params['threshold'])
            # Arrange
            data = self.arrange(signal)
            # Evaluation
            money_change = self.measure_return(data, money = params['money'])

            orders[coin] = data
            progress[coin] = money_change
            tables[coin] = coin_data
            signals[coin] = signal

        if params['results'] == 'all':
            return orders, progress, tables, signals

        return orders, progress #, tables, signals

    def return_table(self, orders, progress):
        # Evaluation
        results = self.return_results(orders, progress)
        print()
        print('DONE')
        return results
```

Date interval is specified as January 2021 to January 2022. This simulation is done in the crypto market with more than 15 different crypto coins. Correlation coefficient threshold is specified as .4 and stochastic RSI time interval as 10.

Coin	Win	Loss	WinPercentage	MinValue	MaxValue	FinalResult
SOLUSDT	40	23	63,492	1157,723	11316,12	9826,073
ONEUSDT	46	22	67,647	1186,465	119900,3	107054,6
XRPUSDT	39	27	59,091	1375,997	3521,322	2903,886
HOTUSDT	53	32	62,353	849,199	11205,46	9148,972
FTMUSDT	40	27	59,701	698,882	3024,614	2714,619
XLMUSDT	43	19	69,355	901,961	2292,006	2153,549
AVAXUSDT	38	26	59,375	1201,712	6752,793	5323,544
ATOMUSDT	45	26	63,38	1075,044	3827,089	3411,163
LINKUSDT	39	19	67,241	1044,057	2362,349	2262,789
SANDUSDT	40	35	53,333	937,605	10816,04	9399,011
BNBUSDT	44	20	68,75	1038,315	5159,556	4111,047
ADAUSDT	42	25	62,687	1087,462	4490,647	3291,191
ENJUSDT	40	23	63,492	884,846	4035,957	3438,53
BTTUSDT	32	25	56,14	973,002	2317,261	1032,284
DOGEUSDT	32	30	51,613	883,015	77801,1	55937,89
ETHUSDT	45	21	68,182	1095,726	3565,095	2975,775
BTCUSDT	50	18	73,529	905,202	2410,517	1981,666
MATICUSDT	38	24	61,29	965,434	3552,756	3357,264

Table 2: Stochastic Swing results

The table above presents, information related to stochastic swing trading results of one year. Starting budged was assumed to be $1000 for each coin. Coin column shows the name of the coin, Win and Loss columns shows amount of trade that resulted with a gain or loss, WinPercentage column shows the ratio of wins for each coin, MinValue and MaxValue are the minimum and the maximum values that coins reached during a year of trading, and FinalResult column shows the very final output of the trading in one year.

According to the table, ONEUSDT and DOGEUSDT have the highest result values. BTCUSDT and XLMUSDT have the highest win percentage results. It is critical to remember that these results are achieved during the rise of the crypto market. Furthermore, DOGEUSDT is well known to be manipulated and, therefore, its results might be considered less reliable.

It is also a trade off considering which metric to rely on more. Win percentage shows that the invested amount may have a steady increase over time, on the other hand, final result has its temptation to take higher risk for a higher profit.

4.4 Causality Matrix

Causality is a renowned concept in statistics. In time series data, a common test is Granger Causality Test. This test may reveal a possible relationship between two time series in a specified lag value. In most machine learning models, on the other hand, correlation tests are applied to understand relationships between different variables. A well-known issue about correlation is, it is not the same as causation.

When correlation exists, there are four possible reasons. Variable A might be causing variable B, variable B might be causing variable A, both variables might be caused by a third variable C or it might just be a coincidence. Therefore, causality can not be stated confidently when a high correlation coefficient is calculated.

4.4.1 Correlation

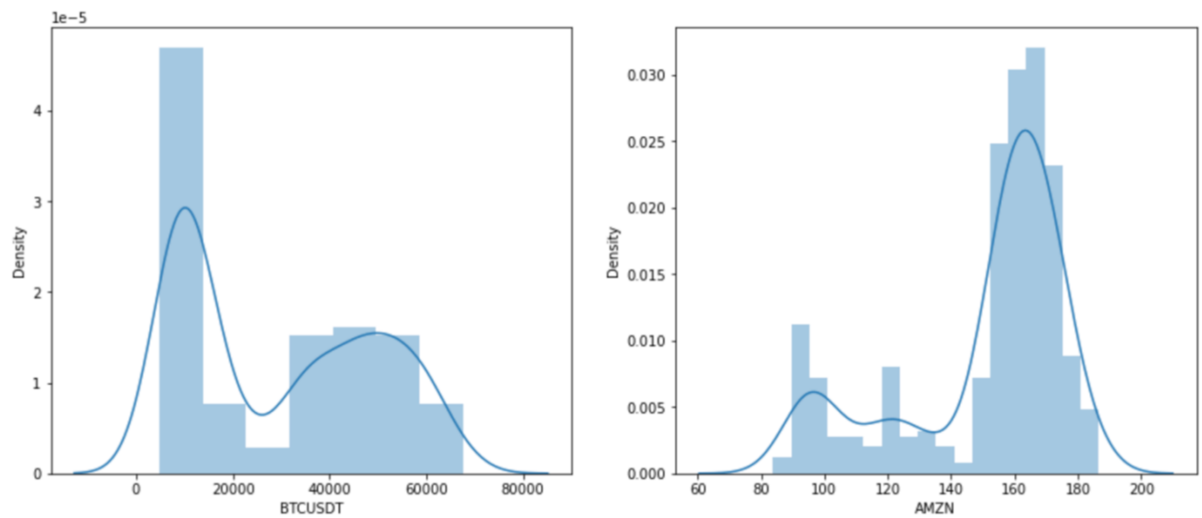

Figure 5: BTCUSDT and AMZN Probability Distribution

Most financial markets do not have a normally distributed data. As normality decreases and heteroscedasticity increases it becomes more challenging to end up with accurate predictions. In linear regression model, logarithmic or exponential transformations could be performed and tested, however, in a time series model this would not be efficient. Furthermore, since data is not distributed normally, it would be more sufficient to apply a non-parametric test like Spearman Correlation.

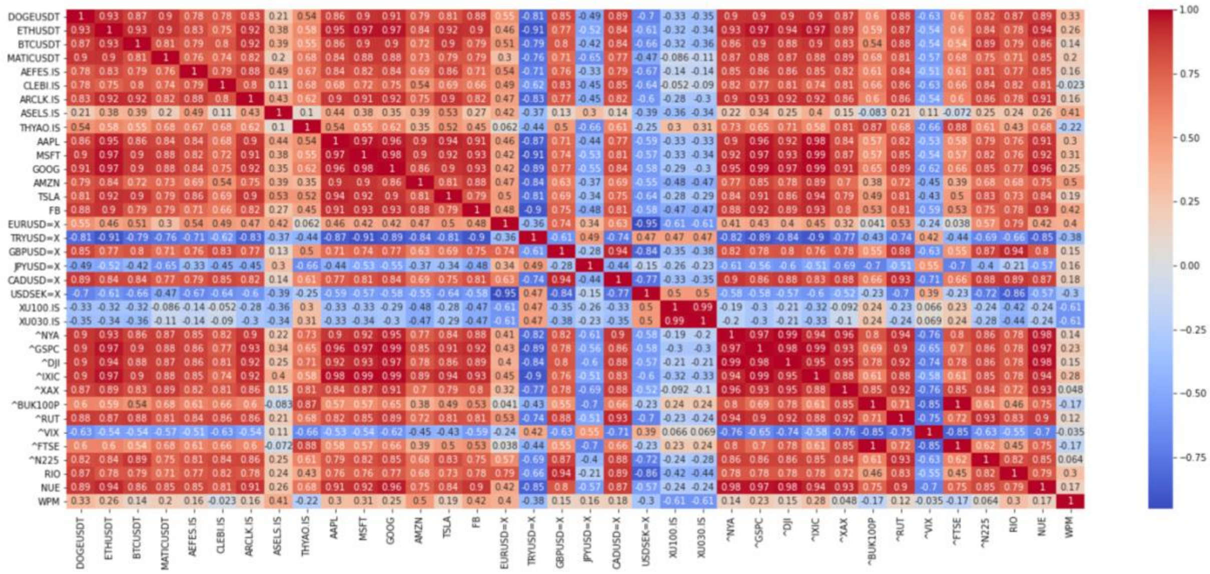

Table 3: Spearman Correlation Heatmap

There are 36 different markets that are included in the correlation matrix. These markets include stock markets, currencies, crypto and more. It is possible to observe that coefficients have high values.

Crypto coins have over .8 coefficients with each other. It is not surprising that most coins, in fact, move in the same direction as bitcoin. Interestingly, renowned indices like Dow Jones and S&P 500 have very high relationships with the crypto coins. Turkish indices XU100 and XU030 have high positive correlation with each other but have negative correlation with all turkish stocks.

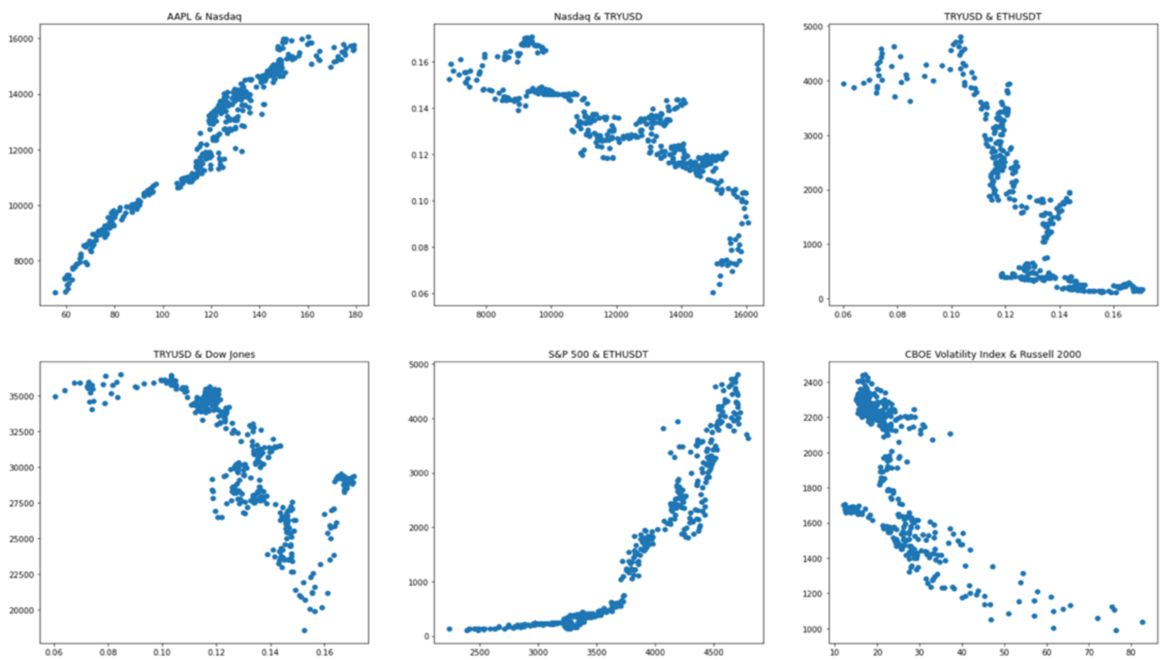

Figure 6: Plotted relationships between various markets

When scatter plots are used, it could be stated that most markets have non-linear relationships even though correlation coefficients are promising. Apple and Nasdaq have very straight linear graph which is not very surprising. On the other hand, TRY-USD data has a negative relationship with Dow Jones, Nasdaq and also Ethereum, and it is possible state that these markets have relationship with each other.

Coefficient values are promising in terms of generating a prediction algorithm. However, correlation is calculated with very same time stamps of each market. Meaning data of one market could be utilized the predict current price of another market which is not assisting at all. In such case, relationships of markets should be evaluated through lagged values of different markets.

Therefore, the first challenge would be deciding and optimizing the lag parameter between different markets. The second challenge would be to decide which markets could be evaluated as an independent variable or dependent variable. To evaluate markets, Granger Causality Test can be applied. But to apply the test, data must be preprocessed first.

4.4.2 Stationarity

Stationary time series data implies that mean and standard deviation of the data does not change over time. Sometimes it is possible to spot stationarity from looking at the time series data on a line plot. Seasonally increasing or decreasing trends imply that data is not yet stationary.

In ARIMA models, integration term (I) refers to the d term which is the differencing term of the algorithm. To apply ARIMA models or Granger Causality Test data has to be stationary. d term implies the how many differencing operations does the data require.

Another way to measure whether a time series data is stationary or not is to use Augmented Dickey Fuller Test. Statsmodel library of Python provides the built in function of adfuller() that could apply a hypothesis testing on a pandas data frame. If p value is smaller than the threshold value 0.05, data is assumed stationary.

Utilizing the adfuller() function in Python stationary can be measured and later ensured when differencing is applied on the data.

```
from statsmodels.tsa.stattools import adfuller


def create_stationarity(df):
    stationary_df = pd.DataFrame()
    df_nnv = df.dropna()
    for column in df.columns:
        transformed_df = df_nnv[column]
        result = adfuller(transformed_df)
        p_value = result[1]
        while p_value > .05:
            transformed_df = transformed_df.diff().dropna()
            result = adfuller(transformed_df)
            p_value = result[1]
        stationary_df = pd.concat([stationary_df, transformed_df], axis = 1)

    return stationary_df
```

4.4.3 Granger Causality Test

Granger Causality Test is applied to observe whether a time series data is causing the other in the specified lag parameter. To see the test results, independent and dependent time series data should be placed into a matrix. To present the results in a matrix, custom function is prepared that utilizes the statsmodels tools for Granger Causality Test. This function utilizes sum of squares chi square test as a causality statistic.

```
from statsmodels.tsa.stattools import grangercausalitytests


def grangers_causation_matrix(data, variables, maxlag = 15, test='ssr_chi2test', verbose=False):

    df = pd.DataFrame(np.zeros((len(variables), len(variables))), columns=variables, index=variables)
    for c in df.columns:
        for r in df.index:
            test_result = grangercausalitytests(data[[r, c]], maxlag=maxlag, verbose=False)
            p_values = [round(test_result[i+1][0][test][1],4) for i in range(maxlag)]

            if verbose: print(f'Y = {r}, X = {c}, P Values = {p_values}')

            min_p_value = np.min(p_values)
            df.loc[r, c] = min_p_value

    df.columns = [var + '_x' for var in variables]
    df.index = [var + '_y' for var in variables]
    return df
```

Independent variable is marked with '_x' and dependent variable is marked with '_y'. Since relationship between 36 different market data is calculated, A huge matrix is presented as a pandas data frame. To analyze and select the most potential markets on the matrix a causality network will be created.

4.5 Causality Network

Granger Causality Test generates a value between 0 and 1. These values mostly range between 0 and 0.2. A network graph consists of all markets and connections between them. If all markets are included on the plot, it would not be possible to differentiate the most useful causality numbers. Therefore, a threshold must be determined. For example, highest 10% of the data or chi square values above 0.5.

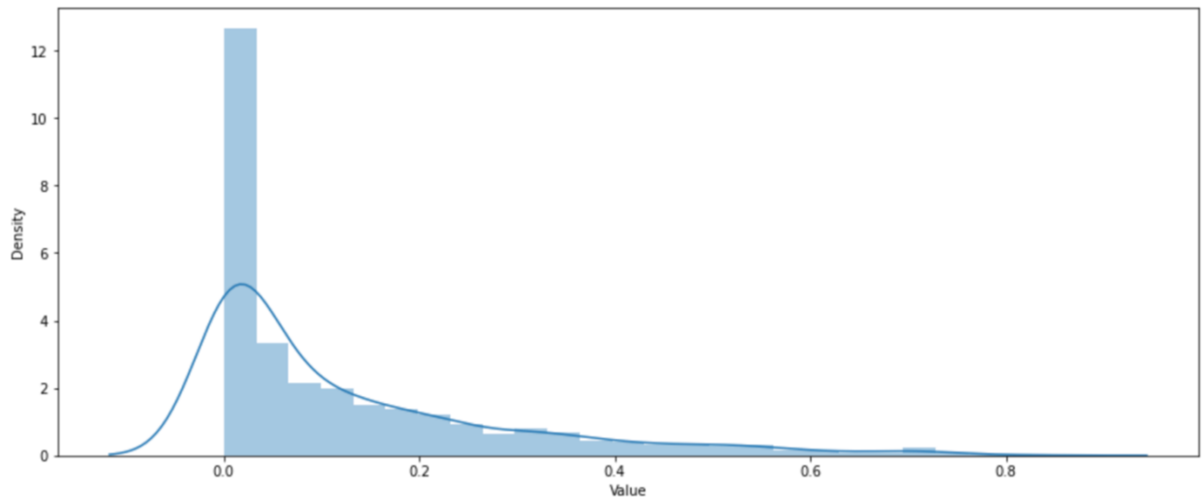

Figure 7: Granger Causality Test Results Distribution

Using 10% as the threshold, instead of using 1260 data only 126 rows are used for the plot. As a result, chi square value threshold is determined as 0.3615 and above. Lowest value seems to be between Tesla and S&P 500 as independent and dependent time series data respectively.

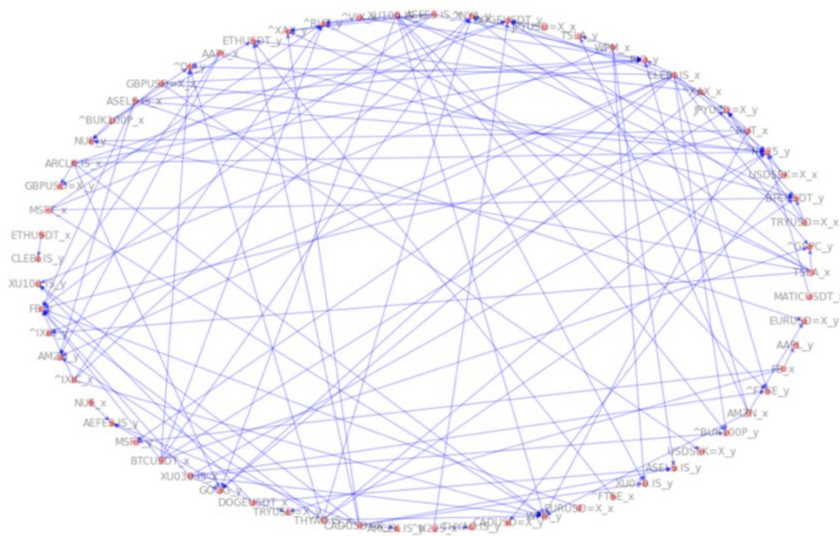

Figure 8: Network Graph 10% Threshold

When 10% percent of the data is used, network graph does not present information clearly. Therefore, a new threshold determined as chi square value 0.5 and above. When these top values are selected on 167 rows remained for the final network graph.

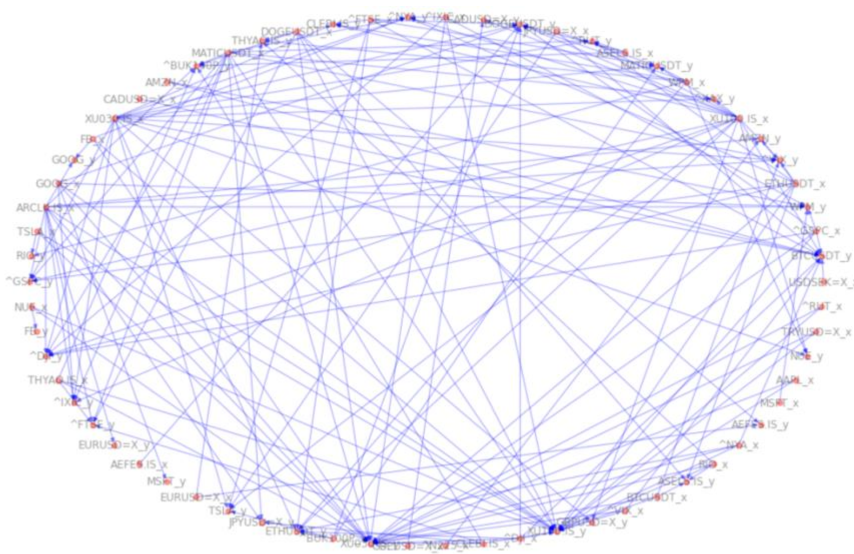

Figure 9: Network Graph 0.5 Threshold

Utilizing the variables acquired from the network graph, a dependent variable and multiple independent variables can be determined. For this project, BTCUSDT is selected as the dependent variable and 16 independent variables are selected including TRYUSD, XU030, S&P 500, Dow Jones, Amazon, and others.

4.6 VARMA – Vector Auto Regressive Moving Average

VARMA – Vector Auto Regressive Moving Average is an ARIMA based algorithm that receives multiple time series data as inputs and generate future predictions while assuming these series are causing each other. With a specified lag parameter, causations are measured between markets using the Granger Causality Test. It is always crucial to remember that these causation values could be nothing more than coincidences. Therefore, predictions have to be tested.

	DOGEUSDT	ETHUSDT	BTCUSDT	MATICUSDT	USDSEK=X	CADUSD=X	JPYUSD=X	TRYUSD=X	XU030.IS	XU100.IS	AMZN	^FTSE	^VIX	^GSPC	^NYA	^DJI
2020-01-02	0,001995	127,19	6965,71	0,01467	9,3394	0,770832	0,009199	0,168172	140681,5	115932,1	94,9005	7604,3	12,47	3257,85	14002,49	28868,8
2020-01-03	0,002024	134,35	7344,96	0,01512	9,3614	0,770238	0,009213	0,167906	138025,3	113684	93,7485	7622,4	14,02	3234,85	13917,05	28634,88
2020-01-06	0,002134	144,15	7758	0,01554	9,40968	0,770024	0,009262	0,167561	135514,1	111412,7	95,144	7575,3	13,85	3246,28	13941,8	28703,38
2020-01-07	0,002175	142,8	8145,28	0,01509	9,4035	0,771373	0,009225	0,167532	136574,7	112599,9	95,343	7573,9	13,79	3237,18	13898,45	28583,68
2020-01-08	0,002089	140,72	8055,98	0,01499	9,46891	0,769047	0,009258	0,167382	137237,8	112876,1	94,5985	7574,9	13,45	3253,05	13934,44	28745,09

Table 4: Independent time series variables

In this project, VARMA is used in a machine learning model building process. Data is splitted into train and test parts. Test data included the last two weeks of market prices. During VARMA training all market data is forecasted at the specified period, however, Bitcoin is selected for the main asset to be forecasted and plotted for tests.

After data is retrieved, first step is to analyze for stationarity with the Augmented Dickey-Fuller test. Later, stationarity is ensured using differencing on the pandas data frame. Unlike VAR algorithm, ARIMA orders for VARMA algorithm could be optimized with pmdarima extension of statsmodels. Finally, train and test data are splitted and model is built. Surprisingly, VARMA algorithm did quite poorly on understanding turbulence of markets.

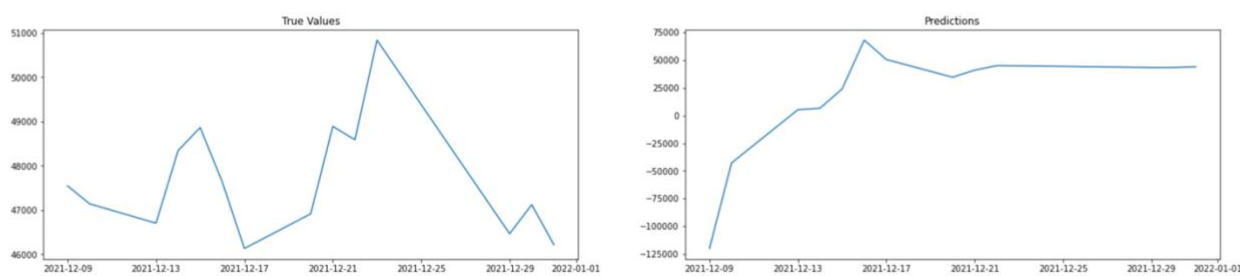

Figure 10: Bitcoin true values & Predictions with VARMA

Since VARMA is a complex algorithm that looks for patterns on the data. It was not expected to have a highly accurate score on a random data like crypto currency. On the other hand, it could have provided accurate information on the direction of the market rather than the degree of its increase or decrease. In such case, it could be stated that technical analysis techniques and trading strategies like Stochastic Swing are more reliable on trading in the financial markets. At least until a reliable model is built on the machine learning subject.

5. DISCUSSION AND RESULTS

In conclusion, several statistical methods applied on financial market data in this project. These include correlation tests, causality tests, a stationarity test, technical indicator simulations and a time series forecasting model called VARMA. There are variety of studies using artificial neural networks, long short-term memory or gated recurrent unit model, machine learning classification strategies and many more.

5.1 Discussion of Research Results

The main goal of this Project was to analyze and understand the concept of turbulence of markets. Assuming asset values are highly dependent on investors' expectations they are presumed to be understood as investors switch their attention to different incentives. Thereof, rise or fall of different markets may provide clues on rise or fall of other financial markets.

Granger Causality Test was assumed to be supportive in terms of ensuring information on causation between different markets. Augmented Dickey-Fuller test is applied to create stationarity on time series, causality matrix and causality networks are created, and VARMA algorithm is applied to predict results.

Furthermore, a simulation is created using technical indicators, trading results are calculated and presented on a pandas table.

5.2 Pedagogical Effects

Investment is an inevitable concept. Even though majority may assume it as risky and scary, there will always be risks in economical perspectives, properties that people own as assets or liabilities they have without realizing. For example, currency is some form of asset. Keeping US dollars on the bank account instead of EURO is an investment decision. Since currency market is very large and without very enormous actions in political or trading world, values of currency assets do not have a high volatility.

It is a monumental aspect in life and is not dependent on the environment people work in or live in. Since investment also includes components of knowledge, risk and optimization; artificial intelligence could be supportive on decision making part. Some of the main obstacles and reasons of failures in investment tasks are lack of knowledge, lack of emotional self-control and not following the events. Thereof, artificial intelligence could be the most assisting tool in the future of investing.

5.3 Results

Even though financial markets are moving in random, both in trend and volatility, there is research presenting high accuracy using machine learning algorithms. Considering most of these research papers present the accurate predictions that they acquired, it is possible to assume most tests would not result in a similar manner. Meaning, complex algorithms like artificial neural networks, discriminant analysis or VARMA may not be the solution for success in investment tasks.

Therefore, it is possible to state that it would be more beneficial to utilized strategies that provide accurate directional movements and volatility. Hence, generating buy and sell trading signals to make profits. It is possible to build simulations to test various trading strategies which might both include machine learning algorithms and technical analysis methodologies.

5.4 Suggestions

Idea of turbulence of markets requires long hours of work, knowledge, and tests. VARMA algorithm was not successful in understanding the turbulence of markets. Other algorithms like Non-linear Auto Regressive Moving Average Exogenous (NARMAX), LSTM or reinforcement learning algorithms like actor critic or Twin-delayed deterministic policy gradient (TD3) could be performed on different technical indicator data.

There is large pool of strategies that could be built using mathematics, econometrics, artificial intelligence, and financial analysis. For example, another perspective on investment strategies would be considering it as a classification task. Since possible decisions consist of three options which are buy, sell, or hold. Making accurate classifications on a time series data, which apparently also dependent on its past data, may provide solutions in the future.

Some other ideas could be;

- Forecasting volatility indicators like ADX or momentum indicators like stochastic RSI to predict the direction.
- Using Yahoo Finance API to retrieve balance sheet data and draw conclusions as positive or negative investor expectations.
- Using different technical indicators prepare possible trading strategies, testing the accuracy and profit of each strategy and utilizing an artificial neural network to classify possible outcomes.

REFERENCES

SCHIERHOLT, Karsten & DAĞLI, Cihan H. (1996) ‘’Stock Market Prediction Using Different Neural Network Classification Architectures’’, Proceedings of the IEEE/IAFE Conference on Computational Intelligence for Financial Engineering, Institute of Electrical and Electronics Engineers (IEEE), Jan, 72-78.

Kimoto, T., Asakawa, K., Yoda, M., & Takeoka, M. (1990). Stock Market Prediction System with Modular Neural Networks. International Joint Conference on Neural Networks (s. 1-6.). San Diego: Institute of Electrical and Electronics Engineers.

Yoon, Y., & Swales, G. (1991). Predicting Stock Price Performance: A Neural Network Approach. Proceedings of the Twenty-Fourth Annual Hawaii International Conference on System Sciences, (s. 156-162). Kauai.

Baba, N., & Kozaki, M. (1992). An Intelligent Forecasting System of Stock Price Using Neural Networks. International Joint Conference on Neural Networks, 1, s. 371-377. Baltimore.

Leung, M. T., Daouk, H., & Chen, A.-S. (2000). Forecasting Stock Indices: A Comparison of Classification and Level Estimation Models. International Journal of Forecasting, 16(2), 173-190.

Olson, D., & Mossman, C. (2003). Neural Network Forecasts of Canadian Stock Returns Using Accounting Ratios. International Journal of Forecasting, 19(3), 453-465.

Mostafa, M. M. (2010). Forecasting Stock Exchange Movements Using Neural Networks: Empirical Evidence From Kuwait. Expert Systems with Applications, 37(9), 6302-6309.

Shah, M., Prabhu, N., & Rao, J. (2014). Performance Analysis of Neural Network Algorithms on Stock Market Forecasting. International Journal of Engineering and Computer Science, 3(9), 8347-8351.

Karaatlı, M., Güngör, İ., Demir, Y., & Kalaycı, Ş. (2005). Hisse Senedi Fiyat Hareketlerinin Yapay Sinir Ağları Yöntemi İle Tahmin Edilmesi. Balıkesir Üniversitesi Bandırma İktisadi ve İdari Bilimler Fakültesi Akademik Fener Dergisi, 2(1), 22-48.

Kutlu, B., & Badur, B. (2009). Yapay Sinir Ağları İle Borsa Endeksi Tahmini. Yönetim Dergisi(63), 25-40.

Ulusoy, T. (2010). İMKB Endeks Öngörüsü İçin İleri Belemeli Ağ Mimarisine Sahip Yapay Sinir Ağı Modellemesi. Uluslararası İktisadi ve İdari İncelemeler Dergisi(5), 21-40.

Diler, A. İ. (2003). İMKB Ulusal-100 Endeksinin Yönünün Yapay Sinir Ağları Hata Geriye Yayma Yöntemi İle Tahmin Edilmesi. İMKB Dergisi, 7(25-26), 66-81.

Altay, E., & Satman, M. H. (2005). Stock Market Forecasting: Artificial Neural Network and Linear Regression Comparison in An Emerging Market. Journal of Financial Management and Analysis, 2(18), 18-33.

RAST, M. (1999), ‘’Forecasting with Fuzzy Neural Networks: A Case Study in Stock Market Crash Situations. Fuzzy Information Processing Society’’, NAFIPS. 18th International Conference of the North American, 418-420.

CHUNG, C. M., CHEONG, W. C. & CHUNG L. C. (2000), ‘’Financial Time Series Forecasting by Neural Network Using Conjugate Gradient Learning Algorithm andMultiple Linear Regression Weight Initialization’’, Computing in Economics and Finance 2000, Society for Computational Economics, no 61.

PHUA, P. K. H., ZHU, X. & KOH, C. H. (2003), ‘’Forecasting Stock Index Increments Using Neural Networks with Trust Region Methods’’, Neural Networks, 2003. Proceedings of the International Joint Conference on, Volume: 1, 260-265.

THAWORNWONG, Suraphan , ENKE, David (2003), ‘’Forecasting Stock Returns with Artificial Neural Networks’’, Neural Networks in Business Forecasting,47- 79

LI Rong J. & XIONG Zhi B. (2005), ‘’Forecasting Stock Market with Fuzzy Neural Networks’’, Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, 18-21 August, 3475-3479.

RODRIGUEZ, J. V. Perez (2005), ‘’STAR and ANN models: Forecasting performance on the Spanish “Ibex-35” stock index’’, Journal of Empirical Finance Volume 12, Issue 3, June, 490-509.

ZHORA, D. V. (2005), ‘’Data Preprocessing for Stock Market Forecasting using RandomSubspace Classifier Network’’, Proceedings of International Joint Conference on Neural Networks, Montreal, Canada, July 31 – August 4, 2549- 2554.

KUTLU, Birgül & BADUR, Bertan (2009), ‘’Yapay Sinir Ağları ile Borsa Endeksi Tahmini’’, Yönetim Yıl:20, Sayı: 63, Haziran, 25-40.

ADEBIYI, A.A.& AYO, C.K. (2011), ‘’Fuzzy-neural model with hybrid market indicators for stock forecasting’’, Int. J. Electronic Finance, Vol. 5, No. 3.

KARA, Yakup, BOYACIOĞLU, Acar, M. & BAYKAN, Ömer Kaan (2011), ‘’ Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange’’, Expert Systems with Applications, 38, 5311–5319.

MACIELL, L., GOMIDE, F. & ROSANGELA B. (2012), ‘’Evolving Fuzzy Modeling for Stock Market Forecasting’’, IPMU 2012, Part IV, CCIS 300, pp. 20–29.