Stock Market Projection Prediction Using Supervised Artificial Neural Networks

Joshua Fortune, Kiran Thapa, Manish Karki

CSE 4990 Computational Intelligence, Mississippi State University, 75 B.S. Hood Rd., Mississippi State, MS, USA

jaf575@msstate.edu, kt1397@msstate.edu, mk1589@msstate.edu

I. Introduction

This is a web development project utilizing supervised artificial neural networks to generate predictions for specified stocks based upon historical stock performance data gathered using freeware APIs. This project hopes to provide stock market investors with a tool that provides guidance when selecting profitable investment opportunities.

A. Background

The main aim of this project is to implement the artificial neural network to predict future stock indices. The system will predict the future stock value based on the historical data and repeated patterns in the history of the stock data. Two separate sources of stock data are used to train the underlying artificial neural network of the software.

B. Proposed Functionality

End users can access the software through a web application (possibly an electron app accessible from the desktop). A user must input a stock name or market code.

Using freeware APIs, the software pulls relevant stock market data such as stock names, prices, dates and timestamps of prices, and respective derivatives of stock prices at each timestamp. These data sets are used as input to train the software's supervised artificial neural network will teach the system to recognize patterns in a stock and allow the system to accurately predict whether a certain stock is a good investment or not.

The output of the supervised artificial neural network will be fed into an expert system to interpret output results. The output of the expert system will be human readable statistics for each respective stock input.

C. Technologies Used

Electron[1] - This is a technology we will use which allows for web development to be brought to the desktop by compiling a cross-platform compatible application composed of HTML, CSS, and JavaScript. The usage of a web application will allow our team to develop a feature-rich and easily updatable GUI. Human interaction with our application will be intuitive and straightforward. Because the application will be cross-platform, usability will be the focus for this Electron technology.

Angular Framework[2] - This is a technology we will use to develop our GUI for the Electron platform. Angular Framework is a JavaScript and TypeScript based framework that improves writeability of front-end applications. The framework is based upon compartmentalizing front-end code into components and allows for easy organization of those components. This will speed up development for unintelligent parts of our application.

Python[3] - This technology is a feature-rich and easily writeable programming language to be used for our team's back-end supervised artificial neural network. Python provides libraries for http and socket functionality (useful for web application approaches), the neural network functionality itself, and many others. If an end user requires a CLI input to use the software, the Django for Python library can be incorporated. Django provides an all-in-one solution for CLI based projects (full-stack development).

CLIPS[4] - This is a technology introduced to our team by Dr. Rahimi during our time taking CSE 4990 - Computational Intelligence at Mississippi State. It is an expert system technology that our team will use to process the trained neural network output into a human-readable format. By defining membership functions and rules, the software can determine whether a stock, fed as input to the front-end, will be a good investment (and just how good of an investment it will be).

Alpha Vantage JSON API[5] - This technology is a freeware API that provides "realtime and historical stock and equity data with over 50 technical indicators [dimensions of data]" [5]. Using this API, our team will pull relevant stock data from the Alpha Vantage network to build known, proven data sets for our supervised artificial neural network training.

World Trading Data API[6] - This technology is a second freeware API that our team will use to pull relevant stock data for the purpose of building data sets used to train our supervised artificial neural network. By employing two APIs, given that they are both freeware, our team will create a verbose data set covering each relevant dimension of the given stock data for training the software.

II. Implementation

D. Division of Labor

Team member Joshua Fortune is tasked with front-end Angular development. Specifically, Joshua will assist team member Kiran with organizing HTML, CSS, and JavaScript into Angular Framework components as well as ensuring data paths and back-end connectors are created and accurate using TypeScript and JavaScript with the Angular Framework. Joshua will also be responsible for importing the finished Angular application into Electron for local use of the software on the desktop. Responsibility for neural network design and implementation will be shared with Kiran; Joshua will design and implement the core neural network and ensure proper functionality.

Team member Kiran Thapa is tasked with designing and generating HTML, CSS, and JavaScript files necessary for GUI operation as well as design and implementation of the interface for the back-end Python neural network component for the software. Kiran's design of the connection between the GUI and neural network will be shared with team member Joshua to ensure the connections necessary to train and query the artificial neural network operate properly.

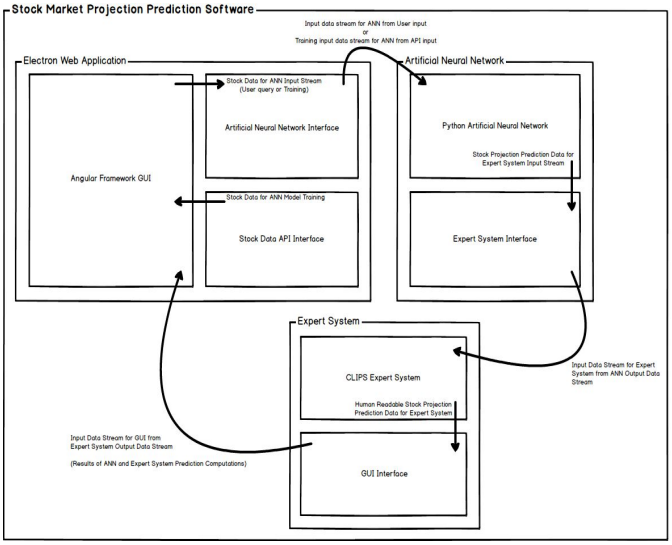
Team member Manish Karki is tasked with the design and implementation of the expert system component of the software. Specifically, Manish will design and implement an

interface allowing the expert system to receive input from the neural network output stream as well as the core expert system itself. This responsibility includes producing rules which accurately interpret the neural network output stream into human-readable format.

E. Architecture Process

To build the system, the project is divided into different phases . In the first phase we will be developing an artificial neural network. The artificial neural which consist input layer, output layer and hidden layers. In the second phase we get the stock data using freeware API's which trains artificial neural network . In the third phase we will develop the platforms and the designs to present the output that we get from the artificial neural network. Expert system will be used to process the output into human-readable format.

F. Architecture Design



III. Discussion and References

A. Case Study [7]

The prediction of stock market depends on the input layer, hidden layer and output layer of the artificial neural network. This project relies on the prediction model proposed by Amin Hedayati Moghaddam, Moein Hedayati Moghaddam, and Mortez Esfandyari. The team used the values of NASDAQ exchange rate to train the data with the stock rate and the day of the week as the input parameters. The use of an artificial neural network for the prediction resulted in very close prediction with 96.22 percentage validation for the data provided to the system. The data gathered from the NASDAQ for a period of 99 days were randomly divided into a training set (60%), validation set (20%), and testing set (20%). OSS (one step secant) was used as a training method while TANGSIG, LOGSIG were used as transfer function. The architecture of the neural network was designed to get the optimized value for the number of hidden layers and number of neurons in each hidden layer. Through multiple experiments, it was found that the system would yield optimal accuracy with two hidden layers of 20-40-20 neurons in the hidden layers with the help of an OSS training method and a LOGSIG transfer function resulting in the 96.22% accuracy. So, our team will be using OSS as the training method and LOGSIG as the transfer function to predict the stock prices.

B. Expected Conclusion and Future Enhancement

The system predicts the stock values fairly accurate based on historical data using artificial neural network. The preliminary goal for this project would be to predict whether the stock price would go up or not, basically a boolean output. On the other hand, the secondary goal would be able to predict the stock price in dollar amount.

C. Abbreviations and Acronyms

API - Application program interface

CLI - Command line interface

GUI - Graphical User Interface

OSS- One step Secant

NASDAQ- National Association of Securities Dealers Automated Quotations

References

[1] "Build cross platform desktop apps with JavaScript, HTML, and CSS.," Electron. [Online]. Available: https://electronjs.org/.

[2] "CLI Overview and Command Reference," Angular. [Online]. Available: https://angular.io/cli.

[3] "Welcome to Python.org," Python.org. [Online]. Available: https://www.python.org/.

[4] G. Riley, "CLIPS Rule Based Programming Language," CLIPS Rule Based Programming Language. [Online]. Available: https://sourceforge.net/projects/clipsrules/.

[5] Alpha Vantage Inc., "Alpha Vantage API Documentation," API Documentation | Alpha Vantage, 2017. [Online]. Available: https://www.alphavantage.co/documentation/.

[6] World Trading Data, "Introduction," API Documentation | World Trading Data, 2019. [Online]. Available: https://www.worldtradingdata.com/documentation/.

[7] A. H. Moghaddam, M. H. Moghaddam, and M. Esfandyari, "Stock Market Index Prediction Using Artificial Neural Network," Journal of Economics, Finance and Administrative Science, vol. 21, no. 41, pp. 89–93, Dec. 2016.