Knowledge-Based Integrated Financial Forecasting System

Andrian Kurniady
School of Computer Science
BINUS International – BINUS University
Jakarta, Indonesia
andrian@kurniady.net

Raymondus Kosala
School of Computer Science
BINUS International – BINUS University
Jakarta, Indonesia
rkosala@binus.edu

Abstract—This paper seeks to implement and test a financial forecasting agent which employs time series, derived time series data, and news that are retrieved and extracted from the Web. This research focuses on the time series data of some individual stocks from the Indonesian Stock Exchange as well as the index data. The financial forecasting agent implemented is based on a Multilayer Neural Network trained with the Genetic Algorithm. An incrementally trained Naïve Bayesian classifier to classify the news stories automatically is also implemented. The agent is measured for its performance on the specified historical stock price data. The first testing surveys its performance against the benchmark agents in a range of stocks. Although the agent does not consistently outperform the benchmarks, it does have its own advantages in terms of stability. The second test tries to find out whether news improves the agent’s prediction accuracy. We find that in certain stocks and the index, the inclusion of news does improve the prediction accuracy.

Keywords—web agent; financial forecasting; news classification

I. INTRODUCTION

Financial markets are being defined by Mishkin [1] as markets in which funds are transferred from parties having excess to parties who have a shortage of funds. Financial market prediction has been a popular area of research in the recent years, shown by many research publications on financial prediction systems, especially by using computational approaches. In general, research in this area can be organized into two categories based on the data types in use by the systems. The first is the research that focuses only on one data type, which is usually the time series data. In this category, a survey paper by Lawrence [7] found that neural networks outperformed the older statistical and regression techniques in predicting stock prices. In a recent research, Huang and Kim [8] concluded that the MACD crossing trade system outperformed the buy-and-hold strategy on a few world stock indices only when the transaction cost is ignored.

The second is the research that focuses on many data types, which includes input such as price, technical indicators, news and even multimedia data. One of the works in this category was done by Becker and Seshadri [9], which uses the genetic programming to evolve the technical trading rules quite successfully. Another research by Farsworth, et al. [10] also uses the genetic programming approach over the price and technical indicators data, which is specialized to several types of moving averages. The work by O’Connor and Madden [11], which trains the neural networks, also consider external factors such as the commodities prices and currency exchange rates. The research by Gidófáli [12] found that news articles have a strong correlation to the price movement for a short period of time before and after the news article become available. Pang et al. [13] shows similar correlation by using a support vector machine to classify text news and use it to predict the movement of Hong Kong stocks.

This paper aims to implement the integrated framework for financial forecasting as proposed by Kosala and Kumaradja in [2]. The framework in [2] advocates the use of multiple data types for input such as price, technical indicators, news and even multimedia data. Then these inputs and some derived time series data will be used for making predictions. This paper is focused on developing a system like the one developed by Farsworth, et al [10], except that instead of using the genetic programming, we use neural networks trained with genetic algorithms. We also use incremental naïve Bayes classifier for the news classification, similar to the work by Gidófáli [12]. Our system, GRAM (Genetically Trained Multilayer Network) agent, was tested using data extracted from the Web such as the Indonesia Stock Market price data, as well as other data. As there is minimum amount of past publication of forecasting systems based on Indonesian stocks, therefore comparison with other works is not viable. In testing the agents’ performance, we also take into account the transaction fees.

II. FOUNDATIONS

A. Data Types in Stock Markets

There are several kinds of data that is available related to the stock markets [2]. These data include time series data, technical indicators data, fundamentals data, as well as news stories. Understanding these types of data is crucial in building an accurate prediction system. The Engineering Statistics Handbook [3] defines time series as an ordered sequence of values of a variable at equally spaced time intervals.

Technical indicators are usually time series data that are derived from the original time series data by means of calculations with certain formulas. Some popular technical indicators are the moving averages, the Moving Average Convergence Divergence (MACD), Parabolic SAR (stop and

978-1-61284-840-2/11/$26.00 ©2011 IEEE

1Currently a Software Engineer in Test at Google Australia.
reverse), Relative Strength Indicator (RSI), and the Bollinger Bands, among many others.

Fundamentals data is also a significant factor influencing the stock prices. These fundamentals data includes microeconomics data about the company itself and its industry, as well as the macroeconomics data of the world and the countries in which the company (and the exchange) operates in.

Relevant news is also a significant factor in determining stock prices. News contains recent changes in the microeconomics and macroeconomics factor that may influence the performance of a company, and thus, influence the price of its stock.

B. Technical Indicators

Moving averages are averages that is calculated based on the last elements of a time series in a given timeframe. The simple moving average (SMA) is the simplest kind of moving average, in which the value is calculated simply by averaging the predefined number of latest elements.

The exponential moving average (EMA), or also known as exponentially weighted moving average, is a modification to the SMA in which the weights of the elements are distributed exponentially. In EMA, the period affects only the smoothing factor (α) and does not denote the number of elements considered. The formula for EMA is shown below.

\[ \alpha = \frac{2}{N+1} \]

\[ \text{SMA}_i = \frac{x_i \cdot (1-\alpha)^{i-1} + (1-\alpha)^{i-2}x_{i-2} + \cdots}{1 + (1-\alpha) + (1-\alpha)^2 + \cdots} \]

In the formula, \( x_i \) denotes the i-th element of the original time series.

The Moving Average Convergence Divergence (MACD) is a technical indicator introduced by Gerald Appel [5], which basically uses the combination of two EMA of suggested period 12 and 26 of the prices. The MACD value is calculated with the following formula.

\[ \text{MACD}_t = \text{EMA}[26]_t - \text{EMA}[12]_t \]

C. Machine Learning Algorithms

The genetic algorithm (GA) is a search algorithm inspired by the biological mechanism of evolution. The goal of this algorithm, as with other searching algorithms, is to find the best exact or approximate solution to optimization and search problems. Genetic algorithms simulate biological evolution by maintaining a set of states in a population and evolve the states in several iterations. It is expected that by the end of the simulation, the population contains several states that are good enough as a solution.

Artificial neural network (ANN), or sometimes called neural network (NN), is yet another concept derived from biology. There are several kinds of network topologies in ANN, e.g. some are categorized as the feed-forward network and the recurrent network. Typically feed-forward network is trained with the back-propagation algorithm, but the research by Riley and Ciesielsky [6] shown that genetic algorithm can also be used.

The naïve Bayes classifier is one of the simplest statistically based classifier. It is based on observed probability of past examples regarding the evidences related to an observed classification. This kind of classifier is possible by means of the use of Bayes’ theorem as described in Russel and Norvig [4]. The classification is derived by comparing the likelihood of each possible classification given the set of evidences.

III. SYSTEM DESIGN

A. High Level Design

Due to the main purpose of the work which is more into the implementation of prediction algorithms and the measurement the algorithms’ performance, the system is to be developed with simple user interface (UI), which is the console UI. The major input and output processes will be done through files which use simple formats. Figure 1 illustrates the high level interactions between the systems and its environment.

![Data Flow Diagram](image)

**Figure 1.** The data flow diagram of the agent

B. News Data Input

The news processor trains its naïve Bayesian classifier by means of an internal simulation through the trading days. It started with an empty classifier (i.e. a classifier which has not been trained at all), and loop through the trading days in a chronological manner. On each day, the classifier is incrementally trained with the recently encountered news stories and its label, which is derived automatically from the price movement following the news story.

For the purpose of classifying news stories in a useful way to the agents using the resulting data, the classifier is designed to classify the news stories into one of the three labels: UP, UNCHANGED, and DOWN. The observed feature for the classification purpose is the individual words contained by the news story. The calculation can be done using the naïve Bayes formulas as shown in Figure 2, in which \( D \) represents the news story and \( \omega \) represents the unique words in the story.

The news chosen for this project is limited to macroeconomics news, which can be reasonably gathered
from online financial news providers. This choice has also an advantage such that it can be used in the simulation for any stocks, because of its generality.

\[
P(UP|D) = \alpha P(UP) \prod_i P(x_i|UP) \\
P(UNCHANGED|D) = \alpha P(UNCHANGED) \prod_i P(x_i|UNCHANGED) \\
P(DOWN|D) = \alpha P(DOWN) \prod_i P(x_i|DOWN)
\]

Figure 2. Naive Bayes formulas

C. The Forecasting Agents Component

There are three types of forecasting agents being implemented in this project: the Buy-and-hold agent, the MACD agent, and our GRAM (Genetically Trained Multilayer Network) agent. The Buy-and-hold only buys the stock with all of its money, while the MACD agent trades based on the MACD indicator. Both of them represent the simple reflex agent model [4]. The GRAM agent is more complicated and represents the learning agent model [4].

The Genetically Trained Multilayer Network (GRAM) agent uses Genetic Algorithms to train Multilayer Neural Networks, and for practical purposes is named the GRAM agent. The Genetic Algorithms is used to evolve the weights of the multilayer network, as a replacement to the traditional Back-Propagation algorithm.

In each of the evolution steps, a succession of the generation, entails a cycle of migration, crossover, mutation, and elitism which is typical to the Genetic Algorithm. Figure 3 illustrates the flow of individuals in each generation. The migration replaces worst individuals with new random individuals. The crossover and mutation follows the typical model in genetic algorithm, which involves a roulette selection method. The elitism takes the best individuals to be copied intact to the next generation.

Figure 3. The flow of individuals in the population during evolution

The predictions are then based on the prediction of the best performing agents in the pool, to achieve better results by reducing noise in the prediction result. Several parameters of the Genetic Algorithm as well as the multilayer network are also specified in the configuration file, which opens the way for extensive parameter optimization in the future.

IV. IMPLEMENTATION AND TESTING

The Forecasting Simulator of the GRAM (Genetically Trained Multilayer Network) agent is implemented in the C++ programming language and is shown in Figure 4. The whole program relies heavily on the C++ Standard Template Library and is written such that it adheres to the GNU C++ standard to ensure portability.

A. Input Data Selection

For tuning and performance evaluation, the stocks data used will cover the trading days between January 1st, 2006 and December 31st, 2008, containing 717 valid trading days. The stock prices and index values are gathered from Yahoo! Finance, which are extracted in the form of CSV files.

A somewhat shorter simulation period is used in the simulations in which the presence of news is being considered. The news that has been gathered and to be used are news stories from Bimis.com, containing both news from printed edition of Bisnis Indonesia newspaper as well as online news, under the category of macroeconomics. The news stories gathered ranges from February 2007 until December 2008, which denotes the period used for the simulations in which news stories are considered.

To obtain the news stories, a PERL script was written to automatically retrieve the stories with the help of Google search and store the stories in TXT files. The method successfully gathered 3181 news stories during the period.

The input channels in use for the GRAM agent are the combination of closing price, trade volume, percent of change, daily trading range, daily index data, some EMA (Exponential Moving Average) and MACD (Moving Average Convergence Divergence) data, and also the news (when used). For the historical price, the window length is set to 5, while for derived and news data, the window length is 1. The window length of 5 used in many of these input channels is derived from the number of trading days in a week, which is normally Monday to Friday. Some derived input channels are used with the window length of 1, because
only the latest value matters in describing the current condition of the stock price. Based on this setting, the input length of the GRAM agent is 38 without news, and 41 with news. Normalization is also done to some of the channels to improve the quality. Due to the limited space, the details of the channels normalization can be found in [14].

The Buy and Hold agent does not need any input data, as it only issue BUY command at all times. The MACD agent uses the MACD Histogram data only. For all of the simulations, the transaction fee is set to 0.25% one-way or 0.5% in total (buying plus selling).

B. Parameter Tuning

The tuning of parameters is important in both neural networks and genetic algorithm to function at their best. Therefore, before proceeding to the actual performance evaluation, the GRAM agent needs to be fine-tuned first.

Table I shows the difference of parameters before and after tuning. Due to the limited space, the explanation of these parameters can be found in [14].

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Initial Value</th>
<th>Tuned Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>minRandom</td>
<td>-10</td>
<td>-10</td>
</tr>
<tr>
<td>maxRandom</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>hiddenNeurons</td>
<td>5</td>
<td>15</td>
</tr>
<tr>
<td>poolSize</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>generationsPerDay</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>maxExamples</td>
<td>100</td>
<td>150</td>
</tr>
<tr>
<td>mutationRate</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>eliteCount</td>
<td>5%</td>
<td>7%</td>
</tr>
<tr>
<td>migrantCount</td>
<td>20%</td>
<td>20%</td>
</tr>
<tr>
<td>learningThreshold</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>consideredIndividuals</td>
<td>5%</td>
<td>3%</td>
</tr>
</tbody>
</table>

C. Performance Testing

After the GRAM agent has been tuned, its performance is tested on a set of different stocks for its performance in general. Along with the TLKM (Telekomunikasi Indonesia, a telecommunication company) stock which is used to tune the GRAM agent, it is also tested on BBCA (Bank Central Asia, a bank), AALI (Astra Agro Lestari, an agriculture company), ISAT (Indosat, a telecommunication company), UNVR (Unilever Indonesia, a consumer goods company), and PTBA (Tambang Batubara Bukit Asam, a mining company) stocks.

All agents start with 1,000,000,000 (one billion) in cash, and its performance is measured based on the return of investment (ROI), that is its asset (cash plus equity) on a particular day compared to the initial cash value. The evaluation will be divided in two parts:

- second part, using 2 years of data along with news stories.

Performance Testing on 3 Years of Data

The first part of the performance evaluation comprises of 717 trading days spanning from January 2006 to December 2008 and is run on the six chosen stocks. This part of the evaluation is aimed to find out about the performance of the agents in general.

The performance in summary can be found in Table II, which also compares the GRAM agent performance to the Buy-and-Hold agent and MACD agent.

<table>
<thead>
<tr>
<th>TABLE II. SUMMARY OF THE AGENTS' ROI</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>-------------------------------</td>
</tr>
<tr>
<td>Buy and Hold</td>
</tr>
<tr>
<td>MACD</td>
</tr>
<tr>
<td>GRAM</td>
</tr>
</tbody>
</table>

The Buy and Hold strategy is the best for BBCA, AALI, and UNVR stocks. The MACD strategy is best for PTBA stock only. The MACD agent wins in ISAT and TLKM. The performance of the MACD agent can be considered unpredictable, as it loses a significant amount of its wealth in 3 out of 6 stocks tested, which suggests that it can make a lot of gain but can also lose a lot. This behavior can be considered as higher risk for some investors. The GRAM agent, on the other hand, makes positive gains in all of the tested stocks, although the gain may be smaller compared to other agents in certain stocks. The reason why the GRAM agent performs well in ISAT, a telecommunication company, might be because it is using TLKM, another telecommunication company, as its dataset when tuning its parameters.

Performance Testing on 2 Years of Data with News

The second part of the testing process is aimed more to investigate the correlation between the inclusion of news and the GRAM agent's performance rather than the details about performance of the agent in individual stocks as seen in the testing in the first part. The news processor successfully classified the 3181 news stories between February 2007 until December 2008, into UP, DOWN, and UNCHANGED labels as shown in the Table III.

The result of running the simulation 20 times for each stock, using the classified news above, can be seen in Table IV, which shows the average ROI of GRAM agent with news and the others. Note that some of the ROI numbers were negative because there was a bearish (declining) trend in the Indonesian stock market during year 2008.

In general, the GRAM agent with news outperformed the GRAM agent without news on TLKM, AALI, ISAT, and PTBA stock. The 4 out of 6 ratio suggests that the presence of news does affect the forecasting performance of the GRAM agent, in a slightly positive way.

<table>
<thead>
<tr>
<th>TABLE III. THE RESULT OF THE NEWS CLASSIFICATION</th>
</tr>
</thead>
</table>

123
Because the news stories are general macroeconomics news, a test on the Jakarta Composite Index (HSGC) is also conducted. Averaged from 40 runs, the GRAM agent with news outperformed the GRAM agent without news (-21.12% vs. -27.74%), while the Buy-and-hold and MACD agent performs at -23.25% and 15.83% respectively.

**TABLE IV. The ROI of the Agents**

<table>
<thead>
<tr>
<th></th>
<th>TLKM</th>
<th>BBCA</th>
<th>AALI</th>
<th>ISAT</th>
<th>UBVVR</th>
<th>PTBA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buy and Hold</td>
<td>-13.20%</td>
<td>-2.52%</td>
<td>-24.70%</td>
<td>-14.50%</td>
<td>16.75%</td>
<td>32.05%</td>
</tr>
<tr>
<td>MACD</td>
<td>-42.65%</td>
<td>5.59%</td>
<td>15.33%</td>
<td>-61.27%</td>
<td>-55.54%</td>
<td>25.68%</td>
</tr>
<tr>
<td>GRAM</td>
<td>-16.02%</td>
<td>4.91%</td>
<td>-14.15%</td>
<td>-8.88%</td>
<td>5.38%</td>
<td>28.34%</td>
</tr>
<tr>
<td>GRAM with News</td>
<td>-12.67%</td>
<td>-7.71%</td>
<td>-10.10%</td>
<td>-4.10%</td>
<td>-1.23%</td>
<td>37.10%</td>
</tr>
</tbody>
</table>

V. CONCLUSION

The use of Genetic Algorithm for training the Multilayer Neural Network, as shown in the GRAM agent, was successful to some extent. There was some observations suggesting that the stock price movements cannot be predicted based on the past training data to a very high accuracy yet.

By doing some testing on the performance of the Buy-and-Hold, MACD, and the GRAM agent, we found that no single strategy outperforms other strategies consistently. However, the newly developed GRAM agent does have its own advantages in terms of stability, in which it provides a more stable gain across the different stocks compared to the others.

The significance of news is also not proven to be consistent. There were evidences that in some individual stocks as well as the index, the news provide a significant performance gain to the GRAM agent. This suggested that the selection of the news as well as the characteristics of the stocks may have played a role in the effect of the incorporation of news into the GRAM agent's decision making process.

Future works may consider investigating the selection of news for certain stocks, as the macroeconomics news used in this thesis seems not to correlate consistently for different types of stocks. The use of more sophisticated classifier as well as a preprocessing (such as to remove stopwords) may be interesting.

REFERENCES