A Machine Learning-based Decision Support Tools for Portfolio Risk Analysis

By

TAN KWEE HOE

A project submitted to the Department of Internet Engineering and Computer Science,

Faculty of Engineering and Science,

Universiti Tunku Abdul Rahman,

In partial fulfillment of the requirements for the degree of

Master of Information Systems

April 2014

TABLE OF CONTENTS

ABSTRACT

There are a lot of investors in stock market. Each of them uses different strategy to define which stock to invest. There are two major types for stock investment, one of it is short term investment and the other is long term investment which investors select investment type based on their needs.

This Machine Learning-based decision support tools is suitable for short term investment in stock market. This research is about Machine Learning-based decision support tools for portfolio risk analysis which helps investors in selecting stock. This Machine Learning-based decision support tools is created by Bayesian Network as the artificial intelligence for the system, Visual C++ which is used to consume the Bayesian Network and display is done by using PHP.

Daily calculation for every single stock in market will be time consuming. Machine Learning-based decision support tools will narrow down potential stocks to invest. Investors only need to calculate the stock based on the return results from the Machine Learning-based decision support tools.

This Machine Learning-based decision support tools for portfolio risk analysis will provide stocks opening price for stock for seven, fourteen and also twenty one days after, so investors will have enough time to take action based on the time given. By having these return results, investors only require having detail calculation on those potentials stock to invest or sold out stocks in hand.

Research purpose for this also proved that Machine Learning-based decision support tools for portfolio risk analysis are able to help investors in stock selection. Machine Learning-based decision support tools for portfolio risk analysis have a lot of potential in stocks selection. It definitely has value to be researched in.

ACKNOWLEDGEMENT

I take this opportunity to thank my supervisor Dr Tay Yong Haur, for his exemplary guidance, monitoring and constant encourage me throughout the course of this project.

I also take this opportunity to thank my relatives that help me to complete my project. Based on their experiences and suggestions for the Machine Learning-based decision support tools for portfolio risk analysis, I manage to add in functions that meet investors need.

I want to my friend Andrew Philip Fredericks that read through my project and helped me to check my project grammatical error, so that my project can be understand by everyone.

Lastly, I thank my parents, sisters and friends for their constant encouragement; because of them, I am able to finish my project in time. Thank you once again for those that helped me finish my project.

APPROVAL SHEET

This project entitled "**A Machine Learning-based Decision Support Tools for Portfolio Risk Analysis**" was prepared by TAN KWEE HOE and submitted as partial fulfillment of the requirements for the degree of Master of Information Systems at Universiti Tunku Abdul Rahman.

Approved by:

_____

(Dr. Tay Yong Haur) Date:……..

Supervisor

Department of Internet Engineering and Computer Science

Faculty of Engineering and Science

Head of Programme (Master of Information Systems)

Chairperson, Centre for Computing and Intelligent Systems (CCIS)

Universiti Tunku Abdul Rahman

FACULTY OF ENGINEERING AND SCIENCE

UNIVERSITI TUNKU ABDUL RAHMAN

Date: _____

SUBMISSION OF PROJECT

It is hereby certified that TAN KWEE HOE (ID No: **11UEM07522**) has completed this project entitled "A Machine Learning-based Decision Support Tools for Portfolio Risk Analysis" under the supervision of Dr Tay Yong Haur (Supervisor) from the Department of Internet Engineering and Computer Science, Faculty of Engineering and Science.

I understand that University will upload softcopy of my project in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

(TAN KWEE HOE)

DECLARATION

I hereby declare that the project is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at UTAR or other institutions.

_____

(TAN KWEE HOE)

Date _____

LIST OF FIGURES

LIST OF FIGURES

1.0 INTRODUCTION

1.1 Introduction

People love to invest in the stock market because it provides a good return. But some of them might lose a lot of money in stock markets because they don't know how to invest in the stock market or some others misfortunate incidents that can happen along the line. This research is about using Machine Learning-based decision support tools that help people invest in the stock market. This research also helps people to understand clearly about the Machine Learning-based decision support tools in portfolio selection.

Traditional decision support tools for portfolio risk requires fund managers or investors to make decisions themselves by looking at the graph or data analysis generated by the decision support tools and making decision themselves. Examples are such as particular stock prices increases or decreases in various time duration. Machine Learning-based decision support tools will automatically generate a report to show that stock price is increasing or decreasing. Machine Learning-based decision support tools will assist investors to invest in stock market in a much easier way.

1.2 Problem statement

There are more than thousands of stocks available in Malaysian stock market. It will be a trouble for investors to calculate the stocks one by one before they purchase or sell out the stocks. Checking through thousand over stock will be

time consuming and also man power consuming as stocks data changes almost every day. Thus, investors or fund company need to go through same process every single day. Only performing calculation on particular stocks will cause the investors to miss the opportunity to invest in potential stocks. This will be a huge losing of investors.

Individual investor will never have time to calculate all the stocks one by one; basically they invest base on news from stock market. News which they get might be fake news, and investors will lose lot of money because of it.

In stock market, whoever can make an accurate prediction, and then he will be the one who get lot of money. This machine Learning-based decision support tools for portfolio risk analysis used to help investors in generate revenue in stock market.

1.3 Objective:

1.3.1 Research Objective:

Objective of this research are:

- To develop a Bayesian Network which is able to return results for seven, fourteen and also twenty one days after opening price for stocks.
- To develop a Visual C++ program which is able to consume the Bayesian Network that has been created.
- To develop PHP that is able to process the raw data and also to display the return results.
- To test and also evaluate the accuracy of the return results accuracy.

Core function of Machine Learning-based decision support tools for portfolio risk analysis is the Bayesian Network. Bayesian Network is used to define the opening price for the stocks in next seven, fourteen and also twenty one days. Reason of provide result after seven, fourteen and also twenty one days is because investors will require time in order to perform action either sell out or purchase the stocks.

1.3.2 Research Tasks

In order to accomplish the above task

- Gather end of day data through internet
- Turn raw data to percentages format, so that the Bayesian Network able supports every stock in market.
- Set different background knowledge in Bayesian Network, and evaluate every single Bayesian Network to select out the most accurate Bayesian Network structure.
- Test the Bayesian Network through Visual C++ in order to make sure both of the language and also Bayesian Network can communicate well with each and other.

Detailed explanation about data preparation, network selection and others is in following chapter. Data preparation, calculation or format that apply in raw data was important, every single format and calculation will affect accuracy of the Bayesian Network. Every stage is important, because any single mistake

that made will affect accuracy of Bayesian Network, so every stages requires testing phase.

1.3.3 Scope

This Machine Learning-based decision support tools for portfolio risk analysis was only suitable for use in short term investment, and it only usable for Malaysia stock market, because it only use Malaysia end of day stock market data to setup the Bayesian Network. It suitable in short term investment because, the return result of it was only up to twenty one days the longest period return result for the Machine Learning-based decision support tools for portfolio risk analysis, so it was only suitable to use in short term investment.

Normally long term investment was referring to blue chip stocks investment. That was only some stocks in markets was blue chip stocks, so Machine Learning-based decision support tools for portfolio risk analysis was only scope for short term investment which having bigger market if compare with long term investment.

In future, can add in even more country stocks market and enable long term investment. This research is use to prove that the Machine Learning-based decision support tools for portfolio risk analysis was able to help fund manager, investment agent, investment broker or investors in stocks selection.

2.0 LITERATURE REVIEW

2.1 System that normally use for portfolio selection

Based on the industry expert suggestion, I have some ideas about on how they make decision on portfolio stock selection. Brokers or fund manager have to define either they want to invest in the portfolio stock for long term or short term because investment in the stock market for long term or short term uses different technique in portfolio stock selection.

Normally for long term investment in stock market, brokers or fund managers will calculate portfolio stock price by using fundamental technique. They will do an analysis for the company. Based on the fundamental calculations, brokers or fund managers will know whether the company will be profitable, progress in growth or both. If the company gets both, then the company's stock will also be growing.

For short term investment in stock market, brokers or fund managers will select stock based on the analysis data or graph that are generated by analysis team or the system that can provide such data. Based on the graph or the analysis data, the brokers or fund managers have to make decision either to invest in the particular stock or not.

Long term or short term investment in stock market uses different technique in order to make decision on portfolio stock selection. But both terms of investment also encounters a same problem which is the final decision. The

final decision is based on the experience of the broker and the fund manager that make decisions on stock selections and the decisions that make by the brokers or fund managers might be affect by their current mood. This trivial point might cause the investors to lose their money because they choose the wrong brokers or fund managers to invest for them.

Different portfolio stock investment firm might use different methods or different systems in order to support the brokers or fund managers in stock selection. This is one of the many methods or ways that the real world portfolio stock investment firm uses for stock selection.

2.2 Investment strategy

Normally there are two types of investment methods which is long term investment or short term investment. Both investment methods and skills are different, therefore investors or fund managers need to know which technique is preferred to be applied for analysis portfolio stock price.

2.2.1 Long term investment in stock market

Long term investment should start earlier. Investors invest in stock market when they are younger so that they can earn more when they are older. This investment also poses a risk as they can also lose money if company goes bankrupt or other unfortunate incidents happened.

As an example:

Investors that plan to end their investment while they reach 65 years old. Investors that invest at 25 years old until the age of 65 years old will have 40 years period of time until the sold out their stock. Investor that start invests at the age of 55 years old will have 10 years period of time until the sold out their stock.

If the stocks increase 8% every years, for those who invest in the age of 25 will earn more than those invest in the age of 55. That's why for those who plan to invest in long term start early will earn more than those who start late.

Long term investment is based on interests that are given by company and the growing of the stock price as a return of investment. Some company will give some return for their shareholders from the company profit at the end of the year; this will be a benefit for long term investor.

Long term investment is safer compared to short term investment because at the beginning, companies will need to spend a lot of money to setup their business, so their stock price will be lower compare with those business which have been running for years or decades. Business will keep growing and if the CEO of the company is competent then the company will grow fast, indirectly causing the stock prices increase faster.

2.2.2 How to find out which long term stocks to buy

1. **Avoid price competitive industry –** do not invest in companies that do not have business strategy. They only compete on the price of products

or services with their competitors. Company that does not have business strategy will eventually face bankruptcy. In order to compete with their competitors, they reduce the price of the products or services. What they are doing is just being competitive on price rather than the product quality or satisfaction of their customers. This will cause companies to spend lot of money and it will be an issue that causes their shareholders to have a lower return rate.

2. **Economic moat –** does the company have strategies that make them have the ability to maintain to compete advantages? This is important to protect long term profits and market shares. For companies such as "Apple", they can have strategy that to protect their product advantages. Those product functions and designs are unique compared with their competitors. Apple pattern up ideas in order to protect it and provide high end technology to end users. Apple is able to maintain the competitive advantages. That is why they were well-known in their business sectors and have lot of end users that support their products.

3. **Growth –** company will continue to grow or regress is another important issue that must be considered. If the company grows consistently or the company grows at an annual pace, the shares of the company can be purchase for long term investment purposes. For those companies that didn't grow or keep losing money then it cannot be invested, because it will cause the stock prices to decrease. The company that does not grow means that the CEO or upper management

didn’t invest their money properly or they don’t know how to run the business. Eventually causing their business to stop growing or facing bankruptcy.

4. **Profitability –** annual profits of the company is high or low? Is there any profit from the business? How much can the company earn every year? Company profits depend on how much fund they have. For an example, if a company that have RM 300,000,000 funds but only earns RM 300,000 per year, it would mean that the company didn’t fully utilize the funds or the company didn’t invest in the right direction. This kind of company even they are earning money they are also considered as a bad investment because they didn’t help the shareholders to gain adequate profits.

5. **Financial health –** company does not have debt might be a good company to invest. At the same time, those companies that have debts are also considered as a good company for investment if they loan for investment in their business. The trend by company these days is to apply loan to invest in their business. This can make sure that they have more funds for investment purposes and by applying loans from bank, they can have better cash flow in hand and help the company to invest in other fields to earn more profit or as a backup for emergency cases. The interest of bank loan is very high. Thus, if the company cannot pay back their loan in time, they would have to pay the interest fees of the

loan. If they are unable to pay back the interest then the company will face bankruptcy because of the debt.

6. **Compare with competitors –** can the company compete with their competitors? The company can compete with competitors or not is also important because if it can't compete with their competitors, the company might go into bankruptcy to the point that the shares of the company will become useless. Investors must know well in the particular field that they plan to invest. They must know about the market and which company is the best or have more potential. This knowledge will help the investor to select the better stock for investment.

7. **Calculate intrinsic value using discounted cash flow or discounted EPS –** before investing in the selected stock, the selling and purchase prices of the stock needs to be calculated first to make sure that the stock is worth to buy or not. Before buying any stock, it was very important that stock analysis is done first though it might take the investor some time to make an analysis. Those analyses can help the investor to choose the correct company to invest in and the time spent on the analysis is definitely worth it.

2.2.3 Short term investment in stock market

Normally people that invest in share market for a short period of time will sell their share within 365 days which mean less than or equal to a year time

because if that share owner keeps the share for more than a year is called long term investment.

Short term investment normally will be high risk if compared with long term investment, as for those shares that can give high return in short period of time but potentially able to wipe out their entire investments in a very short period of time.

2.2.4 How to find out which short term stocks to buy

1. **Watch the moving averages –** a moving average is the average price for the particular stock in a period of time. Traders can use the moving averages to do their own research and analysis. All the moving averages data can be obtained through online.

2. **Understand the overall cycles or patterns –** market is like a cycle. Traders can look for the newsletter in the calendar about the stock price increment and decrement for the stock price. Traders have to guess themselves either to follow other's trends or not.

3. **Get a sense of market trends -** traders can sell a little bit of stock when the trend is negative and do a little buying. If the trend is positive, traders can buy it and do some selling on the particular product. When traders getting more and more experience in buying and selling they will know when to sell and buy the stock based on their experience.

4. **Purchase stock market trading software –** Use trading software to select the stock. The software only can help users to analysis the stock; traders also need to do their own research before they try to select the stock.

2.2.5 Risk in stock market

Risk is something that might cause the investors to lose their investment. When investors start their business, they have to know about risk. How big the risk is and they also have to know the risk for time durations such as monthly and yearly basis. It will help investors define either to invest for long term or short term.

The risk and return are equally important. If the stock is high risk but also have high return then the stock can be invested, but if the stock is high risk but low return then it shouldn't be considered because no one will invest in high risk and low return stock.

People mostly invest in stock market while ignoring the risks and hoping to obtain great return from it is quite impossible to do so. When people ignores the risk, that is the time when it is very dangerous as they don't know what to do when they encounter problems triggered by these risks.

The timing to start investment in stock and also the strategy that needed for stock investment is equally important. If investors don't have any strategy to invest in stock market it will be same as a company that runs their business without strategy. In all means, it does not necessary means that they will fail but then it means that they have a higher chance of failing.

Market is a very dangerous place but it is also a place which can earn lot of money. With an appropriate strategy, people can use market as a tool to help them to earn lot of money.

2.2.6 Type of market risk

1. **Economic risk –** if the economy goes bad the stock price will also follow. Some of the company might face bankruptcy when the economy goes bad.

2. **Inflation –** inflation is something that cannot be avoided for almost every year. The inflation will also cause the investors loss their money because the amount that they earn cannot cover back the amount of inflation. This situation will turn the earning to losing money.

3. **Incidents that happen on the country –** country that faces war will cause the price of the country stock to drop. It might also cause the companies that have business relations with the country to face drop in stock price.

2.3 How fund manager select stock

"The vendor queried sample funds their use of three types of models to evaluate portfolio risk: value at risk, stress testing and scenario analysis" (Gavin Cassar, January 7, 2011). Based on the statement, there's some fund manager using the 3 type of models to evaluate the portfolio risk. That's also some fund manager using other risk analysis like historical simulation, delta normal method, Monte Carlo method, and so on.

2.3.1 Value at risk:

VAR (value at risk) is used to calculate the maximum risk for a stock in normal market fluctuations. VAR will provide maximum loss of stock with given possibility (confidence interval of the loss) and a given numbers of days. That is the three elements of VAR: level of confident (the percentage of the stock that will not lose money), time period (in term of day, month and year for risk calculation), and estimate the loss (the percentage or how much money that will be loss).

Example:

```
level of confident = 99% (which mean that was only 1% chance
might have risk happen)
Time period = 1 day
Deal = 100000
Loss = 1.5% (This 1.5% is the total lose deal that have happen
in 1% of risk chance)
100000 * 1.5% = 1500
```

Based on the example above, it means that companies can have 99% of confident state that the money will not been loss for over 1500 in a day period.

2.3.2 Stress testing:

In portfolio risk analysis, stress testing use to test the stock which in the situation of economic plummeted, increasing of unemployment situation, real estate price plummeted and abnormal change of the market. This is to test whether the stock price will still remain the same or will the particular stock price be fluctuated.

The stress test is used to calculate the 1% out of 99% level of confident in the VAR module, which means it calculates the risk that might happen out of prediction. Stress test can calculate when unpredictable incident happen and how much does the company need to avoid it to be able to not go into bankruptcy.

2.3.3 Scenario analysis:

The scenario analysis uses to analysis the risk of the stock which might happen in the future. Predictions from analysis are such as future incidents, risks and harm that might happen because of the current happenings.

For an example: If the situation of unemployment keep increasing by 10% and continues until the following year, what will be the risk the company will get or how much they will lose if this situation continues.

2.3.4 Historical simulation:

Historical simulation uses previous data which mean historical data to predict the stock price of that particular stock tomorrow. Historical simulation also uses the VAR to calculate what potential loss that might occur for the particular stock the next day.

As an example:

```
level of confident = 99% (which mean there is only 1% chance
risk can happen)
Deal = 100000
Loss = 1.5% (This 1.5% is the loss of total deal that have
happened in 1% of risk chance)
100000 * 1.5% = 1500
```

Based on the example, if it is the historical data that was obtained from the company for the past 100 days, then the fund manager will have 99% of confident for the stock that will not cause loss which is higher than 1500.

2.3.5 Monte Carlo:

Monte Carlo uses random value running for multiple times or test on a particular stock in order to get the approximate probability outcomes. Monte Carlo was named after a casino name because it uses the technique just like gambling in order to calculate the results that might happen in future.

2.3.6 Delta-Normal method:

Delta-Normal method uses an assumption value and a formula to calculate the portfolio price in the future.

2.4 Artificial Intelligence systems for portfolio selection

2.4.1 Bayesian Network:

“Bayesian networks in conjunction with Bayesian statistical techniques facilitate the combination of domain knowledge and data.” (Jordan, 1998, p. 302) By this statement that states that the Bayesian network can work well if used together with the Bayesian statistics.

Bayesian statistics is the replacement of classical approach that uses frequent experiments to get the results. For classical approach, the result will be more accurate if the experiments are repeated for many times. Bayesian statistics are based on possibility of the data combination. The experiment will run until the possibility runs out or meet the final result.

“Bayesian method in conjunction with Bayesian networks and other types of models offers an efficient and principled approach for avoiding the over fitting of data” (Jordan, 1998) because for the Bayesian statistics method that has been using in Bayesian network, once the result from the data falls in one of the sources category, the experiments will stop. Thus, the money that is used to purchase the data will be reduced.

Example for Bayesian statistics:

For a particular portfolio that gets the sources stating that the percentage of the stock will increase its value up to 50% (which is equal to 0.5), another source states the stock will increase its value up to 70% (which was equal to 0.7).

After getting the sources the experiment can be started. If the results that have been return are in between the parameters, the experiments will be repeated as to re-calculate the stock prices. If the results are higher than 0.7 then it will fall into the 0.7 category and the experiments will be stopped.

2.4.2 Neural Network

The neural network was built based on the function of human brain and the neural network can perform calculation like the human brain. Just like a baby, after few times of the baby being taught how to recognize the alphabets or numbers or alpha have been written by using different hand writing, they can still recognize the alphabets or numbers.

"After seeing a number of images of legitimate users' faces, the network needs to determine accurately whether a new image corresponds to the face of a legitimate user or an imposter" (Martin Anthony, Peter L. Bartlett, 2009, p. 1) This statement has defined the neural network needs to be taught before using it. If the neural network has been fed more and more data, the result that it provides will be better. Just like a normal kid, after repeated sessions of learning then he will be cleaver.

The data that need to be fed to the neural network is not cheap. In order to create an accurate neural network, developers need a lot of money to purchase the data in order to fee1d the neural network program. The result might not be accurate if the data that was given to the neural network program is only a small amount of data.

“The data in the test set was used only AFTER all of the prediction and trading models had been trained and optimized” (S.P.Toulson, 1996, p. 6) This is because the neural network program needs to be trained first before it can be tested. The neural network needs to be fed by data, method and the formula in order to test the accuracy of the neural network to predict the future return of the selected portfolio.

“Of course, a neural network with a “sufficient” number of neurons can form maps arbitrarily close to any function” (Wan, 2002) He clearly stated that if the neural network program has been feed by a lot of data it can perform well. The term of sufficient is an unknown number because it is hard to define the level of sufficient, thus it will also be hard to define the results that it provides whether it is accurate or not.

2.5 Compare Artificial Intelligence systems for portfolio selection

The Bayesian network will be a suitable artificial intelligence program that is used to predict the return of portfolio stock. Neural network need to feed

"sufficient" data in order for it to calculate the return of the portfolio stock. If the data have been fed for the neural network is not "sufficient", the result that the neural network returns will not be an accurate result.

Bayesian network only require a small amount of data if compare with neural network in order to get the return of the portfolio. This is because the Bayesian network uses Bayesian statistics, thus it only needs to find out the result that falls in the sources from the previous result and the experiment can be stopped.

The neural network is much more unreliable because the term of "sufficient" is an unknown value. Hence, using the neural network to calculate the return of portfolio stock is risky. "The neural network failed to converge due to the large variation of the training data." (Tseng, 2003) Neural network might fail due to large variation of the training data.

Portfolio market is full of large variation data because the market value will keep changing. This is due to the company daily operations. It will also change because of other issues such as if it been feed large variation data. This will cause the Neural Network to face failure.

"From the experiment results we ran on an under-perform year, we can see that the Bayesian network system works better than C5.0 in a more general situation." (Tseng, 2003) This was the result from the experiment that the author has done. Bayesian network is better than the C5.0 and Neural network. The investment for portfolio will be a large amount of money, so the return for

the portfolio must be an accurate data in order for the investor to earn money from the selected portfolio.

The neural network result cannot be traced back of why it will return to this result. On the other hand, Bayesian network can trace back why the program will return such result. For fund managers, they might not trust the results that is given by the artificial intelligence system, but by using the Bayesian network the fund managers or users can trace back why the artificial intelligence system give such results. Based on the results, the fund managers or users can have the idea of which stock might increase its price in future.

2.6 Research Methodology

To develop this system, the programming language that I will use as an interface is PHP and Visual C++. The processing part with the artificial intelligence program will be the Bayesian Network. I will use the Bayesian Network artificial intelligence program as the main core because the Bayesian Network artificial intelligence program can keep track of why the artificial intelligence program will chose a particular answer rather than another.

Bayesian Network artificial intelligence program can keep track of how the system will give its answers. For an example, my portfolio risk analysis system that uses the Bayesian Network as the core of the system. The Bayesian Network artificial intelligence programs the stock price to either increase or decrease and it can keep track of why the system provides this kind of results.

As an example:

Today stock volume has increased. It might cause future decrease for seven days after stock closing price because other investors start selling off their stocks. The action of a lot of investors selling their stocks in hand at the same time will cause panic for other investors that have invested in same stock.

The training data that is used to train this portfolio stock support tools is downloaded from http://www.klse.info/downloads. Historical daily stock market trading can be attained for free. After the system has been build, the system will be trained by the historical stock market trading data. After training the system, it will be tested by using the historical stock market trading data. The results of the system prediction will be recorded and compared with the real results to check whether the system generates the correct results or not. The result's accuracy of the program will be marked down.

The Machine Learning-based decision support tools for portfolio risk analysis system provide seven days price increasing percentages, fourteen days price increasing percentages, and twenty one days price increasing percentages as the return results. That's because I use end of day stock data to create the Bayesian Network and test the Bayesian Network so today price increasing percentages is an input, not an output for my Bayesian Network.

I have created few prototypes in order to test the selected Bayesian Network structure's accuracy. I compared the Bayesian Network with three outputs, twenty one outputs and twenty outputs. The return result for three outputs having higher accuracy than the other two Bayesian Network. Further elaboration will be done in the compare Bayesian Network section.

2.7 Contributions:

This research will help people to understand about how a Machine Learning-based decision support tools can help them in portfolio selection. Even if results provided by Machine Learning-based decision support tools might not being 100% accurate, this research can give them an idea on how an artificial intelligence technology can help people in portfolio risk selection.

The traditional systems that use to support the brokers or fund managers in stock selection might be out of date. For an example, the rules based system might be out of date because the rules that have been set for the particular support system might not be applicable because the situation might change everything in real world. But the rules are already fixed, few years after it will also be the same set of rules.

Artificial intelligence might help fund managers to select the portfolio stock. By using the artificial intelligence technology, the fund managers does not need to always look at the graph or the analysis data order to select the better portfolio stock.

Artificial intelligence technology might help a lot in the portfolio risk selection fields because it can run the system automatically and the fund managers does not need to look after the data daily or time to time in order to select the portfolio stock. The more important things are the experience of the artificial intelligence system will be improve based on the amount of data that has been fed or train for it.

3.0 MACHINE LEARNING-BASED DECISION SUPPORT TOOLS FOR PORTFOLIO RISK ANALYSIS

3.1 Data Module

Historical data from year 1991 until 2012 that are used to develop Machine Learning-based decision support tools for portfolio risk analysis system was downloaded from http://www.klse.info/downloads.

Historical data from year 1991 until 2012 have been selected to develop the system because it is just approximately 20 years back from 2013, so it still considered fresh to be used to develop the decision support tools for portfolio risk analysis system. There might be some changes in the stock market during the 20 years period. Examples of changes are such as the Malaysian government policy for those companies and some other things that might affect the moving trend for the stock market. I've selected historical data from year 1991 until 2012 to develop the decision support tools for portfolio risk analysis system so that it can predict better of the current stock market.

Historical data for every company will join up to develop the Bayesian Network; join up historical data is required so the Bayesian Network only requires learning the data only once. This will make the Bayesian Network provide an accurate result and also easy to build the data that is used to feed Bayesian Network. Percentages are being used instead of values, so Machine Learning-based decision support tools for portfolio risk analysis are able to support multiple types' portfolio stocks.

As an example:

Stock A opening price is 10 and stock B opening price is 1, increasing of 1 for both stock will have huge different. Increasing 1 for stock A, pricing will only be a minor increment if compared with stock B. In order to provide better and accurate Machine Learning-based decision support tools for portfolio risk, percentages are being used instead of values. By using percentages, it enables the system support analysis for multiple types' portfolio stocks.

Historical data that has been downloaded contains few data which is up to date, open, highest prices, lowest prices, closing prices, volume of the stock and others. Only some historical data have been used to develop the Machine Learning-based decision support tools. Those data will be processed before it can be used to develop the Bayesian Network. History data that has been used to develop the Bayesian Network are open prices, highest prices, lowest prices, closing prices and also volume of the stock.

Data that are used to develop the Bayesian Network was opening price percentages, highest price percentages, lowest price percentages, stock volume percentages, three days average opening price percentages, seven days average opening price percentages, fourteen days average opening price percentages, highest price minus lowest price percentages, price increasing during off market percentages, price increasing percentages, seven days price increasing

percentages, fourteen days price increasing percentages, and twenty one days price increasing percentages.

In order to make sure all those data can create a better and accurate Bayesian Network, they are divided by zero data which will be replaced by NULL. Bayesian Network will generate output even if some of the input data is empty.

3.1.1 Prepare data for Bayesian Network

Bayesian Network is created by using discretize value. The calculated data will be discretizing in order to create the Bayesian Network. After discretize, opening price percentages, highest price percentages, lowest price percentages, stock volume percentages, Three days average opening price percentages, Seven days average opening price percentages, Fourteen days average opening price percentages, highest price minus lowest price percentages, price increasing during off market percentages, price increasing percentages, seven days price increasing percentages, fourteen days price increasing percentages, and twenty one days price increasing percentages will have their own range. Every single data range will be different because their values are not the same. In the end of day, stock data will be processed into consumable data for Bayesian Network.

Data will be converted into Bayesian Network consumable data so when consumed, the Bayesian Network through Visual C++ program will be faster. All range are retrieved from the discretize value range when creating the

Bayesian Network. It has been used to create Bayesian Network as consumable data. As an example, increasing price percentages for network = 0, which means the value from calculated data falls in the first range of increasing price percentages range in Bayesian Network. Bayesian Network allows those unsure data to replace with empty string. This means that if the calculated data is empty or an error value, it will be replaced to an empty string when converting the calculated data into Bayesian Network consumable data. All above is the data that is used to generate results in Bayesian Network.

3.1.2 Results that are generated by Bayesian Network

Seven days increasing percentages, fourteen days price increasing percentages and twenty one days price increasing percentages are the results return from Bayesian Network. Each of the results will have four ranges. The reason of the four ranges are because when discretize, value will use 4 bin count which also means the data will be separated into 4 ranges. When developing the Bayesian Network, raw data has been discretized and it has been separate into 4 ranges. Thus, results that have been provided by Bayesian Network will also have 4 ranges. The result will show the possibility of the stock opening in future, highest percentages for the range represents the stock opening price has higher chances to fall into that range.

Note: This is the result that is generated by Bayesian Network by using real time data. All the results will be stored into a text file as a reference purpose. First result means it is the first range for the result. Followed by, the second

result which means it is second range for the result and so on. Results have been retrieved from Bayesian Network through Visual C++. Results that are generated from Bayesian Network will multiply by 100 in order to get the percentages for the results. All the results will be recorded into a text file that will be shown to clients. Data that are used for the test of Bayesian Network will also be discretized using 4 bin counts. Both data that are used to discretize data for testing and developing Bayesian Network are the same one in order for the Bayesian Network to be tested.

3.2 Technical Module

Machine Learning-based decision support tools for portfolio risk analysis have two parts. The first is used to determine the accuracy of the Bayesian Network and the other is used to be consumed by the Bayesian Network for real time data. The reason of separating the technical module into two parts is because it will enable me to test the accuracy of the system faster. The Bayesian Network that is used in testing and consuming is the same.

Machine Learning-based decision support tools for portfolio risk analysis Bayesian Network selected among 71 Bayesian Network that was created and tested. The selected Bayesian Network has the highest accuracy rate among other Bayesian Networks. The Bayesian Network selected has created automatic in Genie and Smile through the raw data that have been fed for it. I have set the background knowledge for the Bayesian Network when I develop

it. The accuracy of Bayesian Network is tested by using Genie and Smile builds in tools to check the results.

The Machine Learning-based decision support tools for portfolio risk analysis system are created by few types programming language. I choose Bayesian Network as artificial intelligence for the system. It is produced by using Genie and Smile. The program that is used to consume the Bayesian Network is Visual C++ 2010 and to for process to convert raw data into consumable data is done by using PHP.

Technical model have two steps. First is to prepare testing, developing data, and accuracy for the Bayesian Network. The other is the program that is used to consume the Bayesian Network. I used PHP and Visual C++ in order to develop the programs, so each of them will also have 2 parts. The first part is to test the Bayesian Network and the other is to consume the Bayesian Network.

3.2.1 Bayesian Network

Data that are used to develop the Bayesian Network is the opening price percentages, highest price percentages, lowest price percentages, stock volume percentages, three days average opening price percentages, seven days average opening price percentages, fourteen days average opening price percentages, highest price minus lowest price percentages, price increasing during off market percentages, price increasing percentages, seven days price increasing

percentages, fourteen days price increasing percentages, and twenty one days price increasing percentages.

Those data are used to develop the Bayesian Network because they only have limited historical information about the stock and it can be retrieved from online for free thus chosen as ideal data to develop the Bayesian Network. Besides that, those default historical data that I have retrieved, I add in some formula in it, in order to provide a better value to develop the Bayesian Network.

Some of the books stated that those data are also important to use for prediction in the future stock prices. Based on “Winning stocks seldom drop 8% below a correct pivot-point buying price” (O'Neil, 1995, p. 103) which means that if the lowest price percentages are more than 8% compared with the opening price, it is not a good stock to select and it might face price decrease in future. On the other hand, it also means that lowest price percentages will affect stock price in future.

Besides that, based on “If after a stock's price is extended from a proper base, its price closes for a larger increase than on any previous up days, watch out! This move usually occurs at or very close to a stock's peak” (O'Neil, 1995, p. 104) which means that the highest price percentages was higher than the opening price, stock price will be increase in future due to high demand for the particular stock. Thus, highest price percentages will affect the stock price in future.

Based on “The ultimate top may occur on the heaviest volume day since the beginning of the advance” (O'Neil, 1995, p. 104) and also “New highs on decreased or poor volume means there is temporarily no demand for the stock at that level and selling may soon overcome the stock” (O'Neil, 1995, p. 104) volume of the stock will also affect the stock price in future. The more people who purchase the stock will cause the stock volume to decrease. In result, it will also cause the stock price increase, since it is high demand but low supply. On the other hand, if stock have large volume and the volume level keeps increasing which mean investors start selling off that stock, it will result in low demand high supply which will cause the stock price decrease. Based on this two points, I added in the stock volume percentages in develop the Bayesian Network.

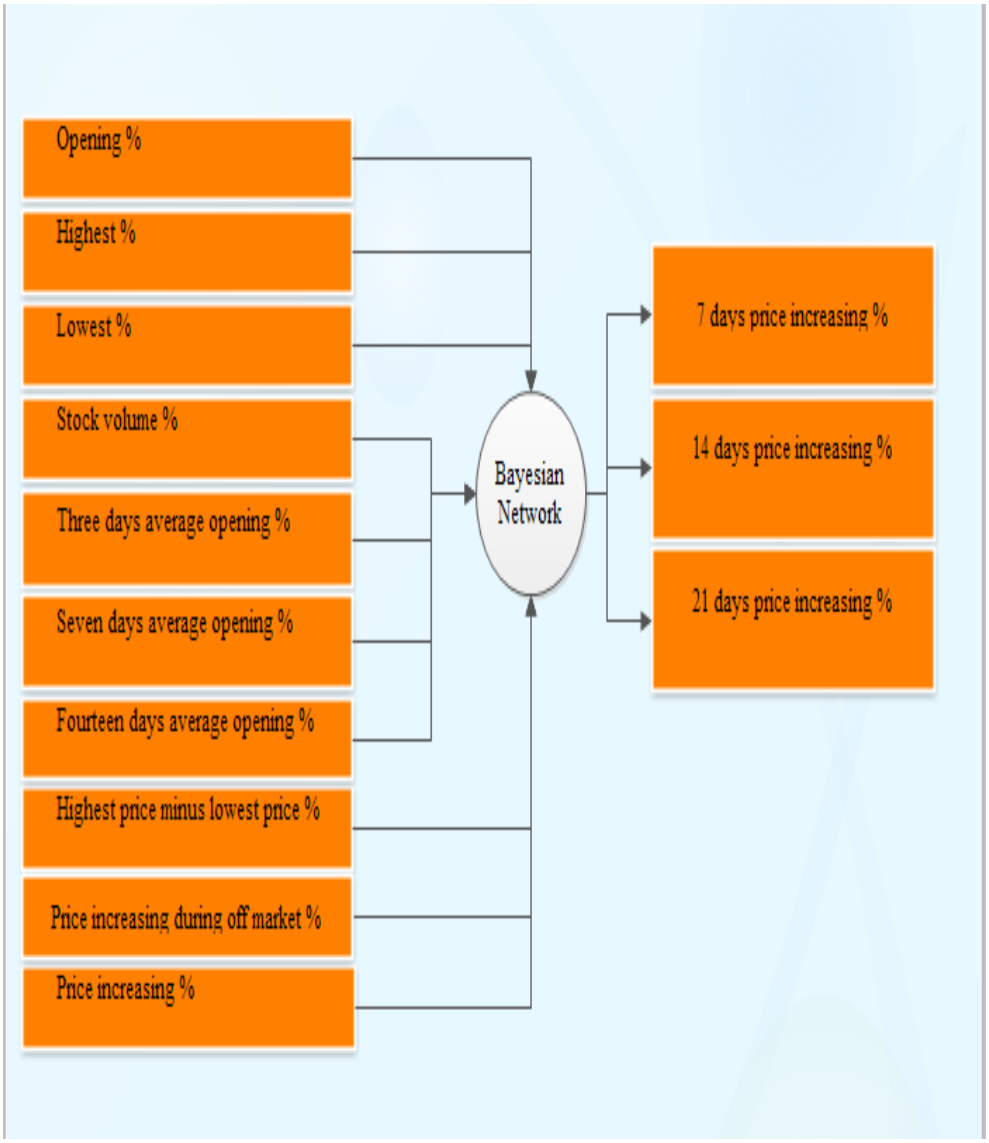
Opening price percentages, closing price percentages and also price increasing during off-market percentages will also affect the stock price in future thus it is included in developing the Bayesian Network. Based on “… extended from its base and opens on a gap up in price, the advance is probably near its peak. A two-point gap in a stock's price would occur if it closed at its high of $50 for the day and the next morning opened at $52 and held above $52 during the day” (O'Neil, 1995, p. 104) which means that the opening price, closing price and the increment during off-market is important to determine stock peak. Stock peak means that the stock is almost at the highest point for that cycle, so it is the time for the stock price to start falling down.

For three days average opening price percentages, seven days average opening price percentages, fourteen days average opening price percentages is selected to develop the Bayesian Network. This is because based on “Sell if a stock advance gets so active that it has a rapid price run-up for two or three weeks (eight to twelve days). This is called climax (blow-off) top activity.” (O'Neil, 1995, p. 104). The stock opening price run up to for two to three weeks as it might cause the change of the stock price in future.

The results was seven days, fourteen days and twenty one days but no shorter or longer period as investors who invest in the market, need some time in order to sell out the on hand stocks and also to purchase those potential stock which might bring them profit in future. In stock market, whoever that wants to sell out the stock might need to have someone who is willing to buy it. If the condition is not met, the stock cannot be sold out.

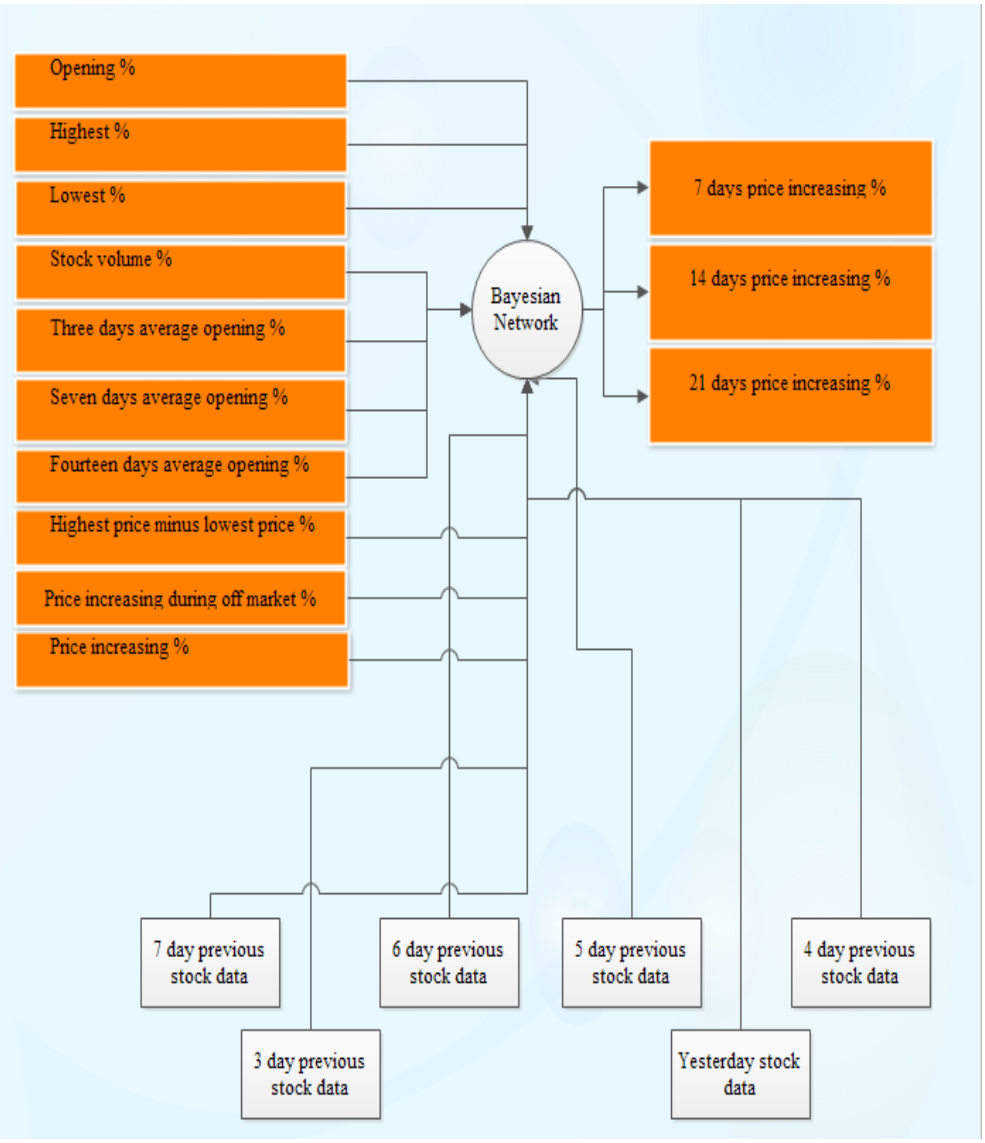
Base on “Institutional sponsorship. Buy stocks with at least few institutional sponsors and having better than average recent performance records” (O'Neil, 1995, p. 78), investors can sell out the company stocks that have institutional sponsorship because the institutional sponsorship will purchase most of the stocks for that company from market. Due to time concern, the results that are provided by the system is seven days, fourteen days and twenty one days, three weeks period as a good time for investors to purchase and also to sell out stocks.

3.2.1.1 Bayesian Network with only today data overview diagram



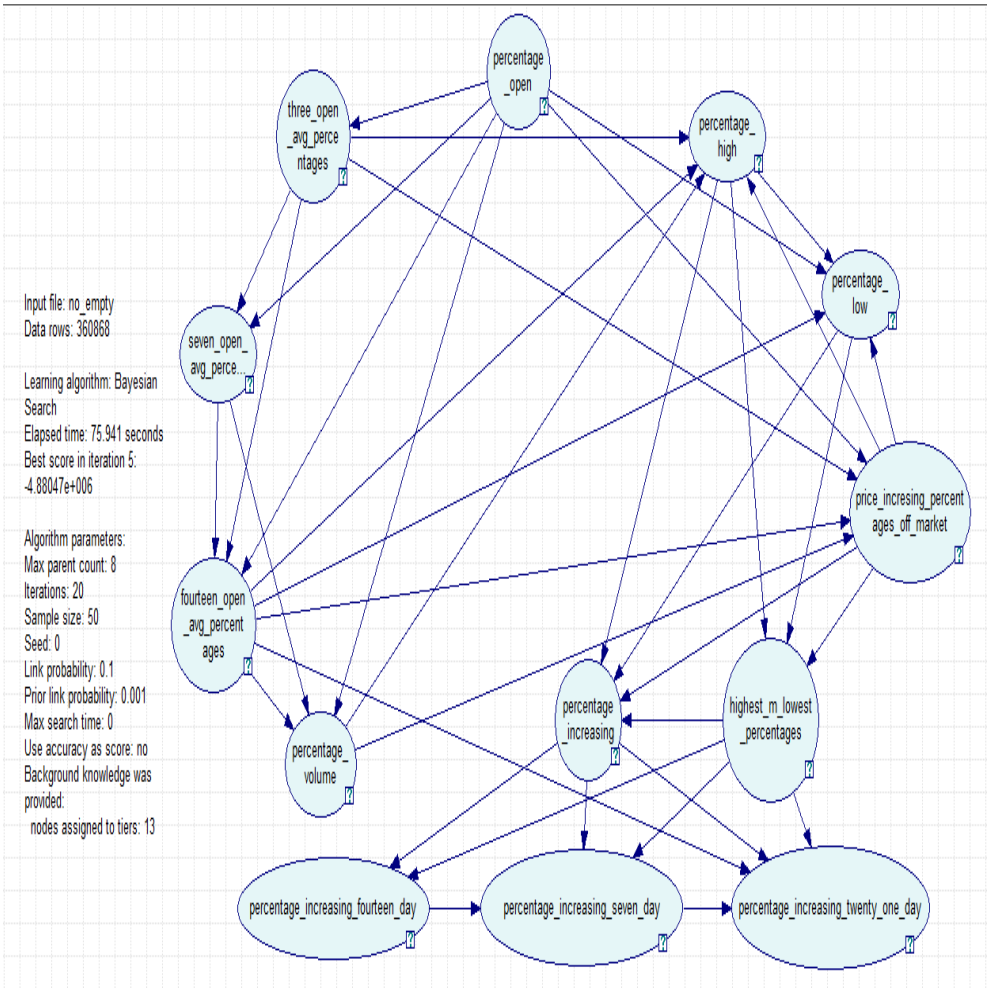
Besides the chosen Bayesian Network, I tried to build others Bayesian network by using other input but the accuracy of the result is lower than the selected Bayesian Network with the above input. I have added in seven day previous, six day previous, five day previous, four day previous, three day previous, and yesterday data set as the input for the New Bayesian Network.

3.2.1.2 Bayesian Network with seven days data overview diagram



Seven days previous stock data, six days previous stock data, five days previous stock data, four days previous stock data, three days previous stock data, yesterday stock data was previously stock data for opening price percentages, highest price percentages, lowest price percentages, stock volume percentages, three days average opening price percentages, seven days average opening price percentages, fourteen days average opening price percentages, highest price minus lowest price percentages, price increasing during off market percentages and price increasing percentages.

3.2.1.3 Most accurate Bayesian Network (with set background knowledge)



3.2.1.3.1 Nodes represent

Percentage_open = opening price percentages

Percentage_high = highest price percentages

Percentage_low = lowest price percentages

Percentage_volume = stock volume

Three_open_avg_percentage = three days average opening price percentages

Seven_open_avg_percentage = seven days average opening price percentage

Fourteen_open_avg_percentage = fourteen days average opening price percentages

Highest_m_lowest_percentages = highest price minus lowest price percentages

Price_increasing_percentages_off_market = price increasing during off market percentages

Percentage_increasing = price increasing percentages

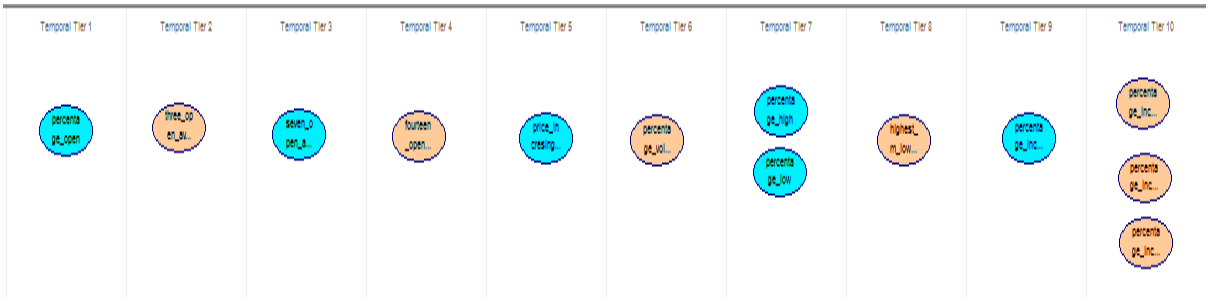
Percentage_increasing_seven_day = seven days price increasing percentages

Percentage_increasing_fourteen_day = fourteen days price increasing percentages

Percentage_increasing_twenty_one_day = twenty one days price increasing percentages

3.2.1.4 Background knowledge for Bayesian Network

3.2.1.4.1 Most accurate background knowledge for Bayesian Network with only today data



3.2.1.4.1.1 Most accurate background knowledge for Bayesian Network with only today data

Temporal Tier	Data

1	• Today opening price percentages
2	• Today three days average opening price percentages
3	• Today seven days average opening price percentages
4	• Today fourteen days average opening price percentages
5	• Today price increasing during off market percentages
6	• Today stock volume percentages
7	• Today highest price percentages • Today lowest price percentages
8	• Today highest price minus lowest price percentages
9	• Today price increasing percentages
10	• Today seven days price increasing percentages • Today fourteen days price increasing percentages • Today twenty one days price increasing percentages

Today opening price percentages is the first which have been set as the first item in background knowledge for developing the Bayesian Network. The reason is because it will affect other nine data. Almost all of the Bayesian Network data uses opening price to create the percentages, so opening price definitely will affect other data that are used to develop the Bayesian Network.

Bayesian Network that place opening price percentages on first place will create higher accuracy Bayesian Network than the Bayesian Network which place fourteen days average opening price percentages in first temporal tier,

seven days average opening price percentages in second temporal tier, three days average opening price percentages in third temporal tier and opening price percentages in fourth temporal tier.

Besides that, Bayesian Network which place opening price percentages place in first temporal tier is followed by three days average opening price percentages, seven days average opening price percentages, and fourteen days average opening price percentages for having higher accuracy than Bayesian Network which place fourteen days average opening price percentages, seven days average opening price percentages, three days average opening price percentages and opening price percentages place in same tier

Fifth temporal tier price increased during off market percentages. Followed by stock volume percentages and the reason why I place price increasing during off market percentages follow by stock volume percentages is because if the price increase during off market percentages is changing, it mean someone is selling or purchasing the stock during after-market hours. Large changes on price increase during off market percentages will definitely occur after the stock volume.

Based on "…pre-market, regular market and after-hours market. The regular market trades between 9:30am and 4pm EST; the pre-market trades from 8am to 9:30am EST; and the after-hours market trades from 4:30pm to 8pm EST" (Investopedia Staff, 2009) which means besides the regular market trades, there are after-hours market trades and pre-market trades. Data that I used to develop

the Bayesian Network is the end of day data. End of day data means it will be generated after the regular trading time, if any trade was making on after-hours market trades and pre-market trades it will affect price increasing during off market.

First temporal tier up to sixth temporal tier have been set; now follow with the seventh temporal tier. Stock volume percentages place at sixth temporal tier and followed by highest and lowest price percentages because stock volume percentages will affect highest and lowest price percentages. If stock volume increase, it means investors is selling their stocks in hand. If stock volume increase with a very fast rate, it means many investors are selling their stock in hand, and it will cause the stock price to drop due to the amount of selling is greater than amount of purchasing. This action will definitely affect the highest and lowest price percentages.

Highest and lowest price percentages being place in same temporal tier is because both of them are actually the same level. Highest price percentages will not affect lowest price percentages. This have been proven by the experiment, Bayesian network with highest price percentages in eighth temporal tier and followed by lowest price percentages in ninth temporal tier or Bayesian Network with lowest price percentages in eighth temporal tier follow by lowest price percentages in ninth tier have the lowest accuracy compared to the Bayesian Network that place highest and lowest price percentages at the same temporal tier.

Highest price minus lowest price percentages is placed after highest and lowest price percentages because highest price minus lowest price percentages is the result that comes from highest price minus with lowest price. It will definitely be affected by highest and lowest price percentages.

Price increasing percentages placed after highest price minus lowest price percentages because the closing price of the day will be affected by all other values and closing price is the last value for the day. Hence, price increasing percentages should be placed at the last.

Seven days price increasing percentages, fourteen days price increasing percentages, twenty one days price increasing percentages place after price increasing percentages is because those data will be affected by price increasing percentages. That is the few prototypes that I have created which place price increasing percentages, seven days price increasing percentages, fourteen days price increasing percentages, twenty one days price increasing percentages at different places.

First, I placed all of the four prices of increasing percentages at same temporal tier. Followed by, the second prototype I placed the price increasing percentages in the first temporal tier followed by seven days price increasing percentages in second temporal tier and so on. The others experiment that make the return result are also included. Finally I have found out that the Bayesian Network which the return higher accuracy is Bayesian Network which placed price increasing percentages in one temporal tier and seven days

price increasing percentages, fourteen days price increasing percentages, twenty one days price increasing percentages together at another temporal tier.

3.2.1.4.1.2 Most accurate background knowledge for Bayesian Network with seven days data

Temporal Tier	Data
1	• Seven day previous opening price percentages
2	• Seven day previous three days average opening price percentages
3	• Seven day previous seven days average opening price percentages
4	• Seven day previous fourteen days average opening price percentages
5	• Seven day previous price increasing during off market percentages
6	• Seven day previous stock volume percentages
7	• Seven day previous highest price percentages • Seven day previous lowest price percentages
8	• Seven day previous highest price minus lowest price percentages
9	• Seven day previous price increasing percentages
10	• Six day previous opening price percentages
11	• Six day previous three days average opening price

	percentages
12	• Six day previous seven days average opening price percentages
13	• Six day previous fourteen days average opening price percentages
14	• Six day previous price increasing during off market percentages
15	• Six day previous stock volume percentages
16	• Six day previous highest price percentages • Six day previous lowest price percentages
17	• Six day previous highest price minus lowest price percentages
18	• Six day previous price increasing percentages
19	• Five day previous opening price percentages
20	• Five day previous three days average opening price percentages
21	• Five day previous seven days average opening price percentages
22	• Five day previous fourteen days average opening price percentages
23	• Five day previous price increasing during off market percentages
24	• Five day previous stock volume percentages

25	• Five day previous highest price percentages • Five day previous lowest price percentages
26	• Five day previous highest price minus lowest price percentages
27	• Five day previous price increasing percentages
28	• Four day previous opening price percentages
29	• Four day previous three days average opening price percentages
30	• Four day previous seven days average opening price percentages
31	• Four day previous fourteen days average opening price percentages
32	• Four day previous price increasing during off market percentages
33	• Four day previous stock volume percentages
34	• Four day previous highest price percentages • Four day previous lowest price percentages
35	• Four day previous highest price minus lowest price percentages
36	• Four day previous price increasing percentages
37	• Three day previous opening price percentages
38	• Three day previous three days average opening price percentages

39	• Three day previous seven days average opening price percentages
40	• Three day previous fourteen days average opening price percentages
41	• Three day previous price increasing during off market percentages
42	• Three day previous stock volume percentages
43	• Three day previous highest price percentages • Three day previous lowest price percentages
44	• Three day previous highest price minus lowest price percentages
45	• Three day previous price increasing percentages
46	• Yesterday opening price percentages
47	• Yesterday three days average opening price percentages
48	• Yesterday seven days average opening price percentages
49	• Yesterday fourteen days average opening price percentages
50	• Yesterday stock volume percentages
51	• Yesterday price increasing during off market percentages
52	• Yesterday highest price percentages

	• Yesterday lowest price percentages
53	• Yesterday highest price minus lowest price percentages
54	• Yesterday price increasing percentages
55	• Today opening price percentages
56	• Today three days average opening price percentages
57	• Today seven days average opening price percentages
58	• Today fourteen days average opening price percentages
59	• Today stock volume percentages
60	• Today price increasing during off market percentages
61	• Today highest price percentages • Today lowest price percentages
62	• Today highest price minus lowest price percentages
63	• Today price increasing percentages
64	• Seven day previous seven days price increasing percentages • Seven day previous fourteen days price increasing percentages • Seven day previous twenty one days price increasing percentages
65	• Six day previous seven days price increasing percentages

	• Six day previous fourteen days price increasing percentages • Six day previous twenty one days price increasing percentages
66	• Five day previous seven days price increasing percentages • Five day previous fourteen days price increasing percentages • Five day previous twenty one days price increasing percentages
67	• Four day previous seven days price increasing percentages • Four day previous fourteen days price increasing percentages • Four day previous twenty one days price increasing percentages
68	• Three day previous seven days price increasing percentages • Three day previous fourteen days price increasing percentages • Three day previous twenty one days price increasing percentages
69	• Yesterday seven days price increasing percentages • Yesterday fourteen days price increasing percentages

	• Yesterday twenty one days price increasing percentages
70	• Today seven days price increasing percentages • Today fourteen days price increasing percentages • Today twenty one days price increasing percentages

3.2.3 PHP

3.2.3.1 PHP used to develop Bayesian Network

Raw data will be converted into percentages before it can be used to develop the Bayesian Networks. Those raw data are be processed by using PHP. In PHP first step, raw data are separated. After that, it is grouped up into companies based. The reason of grouping them by companies is because it will be easier to do the calculation. Due to the size of the historical data, the program is run for four times. Every time the program runs, the program will process 5 years' worth of data. After that, every historical data that have been separated will place into another folder called as company data for the second step. In first step, those useless data will be removed from the historical data and then the new process data will be saved into new a file that based on the stock's name. Useless data are such as date, company name and other data that which is not in use in developing the Bayesian Network.

For the second step, PHP will read all files that are inside company data file and will process all the files. It will then turn all the raw data into process data. These processed data will be stored into different direction based on the

completeness of the data. Each column of data in the same row if do not have empty value then it will be stored into no empty data file, but if it has empty value in it then it will be stored into got empty value file.

Processed data has been separate in to no empty data and having empty data. No empty processed data is used to develop the Bayesian Network. When building the Bayesian Network, empty value is disallowed in the data file and when validating the Bayesian Network, empty value is allowed in the data file. Data that are used to develop the Bayesian Network and data that are used to test the Bayesian Network are totally different so the generate results will be trustable.

As an example:

```
$thisdir = getcwd();

$raw_data_dir                                =
$thisdir."/raw_data/";
$company_data_dir                                  =
$thisdir."/company_data/";
$no_empty_dir                                =
$thisdir."/no_empty/";
$got_empty_dir                               =
$thisdir."/got_empty/";
```

The above codes are all those file direction for the raw data, company data and the processed data storage location. $thisdir = getcwd(); this will return the based location for the file, so when selecting data or store data, there are no need to declare full path of the files location. $raw_data_dir is the

direction stores of all the raw data that before separate it based on company. $company_data_dir is the direction stores all the separate companies' data, which will be used in step two.

Data that have been processed in step two will store in two different directions, which are $no_empty_dir and $got_empty_dir. $no_empty_dir This folder will store those data that did not have any empty value for opening price percentages, highest price percentages, lowest price percentages, stock volume percentages, three days average opening price percentages, seven days average opening price percentages, fourteen days average opening price percentages, highest price minus lowest price percentages, price increasing during off market percentages, price increasing percentages, seven days price increasing percentages, fourteen days price increasing percentages, and twenty one days price increasing percentages. If data is empty for one of the above, that data will be stored in $got_empty_dir.

```
foreach(scandir($company_data_dir) as $file_name){
      …
}
```

Above coding was used to read all the data from company data folder. Scandir means that to scan the entire files inside the direction. $company_data_dir represent the company data direction, but the folder direction is declared on top of the code.

3.2.3.2 PHP used to test Bayesian Network

PHP program that is used to test the Bayesian Network is actually the same one that is used to create the process data to be consumed in Bayesian Network. In order to speed up the process for testing the Bayesian Network accuracy through Visual C++, the process data has been chopped 30000 data into one

file, so every time the Visual C++ program run, it only need to process 30000 data but not all in one shot.

Daily data or historical daily data will be processed in this step. It is different from the data that have been processed for usage of testing the accuracy of Bayesian Network and develop the Bayesian Network in Genie and Smile. Process data that is used in Genie and Smile are the actual value, which means decimal or float. Those data will be changed to discretise data in order to test and develop Bayesian Network in Genie and Smile.

Daily data or historical data that is used to test or consume the Bayesian Network through Visual C++ will be processed and the data will be fitted in into the range which where they belong. Zero, one, two three and empty string will replace the decimal value, so it is not decimal value like those data that is used in the development and testing of the Bayesian Network in Genie and Smile.

As an example:

```
If (Opening price percentages === "") {
      Opening price percentages for network = ""
} else if (Opening price percentages < -2.272727) {
      Opening price percentages for network = 0
} else if (Opening price percentages >= -2.272727 AND Opening
price percentages < 0.3496504) {
      Opening price percentages for network = 1
} else if (Opening price percentages >= 0.3496504 AND Opening
price percentages < 2.673797) {
```

```
        Opening price percentages for network = 2
} else if (Opening price percentages >= 2.673797) {
        Opening price percentages for network = 3
} else {
        Opening price percentages for network = “”
}
```

This is the process data that is used to test the Bayesian Network through Visual C++. The processed data is used to check which category the data belongs to. This data will be used for testing the Bayesian Network. Data like 2.673797 that has been used in Visual C++ is discretizing data from Bayesian Network. There are 4 ranges for the results because when discretize data for developing Bayesian Network, I used Bin count 4 to discretize data so it will only provide 4 data ranges for every single data that have been used to develop the Bayesian Network. This includes the results that have been generated by the Bayesian Network which is also 4 data ranges.

3.2.3.3 PHP used to display the result

After all the data that has been processed in Visual C++, it will be stored into a file. This is done in order for the PHP to select all those data that are inside the file to be used as a display purpose. By using Ajax function to call the result action file and through the action file to get the result from text file and then display it on web.

3.2.4 Visual C++

3.2.4.1 Visual C++ used to test Bayesian Network

In Visual C++, the first step is to retrieve the entire processed data. Processed data is processed by PHP program and then it will be stored at a location. Visual C++ will retrieve those data from that location for validation. Beside that data, Visual C++ has to enable Smile library in it, so that Visual C++ can consume the Bayesian Network through the Smile library. Bayesian Network location also have to be included in Visual C++ program, so it will know where is the Bayesian Network location and can consume that Bayesian Network.

In Visual C++, that is a header file that contains the functions that using Smile library to communicate with Bayesian Network. After retrieving the processed data and included the Bayesian Network into the Visual C++ program, it will then function calls to perform integrate the data with Bayesian Network.

For the function, the first thing that it has to do was to declare the entire possible variables that will be used in the program. After that, the declared data will be assigned with nodes in Bayesian Network. Then those nodes will fill the data from the processed data and each result that return from Bayesian Network will be assigned into a value in Visual C++.

This is followed by the result which will be compared. If the actual result falls in between 2 range of the option in Bayesian Network then the second range

result must be higher compared with other range result. Then only it can be counted as a correct result.

3.2.4.2 Visual C++ used to consume Bayesian Network

Visual C++ for consume Bayesian Network coding is almost similar with Visual C++ for test Bayesian Network. But there is some difference in it. Visual C++ for Bayesian Network to consume, records down all the result into a text file. The text file will be used to display at the website through PHP coding. In order to store the result in text file, Visual C++ for consummation purposes by Bayesian Network will definitely require the location for the text file to save.

As an example:

```
char today_resultDir[300]          =              "C:\\Program
Files\\wamp\\www\\klse\\result\\"; /* Direction of where the
result being store */
strcat(today_resultDir,buffer);
```

```
The above code that has been shown is the code that uses to
point the location which the result text files will be store at.
```

```
fprintf(fp_result, "%s",stock_name);
fprintf(fp_result, "%s",",");
fprintf(fp_result, "%lf",result1_seven_day*100);
fprintf(fp_result, "%s",",");
fprintf(fp_result, "%lf",result2_seven_day*100);
fprintf(fp_result, "%s",",");
fprintf(fp_result, "%lf",result3_seven_day*100);
fprintf(fp_result, "%s",",");
```

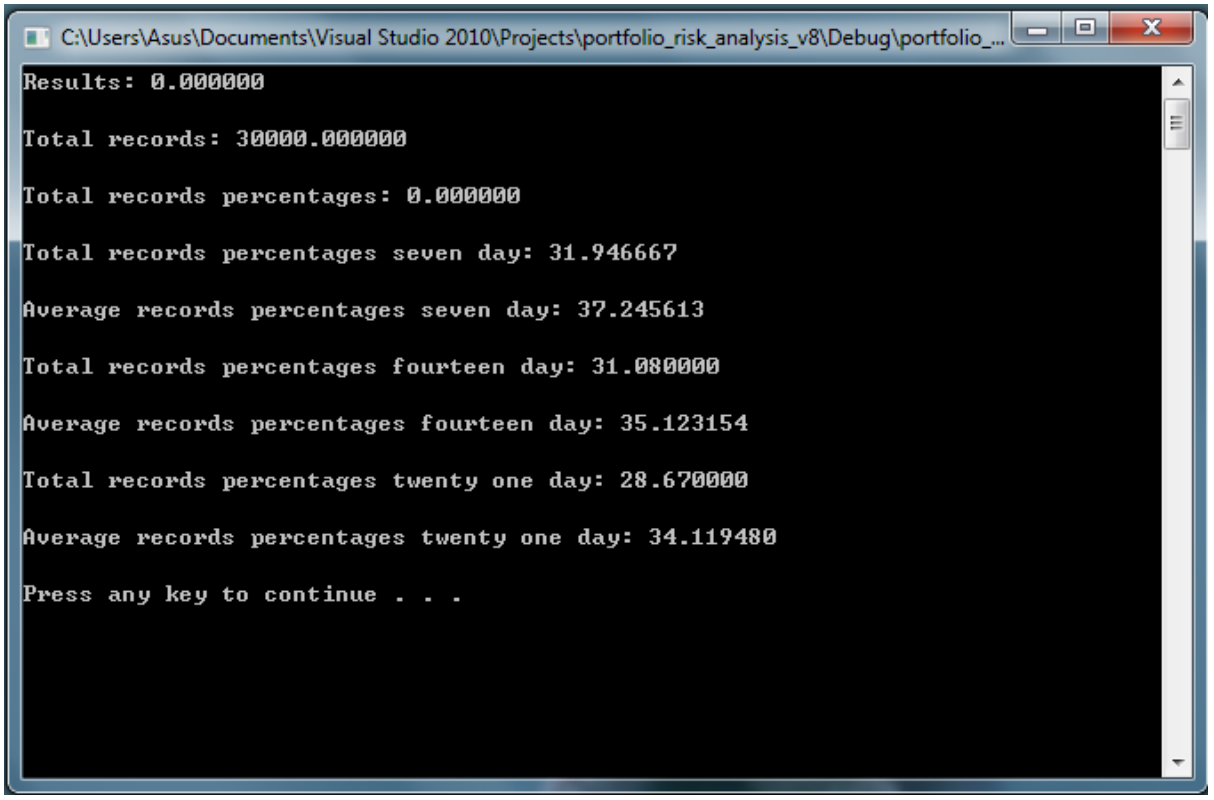
```
fprintf(fp_result, "%lf",result4_seven_day*100);
fprintf(fp_result, "%s",",");
```

Stock name will be stored into text file, so that users can know which of the results it belongs with are. There are 4 results range which will return from Bayesian Network for every single result. The return results will show the possibility of the stock price in future that falls in which range. Every single range will have its own possible percentages. The higher percentages which mean the stock price will have higher chances to fall into the particular price range.

3.3 Testing Module

There will be a testing Module for the Bayesian Network that is used in Machine Learning-based decision support tools for portfolio risk analysis system. The actual Bayesian Network will be tested by using two different methods. One of it is the testing on Genie. There is a function in Genie which is after users open the data field and network, they only have to click on the validation function and then select the validate field, and finally the results will be generated. Plus, the confusion matrix for the selected fields will also be provided. This is the easiest way to test the Bayesian Network. The example result that is provided by Genie is show on Bayesian Network part on top.

The second way to test the Bayesian Network is through the Visual C++ that consumes Bayesian Network by using Smile library. Precision of Bayesian Network can be tested through Visual C++. Results generated by Bayesian Network will be displayed in Visual C++.



The above is the example result that is generated by Visual C++ for testing purpose.

Total records percentages = the accuracy of the result / total record

Average records percentages = sum up percentages of particular stock / results which correctly falls into that range.

The result is generated by using 30000 of portfolio stocks historical data that have been processed by PHP. Using 30000 of portfolio stocks historical data but not all historical data is because it was used to test the accuracy of Bayesian Network and the connection between Visual C++ and Bayesian

Network. Thus there is no need to use all historical data as 30000 historical data will be more than enough.

4.0 RESULTS AND ANALYSIS

4.1 Bayesian Network with only today data without learning parameter (without set background knowledge)

Accuracy for all 3 nodes = 0.363

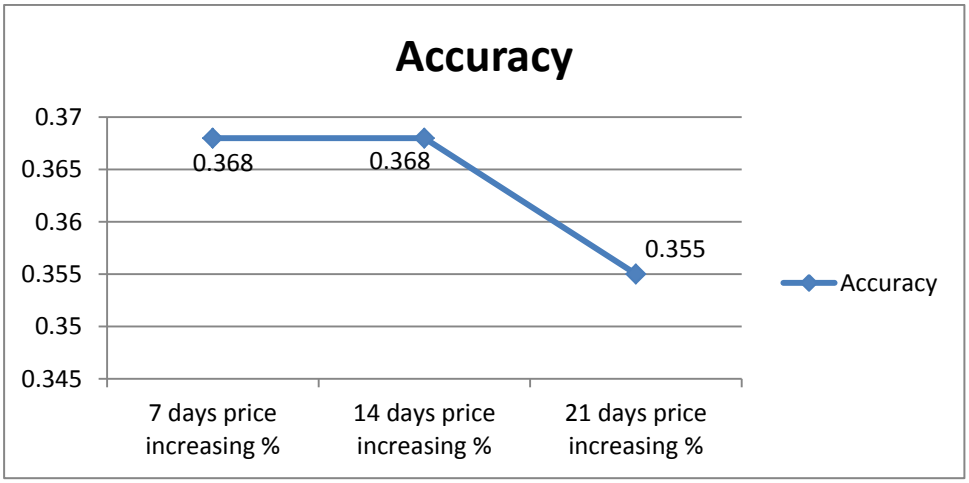


Figure 4-1

4.2 Bayesian Network with only today data without learning parameter

Accuracy for all 3 nodes = 0.464

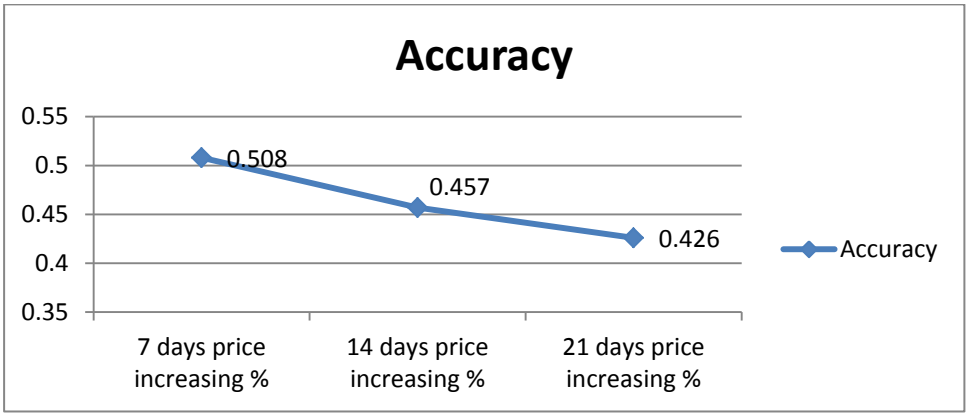


Figure 4-2

4.3 Bayesian Network with only today data with learning parameter

Accuracy for all 3 nodes = 0.476

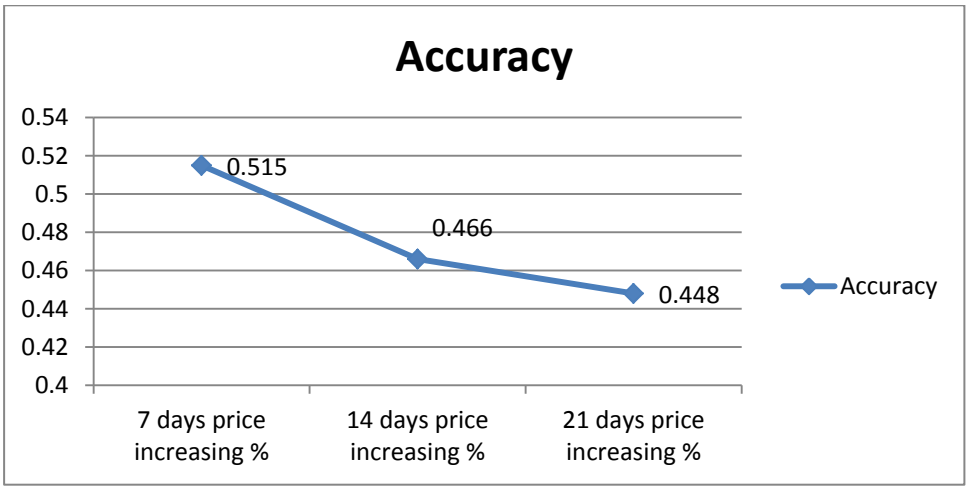


Figure 4-3

4.4 Bayesian Network with only today data with learning parameter (include price increasing percentages)

Accuracy for all 4 nodes = 0.562

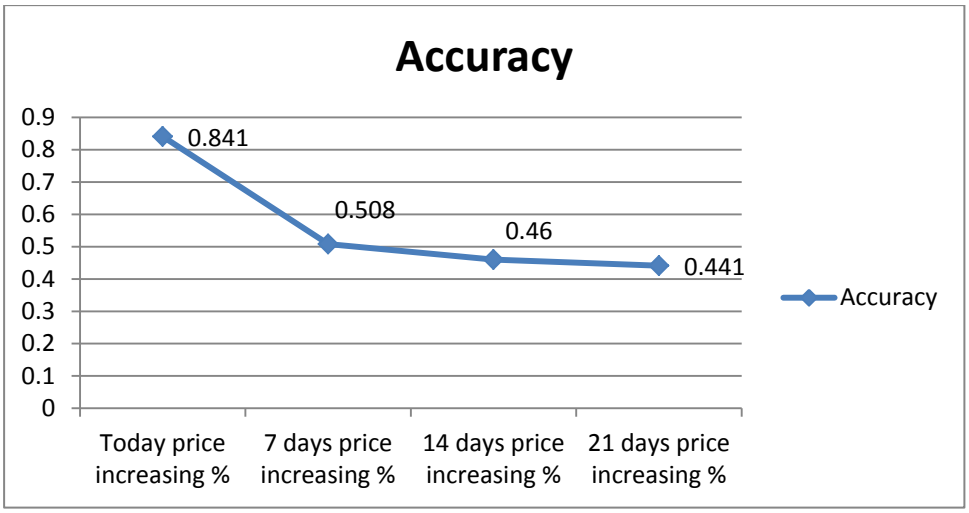


Figure 4-4

4.5 Bayesian Network with seven days data without learning parameter

Accuracy for all 21 nodes = 0.425

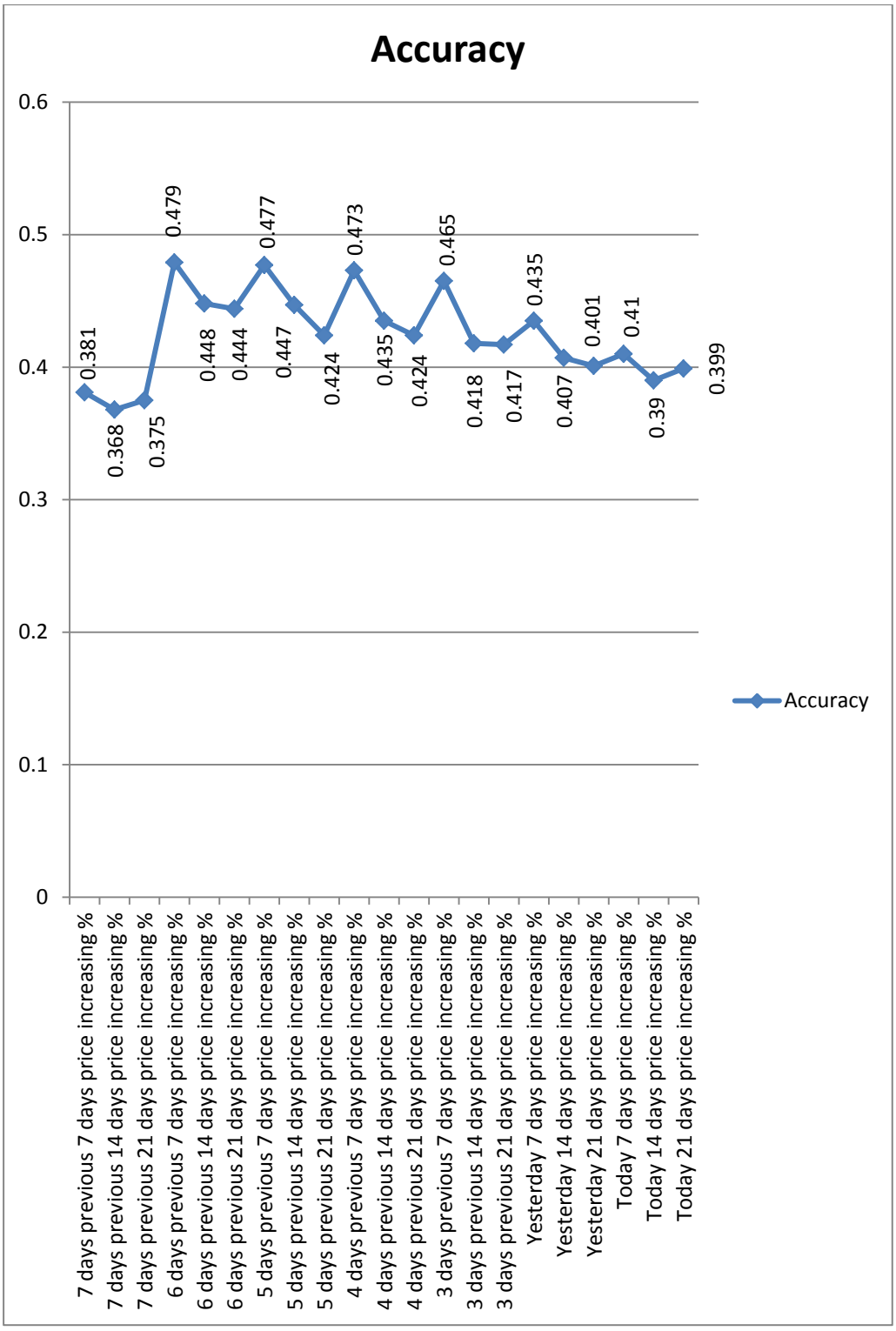


Figure 4-5

4.6 Bayesian Network with seven days data with learning parameter

Accuracy for all 21 nodes = 0.425

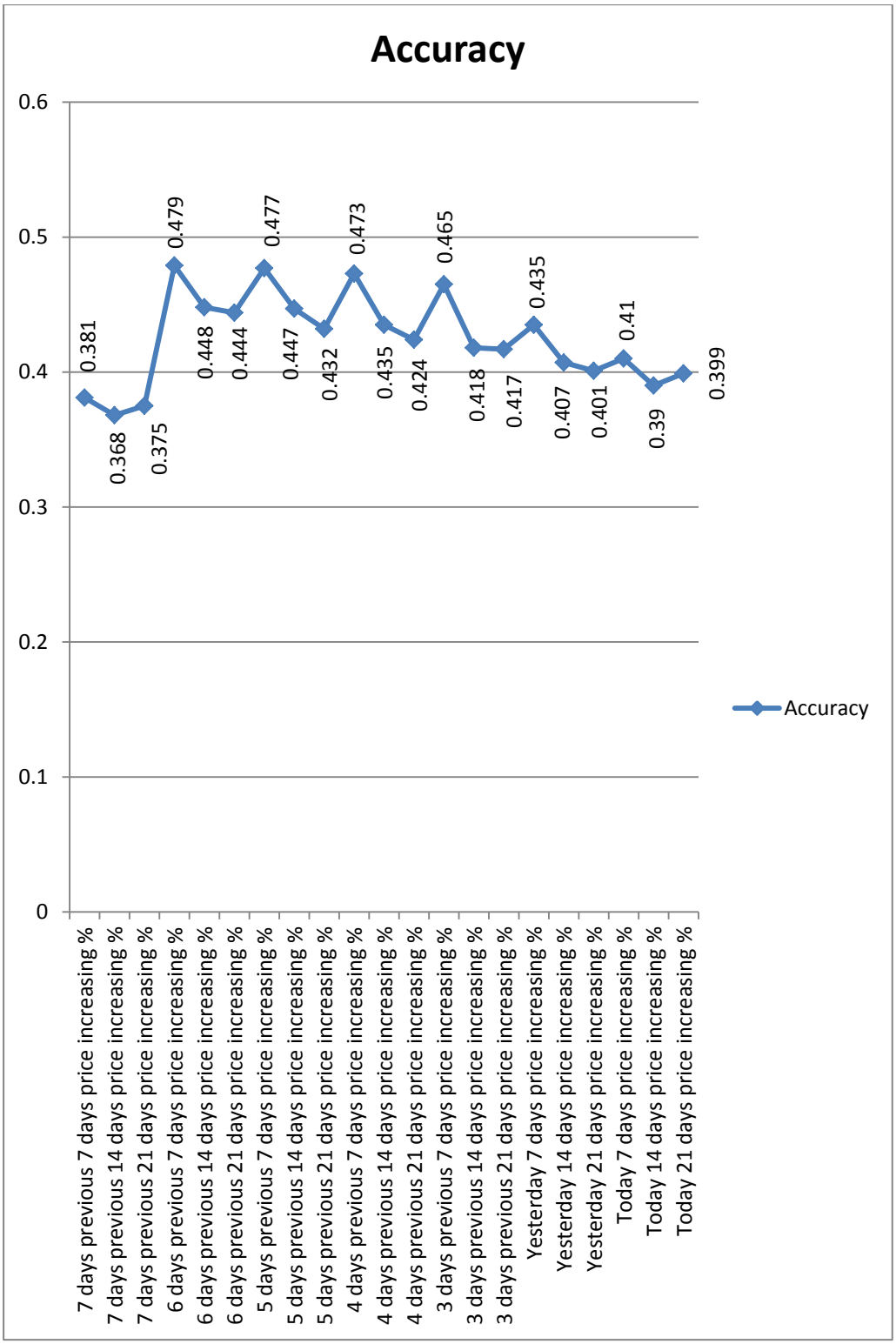


Figure 4-6

4.7 Bayesian Network with seven days data without learning parameter (without seven days previous seven days price increasing percentages)

Accuracy for all 20 nodes = 0.474

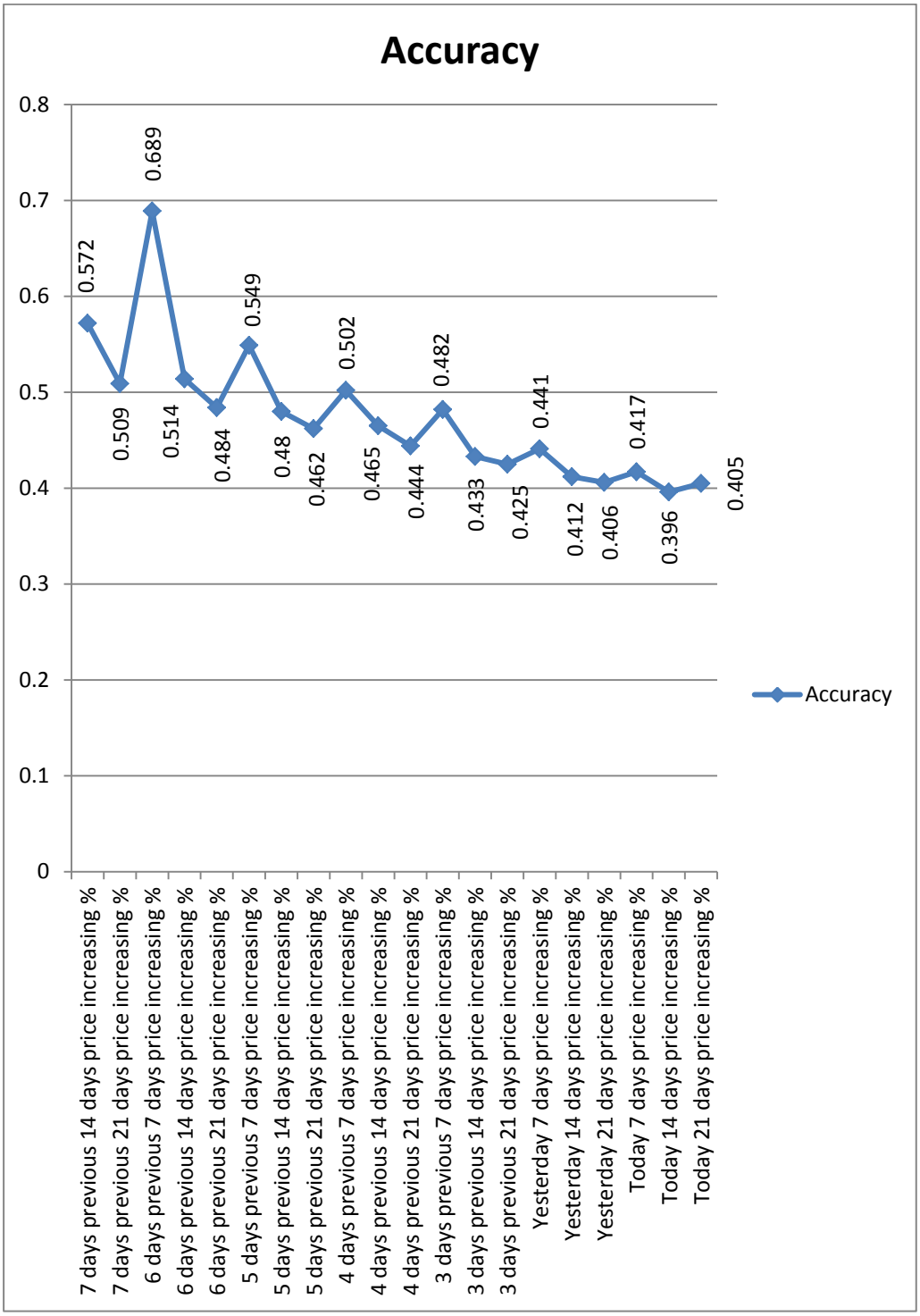

Accuracy
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0.572
0.509
0.689
0.514
0.484
0.549
0.48
0.462
0.502
0.465
0.444
0.482
0.433
0.425
0.441
0.412
0.406
0.417
0.396
0.405
7 days previous 14 days price increasing %
7 days previous 21 days price increasing %
6 days previous 7 days price increasing %
6 days previous 14 days price increasing %
6 days previous 21 days price increasing %
5 days previous 7 days price increasing %
5 days previous 14 days price increasing %
5 days previous 21 days price increasing %
4 days previous 7 days price increasing %
4 days previous 14 days price increasing %
4 days previous 21 days price increasing %
3 days previous 7 days price increasing %
3 days previous 14 days price increasing %
3 days previous 21 days price increasing %
Yesterday 7 days price increasing %
Yesterday 14 days price increasing %
Yesterday 21 days price increasing %
Today 7 days price increasing %
Today 14 days price increasing %
Today 21 days price increasing %
Accuracy

Figure 4-7

4.8 Bayesian Network with seven days data without learning parameter with 4 return results

Accuracy for all 4 nodes = 0.760

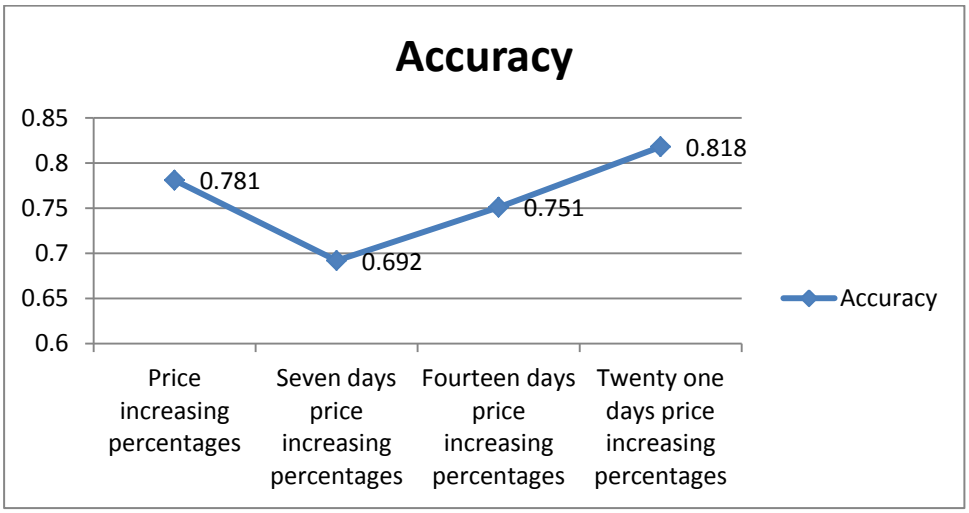


Figure 4-8

4.9 Bayesian Network with seven days data without learning parameter with 3 return results

Accuracy for all 3 nodes = 0.753

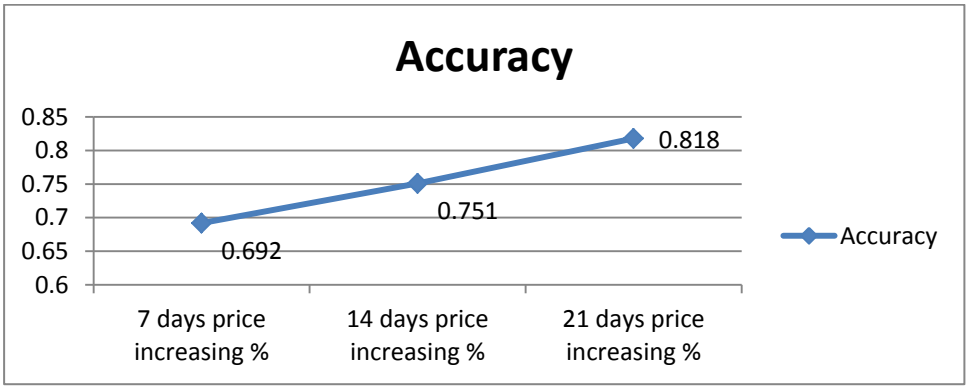


Figure 4-9

4.10 Results Explanation

Bayesian Network with learning parameter learns got empty value parameter. Got empty value parameter is retrieved from got empty value file. Data from got empty value file will be discretized. Discretize range needs to follow the number of empty data file discretize range and value. This is done so that the parameter that has been created can be used to test accuracy of the Bayesian Network or used to learn the Bayesian Network. (Data process step will be further elaborated in PHP section)

Bayesian Network that has been created by using only a day data and also with 7 days data uses historical stock data from year 1991 until 2012. All those data are processed before it can be used to develop and test the Bayesian Network. Most accurate Bayesian Network is selected after being tested with the got empty parameter. Most accurate Bayesian Network which means it has the highest accuracy rate among others. Above result for Bayesian Network is tested by the market data of year 2013 stock. The stock data of year 2013 is not used in development of Bayesian Network so it will be fair for testing Bayesian Network accuracy.

4.10.1 Comparison between Bayesian Network result with only a day stock data

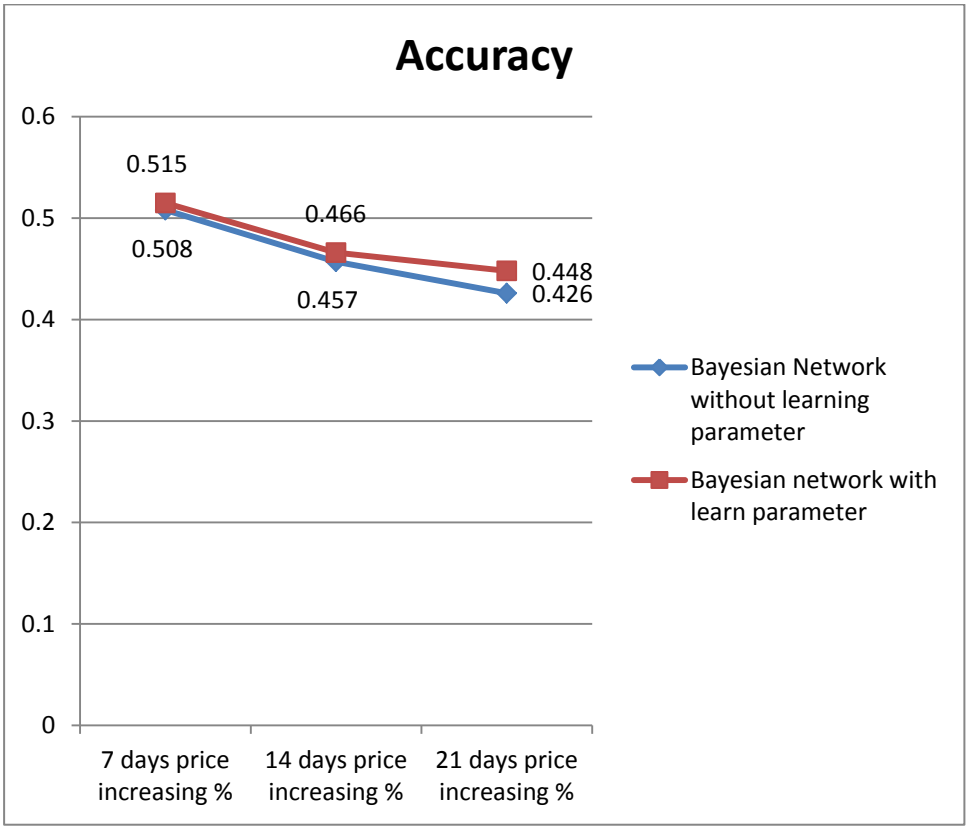


Figure 4-10

Average accuracy for most accurate Bayesian Network without learning parameter is 46.4% and Bayesian Network with learning parameter is 47.6264%. Based on the experiments results, Bayesian network with learn parameter will increase the accuracy of the Bayesian Network.

4.10.2 Comparison between Bayesian Network result with seven days stock data

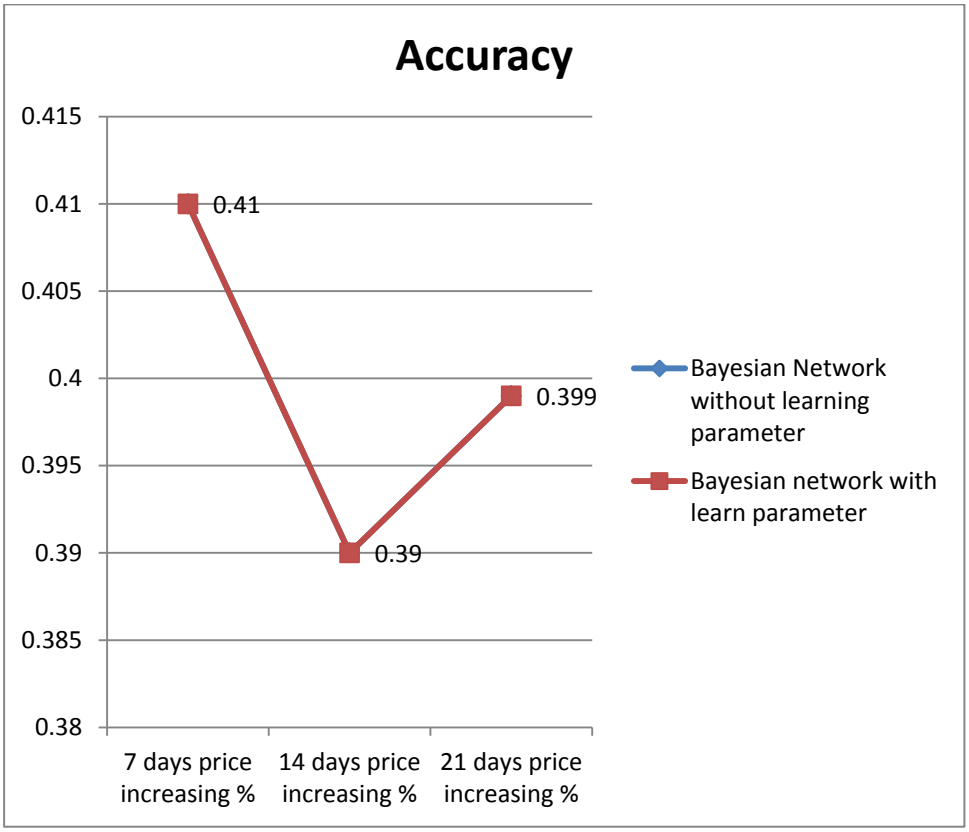


Figure 4-11

Accuracy for the most accurate Bayesian Network is generated by using seven days of stock data without learning parameter is 42.4998% and accuracy for the Bayesian Network generated by using seven days of stock data with learning parameter is 42.4993%. Based on the experiments results, Bayesian Network generated by using seven days of stock data without learning parameter will produce higher accuracy compared to the ones with learning parameter. Based on experiment result, it proved that not necessarily that the Bayesian Network with learning parameter will generate higher accuracy

compared with the original does not have learning parameter. Sometimes the accuracy might drop after learning the parameter.

Besides that, comparison between the accuracy for Bayesian Network that learn the parameter and Bayesian Network that did not learn the parameter, I compared the accuracy for both highest accurate Bayesian Network with only a day stock data and Bayesian Network generate by using seven days of stock data. Results for the Bayesian Network accuracy with only a day stock data is 47.6264% and Bayesian Network results with seven days of stock data is 42.4998%. The 47.6264% accuracy and 42.4998% accuracy are the overall accuracy from all the return result.

4.10.3 Comparison between Bayesian Network result with only a day stock data and Bayesian Network result with seven days of stock data

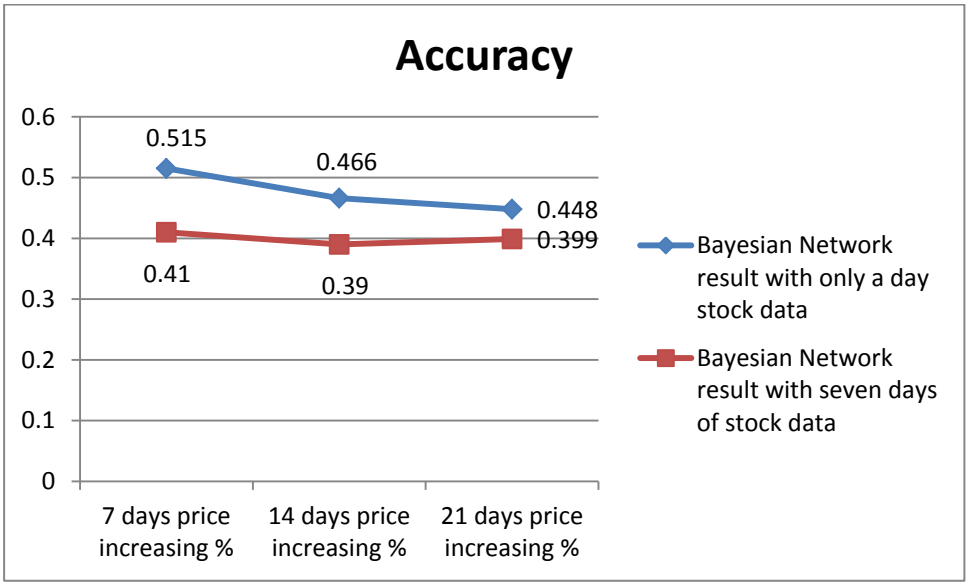


Figure 4-12

In order to make sure the comparison is fair, the accuracy of Bayesian Network comparison with seven days price increasing percentages, fourteen days price increasing percentages and twenty one days increase price percentages is selected because these three results are the main results that will be generated to the users for this AI system. Based on the experiment Bayesian Network result with only a day stock data has higher accuracy compared to Bayesian Network result with seven days of stock data.

4.10.4 Comparison between Bayesian Network with only a day stock data result for 3 return results and 4 return results

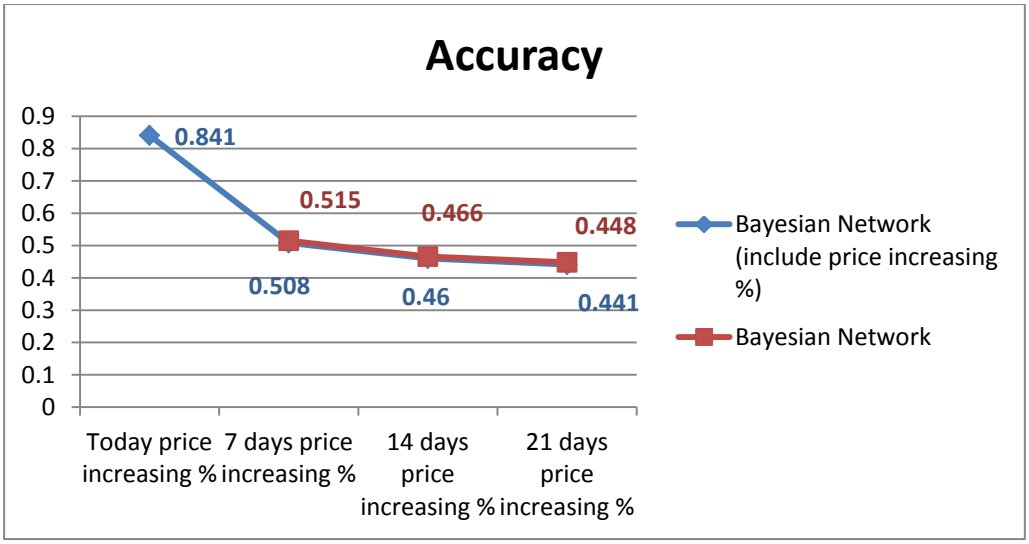


Figure 4-13

Figure 4-13 is another experiment that was conducted with the Bayesian Network. Same Bayesian Network which is set to return different output will also affect the accuracy. The table above shows comparison between Bayesian Network result with three return results and four return results. Bayesian Network which is developed by using a day data and today price increasing

percentages as a return result will have lower accuracy compared with that Bayesian Network that does not include today price increasing percentages. Based on the experiment, it means that the input data will affect the accuracy of the Bayesian Network. More data which is pumped into the Bayesian Network will create a more accurate Bayesian Network.

4.10.5 Comparison between Bayesian Network with seven days of stock data result for 3 return results and 21 return results

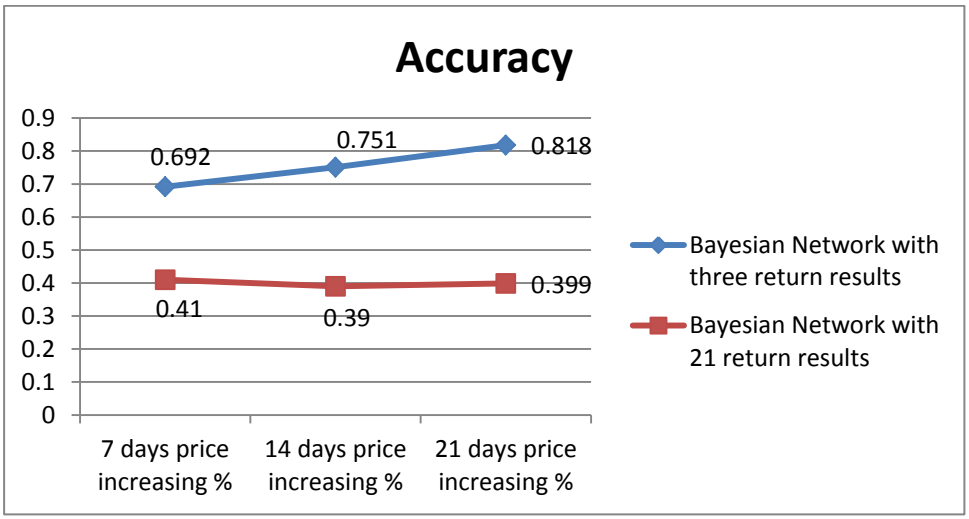


Figure 4-14

4.10.6 Comparison between Bayesian Network with seven days of stock data result for 20 return results and 21 return results

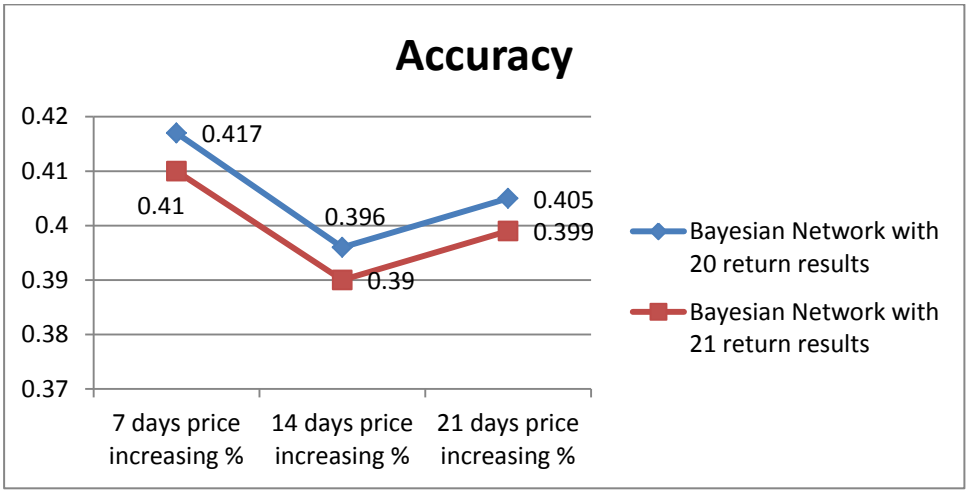


Figure 4-15

4.10.7 Comparison between Bayesian Network with seven days of stock data result for 3 return results and 4 return results

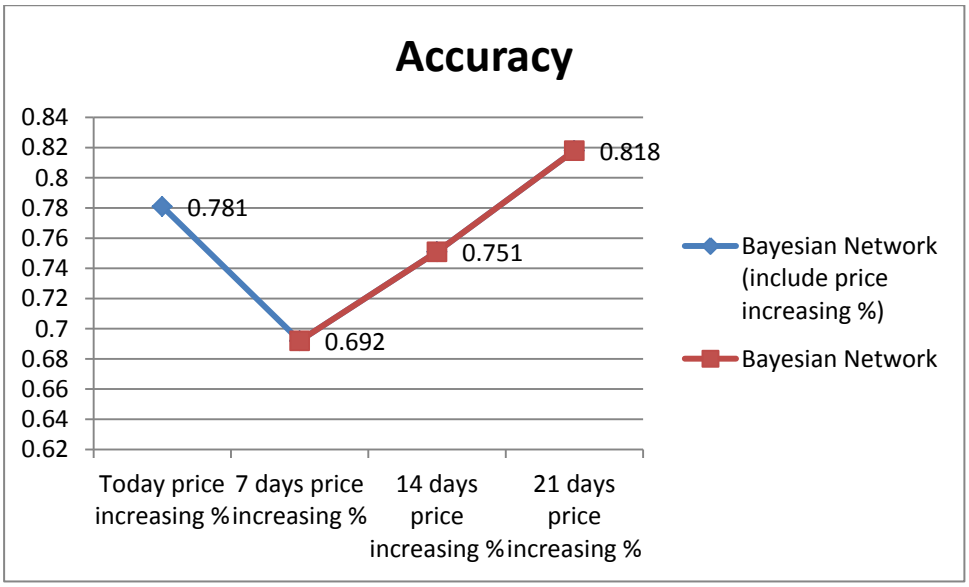


Figure 4-16

Figure 4-14, Figure 4-15, Figure 4-16 are few comparisons on Bayesian Network with seven days stock data. Based on the experiment on the Bayesian Network with only a day stock data, the input data will affect the accuracy of Bayesian Network.

Bayesian Network with seven days stock data with four output accuracy is actually the same with the Bayesian Network with only three outputs. This is because today price increasing might not be important for Bayesian Network with seven days stock data, that's why it makes no different. Comparison between Bayesian Network with seven days of stock data result for twenty return results and twenty one return results shows that the accuracy is different as Bayesian Network with twenty day return result has higher accuracy compared with Bayesian Network that with twenty one return results.

Based on the experiment, input data will affect the accuracy of Bayesian Network, but it also depends on the output data which is important or not for the Bayesian Network. Data that is used to develop Bayesian Network must be selected carefully because every single data will affect the accuracy of Bayesian Network.

4.10.7.1 Result time frame

Days	**Result**
Today	Seven day previous seven days price increasing percentages
2	Six day previous seven days price increasing percentages

3	Five day previous seven days price increasing percentages
4	Four day previous seven days price increasing percentages
5	Three day previous seven days price increasing percentages
6	Yesterday seven days price increasing percentages
7	Today seven days price increasing percentages
8	Seven day previous fourteen days price increasing percentages
9	Six day previous fourteen days price increasing percentages
10	Five day previous fourteen days price increasing percentages
11	Four day previous fourteen days price increasing percentages
12	Three day previous fourteen days price increasing percentages
13	Yesterday fourteen days price increasing percentages
14	Today fourteen days price increasing percentages
15	Seven day previous twenty one days price increasing percentages
16	Six day previous twenty one days price increasing percentages
17	Five day previous twenty one days price increasing percentages
18	Four day previous twenty one days price increasing percentages
19	Three day previous twenty one days price increasing percentages
20	Yesterday twenty one days price increasing percentages
21	Today twenty one days price increasing percentages

Table 4.1

Bayesian Network with seven days of stock data is for testing purpose. If the system run in real time, it will not be workable for Bayesian Network with seven days of stock data with only price increasing percentages, seven days price increasing percentages, fourteen days price increasing percentages and twenty one days price increasing percentages, because seven day previous

fourteen days price increasing percentages is an unknown data. Only seven day previous seven days price increasing percentages is equal to today price increasing percentages. Thus, it can be included in Bayesian Network with seven days of stock data as an input data.

4.10.8 Comparison between Bayesian Network result with background knowledge and without background knowledge

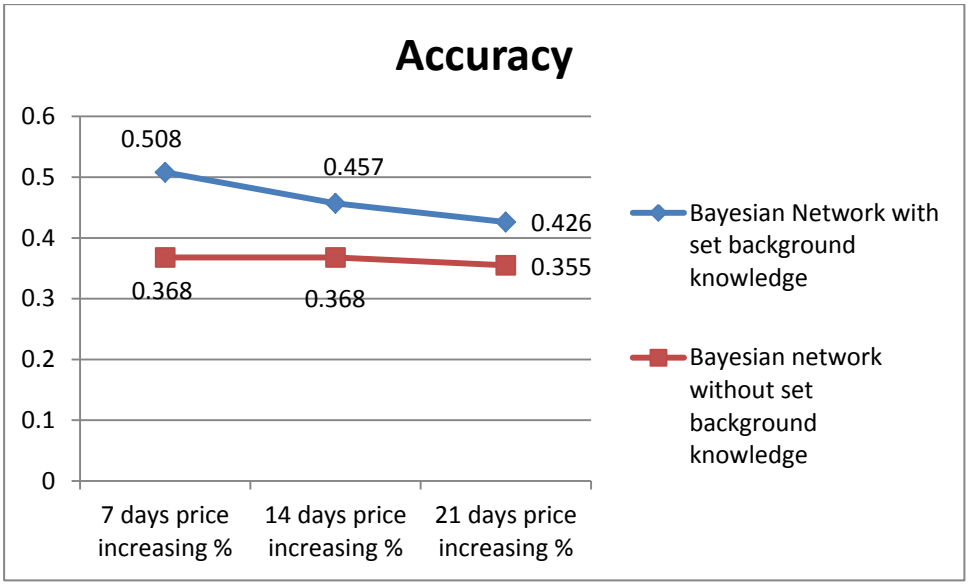


Figure 4-17

Besides the data that was used to develop Bayesian Network, background knowledge that have been set when developing Bayesian Network is also important in order to create higher accuracy for Bayesian Network. There are 71 Bayesian Network that have been created with different Background knowledge. The selected Bayesian Network has the highest accuracy results among others Bayesian Network with different Background knowledge. Bayesian Network that has been developed without Background knowledge

will return lower accuracy rate compared with those with has been set with Background knowledge.

4.11 Overall result comparison

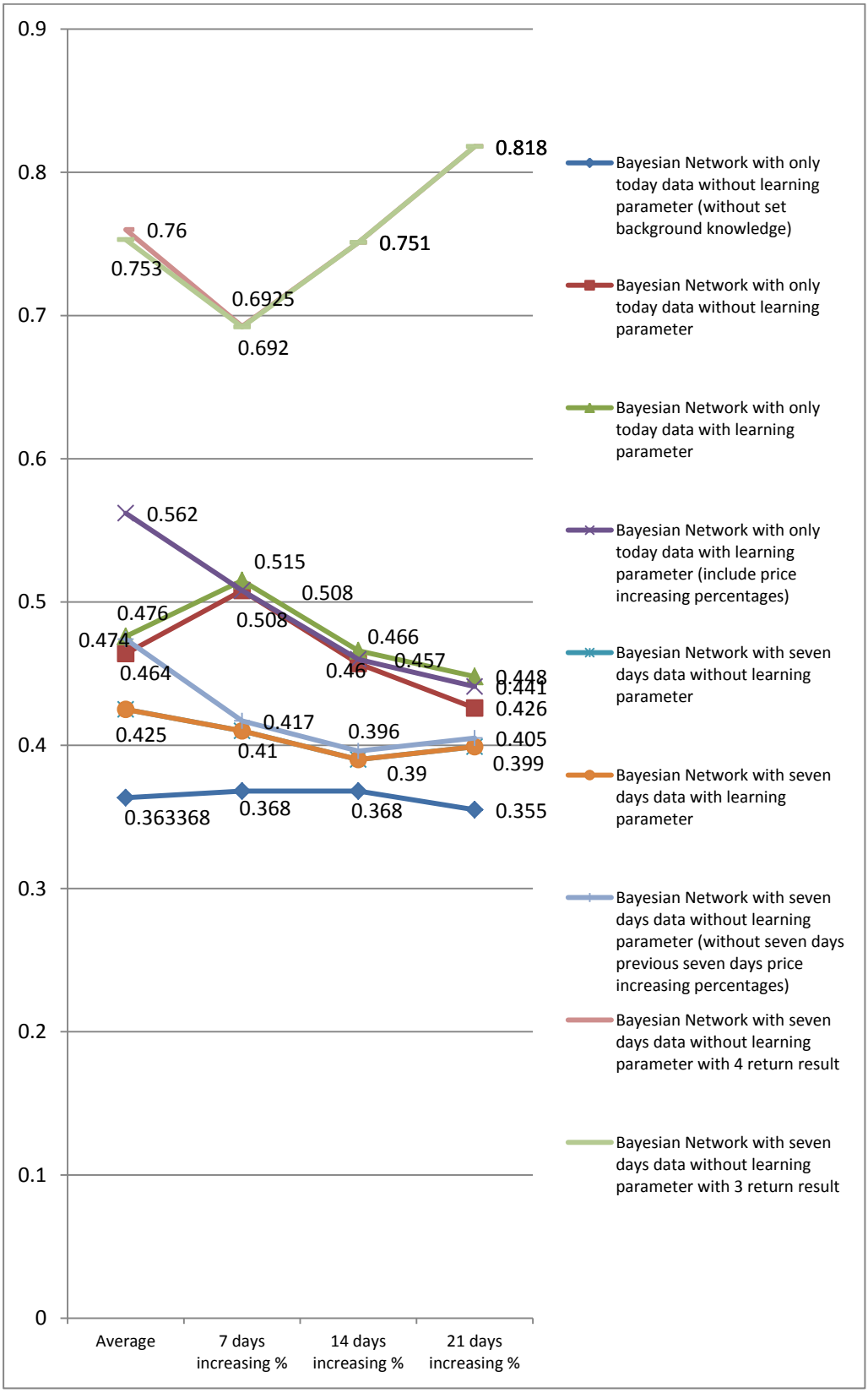


Figure 4-18

Based on the overall result, exclude the Bayesian Network with seven days data without learn parameter with 3 results and 4 four results because some of the input data are invalid. I have explain it on top about the input data for Bayesian Network with seven days data, the highest average accuracy is the Bayesian Network with only today data with learning parameter. Bayesian Network with only today data with learning parameter 4 return result have higher average accuracy is because it includes today price increasing percentages.

5.0 COMPARISON

There are some Machine Learning-based decision supports tools for portfolio risk analysis that already exist outside the market. Based on the research, I have found out that Dhaka stock market timing decisions by Hybrid Machine Learning technique (Shipra Banik, 2012), using Machine Learning in the development of the support tools for portfolio risk analysis. Dhaka stock market timing decisions by Hybrid Machine Learning technique used neural network as their core network but for mine, I used Machine Learning-based decision support tools for portfolio risk analysis. This is one of the differences between two systems.

Dhaka stock market timing decisions tools used neural network. They do not know how the results have been generated. Plus, same data set that has been pump into the system might provide two different results. Investors will be afraid if every time they execute the system, the return result is different. It will be a big problem when it comes to convincing investors to use their system. My decision support tools for portfolio risk analysis is different because by using Bayesian Network, investors will know the reason of the outcome, why the result produced is in that manner and which nodes are affecting it.

Besides that, based on "… the hybrid model can be recommended to predict the daily Dhaka stock movements that would guide investors, buyers, sellers and others when to buy, sell or hold a share" (Shipra Banik, 2012) which means that the results that was provided by Dhaka stock market timing decisions tools is daily result. It will provide results for tomorrow. Daily results

sometime are useless because investors need time to sell out those stocks in hand or buying stocks which the plan to invest in. A day period of time is too rushed for them to perform a response. If everyone sells at the same time, it will make the stock price drop faster than normal speed. My decision support tools for portfolio risk analysis provide results after seven days, fourteen days and twenty one days so that investors have enough time to make decision. Even if they want to sell their stocks or purchase stocks, they will at least have some time to collect money to purchase or to find new buyers for their stocks in hand.

5.1 Confusion matrix for the ANN model

	Predicted		
Actual	**Fall (-1)**	**Rise(+1)**	**Accuracy (%)**
Fall (-1)	290	108	0.729
Rise(+1)	97	453	0.824
Accuracy (%)	0.749	0.807	0.784

5.2 Confusion matrix for the ANN_RS model

	Predicted		
Actual	**Fall (-1)**	**Rise(+1)**	**Accuracy (%)**
Fall (-1)	401	17	0.960
Rise(+1)	3	527	0.994
Accuracy (%)	0.993	0.969	0.979

Confusion matrix for ANN model and ANN_RS model from (Shipra Banik, 2012)

5.3 Confusion matrix for my Bayesian Network

5.3.1 Seven days price increasing percentages confusion matrix for the most accurate Bayesian Network (with set background knowledge)

	Result < -5.556	**Result >= -5.556 and result < -0.746**	**Result >= -0.746 and result < 4.934**	**Result >= 4.934**
Result < -5.556	123386	60142	329298	36287
Result >= -5.556 and result < -0.746	51455	75638	593497	38992
Result >= -0.746 and result < 4.934	49433	52542	1035911	98110
Result >= 4.934	44125	28101	410270	159262

5.3.2 Fourteen days price increasing percentages confusion matrix for the most accurate Bayesian Network (with set background knowledge)

	Result < -7.813	**Result >= -7.813 and result < -1.049**	**Result >= -1.049 and result < 6.667**	**Result >= 6.667**
Result < -7.813	107913	54492	367721	28080
Result >= -7.813 and result < -1.049	59818	72829	627180	25624
Result >= -1.049 and result < 6.667	51890	56096	1007063	51387
Result >= 6.667	48778	33006	472718	91854

5.3.3 Twenty one days price increasing percentages confusion matrix for the most accurate Bayesian Network (with set background knowledge)

	Result < -9.382	**Result >= -9.382and result < -**	**Result >= -1.376 and result <**	**Result >= 7.870**

		1.376	**7.870**	
Result < -9. 382	119628	59934	362263	40172
Result >= -9. 382 and result < -1. 376	75035	83932	596263	39588
Result >= -1. 376 and result < 7. 870	64320	69126	916678	64839
Result >= 7. 870	63194	46549	453533	101395

Based on the confusion matrix above for both systems, Dhaka stock market timing decisions tools provide higher accuracy than my decision support tools for portfolio risk analysis. But as I mentioned earlier on although the accuracy is higher for daily prediction, it is useless for those investors because a day period is hard for them to make decision, to sell and purchase stocks. Here I state that my decision support tool for portfolio risk analysis is better than Dhaka stock market timing decisions tools. Different developers will have their different styles in developing the system. Their network that has been used in the systems might also be different, so the accuracy of the results might differ compared to results I have obtain from my system.

6.0 CONCLUSION AND FUTURE IMPROVEMENT

As a conclusion, the Machine Learning-based decision support tools for portfolio risk analysis system can help fund manager or investors in selecting stock, sell and purchase or hold the stocks. The results provided by the Machine Learning-based decision support tools is not a hundred per cent accurate or half of the full accuracy but then it can be a great guide to help fund managers or investors in selecting stocks. After this research, people might realize the advantages of Machine Learning-based decision support tools in helping them to select stock.

Machine Learning-based decision support tools for portfolio risk analysis system can be improved in its accuracy. Due to time concern, current Machine Learning-based decision support tools for portfolio risk analysis system is using only limited historical data that can be downloaded from internet.

In the future, developers can collect daily on-going stock data such as the accuracy of Bayesian Network that uses on-going stock data might be higher than Bayesian Network that is created by end of day data. Based on research and the books that I have study, daily stock data is important to predict future stock's price. "In a few cases, you should sell if a stock hits its upper channel line. (Channel lines are drawn to connect the lows and connect the highs on a stock's price chart.) Stocks surging above their upper channel lines should normally be sold." (O'Neil, 1995, p. 106), on-going stock data is required in order to check is that the stocks surging above their upper channel lines or not.

Daily on-going data is used as the base to develop the Bayesian network for a higher probability of higher accuracy for Machine Learning-based decision support tools for portfolio risk analysis system. This is because the changing of the data and also the stock data will be more and more specific. The more information that you feed in to the Bayesian network to learn, will help it provide more accurate Bayesian network. Just like the above Bayesian network that I have shown, for those Bayesian network that have been set with the background knowledge, it will provide higher accuracy results compared with those Bayesian network that does not provide background knowledge.

That's another part can be improved so that the results that are provided by Machine Learning-based decision support tools for portfolio risk analysis system can be more accurate is the calculation part. Different people have different style in calculating the stock price. It is just like different fund managers have their own style to select the stocks for their clients. The method that they use will also be different among each other. Some of them might be using Monte Carlo or Delta-Normal methods. Same goes to developing the Bayesian Network for Machine Learning-based decision support tools for portfolio risk analysis system.

In the future, formulas that are applied in historical stock data can be change using advance formula to process the historical data. Current Machine Learning-based decision support tools for portfolio risk analysis system are just using some simple mathematic formula to process all the historical data. Thus, advance mathematic formula that are used in future might also increase the

accuracy of the Machine Learning-based decision support tools for portfolio risk analysis system results.

New functions should be include in Machine Learning-based decision support tools for portfolio risk analysis system in future in order to meet investors needs, so that this system can fully support investors. In order to attract investors to use this system, it should be made to fulfil the investors needed, so that they will use it.

REFERENCES

Capterra, Inc. (2012). *Capterra the smart way to find software*. Retrieved 3 12, 2012, from http://www.capterra.com/compare-financial-risk-management-software/CreditPoint-Software-vs-DealerCenter-vs-F3-vs-IMS/24348-79449-114490-120371?parsed=f

Gavin Cassar, J. G. (January 7, 2011). How do Hedge Funds Manage Portfolio Risk? *How do Hedge Funds Manage Portfolio Risk?*, 9.

Investopedia Staff. (2009, 02 26). *Frequently Asked Question*. Retrieved 02 20, 2014, from Investopedia: http://www.investopedia.com/ask/answers/05/saleafterhours.asp

Jordan, M. I. (1998). *Learning in Graphical Models* (First MIT Press edition ed.). United States of America: Boston: Kluwer Academic Publishers.

Martin Anthony, Peter L. Bartlett. (2009). *Neural Network Learning: Theoretical Foundations.* United States of America: Cambridge University Press.

O'Neil, W. J. (1995). *How To Make Money In Stocks: A Winning System in Good Times or Bad, 2nd Edition.* McGraw-Hill.

S.P.Toulson, D. (1996). Use Of Neural Network Ensembles for Portfolio Selection and Risk Management. 6.

Shipra Banik, A. K. (2012). Dhaka Stock Market Timing Decisions by Hybrid Machine Learning Technique. *Computer and Information Technology (ICCIT), 2012 15th International Conference on Date of Conference: 22-24 Dec. 2012*, 384 - 389.

Tseng, C.-C. (2003). Comparing Artificial Intelligence Systems for Stock Portfolio Selection.

Wan, E. (2002). Neural network classification: a Bayesian interpretation.

APPENDIX A

CANSLIM method

Technique that I plan to use in the system was abstract from (O'Neil, 1995, p. 78) :

C = Current quarterly earnings per share must be increased at least 20%.

A = Last five years, the annual earnings per share must show growth.

N = N stand for New. Businesses that have been come out new products, new services, new management, or any new change that will help the company grow, and those stock that already create their new high in market.

S = Must have some outstanding shares, and when the stock's price increase, check for its stock volume.

L = Leaders. Market leaders are those companies that have the largest market shares and have high profit margin.

I = Institutional sponsorship. Purchase stocks that have better than recent average performance and have institutional sponsorship because when you want to sell out your stocks, at least there will be someone buying it.

M = Market. Understand the market.

APPENDIX B

Calculation for the historical data:

Opening price percentages:

Opening price percentages = ((opening price – yesterday price) / opening price) * 100

- Today opening price minus yesterday price will know the opening price either increase or decrease. After that, divide by today opening price and multiply 100 to get opening price percentages.

Highest price percentages:

Highest price percentages = ((highest price – opening price) / opening price) * 100

- Today highest price minus opening price will get today highest stock price. After that, divide today opening price and multiply 100 to get the highest price percentages.

Lowest price percentages:

Lowest price percentages = ((lowest price – opening price) / opening price) * 100

- Today lowest price minus opening price will get today lowest stock price. After that, divide today opening price and multiply 100 to get the lowest price percentages.

Stock volume percentages:

Stock volume percentages = ((stock volume – yesterday stock volume) / yesterday stock volume) * 100

- Today stock volume minus yesterday stock volume will get the stock volume either increases or decreases. After that, divide by yesterday stock volume and multiply 100 to get stock volume percentages.

Three days average opening price percentages:

Three days average opening price percentages = ((((previous 2 days opening price + yesterday opening price + today opening price) / 3) - today opening price) / today opening price) * 100

- Sum up the previous 2 days opening price and today opening price then divide by three in order to get the three days average opening price. After that, the three days opening price average will be divided by today opening price and then multiply by 100 to get Three days average opening price percentages

Seven days average opening price percentages:

Seven days average opening price percentages = ((((previous 6 days opening price + previous 5 days opening price + previous 4 days opening price + previous 3 days opening price + previous 2 days opening price + yesterday opening price + today opening price) / 7) - today opening price) / today opening price) * 100

- Sum up previous 6 days opening price and today opening price then divide by seven in order to get the seven days average opening price. After that, the seven days opening price average will be divided by

today opening price and then multiply by 100 to get Seven days average opening price percentages

Fourteen days average opening price percentages:

Fourteen days average opening price percentages = ((((previous 13 days opening price + previous 12 days opening price + previous 11 days opening price + previous 10 days opening price + previous 9 days opening price + previous 8 days opening price + previous 7 days opening price + previous 6 days opening price + previous 5 days opening price + previous 4 days opening price + previous 3 days opening price + previous 2 days opening price + yesterday opening price + today opening price) / 14) - today opening price) / today opening price) * 100

- Sum up previous 13 days opening price and today opening price then divide by fourteen in order to get the fourteen days average opening price average. After that the fourteen days opening price average will be divided by today opening price and then multiply by 100 to get fourteen days average opening price percentages

Highest price minus lowest price percentages:

Highest price minus lowest price percentages = ((highest price – lowest price) / opening price) * 100

- Today highest stock price minus today lowest stock price will be the difference between highest and lowest. After that, use the difference value divide by today opening price and multiply by 100 to get highest price minus lowest price percentages.

Price increasing during off market percentages:

Price increasing during off market percentages = ((yesterday closing price – opening price) / opening price) * 100

- Yesterday opening price minus with today opening price. After that, divide by today opening price multiply by 100 to check the percentages of the price increasing during off market.

Price increasing percentages:

Price increasing percentages = ((closing price – opening price) / opening price) * 100

- Today closing price minus today opening price then will know the price is decreasing or increasing. After that, divide by today opening price and the multiply by 100 will get today price increasing percentages.

Seven days price increasing percentages:

Seven days price increasing percentages = ((closing price after 7 days - opening price) / opening price) * 100

- Closing price after 7 days minus with today opening price then will know the price is decreasing or increasing after 7 days. After that, divide by today opening price and then multiply by 100 will get price increasing percentages after 7 days.
- **Note:** seven days price increasing percentages data will be generated when developing the Bayesian Network. Actually it was one of the

results that are generated by Machine Learning-based decision support tools.

Fourteen days price increasing percentages:

Fourteen days price increasing percentages = ((closing price after 14 days - opening price) / opening price) * 100

- Closing price after 14 days minus with today opening price then will know the price is decreasing or increasing after 14 days. After that, divide by today opening price and the multiply by 100 will get price increasing percentages after 14 days.
- **Note:** fourteen days price increasing percentages data will be generated when developing the Bayesian Network. Actually it was one of the results that are generated by Machine Learning-based decision support tools.

Twenty one days price increasing percentages:

Twenty one days price increasing percentages = ((closing price after 21 days - opening price) / opening price) * 100

- Closing price after 21 days minus with today opening price then will know the price is decreasing or increasing after 21 days. After that, divide by today opening price and the multiply by 100 will get price increasing percentages after 21 days.
- **Note:** twenty one days price increasing percentages data will be generated when developing the Bayesian Network. Actually it was one

of the results that are generated by Machine Learning-based decision support tools.

Calculation for the historical data coding:

Opening price percentages:

If (Opening price percentages === “”) {

Opening price percentages for network = “”

} else if (Opening price percentages < -2.272727) {

Opening price percentages for network = 0

} else if (Opening price percentages >= -2.272727 AND Opening price percentages < 0.3496504) {

Opening price percentages for network = 1

} else if (Opening price percentages >= 0.3496504 AND Opening price percentages < 2.673797) {

Opening price percentages for network = 2

} else if (Opening price percentages >= 2.673797) {

Opening price percentages for network = 3

} else {

Opening price percentages for network = “”

}

Highest price percentages:

If (Highest price percentages === “”) {

Highest price percentages for network = “”

} else if (Highest price percentages < 0.8832188) {

Highest price percentages for network = 0

} else if (Highest price percentages >= 0.8832188 AND Highest price percentages < 1.745455) {

Highest price percentages for network = 1

} else if (Highest price percentages >= 1.745455 AND Highest price percentages < 3.4482760) {

Highest price percentages for network = 2

} else if (Highest price percentages >= 3.448276) {

Highest price percentages for network = 3

}

Lowest price percentages:

If (Lowest price percentages === “”) {

Lowest price percentages for network = “”

} else if (Lowest price percentages < -3.225806) {

Lowest price percentages for network = 0

} else if (Lowest price percentages >= -3.225806 AND Lowest price percentages < -1.7241380) {

Lowest price percentages for network = 1

} else if (Lowest price percentages >= -1.724138 AND Lowest price percentages < -0.8849558) {

Lowest price percentages for network = 2

} else if (Lowest price percentages >= -0.8849558){

Lowest price percentages for network = 3

} else {

Lowest price percentages for network = “”

}

Stock volume percentages:

If (Stock volume percentages === “”) {

Stock volume percentages for network = “”

} else if (Stock volume percentages < -38.28125) {

Stock volume percentages for network = 0

} else if (Stock volume percentages >= -38.28125 AND Stock volume percentages < 7.920792) {

Stock volume percentages for network = 1

} else if (Stock volume percentages >= 7.920792 AND Stock volume percentages < 98.611110) {

Stock volume percentages for network = 2

} else if (Stock volume percentages >= 98.61111) {

Stock volume percentages for network = 3

} else {

Stock volume percentages for network = “”

}

Three days opening price average percentages:

If (Three days opening price average percentages === “”) {

Three days opening price average percentages for network = “”

} else if (Three days opening price average percentages < -1.851852) {

Three days opening price average percentages for network = 0

} else if (Three days opening price average percentages >= -1.851852 AND Three days opening price average percentages < -1.586033e-014) {

Three days opening price average percentages for network = 1

} else if (Three days opening price average percentages >= -1.586033e-014 AND Three days opening price average percentages < 1.612903) {

Three days opening price average percentages for network = 2

} else if (Three days opening price average percentages >= 1.612903) {

Three days opening price average percentages for network = 3

} else {

Three days opening price average percentages for network = “”

}

Seven days opening price average percentages:

If (Seven days opening price average percentages === “”) {

Seven days opening price average percentages for network = “”

} else if (Seven days opening price average percentages < -3.246753) {

Seven days opening price average percentages for network = 0

} else if (Seven days opening price average percentages >= -3.246753 AND Seven days opening price average percentages < -0.06523157) {

Seven days opening price average percentages for network = 1

} else if (Seven days opening price average percentages >= -0.06523157 AND Seven days opening price average percentages < 2.710333) {

Seven days opening price average percentages for network = 2

} else if (Seven days opening price average percentages >= 2.710333) {

Seven days opening price average percentages for network = 3

} else {

Seven days opening price average percentages for network = “”

}

Fourteen days opening price average percentages:

If (Fourteen days opening price average percentages === “”) {

Fourteen days opening price average percentages for network = “”

} else if (Fourteen days opening price average percentages < -5.037594) {

Fourteen days opening price average percentages for network = 0

} else if (Fourteen days opening price average percentages >= -5.037594 AND Fourteen days opening price average percentages < -0.3139717) {

Fourteen days opening price average percentages for network = 1

} else if (Fourteen days opening price average percentages >= -0.3139717 AND Fourteen days opening price average percentages < 3.781512) {

Fourteen days opening price average percentages for network = 2

} else if (Fourteen days opening price average percentages >= 3.781512) {

Fourteen days opening price average percentages for network = 3

} else {

Fourteen days opening price average percentages for network = “”

}

Highest price minus lowest price percentages:

If (Highest price minus lowest price percentages === “”) {

Highest price minus lowest price percentages for network = “”

} else if (Highest price minus lowest price percentages < 2.564103) {

Highest price minus lowest price percentages for network = 0

} else if (Highest price minus lowest price percentages >= 2.564103 AND Highest price minus lowest price percentages < 4.148279) {

Highest price minus lowest price percentages for network = 1

} else if (Highest price minus lowest price percentages >= 4.148279 AND Highest price minus lowest price percentages < 6.666667) {

Highest price minus lowest price percentages for network = 2

} else if (Highest price minus lowest price percentages >= 6.666667) {

Highest price minus lowest price percentages for network = 3

} else {

Highest price minus lowest price percentages for network = “”

}

Price increasing during off market percentages:

If (Price increasing during off market percentages === “”) {

Price increasing during off market percentages for network = “”

} else if (Price increasing during off market percentages < -1.31397) {

Price increasing during off market percentages for network = 0

} else if (Price increasing during off market percentages >= -1.31397 AND Price increasing during off market percentages < -0.4905396) {

Price increasing during off market percentages for network = 1

} else if (Price increasing during off market percentages >= -0.4905396 AND Price increasing during off market percentages < 1.25) {

Price increasing during off market percentages for network = 2

} else if (Price increasing during off market percentages >= 1.25) {

Price increasing during off market percentages for network = 3

} else {

Price increasing during off market percentages for network = “”

}

Price increasing percentages:

If (Price increasing percentages === “”) {

Price increasing percentages for network = “”

} else if (Price increasing percentages < -1.6) {

Price increasing percentages for network = 0

} else if (Price increasing percentages >= -1.6 AND Price increasing percentages < 0.4016064) {

Price increasing percentages for network = 1

} else if (Price increasing percentages >= 0.4016064 AND Price increasing percentages < 1.851852) {

Price increasing percentages for network = 2

} else if (Price increasing percentages >= 1.851852) {

Price increasing percentages for network = 3

} else {

Price increasing percentages for network = “”

}

Result that generate by Bayesian Network

Seven days price increasing percentages:

First result for seven days price increasing percentages: First result for seven days price increasing percentages * 100

Second result for seven days price increasing percentages: Second result for seven days price increasing percentages * 100

Third result for seven days price increasing percentages: Third result for seven days price increasing percentages * 100

Fourth result for seven days price increasing percentages: Fourth result for seven days price increasing percentages * 100

Fourteen days price increasing percentages:

First result for fourteen days price increasing percentages: First result for fourteen days price increasing percentages * 100

Second result for fourteen days price increasing percentages: Second result for fourteen days price increasing percentages * 100

Third result for fourteen days price increasing percentages: Third result for fourteen days price increasing percentages * 100

Fourth result for fourteen days price increasing percentages: Fourth result for fourteen days price increasing percentages * 100

Twenty one days price increasing percentages:

First result for twenty one days price increasing percentages: First result for twenty one days price increasing percentages * 100

Second result for twenty one days price increasing percentages: Second result for twenty one days price increasing percentages * 100

Third result for twenty one days price increasing percentages: Third result for twenty one days price increasing percentages * 100

Fourth result for twenty one days price increasing percentages: Fourth result twenty one fourteen days price increasing percentages * 100

APPENDIX C

Result

Most accurate Bayesian Network with only today data without learning parameter

Accuracy for all 3 nodes = 0.463728 (509465/1098630)

Seven days price increasing percentages = 0.508397 (186180/366210)

- Result < -5.555555 = 0.310172 (19141/61711)
- Result >= -5.555555 and result < -0.7462686 = 0.169831 (10402/61249)
- Result >= -0.7462686 and result < 4.93421 = 0.812927 (131721/162033)
- Result >= 4.93421 = 0.306783 (24916/81217)

Fourteen days price increasing percentages = 0.456735 (167261/366210)

- Result < -7.8125 = 0.265858 (16329/61420)
- Result >= -7.8125 and result < -1.048951 = 0.14781 (10240/69278)
- Result >= -1.048951 and result < 6.666667 = 0.790724 (119518/151150)
- Result >= 6.666667 = 0.25099 (21174/84362)

Twenty one days price increasing percentages = 0.426051 (156024/366210)

- Result < -9.38248 = 0.268308 (16623/61955)
- Result >= -9.38248 and result < -1.376147 = 0.170394 (11727/68823)

- Result >= -1.376147 and result < 7.869742 = 0.737848 (107762/146049)
- Result >= 7.869742 = 0.222772 (19912/89383)

Seven days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data without learning parameter (with set background knowledge)

	Result < -5.555555	Result >= -5.555555 and result < -0.7462686	Result >= -1.048951 and result < 4.93421	Result >= 4.93421
Result < -5.555555	19141	5590	31538	5442
Result >= -5.555555 and result < -0.7462686	6350	10402	40549	3948
Result >= -1.048951 and result < 4.93421	9512	8385	131721	12415
Result >= 4.93421	7395	4526	44380	24916

Fourteen days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data without learning parameter (with set background knowledge)

	Result < -7.8125	Result >= -7.8125 and result < -1.048951	Result >= -1.048951 and result < 6.666667	Result >= 6.666667
Result < -7.8125	16329	4864	34093	6134
Result >= -7.8125 and result < -1.048951	8695	10240	44744	5599
Result >= -1.048951 and result < 6.666667	9835	8527	119518	13270
Result >= 6.666667	8083	5272	49833	21174

Twenty one days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data without learning parameter (with set background knowledge)

	Result < -	Result >= -	Result >= -	Result >=

	9.38248	9.38248 and result < -1.376147	1.376147 and result < 7.869742	7.869742
Result < -9.38248	16623	10021	28741	6570
Result >= -9.38248 and result < -1.376147	10839	11727	40726	5531
Result >= -1.376147 and result < 7.869742	11564	15147	107762	11576
Result >= 7.869742	10384	11247	47840	19912

Most accurate Bayesian Network with only today data with learning parameter

Accuracy for all 3 nodes = 0.476264 (523238/1098630)

Seven days price increasing percentages = 0.514997 (188597/366210)

- Result < -5.555555 = 0.247071 (15247/61711)
- Result >= -5.555555 and result < -0.7462686 = 0.122892 (7527/61249)

- Result >= -0.7462686 and result < 4.93421 = 0.873686 (141566/162033)
- Result >= 4.93421 = 0.298669 (24257/81217)

Fourteen days price increasing percentages = 0.465864 (170604/366210)

- Result < -7.8125 = 0.149332 (9172/61420)
- Result >= -7.8125 and result < -1.048951 = 0.157813 (10933/69278)
- Result >= -1.048951 and result < 6.666667 = 0.890083 (134536/151150)
- Result >= 6.666667 = 0.18922 (15963/84362)

Twenty one days price increasing percentages = 0.447932 (164037/366210)

- Result < -9.38248 = 0.194512 (12051/61955)
- Result >= -9.38248 and result < -1.376147 = 0.129957 (8944/68823)
- Result >= -1.376147 and result < 7.869742 = 0.841129 (122846/146049)
- Result >= 7.869742 = 0.225949 (20196/89383)

Seven days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -5.555555	Result >= -5.555555 and result < -	Result >= -1.048951 and result <	Result >= 4.93421

		0.7462686	4.93421	
Result < -5.555555	15247	3924	37434	5106
Result >= -5.555555 and result < -0.7462686	4445	7527	45809	3468
Result >= -1.048951 and result < 4.93421	4119	4763	141566	11585
Result >= 4.93421	3899	2620	50441	24257

Fourteen days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -7.8125	Result >= -7.8125 and result < -1.048951	Result >= -1.048951 and result < 6.666667	Result >= 6.666667
Result < -7.8125	9172	6788	41172	4288
Result >= -	2388	10933	53278	2679

7.8125 and result < -1.048951				
Result >= -1.048951 and result < 6.666667	2007	7514	134536	7093
Result >= 6.666667	2496	5246	60657	15963

Twenty one days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -9.38248	Result >= -9.38248 and result < -1.376147	Result >= -1.376147 and result < 7.869742	Result >= 7.869742
Result < -9.38248	12051	4907	38533	6464
Result >= -9.38248 and result < -1.376147	5305	8944	49114	5460
Result >= -	4570	6998	122846	11635

1.376147 and result < 7.869742				
Result >= 7.869742	5783	4937	58467	20196

Most accurate Bayesian Network with only today data with learning parameter (include price increasing percentages)

Accuracy for all 4 nodes = 0.562472 (823932/1464840)

Price increasing percentages= 0.841086 (308014/366210)

- Result < -1.6= 0.823801 (38343/46544)
- Result >= -1.6 and result < 0.4016064= 0.875752 (197793/225855)
- Result >= 0.4016064 and result < 1.851852= 0.466077 (15223/32662)
- Result >= 1.851852 = 0.926507 (56655/61149)

Seven days price increasing percentages = 0.507788 (185957/366210

- Result < -5.555555 = 0.26966 (16641/61711)
- Result >= -5.555555 and result < -0.7462686 = 0.0888341 (5441/61249)
- Result >= -0.7462686 and result < 4.93421 = 0.851975 (138048/162033)
- Result >= 4.93421 = 0.318 (25827/81217)

Fourteen days price increasing percentages = 0.460067 (168481/366210)

- Result < -7.8125 = 0.22675 (13927/61420)

- Result >= -7.8125 and result < -1.048951 = 0.0813967 (5639/69278)
- Result >= -1.048951 and result < 6.666667 = 0.870658 (131600/151150)
- Result >= 6.666667 = 0.205246 (17315/84362)

Twenty one days price increasing percentages = 0.440949 (161480/366210)

- Result < -9.38248 = 0.207861 (12878/61955)
- Result >= -9.38248 and result < -1.376147 = 0.0973512 (6700/68823)
- Result >= -1.376147 and result < 7.869742 = 0.842019 (122976/146049)
- Result >= 7.869742 = 0.21174 (18926/89383)

Price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -1.6	Result >= -1.6 and result < 0.4016064	Result >= 0.4016064 and result < 1.851852	Result >= 1.851852
Result < -1.6	38343	6572	0	1629
Result >= -1.6 and result < 0.4016064	11197	197793	6272	10593
Result >= 0.4016064	542	11673	15223	5224

and result < 1.851852				
Result >= 1.851852	708	1571	2215	56655

Seven days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -5.555555	Result >= -5.555555 and result < -0.7462686	Result >= -1.048951 and result < 4.93421	Result >= 4.93421
Result < -5.555555	16641	3452	34644	6974
Result >= -5.555555 and result < -0.7462686	5163	5441	46691	3954
Result >= -1.048951 and result < 4.93421	5992	3946	138048	14047
Result >= 4.93421	5081	2192	48117	25827

Fourteen days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -7.8125	Result >= -7.8125 and result < -1.048951	Result >= -1.048951 and result < 6.666667	Result >= 6.666667
Result < -7.8125	13927	3087	38622	5784
Result >= -7.8125 and result < -1.048951	7033	5639	53321	3285
Result >= -1.048951 and result < 6.666667	6356	4206	131600	8988
Result >= 6.666667	5771	2747	58529	17315

Twenty one days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -9.38248	Result >= -9.38248 and result < -1.376147	Result >= -1.376147 and result < 7.869742	Result >= 7.869742
Result < -9.38248	12878	3562	38000	7515
Result >= -9.38248 and result < -1.376147	7368	6700	49779	4976
Result >= -1.376147 and result < 7.869742	6252	5643	122976	11178
Result >= 7.869742	6544	3915	59998	18926

Most accurate Bayesian Network with seven days data without learning parameter

Accuracy for all 21 nodes = 0.424998 (3141191/7391067)

Seven days previous seven days price increasing percentages = 0.380687 (133979/351940)

- Result < -5.536332 = 0.457495 (27059/59146)
- Result >= -5.536332 and result < -0.7246377 = 0.461606 (27244/59020)
- Result >= -0.7246377 and result < 4.991681 = 0.312903 (48550/155160)
- Result >= 4.991681 = 0.395935 (31126/78614)

Seven days previous fourteen days price increasing percentages = 0.36823 (129595/351940)

- Result < -7.777778 = 0.43089 (25376/58892)
- Result >= -7.777778 and result < -1.007557 = 0.379142 (25277/66669)
- Result >= -1.007557 and result < 6.773334 = 0.332937 (48158/144646)
- Result >= 6.773334 = 0.376641 (30784/81733)

Seven days previous twenty one days price increasing percentages = 0.375092 (132010/351940)

- Result < -9.340659 = 0.437922 (25978/59321)
- Result >= -9.340659 and result < -1.315789 = 0.349674 (23111/66093)
- Result >= -1.315789 and result < 8 = 0.365444 (51137/139931)
- Result >= 8 = 0.367042 (31784/86595)

Six days previous seven days price increasing percentages = 0.478855 (168543/351971)

- Result < -5.528613 = 0.452645 (26778/59159)
- Result >= -5.528613 and result < -0.7246377 = 0.322995 (19099/59131)
- Result >= -0.7246377 and result < 4.975124 = 0.603091 (93622/155237)
- Result >= 4.975124 = 0.370251 (29044/78444)

Six days previous fourteen days price increasing percentages = 0.447702 (157578/351971)

- Result < -7.777778 = 0.427708 (25127/58748)
- Result >= -7.777778 and result < -1.010101 = 0.27396 (18260/66652)
- Result >= -1.010101 and result < 6.75583 = 0.591455 (85603/144733)
- Result >= 6.75583 = 0.349324 (28588/81838)

Six days previous twenty one days price increasing percentages = 0.444338 (156394/351971)

- Result < -9.338521 = 0.419897 (24826/59124)
- Result >= -9.338521 and result < -1.324503 = 0.242756 (16011/65955)
- Result >= -1.324503 and result < 7.972379 = 0.607364 (85178/140242)
- Result >= 7.972379 = 0.350594 (30379/86650)

Five days previous seven days price increasing percentages = 0.47661 (167794/352057)

- Result < -5.527638 = 0.454229 (26899/59219)
- Result >= -5.527638 and result < -0.7246377 = 0.279243 (16546/59253)

- Result >= -0.7246377 and result < 4.964539 = 0.604684 (93896/155281)
- Result >= 4.964539 = 0.388907 (30453/78304)

Five days previous fourteen days price increasing percentages = 0.4473 (157475/352057)

- Result < -7.777778 = 0.42681 (25020/58621)
- Result >= -7.777778 and result < -1.01833 = 0.259226 (17252/66552)
- Result >= -1.01833 and result < 6.730769 = 0.58756 (85171/144957)
- Result >= 6.730769 = 0.36657 (30032/81927)

Five days previous twenty one days price increasing percentages = 0.432294 (152192/352057)

- Result < -9.343003 = 0.401985 (23689/58930)
- Result >= -9.343003 and result < -1.333333 = 0.22215 (14641/65906)
- Result >= -1.333333 and result < 7.936508 = 0.597378 (83894/140437)
- Result >= 7.936508 = 0.345317 (29968/86784)

Four days previous seven days price increasing percentages = 0.47277 (166448/352070)

- Result < -5.528846 = 0.431527 (25568/59250)
- Result >= -5.528846 and result < -0.7264802 = 0.322313 (19128/59346)
- Result >= -0.7264802 and result < 4.951857 = 0.583292 (90573/155279)

- Result >= 4.951857 = 0.398734 (31179/78195)

Four days previous fourteen days price increasing percentages = 0.434879 (153108/352070)

- Result < -7.777778 = 0.393429 (22990/58435)
- Result >= -7.777778 and result < -1.025641 = 0.177773 (11805/66405)
- Result >= -1.025641 and result < 6.705969 = 0.612597 (88943/145190)
- Result >= 6.705969 = 0.357996 (29370/82040)

Four days previous twenty one days price increasing percentages = 0.423887 (149238/352070)

- Result < -9.349593 = 0.35172 (20647/58703)
- Result >= -9.349593 and result < -1.345291 = 0.20102 (13209/65710)
- Result >= -1.345291 and result < 7.900677 = 0.622626 (87591/140680)
- Result >= 7.900677 = 0.319521 (27791/86977)

Three days previous seven days price increasing percentages = 0.464592 (163516/351956)

- Result < -5.531915 = 0.422976 (25063/59254)
- Result >= -5.531915 and result < -0.729927 = 0.234049 (13899/59385)
- Result >= -0.729927 and result < 4.938272 = 0.607115 (94220/155193)
- Result >= 4.938272 = 0.38828 (30334/78124)

Three days previous fourteen days price increasing percentages = 0.418095 (147151/351956)

- Result < -7.777778 = 0.345304 (20087/58172)
- Result >= -7.777778 and result < -1.032541 = 0.183721 (12146/66111)
- Result >= -1.032541 and result < 6.666667 = 0.583945 (85002/145565)
- Result >= 6.666667 = 0.364349 (29916/82108)

Three days previous twenty one days price increasing percentages = 0.416595 (146623/351956)

- Result < -9.352518 = 0.353654 (20664/58430)
- Result >= -9.352518 and result < -1.35267 = 0.187207 (12269/65537)
- Result >= -1.35267 and result < 7.874016 = 0.602514 (84849/140825)
- Result >= 7.874016 = 0.330882 (28841/87164)

Yesterday seven days price increasing percentages = 0.43466 (152949/351882)

- Result < -5.536332 = 0.368545 (21830/59233)
- Result >= -5.536332 and result < -0.7352941 = 0.177834 (10550/59325)
- Result >= -0.7352941 and result < 4.918033 = 0.605287 (93932/155186)
- Result >= 4.918033 = 0.340897 (26637/78138)

Yesterday fourteen days price increasing percentages = 0.407225 (143295/351882)

- Result < -7.777778 = 0.325109 (18829/57916)

- Result >= -7.777778 and result < -1.041667 = 0.178577 (11762/65865)
- Result >= -1.041667 and result < 6.666667 = 0.578229 (84389/145944)
- Result >= 6.666667 = 0.344645 (28315/82157)

Yesterday twenty one days price increasing percentages = 0.401061 (141126/351882)

- Result < -9.356725 = 0.302768 (17609/58160)
- Result >= -9.356725 and result < -1.369863 = 0.153899 (10037/65218)
- Result >= -1.369863 and result < 7.841098 = 0.601497 (84864/141088)
- Result >= 7.841098 = 0.327354 (28616/87416)

Today seven days price increasing percentages = 0.410482 (144413/351813)

- Result < -5.540166 = 0.35051 (20692/59034)
- Result >= -5.540166 and result < -0.7407407 = 0.149669 (8839/59057)
- Result >= -0.7407407 and result < 4.907407 = 0.568198 (88302/155407)
- Result >= 4.907407 = 0.339399 (26580/78315)

Today fourteen days price increasing percentages = 0.390381 (137341/351813)

- Result < -7.783019 = 0.339354 (19559/57636)
- Result >= -7.783019 and result < -1.048951 = 0.176043 (11575/65751)
- Result >= -1.048951 and result < 6.626506 = 0.533251 (77340/145035)
- Result >= 6.626506 = 0.346164 (28867/83391)

Today twenty one days price increasing percentages = 0.399141 (140423/351813)

- Result < -9.356725 = 0.32375 (18739/57881)
- Result >= -9.356725 and result < -1.385681 = 0.0876896 (5705/65059)
- Result >= -1.385681 and result < 7.792208 = 0.615775 (86972/141240)
- Result >= 7.792208 = 0.331005 (29007/87633)

Seven days previous seven days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -5.536332	Result >= -5.536332 and result < -0.7246377	Result >= -0.7246377 and result < 4.991681	Result >= 4.991681
Result < -5.536332	27059	17483	5412	9192
Result >= -5.536332 and result < -0.7246377	9193	27244	16892	5691
Result >= -0.7246377 and result < 4.991681	16293	72644	48550	17673

Result >= 4.991681	12036	14511	20941	31126

Seven days previous fourteen days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -7.777778	Result >= -7.777778 and result < -1.007557	Result >= -1.007557 and result < 6.773334	Result >= 6.773334
Result < -7.777778	25376	14622	7932	10962
Result >= -7.777778 and result < -1.007557	11212	25277	21671	8509
Result >= -1.007557 and result < 6.773334	16821	59604	48158	20063
Result >= 6.773334	15399	15559	19991	30784

Seven days previous twenty one days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -9.340659	Result >= -9.340659 and result < -1.315789	Result >= -1.315789 and result < 8	Result >= 8
Result < -9.340659	25978	14667	8188	10488
Result >= -9.340659 and result < -1.315789	11853	23111	22785	8344
Result >= -1.315789 and result < 8	15689	54209	51137	18896
Result >= 8	15732	17045	22034	31784

Six days previous seven days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -5.528613	Result >= -5.528613 and result < -	Result >= -0.7246377 and result <	Result >= 4.975124

		0.7246377	4.975124	
Result < -5.528613	26778	12163	9256	10962
Result >= -5.528613 and result < -0.7246377	9684	19099	24234	6114
Result >= -0.7246377 and result < 4.975124	20961	21601	93622	19053
Result >= 4.975124	15113	6109	28178	29044

Six days previous fourteen days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -7.777778	Result >= -7.777778 and result < -1.010101	Result >= -1.010101 and result < 6.75583	Result >= 6.75583
Result < -7.777778	25127	9888	12179	11554
Result >= -	11761	18260	28035	8596

7.777778 and result < -1.010101				
Result >= -1.010101 and result < 6.75583	18386	21640	85603	19104
Result >= 6.75583	17199	9968	26083	28588

Six days previous twenty one days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -9.338521	Result >= -9.338521 and result < -1.324503	Result >= -1.324503 and result < 7.972379	Result >= 7.972379
Result < -9.338521	24826	8872	13754	11672
Result >= -9.338521 and result < -1.324503	12138	16011	28559	9247
Result >= -	16532	18678	85178	19854

1.324503 and result < 7.972379				
Result >= 7.972379	16826	9145	30300	30379

Five days previous seven days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -5.527638	Result >= -5.527638 and result < -0.7246377	Result >= -0.7246377 and result < 4.964539	Result >= 4.964539
Result < -5.527638	26899	8879	12219	11222
Result >= -5.527638 and result < -0.7246377	9256	16546	26891	6460
Result >= -0.7246377 and result < 4.964539	20706	20375	93896	20304
Result >=	14660	6238	26953	30453

4.964539				

Five days previous fourteen days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -7.777778	Result >= -7.777778 and result < -1.01833	Result >= -1.01833 and result < 6.730769	Result >= 6.730769
Result < -7.777778	25020	8217	13454	11930
Result >= -7.777778 and result < -1.01833	12372	17252	28037	8891
Result >= -1.01833 and result < 6.730769	17882	21655	85171	20249
Result >= 6.730769	16192	8524	27179	30032

Five days previous twenty one days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -9.343003	Result >= -9.343003 and result < -1.333333	Result >= -1.333333 and result < 7.936508	Result >= 7.936508
Result < -9.343003	23689	7397	14983	12861
Result >= -9.343003 and result < -1.333333	12347	14641	28820	10098
Result >= -1.333333 and result < 7.936508	17565	18554	83894	20424
Result >= 7.936508	17781	8482	30553	29968

Four days previous seven days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -	Result >= -	Result >= -	Result >=

	5.528846	5.528846 and result < -0.7264802	0.7264802 and result < 4.951857	4.951857
Result < -5.528846	25568	11006	11408	11268
Result >= -5.528846 and result < -0.7264802	8552	19128	25829	5837
Result >= -0.7264802 and result < 4.951857	15007	30334	90573	19365
Result >= 4.951857	12349	12628	22039	31179

Four days previous fourteen days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -7.777778	Result >= -7.777778 and result < -1.025641	Result >= -1.025641 and result < 6.705969	Result >= 6.705969
Result < -	22990	6828	14626	13991

7.777778				
Result >= -7.777778 and result < -1.025641	13028	11805	31473	10099
Result >= -1.025641 and result < 6.705969	19554	14363	88943	22330
Result >= 6.705969	18680	6148	27842	29370

Four days previous twenty one days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -9.349593	Result >= -9.349593 and result < -1.345291	Result >= -1.345291 and result < 7.900677	Result >= 7.900677
Result < -9.349593	20647	8481	17207	12368
Result >= -9.349593 and result < -	11608	13209	31532	9361

1.345291				
Result >= -1.345291 and result < 7.900677	14774	19005	87591	19310
Result >= 7.900677	16090	10803	32293	27791

Three days previous seven days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -5.531915	Result >= -5.531915 and result < -0.729927	Result >= -0.729927 and result < 4.938272	Result >= 4.938272
Result < -5.531915	25063	7129	13813	13249
Result >= -5.531915 and result < -0.729927	10192	13899	28067	7227
Result >= -0.729927 and result <	21122	17139	94220	22712

4.938272				
Result >= 4.938272	16260	5647	25883	30334

Three days previous fourteen days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -7.777778	Result >= -7.777778 and result < -1.032541	Result >= -1.032541 and result < 6.666667	Result >= 6.666667
Result < -7.777778	20087	6244	14934	16907
Result >= -7.777778 and result < -1.032541	12817	12146	28963	12185
Result >= -1.032541 and result < 6.666667	18011	16264	85002	26288
Result >= 6.666667	18467	6555	27170	29916

Three days previous twenty one days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -9.352518	Result >= -9.352518 and result < -1.35267	Result >= -1.35267 and result < 7.874016	Result >= 7.874016
Result < -9.352518	20664	5837	16321	15608
Result >= -9.352518 and result < -1.35267	12730	12269	30386	10152
Result >= -1.35267 and result < 7.874016	18077	16184	84849	21715
Result >= 7.874016	19313	7654	31356	28841

Yesterday seven days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -	Result >= -	Result >= -	Result >=

	5.536332	5.536332 and result < -0.7352941	0.7352941 and result < 4.918033	4.918033
Result < -5.536332	21830	4997	15878	16528
Result >= -5.536332 and result < -0.7352941	10400	10550	30081	8294
Result >= -0.7352941 and result < 4.918033	21765	14415	933932	25074
Result >= 4.918033	18782	5378	27341	26637

Yesterday fourteen days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -7.777778	Result >= -7.777778 and result < -1.041667	Result >= -1.041667 and result < 6.666667	Result >= 6.666667
Result < -	18829	5502	15482	18103

7.777778				
Result >= -7.777778 and result < -1.041667	12248	11762	29855	12000
Result >= -1.041667 and result < 6.666667	18954	16809	84389	25792
Result >= 6.666667	19391	6993	27458	28315

Yesterday twenty one days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -9.356725	Result >= -9.356725 and result < -1.369863	Result >= -1.369863 and result < 7.841098	Result >= 7.841098
Result < -9.356725	17609	5037	17063	18451
Result >= -9.356725 and result < -	12003	10037	31071	12107

1.369863				
Result >= -1.369863 and result < 7.841098	17621	15004	84864	23599
Result >= 7.841098	19750	7338	31712	28616

Today seven days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -5.540166	Result >= -5.540166 and result < -0.7407407	Result >= -0.7407407 and result < 4.907407	Result >= 4.907407
Result < -5.540166	20692	3386	15749	19207
Result >= -5.540166 and result < -0.7407407	10683	8839	28832	10703
Result >= -0.7407407 and result <	21227	15188	88302	30690

4.907407				
Result >= 4.907407	20727	6350	24658	36580

Today fourteen days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -7.783019	Result >= -7.783019 and result < -1.048951	Result >= -1.048951 and result < 6.626506	Result >= 6.626506
Result < -7.783019	19559	3832	14792	19453
Result >= -7.783019 and result < -1.048951	13595	11575	27774	12807
Result >= -1.048951 and result < 6.626506	20539	19385	77340	27771
Result >= 6.626506	21067	8070	25387	28867

Today twenty one days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -9.356725	Result >= -9.356725 and result < -1.385681	Result >= -1.385681 and result < 7.792208	Result >= 7.792208
Result < -9.356725	18739	2074	17391	19677
Result >= -9.356725 and result < -1.385681	13578	5705	32249	13527
Result >= --1.385681 and result < 7.792208	19179	9134	86972	25955
Result >= 7.792208	21594	4892	32140	29007

Most accurate Bayesian Network with seven days data with learning parameter

Accuracy for all 21 nodes = 0.424993 (3141150/7391067)

Seven days previous seven days price increasing percentages = 0.380826 (134028/351940)

- Result < -5.536332 = 0.457546 (27062/59146)
- Result >= -5.536332 and result < -0.7246377 = 0.461606 (27244/59020)
- Result >= -0.7246377 and result < 4.991681 = 0.313496 (48642/155160)
- Result >= 4.991681 = 0.395349 (31080/78614)

Seven days previous fourteen days price increasing percentages = 0.368202 (129585/351940)

- Result < -7.777778 = 0.43084 (25373/58892)
- Result >= -7.777778 and result < -1.007557 = 0.379082 (25273/66669)
- Result >= -1.007557 and result < 6.773334 = 0.332902 (48153/144646)
- Result >= 6.773334 = 0.376665 (30786/81733)

Seven days previous twenty one days price increasing percentages = 0.375084 (132007/351940)

- Result < -9.340659 = 0.437906 (25977/59321)
- Result >= -9.340659 and result < -1.315789 = 0.349689 (23112/66093)
- Result >= -1.315789 and result < 8 = 0.365437 (51136/139931)
- Result >= 8 = 0.367019 (31782/86595)

Six days previous seven days price increasing percentages = 0.47886 (168545/351971)

- Result < -5.528613 = 0.452712 (26782/59159)
- Result >= -5.528613 and result < -0.7246377 = 0.322961 (19097/59131)
- Result >= -0.7246377 and result < 4.975124 = 0.603084 (93621/155237)
- Result >= 4.975124 = 0.370264 (29045/78444)

Six days previous fourteen days price increasing percentages = 0.447699 (157577/351971)

- Result < -7.777778 = 0.427691 (25126/58748)
- Result >= -7.777778 and result < -1.010101 = 0.273975 (18261/66652)
- Result >= -1.010101 and result < 6.75583 = 0.591413 (85597/144733)
- Result >= 6.75583 = 0.349385 (28593/81838)

Six days previous twenty one days price increasing percentages = 0.444366 (156404/351971)

- Result < -9.338521 = 0.419863 (24824/59124)
- Result >= -9.338521 and result < -1.324503 = 0.242832 (16016/65955)
- Result >= -1.324503 and result < 7.972379 = 0.607386 (85181/140242)
- Result >= 7.972379 = 0.350641 (30383/86650)

Five days previous seven days price increasing percentages = 0.476636 (167803/352057)

- Result < -5.527638 = 0.454195 (26897/59219)
- Result >= -5.527638 and result < -0.7246377 = 0.279345 (16552/59253)

- Result >= -0.7246377 and result < 4.964539 = 0.604665 (93893/155281)
- Result >= 4.964539 = 0.38901 (30461/78304)

Five days previous fourteen days price increasing percentages = 0.447274 (157466/352057)

- Result < -7.777778 = 0.426758 (25017/58621)
- Result >= -7.777778 and result < -1.01833 = 0.259256 (17254/66552)
- Result >= -1.01833 and result < 6.730769 = 0.58745 (85155/144957)
- Result >= 6.730769 = 0.366668 (30040/81927)

Five days previous twenty one days price increasing percentages = 0.432282 (152188/352057)

- Result < -9.343003 = 0.402002 (23690/58930)
- Result >= -9.343003 and result < -1.333333 = 0.22221 (14645/65906)
- Result >= -1.333333 and result < 7.936508 = 0.597293 (83882/140437)
- Result >= 7.936508 = 0.345352 (29971/86784)

Four days previous seven days price increasing percentages = 0.472744 (166439/352070)

- Result < -5.528846 = 0.431662 (25576/59250)
- Result >= -5.528846 and result < -0.7264802 = 0.322145 (19118/59346)
- Result >= -0.7264802 and result < 4.951857 = 0.583215 (90561/155279)

- Result >= 4.951857 = 0.398798 (31184/78195)

Four days previous fourteen days price increasing percentages = 0.434877 (153107/352070)

- Result < -7.777778 = 0.393548 (22997/58435)
- Result >= -7.777778 and result < -1.025641 = 0.177848 (11810/66405)
- Result >= -1.025641 and result < 6.705969 = 0.612473 (88925/145190)
- Result >= 6.705969 = 0.358057 (29375/82040)

Four days previous twenty one days price increasing percentages = 0.423887 (149238/352070)

- Result < -9.349593 = 0.351737 (20648/58703)
- Result >= -9.349593 and result < -1.345291 = 0.20102 (13209/65710)
- Result >= -1.345291 and result < 7.900677 = 0.622597 (87587/140680)
- Result >= 7.900677 = 0.319556 (27794/86977)

Three days previous seven days price increasing percentages = 0.464555 (163503/351956)

- Result < -5.531915 = 0.423094 (25070/59254)
- Result >= -5.531915 and result < -0.729927 = 0.233763 (13882/59385)
- Result >= -0.729927 and result < 4.938272 = 0.607063 (94212/155193)
- Result >= 4.938272 = 0.388344 (30339/78124)

Three days previous fourteen days price increasing percentages = 0.418103 (147154/351956)

- Result < -7.777778 = 0.34563 (20106/58172)
- Result >= -7.777778 and result < -1.032541 = 0.183479 (12130/66111)
- Result >= -1.032541 and result < 6.666667 = 0.583842 (84987/145565)
- Result >= 6.666667 = 0.364532 (29931/82108)

Three days previous twenty one days price increasing percentages = 0.416586 (146620/351956)

- Result < -9.352518 = 0.353722 (20668/58430)
- Result >= -9.352518 and result < -1.35267 = 0.187329 (12277/65537)
- Result >= -1.35267 and result < 7.874016 = 0.6024 (84833/140825)
- Result >= 7.874016 = 0.330893 (28842/87164)

Yesterday seven days price increasing percentages = 0.434606 (152930/351882)

- Result < -5.536332 = 0.36846 (21825/59233)
- Result >= -5.536332 and result < -0.7352941 = 0.177868 (10552/59325)
- Result >= -0.7352941 and result < 4.918033 = 0.605222 (93922/155186)
- Result >= 4.918033 = 0.34082 (26631/78138)

Yesterday fourteen days price increasing percentages = 0.407216 (143292/351882)

- Result < -7.777778 = 0.325057 (18826/57916)
- Result >= -7.777778 and result < -1.041667 = 0.17876 (11774/65865)
- Result >= -1.041667 and result < 6.666667 = 0.578085 (84368/145944)
- Result >= 6.666667 = 0.344755 (28324/82157)

Yesterday twenty one days price increasing percentages = 0.401112 (141144/351882)

- Result < -9.356725 = 0.302871 (17615/58160)
- Result >= -9.356725 and result < -1.369863 = 0.154037 (10046/65218)
- Result >= -1.369863 and result < 7.841098 = 0.601433 (84855/141088)
- Result >= 7.841098 = 0.327492 (28628/87416)

Today seven days price increasing percentages = 0.410428 (144394/351813)

- Result < -5.540166 = 0.350595 (20697/59034)
- Result >= -5.540166 and result < -0.7407407 = 0.149601 (8835/59057)
- Result >= -0.7407407 and result < 4.907407 = 0.567954 (88264/155407)
- Result >= 4.907407 = 0.339628 (26598/78315)

Today fourteen days price increasing percentages = 0.390355 (137332/351813)

- Result < -7.783019 = 0.339371 (19560/57636)
- Result >= -7.783019 and result < -1.048951 = 0.176089 (11578/65751)
- Result >= -1.048951 and result < 6.626506 = 0.533147 (77325/145035)
- Result >= 6.626506 = 0.346188 (28869/83391)

Today twenty one days price increasing percentages = 0.399059 (140394/351813)

- Result < -9.356725 = 0.323768 (18740/57881)
- Result >= -9.356725 and result < -1.385681 = 0.0877972 (5712/65059)
- Result >= -1.385681 and result < 7.792208 = 0.615534 (86938/141240)
- Result >= 7.792208 = 0.330971 (29004/87633)

Seven days previous seven days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -5.536332	Result >= -5.536332 and result < -0.7246377	Result >= -0.7246377 and result < 4.991681	Result >= 4.991681
Result < -5.536332	27062	17479	5415	9190
Result >= -5.536332 and result < -0.7246377	9193	27244	16910	5673
Result >= -0.7246377 and result <	16293	72644	48642	17581

4.991681				
Result >= 4.991681	12038	14509	20987	31080

Seven days previous fourteen days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -7.777778	Result >= -7.777778 and result < -1.007557	Result >= -1.007557 and result < 6.773334	Result >= 6.773334
Result < -7.777778	25373	14621	7934	10964
Result >= -7.777778 and result < -1.007557	11212	25273	21673	8511
Result >= -1.007557 and result < 6.773334	16820	59609	48153	20064
Result >= 6.773334	15400	15553	19994	30786

Seven days previous twenty one days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -9.340659	Result >= -9.340659 and result < -1.315789	Result >= -1.315789 and result < 8	Result >= 8
Result < -9.340659	25977	14666	8190	10488
Result >= -9.340659 and result < -1.315789	11851	23112	22786	8344
Result >= -1.315789 and result < 8	15691	54209	51136	18895
Result >= 8	15736	17046	22031	31782

Six days previous seven days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -5.528613	Result >= -5.528613 and result < -	Result >= -0.7246377 and result <	Result >= 4.975124

		0.7246377	4.975124	
Result < -5.528613	26782	12158	9254	10965
Result >= -5.528613 and result < -0.7246377	9685	19097	24235	6114
Result >= -0.7246377 and result < 4.975124	20960	21603	93621	19053
Result >= 4.975124	15112	6108	28179	29045

Six days previous fourteen days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -7.777778	Result >= -7.777778 and result < -1.010101	Result >= -1.010101 and result < 6.75583	Result >= 6.75583
Result < -7.777778	25126	9887	12178	11557
Result >= -	11760	18261	28034	8597

7.777778 and result < -1.010101				
Result >= -1.010101 and result < 6.75583	18383	21643	85597	19110
Result >= 6.75583	17198	9967	26080	28593

Six days previous twenty one days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -9.338521	Result >= -9.338521 and result < -1.324503	Result >= -1.324503 and result < 7.972379	Result >= 7.972379
Result < -9.338521	24824	8872	13754	11674
Result >= -9.338521 and result < -1.324503	12136	16016	28556	9247
Result >= -	16534	18677	85181	19850

1.324503 and result < 7.972379				
Result >= 7.972379	16827	9145	30295	30383

Five days previous seven days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -5.527638	Result >= -5.527638 and result < -0.7246377	Result >= -0.7246377 and result < 4.964539	Result >= 4.964539
Result < -5.527638	26897	8874	12216	11232
Result >= -5.527638 and result < -0.7246377	9358	16552	26883	6460
Result >= -0.7246377 and result < 4.964539	20701	20378	93893	20309
Result >=	14664	6232	26947	30461

4.964539				

Five days previous fourteen days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -7.777778	Result >= -7.777778 and result < -1.01833	Result >= -1.01833 and result < 6.730769	Result >= 6.730769
Result < -7.777778	25017	8218	13454	11932
Result >= -7.777778 and result < -1.01833	12373	17254	28032	8893
Result >= -1.01833 and result < 6.730769	17883	21658	85155	20261
Result >= 6.730769	16190	8532	27165	30040

Five days previous twenty one days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -9.343003	Result >= -9.343003 and result < -1.333333	Result >= -1.333333 and result < 7.936508	Result >= 7.936508
Result < -9.343003	23690	7395	14984	12861
Result >= -9.343003 and result < -1.333333	12347	14645	28815	10099
Result >= -1.333333 and result < 7.936508	17568	18560	83882	20427
Result >= 7.936508	17785	8483	30545	29971

Four days previous seven days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -	Result >= -	Result >= -	Result >=

	5.528846	5.528846 and result < -0.7264802	0.7264802 and result < 4.951857	4.951857
Result < -5.528846	25576	10993	11406	11275
Result >= -5.528846 and result < -0.7264802	8553	19118	25831	5844
Result >= -0.7264802 and result < 4.951857	15008	30328	90561	19382
Result >= 4.951857	12355	12619	22037	31184

Four days previous fourteen days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -7.777778	Result >= -7.777778 and result < -1.025641	Result >= -1.025641 and result < 6.705969	Result >= 6.705969
Result < -	22997	6820	14627	13991

7.777778				
Result >= -7.777778 and result < -1.025641	13034	11810	31462	10099
Result >= -1.025641 and result < 6.705969	19557	14368	88925	22340
Result >= 6.705969	18684	3147	27834	29375

Four days previous twenty one days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -9.349593	Result >= -9.349593 and result < -1.345291	Result >= -1.345291 and result < 7.900677	Result >= 7.900677
Result < -9.349593	20648	8482	17205	12368
Result >= -9.349593 and result < -	11611	13209	31527	9363

1.345291				
Result >= -1.345291 and result < 7.900677	14776	19007	87587	19310
Result >= 7.900677	16094	10792	32297	27794

Three days previous seven days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -5.531915	Result >= -5.531915 and result < -0.729927	Result >= -0.729927 and result < 4.938272	Result >= 4.938272
Result < -5.531915	25070	7122	13812	13250
Result >= -5.531915 and result < -0.729927	10198	13882	28074	7231
Result >= -0.729927 and result <	21130	17128	94212	22723

4.938272				
Result >= 4.938272	16268	5640	25877	30339

Three days previous fourteen days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -7.777778	Result >= -7.777778 and result < -1.032541	Result >= -1.032541 and result < 6.666667	Result >= 6.666667
Result < -7.777778	20106	6219	14930	16917
Result >= -7.777778 and result < -1.032541	12833	12130	28955	12193
Result >= -1.032541 and result < 6.666667	18026	16252	84987	26300
Result >= 6.666667	18473	6546	27158	29931

Three days previous twenty one days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -9.352518	Result >= -9.352518 and result < -1.35267	Result >= -1.35267 and result < 7.874016	Result >= 7.874016
Result < -9.352518	20668	5835	16317	15610
Result >= -9.352518 and result < -1.35267	12727	12277	30381	10152
Result >= -1.35267 and result < 7.874016	18079	16195	84833	21718
Result >= 7.874016	19318	7660	31344	28842

Yesterday seven days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -	Result >= -	Result >= -	Result >=

	5.536332	5.536332 and result < -0.7352941	0.7352941 and result < 4.918033	4.918033
Result < -5.536332	21825	5005	15869	16534
Result >= -5.536332 and result < -0.7352941	10401	10552	30077	8295
Result >= -0.7352941 and result < 4.918033	21764	14419	93922	25081
Result >= 4.918033	18788	5385	27334	26631

Yesterday fourteen days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -7.777778	Result >= -7.777778 and result < -1.041667	Result >= -1.041667 and result < 6.666667	Result >= 6.666667
Result < -	18826	5513	15473	18104

7.777778				
Result >= -7.777778 and result < -1.041667	12245	11774	29838	12008
Result >= -1.041667 and result < 6.666667	18947	16825	84368	25804
Result >= 6.666667	19382	7006	27445	28324

Yesterday twenty one days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -9.356725	Result >= -9.356725 and result < -1.369863	Result >= -1.369863 and result < 7.841098	Result >= 7.841098
Result < -9.356725	17615	5036	17060	18449
Result >= -9.356725 and result < -	11999	10046	31062	12111

1.369863				
Result >= -1.369863 and result < 7.841098	17630	15009	84855	23594
Result >= 7.841098	19752	7342	31694	28628

Today seven days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -5.540166	Result >= -5.540166 and result < -0.7407407	Result >= -0.7407407 and result < 4.907407	Result >= 4.907407
Result < -5.540166	20697	3383	15726	19228
Result >= -5.540166 and result < -0.7407407	10688	8835	28813	10721
Result >= -0.7407407 and result <	21237	15188	88264	30718

4.907407				
Result >= 4.907407	20733	6343	24641	26598

Today fourteen days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -7.783019	Result >= -7.783019 and result < -1.048951	Result >= -1.048951 and result < 6.626506	Result >= 6.626506
Result < -7.783019	19560	3828	14787	19461
Result >= -7.783019 and result < -1.048951	13595	11578	27759	12819
Result >= -1.048951 and result < 6.626506	20553	19385	77325	27772
Result >= 6.626506	21074	8074	25374	28869

Today twenty one days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -9.356725	Result >= -9.356725 and result < -1.385681	Result >= -1.385681 and result < 7.792208	Result >= 7.792208
Result < -9.356725	18740	2067	17383	19691
Result >= -9.356725 and result < -1.385681	13577	5712	32240	13530
Result >= --1.385681 and result < 7.792208	19185	9158	86938	25959
Result >= 7.792208	21599	4907	32123	29004

Most accurate Bayesian Network with seven days data without learning parameter (without seven days previous seven days price increasing percentages)

Accuracy for all 20 nodes = 0.474472 (3356102/7073341)

Seven days previous fourteen days price increasing percentages = 0.57234 (202412/353657)

- Result < -7.777778 = 0.563425 (33397/59275)
- Result >= -7.777778 and result < -1.007557 = 0.425277 (28525/67074)
- Result >= -1.007557 and result < 6.773334 = 0.654843 (95150/145302)
- Result >= 6.773334 = 0.552886 (45340/82006)

Seven days previous twenty one days price increasing percentages = 0.509324 (180126/353657)

- Result < -9.340659 = 0.51416 (30736/59779)
- Result >= -9.340659 and result < -1.315789 = 0.352194 (23435/66540)
- Result >= -1.315789 and result < 8 = 0.616974 (86688/140505)
- Result >= 8 = 0.452213 (39267/86833)

Six days previous seven days price increasing percentages = 0.688734 (243597/353688)

- Result < -5.528613 = 0.705827 (41948/59431)
- Result >= -5.528613 and result < -0.7246377 = 0.559625 (33216/59354)
- Result >= -0.7246377 and result < 4.975124 = 0.694494 (108373/156046)
- Result >= 4.975124 = 0.761632 (60060/78857)

Six days previous fourteen days price increasing percentages = 0.514142 (181846/353688)

- Result < -7.777778 = 0.5293 (31306/59146)
- Result >= -7.777778 and result < -1.010101 = 0.36328 (24367/67075)
- Result >= -1.010101 and result < 6.75583 = 0.585394 (85101/145374)
- Result >= 6.75583 = 0.500311 (41072/82093)

Six days previous twenty one days price increasing percentages = 0.484441 (171341/353688)

- Result < -9.338521 = 0.481655 (28724/59636)
- Result >= -9.338521 and result < -1.324503 = 0.315681 (20965/66412)
- Result >= -1.324503 and result < 7.972379 = 0.598216 (84221/140787)
- Result >= 7.972379 = 0.43097 (37431/86853)

Five days previous seven days price increasing percentages = 0.548622 (194081/353761)

- Result < -5.527638 = 0.543547 (32328/59476)
- Result >= -5.527638 and result < -0.7246377 = 0.324135 (19287/59503)
- Result >= -0.7246377 and result < 4.964539 = 0.639202 (99773/156090)
- Result >= 4.964539 = 0.542533 (42693/78692)

Five days previous fourteen days price increasing percentages = 0.480375 (169938/353761)

- Result < -7.777778 = 0.483908 (28567/59034)
- Result >= -7.777778 and result < -1.01833 = 0.270505 (18116/66971)
- Result >= -1.01833 and result < 6.730769 = 0.60222 (87679/145593)
- Result >= 6.730769 = 0.432993 (35576/82163)

Five days previous twenty one days price increasing percentages = 0.462196 (163507/353761)

- Result < -9.343003 = 0.447802 (26629/59466)
- Result >= -9.343003 and result < -1.333333 = 0.238697 (15839/66356)
- Result >= -1.333333 and result < 7.936508 = 0.608866 (85830/140967)
- Result >= 7.936508 = 0.404831 (35209/86972)

Four days previous seven days price increasing percentages = 0.502056 (177617/353779)

- Result < -5.528846 = 0.459963 (27383/59533)
- Result >= -5.528846 and result < -0.7264802 = 0.348771 (20800/59638)
- Result >= -0.7264802 and result < 4.951857 = 0.605097 (94435/156066)
- Result >= 4.951857 = 0.445609 (34999/78542)

Four days previous fourteen days price increasing percentages = 0.465454 (164668/353779)

- Result < -7.777778 = 0.448434 (26415/58905)
- Result >= -7.777778 and result < -1.025641 = 0.192851 (12895/66865)

- Result >= -1.025641 and result < 6.705969 = 0.622472 (90729/145756)
- Result >= 6.705969 = 0.421006 (34629/82253)

Four days previous twenty one days price increasing percentages = 0.444113 (157118/353779)

- Result < -9.349593 = 0.386466 (22902/59260)
- Result >= -9.349593 and result < -1.345291 = 0.203175 (13440/66150)
- Result >= -1.345291 and result < 7.900677 = 0.636715 (89901/141195)
- Result >= 7.900677 = 0.354177 (30875/87174)

Three days previous seven days price increasing percentages = 0.482051 (170485/353666)

- Result < -5.531915 = 0.440533 (26232/59546)
- Result >= -5.531915 and result < -0.729927 = 0.24027 (14341/59687)
- Result >= -0.729927 and result < 4.938272 = 0.618537 (96475/155973)
- Result >= 4.938272 = 0.426166 (33437/78460)

Three days previous fourteen days price increasing percentages = 0.433318 (153250/353666)

- Result < -7.777778 = 0.374291 (21966/58687)
- Result >= -7.777778 and result < -1.032541 = 0.181986 (12115/66571)
- Result >= -1.032541 and result < 6.666667 = 0.588939 (86047/146105)
- Result >= 6.666667 = 0.40244 (33122/82303)

Three days previous twenty one days price increasing percentages = 0.425277 (150406/353666)

- Result < -9.352518 = 0.359855 (21221/58971)
- Result >= -9.352518 and result < -1.35267 = 0.192212 (12681/65974)
- Result >= -1.35267 and result < 7.874016 = 0.606511 (85737/141361)
- Result >= 7.874016 = 0.352186 (30767/87360)

Yesterday seven days price increasing percentages = 0.44122 (156012/353592)

- Result < -5.536332 = 0.381544 (22708/59516)
- Result >= -5.536332 and result < -0.7352941 = 0.182695 (10895/59635)
- Result >= -0.7352941 and result < 4.918033 = 0.609788 (95116/155982)
- Result >= 4.918033 = 0.347863 (27293/78459)

Yesterday fourteen days price increasing percentages = 0.412227 (145760/353592)

- Result < -7.777778 = 0.328981 (19223/58432)
- Result >= -7.777778 and result < -1.041667 = 0.183973 (12204/66336)
- Result >= -1.041667 and result < 6.666667 = 0.580143 (84988/146495)
- Result >= 6.666667 = 0.356436 (29345/82329)

Yesterday twenty one days price increasing percentages = 0.405909 (143526/353592)

- Result < -9.356725 = 0.308664 (18113/58682)

- Result >= -9.356725 and result < -1.369863 = 0.159005 (10442/65671)
- Result >= -1.369863 and result < 7.841098 = 0.602619 (85352/141635)
- Result >= 7.841098 = 0.338101 (29619/87604)

Today seven days price increasing percentages = 0.416875 (147375/353523)

- Result < -5.540166 = 0.349724 (20810/59504)
- Result >= -5.540166 and result < -0.7407407 = 0.143407 (8533/59502)
- Result >= -0.7407407 and result < 4.907407 = 0.580921 (90622/155997)
- Result >= 4.907407 = 0.349083 (27410/78520)

Today fourteen days price increasing percentages = 0.395923 (139968/353523)

- Result < -7.783019 = 0.334319 (19450/58178)
- Result >= -7.783019 and result < -1.048951 = 0.167369 (11082/66213)
- Result >= -1.048951 and result < 6.626506 = 0.546895 (79611/145569)
- Result >= 6.626506 = 0.356916 (29825/83563)

Today twenty one days price increasing percentages = 0.404695 (143069/353523)

- Result < -9.356725 = 0.319816 (18675/58393)
- Result >= -9.356725 and result < -1.385681 = 0.0854617 (5598/65503)
- Result >= -1.385681 and result < 7.792208 = 0.626079 (88773/141792)
- Result >= 7.792208 = 0.341811 (30023/87835)

Seven days previous fourteen days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -7.777778	Result >= -7.777778 and result < -1.007557	Result >= -1.007557 and result < 6.773334	Result >= 6.773334
Result < -7.777778	33397	10769	9141	5968
Result >= -7.777778 and result < -1.007557	11669	28525	20986	5894
Result >= -1.007557 and result < 6.773334	8745	19431	95150	21976
Result >= 6.773334	5987	5248	25431	45340

Seven days previous twenty one days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -	Result >= -	Result >= -	Result >= 8

	9.340659	9.340659 and result < -1.315789	1.315789 and result < 8	
Result < -9.340659	30736	10235	11054	7754
Result >= -9.340659 and result < -1.315789	12773	23435	23043	7289
Result >= -1.315789 and result < 8	12349	19819	86688	21649
Result >= 8	10454	8011	29101	39267

Six days previous seven days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -5.528613	Result >= -5.528613 and result < -0.7246377	Result >= -0.7246377 and result < 4.975124	Result >= 4.975124
Result < -5.528613	41948	10421	2905	4157
Result >= -	7148	33216	15000	3990

5.528613 and result < -0.7246377				
Result >= -0.7246377 and result < 4.975124	7144	15811	108373	24718
Result >= 4.975124	3111	1881	13805	60060

Six days previous fourteen days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -7.777778	Result >= -7.777778 and result < -1.010101	Result >= -1.010101 and result < 6.75583	Result >= 6.75583
Result < -7.777778	31306	10003	8834	9003
Result >= -7.777778 and result < -1.010101	13195	24367	21328	8185
Result >= -	14472	22064	85101	23737

1.010101 and result < 6.75583				
Result >= 6.75583	10514	6483	24024	41072

Six days previous twenty one days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -9.338521	Result >= -9.338521 and result < -1.324503	Result >= -1.324503 and result < 7.972379	Result >= 7.972379
Result < -9.338521	28724	9143	11214	10555
Result >= -9.338521 and result < -1.324503	13042	20965	23312	9093
Result >= -1.324503 and result < 7.972379	14860	19296	84221	22410
Result >=	13692	7400	28330	37431

7.972379				

Five days previous seven days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -5.527638	Result >= -5.527638 and result < -0.7246377	Result >= -0.7246377 and result < 4.964539	Result >= 4.964539
Result < -5.527638	32328	11875	6687	8586
Result >= -5.527638 and result < -0.7246377	9267	19287	24933	5916
Result >= -0.7246377 and result < 4.964539	17624	16595	99773	22098
Result >= 4.964539	8873	3471	23655	42693

Five days previous fourteen days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -7.777778	Result >= -7.777778 and result < -1.01833	Result >= -1.01833 and result < 6.730769	Result >= 6.730769
Result < -7.777778	28567	9008	10642	10817
Result >= -7.777778 and result < -1.01833	13228	18116	26824	8803
Result >= -1.01833 and result < 6.730769	17035	18424	87679	22455
Result >= 6.730769	13396	6343	26848	35576

Five days previous twenty one days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -	Result >= -	Result >= -	Result >=

	9.343003	9.343003 and result < -1.333333	1.333333 and result < 7.936508	7.936508
Result < -9.343003	26629	7820	12631	12386
Result >= -9.343003 and result < -1.333333	13160	15839	27171	10186
Result >= -1.333333 and result < 7.936508	16462	16043	85830	22632
Result >= 7.936508	15635	6745	29383	35209

Four days previous seven days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -5.528846	Result >= -5.528846 and result < -0.7264802	Result >= -0.7264802 and result < 4.951857	Result >= 4.951857
Result < -	27383	14022	8687	9441

5.528846				
Result >= -5.528846 and result < -0.7264802	8867	20800	24510	5461
Result >= -0.7264802 and result < 4.951857	14390	27965	94435	19276
Result >= 4.951857	10263	8348	24932	34999

Four days previous fourteen days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -7.777778	Result >= -7.777778 and result < -1.025641	Result >= -1.025641 and result < 6.705969	Result >= 6.705969
Result < -7.777778	26415	5579	14084	12827
Result >= -7.777778 and result < -	13471	12895	30215	10284

1.025641				
Result >= -1.025641 and result < 6.705969	16594	13224	90729	25209
Result >= 6.705969	14943	4952	27729	34629

Four days previous twenty one days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -9.349593	Result >= -9.349593 and result < -1.345291	Result >= -1.345291 and result < 7.900677	Result >= 7.900677
Result < -9.349593	22902	7120	17480	11758
Result >= -9.349593 and result < -1.345291	11979	13440	31104	9627
Result >= -1.345291 and result <	14413	16528	89901	20353

7.900677				
Result >= 7.900677	14664	8287	33348	30875

Three days previous seven days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -5.531915	Result >= -5.531915 and result < -0.729927	Result >= -0.729927 and result < 4.938272	Result >= 4.938272
Result < -5.531915	26232	7471	13450	12393
Result >= -5.531915 and result < -0.729927	10307	14341	28156	6883
Result >= -0.729927 and result < 4.938272	18428	17250	96475	23820
Result >= 4.938272	13362	5228	26433	33437

Three days previous fourteen days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -7.777778	Result >= -7.777778 and result < -1.032541	Result >= -1.032541 and result < 6.666667	Result >= 6.666667
Result < -7.777778	21966	5669	14343	16709
Result >= -7.777778 and result < -1.032541	13328	12115	28918	12210
Result >= -1.032541 and result < 6.666667	17333	15080	86047	27645
Result >= 6.666667	16976	5642	26563	33122

Three days previous twenty one days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -	Result >= -	Result >= -	Result >=

	9.352518	9.352518 and result < -1.35267	1.35267 and result < 7.874016	7.874016
Result < -9.352518	21221	6215	16201	15334
Result >= -9.352518 and result < -1.35267	12577	12681	30318	10398
Result >= -1.35267 and result < 7.874016	17030	16411	85737	22183
Result >= 7.874016	17443	7754	31396	30767

Yesterday seven days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -5.536332	Result >= -5.536332 and result < -0.7352941	Result >= -0.7352941 and result < 4.918033	Result >= 4.918033
Result < -	22708	4984	15883	15941

5.536332				
Result >= -5.536332 and result < -0.7352941	10536	10895	30159	8045
Result >= -0.7352941 and result < 4.918033	21878	14152	95116	24836
Result >= 4.918033	18238	5228	27700	27293

Yesterday fourteen days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -7.777778	Result >= -7.777778 and result < -1.041667	Result >= -1.041667 and result < 6.666667	Result >= 6.666667
Result < -7.777778	19223	6061	15233	17915
Result >= -7.777778 and result < -	12184	12204	30010	11938

1.041667				
Result >= -1.041667 and result < 6.666667	17901	17503	84988	26103
Result >= 6.666667	18152	7416	27416	29345

Yesterday twenty one days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -9.356725	Result >= -9.356725 and result < -1.369863	Result >= -1.369863 and result < 7.841098	Result >= 7.841098
Result < -9.356725	18113	5473	16768	18328
Result >= -9.356725 and result < -1.369863	11990	10442	31099	12140
Result >= -1.369863 and result <	16797	15486	85352	24000

7.841098				
Result >= 7.841098	18642	7679	31664	29619

Today seven days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -5.540166	Result >= -5.540166 and result < -0.7407407	Result >= -0.7407407 and result < 4.907407	Result >= 4.907407
Result < -5.540166	20810	3474	16161	19059
Result >= -5.540166 and result < -0.7407407	10504	8533	29722	10693
Result >= -0.7407407 and result < 4.907407	20309	14090	90622	30976
Result >= 4.907407	19833	5630	25647	27410

Today fourteen days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -7.783019	Result >= -7.783019 and result < -1.048951	Result >= -1.048951 and result < 6.626506	Result >= 6.626506
Result < -7.783019	19450	3850	15211	19667
Result >= -7.783019 and result < -1.048951	13336	11082	28948	12847
Result >= -1.048951 and result < 6.626506	19317	18196	79611	28445
Result >= 6.626506	20003	7285	26450	29825

Today twenty one days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -	Result >= -	Result >= -	Result >=

	9.356725	9.356725 and result < -1.385681	1.385681 and result < 7.792208	7.792208
Result < -9.356725	18675	1960	17933	19825
Result >= -9.356725 and result < -1.385681	13201	5598	33137	13567
Result >= --1.385681 and result < 7.792208	18405	8234	88773	26380
Result >= 7.792208	20545	4430	32837	30023

Most accurate Bayesian Network with seven days data without learning parameter (with today price increasing percentages, seven days price increasing percentages, fourteen days price increasing percentages , twenty one days price increasing percentages result only)

Accuracy for all 4 nodes = 0.760251 (1075780/1415032)

Today price increasing percentages = 0.780556 (276128/353758)

- Result < -1.587302 = 0.795248 (35977/45240)

- Result >= -1.587302 and result < 0.4016064 = 0.777584 (169287/217709)
- Result >= 0.4016064 and result < 1.834862 = 0.606697 (18989/31299)
- Result >= 1.834862 = 0.871702 (51875/59510)

Today seven days price increasing percentages = 0.692372 (244932/353758)

- Result < -5.540166 = 0.74974 (44623/59518)
- Result >= -5.540166 and result < -0.7407407 = 0.531598 (31654/59545)
- Result >= -0.7407407 and result < 4.907407 = 0.678293 (105906/156136)
- Result >= 4.907407 = 0.79875 (62749/78559)

Today fourteen days price increasing percentages = 0.750533 (265507/353758)

- Result < -7.783019 = 0.841315 (48962/58197)
- Result >= -7.783019 and result < -1.048951 = 0.591888 (39225/66271)
- Result >= -1.048951 and result < 6.626506 = 0.739062 (107671/145686)
- Result >= 6.626506 = 0.833082 (69649/83604)

Today twenty one days price increasing percentages = 0.817545 (289213/353758)

- Result < -9.356725 = 0.873457 (51023/58415)
- Result >= -9.356725 and result < -1.385681 = 0.740527 (48546/65556)

- Result >= -1.385681 and result < 7.792208 = 0.791939 (112392/141920)
- Result >= 7.792208 = 0.879192 (77252/87867)

Today price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -1.587302	Result >= -1.587302 and result < 0.4016064	Result >= 0.4016064 and result < 1.834862	Result >= 1.834862
Result < -1.587302	35977	6907	573	1783
Result >= -1.587302 and result < 0.4016064	11050	169287	27207	10165
Result >= 0.4016064 and result < 1.834862	958	7687	18989	3665
Result >= 1.834862	1138	99	6398	51875

Today seven days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -5.540166	Result >= -5.540166 and result < -0.7407407	Result >= -0.7407407 and result < 4.907407	Result >= 4.907407
Result < -5.540166	44623	7478	2586	4831
Result >= -5.540166 and result < -0.7407407	7827	31654	16182	3882
Result >= -0.7407407 and result < 4.907407	7789	21806	105906	20635
Result >= 4.907407	3297	1731	10782	32749

Today fourteen days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -	Result >= -	Result >= -	Result >=

	7.783019	7.783019 and result < -1.048951	1.048951 and result < 6.626506	6.626506
Result < -7.783019	48962	5667	1426	2142
Result >= -7.783019 and result < -1.048951	8815	39225	15156	3075
Result >= -1.048951 and result < 6.626506	5466	16523	107671	16026
Result >= 6.626506	1702	1249	11004	69649

Today twenty one days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -9.356725	Result >= -9.356725 and result < -1.385681	Result >= -1.385681 and result < 7.792208	Result >= 7.792208
Result < -	51023	5057	1474	861

9.356725				
Result >= -9.356725 and result < -1.385681	6737	48546	9058	1215
Result >= --1.385681 and result < 7.792208	3000	15684	112392	10844
Result >= 7.792208	890	1070	8655	77252

Most accurate Bayesian Network with seven days data without learning parameter (with today seven days price increasing percentages, fourteen days price increasing percentages , twenty one days price increasing percentages result only)

Accuracy for all 3 nodes = 0.753483 (799652/1061274)

Today seven days price increasing percentages = 0.692372 (244932/353758)

- Result < -5.540166 = 0.74974 (44623/59518)
- Result >= -5.540166 and result < -0.7407407 = 0.531598 (31654/59545)
- Result >= -0.7407407 and result < 4.907407 = 0.678293 (105906/156136)
- Result >= 4.907407 = 0.79875 (62749/78559)

Today fourteen days price increasing percentages = 0.750533 (265507/353758)

- Result < -7.783019 = 0.841315 (48962/58197)
- Result >= -7.783019 and result < -1.048951 = 0.591888 (39225/66271)
- Result >= -1.048951 and result < 6.626506 = 0.739062 (107671/145686)
- Result >= 6.626506 = 0.833082 (69649/83604)

Today twenty one days price increasing percentages = 0.817545 (289213/353758)

- Result < -9.356725 = 0.873457 (51023/58415)
- Result >= -9.356725 and result < -1.385681 = 0.740527 (48546/65556)
- Result >= -1.385681 and result < 7.792208 = 0.791939 (112392/141920)
- Result >= 7.792208 = 0.879192 (77252/87867)

Today seven days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -5.540166	Result >= -5.540166 and result < -0.7407407	Result >= -0.7407407 and result < 4.907407	Result >= 4.907407
Result < -	44623	7478	2586	4831

5.540166				
Result >= -5.540166 and result < -0.7407407	7827	31654	16182	3882
Result >= -0.7407407 and result < 4.907407	7789	21806	105906	20635
Result >= 4.907407	3297	1731	10782	62749

Today fourteen days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

	Result < -7.783019	Result >= -7.783019 and result < -1.048951	Result >= -1.048951 and result < 6.626506	Result >= 6.626506
Result < -7.783019	48962	5667	1426	2142
Result >= -7.783019 and result < -	8815	39225	15156	3075

1.048951				
Result >= -1.048951 and result < 6.626506	5466	16523	107671	16026
Result >= 6.626506	1702	1249	11004	69649

Today twenty one days price increasing percentages confusion matrix for the most accurate Bayesian Network with only today data with learning parameter (with set background knowledge)

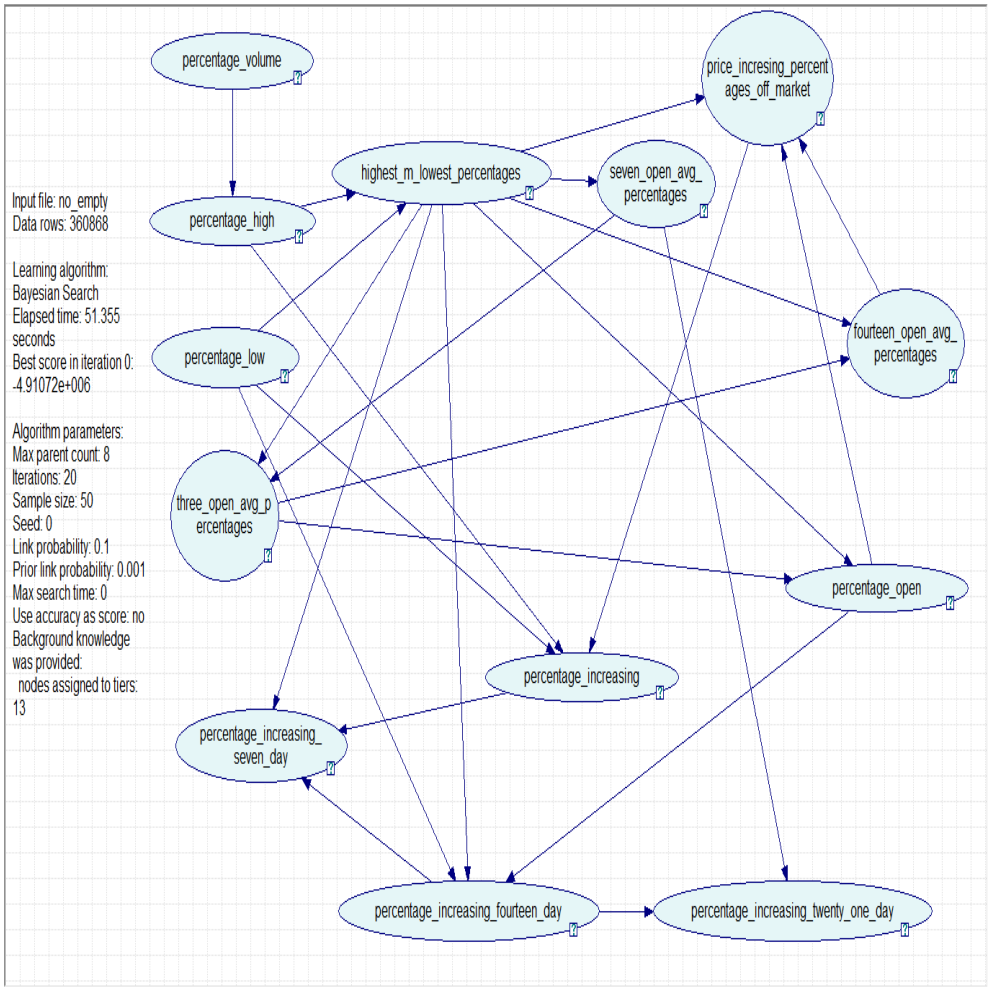
	Result < -9.356725	Result >= -9.356725 and result < -1.385681	Result >= -1.385681 and result < 7.792208	Result >= 7.792208
Result < -9.356725	51023	5057	1474	861
Result >= -9.356725 and result < -1.385681	6737	48546	9058	1215
Result >= --1.385681 and result <	3000	15684	112392	10844

7.792208				
Result >= 7.792208	890	1070	8655	77252

APPENDIX D

Bayesian Network

Second Bayesian Network (with set background knowledge)



Result for second Bayesian Network (with set background knowledge)

Accuracy for all 3 nodes = 0.408201 (3865394/9469347)

Seven days price increasing percentages = 0.441754 (1394373/3156449)

- Result < -5.555555 = 0.284832 (156405/549113)

- Result >= -5.555555 and result < -0.7462686 = 0.235599 (171889/729582)
- Result >= -0.7462686 and result < 4.93421 = 0.721107 (891285/1235996)
- Result >= 4.93421 = 0.272367 (174794/641758)

Fourteen days price increasing percentages = 0.401065 (1265942/3156449)

- Result < -7.8125 = 0.246198 (137429/558206)
- Result >= -7.8125 and result < -1.048951 = 0.178851 (140479/785451)
- Result >= -1.048951 and result < 6.666667 = 0.721333 (841389/1166436)
- Result >= 6.666667 = 0.22688 (146645/646356)

Twenty one days price increasing percentages = 0.381783 (1205079/3156449)

- Result < -9.38248 = 0.249558 (145242/581997)
- Result >= -9.38248 and result < -1.376147 = 0.140037 (111304/794818)
- Result >= -1.376147 and result < 7.869742 = 0.70821 (789628/1114963)
- Result >= 7.869742 = 0.239073 (158905/664671)

Seven days price increasing percentages confusion matrix for second Bayesian Network (with set background knowledge)

	Result < -5.555555	Result >= -5.555555 and	Result >= -1.048951 and	Result >= 4.93421

		result < -0.7462686	result < 4.93421	
Result < -5.555555	156405	98507	246607	47594
Result >= -5.555555 and result < -0.7462686	83246	171889	425518	48929
Result >= -1.048951 and result < 4.93421	74706	154793	891285	115212
Result >= 4.93421	59786	72094	335084	174794

Fourteen days price increasing percentages confusion matrix for second Bayesian Network (with set background knowledge)

	Result < -7.8125	Result >= -7.8125 and result < -1.048951	Result >= -1.048951 and result < 6.666667	Result >= 6.666667
Result < -7.8125	137429	81691	284812	54274
Result >= -	92092	140479	492722	60158

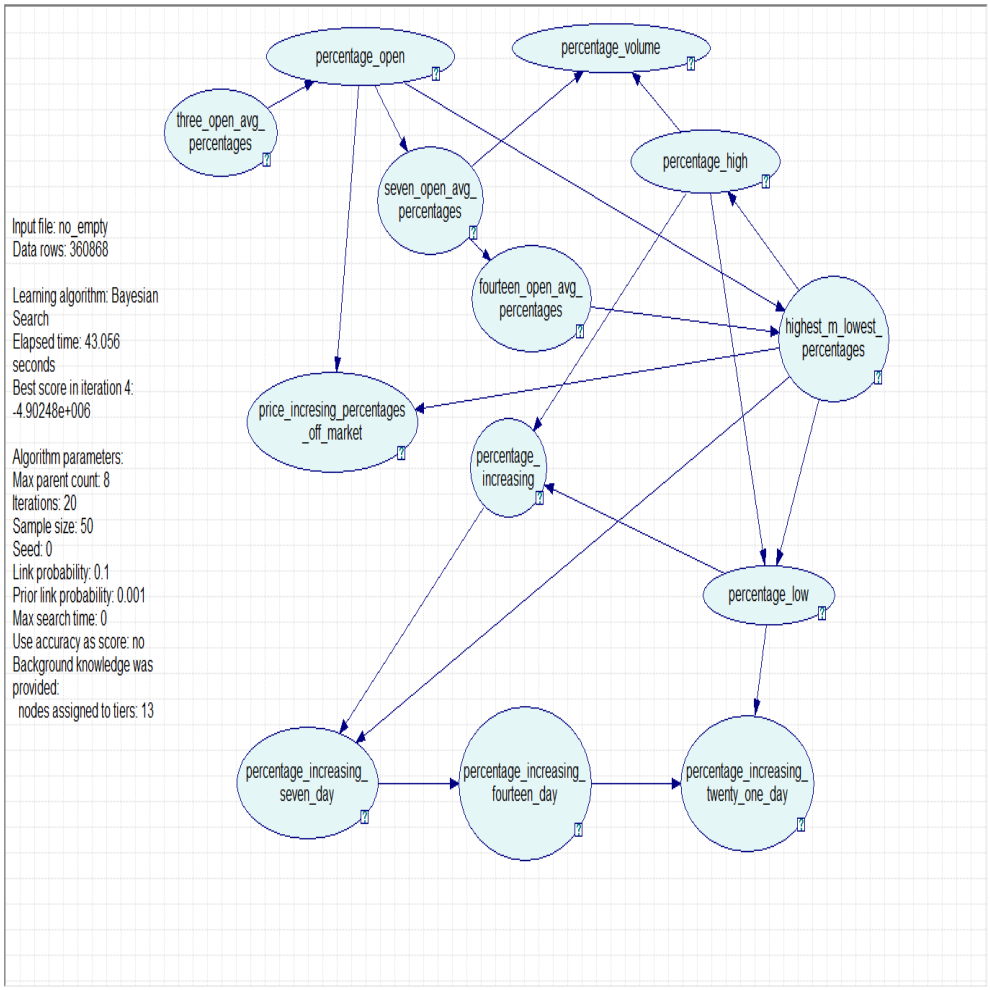
7.8125 and result < -1.048951				
Result >= -1.048951 and result < 6.666667	83692	128320	841389	113035
Result >= 6.666667	71919	65890	361902	146645

Twenty one days price increasing percentages confusion matrix for second Bayesian Network (with set background knowledge)

	Result < -9.38248	Result >= -9.38248 and result < -1.376147	Result >= -1.376147 and result < 7.869742	Result >= 7.869742
Result < -9.38248	145242	83995	267238	85522
Result >= -9.38248 and result < -1.376147	105643	111304	487576	90295
Result >= -1.376147 and	93519	101785	789628	130031

result < 7.869742				
Result >= 7.869742	85506	61521	358739	158905

Third Bayesian Network (with set background knowledge)



Result for third Bayesian Network (with set background knowledge)

Accuracy for all 3 nodes = 0.379519 (3593797/9469347)

Seven days price increasing percentages = 0.440069 (1389054/3156449)

- Result < -5.555555 = 0.256463 (140827/549113)
- Result >= -5.555555 and result < -0.7462686 = 0.153348 (111880/729582)
- Result >= -0.7462686 and result < 4.93421 = 0.784297 (969388/1235996)
- Result >= 4.93421 = 0.260159 (166959/641758)

Fourteen days price increasing percentages = 0.321292 (1014142/3156449)

- Result < -7.8125 = 0.286794 (160090/558206)
- Result >= -7.8125 and result < -1.048951 = 0.615825 (483700/785451)
- Result >= -1.048951 and result < 6.666667 = 0.195576 (228127/1166436)
- Result >= 6.666667 = 0.220041 (142225/646356)

Twenty one days price increasing percentages = 0.377196 (1190601/3156449)

- Result < -9.38248 = 0.271464 (157991/581997)
- Result >= -9.38248 and result < -1.376147 = 0.190929 (151754/794818)
- Result >= -1.376147 and result < 7.869742 = 0.704252 (785215/1114963)
- Result >= 7.869742 = 0.143892 (95641/664671)

Seven days price increasing percentages confusion matrix for third Bayesian Network (with set background knowledge)

	Result < -5.555555	Result >= -5.555555 and result < -0.7462686	Result >= -1.048951 and result < 4.93421	Result >= 4.93421
Result < -5.555555	140827	85075	281422	41789
Result >= -5.555555 and result < -0.7462686	66817	111880	505434	45451
Result >= -1.048951 and result < 4.93421	68244	90718	969388	107646
Result >= 4.93421	58456	50926	365417	166959

Fourteen days price increasing percentages confusion matrix for third Bayesian Network (with set background knowledge)

	Result < -7.8125	Result >= -7.8125 and result < -	Result >= -1.048951 and result <	Result >= 6.666667

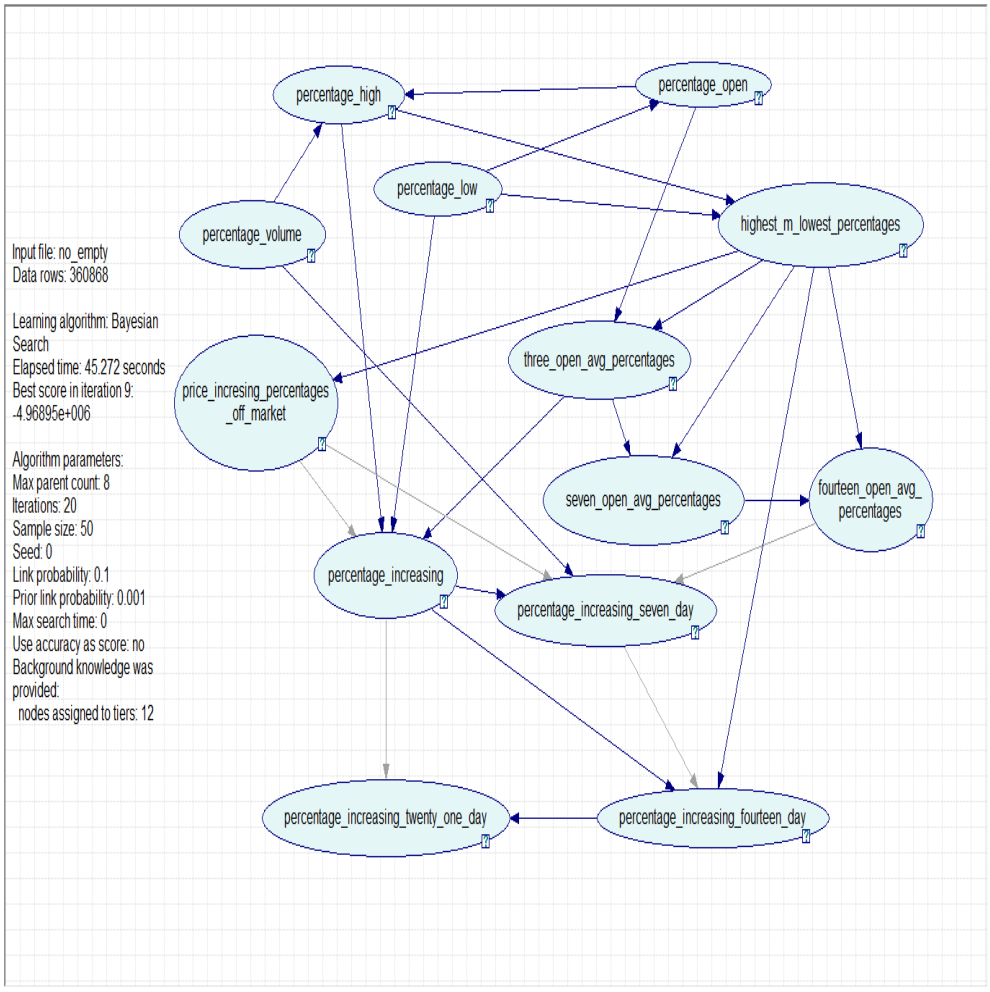
		1.048951	6.666667	
Result < -7.8125	160090	296691	50504	50921
Result >= -7.8125 and result < -1.048951	119293	483700	124380	58078
Result >= -1.048951 and result < 6.666667	101886	725802	228127	110621
Result >= 6.666667	82826	309521	111784	142225

Twenty one days price increasing percentages confusion matrix for third Bayesian Network (with set background knowledge)

	Result < -9.38248	Result >= -9.38248 and result < -1.376147	Result >= -1.376147 and result < 7.869742	Result >= 7.869742
Result < -9.38248	157991	86464	297894	39648
Result >= -9.38248 and	123876	151754	476766	42422

result < -1.376147				
Result >= -1.376147 and result < 7.869742	103714	155556	785215	70478
Result >= 7.869742	88638	84682	395710	95641

Fourth Bayesian Network (with set background knowledge)



Result for fourth Bayesian Network (with set background knowledge)

Accuracy for all 3 nodes = 0.318566 (3016611/9469347)

Seven days price increasing percentages = 0.173965 (549113/3156449)

- Result < -5.555555 = 1 (549113/549113)
- Result >= -5.555555 and result < -0.7462686 = 0 (0/729582)
- Result >= -0.7462686 and result < 4.93421 = 0 (0/1235996)
- Result >= 4.93421 = 0 (0/641758)

Fourteen days price increasing percentages = 0.401766 (1268155/3156449)

- Result < -7.8125 = 0.22625 (126294/558206)
- Result >= -7.8125 and result < -1.048951 = 0.139172 (109313/785451)
- Result >= -1.048951 and result < 6.666667 = 0.765021 (892348/1166436)
- Result >= 6.666667 = 0.216908 (140200/646356)

Twenty one days price increasing percentages = 0.379966 (1199343/3156449)

- Result < -9.38248 = 0.269665 (156944/581997)
- Result >= -9.38248 and result < -1.376147 = 0.0815784 (64840/794818)
- Result >= -1.376147 and result < 7.869742 = 0.758643 (845859/1114963)
- Result >= 7.869742 = 0.198143 (131700/664671)

Seven days price increasing percentages confusion matrix for fourth Bayesian Network (with set background knowledge)

	Result < -5.555555	Result >= -5.555555 and result < -0.7462686	Result >= -1.048951 and result < 4.93421	Result >= 4.93421
Result < -5.555555	549113	0	0	0
Result >= -5.555555 and result < -0.7462686	729582	0	0	0
Result >= -1.048951 and result < 4.93421	1235996	0	0	0
Result >= 4.93421	641758	0	0	0

Fourteen days price increasing percentages confusion matrix for fourth Bayesian Network (with set background knowledge)

	Result < -7.8125	Result >= -7.8125 and result < -	Result >= -1.048951 and result <	Result >= 6.666667

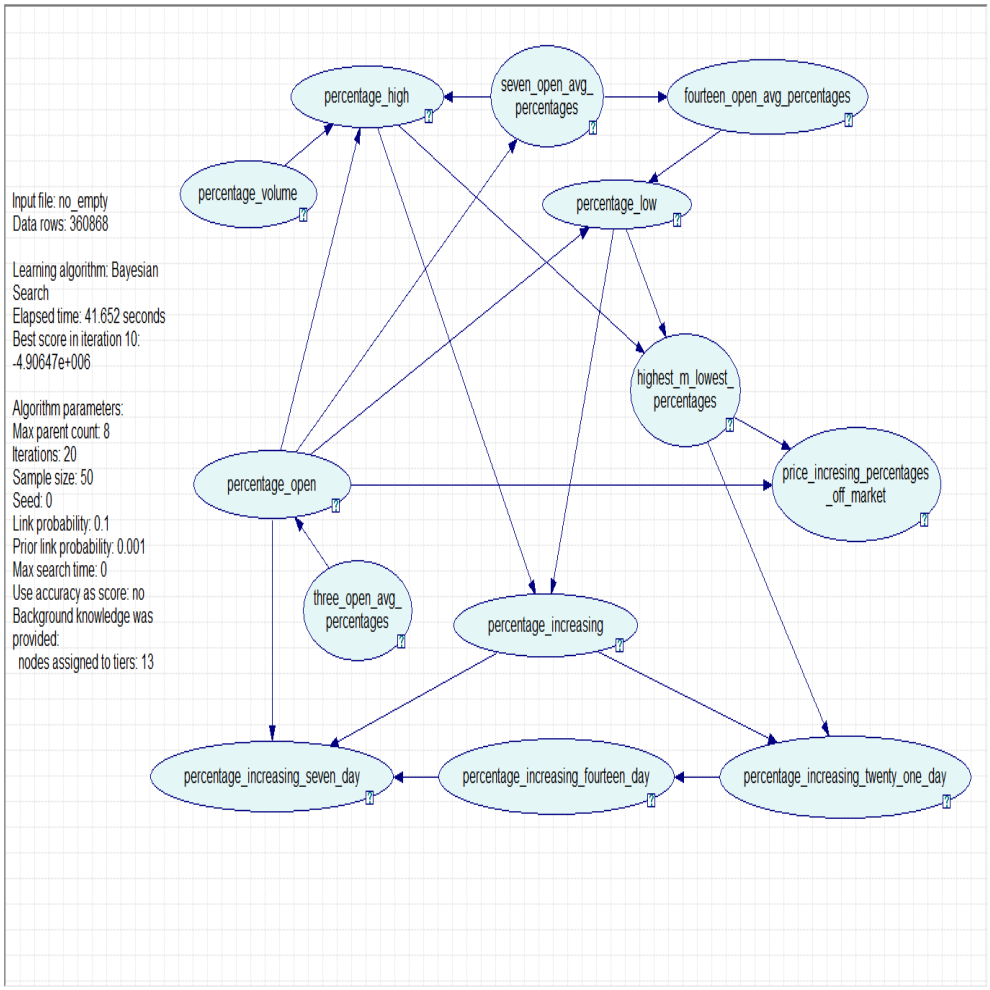
		1.048951	6.666667	
Result < -7.8125	126294	79382	303325	49205
Result >= -7.8125 and result < -1.048951	78222	109313	541503	56413
Result >= -1.048951 and result < 6.666667	72558	93017	892348	108513
Result >= 6.666667	64784	56887	384485	140200

Twenty one days price increasing percentages confusion matrix for fourth Bayesian Network (with set background knowledge)

	Result < -9.38248	Result >= -9.38248 and result < -1.376147	Result >= -1.376147 and result < 7.869742	Result >= 7.869742
Result < -9.38248	156944	45132	325363	54558
Result >= -9.38248 and	121401	64840	546302	62275

result < -1.376147				
Result >= -1.376147 and result < 7.869742	103361	59945	845859	105789
Result >= 7.869742	89903	38931	404137	131700

Fifth Bayesian Network (with set background knowledge)



Result for fifth Bayesian Network (with set background knowledge)

Accuracy for all 3 nodes = 0.397319 (3762349/9469347)

Seven days price increasing percentages = 0.410651 (1296200/3156449)

- Result < -5.555555 = 0.34072 (187094/549113)
- Result >= -5.555555 and result < -0.7462686 = 0.274234 (200076/729582)
- Result >= -0.7462686 and result < 4.93421 = 0.566085 (699679/1235996)
- Result >= 4.93421 = 0.326215 (209351/641758)

Fourteen days price increasing percentages = 0.399478 (1260931/3156449)

- Result < -7.8125 = 0.289868 (161806/558206)
- Result >= -7.8125 and result < -1.048951 = 0.0847628 (66577/785451)
- Result >= -1.048951 and result < 6.666667 = 0.765021 (892348/1166436)
- Result >= 6.666667 = 0.216908 (140200/646356)

Twenty one days price increasing percentages = 0.381827 (1205218/3156449)

- Result < -9.38248 = 0.209266 (121792/581997)
- Result >= -9.38248 and result < -1.376147 = 0.133197 (105867/794818)
- Result >= -1.376147 and result < 7.869742 = 0.758643 (845859/1114963)

- Result >= 7.869742 = 0.198143 (131700/664671)

Seven days price increasing percentages confusion matrix for fifth Bayesian Network (with set background knowledge)

	Result < -5.555555	Result >= -5.555555 and result < -0.7462686	Result >= -1.048951 and result < 4.93421	Result >= 4.93421
Result < -5.555555	187094	125918	183030	53071
Result >= -5.555555 and result < -0.7462686	131569	200076	326412	71525
Result >= -1.048951 and result < 4.93421	104220	260613	699679	171484
Result >= 4.93421	74777	133216	224414	209351

Fourteen days price increasing percentages confusion matrix for fifth Bayesian Network (with set background knowledge)

	Result < -	Result >= -	Result >= -	Result >=

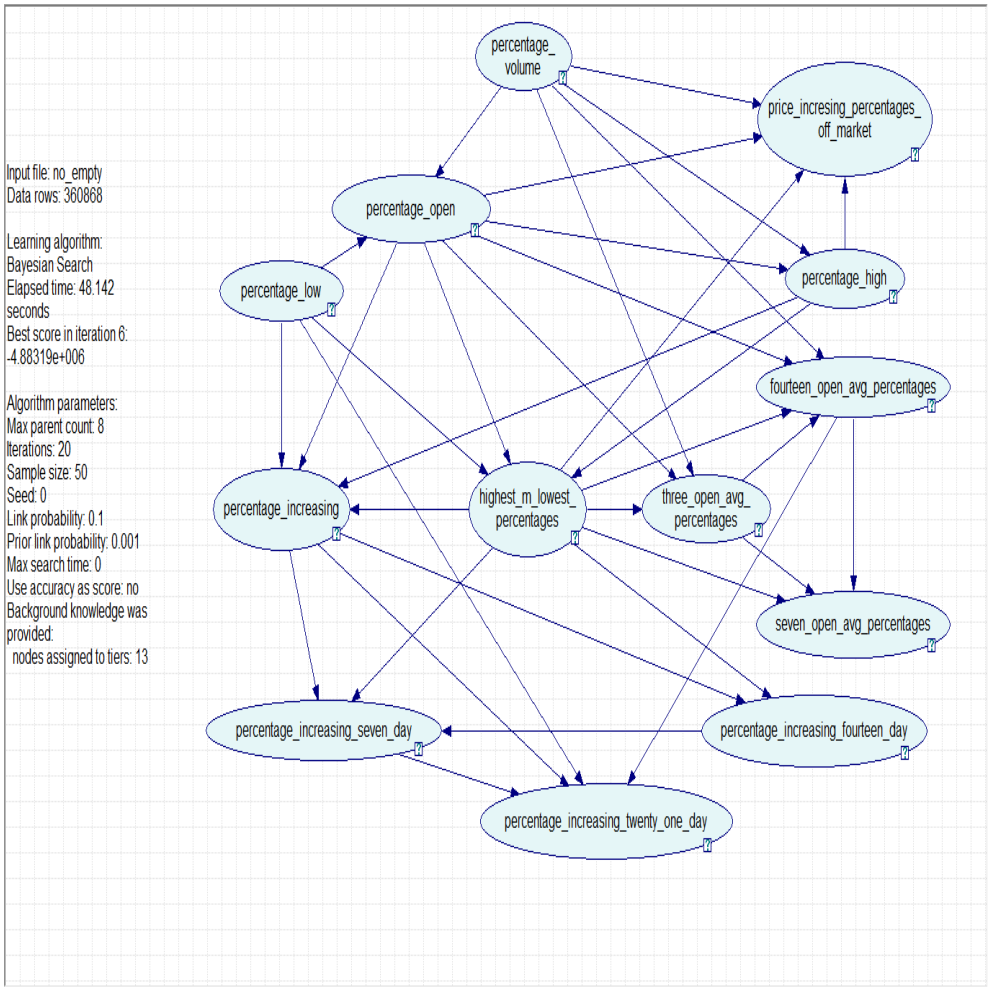
	7.8125	7.8125 and result < -1.048951	1.048951 and result < 6.666667	6.666667
Result < -7.8125	161806	43870	303325	49205
Result >= -7.8125 and result < -1.048951	120958	66577	541503	56413
Result >= -1.048951 and result < 6.666667	103994	61581	892348	108513
Result >= 6.666667	84851	36820	384485	140200

Twenty one days price increasing percentages confusion matrix for fifth Bayesian Network (with set background knowledge)

	Result < -9.38248	Result >= -9.38248 and result < -1.376147	Result >= -1.376147 and result < 7.869742	Result >= 7.869742
Result < -9.38248	121792	80284	325363	54558

Result >= -9.38248 and result < -1.376147	80374	105867	546302	32275
Result >= -1.376147 and result < 7.869742	71434	91872	845859	105798
Result >= 7.869742	68258	60576	404137	131700

Sixth Bayesian Network (with set background knowledge)



Result for sixth Bayesian Network (with set background knowledge)

Accuracy for all 3 nodes = 0.407841 (3861987/9469347)

Seven days price increasing percentages = 0.440069 (1389054/3156449)

- Result < -5.555555 = 0.256463 (140827/549113)
- Result >= -5.555555 and result < -0.7462686 = 0.153348 (111880/729582)

- Result >= -0.7462686 and result < 4.93421 = 0.784297 (969388/1235996)
- Result >= 4.93421 = 0.260159 (166959/641758)

Fourteen days price increasing percentages = 0.401766 (1268155/3156449)

- Result < -7.8125 = 0.22625 (126294/558206)
- Result >= -7.8125 and result < -1.048951 = 0.139172 (109313/785451)
- Result >= -1.048951 and result < 6.666667 = 0.765021 (892348/1166436)
- Result >= 6.666667 = 0.216908 (140200/646356)

Twenty one days price increasing percentages = 0.381688 (1204778/3156449)

- Result < -9.38248 = 0.26546 (154497/581997)
- Result >= -9.38248 and result < -1.376147 = 0.177389 (140992/794818)
- Result >= -1.376147 and result < 7.869742 = 0.682187 (760613/1114963)
- Result >= 7.869742 = 0.223684 (148676/664671)

Seven days price increasing percentages confusion matrix for sixth Bayesian Network (with set background knowledge)

	Result < -5.555555	Result >= -5.555555 and result < -0.7462686	Result >= -1.048951 and result < 4.93421	Result >= 4.93421

Result < -5.555555	140827	85075	281422	41789
Result >= -5.555555 and result < -0.7462686	66817	111880	505434	45451
Result >= -1.048951 and result < 4.93421	68244	90718	969388	107646
Result >= 4.93421	58456	50926	365417	166959

Fourteen days price increasing percentages confusion matrix for sixth Bayesian Network (with set background knowledge)

	Result < -7.8125	Result >= -7.8125 and result < -1.048951	Result >= -1.048951 and result < 6.666667	Result >= 6.666667
Result < -7.8125	126294	79382	303325	49205
Result >= -7.8125 and result < -	78222	109313	541503	56413

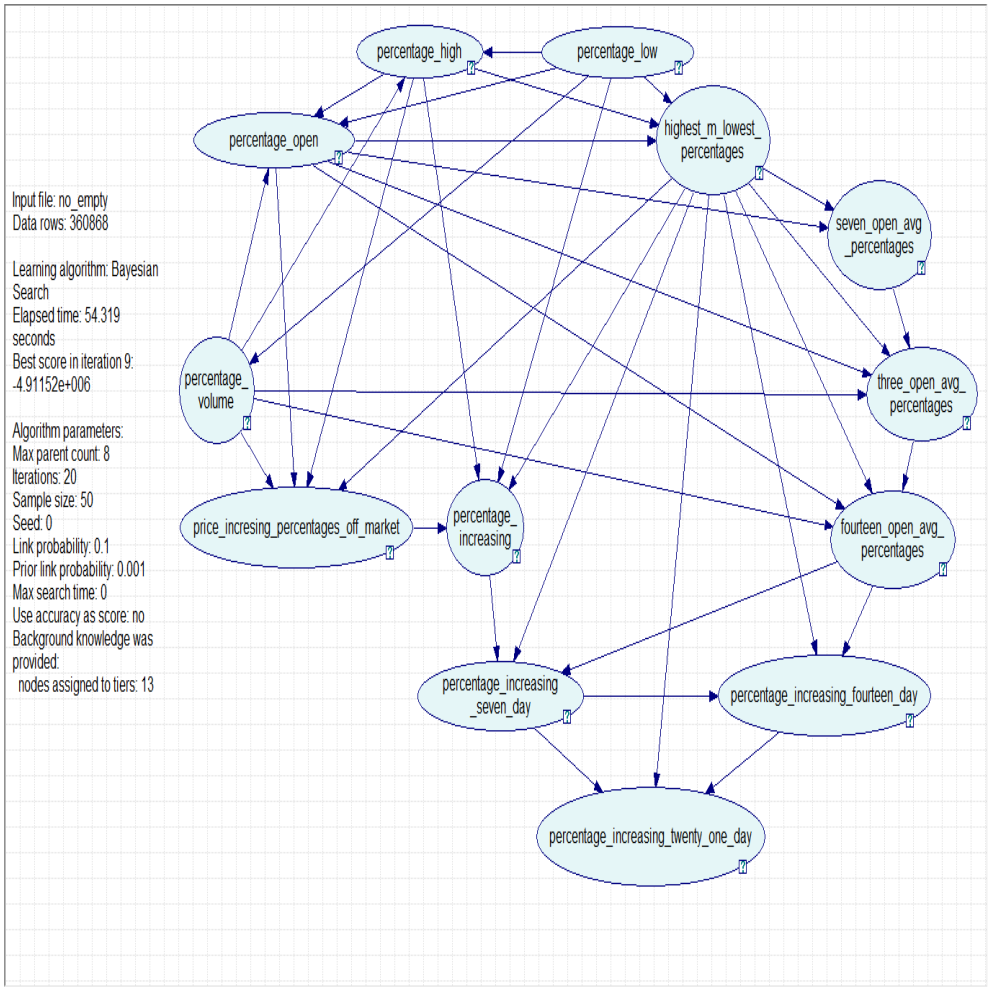
1.048951				
Result >= -1.048951 and result < 6.666667	72558	93017	892348	108513
Result >= 6.666667	64784	56887	384485	140200

Twenty one days price increasing percentages confusion matrix for sixth Bayesian Network (with set background knowledge)

	Result < -9.38248	Result >= -9.38248 and result < -1.376147	Result >= -1.376147 and result < 7.869742	Result >= 7.869742
Result < -9.38248	154497	104958	260100	62442
Result >= -9.38248 and result < -1.376147	112406	140992	467404	74016
Result >= -1.376147 and result < 7.869742	96013	139618	760613	118719

Result >= 7.869742	86139	87689	342167	148676

Seventh Bayesian Network (with set background knowledge)



Result for seventh Bayesian Network (with set background knowledge)

Accuracy for all 3 nodes = 0.401505 (3801986/9469347)

Seven days price increasing percentages = 0.42026 (1326530/3156449)

- Result < -5.555555 = 0.310362 (170424/549113)

- Result >= -5.555555 and result < -0.7462686 = 0.224657 (163906/729582)
- Result >= -0.7462686 and result < 4.93421 = 0.673001 (831826/1235996)
- Result >= 4.93421 = 0.249898 (160374/641758)

Fourteen days price increasing percentages = 0.402264 (1269727/3156449)

- Result < -7.8125 = 0.273736 (152801/558206)
- Result >= -7.8125 and result < -1.048951 = 0.145404 (114208/785451)
- Result >= -1.048951 and result < 6.666667 = 0.731754 (853544/1166436)
- Result >= 6.666667 = 0.230792 (149174/646356)

Twenty one days price increasing percentages = 0.381989 (1205729/3156449)

- Result < -9.38248 = 0.253041 (147269/581997)
- Result >= -9.38248 and result < -1.376147 = 0.103093 (81940/794818)
- Result >= -1.376147 and result < 7.869742 = 0.742651 (828028/1114963)
- Result >= 7.869742 = 0.223407 (148492/664671)

Seven days price increasing percentages confusion matrix for seventh **Bayesian Network (with set background knowledge)**

	Result < -	Result >= -	Result >= -	Result >=

	5.555555	5.555555 and result < -0.7462686	1.048951 and result < 4.93421	4.93421
Result < -5.555555	170424	155205	181695	41789
Result >= -5.555555 and result < -0.7462686	90885	163906	431751	43040
Result >= -1.048951 and result < 4.93421	87506	220917	831826	95747
Result >= 4.93421	75628	133678	272078	160374

Fourteen days price increasing percentages confusion matrix for seventh **Bayesian Network (with set background knowledge)**

	Result < -7.8125	Result >= -7.8125 and result < -1.048951	Result >= -1.048951 and result < 6.666667	Result >= 6.666667
Result < -7.8125	152801	97605	255112	52688

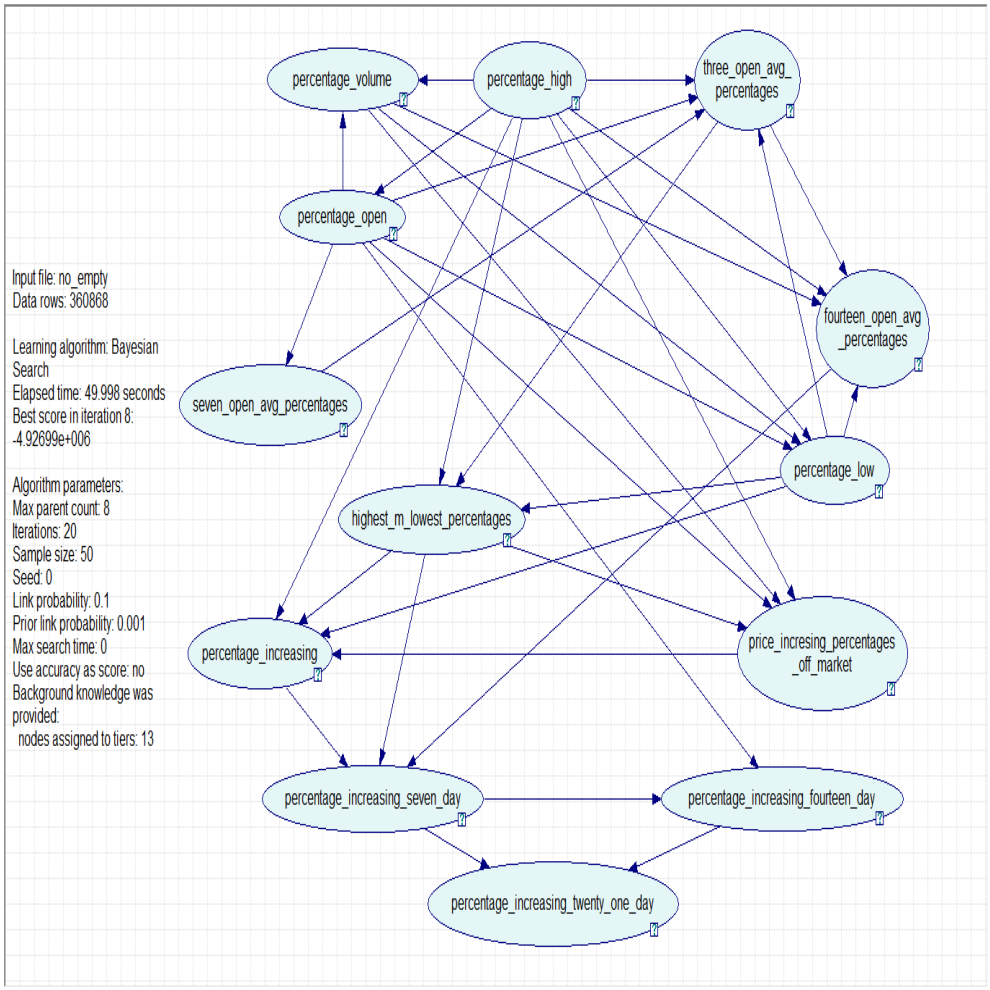
Result >= -7.8125 and result < -1.048951	102814	114208	507197	61232
Result >= -1.048951 and result < 6.666667	94997	107561	853544	110334
Result >= 6.666667	83523	65562	348097	149174

Twenty one days price increasing percentages confusion matrix for seventh **Bayesian Network (with set background knowledge)**

	Result < -9.38248	Result >= -9.38248 and result < -1.376147	Result >= -1.376147 and result < 7.869742	Result >= 7.869742
Result < -9.38248	147269	55041	318579	61108
Result >= -9.38248 and result < -1.376147	104935	81940	534434	73509
Result >= -	93595	70643	828028	122697

1.376147 and result < 7.869742				
Result >= 7.869742	88057	40894	387228	148492

Eighth Bayesian Network (with set background knowledge)



Result for eighth Bayesian Network (with set background knowledge)

Accuracy for all 3 nodes = 0.395603 (3746099/9469347)

Seven days price increasing percentages = 0.420332 (1326755/3156449)

- Result < -5.555555 = 0.310652 (170583/549113)
- Result >= -5.555555 and result < -0.7462686 = 0.224526 (163810/729582)
- Result >= -0.7462686 and result < 4.93421 = 0.673129 (831985/1235996)
- Result >= 4.93421 = 0.249903 (160377/641758)

Fourteen days price increasing percentages = 0.391324 (1235194/3156449)

- Result < -7.8125 = 0.311779 (174037/558206)
- Result >= -7.8125 and result < -1.048951 = 0.187668 (147404/785451)
- Result >= -1.048951 and result < 6.666667 = 0.650182 (758396/1166436)
- Result >= 6.666667 = 0.240358 (155357/646356)

Twenty one days price increasing percentages = 0.375153 (1184150/3156449)

- Result < -9.38248 = 0.314904 (183273/581997)
- Result >= -9.38248 and result < -1.376147 = 0.201563 (160206/794818)
- Result >= -1.376147 and result < 7.869742 = 0.619767 (691017/1114963)
- Result >= 7.869742 = 0.225155 (149654/664671)

Seven days price increasing percentages confusion matrix for eighth Bayesian Network (with set background knowledge)

	Result < -5.555555	Result >= -5.555555 and result < -0.7462686	Result >= -1.048951 and result < 4.93421	Result >= 4.93421
Result < -5.555555	170583	154901	181839	41790
Result >= -5.555555 and result < -0.7462686	90952	163810	431779	40341
Result >= -1.048951 and result < 4.93421	87562	220703	831985	95746
Result >= 4.93421	75674	133488	272219	160377

Fourteen days price increasing percentages confusion matrix for eighth Bayesian Network (with set background knowledge)

	Result < -7.8125	Result >= -7.8125 and result < -	Result >= -1.048951 and result <	Result >= 6.666667

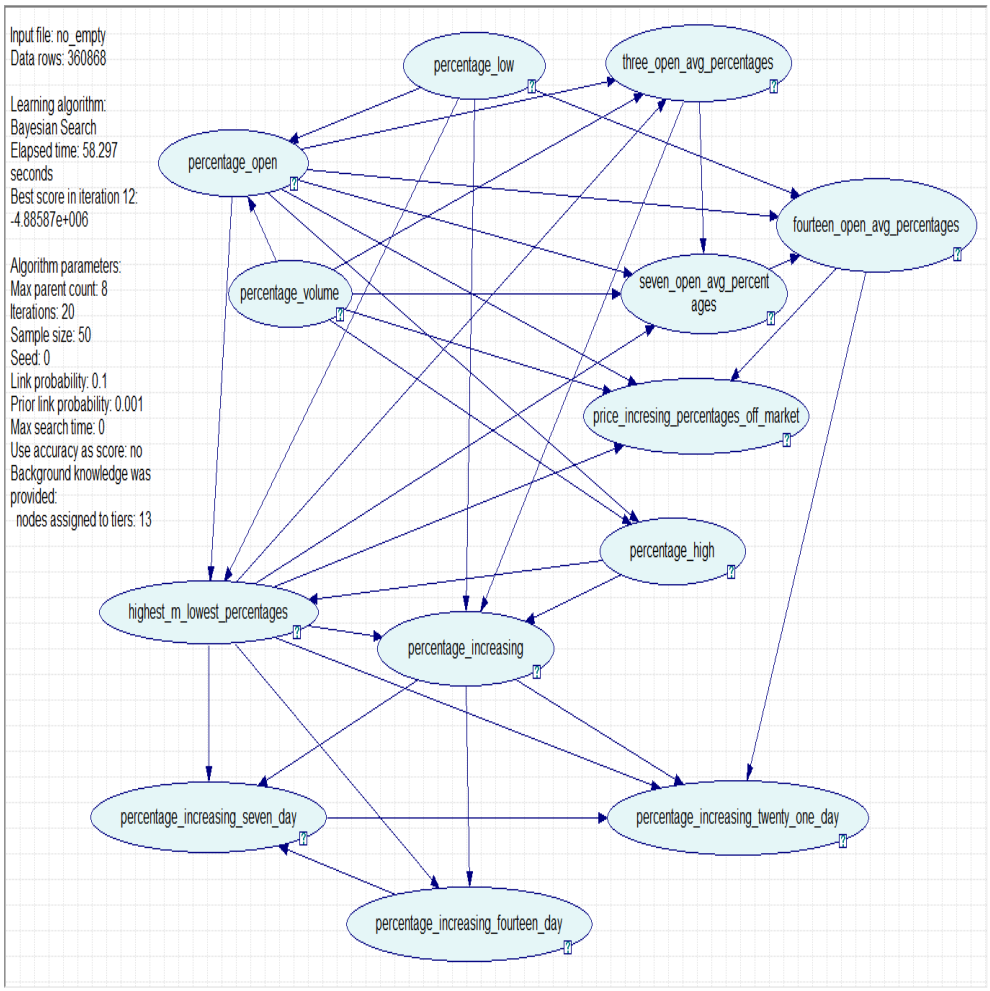
		1.048951	6.666667	
Result < -7.8125	174037	106426	208698	69045
Result >= -7.8125 and result < -1.048951	124487	147404	440441	73119
Result >= -1.048951 and result < 6.666667	122572	162807	758396	122661
Result >= 6.666667	111252	102269	277478	155357

Twenty one days price increasing percentages confusion matrix for eighth Bayesian Network (with set background knowledge)

	Result < -9.38248	Result >= -9.38248 and result < -1.376147	Result >= -1.376147 and result < 7.869742	Result >= 7.869742
Result < -9.38248	183273	136281	198849	63594
Result >= -9.38248 and	134665	160206	424454	75493

result < -1.376147				
Result >= -1.376147 and result < 7.869742	127542	174878	691017	121526
Result >= 7.869742	122554	121877	270586	149654

Ninth Bayesian Network (with set background knowledge)



Result for the ninth Bayesian Network (with set background knowledge)

Accuracy for all 3 nodes = 0.408617 (3869333/9469347)

Seven days price increasing percentages = 0.440069 (1389054/3156449)

- Result < -5.555555 = 0.256463 (140827/549113)
- Result >= -5.555555 and result < -0.7462686 = 0.153348 (111880/729582)
- Result >= -0.7462686 and result < 4.93421 = 0.784297 (969388/1235996)
- Result >= 4.93421 = 0.260159 (166959/641758)

Fourteen days price increasing percentages = 0.401766 (1268155/3156449)

- Result < -7.8125 = 0.22625 (126294/558206)
- Result >= -7.8125 and result < -1.048951 = 0.139172 (109313/785451)
- Result >= -1.048951 and result < 6.666667 = 0.765021 (892348/1166436)
- Result >= 6.666667 = 0.216908 (140200/646356)

Twenty one days price increasing percentages = 0.384015 (1212124/3156449)

- Result < -9.38248 = 0.250749 (145935/581997)
- Result >= -9.38248 and result < -1.376147 = 0.14746 (117204/794818)
- Result >= -1.376147 and result < 7.869742 = 0.738941 (823892/1114963)

- Result >= 7.869742 = 0.188203 (125093/664671)

Seven days price increasing percentages confusion matrix for the ninth Bayesian Network (with set background knowledge)

	Result < -5.555555	Result >= -5.555555 and result < -0.7462686	Result >= -1.048951 and result < 4.93421	Result >= 4.93421
Result < -5.555555	140827	85075	281422	41789
Result >= -5.555555 and result < -0.7462686	66817	111880	505434	45451
Result >= -1.048951 and result < 4.93421	68244	90718	969388	107646
Result >= 4.93421	58456	50926	365417	166959

Fourteen days price increasing percentages confusion matrix for the ninth Bayesian Network (with set background knowledge)

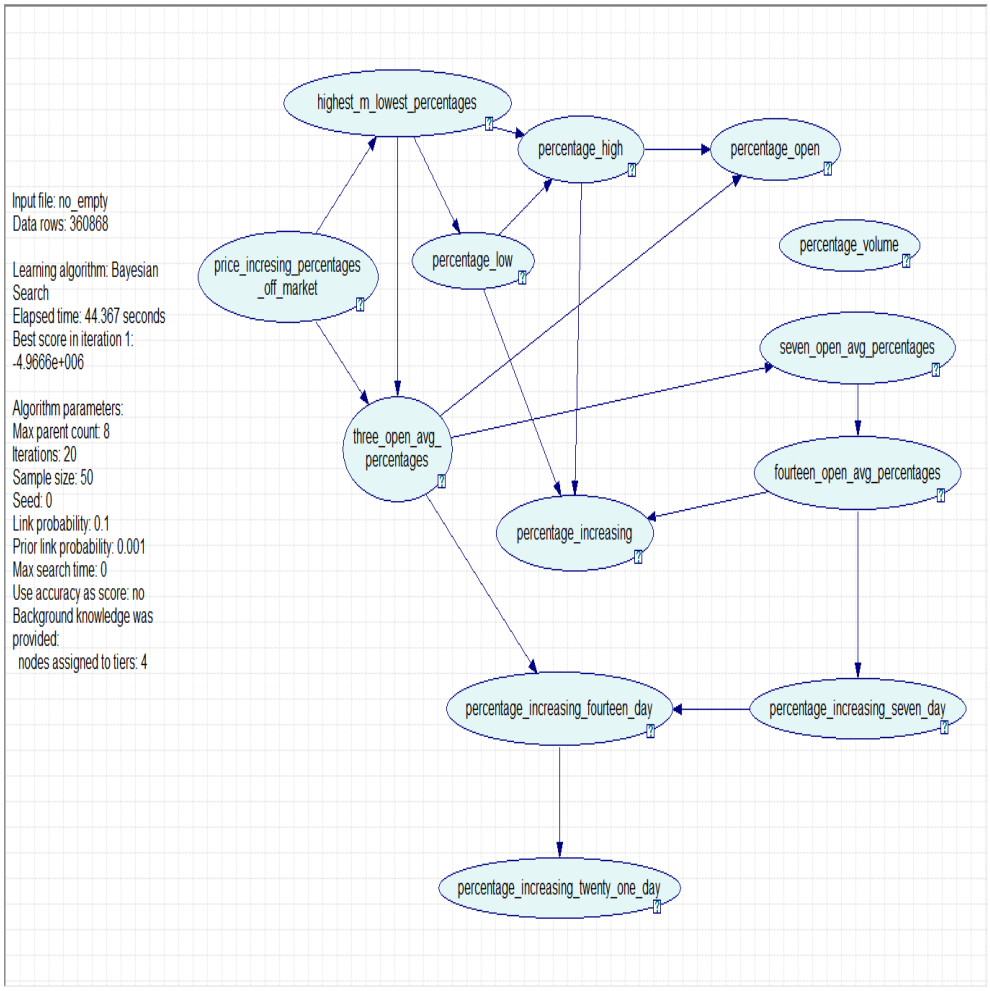
	Result < -7.8125	Result >= -7.8125 and result < -1.048951	Result >= -1.048951 and result < 6.666667	Result >= 6.666667
Result < -7.8125	126294	79382	303325	49205
Result >= -7.8125 and result < -1.048951	78222	109313	541503	56413
Result >= -1.048951 and result < 6.666667	72558	93017	892348	108513
Result >= 6.666667	64784	56887	384485	140200

Twenty one days price increasing percentages confusion matrix for the ninth Bayesian Network (with set background knowledge)

	Result < -9.38248	Result >= -9.38248 and result < -	Result >= -1.376147 and result <	Result >= 7.869742

		1.376147	7.869742	
Result < -9.38248	145935	103138	282438	50486
Result >= -9.38248 and result < -1.376147	103077	117204	519968	54569
Result >= -1.376147 and result < 7.869742	93536	107651	823892	89884
Result >= 7.869742	86078	75388	378112	125093

Tenth Bayesian Network (without set background knowledge)



Result for tenth Bayesian Network (with set background knowledge)

Accuracy for all 3 nodes = 0.363368 (3440861/9469347)

Seven days price increasing percentages = 0.367827 (1161028/3156449)

- Result < -5.555555 = 0.231479 (127108/549113)
- Result >= -5.555555 and result < -0.7462686 = 0.332386 (242503/729582)

- Result >= -0.7462686 and result < 4.93421 = 0.479312 (592428/1235996)
- Result >= 4.93421 = 0.310069 (198989/641758)

Fourteen days price increasing percentages = 0.367706 (1160646/3156449)

- Result < -7.8125 = 0.339131 (189305/558206)
- Result >= -7.8125 and result < -1.048951 = 5.60188e-005 (44/785451)
- Result >= -1.048951 and result < 6.666667 = 0.782423 (912646/1166436)
- Result >= 6.666667 = 0.090741 (58651/646356)

Twenty one days price increasing percentages = 0.354572 (1119187/3156449)

- Result < -9.38248 = 0.342108 (199106/581997)
- Result >= -9.38248 and result < -1.376147 = 1.25815e-005 (10/794818)
- Result >= -1.376147 and result < 7.869742 = 0.780324 (870032/1114963)
- Result >= 7.869742 = 0.0752839 (50039/664671)

Seven days price increasing percentages confusion matrix for tenth Bayesian Network (with set background knowledge)

	Result < -5.555555	Result >= -5.555555 and result < -0.7462686	Result >= -1.048951 and result < 4.93421	Result >= 4.93421

Result < -5.555555	127108	124676	155158	142171
Result >= -5.555555 and result < -0.7462686	69949	242503	300096	117034
Result >= -1.048951 and result < 4.93421	97341	332220	592428	214007
Result >= 4.93421	95428	157594	189747	198989

Fourteen days price increasing percentages confusion matrix for tenth Bayesian Network (with set background knowledge)

	Result < -7.8125	Result >= -7.8125 and result < -1.048951	Result >= -1.048951 and result < 6.666667	Result >= 6.666667
Result < -7.8125	189305	56	309188	59657
Result >= -7.8125 and result < -	136735	44	578799	69873

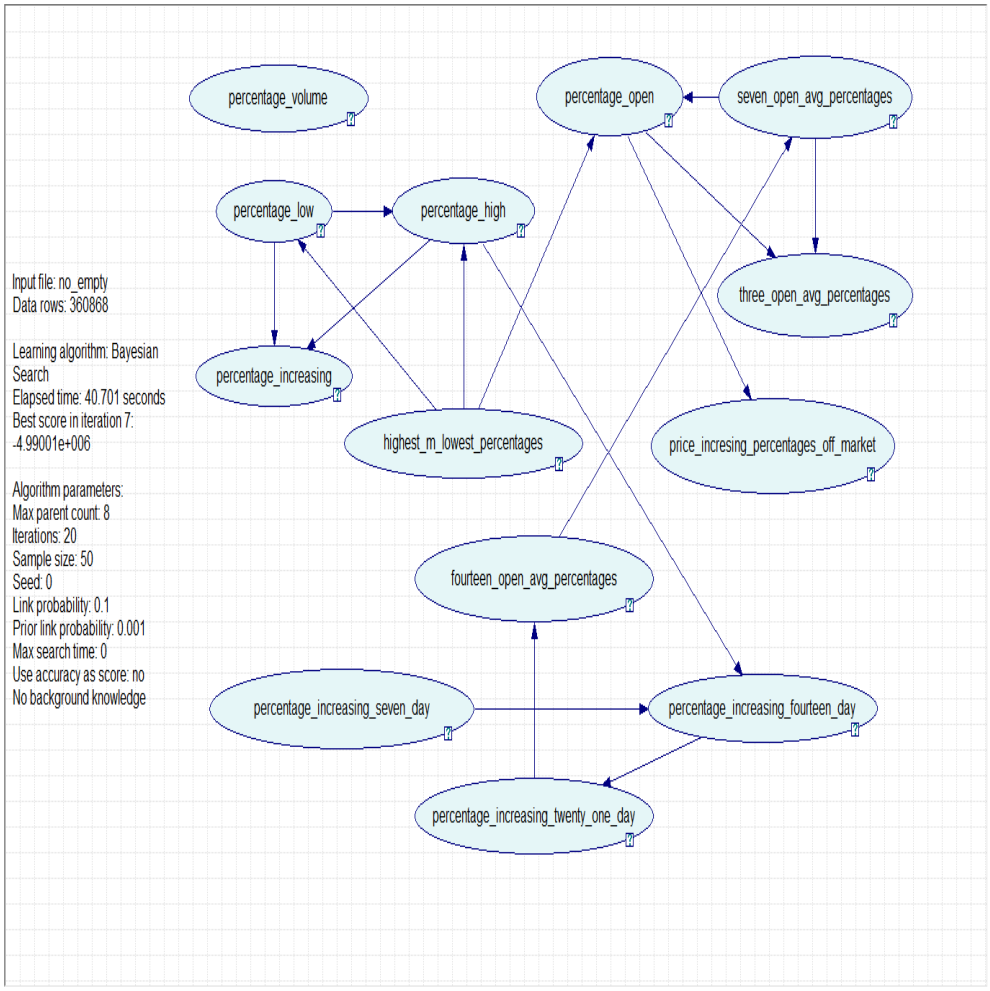
1.048951				
Result >= -1.048951 and result < 6.666667	181339	50	912646	72401
Result >= 6.666667	198722	42	388941	58651

Twenty one days price increasing percentages confusion matrix for tenth Bayesian Network (with set background knowledge)

	Result < -9.38248	Result >= -9.38248 and result < -1.376147	Result >= -1.376147 and result < 7.869742	Result >= 7.869742
Result < -9.38248	199106	21	332529	50341
Result >= -9.38248 and result < -1.376147	149116	10	580539	65153
Result >= -1.376147 and result < 7.869742	181895	12	870032	63024

Result >= 7.869742	208031	29	406572	50039

Eleventh Bayesian Network (without set background knowledge)



Result for eleventh Bayesian Network (with set background knowledge)

Accuracy for all 3 nodes = 0.362697 (3434508/9469347)

Seven days price increasing percentages = 0.378014 (1193183/3156449)

- Result < -5.555555 = 0.488763 (268386/549113)

- Result >= -5.555555 and result < -0.7462686 = 0 (0/729582)
- Result >= -0.7462686 and result < 4.93421 = 0.748068 (924609/1235996)
- Result >= 4.93421 = 0.000292945 (188/641758)

Fourteen days price increasing percentages = 0.37859 (1195000/3156449)

- Result < -7.8125 = 0.256196 (143010/558206)
- Result >= -7.8125 and result < -1.048951 = 0.000655674 (515/785451)
- Result >= -1.048951 and result < 6.666667 = 0.785717 (916489/1166436)
- Result >= 6.666667 = 0.208842 (134986/646356)

Twenty one days price increasing percentages = 0.331488 (1046325/3156449)

- Result < -9.38248 = 0.46458 (270384/581997)
- Result >= -9.38248 and result < -1.376147 = 0.314238 (249762/794818)
- Result >= -1.376147 and result < 7.869742 = 0.471227 (525401/1114963)
- Result >= 7.869742 = 0.0011705 (778/664671)

Seven days price increasing percentages confusion matrix for eleventh Bayesian Network (with set background knowledge)

	Result < -5.555555	Result >= -5.555555 and	Result >= -1.048951 and	Result >= 4.93421

		result < -0.7462686	result < 4.93421	
Result < -5.555555	2683386	0	280498	229
Result >= -5.555555 and result < -0.7462686	186818	0	542613	151
Result >= -1.048951 and result < 4.93421	311147	0	924609	240
Result >= 4.93421	293971	0	347599	188

Fourteen days price increasing percentages confusion matrix for eleventh Bayesian Network (with set background knowledge)

	Result < -7.8125	Result >= -7.8125 and result < -1.048951	Result >= -1.048951 and result < 6.666667	Result >= 6.666667
Result < -7.8125	143010	1027	364412	49757
Result >= -	109393	515	615751	59792

7.8125 and result < -1.048951				
Result >= -1.048951 and result < 6.666667	129589	646	916489	119712
Result >= 6.666667	125187	769	385414	134986

Twenty one days price increasing percentages confusion matrix for eleventh Bayesian Network (with set background knowledge)

	Result < -9.38248	Result >= -9.38248 and result < -1.376147	Result >= -1.376147 and result < 7.869742	Result >= 7.869742
Result < -9.38248	270384	124629	186286	698
Result >= -9.38248 and result < -1.376147	219434	249762	325315	307
Result >= -1.376147 and	280255	308909	525401	398

result < 7.869742				
Result >= 7.869742	286302	172606	204985	778

APPENDIX E

Visual C++ Code

As an example:

```
Char const * constfile Name=
"C:\\Users\\Asus\\Desktop\\KLSE\\klse\\network_consume\\network
_consume_3.txt";
```

Above code used to retrieve the process data.

```
char const* const networkName =
"C:\\Users\\Asus\\Desktop\\KLSE\\klse\\network_with_set_backgro
und_knowledge.xdsl";
```

Above code call the Bayesian Network

```
int int_open;
int int_highest;
int int_lowest;
int int_close;
int int_volume;
```

Above code was part of the code that use to declare the variable in Visual C++

```
int node_open     = theNet.FindNode("percentage_open");
int node_highest = theNet.FindNode("percentage_high");
int node_lowest = theNet.FindNode("percentage_low");
int node_volume = theNet.FindNode("percentage_volume");
```

Above code is get node data from Bayesian Network and then assign it into a variable in Visual C++.

```
while (pch != NULL){
      if(dc == 1){
            int_open = atof(pch);
      }else if(dc == 2){
            int_highest = atof(pch);
      }else if(dc == 3){
            int_lowest = atof(pch);
      …

      pch = strtok (NULL, ",");
      dc++;
}
```

When loop the process data, the process data will be split by using comma as the split symbol in the processed data files, so that the data will be separated by category and use to process the result in Bayesian Network.

```
theNet.GetNode(node_open)->Value()->SetEvidence(int_open);
theNet.GetNode(node_highest)->Value()->SetEvidence(int_highest);
theNet.GetNode(node_lowest)->Value()->SetEvidence(int_lowest);
```

Above coding was assigned in to the node of Bayesian Network to produce results.

```
DSL_sysCoordinates
theCoordinates_seven_day(*theNet.GetNode(node_price_incresing_s
even_day)->Value());
theCoordinates_seven_day.GoToCurrentPosition();
result1_seven_day = theCoordinates_seven_day.UncheckedValue();
theCoordinates_seven_day.Next();
result2_seven_day = theCoordinates_seven_day.UncheckedValue();
theCoordinates_seven_day.Next();
result3_seven_day = theCoordinates_seven_day.UncheckedValue();
```

```
theCoordinates_seven_day.Next();
result4_seven_day = theCoordinates_seven_day.UncheckedValue();
```

Above code is the code that is used to retrieve results that have been return from Bayesian Network, after all those Bayesian Network node is being set value for them.

```
result1_seven_day = theCoordinates_seven_day.UncheckedValue();
```

This is retrieving the first range percentages that return by Bayesian Network.

```
theCoordinates_seven_day.Next();
```

Above code is use to check the next results that return from Bayesian Network.

```
if(float_result_seven_day < -5.555555){
      if(result1_seven_day      >      result2_seven_day      &&
result1_seven_day  >  result3_seven_day  &&  result1_seven_day  >
result4_seven_day ){
            accuracy_count_seven_day++;
            total_accuracy_count_seven_day                    =
            total_accuracy_count_seven_day + result1_seven_day;
      }
}
```

Above code is used to calculate the accuracy data for the results that have been generated by Bayesian Network. Those results will be displayed out for checking purposes. That is another reason to create this testing module in Visual C++ which is because to test the accuracy of the Bayesian Network, does the Bayesian Network generates the same results or not, Or if the Bayesian Network is consumed through Visual C++, does

the value -5.555555 pops up in the results when I discretize the data for developing the Bayesian Network.