Investigating the ability of an artificial neural network to forecast stock market movement based on news events.

By Marcus Joiner (073515764)

05 May 2010

CSC3095 - G400 Computing Science
Supervisor: Dr Peter Andras

Word Count: 20860

INFORMATION

Declaration

I declare that this dissertation represents my own work except where otherwise stated.

Acknowledgements

I would like to thank Dr Peter Andras for his guidance, support, patience and trust. I have thoroughly enjoyed working with him and his team.

ABSTRACT

Artificial intelligence, and particularly neural networking, promises to provide solutions in application areas previously reserved for human judgement. Whilst capable, existing neural networks have several issues and areas that can be improved; and this document reviews their effectiveness and draws comparison with biological intelligent systems such as the brain. Taking a conceptual approach I propose a new design to tackle some of the issues using a communication-based neural network; attempting to mimic the biological function of the human nervous system. Having produced a workable neural network I perform tests using financial share data and news events to try and predict future stock movements, and have a reasonable degree of success. Although having insufficient data to draw sound conclusions, I do support my hypothesis that environmental awareness for a neural network can replace pre-programmed rules. Finally I look at improvements that could be made to my neural network design to increase performance and accuracy, and suggest avenues for continuation of the project in the future.

TABLE OF CONTENTS

CHAPTER 1 - INTRODUCTION

In September 2009 I embarked on an ambitious final year project with little idea where it would take me, and the outcomes I would achieve. Over the course of the project I have seen the aims and motivations change; learnt about and become fascinated with neurobiology; lost many hours of sleep producing test results; and presented what I believe to be a sound argument questioning the nature of existing neural networks. Through the pages of this document I try to condense the project into a concise but detailed report for those with little or no prior knowledge of artificial intelligence.

To help with navigating through the document I have provided a brief document map below. Those with specific interests can use the map to jump to the appropriate chapter easily and quickly. For each chapter I provide an introduction to its content and the areas that are addressed.

Before reading any further than this introduction you should familiarise yourself with the terminology I use throughout. In some cases I have used shorthand or abbreviated terms and phrases which require clarification; and in others I want to be sure there is no ambiguity.

I then move on to the factors that influenced my choice of project, with a brief narrative describing how my proposal changed and the reasons for those changes.

This leads directly into my hypothesis, and the aims that I produced in order to keep my development focused towards it. At the end of this chapter I then highlight some of the changes in my motivations that have occurred throughout the project, mainly due to a lack of understanding and appreciation at the very beginning. It try to present in summary the main theme that is present throughout the document, and what I hope will be the result for anyone that reads it.

1.1. Document map

This document is divided into the following chapters:

- **Chapter 1 – Setting the scene**

I begin by briefly describing the origins for my interest in AI and the motivations for choosing this topic as my final year project. This hopefully provides context and understanding to some of the decisions I discuss in later sections. I also introduce my initial aims and hypothesis; setting out what I wanted to achieve and the approach I thought that I would take.

- **Chapter 2 – Background**

Chapter 2 starts by offering a brief history of Artificial Intelligence (AI). I look at how AI has developed over the years, and some of the concepts that have become common place. This leads to a review of some current projects and applications, indentifying the leading areas of research and development and the tools and techniques they are promoting.

- **Chapter 3 – Developing a solution**

Having reviewed the existing developments I remained unconvinced of their suitability for the task at hand, and so chapter 3 justifies that decision and provides a detailed account of the solution I developed. This begins by looking at the conceptual flaws in existing neural networks and how they might be addressed, and then moves on to a practical definition of how my neural network overcome these flaws. I finally consider the operational implications of my neural network and the supporting architecture that links it together.

- **Chapter 4 – Results**

In order to investigate the hypothesis and provide judgement on my neural network performance I conducted a number of tests using real-world data. Chapter 4 shows the consolidated results from the testing, in a user-friendly and easily understandable format. In conjunction with the results I also use this chapter to explain how I conducted the tests; the reliability and accuracy of the data used; and the many issues and problems encountered. Although thorough evaluation is covered in the next chapter, during chapter 4 I highlight notable trends in the results to draw attention to them.

- **Chapter 5 – Evaluation**

Leading on from the test results I use chapter 5 to provide interpretation of them. I look at two key areas that relate back to the initial aims: Making predictions and improving predictions with experience. In both cases I justify my evaluation with constant reference back to the results; and although considering my testing to be inconclusive I suggest possible trends that may be appearing, whilst avoiding speculation or assumption.

Having considered the tests I then review the project and my approach to it. Conscious of the disappointing lack of a conclusion I try to concentrate on improvements that could be made to future developments. In all evaluations I am critical and constructive, and in most instances provide a detailed account of new changes.

- **Chapter 6 – Conclusion**

In final review of the project I summarise the findings of my evaluation in relation to the original aims and hypothesis. I consider each aim in turn and whether it has been achieved, making recommendations for improvements that would better serve the hypothesis. After detailing my final conclusions I then suggest an approach for anyone wishing to continue the work I have started, again offering practical and detailed suggestions drawing from my thoughts and experiences.

- **Appendix Pack**

Throughout the writing of this document I have deliberately removed development specific details and technical considerations to better support the projects conceptual nature. For those wishing to review some of the finer detail that has been omitted an accompanying CD contains an Appendix Pack with the following content:

a) Appendix A – Raw data collected.

   All of the raw data collected on news articles and share price updates, as used for testing.

b) Appendix B – Input files.

   Comma separated input files containing cleaned versions of the raw data collected, in a suitable format to be imported into my neural network via sensors.

c) Appendix C – Raw Predicted Files.

   Unedited output files from my neural network each time a pause was defined and a prediction asked for.

d) Appendix D – Variation Comparisons.

   A collated view of the test results including the functions and calculations used to determine averages and variations.

e) Appendix E – Project Source Code

   A NetBeans project folder containing both compiled and non-compiled versions of my neural network, including the various drivers required.

1.2. Terminology

Below follows a list of terms commonly used in this document:

Term	Definition
AI	Artificial Intelligence (AI) is the replication of intelligent behaviour, such as human intelligence, using man-made devices, most commonly now using the digital computer.
Biological system	The term biological system throughout this document refers to biological intelligent systems; such as the human nervous system.
LSE	London Stock Exchange
Neural Network	A neural network is a network of neurons, be they biological or artificial; however unless otherwise specified the term refers to an artificial neural network throughout this document.
Neuron	A biological neuron.
$\forall$	For all.
$\rightarrow$	Implies

1.3. In the beginning

The concept of AI has fascinated me from an early age, inspired no doubt by its relentless use in science fiction. In recent years AI has secured itself a place in conventional computing, with advancements in technology and approach allowing for AI based applications to provide functionality previously unthinkable. AI fetches the results when we Google; intelligently routes our telephone calls; competes against us in computer games and directs us when driving.

Despite all these applications, none however mimic the walking-talking automatons that science fiction promises us, and largely the AI community has moved away from replicating human intelligence in favour of more abstract models that can provide applications to specific and focused problems.

Appreciating that any future work in AI that I conduct may also be bound by similar pressures to produce useable application, I wanted to take maximum advantage of the freedom this final year project offered me. Convinced I wanted to explore replicating biological function in computing, my initial proposal ideas were very different from the final project choice.

I too feel that human intelligence is beyond our current technological ability; but felt that biological functions that we consider sub-conscious must be controlled in a similar manner by the brain, but in a more simplistic or basic method than higher brain functions. My initial proposal was therefore to replicate the cardio respiratory system, using no pre-programmed logic or function. The assumption was that if I could produce a generic, neural-mimicking system that could match biological behaviour and function it might provide a foundation for gradually building more intelligent systems.

The basis for my proposal was simple. I felt that the brain receives a number of environmental sensors that provide it with information on the state of the body, and it has a number of actions that it can perform using muscles. This is often referred to as the feedback mechanism (Wiener, 1948). By monitoring and remembering the changes in the environmental state after performing actions the brain could, without any pre-programming, determine which combinations of movements would achieve desired states. Initially using random combinations of muscle movements, the model would quickly learn: expanding the diaphragm muscles increases the availability of oxygen, and contracting decreases the level of toxic carbon dioxide, for example.

I hypothesised that the brain would begin with its most urgent need – pumping oxygenated blood around the body; and this could be modelled using a neural network. This initial project proposal had one considerable flaw: In order to test any neural network model I constructed I would need to also create a simulated environment. The simulated environment could potentially have become more difficult to construct that the network itself; and any results would always be questionable due to abstractions in the simulated environment. Considering the time constraints this appeared to be an unachievable aim.

Convinced that environmental factors can drive decision making without pre-defined programming the requirement was to find a similarly complex system where a real-world environment could be used for testing, without the need for simulation. I began by formulating my hypothesis using generic references whilst I continued to search for a suitable testing environment.

1.4. Aims and hypothesis

Having removed environmental specifics the hypothesis I wanted to prove became:

> *"Provided with sufficient awareness of its environment, an intelligent computer model could identify patterns within a complex system, forecast future states based on those patterns; and learn and develop those forecasts to improve accuracy based on experience."*

After researching many environments, including metrological systems, network traffic models and computer gaming I decided upon using the financial markets environment. There were several key justifications that led me to this decision:

- Availability of data in an electronic format.
- Simple, limited movement options (increase in price, decrease in price, and no change).
- Easily available reference material on the subject.
- Many comparable systems already in use, with varying and questionable success rates.

I then turned to creating a series of aims and objectives that would assist in the development of a system to prove the hypothesis.

The main aim of this project was to attempt to prove the hypothesis, using the context of the Financial Markets, by constructing a system that could:

1. Gather and store, in real-time, News and Share price data from multiple, disparate sources;
2. Store this data in a format that allows for pattern recognition and identification both within and across the sources;
3. Produce future trend forecasts based on identified patterns and computed logic;
4. Improve the accuracy of forecasts over time based on feedback and experience.

Additionally the system was to be generic in design; such that input sources from a different system environment could be associated. It would then continue to produce forecasts based on the new environment.

1.5. Changes

Having confirmed what I wanted to achieve, focus then switched to how to achieve it. My initial thoughts and ideas went through many changes, and as a result by the end of the project my motivations were considerably different.

Although I still evaluate the project against the initial aims and hypothesis, I also consider my success in areas that I had not initially factored.

The main change in focus for the project was a switch from results to method. The hypothesis concerns the production of results and looking at their accuracy; and the demonstration of learning through improvement in the results. As the project progressed I became less concerned with the quality of results, and move intrigued by the conceptual problems with existing neural networks, and how they might be corrected.

My development became less about producing an application of solution to an existing problem, and more towards producing a prototype that would experiment with new techniques and promote questions and discussions on how we currently implement AI.

For this reason my evaluation concentrates not on the results achieved, which were somewhat inconclusive. Instead I draw out the areas in the prototype that worked and those that didn't – including how to improve them.

I try to promote thought and questioning of existing techniques not to suggest that my solution is better, but so that aspects of my approach might be discussed and considered to improve future developments in AI.

CHAPTER 2 - BACKGROUND

AI as a concept has existing in human society for centuries, dating back to ancient Greek mythology with legends such as Talos the bronze protector of Europa to Hephaestus and his mechanical servants. In the mid 1940's after the invention of the digital computer the prospect of a thinking machine became more of a reality; and in the decades since progress has been considerable.

In this chapter I will introduce the brief history of AI and the key milestones that have marked its progression to where we are today. This provides a prelude to my review of current approaches to AI, including some of the latest methods of neural network simulation.

2.1. History of AI

AI as we would now recognise it has its origins in the mid 1940's where a series of advancements fuelled by the arrival of the digital computer lent serious consideration to machines performing advanced tasks that could be associated with intelligent behaviour. The term "Artificial Intelligence" was officially coined by John McCarthy in 1956 at the Dartmouth conference he organised to promote scientific thinking on the subject, but research and development was already underway prior to this.

Comparisons between computers and the human brain were already being made, laying the foundation for future neural networks. Consideration into how neurons work led to the construction of primitive neural networks (McCulloch & Pitts, 1943), enhanced through further theories into how learning is performed "not by establishing new connections but a selective reinforcement of connections already capable of functioning" (Hebb, 1949. pp.132) – a concept still prevalent today.

Through the early 1950's AI and neural network development took a backseat to mainstream computing, fuelled with the increasing improvements in hardware and the immediate practical advantages that they offered. This remained the case until Frank Rosenblatt developed the first working neural network, the "Perceptron" (Rosenblatt, 1958). With the concept of neural networking now firmly established the next two decades saw the introduction of several networking models that expanding on the initial theme; each with their own advantages, disadvantages, and preferred application areas.

The mid 1980's saw significant advances in neural network, with the publishing first of the Hopfield neural network (Hopefield, 1982) followed by the Kohonen neural network, often referred to as self-organising maps (Kohonen, 1984). These two implementations provide for the use of neural networks in commercial applications, with proven learning capability. Almost all neural networks developed since have been variations on the principle concepts developed by these two networks; but things are now beginning to change.

2.2. Current research and development

Although all of the commercial neural applications I have found are variations on Hopefield or Kohonen networks, research and development is expanding new concepts and approaches. These can be divided into the following categories, which are not mutually exclusive to each other:

- Generic approach to neural networking

  Existing neural networks in commercial use are almost entirely bespoke to the environment they operate in. With neural networks being expanded into increasingly complex operating environments, financial return motivations are forcing the consideration of reusable neural network models. Community research projects such as NEOSIM[1], NeuroML[2] and NEURON[3] strive to promote the concept of a “plug-and-play” (Goddard et al, 2001) solution to neural networking; but are hampered by lack of agreement in standards and approaches.

- Massively parallel computing

  As neural networks are introduced to increasingly complex environments, the demands on their performance are also increasing. Neural networks are expected to process more data, produce more results, and work increasingly in real-time or near real-time environments. To tackle this approach researchers are attempting to model the parallel operational of the biological systems; using two methods.

  The first sees the use of super computers that combine multiple processors through an interconnected bus to maximise centralised power; whilst the second approach relies on distributed computers across a network to provide processing improvements. In either case the aim is to allow for concurrent firing of neurons and thus an increase in performance.

- Reverse engineering of brain function

  The third area of research moves away from the practical application of neural networking, and focuses instead on attempting to better understand and model biological systems. Projects such as Blue Brain Project[4] believe that by understanding the connections between neurons and the transmission of signals we can better construct artificial neural networks. Models have already been constructed for the neo-cortex of rats; and as hardware improvements are made it is hoped that higher order creatures can be modelled, eventually leading to humans. The ultimate aim is to reverse engineer an intelligent system through the study of brain function.

[1] NEOSIM – a parallel kernel for spiking neurons. [Online].
NeuroGEMS – Software for neuroinformatics. Available at: http://www.neurogems.org/neosim2/ [last accessed 01 May 2010].
[2] Model Descriptions for Computational Neuroscience. [Online].
NeuroML. Available at: http://neuroml.org [last accessed 01 May 2010].
[3] NEURON for empirically-based simulations of neurons and networks of neurons. [Online].
NEURON – Yale University. Available at: http://www.neuron.yale.edu/neuron/ [last accessed 01 May 2010].
[4] Blue Brain Project. [Online].
Ecole Polytechnique Federale de Lausanne. Available at: http://bluebrain.epfl.ch/ [last accessed 01 May 2010].

CHAPTER 4 - DEVELOPING A SOLUTION

In this chapter I will explain in detail the solution I have developed to meet the aims of the project.

I start by identifying and justifying the principles I set out before beginning my design. This is important as it provides insight and support into many of the design decisions I made.

I then briefly look into existing systems, offering analysis of their weaknesses and supporting my decision not to use any of them.

This leads on to the system I have implemented. Firstly I look at how I countered the problems with existing systems using design guidelines at a conceptual level, and then moving to the detailed design.

In documenting my design I try to gradually introduce the component features and operations of the system, building the picture in detail whilst explaining the motivations behind decisions.

Where possible I have tried to mathematically explain, justify or prove elements to remove ambiguity and provide a sound basis for continued development or critic.

The foundation of any neural network should be the neuron, and so I begin with how information is stored within my neuron design and how it associates with other neurons to form the network.

The neuron alone will not meet the aims of the project, and so I then progress to looking at the supporting network model that includes:

- Detection – How values are introduced to the network.
- Firing – How neurons are activated and activate the neurons they are connected to.
- Forecasting – How activating neurons can be interpreted and returned as a prediction.

With my neural network model outlined I move to the supporting architecture required to best enable it, focusing on the areas of neuron distribution across a network of computers and fault tolerance.

4.1. Approach principles

Before commencing with design I imposed several approach principles to provide direction, guidance and constraint. These principles were derived from my own beliefs and theories around AI; and the personal objectives I wanted to achieve from the project.

Their scope had significant influence over key considerations and issues that arose, whilst not deviating from the aims. In many instances they dictated how I wanted to tackle a problem. In some instances this direction led me away from conventional and accepted practises. Although conscious of the effects of such deviation, adherence to these principles has made this project uniquely personal.

The principles I decided upon can be broadly defined under 2 categories: Mimicking biology and environmental awareness.

4.1.1 Mimicking Biology

Initial attempts at AI in the 1950s (such as Rosenblatt, 1958) were abstract developments based on the biological workings of the brain and other biological systems - as these provide our only known reference. Even now neural networks are abstract models of biological systems, to varying degrees.

Despite this, and an increased understanding of biology and how biological systems function, AI models have become increasingly abstract. Rather than continuing to improve the emulation of biological systems, they instead focus on the replication of function and result only.

This divergence of biology and AI can be attributed to 3 key assumptions on: Technology, Suitability, and Application:

a) Technology.

Despite the continued advancement of technology it is still widely considered that modern computers are no match for biological systems. Many in the field of AI consider the goal of replicating biological intelligence to be beyond our current capability. Those that continue to model biological systems have, in the most part, moved away from the complex systems of humans and other intelligent mammals. Instead they focus on simpler models of lower order creatures, believing them more appropriate for our technological capabilities.

Closer investigation and conceptual thinking, in my opinion, negates this argument, as I demonstrate below.

A neuron is able to cycle, on average, at 200 Hz. In comparison the computer that I am using to write this document is able to cycle at 9.6×10^9 Hz. This is an overly simplistic comparison; but that does not detract from the point that processor technology is beyond neuron capability by many orders of magnitude.

In terms of capacity we have yet to determine how a single neuron stores information; or how much information it can store. Comparisons between neurons and digital units of storage therefore serve only to confuse the issue; but we can consider that neurons in humans are almost identical in structure to neurons in other organisms. It is therefore reasonable to assume that the volume of information they contain must be of a similar order of magnitude. This suggests that it is not the capacity of a neuron, but the complex connectivity and volume of neurons, that foster intelligent behaviour.

If this is the case, the issue becomes not how much a neuron can store, but how much information can be accessed in a firing cycle.

Standard personal computer memory currently operates at 8×10^8 Hz. Based on the average cycle time for a neuron this allows the computer to access 4Mb of data in the same period. If we assume that each neuron contains less than 4Mb (about the size of a 3 minute song in MP3 format), then computers do have the capacity to match biological neurons.

Having demonstrated that computers can match neuron capability and neuron capacity, the remaining and key issue is volume and connectivity. The human brain is estimated to contain 1×10^{11} neurons, with each neuron connected to as many as 7000 other neurons. In extremis, due to the asynchronous way in which neurons operate, we could assume a worst case scenario of all 1×10^{11} neurons firing simultaneously, and continuously, at 200 Hz. This results in 2×10^{13} neuron firing events every second; which is several orders of magnitude greater than the raw power of the modern computer.

This is not surprising. In the comparison above I compared the computer to an individual neuron. When scaled this leads to the human brain being akin to a network of computers. Although even the largest network known to man, the Internet, falls short of the required size with estimates at 1×10^9 computers[5], networks using multi-tasking computers provide the potential scope to realistically model biological systems. Capability and capacity constraints can be overcome with additional machines. The key issue is then performance throughout the network, which determines the capability and speed of the model.

Network speeds may not be able to replicate the near instant transmission times between neurons, but reduced speeds only serve to make any resultant model slower – not less intelligent. Biological systems operate in the real world where they are required to process information and respond in real time. There are other environments where responses are not as time critical. The flow of information can be regulated to meet performance constraints; which does not detract from the implementation and accuracy of the model.

b) Suitability.

Biological systems have evolved over time to their present state. They have developed in conjunction with the other systems that comprise an organism, and the environment in which it functions. The digital computer is fundamentally different. In function similarities can be drawn: both appear to store information and retrieve it; and both appear to be able to conduct processing of information.

In operation however the two systems share little commonality.

[5] "Internet Users (per 100 people)" [Data].
Data Finder, World Bank. Available at: http://datafinder.worldbank.org/ [last accessed 25 March 2010]

Biological systems operate in analogue, whilst computers are digital. A single transmission from a neuron can encode a range of values. In comparison each transmission from a computer is restricted to binary and therefore carries either a 1 or 0.

Most importantly for neural networks, a computer processor is a serial device. Instructions are parsed one by one. Biological neurons are able to operate asynchronously, allowing for many to process at the same time.

With such disparity between technology and biology many in the field of AI consider it conceptually inappropriate to try and model biological systems. Instead they promote new approaches and ideas, using only the function of biological systems for their abstraction.

This is undeniable, and has historic precedence outside of computing. In the late 19th century attempts at manned flight modelled birds; and were unsuccessful. It was only when the Wright brothers considered the science behind flight and moved away from replicating nature that success was achieved. However, whilst being unsuccessful, the attempts to model birds were not useless. The information they provided shaped scientific thinking and prompted questions on the nature of flight. Every failed attempt contributed to scientific knowledge and understanding.

Had we not attempted to replicate bird function, would we have learnt enough about flight to produce the aeroplane?

I am certain that modelling AI on biological systems will not provide an optimal solution. It is highly likely that it will fail to achieve intelligence anywhere near the same scale as human intelligence. Despite this we must consider that biological systems are the only reference we have, and that modelling them will likely improve our knowledge and understanding of both AI and biological systems.

These two concerns, complemented with recent successes in AI development have resulted in a trend away from modelling biological systems.

There are sound arguments both fore and against modelling AI on biological systems. In summary I believe that computers are comparable with neurons for capacity and capability, but many networked computers will be required to simulate a biological system. Even if successful in modelling a biological system using a network of computers, it is almost certain that this will be a sub-optimal solution.

Despite this, the knowledge learnt from failed models not based on biological systems provides only an insight into computers and AI. In contrast attempts to model biological systems, even if unsuccessful, have the potential to expand our understanding of both AI and biological systems. It is for this reason that I have tried to mimic biology throughout this project.

4.1.2 Environmental Awareness

Biological systems are comprised of many cells, but those most considered to contribute to intelligence are Neurons and Glia. Our understanding of Glia is limited, but it is thought that they are facilitators to neurons. It is the neuron that stores, processes and communicates information and instructions.

Biological neurons can be classified by many criteria. Common examples being reference to the number of Neurites they have; or the Neurotransmitters they contain. Neurons of all classifications are present throughout biological systems, although concentration may vary.

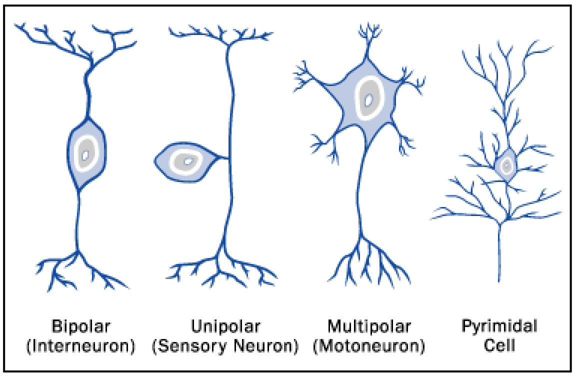

Figure 1 - Neuron categories based on Neurite configuration[6]

Although areas of the brain have been attributed to different senses, to date no neuron has been associated with a particular sense. Despite the differences in information that the visual cortex and the auditory cortex must store and process, they share similar neuron types. This implies that the types of neurons relate to task or function, and not the nature of the information they store.

This generic assumption of neurons can be translated into 2 scopes in AI:

a) Generic environmental inputs.

All of the practical instances of existing neural networks that I have researched are designed to work in specific environments. Whilst they may share similar algorithms, they are configured to operate on certain information within set bounds.

As an example, neural networks designed to perform data mining on text information are not able to switch to visual images without modification. A truly generic implementation of a neural network would be able to interchange between any data type, just as a biological neuron appears able to.

In addition most neural networks contain an element of pre-programmed logic. Logic rules relating to conditions that have been identified external to the network are imposed on its operation. For example AI implementations of search engines are programmed to ignore superfluous words such as “and” and “the”. A truly generic solution would allow the neural network to discover for itself that these are superfluous words, and develop its own self imposed rules. Biological

[6] “Basic Neuron Types” [Image].
How Stuff Works. Available at: http://static.howstuffworks.com/gif/brain-neuron-types.gif [last accessed 21 March 2010]

systems, aside from the questionably possible transmission of basic instincts, do not receive such logic programming.

b) Expandable environment.

Existing neural networks, in general, consider single streams of information. Using the stock market as an example, many neural networks monitor share price, whilst using pre-defined logic to interrupt movements from other contributing data such as exchange rates. Neural networks do not have the flexibility to add additional, potentially disparate, information sources.

Biological systems are not constrained in such ways. They are exposed to the entire environmental spectrum, within the capabilities of their senses. Data is not selected for its appropriateness; and is not governed by logic rules.

Whilst the weather outside might not be influential to stock market movements; it is for the network to determine this, not the programmer

By increasing a neural networks awareness of the environment, including a full range of input types, it will be able to make more informed judgements and decisions. Current methods may be forced by technology constraints (due to their design), but this is self serving in the pursuit of AI.

I am certain that greater environmental awareness will result in a more intelligent system, and therefore I have aimed to develop an open and generic approach to environmental inputs.

4.2. Existing neural networks

Neural Networks form a leading edge of our AI developments. Whilst there are many types, some of which are detailed below, they all share common traits.

Unlike traditional computer programs that follow pre-defined process steps to arrive at a solution, neural networks have the ability to process non-linear data where process steps are difficult or impossible to define. For this reason neural networks have proven useful in tasks such as data mining and recognition.

Existing neural networks generally consist of layers. Whilst the connections and configurations vary between types, the fundamental principles remain the same. Each neural network generally contains an input layer, one or more hidden layers which perform processing, and an output layer.

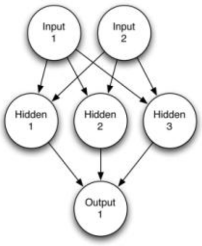

Figure 2 - A Typical feed-forward neural network with a single hidden layer[7]

Environmental states are submitted to the neural network through the input layer. Each input neuron represents an environment state. Once the input layer has been submitted the neurons are processed through the hidden layer(s) which either provide recognition or conduct logic processing (AND, OR, XOR).

Recognition is achieved using weighted connections and thresholds. If the connection exceeds the threshold the neuron will fire, indicating successful recognition. Logic processing is conducted using combinations of neurons to produce outputs.

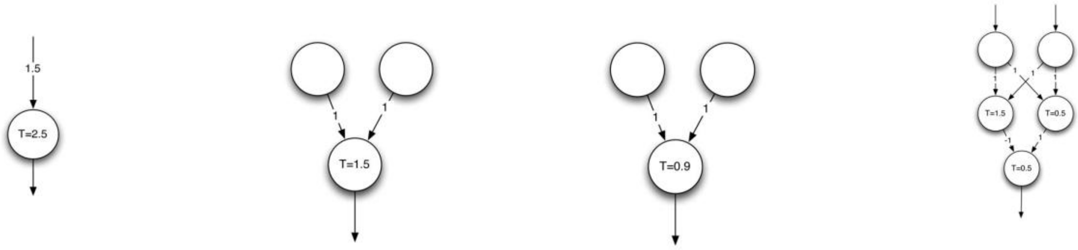

Figure 3 – Examples of layer configuration to produce different results. From left to right: a weighted connection and neuron threshold used for recognition, an AND operator, an OR operator, and an XOR operator.[8]

[7] Heaton, J. 2008. "Introduction to neural networks for Java" [Image].
Heaton Research. Available at: http://www.heatonresearch.com/node/704 [last accessed 30 March 2010]
[8] Heaton, J. 2008. "Introduction to neural networks for Java" [Image].
Heaton Research. Available at: http://www.heatonresearch.com/online/introduction-neural-networks-java-edition-2/chapter-1/page4.html [last accessed 30 March 2010]

A key feature of neural networks is their ability to learn. This is achieved through the adjustment of weights and thresholds. Learning can take place in two ways: supervised or unsupervised.

Supervised learning requires the construction of training sets demonstrating correct outcomes. These sets are then processed by the neural network and are used to adjust the weighted connections between neurons. Once training is complete the weights are fixed. Due to the requirement of known results supervised learning is best suited to recognition tasks.

Unsupervised learning is used for environments where correct operation can't be prescribed or defined. Instead weight adjustments are made by the neural network based on actual environmental data. This makes unsupervised learning more appropriate for prediction tasks.

Whilst there are many types and variations of neural networks, 3 of the most common are:

a) Hopfield neural network (Hopfield, 1982).

   Hopfield neural networks are single layer neural networks in which every neuron is connected to every other neuron (Fully connected). Requiring supervised learning, they are suited more to pattern recognition rather than prediction.

b) Feed-forward, back-propagation neural network.

   Feed-forward neural networks are the most common implementation architecture used due to their ability to suit many tasks. Neurons in feed-forward neural networks are connected by directed edges, meaning processing must pass from the input layer, through hidden layers, to the output layer. Cycles are not permitted. Similar to Hopfield neural networks they require supervised learning.

c) Kohonen neural network (Kohonen, 1984).

   Often called self-organising maps, Kohonen neural networks differ from feed-forward neural networks in two key areas. Firstly they do not produce a range of outputs. Instead a single output is produced, often referred to as "the winner". This allows Kohonen neural networks to consider many possible outputs, and determine the most suitable. Instead of using functions or threshold values, weights are normalised and translated through the various layers. The output neuron is decided by the largest weight. Secondly Kohonen neural networks do not require supervised learning.

Whilst neural networks have proven success in many application areas, they are not without their problems and constraints.

a) Process-based and not communication-based.

   Existing neural network architectures are still process based. Similar in design and operation to graph algorithms, they process through nodes (artificial neurons) and their connections (edges). This fundamental trait means that they are serial algorithms. Each node is evaluated in sequence, moving through each layer until the output layer is reached.

   Consider the feed-forward neural network in Figure 2. The processing algorithm must evaluate each node in the hidden layer against each node in the input layer to produce an output. This results in a minimum of 6 evaluations.

This can be expressed as:

$$x = i.h^n$$

x - number of evaluations
i - number of input nodes
h - number of hidden layers
n - number of nodes in hidden layers

Whilst this function is a simplification of the problem, and assumes that each hidden layer has the same number of nodes, it highlights that the result is of an exponential order. Adding a second hidden layer to Figure 2 will result in 16 evaluations, a third layer 54, and so on.

For this reason large neural networks are constrained by the number of hidden layers they can contain. It is common for neural networks to only contain 1 hidden layer, whilst more than 2 is rare. To have no limit on the number of layers using current approaches would risk memory overflows or restrict performance to such an extent that the system would not be able to operate in real-time environments.

Biological systems are not constrained in such a manor. Neurons are connected to many other neurons through many layers, which can include cycles. This is possible because biological systems are not process based. Instead they are communication based. Neurons communicate their signals to other neurons, which can then independently begin evaluation, not having to wait for a serial algorithm to reach them.

b) Non flexible environments.

When developing neural network architectures the types of environmental inputs must be carefully considered. Comparing disparate sources that have no defined connection should be a task well suited to a neural network, but complications with adding additional and non-compatible input sources has seen targeted neural network developments that lack flexibility.

This is partly due to the performance issues that arise when increasing inputs and layers, and partly due to the requirement to normalise data types into comparable units.

Biological systems do not have this restriction. Senses contribute to an overall environmental state. Although certain senses may have primacy, absence of some sensory information does not eliminate recognition. Equally biological systems appear to be able to incorporate new sense information to aid recognition. An example is initial recognition by visual means that is later supported by smell or taste.

c) Limited temporal awareness.

Despite the ability of neural networks to perform basic prediction, they are more suited to recognition tasks where defined solutions can be specified, and the network trained to identify those solutions. Some recognition based neural network architectures are modified to produce forecasts by connecting the output layer to the input layer and creating a repeatable cycle (such as Elman networks (Elman, 1990)), but this is rare in practical implementations.

Recognition requires set inputs which are non-consequential. The current set of inputs has no influence on the next input and its solution, except in the use of re-training the neural network to improve its recognition.

Forecasting inputs are consequential. The next input may depend on some or all previous inputs. This introduces a temporal element to forecasting that existing neural networks often struggle to incorporate. This temporal element becomes more complex when considering multiple, disparate input sources. Just as "the butterfly effect"[9] describes how seemingly unconnected events may hold influential relationships, future forecasts for one sense will have influence over future forecasts for others (Sutton & Barto, 1990).

Having considered the problems and constrains that existing neural network architectures are bound by I determined they would not be suitable for this project. This meant a requirement to develop a new style of neural network.

[9] The butterfly effect is a common example of Chaos theory, which dictates that complex and unpredictable outcomes can result from the smallest difference in input states.

4.3. Solution guidelines

In order to overcome the problems and constraints of existing neural networks I established several design guidelines:

a) Distributed communication based system.

   The main drawback with existing neural networks is their serial operation. Whilst improvements to use parallel processing are now being examined (McClelland, 2010), they only offset the issue instead of removing it. To truly mimic biology, and achieve performance on a similar scale, a neural network must be able to allow the independent processing of each neuron. To support a communication based architecture a network of computers must provide the processing power to meet the performance demands.

   Whilst unrealistic to expect an individual computer for each neuron, a distributed solution would provide the expandability required:

   - Maximum use of the parallel capacity of modern computers with their superior speed over biological neurons;
   - The ability to easily integrate additional computers into the model;
   - A distributed approach that sees load balancing and performance maintenance across the network.

b) Flexible environment.

   The complex financial markets environment has many influential factors. Although this project has only examined the connection between share price and world news, future improvements may require additional input sources of any data type.

   A scalable system would be able to increase the number of inputs it receives without requiring modification. The addition of new input sources during runtime provides an extra consideration to support existing decision making; seamlessly integrating with existing data.

   In addition input sources must be able to contain any data type to support both a scalable and generic approach. These data types must allow not only for the standard computing types such as numbers, characters and Booleans; but conceptual user-defined inputs such as colours or emotive descriptors.

c) Temporal awareness

   In order to produce forecasts for multiple inputs a solution must be temporally aware. More importantly this must be across the environment and not specific to individual inputs. Forecasting requires a temporal view of the previous environment, current environment, and consideration of the predicted future environment (Connor & Douglas, 1994). This is because the forecast state of a particular input source may have more influence over a second source than its own.

4.4. The neuron

The basic building blocks of biological systems are neurons. Activation signals are received across synaptic gaps between the neurons dendrites and the firing neurons axon. These signals travel through the neurons cell body and are then retransmitted along the axon.

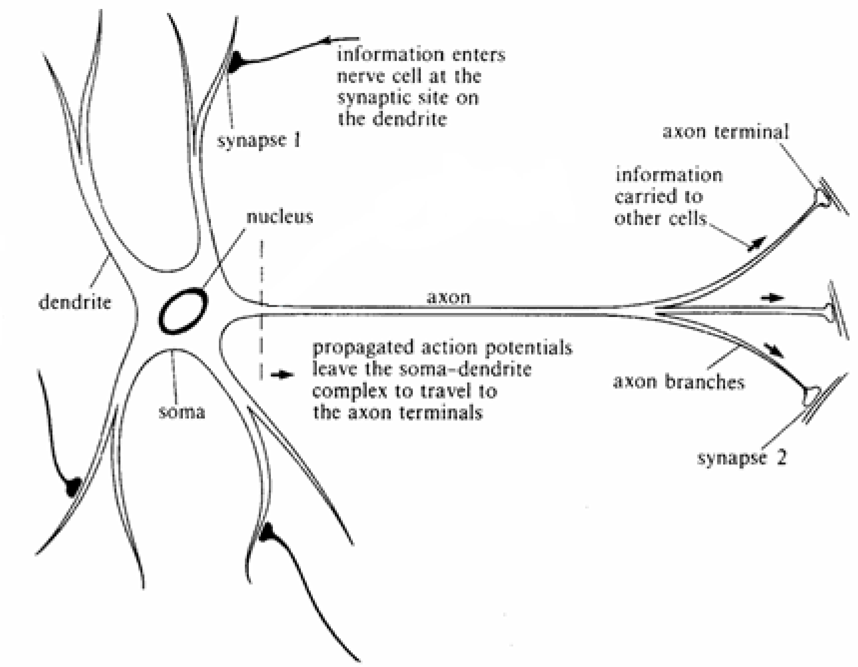

Figure 4 - Basic Neuron structure[10]

The axon has many branches that connect to the dendrites of other neurons. The electro-chemical manor in which signals are transmitted across the synaptic gap is beyond the abstraction required for this project, but some key observations can be made.

i. The central cell body does not appear to have duplication in its internal structure. I therefore assumed that a neuron represents a single environmental state.

ii. A neuron has many dendrites, connecting to many axons, and so can be triggered by many other neurons.

iii. A neuron has a single axon through which to retransmit its signal, however the axon is branched in many places and connects with many dendrites from other neurons.

In most existing neural networks the neuron is a primitive entity similar to a node in graph structures. Depending on the type of neural network it may contain activation functions or threshold comparators.

My first task was to create a more complex neuron implementation that could better approximate the behaviour of a biological neuron.

[10] "A Brain Neuron" [Image]. Code Project,
Available at:http://www.codeproject.com/KB/recipes/NeuralNetwork_1/BrainNeuron.png [last accessed 24 March 2010]

4.4.1 Percepts

Based on the assumption that a biological neuron represents a single state from the environment, an artificial neuron must be able to store or associate itself with that state. AI terminology refers to an environmental state as a percept (Russell & Norvig, 2003). In biological terms this may be a component of a colour, a smell, or a taste. A percept in the financial markets environment will be a price update; and in news terms it may be an article, a paragraph, a word or a letter.

Percepts are generated by the external environment and then passed to the system, which stores, recognises and processes them.

To ensure a generic approach I developed a percept interface. The interface acts as a wrapper to allow any data type to be associated with a neuron, providing it meets the following requirements:

i. Object based environmental state.

   The environmental state is stored as an object inside the percept wrapper. This allows for future users to develop and implement their own environmental states, not limiting the system to the pre-defined computer norms of numbers, characters and Booleans.

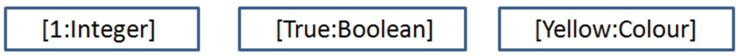

Figure 5 – Percept wrappers containing the standard Integer and Boolean objects, along with a user defined Colour object.

ii. Comparable.

   To allow for the recognition of repeated environmental states each percept must be comparable to other percepts. This is accomplished in two ways. To allow for rapid retrieval of neurons an indexing system was required, which has been implemented using hashcode. Each percept must be capable of generating a unique and consistent hashcode. In addition percepts must, as a minimum, implement an equality comparator.

iii. Divisible.

   In the real world percepts are not isolated from each other. A colour we observe is a composite of the 3 primary colours: red, green and blue. In the news environment paragraphs are composites of words, whilst they in turn are composites of letters. Simple experiments have shown that when we read words on a page we are actually recognising patterns of letters (Larson, 2004). Spelling mistakes and letter translations do not stop our recognition.

   This implies that whilst a neuron may store a word as a percept, it is also associated with the letters that comprise it. For this reasons the interface allows for percepts to be divisible.

Percepts must declare if they are divisible, allowing the system to automatically perform a break-up to the component parts. The iterative way in which percepts are divisible allows for multiple layers of division.

For example: News article => sentences => words => letters or characters.

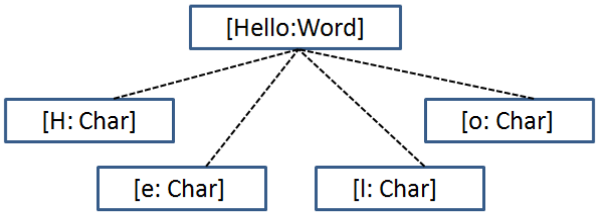

Figure 6 - Divisible quality of a user defined percept Word

Having defined the percept and provided an implementation for it, I considered the wider environment. Percepts are generated by an input source, and represent a single state at a fixed moment in time.

Consider a percept p as the state of an input source I at time interval t. Assigning t_c as the current time, we can state:

$$I = \{p_t |\, 0 > t < t_c \}$$

$$p_{t_c} \in I$$

In an environment with only a single input this would be sufficient, however we must consider that multiple input source $I_o \ldots I_n$ may exist. When a change in the environment is detected we can now express both the changed percept p_{t_c} associated with input source I_i and the wider environmental state e_{t_c}:

$$I_i\, p_{t_c} \in I_i$$

$$e_{tc} = \{I_i p_{t_c} \,|\, 0 > i < n\}$$

In a fully environmentally aware system we acknowledge that changes in the state of input source I_i can have subsequent effects on the state of any other input source; until the state of I_i changes again. These subsequent percepts (neurons[11]) s can be stated as:

$$s = \{I_x p_t \,|t_c > t < (t_c + 1)\; and\; 0 > x < n\; and\; x \neq i\}$$

[11] Having already defined that a neuron is associated with an individual percept, we can easily interchange the two terms.

In summary when input source I_i changes:

New percept:	$I_i\, p_{t_c} \in I_i$
Environment:	$e_{t_c} = \{I_i p_{t_c} \mid 0 > i < n\}$
Subsequent neurons:	$s = \{I_x p_t \mid t_c > t < (t_c + 1)\ and\ 0 > x < n\ and\ x \neq i\}$

4.4.2 Axons

In graphical terms we can now represent the neuron and its subsequent percept connections as:

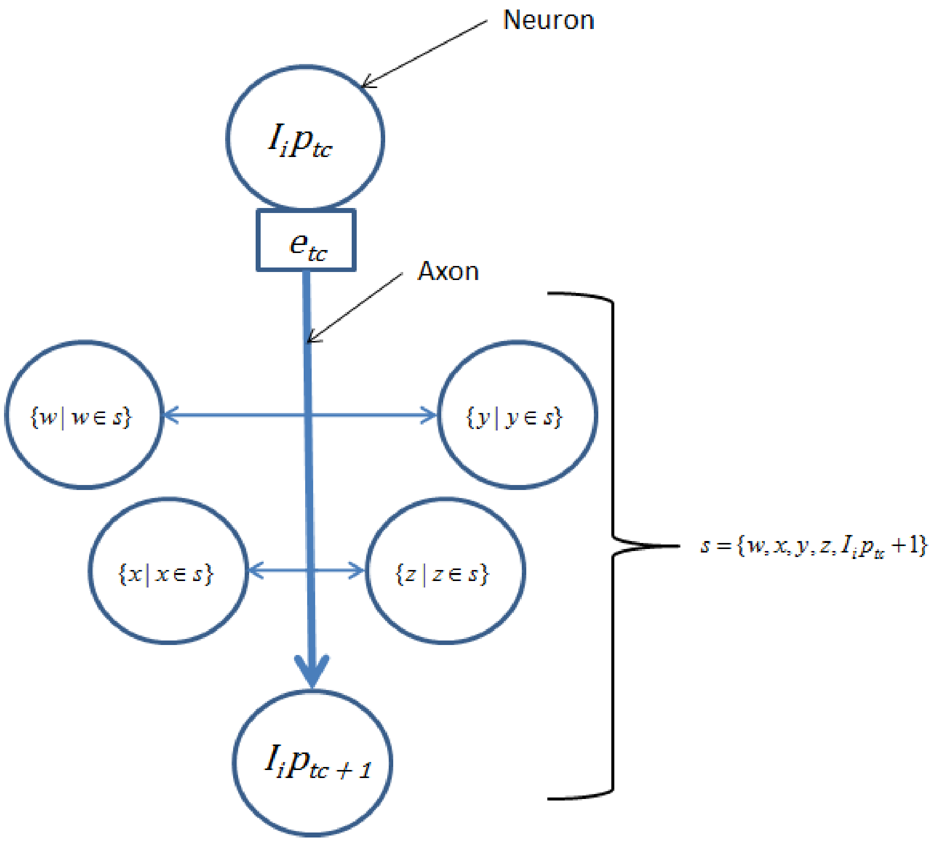

Figure 7 - Neuron connections for a new Percept

This model supports the association of percepts to neurons. It also supports connectivity between neurons independent of their input source or type. Using this model receipt of percept $I_i\, p_{t_c}$ will result in the trigger of all subsequent neurons in s.

Whilst this may be true at time t_c, lets consider future occurrences of the same percept at any time Δt. It is not guaranteed that the environment will always be the same, and subsequent neurons may also differ:

$$I_i\, p_{\Delta t} \equiv I_i\, p_{t_c}\ \forall\ \Delta t$$

$$e_{\Delta t} \neq e_{t_c} \forall\, \Delta t$$

$$s_{\Delta t} \neq s_{t_c} \forall\, \Delta t$$

Modification of the model to allow for different environmental states and different subsequent neurons results in:

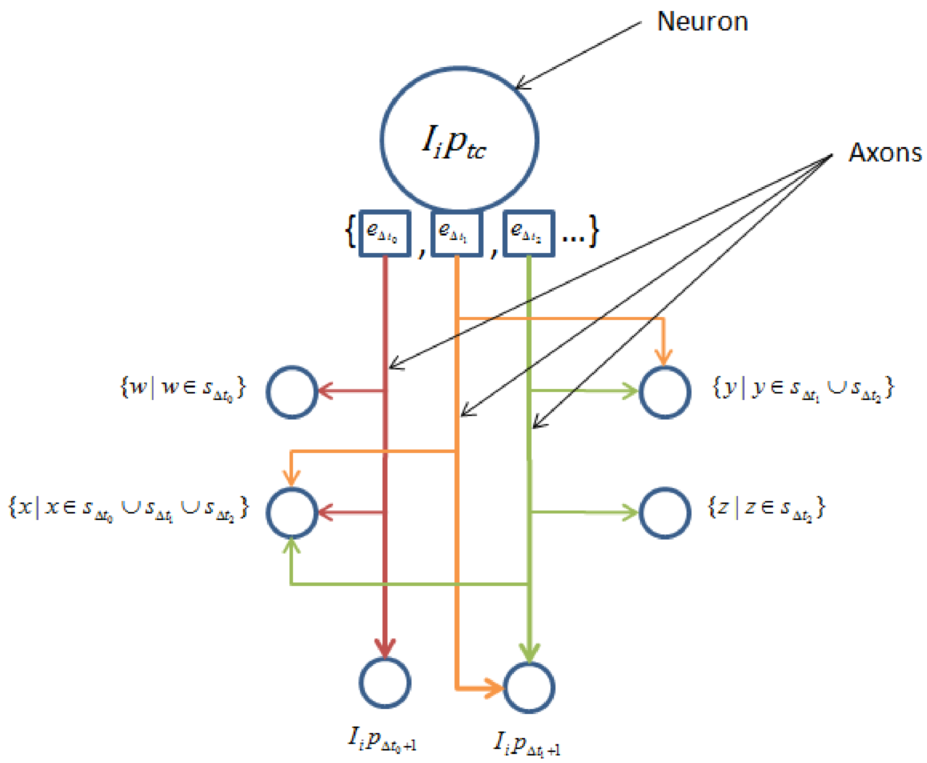

Figure 8 - Neuron associated with multiple environments

Whilst an improvement, this new model makes a flawed assumption that for every occurrence x of an environmental state e the subsequent neurons s always occur, such that:

$$e_x \rightarrow s_x \;\forall\, x$$

If all subsequent neurons were consequential to the originating percept this may be correct, but in an environment that contains multiple, disparate input sources we can conceive that not all of the input sources will be consequential to all of the others.

As such the subsequent neurons s_x for different occurrences of the same environmental state e_x at time Δt may be different, although we would expect them to share common elements:

$$e_{x\Delta t} \rightarrow \{s_{x\Delta t} \mid s_{x\Delta t} \subset s_x\}$$

From this we can determine the likelihood or probability ₱ of a subsequent neuron n' occurring:

$$n' \in s_x$$

$$₱_{n'} = \frac{\boldsymbol{n}(n' \in s_x \; \forall \, x)}{\boldsymbol{n}(e_x \in e \; \forall \, x)}$$

*Note that in set notation $\boldsymbol{n}(x)$ refers to the cardinality of the set x.

This probability can be used when forecasting to determine the likelihood of a subsequent neuron occurring, within the range 0 > ₱< 0.99[12]. Percepts received direct from the environmental are attributed 0.99. A probability of 0 implies the percept will never occur.

At this point I can also introduce an average transition time between receipt of a percept and subsequent neurons.

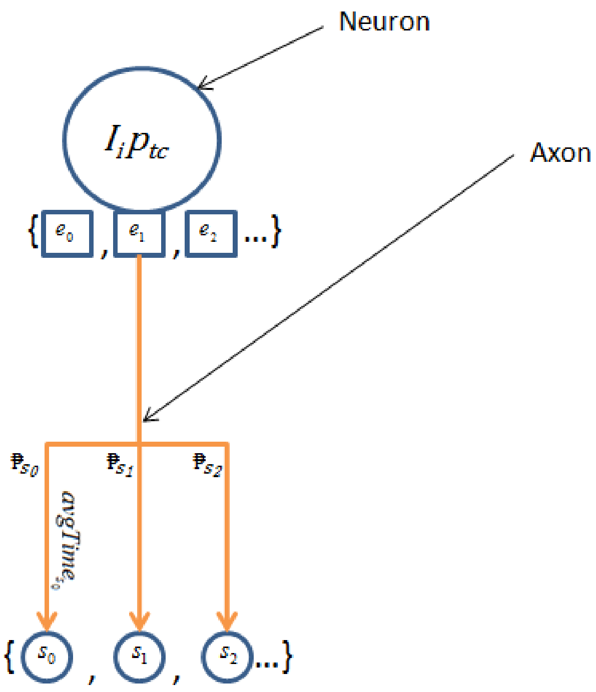

Figure 9 - Axon showing how probability and average time are attributed to connections

To recap we now have a model for a neuron which is associated with a percept, and can be identified on repeat occurrences of the same percept. The neuron can have any number of axons, with each axon associated with an environmental state and connected to any number of subsequent neurons. Each subsequent neuron has a probability and an average transition time.

[12] Although probability is normally measured within the bounds of 0 to1, for practical implementation reasons it is easier to restrict the upper limit to 0.99. When calculating the probability for subsequent neurons by multiplying the probability of their originating neurons, the result must be a diminishing value. In the case of neuron cycles ($a \to b \to a$) this ensures an infinite loop is never entered.

Whilst suitable, this model has limited learning ability, and can have its efficiency improved by deducing logic from the state of its connections.

4.4.3 Logic

In the current model subsequent neurons can be associated with a number of environmental states, with multiple axons connecting to the same subsequent neuron. This is expected.

If we consider a number of axons a_0 ... a_x, each having different environmental states, are connected to a subsequent neuron n':

$$\{a_0 \ \dots \ a_x\} \rightarrow n'$$

It could be that these environmental states are unconnected, but more likely they will share some common, decisive percepts E:

$$E = \{e_x \mid e_x \ \rightarrow a_x \ and \ a_x \ni n'\}$$

$$\neg E = \{e_x \mid e_x \ \rightarrow \ a_x \ and \ e_x \not\owns E\}$$

$$E \ \cap \neg E \rightarrow n' \mid e_x \rightarrow n' \ \forall \ e_x \ \in E$$

As an example, the sets below represent 3 environmental states at which a percept is encountered, the subsequent neurons, and one possible logic rule that can be derived from them:

$$e_0 = \{x, y\} \rightarrow \ a_0 = \{\alpha, \beta, \theta\}$$

$$e_1 = \{x, z\} \rightarrow \ a_1 = \{\alpha, \beta\}$$

$$e_2 = \{x, y, z\} \rightarrow \ a_2 = \{\alpha, \theta\}$$

$$(e_0 \ \cap \ e_2) \cup \ e_1 = \{x, y\}$$

$$(a_0 \ \cap \ \ a_2) \cup a_1 = \{\theta\}$$

$$\therefore \{x, y\} \rightarrow \{\theta\}$$

The first logic rule therefore identifies percepts that always precede a subsequent neuron, and assumes a direct correlation.

We can also consider that some neurons may always generate a subsequent neuron irrespective of the state of the environment. We can express this as:

$$e_x \ \rightarrow \ n' \ \forall \ x \mid n' \in \{a_0 \cap \ a_1 \ \cap \dots \ a_x\}$$

Using the previous example:

$$a_0 \cap \ a_1 \cap \ a_2 = \{\alpha\}$$

$$e_x \ \rightarrow \{\alpha\} \, \forall \, x$$

The second logic rule therefore identifies subsequent neurons that always succeed a neuron.

The scope of the project did not permit time for additional logic rules to be implemented, but many more could be derived in the future. These rules differ from the rules pre-defined in existing neural networks. They are not directly related or linked to a specific external factor in the environment. In contrast they are produced by the network itself using generic and logical mathematical rules and derivations.

4.5. Neural network model

Once I had defined the neuron implementation it was necessary to consider a new neural network architecture to support it. To fully explain the network, and the considerations that influenced the design, I shall gradually introduce and build on each of the areas and their features.

I broadly class these areas as: Detection, Firing, and Forecasting. It is easier to consider them in reverse order:

4.5.1 Forecasting

Imagine a percept p_{t_c} is received, and is associated with a neuron n. Based on its axons this may have subsequent neurons of n' and n''.

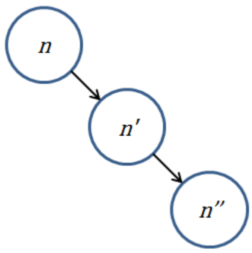

Figure 10 - A forecast list

In this instance the forecast could be maintained as a list of the subsequent neurons, in the order that they will occur. Unfortunately as we complicate the example we begin to see that a list is no longer sufficient.

Imagine the same percept has occurred several times, and each time an axon has been created with differing subsequent neurons. There are many subsequent neurons that may occur as a result, and therefore the forecast is more suited to a tree. The neuron n will be the root of the tree.

We must also consider that multiple input sources will provide environmental data, and therefore the forecaster will be required to maintain multiple forecast trees.

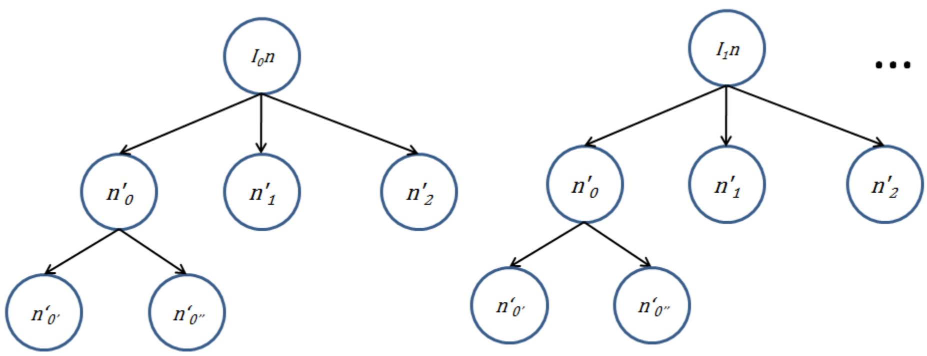

Figure 11 - Forecast trees representing different input streams

These forecast trees represent the range of possible subsequent neurons that may result, but they are not a prediction. To produce a prediction we must convert the tree into a single output list. This is done using the probability factor that is associated to each subsequent neuron. It is also now an appropriate time to introduce the average time value from the axon.

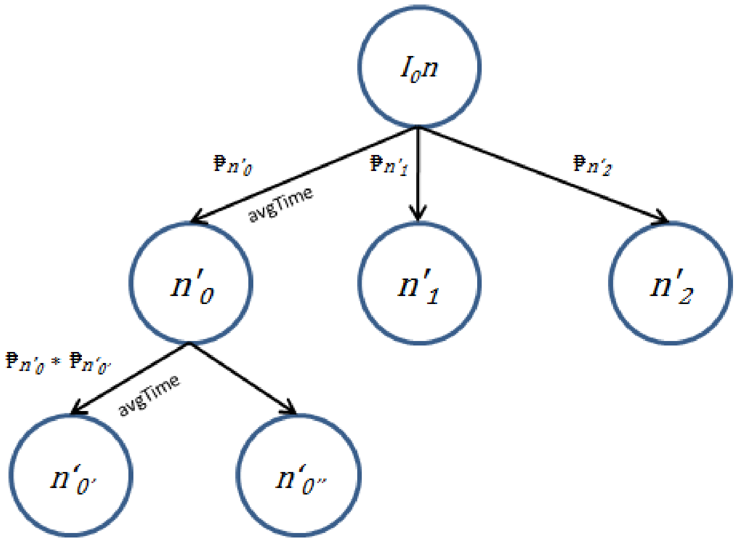

Figure 12 – A forecast tree including probability and average time

If we navigate the tree by only expanding the child node with the highest probability, and always increase depth, we will return the most likely subsequent neurons and therefore our prediction.

Having identified the need to generate a forecast tree for each input stream, and how to produce a prediction from the forecast tree, we must turn to the construction and maintenance of the tree.

We have already stated that the root of the tree will be the neuron associated with percept $I_i p_{t_c}$, however this will continually change as new percepts are received.

One option would be to reconstruct the forecast tree each time a new percept is received. This would be inefficient and in a system that requires rapid response would likely lead to unacceptable delays in performance, or insufficient time between percepts to generate the complete forecast tree.

To counter this we must have faith in the accuracy of the forecast tree. If our forecast trees are correct, we can assume that the neuron associated with $I_i p_{t_c+1}$ exists in the tree. In the best case it will be the highest probability child node from the root. In the example above this would mean that $I_i p_{t_c+1} \equiv n'_0$. We can set n'_0 as the root of the tree, and all its subsequent neurons and their connections remain extant. All other nodes are pruned from the tree, and an update process can adjust the probability values to reflect the change.

In more complicated instances it may be that $I_i p_{t_c+1}$ exists many times in the forecast tree, at different levels. In this case all instances of nodes that are equal to $I_i p_{t_c+1}$ are merged into a single root node.

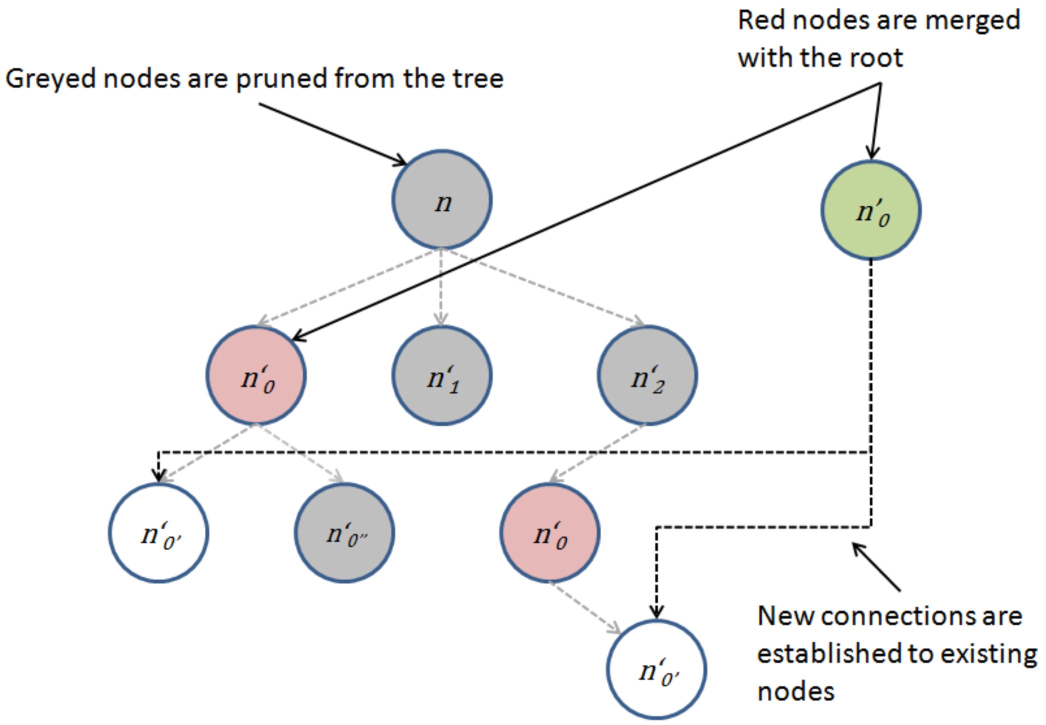

Figure 13 - Integration of a new neuron into an existing forecast tree

The forecaster is now able to construct forecast trees for any number of input streams, adjusting efficiently as new percepts are received; and able to produce predictions of subsequent neurons from the forecast trees.

There is one final feature that the forecaster must incorporate: the concept of a temporal fringe.

When defining the neuron implementation and its axons I stated that $e_x \rightarrow s_x$, showing how the environmental states influence the subsequent neurons. At $I_i p_{t_c}$ we can expect to know the state of the environment e_x, and have expressed it as $e_x = \{I_x p_{t_c} | 0 > x < n\}$.

This is only true at time t_c. Once our forecast trees moves to subsequent neurons, the environment will have changed. We do not know what the environment will be at any given time Δt, but using the forecast trees we can attempt to predict it.

For any time Δt we predict that the state of input source I_i will be one of the subsequent neurons in its forecast tree, within a timed range equal to Δt. Furthermore we know that many subsequent neurons may be predicted to occur at time Δt, and we know the probability of them occurring.

This information allows the construction of a temporal fringe: a set of the possible environmental states and their probability. Predicted future states can therefore influence subsequent predictions.

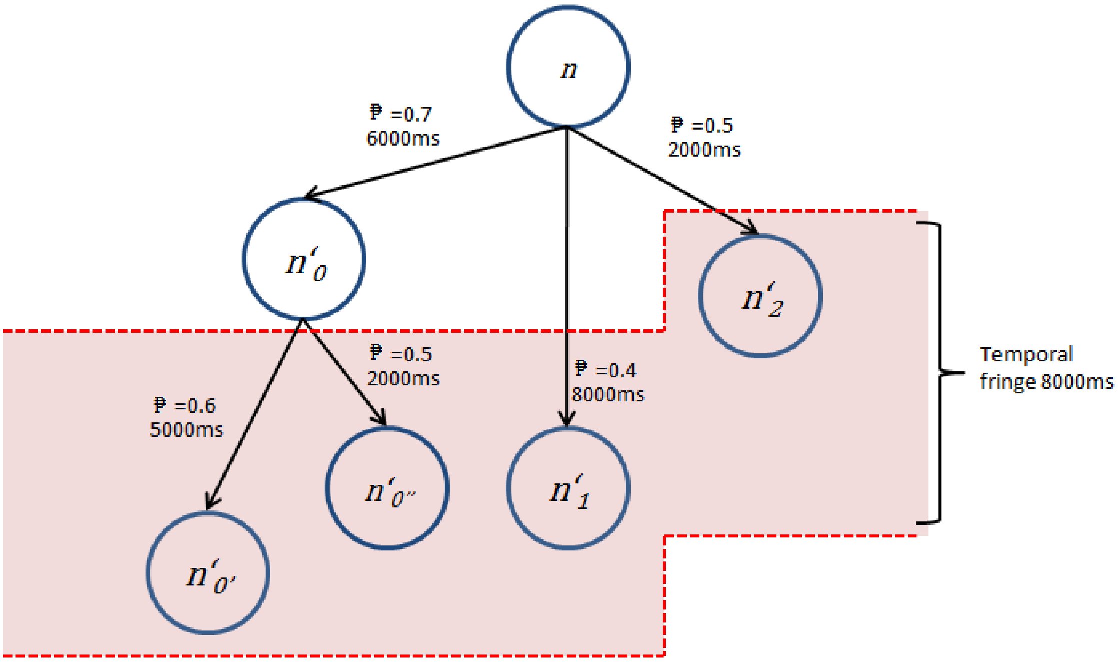

Figure 14 – A temporal fringe of 8000ms

If the temporal fringe for all input sources is combined we have a complete predicted environmental state $e_{\Delta t}$.

4.5.2 Firing

In order to populate a forecast tree the forecaster must be aware of the neurons and subsequent neurons as they are activated. In biological terms, this activation is known as a neuron firing.

In a serial process an algorithm would iterate through the neurons, activating each one and systematically populating the tree. In a distributed environment where neurons can fire independently, a signal or trigger is needed to inform the neuron when to fire.

Consider a new percept is received and associated with a neuron that has several axons a_0, a_1 and a_2, connecting many subsequent neurons.

It is unlikely that the current environmental state e_{t_c} will exactly match one of those associated with the axons, and therefore firing a single axon would be inappropriate.

Instead the neuron must fire each of its associated axons, but apportion a level of appropriateness to them based on how similar they are to the current environment.

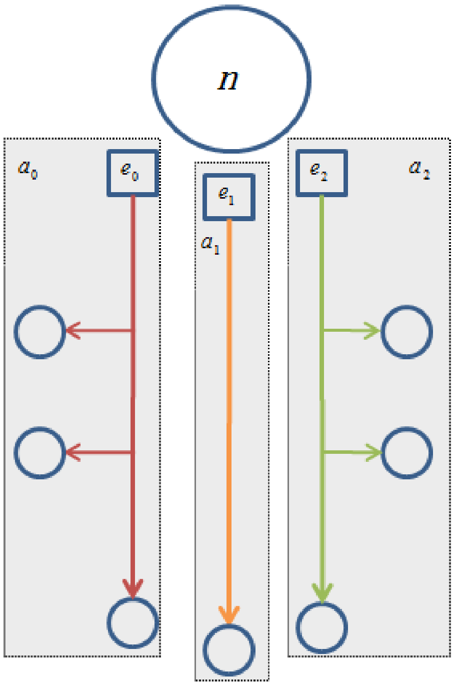

Figure 15 - Received neuron with 3 axons

If the environment is comprised of percepts from 10 input sources, and e_{t_c} shares 5 similar percepts to e_0, we can say that a_0 is 50% appropriate. This can be stated as:

$$Ᵽ_{a_x} = \frac{e_{t_c} \cap e_x}{n(e_x)}$$

Having already assigned probability factors to subsequent neurons as part of the axon, we can now combine this with the appropriateness of the axon itself. Receipt of a new percept has a probability $Ᵽr$ of 0.99. The probability of a subsequent neuron is therefore the received probability $Ᵽr$ multiplied by the axon probability $Ᵽ_{a_x}$, and then multiplied by the neuron probability $Ᵽ_{n'}$:

$$Ᵽ = Ᵽr * Ᵽ_{a_x} * Ᵽ_{n'}$$

The neuron can now generate triggers for each of the subsequent neurons in each of its axons, weighted with a signal strength calculated on the probability of a subsequent neuron occurring.

As subsequent neurons receive this trigger they begin to create their own triggers. This presents a problem. The axon probability was initially calculated using the current environment e_{t_c}. The subsequent neuron now represents a future time Δt, and as such the environment will most likely have changed.

The forecaster is able to produce the predicted environment state $e_{\Delta t}$ using the temporal fringe method; and so on receipt of a trigger the neuron must request the predicted environment from the forecaster. Each predicted state is associated with a probability, and therefore the axon probability can be stated as:

$$Ᵽ_{a_x} = \frac{(e_{\Delta t} \cap e_x) * Ᵽ_{e_{\Delta t}}}{n(e_x)}$$

As the forecaster receives requests for $e_{\Delta t}$ it becomes aware that a neuron has been fired, and can add it to the forecast tree. To do this the forecaster needs to know which node the neuron is to become a child of. Neurons triggered from changed percepts become the root node and have no parent, whilst all subsequent neurons become children of it.

I overcame this problem by including a reference to the parent node with the trigger.

A trigger received from a changed percept will not have a parent node, instructing the forecaster to allocate a new node as the root. A reference to this new root node can be returned to the neuron and then included in the trigger to subsequent neurons.

On receipt these neurons will announce themselves to the forecaster, including the parent node reference. The forecaster creates a new node for them, attaches it as a child to the parent node, and then returns a reference to this newly created node to be the parent for all subsequent neurons. The process is then repeated.

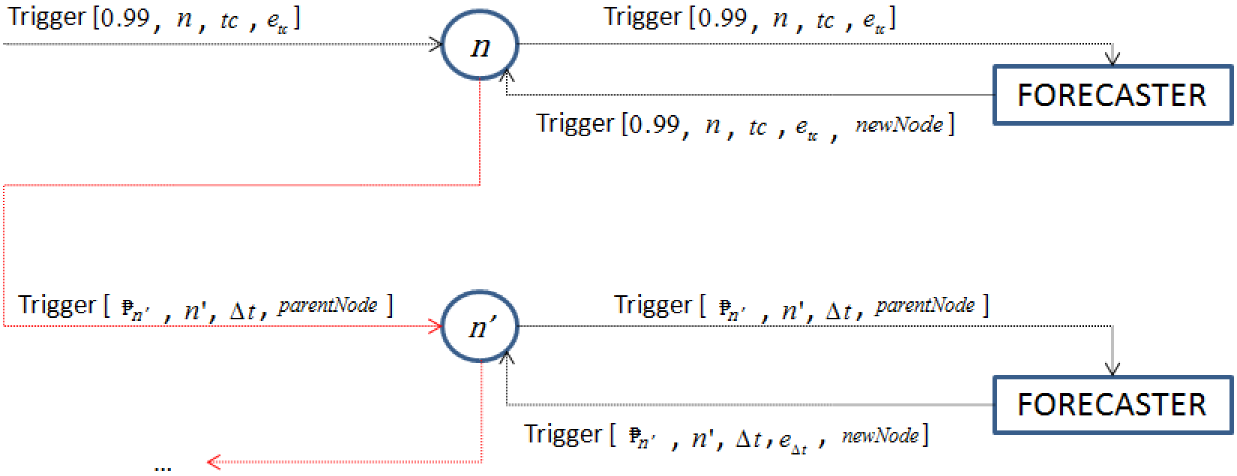

Figure 16 – Trigger communication between neurons and the forecaster

An advantage of this model is that the forecaster is able to regulate the firing of neurons. When a new percept is received and the root node changes, forecasts that were wrong will be pruned from the tree. Any future triggers pointing to these pruned nodes are simply rejected. This ensures that neurons do not fire indefinitely, providing new percepts are being received.

4.5.3 Detection

Having explained how neurons fire, and how as they fire they are tracked by the forecaster to produce a prediction, the remaining element to my neural network model is how neurons are created and how axons are assigned to them.

The first requirement is to allow the system to interact with the environment so that it can be made aware of changes as they occur. Each input source is represented by a sensor. The sensor observes changes in its state and converts these changes into percepts that can be interpreted. A financial markets sensor may detect changes in share price, whilst a news sensor may activate when new articles are published.

When a sensor observes a change and a percept is generated there are two possible actions that may be required:

- If the percept has been encountered by the system before, it will already have a neuron associated with it, and this neuron should therefore be triggered.
- If this is the first occurrence of the percept then a new neuron is created, and associated with it.

In either case, we are at the point where a neuron is associated with the percept. The task now turns to creation of axons.

By definition the axon is associated with the state of the environment, which means that all the current sensor states must be known. To centrally collate sensor information we can introduce a detector. Each sensor passes its percepts to the detector for processing.

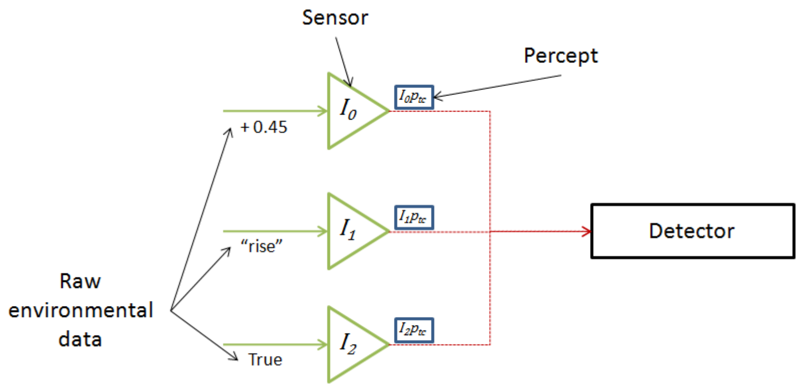

Figure 17 - Detection system

On receipt of a percept the detector creates a new axon that is associated with the sensor. The axon is assigned the current environmental state, and should then be considered "under construction".

As new environmental changes are received from other sensors, the time delay and a reference to their associated neuron is added to the axon. The axon continues to build until its creation sensor changes again. At this point a final subsequent neuron is added for the new percept, and the axon is complete.

Although the axon is at the detector, the neuron may be elsewhere; and may have fired many times whilst the axon was being constructed. This presents a need to locate the neuron and integrate the new axon.

Once located, if the new axons environmental state matches that of an existing axon associated with the neuron, the two are merged; otherwise the new axon is appended to the neuron.

4.6. Support architecture

Having developed a new model for a communications-based neural network, I identified the need for a supporting architecture required to enable the transmission of messages and activation of objects in the manor described. This architecture needed to support distributed processing, harnessing the power of multiple computers over a network.

4.6.1 Neuron server

True distributed processing to simulate a biological system, in the extreme, would require each neuron to have a dedicated processor; but this is impractical and unnecessary. Instead, acknowledging the parallel processing and multi-tasking capability of modern computers, I decided that each neuron should operate as a single processing thread. This enables each neuron to receive an equal share of the processors time when it requires it. Providing the number of neurons associated with each computer is managed this should provide sufficient performance.

The decision to associate multiple neurons with each computer implies a degree of management, and led to the concept of a Neuron Server. The Neuron server facilitates the following functions:

- Tracking of neurons located on the host computer.
- Passing of received messages to the intended recipient neuron.
- Termination of neurons that have expired (not been fired within a set period).

Additionally the neuron server is able to provide performance management and prioritisation to the neuron threads that it hosts. In burst periods (where many neurons are firing multiple connections at once) priority can be given to those neurons with higher signal ratings (those considered a higher probability of being correct). The neuron server is also able to provide a degree of resilience; restarting any neuron threads that encounter difficulty and resolving any errors and faults before they become systemic.

Finally, to load balance neurons across all available neuron servers, they have the ability to prematurely terminate neurons. This releases them to another neuron server under less load.

4.6.2 Directory server

With the likelihood of neurons being spread across multiple neuron servers; and with the neurons able to be transferred between neuron servers; a method to ensure messages are passed to the correct destination was needed. Drawing inspiration from Domain Name Service (DNS) and how it operates, I introduced the concept of a Directory server.

Each implementation of my neural network model requires a single directory server, whose address is known by all elements of the network. More than simply a message forwarder, the directory server has a key role to play in the distribution of neurons and messages across the network. As the central focus for the network, it was logical to base neuron storage with the directory server. With this key responsibility, the directory server facilitates the following functions:

- Maintenance of a permanent, persistent database store for all neurons.
- Pairing of incoming percepts to existing neurons, or the creation of new neurons for percepts.
- Distribution of neurons to neuron servers, and on their termination receipt of neurons from the neuron servers to commit them back to the permanent database store until needed again.
- Maintenance of a mapping of neurons to servers to enable routing of messages to the correct location, adjusting the mapping in real-time as neurons are transmitted and released.

Concerned again with the possibility of systemic failure of the network, especially in view of the single point of failure that the directory server represents, great care was taken to ensure a robust implementation. Noting the distributed nature of the neuron servers and the possibility of neuron servers become disconnected whilst processing, the directory server has procedures to identify non-responding neuron servers and re-distribute their neurons to other machines. Whilst this may incur data loss, the focus for all error and fault identification and rectification was to ensure the continued operation of the neural network.

4.6.3 Detector

Having identified and defined the directory server, my neural network model pointed to the requirement of a server to collate sensor information and maintain the environmental picture. The detector is the simplest of all the server elements in the supporting architecture, supporting the following functions:

- Central receipt and collation of sensors into a complete environmental picture.
- Matching of incoming sensor percepts to existing neurons via the directory server.
- Creation of axons and the communication of axons to neurons via the directory server.

4.6.4 Forecast server

The final element to the supporting architecture is the forecast server. As discussed when defining the model operation, neurons communicate to a forecaster both to inform it of their activation (and the percept they represent), and to retrieve the probably environmental state for their subsequent triggers. This requires a separate machine, able to receive communication from neurons located on any neuron server, and transmit results back to the neurons via the directory server.

The Forecast server therefore performs the following functions:

- Creation of forecast trees (graphs) for each sensor.
- Addition of nodes (percepts) to the forecast trees.
- Pruning of the forecast trees as actual percepts are received from the detector, via the directory server and neuron servers.
- Generation of temporal fringes to indicate probable environmental states across the forecast trees.
- Generation of a prediction by navigating the forecast trees using the highest probability branches, always progressing in depth.

When combined together these individual components make up the supporting architecture operating over a TCP-IP network and providing distributed, parallel processing for the communications based neural network.

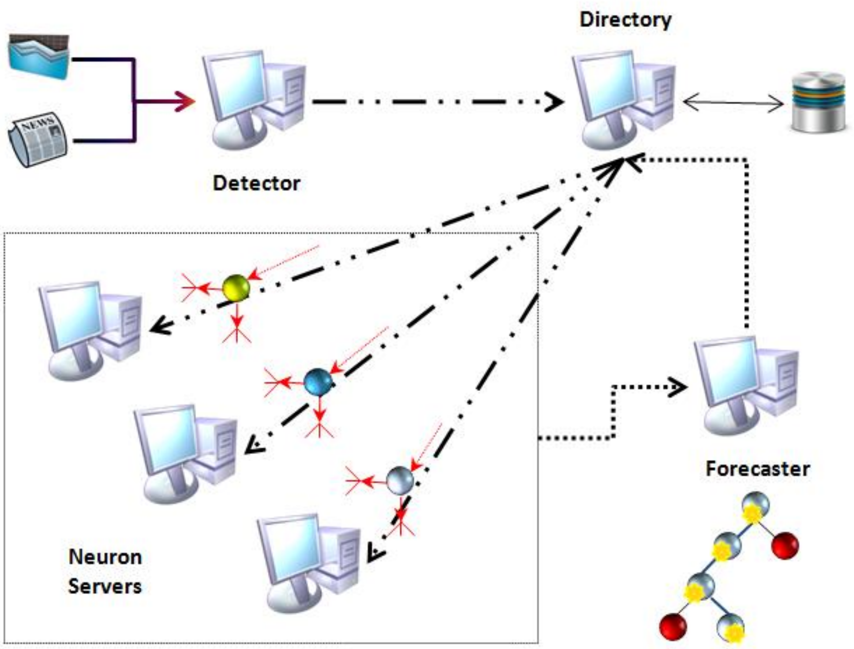

Figure 18 Supporting architecture

CHAPTER 5 - RESULTS

In this chapter I will explain in detail the testing undertaken with my neural network, and highlight the key results obtained, before evaluating them in the next chapter.

I begin by describing the practical elements of the testing, including the approach and environment used. This is both to provide context to the results, and enable replication of the results should the project be revisited in the future.

I then briefly describe the selection of data used for testing; including the justification used in narrowing the available data scope. This covers the collection method, which should provide sufficient detail to enable collection to continue or be expanded should the project be continued.

Before moving on to the results themselves I then briefly discuss some of the key issues encountered during testing, their implications on the testing, and how they may have influenced the results. This is a prelude to my evaluation in the next chapter where I expand on the issues and make recommendations for future improvements.

Having provided context to the results I examine 3 key areas, drawing direction from the original aims and hypothesis of the project. I present the results, with explanation as to what they indicate and any caveats that apply, in graphical form. The tabulated results can be found in Annex A.

The 3 areas are:

- Ability to produce predictions.
- Directional accuracy.
- Accuracy and improvement with experience (demonstrating self-learning).

5.1. Testing Procedure

Initial testing of my neural network focused on ensuring that the network operated correctly under test conditions. This involved supplying known data such as multiplication tables and prime numbers to ensure that prediction and learning behaved as expected. Once happy with the basic operation, testing required the introduction of actual news and share data.

5.1.1 Data Collection

Before commencing with design of my neural network model I identified that to receive meaningful results a large volume of data would be required, spreading over the largest possible time period. This prompted a need to begin data collection at the earliest point to increase the availability of test data.

With the focus of the project being the neural network, and not data collection, a compromise between collection from many sources and time constraints led me to narrow the scope of data. This is because collection from each source required the development of an automated routine, and would lead to a data cleansing and integration time cost (all of which would reduce the time available for design and development of the neural network).

With this in mind I quickly eliminated any data sources that could not be freely retrieved; and that were not available in an electronic form that lent themselves to automated collection and import (which quickly eliminated all but the internet based sources). Having narrowed the scope to internet data providers it was desirable to retrieve both news and share data from the same provider. This added the advantage that their format would be similar and cleansing requirements would be consistent throughout the data.

Using this method to narrow the scope pointed to two obvious providers: Yahoo and Google; and with Yahoo data streamed via XML in a format that lent itself to easier collection I opted to use Yahoo News and Yahoo Financial for data collection.

Despite this narrowing, it still presented a vast amount of data that I felt needed further reduction.

Firstly I wanted to reduce the number of shares being investigated, and therefore considered which I believed would likely be effected most by news events. With the unique circumstances that the global financial crisis presented and the social-political focus on banking I decided that I would select 5 UK based banks comprising[13]:

- BARCLAYS (LSE: BARC.L / INSI GB 031348658)
- HSBC HLDG (LSE: HSBA.L / INSI GB 0005405286)
- LLOYDS BANKING GROUP (LSE: LLOY.L / INSI GB 0008706128)
- STANDARD CHARTERED (LSE: STAN.L / INSI GB 0004082847)

[13] "UK Indices", Yahoo! Finance.
Available at: http://uk.finance.yahoo.com/m8 (last accessed 24 April 2010)

- ROYAL BK SCOTL GR (LSE: RBS.L / INSI GB 0007547838)[14]

In consideration of the shares selected I felt that the news categories could also be reduced. Whilst the true environmental picture would include all possible sources of information, news categories such as “Health and Beauty” I determined have little impact on banking shares and are therefore, for the purpose of this project, inconsequential. This led me to elect to monitor the following news categories:

- UK Banking Financial
- UK Business News
- UK Consumer Products
- UK Domestic News
- UK Funds
- UK Stocks and Shares
- UK Top News
- UK World News

Having determined the data to be used I developed an automated collection routine that retrieved updates from the XML feeds provided by Yahoo!, and integrated them into a database.

Copies of the raw data collected can be found in Appendix A in the Appendix Pack.

5.1.2 Method

Initially I had planned to conduct 3 independent tests on my neural network. These were:

a) Share data only.

   By using only share data to produce predictions I planned to provide a benchmark from which to assess the effect the inclusion of news data had.

b) Share and News data.

   Combining both data sources I planned to test the hypothesis and demonstrate an improved accuracy over the results of a).

c) Key news events.

   Using closer inspection to observe and report neural network performance after key news events.

[14] “UK & Ireland News”, Yahoo! Finance
Available at:http://uk.news.yahoo.com/ (last accessed 24 April 2010)

Unfortunately time constraints precluded test c) being conducted. The remaining two tests were performed in an identical manor, with the only difference being the input data provided.

A sensor routine provided automated retrieval of data from a comma-separated file produced from the collected data. This read in the next available line of data, determined its nature, correctly formatted the required percept, processed it, and then paused for a set period until repeating the process.

In both tests training sets of data covering late October 2009 and November 2009 were processed, with forecasting disabled. This enabled the creation of neurons and axons, without the processing overhead of forecasting; reducing the required pause period between inputs.

The remaining data for December 2009 and January 2010 was divided into two monthly input files.

The December 2009 file contained 10 pre-defined pause points. On reaching these points the input would wait for user acknowledgement, allowing for the output of predictions and the backup of the database. These pause points were taken through a 5 day period, at 1100hrs and 1500hrs respectively.

The January 2010 file contained 18 predefined pause points, covering 9 days. Again these were at 1100hrs and 1500hrs respectively.

Copies of the input files can be found in Appendix B in the Appendix Pack.

The pause points were selected because they represented consecutive working days, where data collection had been complete. In the case of December, the Christmas period was avoided to ensure consistency of data, hence the reduced number of pause points.

Initially I had hoped to produce forecasts at the end of each working day, however early tests showed that predictions concentrated on only the next few minutes of movements, and therefore did not cover the hours required to the start of the next working day. To meet this unforeseen outcome the timings were changed so that actual movement data was available to match the period of prediction.

5.1.3 Environment

For all testing the following environment was used:

Role	Make / Model	Specification
Directory Server	HP Pavilion m7755.uk	• Intel Core 2 6600 2.40 GHz • 4 GB Ram • Windows Vista 32-bit (SP2)
Forecast Server	HP Pavilion dv7	• Intel Core 2 6600 2.20 GHz • 4 GB Ram • Windows 7 64-bit
Sensors & Detector	HP Mini 110-1100	• Intel Atom N270 1.60 GHz • 1 GB Ram • Windows 7 Starter 32-bit
Neuron Server	HP s3010.d	• Intel Celeron Dual Core 3.4GHz • 1 GB Ram • Windows Vista 32-bit (SP2)
Neuron Server	HP Pavilion tx2500	• AMD Turion X2 Mobile ZM-80 2.10 GHz • 3 GB Ram • Windows Vista 32-bit (SP2)

Network: 100BaseT Wired network

Database: MySQL v5.0.90 Community

5.1.4 Issues

Whilst undertaking the collection of data and testing several issues arose. Two of these warrant discussion before we consider the results produced:

a) Collected data quality

During the automated data collection several problems occurred. A memory issue with the routine meant that as it processed more data, available resources on the machine began to reduce until a critical point was reached; resulting in a crash. Whilst this did not affect the data already collected, until the machine was restarted no additional data could be collected.

Initial attempts to remove the problem from the routine had no success and in view of time constraints I decided that the only workable solution was to restart the routine every 3-4 days; which generally was sufficient. Unfortunately over busy weekends when lots of news items occurred, or over University holiday periods where access to the computer was not possible, it would crash and there would be a period of time before it could be restarted. This is why gaps appear at certain periods in the data files; and whilst I do not believe that these gaps have any negative effect on the results, they did force restrictions on when pause periods could be implemented.

In addition, once the data had been collected, the cleansing process revealed an issue with share update information. Share data collected after 1200hrs on any given day reverted to a time base of 0001hrs, an offset of 12 hours. Examination of the collection routine did not reveal any issues, suggesting that the problem was either with the received XML file or more likely a discrepancy with time zone conversions.

Having identified the issue I was able to use Microsoft Excel to identify the faulty times and correct them, reducing any affect they may have had on the results. This should explain the discrepancy seen in the raw files contained within the Appendix pack.

b) Consistency checks

Due to the unexpected volume of data, and the time taken to process it, time constraints became the biggest obstacle for testing. For this reason I had to disregard my planned test to look at share predictions after key news events, but more importantly I was only able to conduct my other tests a single time.

This has a considerable impact on the results and conclusions that can be drawn from them. Without any check to prove consistency it is possible that these results are one-off, or flawed. To be considered a scientific evaluation the tests should be repeated at least 3 times, and preferably under different environments (different testing data). For this reason my evaluation and conclusions are only able to support my hypothesis, and not prove it.

Whilst completely undesirable, the ambitious nature of the project and the volumes of data concerned left no alternative; and this only became clear once testing had commenced. To replicate the tests concurrently in the time available a significant uplift in hardware would have be required, which was not practical. I discuss this in further detail in my evaluation in the next chapter.

When testing, predictions were exported from the forecast server at each pause point. These predictions have been combined for analysis, however the raw exported data can be found in Appendix C of the Appendix Pack.

They contain comma separated files for each share at each point, detailing the current state, and predicted future states. The future states include:

- Time in milliseconds until the predicted change
- Predicted Change
- Probability

Annex A contains a summary of all the predictions returned, and their direct comparison to actual movements. It is this data that has been used to determine the following results in the next sections.

Each prediction is compared with the next available actual movement that is either equal to or greater than the predicted time span.

5.2. Predictions

At each pause period the forecast server was instructed to produce a prediction by navigating its forecast tree. The root node for the forecast tree would always be the last percept received for that sense, with any subsequent nodes being future predictions.

If the prediction only consists of a single entry matching the root node, the forecaster has been unable to produce a future prediction. If the prediction output contains more than a single entry, it has been able to produce a future prediction.

	December		January	
	Shares Only	Shares & News	Shares Only	Shares & News
BARC	100%	90%	89%	83%
HSBA	100%	80%	83%	67%
LLOY	80%	70%	67%	67%
RBS	90%	90%	78%	72%
STAN	100%	100%	83%	78%
Predicted	***94%***	***86%***	***80%***	***73%***
Not Predicted	***6%***	***14%***	***20%***	***27%***

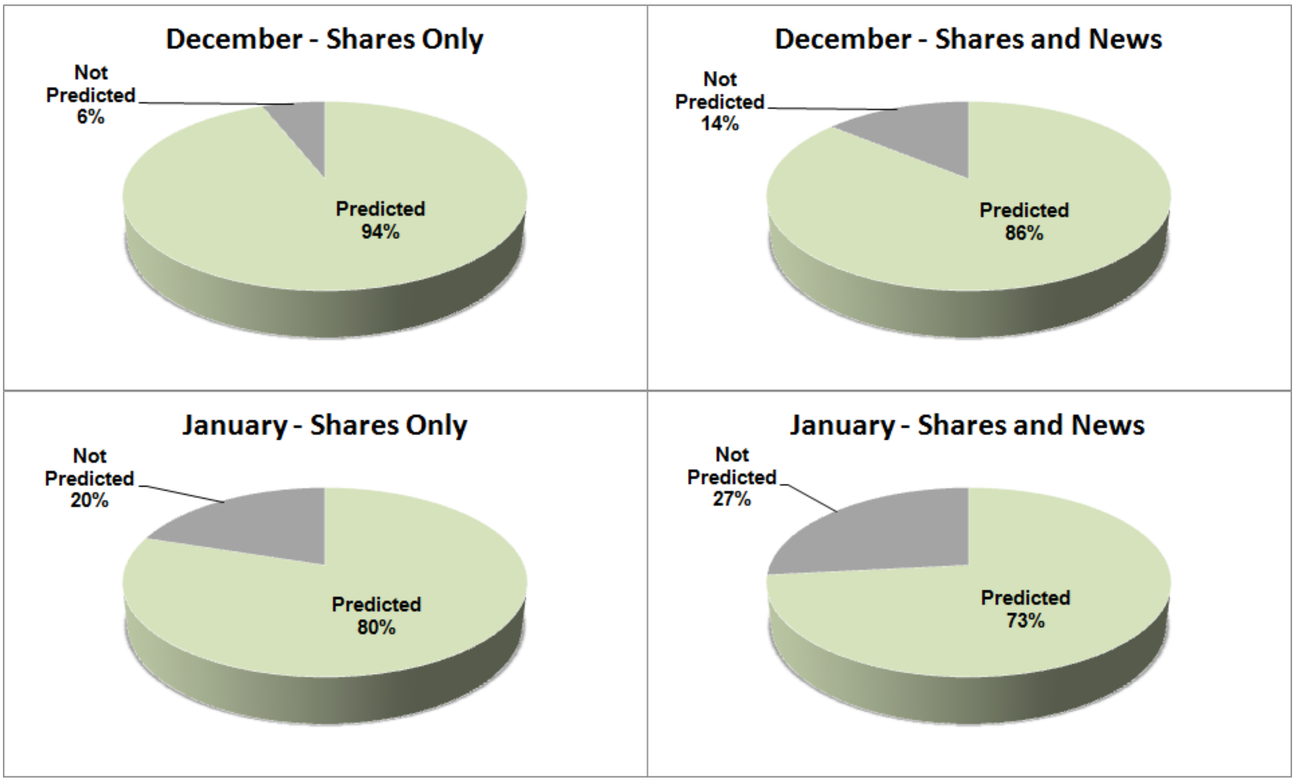

Figure 19 Prediction results and comparison for December and January

These results show that in both cases the rate of prediction was high, but with a noticeable reduction in the prediction rate when using both share and news data. They also suggest that as the system becomes more experienced (has processed more data), the rate of prediction reduces. This is shown both in the difference between tests, and the differences between December and January results.

If we calculate the mean average for the prediction results we get:

Mean average	Shares Only	Shares and News
Predicted	***87%***	***80%***
Not Predicted	***13%***	***20%***

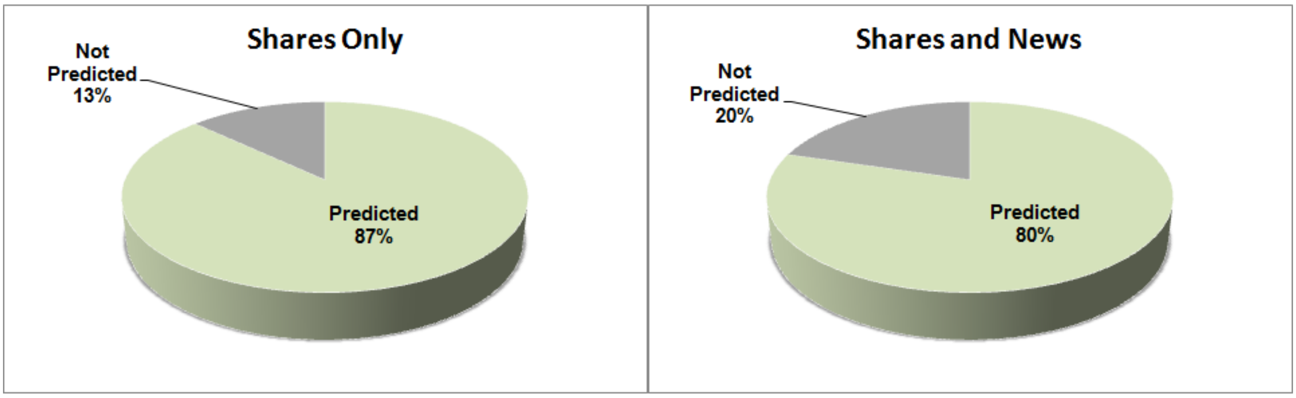

Figure 20 Mean average prediction results and comparison

These support the claim that as the system becomes aware of more data, the prediction rate is reduced.

5.3. Directional accuracy

At each pause point *__where a prediction was produced__* the results below compare the direction of movement predicted by the forecaster, and actual movements. The results are grouped under one of the following classifications:

i. Correct – Direction and magnitude of movement were exactly equal.

ii. Direction – The prediction was in the correct direction, but the predicted movement magnitude was incorrect.

iii. Marginal – The prediction was exactly 0 (no movement direction), and the actual movement was less than 0.1% of the share value (in either direction).

iv. Incorrect – Predictions that are incorrect in both direction and magnitude; and are outside of the permitted 0.1% margin of error for non-movement predictions.

December	Shares Only				Shares and News			
	Predicted	Correct	Direction	Marginal	Predicted	Correct	Direction	Marginal
BARC	*10*	0%	50%	0%	*9*	0%	33%	0%
HSBA	*10*	0%	30%	30%	*8*	0%	25%	50%
LLOY	*8*	0%	25%	38%	*7*	0%	14%	29%
RBS	*8*	13%	38%	0%	*9*	0%	67%	0%
STAN	*10*	20%	10%	0%	*10*	0%	60%	30%
Correct	***7%***				***0%***			
Direction	***31%***				***40%***			
Marginal	***14%***				***22%***			
Incorrect	***50%***				***38%***			

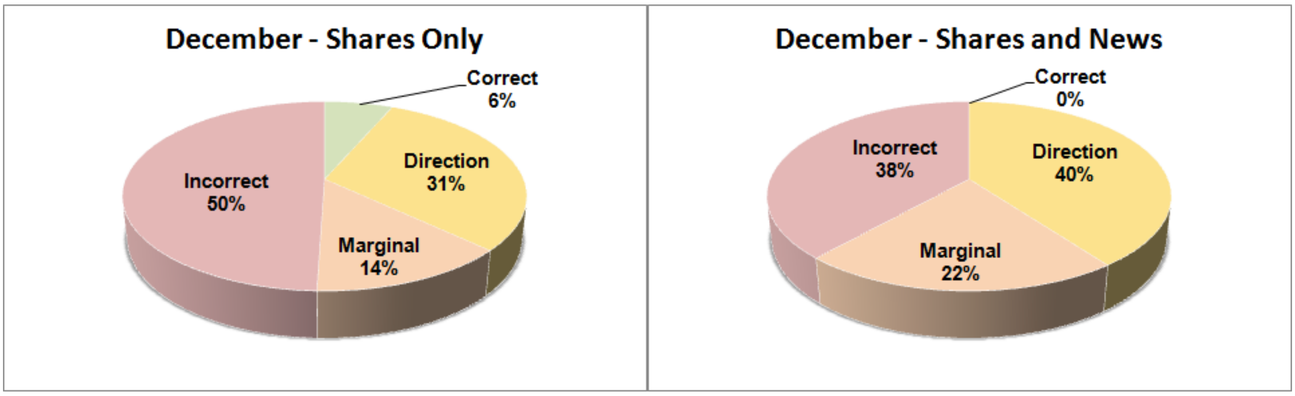

Figure 21 Directional accuracy results and comparisons for December

The results for December show an increase in accuracy when using share and news data combined, despite a drop in the number of exactly correct predictions.

January	Shares Only				Shares and News			
	Predicted	Correct	Direction	Marginal	Predicted	Correct	Direction	Marginal
BARC	*16*	6%	56%	6%	*15*	7%	47%	13%
HSBA	*11*	9%	18%	9%	*11*	0%	27%	27%
LLOY	*12*	0%	25%	33%	*12*	0%	17%	25%
RBS	*14*	0%	29%	21%	*13*	0%	38%	8%
STAN	*15*	0%	33%	0%	*15*	7%	20%	33%
Correct	***3%***				***3%***			
Direction	***32%***				***30%***			
Marginal	***14%***				***21%***			
Incorrect	***51%***				***46%***			

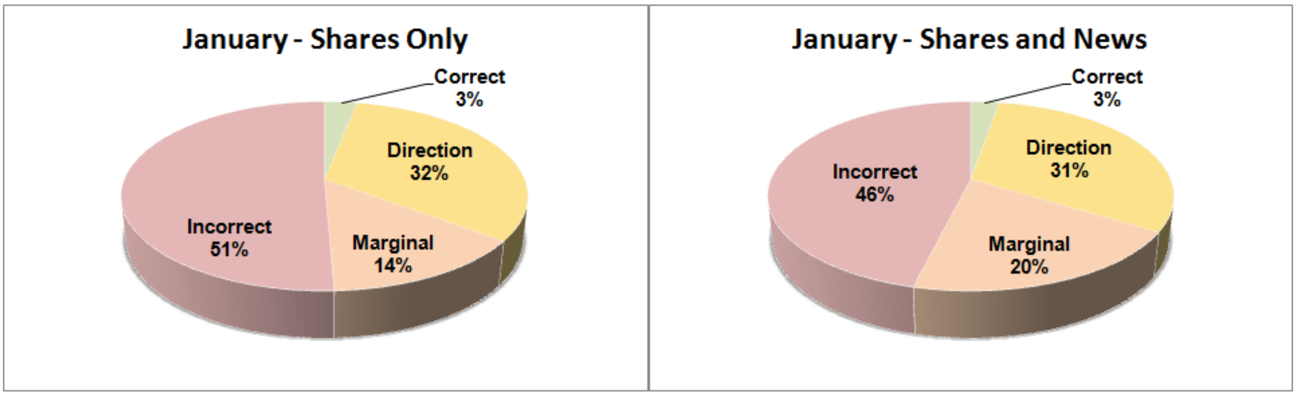

Figure 22 Directional accuracy results and comparisons for January

Supporting the results from December, the January results also show an improvement in accuracy when using share and news data combined. This improvement is less pronounced in the January results.

The most notable observation from these results is the increase in the number of incorrect predictions between December and January when using the combined data.

Also of note is the virtually identical nature of results between December and January when using only share data.

In both tests the results categorised as Marginal remained similar between December and January.

If we calculate the mean average for the prediction results we get:

Mean average	Shares Only	Shares and News
Correct	***5%***	***1%***
Direction	***31%***	***35%***
Marginal	***14%***	***22%***
Incorrect	***50%***	***42%***

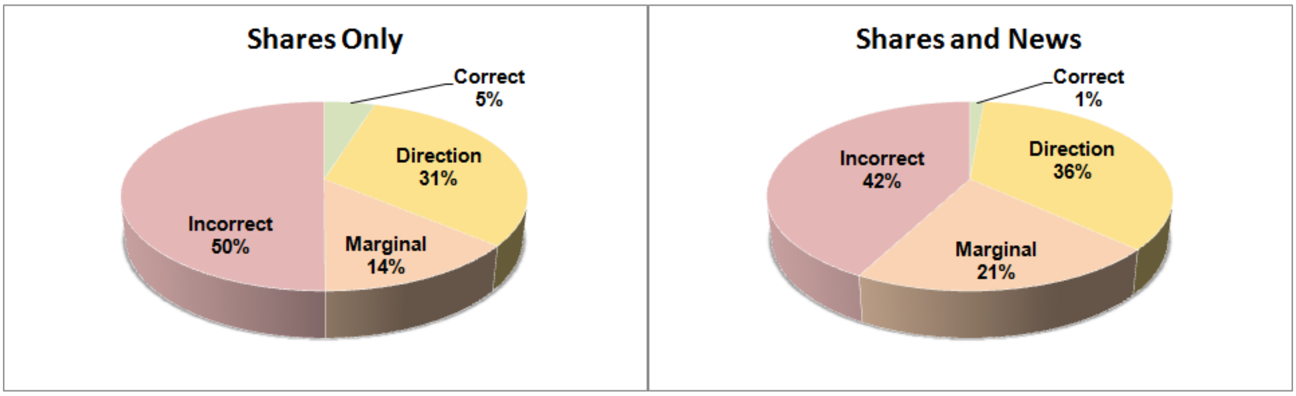

Figure 23 Mean average directional accuracy results and comparisons

These results show that the combined use of share and news data improves the overall accuracy of predictions. Furthermore the improvement seems to be the result of more predictions falling into the marginal category, with correct and directional predictions (if considered as a combined entity) remaining almost constant.

5.4. Accuracy, Improvements and Learning

In the previous section accuracy was considered as a category based on predicted and actual direction of movement; however it is possible to consider the relationship between the actual and predicted values directly. Using the linear difference between the two values does not allow for comparison between each of the shares, and therefore to measure accuracy I have normalised the data as a percentage of the share value – taken at the time of the prediction.

By converting the actual and predicted values into percentages we can then consider the variation between the two:

- A 0% variation means the prediction was exactly correct.
- A positive variation (> 0%) indicates that the prediction was in the correct direction of movement, with the percentage representing the order of magnitude that the prediction was out.
- A negative variation (< 0 %) indicates that the prediction was in the wrong direction, with the percentage representing the order of magnitude that the prediction was out.

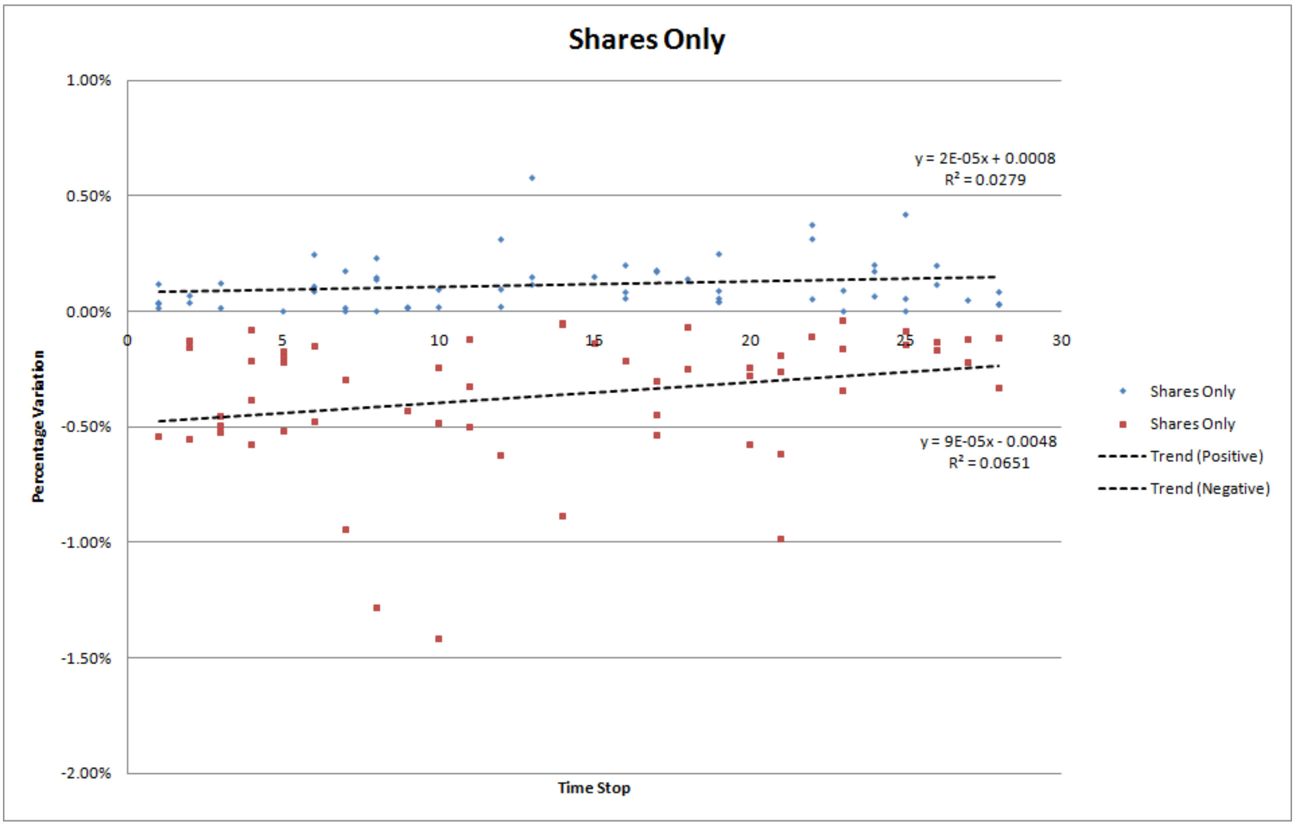

Figure 24 Variation of predicted changes from actual changes when using share data only

Looking at the trend data from these results we can see that the best-fit trend line for the negative results is improving with experience, drawing closer to 0. The results in the positive direction are moving away from 0, contradicting the assumption of learning; however the gradient is significantly less.

Coupled with this it is worth noting that the positive trend line has an R^2 value significantly less than the negative trend line, suggesting the negative trend line is a more accurate representation. In both cases the R^2 value is particularly low, suggesting that before any conclusions can be drawn a greater number of results will be required.

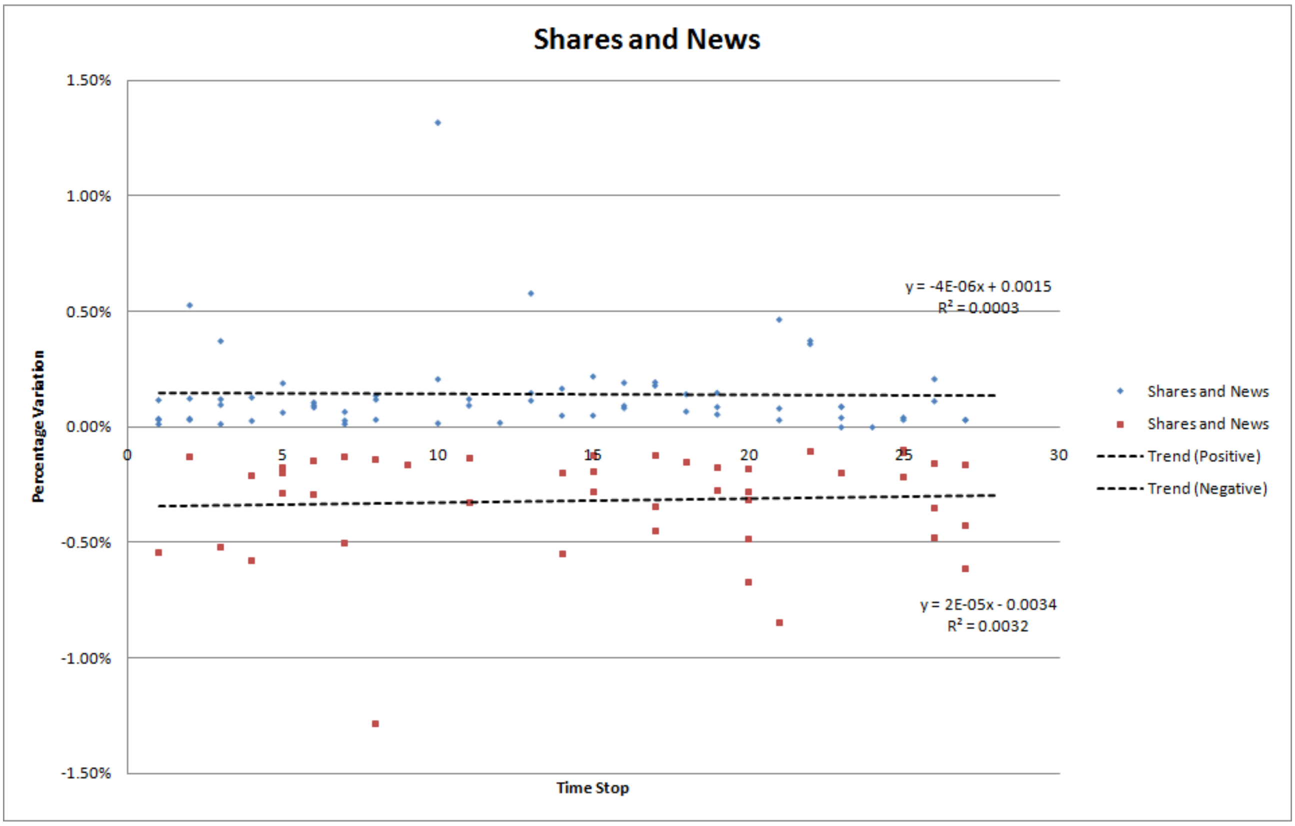

Figure 25 Variation of predicted changes from actual changes when using share and news data

When using the combination of share and news data both trend lines tend towards 0. In both cases the gradient is less than with share data alone, to the extent that the positive trend line appears to be almost parallel with the axis.

Whilst appearing to support the assumption of learning and be an improvement on the share only results, the R^2 value for both the positive and negative trend lines are reduced, suggesting that the trend line is a less accurate representation.

The associated data and graphs used to compare the variations can be found in Appendix D of the Appendix Pack.

CHAPTER 6 - EVALUATION

Having presented the results from testing, in this chapter I will begin by interpreting the results and justifying the conclusions I believe can be drawn from them. In each case I focus from the perspective of the original aims of the project, and the hypothesis. Having reviewed the content of the results, I then consider their context.

I then move to a more generic evaluation of the project, and my approach to it. For familiarity I have chosen to divide this into each stage of the software engineering lifecycle.

This begins with research and specification, where I look at the direction I took and the areas that could have been explored in greater detail.

I then discuss the design elements of the project, focusing on the neural network model and changes that I would make in future developments. These include changes to the neuron model, percepts, and the supporting architecture. In each case I have been as critical as possible to provide a platform for improvement.

In writing this document the details of my actual implementation has been mostly omitted to ensure brevity and avoid confusing the main approach and concepts with the practical implementation issues. At this point however I will briefly consider some of the implementation problems encountered, the issues they caused, and recommendations that would provide for improved performance and results. This can be read in conjunction with the project source code located in Appendix E of the Appendix Pack.

Finally I consider testing; both the methods used and the quality of the data. I make critical assessments on the reliability, accuracy and appropriateness of the results. At all times I consider how this effects the conclusions that can be drawn, and suggest improvements and changes.

6.1. Results

Before commencing with an evaluation of the test results it is worth recapping on the hypothesis that the project set out of prove:

> *"Provided with sufficient awareness of its environment, an intelligent computer model could identify patterns within a complex system, forecast future states based on those patterns; and learn and develop those forecasts to improve accuracy based on experience".*

From this we can draw out the two key elements: the forecasting of predictions, and improvement with experience (or unsupervised learning).

6.1.1 Forecasting of predictions

Looking at the rate of predictions we can clearly see that the model is able to identify patterns. This applies when using share data independently, and when combining it with news data. More interesting is the decline in the prediction rate as more data and experience is available. This initially appears contrary to the hypothesis, and against common thinking. I believe this is not the case though.

Share movements are chaotic, and the model is only able to identify patterns when sequences of environmental states are repeated. For example, after the first occurrence of a pattern we could, in simple terms, state:

$$a \rightarrow b \rightarrow c$$

Any time the model encounters a again, the forecaster will predict b followed by c; and the probability will be considered certain as it has no alternatives to compare against. If the environmental states are encountered again with differing results, the certainty begins to become diluted. After 4 instances of the same state we could have:

$$a \rightarrow x : x \in \{b, c, d, e\}$$

As a state is encountered more frequently without consistency in outcome the possible resultant states increases, and therefore the probability for forecasting decreases. Coupled with the possibility of the wider environment being different each time the state is encountered, it is likely that the probability of future states will drop, until it is below a threshold where the forecaster is unable to produce predictions.

The results therefore imply that instead of providing a widely inaccurate or unlikely event, the forecaster does not produce a prediction. In reality this is far more useful than speculation.

For the testing conducted the threshold was set at a probability of 5%. Any lower would result in a non-prediction (the neuron would fail to perpetuate firing due to the low signal).

Unfortunately this is only my assumption, as there is insufficient evidence to fully support the conclusion. Additional testing that varies the threshold level would provide interesting comparison, coupled with several more months' worth of data to see if the trend continues.

6.1.2 Improving accuracy based on experience

To demonstrate unsupervised learning the test results would need to reflect an increase in accuracy as time progressed; and this is best applied using a best-fit trend line.

A trend line that tends towards 0 as time increases shows improvement that could be interpreted as learning. Due to the positive and negative trend lines representing directional accuracy, this means that negative trend lines should have a positive function, and positive trend lines should have a negative function.

Reviewing the negative trend lines for both tests reveals that as time progresses those predictions that were directionally incorrect have a reduced variation.

In contrast the positive trend lines for both tests are not as conclusive. When using share data only the trend line actually increases, suggesting that predictions get worse. When combined with news data the trend line does tend towards 0.

From this we can say that the results using shares and news data support the claim that the neural network model is learning from experience; and that using share data only the results are inconclusive either for or against the claim.

In both cases the poor R^2 value makes any claim questionable. The removal or changing of only a few points that may be discrepancies has drastic effects on both the trend line function and the R^2 value.

In view of this it is not possible to state if the model is supporting unsupervised learning or not.

I can conclude that the results do not disprove or discount the possibility that learning is taking place; in fact they add support to the assumption. This support is however not sufficient to make any sound conclusions.

6.2. Research and specification

Initially the nature of this project was undefined; nothing more than a suggested area of investigation that had mutated several times after various discussions. My Initial research focused on the financial markets, share movements, and the forces that drive them. As the project progressed it quickly changed to be neural network focused, with the financial markets simply providing an environment for testing.

With time constraints being prevalent, this consequentially led to less research being conducted in other areas. With the benefit of hindsight this would have been more beneficial. The key area that I believe required more research was biological neuroscience – the structure and function of biological neurons.

Whilst I believe that the concepts contained within my neural network model are sound despite my lack of knowledge in neuroscience, subsequent research that I have completed in these areas (since developing my neural network) has highlighted particular changes that I would make.

6.2.1 Revised axon design

In my model neurons contain multiple axons, each representing the occurrence of the percept under different environmental states. As a neuron fires a process iterates through the axons, calculating the association between the current environment and the axons environment to determine signal strength. This signal strength is then transmitted to the subsequent neurons. This means that for a neuron with x axons taking t seconds to fire, the total firing time T is:

$$T = xt$$

As the number of axons increases, the total time required to fire the neuron increases, delaying the firing process and slowing the system. Key neurons that are encountered frequently are therefore drastically slowed whereas the subsequent neurons that may only occur infrequently fire optimally.

Biological neurons operate differently. Although we do not know if or how they store environmental states, we do know that each neuron only has a single axon. These axons are joined by dendrites that receive signals for the subsequent neuron. Using this approach, an improved model would be for the environmental state to relate to the subsequent neuron, not the present neuron.

When firing a neuron should transmit the signal to all its subsequent neurons, with the present environmental state embedded within the signal, firing in constant time.

The subsequent neuron receives the signal and associates the signal with the most appropriate dendrite, based on the environmental state. Coupled with a change in how environmental states are maintained this would dramatically improve firing performance.

6.3. Design

At a very early stage I decided the development strategy that I would use for the project. For the most part this was forced on me by circumstances and the nature of the development: the conceptual nature of the project; my unfamiliarity with the development areas (TCP/IP socket programming and serial transmission of objects; Distributed concurrency; massively-multi threaded systems, Object database integration etc); my skills and personal preference in development style; and my previous experience in development environments.

As such I opted for a Rapid Application Development (RAD) approach: preferring the evolution of a prototype from initial concepts instead of a top-down pre-designed approach. I do not believe that I would have achieved success under any other approach; however this does not mean that design flaws are not present in my development.

One of the key areas that I believe to be flawed is the manner in which environmental states are stored. A more thoughtful design process may have highlighted this.

6.3.1 Revised percept design

Percepts are objects representing individual states in the environment from sensors which can be introduced and removed from the overall picture at any point. The wider environmental state is represented by percepts from any number of sensors that exist at the time; stored as an array of percepts. The handling of percepts as objects provides generic capability, but adds management overheads. The array of percepts also adds complexity. Both of these place a drain on performance. I am certain the design can be improved, and a possible example of this follows:

For most sensors there are a limited number of variations, for example share movements. These are rarely greater than 2% in either direction.

We can also determine that we do not need to store any more than 2 decimal places of accuracy, leaving us with a range for the percept p of:

$$-2.00 \geq p \leq \ +2.00$$

This leaves a range of 400 distinct values, and using a mapping we can associate each value with an index number: -2.00 becomes 0, -1.99 becomes 1 and so on. If we convert this to binary, we could always be assured to represent the percept in 9 bits (511 possible distinct values). A share movement of -0.15 would be represented as:

$$-0.15 \ \rightarrow decimal(186) \ \rightarrow \ binary(010111010)$$

Representing the percept as a number makes management, processing and transmission far less complex, and significantly quicker. This does not detract from a generic approach, because each percept would simply require a translational mapping to convert indexed values to actual states.

We can now consider the wider environment with the introduction of news data. Using the composite neuron approach already implemented news articles are comprised of paragraphs, sentences, words and ultimately letters. A letter percept, if we ignore case, would only require 26 distinct values to represent the basic modern Latin alphabet, or 5 bits: ‘a’ becomes 0, ‘b’ becomes 1, ‘c’ becomes 2 and so on.

$$'g' \rightarrow decimal(6) \rightarrow binary(00110)$$

If as sensors are introduced they are provided with an offset value, which once set is always applicable, we can combine the percept numbers and still be able to determine each individual state from the environment. For this example share movements would have an offset 0 to 8, whilst letters are 9 to 13.

In simplistic terms we can therefore represent the above environment as the decimal number 3258:

Percept: Letter					**Percept: Share Movement**								
‘g’					**-0.15**								
8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
0	**0**	**1**	**1**	**0**	**0**	**1**	**0**	**1**	**1**	**1**	**0**	**1**	**0**

This is a simplified view of the concept, and skips over some of the issues that do arise, but the principle applies.

The benefit to performance and the reduction in complexity would be highly advantageous, whilst the expandable and generic approach to the environment is not compromised.

6.4. Implementation

Under a RAD approach implementation is an evolutionary process from prototype to final product. Improvements can almost always be made, but at a point in the cycle – normally dictated by time or financial constraints – improvements cease. I also experienced this. With improvements that I wanted to make, but impending deadlines and time constraints, I had to finalise development and proceed with testing.

Whilst I accept that the conceptual nature of the project means that the implementation will be more akin to a developed prototype, there are several areas that could be modified to improve performance and reduce complexity. These only came to light once testing had begun, and although they do not alter the results generated, they would make for a more efficient and robust system.

6.4.1 Neuron Server

The concept of a neuron server was simple, and seemed to evolve from my neural network model: Many neurons on a single machine require a central agent to supervise and manage them. Whilst this approach has worked, decentralisation is both possible and has advantages. Each neuron operates via an independent thread, and therefore is capable of managing itself. The neuron server adds a level of complexity, and hence performance delay, which is not required.

A small residual program would still be required to accept neurons from the directory server and instantiate them on the host machine, but once instantiated the neurons could directly communicate with the directory server.

An independently processing neuron would be able to detect available TCP/IP ports and establish using them. It could notify the directory server which port it can be contacted on. The neuron would then be able to accept its own triggers directly without the need for a queue at the neuron server, and finally when terminating the neuron could pass its new data back to the directory server and self-terminate its thread.

Removing the neuron server would not only improve performance, but opens the possibility of a peer based distributed system where idle processors can elect to accept neurons, and then refuse them when busy. This approach has been used in the past for processing large volumes of data, taking advantage of many idle machines across vast networks or the Internet, and would be ideal for the nature of my neural network.

6.4.2 Recoverability

Considering the challenges and difficulties present with this project the final solution is relatively robust and reliable; but only having overcome many issues and problems. Many of the problems encountered are not due to programming faults, but instead known issues with an element of the development environment such as the JVM or MySQL Connector. Particularly problematic were memory leaks, suspected to be caused by the MySQL connector; and the thread priority given to the Java garbage collector and its operational quirks.

Due to the time requirements for processing data, failures of the system caused significant delays to the project, and the key area that I would wish to address for future developments would be the ability to save the state of the system at more frequent intervals. Presently neurons are saved to the database when they terminate, after a pre-defined idle time. Should a neuron be frequently called, it could remain distributed and not returned to the directory server for a long period of time. Any failure of the system before the neuron returns results in new axons that have been created being lost.

Pauses defined in the input file allow for execution to stop and all neurons to be relinquished back to the directory server, but this involves user interaction, and is slow.

To be truly robust the system should be able to recall neurons after a set period has expired, even if they have been recently fired. This would ensure that only a small amount of data is lost due to a failure, but does add a performance overhead to the system.

Additionally the detector maintains current and previous states of the environment, critical for the generation of axons. Should the detector fail there is no way to reinstate the environment, and instead data resumes from a blank environment. This has the potential to disrupt key chains of subsequent neurons. To remedy this possibility the detector should implement a persistent approach for storing present and previous environmental states.

6.5. Testing

Of all the elements of this project the testing stage has been the most disappointing; and with hind sight would be subject to the most change. Arguably too ambitious for an academic year, insufficient time was available for both the collection and processing of test data. The issues encountered thus fall broadly into two categories: The quality and availability of data, and the number and range of tests conducted.

6.5.1 Quality and availability of data

Despite prioritising the development of an automated data collection routine the process did not commence in earnest until November 2009. Under pressure to begin testing to allow sufficient time for evaluation of the results the collection had to cease at the end of January 2010. With only 3 months worth of data, including the overlap of the Christmas period, it was apparent from the beginning that there would be insufficient data to produce meaningful conclusions.

Allowing for periodic cycles and fluctuations I would estimate that to draw confident conclusions testing would need to commence on at least 2 years worth of data. This was identified early on in the project, but a search for suitable data already available was not successful.

Historic news data is available through many sources, but in most cases as free-form text and not in a downloadable, easily importable format. Historic and archived news also tended to be specific to the host provider and not centralised by a single generic provider.

Share data is more widely available, and kept in formats that can be imported easily. Unfortunately these sources also have drawbacks. Freely available information is available on a daily summary basis (taken at close of business). These movements would allow a neural network to predict day-end states, however to process similar amounts of information as the present system data for many years worth would be required. In this instance related news data would not be available and the hypothesis would not be testable.

Detailed share movement data covering hourly changes is also available, but at significant cost. Enquiries to the various sources including the LSE highlighted that whilst basic data could be provided to students for free, the detailed data was prohibitively expensive.

Faced with this knowledge the only available option was to proceed with collecting the data personally, understanding the restrictions this would incur and the effects to subsequent conclusions. In response that is why this projects focus is more towards the conceptual nature of the neural network, and not the accuracy of the predictions.

6.5.2 Number and range of tests

Although I had identified that the volume of data available for testing would be lacking, I had not foreseen the length of time that testing would take. Below are some approximate calculations that show the extent of the problem encountered:

The data files containing share and news data between November 2009 and January 2010 comprise approximately 40,000 records. To facilitate the automated import of these records a sensor reads a line from the file, and then waits a period of time to complete processing before commencing with the next line.

Share movements can be processed using a single neuron transaction (either successfully finding the neuron in the database or creating a new neuron; and then creating an axon and generating a trigger), whereas a news article requires splitting to its composites: sentences, words and then letters. Each composite element requires at least a single neuron transaction.

In general we can assume it takes 1 second to process a single neuron transaction and submit to the directory server. In addition we know there are other processing overheads, which to avoid overloading must be considered as an additional 1 seconds per transaction. This is before we have even considered allowing time for the directory server, neuron servers and forecast server to process future predictions, pauses for backups, and outputting predictions. The result in reality was a requirement for at least 3 seconds between the importing of records, allowing for busy periods processing groups of news data (time periods where the LSE is closed and the only data is news).

A quick breakdown of the records reveals approximately 24,000 share movements and 16,000 news articles. Analysis of the news articles reveals that, on average, they are comprised of 105 letters.

We can use these figures to calculate the minimum required time to complete a single test:

$$(24{,}000\ shares\ * 3\ secs) + (16{,}000\ news * 105 * 3\ secs) = 5{,}112{,}000\ seconds$$

$$\frac{5{,}112{,}000\ seconds}{60} \sim 85{,}200\ minutes$$

$$\frac{85{,}200\ minutes}{60} \sim 1{,}420\ hours$$

$$\frac{1{,}420\ hours}{24} \sim 60\ days$$

These calculations refer to the combined test for share data and news. Admittedly when testing share data only the test can be performed in reduced time, but this does not detract from the excessive requirement for testing time.

In view of this I was unable to repeat the testing conducted, or complete variations on the testing as originally planned. Additional hardware would have enabled concurrent testing, but at a level that would prove to be impractical.

If the project were to continue I would recommend, as a minimum, the following additional tests be completed, preferably in conjunction with additional data as discussed previous:

i. Confirmation tests.

   The repeat of tests already conducted at least 3 times to provide a measure of consistency in results and confirm conclusions.

ii. Key event monitoring.

   Instead of a blanket check at 1100hrs and 1500hrs each day during testing I would conduct checks before and after key news events, as initially planned.

iii. Different environment.

   I designed my neural network to operate generically, in any complex environment. To truly evaluate the networks performance it would require testing in multiple environments. Examples of additional environments where comparisons could be made are: Metrological data; Traffic flow; Usage statistics.

Finally there is one key area of analysis which I have been unable to perform but would be essential before any conclusions on the performance of the neural network can be drawn. Operating the new neural network model in parallel with an existing neural network, using the same or similar data, would provide an excellent means of performance comparison.

CHAPTER 7 - CONCLUSIONS

In my final chapter I look to draw the project, and this document, to a close. I review the successes that have been achieved, and the areas that have fallen short of initial expectations.

I begin by examining the initial aims and hypothesis of the project, and considering if they have been achieved and to what degree. I review their appropriateness, how my results and evaluation support them, and changes that should have been made.

With most projects, final direction often deviates from plans made at the start - and this project is no exception. The original focus on accurate share movement predictions shifted to neural networking concepts, and so I consider the wider achievements of this project and some of the lessons and conclusions that can be drawn from its approach.

Finally before closing with a summary of the project I recap on some of the recommendations made in the evaluation, suggesting future improvements and additional work that could lead to greater achievements.

7.1. Achieving the Aims

From the outset this project had 4 specified aims. These I developed as measurable targets that I felt, if achieved, would ensure a system capable of testing the hypothesis. They were deliberately worded to provide direction in the development of the system, whilst sufficiently generic to not narrow the projects scope.

If we consider each of these aims in turn:

7.1.1 Gather and store News and Share price data

> *"Gather and store, in real-time, News and Share price data from multiple, disparate sources*;"

The first aim was easy and quick to specify, as very early into the project it became clear that a pre-requisite to any implementation would require quality test data.

This aim was, arguably, achieved. The automated collection routine, once provided with the locations of online data feeds (XML), extracted data in real-time and committed it to a database. The data collected between November 2009 and January 2010 provided the basis for all the testing and evaluation performed during this project, and if left running would be able to continue to do so until stopped. Although not required for this project, the development of the collection routine also provided for direct connection to the neural network, creating the possibility for real-time processing from data retrieval to prediction.

Despite this apparent achievement, the routine was not without fault. A memory leak which I was unable to specifically locate caused the routine to crash after any more than a single weeks processing, requiring constant supervision and resetting. The result was gaps in the data collected, ranging from a few minutes to several days depending on when the crash occurred and my availability to reset the routine.

The routine was also dependent on pre-definition of the XML locations and formats. For the duration of this project that was sufficient, but for longer term processing a routine capable of determining the format of XML feeds automatically would be advantageous. This would ensure that changes made by the feed provider would not cause failure in the routine.

The ideal solution would be a collection routine capable of automated data feed location, and self definition. This would enable the routine to be set and left indefinitely, but was far beyond the scope for this project.

7.1.2 Allow for pattern recognition and identification

> *"Store this data in a format that allows for pattern recognition and identification both within and across the sources;"*

In my neural network model storage of data for recognition and identification relates to the neuron, and more precisely the percept concept that is associated with it. For this reason the aim has been achieved in nearly all respects.

Where the current percept implementation falls short is with non-exact match identification. The process of using unique hash codes to recognise repeat percepts is suitable where exact matches are encountered, but with this method the slightest variation in a percept is treated as unique.

In my evaluation I have already suggested improvements to the manor in which percepts, and the wider environment, are represented. Using a binary offset would improve performance and reduce complexity, but in addition a method of identifying similar percepts would be useful.

7.1.3 Produce future trend forecasts

> *"Produce future trend forecasts based on identified patterns and computed logic;"*

Using the results obtained my neural network was able to produce forecasts and predictions, in the majority of cases. As the amount of data available increased the rate of prediction decreased, but over the volume of test data used for this project the rate still remained high (above 70% in all tests).

Considering the conceptual nature of the development it was entirely possible that no forecasts would be produced, and therefore I believe the aim has been achieved.

Less clear is, reviewing the semantics of the aim, the effect computed logic had on the forecasting.

Although only implementing two logic rules (many more could have been specified), the effect they had is unknown. Had time permitted it would have been useful to repeat testing with the logic rules disabled and enabled; possibly even building up each logic rule per test to see its individual effect.

7.1.4 Improve accuracy based on feedback and experience

> *"Improve the accuracy of forecasts over time based on feedback and experience."*

The final aim is the most ambitious, but also the most important when considering if the neural network is capable of learning (and therefore intelligent). Evidence of unsupervised learning would support further development and provide exciting opportunities for other areas of application. In contrast without the ability to learn a neural network shows little potential, and warrants no further investigation save to examine the mistakes made.

Unfortunately I am unable to make a conclusion in either respect. As discussed in the evaluation of the results, the data suggests the presence of learning to some degree, suggesting the aim has been achieved. Certainly the data does not suggest that learning is not taking place, and therefore does not imply that the aim has not been achieved.

I must caveat this by stating that insufficient test data was available to draw meaningful analysis from the results.

In summary I would therefore conclude that my first 3 aims have been achieved, although each has areas for improvement.

The final aim has been supported, but not to the extent that a conclusion can be drawn.

Each of the aims of the project was produced to support and direct development towards proving my hypothesis:

> *"Provided with sufficient awareness of its environment, an intelligent computer model could identify patterns within a complex system, forecast future states based on those patterns; and learn and develop those forecasts to improve accuracy based on experience."*

With the questionable state of my final aim I must also conclude that the hypothesis has not been proven. Instead it has been supported by the results obtained through the project; requiring further study to reach any solid conclusions.

7.2. Neural network concept

As the project developed my focus changed. Initial motivation towards predictions and their accuracy was replaced by the conceptual nature of the neural networks.

In my opinion the most significant element of the project was the move from a process-based neural network to a communications-based neural network. The potential advantages with such an approach made the obstacles it presented worth tackling. My main aim therefore became demonstrating that a communications-based neural network could be implemented, and would achieve results comparable with existing neural networks without the constraints and limits that bind them.

Although I have not had sufficient time to compare the communications-based neural network with existing neural networks, I can conclude that the concept is achievable. Communications-based neural networks, distributed over many machines, can produce forecasts and therefore presumably can complete other tasks that neural networks are capable of.

Whilst thresholds were in place during my testing to maintain performance, none of the connection limits that are imposed to many existing neural networks, such as constrained hidden layers, were present.

In a performance comparison I am certain that existing neural networks would probably out perform my model at present. That said, I consider my current model to be only a proto-type. The improvements suggested in my evaluation would likely see a dramatic jump in performance. Additional design effort, and greater research into biological systems, would yield even greater increases.

Despite having insufficient results to prove my hypothesis and that greater environmental data tends towards a more intelligent solution, I believe this project has demonstrated that environmentally aware neural networks are achievable. Pre-defined logic and programming can be replaced which, in the future, has the potential to allow for generic neural network applications.

7.3. Continuation

I believe that this project has successfully demonstrated the potential of communication-based neural networks that consider a wider environmental picture instead of pre-defined programming. Unfortunately the lack of test data and results stops me from making solid conclusions, and therefore further work is required to develop the ideas discussed in this project.

Should this further work be undertaken, there are several recommendations I would make:

i. Additional testing

   Before making any modifications to the system I would strongly advise that more testing be conducted. My evaluation discusses tests that should be conducted, including confirmatory tests for those already performed and new tests to support existing results. This would, ideally, be complemented with larger, more complete data.

ii. Percept change

   Assuming the additional testing supports the conclusions and assumptions I have made, the next area to review would be the way in which percepts are stored, communicated and managed. Again this is discussed in more detail in my evaluation. The percept, and the environmental state, is key to every element of my neural network and therefore improvements in performance and functionality in this area will have cumulative effects for the network as a whole.

iii. Neuron server

   As the volume of test data increases the systems performance will become more critical. I am certain that a peer-based distributed approach that can take advantage of idle processors on many networked machines will provide a more appropriate supporting architecture than the neuron server concept. This will require both conceptual level changes to the communications model, and practical changes. Firewall and security issues which I largely bypassed through my sanitised network will become increasingly prevalent; and the robust handling of disconnected neurons will require greater focus.

7.4. Summary

The aims this project set out to achieve, in hindsight, were too ambitious for the time available. The complex nature of the problem; the unfamiliarity of the techniques and technologies; the volume of data required for testing and the time testing takes should have all been identified at the beginning of the project. More conservative aims and a revised hypothesis may have been more achievable.

Despite this I have been able to support my hypothesis, suggesting further testing would provide more concrete conclusions.

In addition to the original project aims I have challenged the design of neural networks; showing the viability of communications-based neural networks and the benefits they could hold for future developments.

I have also begun to show that environmentally aware systems are possible; and that pre-programmed understanding can be replaced by this awareness. In the future this has the potential to provide generic neural networks able to work in many application areas without modification.

This project almost certainly raises more questions than it answers. If considered as a solution then the results are disappointing. In contrast as a conceptual prototype the project provides support for the assumptions I have made, and a basis for further development. I have tried throughout this document to support this approach by being critical at all times and were possible suggesting future changes and improvements.

In final summary I would say that this project has been successful in developing a working prototype of a communication-based, environmentally aware neural network that challenges some of the concepts of existing neural networks. It provides evidence to support my hypothesis, and sound recommendations that warrant further investigation.

From a personal perspective the project has inspired me to learn more about biological systems and neuroscience; and has provided motivation to continue the work I have begun at some point in the future.

REFERENCES

Figures

Bibliography

1. Atlas, L.E., Conner, J.T., Douglas, R., 1994. "Recurrent Neural Networks and robust time series prediction". IEEE Transactions on Neural Networks, 5 (2), March 1994.

2. Baestaens, E., Max van den Bergh, W., Wood, D., 1994. "Neural network solutions for trading in financial markets". London: Financial Times.

3. Barto, A.G., Sutton, R.S., 1990. "Time derivative models of Pavlov ion reinforcement". In: M.Gagriel & J.Moore (eds) "Learning and Computational neuroscience: Foundations of adaptive networks". Cambridge, MA: MIT Press. pp. 497-537

4. Beale, R., Jackson, T., 1990. "Neural Computing: an Introduction". London: IOP Publishing Ltd.

5. Bear, M. F., Connors, B.W., Paradiso, M.A., 2007. "Neuroscience: Exploring the Brain". 4th ed. Lippincott Williams & Wilkins.

6. Elman, J., 1990. "Finding Structure in Time". Cognitive Science, 14. pp.179-211.

7. Heaton, J., 2008. "Introduction to Neural Networks for Java". 2nd ed. Heaton Research.

8. Hebb, D.O., 1949. "The Organization of behaviour: A neuropsychological theory". New York: Wiley.

9. Hopfield, J., 1982. "Neural networks and physical systems with emergent collective computational abilities". Proceedings of the National Academy of Science USA, 79, pp.2554-2558.

10. Judd, J.S., 1990. "Neural network design and the complexity of learning". Cambridge (MA): The MIT press.

11. Kohonen, T., 1984. "Self-Organization and Associative Memory". Berlin: Springer-Verlag; New York: Heidelberg.

12. Laron, K., 2004. "The Science of Word Recognition". Available at: http://www.microsoft.com/typography/ctfonts/WordRecognition.aspx (last accessed 03 May 2010).

13. Rosenblatt, F., 1958. "The Perceptron: A probabilistic model for information storage and organisation in the brain". Cornell Aeronautical Laboratory, Psychological review, 65 (6), pp.386-408.

14. McClelland, J., 2010. "Explorations in Parallel Distributed Processing: A Handbook of Models, Programs, and Exercises". Available at: http://www.standford.edu/group/pdplab/pdphandbook/handbook.pdf (last accessed 28 April 2010).

15. Norvig, P., Russell, S., 2003. "Artificial Intelligence: A modern approach". 2nd ed. London: Prentice Hall.

16. Sedgewick, R., 2008. “Algorithms in Java: Part 5 – Graph Algorithms”. 3rd ed. Addison Wesley.

17. Wiener, N., 1948. “Cybernetics: Or Control and Communication in the Animal and the Machine”. Cambridge, MA: MIT Press.

18. Whitby, B., 2003. “Artificial Intelligence”. Oxford: One World Publications.

ANNEX A - RESULTS

1. Share only results

	BARC								
Stop	Date	Value	Change		Predicted			Comparison	
			Pts	%	Pts	%	Prob	Variation	Direction
1	01/12/2009 10:56	296.6							
	01/12/2009 11:00	295.75	-0.85	-0.29%					
	01/12/2009 11:01				0.57	0.19%	38.81%		
	01/12/2009 11:03	295.55	-0.2	-0.07%					
			-1.05	**-0.35%**	**0.57**	**0.19%**		**0.55%**	**Incorrect**
2	01/12/2009 14:59	299.04							
	01/12/2009 14:59				0	0.00%	5.54%		
	01/12/2009 15:01				-0.08	-0.03%	1.19%		
	01/12/2009 15:03	299.35	0.31	0.10%					
			0.31	**0.10%**	**-0.08**	**-0.03%**		**0.13%**	**Incorrect**
3	02/12/2009 10:57	288.9							
	02/12/2009 10:59				0.6	0.21%	19.41%		
	02/12/2009 10:59				0	0.00%	7.85%		
	02/12/2009 11:00	289.15	0.25	0.09%					
			0.25	**0.09%**	**0.6**	**0.21%**		**0.12%**	**Direction**
4	02/12/2009 14:59	292.15							
	02/12/2009 14:59				-0.04	-0.01%	38.81%		
	02/12/2009 15:04	293.8	1.65	0.56%					
			1.65	**0.56%**	**-0.04**	**-0.01%**		**0.58%**	**Incorrect**
5	03/12/2009 10:57	311.6							
	03/12/2009 10:59				0.45	0.14%	19.41%		
	03/12/2009 11:00	311.5	-0.1	-0.03%					
			-0.1	**-0.03%**	**0.45**	**0.14%**		**0.18%**	**Incorrect**
6	03/12/2009 14:59	308.85							
	03/12/2009 15:01	308.55	-0.3	-0.10%					
	03/12/2009 15:03	307.45	-1.1	-0.36%	-0.64	-0.21%	58.22%		
			-1.4	**-0.45%**	**-0.64**	**-0.21%**		**0.25%**	**Direction**
7	04/12/2009 10:59	299							
	04/12/2009 11:01				-0.75	-0.25%	19.41%		
	04/12/2009 11:02	299.15	0.15	0.05%					
			0.15	**0.05%**	**-0.75**	**-0.25%**		**0.30%**	**Incorrect**
8	04/12/2009 14:58	309.15							
	04/12/2009 15:03	309.95	0.8	0.26%	0.35	0.11%	19.41%		
			0.8	**0.26%**	**0.35**	**0.11%**		**0.15%**	**Direction**
9	08/12/2009 10:57	293.75							
	08/12/2009 11:00	293.7	-0.05	-0.02%	0	0.00%	5.30%		
			-0.05	**-0.02%**	**0**	**0.00%**		**0.02%**	**Marginal**
	08/12/2009 11:04	293.55	-0.15	-0.05%	-0.15	-0.05%	18.20%		
			-0.2	**-0.07%**	**-0.15**	**-0.05%**		**0.02%**	**Direction**
10	08/12/2009 14:59	288.6							
	08/12/2009 15:00				-0.35	-0.12%	38.81%		
	08/12/2009 15:05	288.3	-0.3	-0.10%					

			-0.3	**-0.10%**	**-0.35**	**-0.12%**		**0.02%**	**Direction**
11	04/01/2010 10:57	278.84							
	04/01/2010 11:01	279.1	0.26	0.09%					
			0.26	**0.09%**					**None**
12	04/01/2010 14:59	279.3							
	04/01/2010 15:01	279.35	0.05	0.02%	-0.55	-0.20%	19.41%		
			0.05	**0.02%**	**-0.55**	**-0.20%**		**0.21%**	**Incorrect**
	04/01/2010 15:04	279.2	-0.15	-0.05%					
	04/01/2010 15:07				-1.1	-0.39%	6.71%		
	04/01/2010 15:09	279.4	0.2	0.07%					
			0.1	**0.04%**	**-1.65**	**-0.59%**		**0.63%**	**Incorrect**
13	05/01/2010 10:58	294.5							
	05/01/2010 10:58				-0.31	-0.11%	19.41%		
	05/01/2010 11:01	293.85	-0.65	-0.22%					
			-0.65	**-0.22%**	**-0.31**	**-0.11%**		**0.12%**	**Direction**
14	05/01/2010 14:57	296.5							
	05/01/2010 14:59				0.05	0.02%	19.41%		
	05/01/2010 15:02	296.4	-0.1	-0.03%					
			-0.1	**-0.03%**	**0.05**	**0.02%**		**0.05%**	**Incorrect**
15	06/01/2010 10:58	296							
	06/01/2010 11:02	296.4	0.4	0.14%					
			0.4	**0.14%**					**None**
16	06/01/2010 14:58	300.6							
	06/01/2010 15:00				0.45	0.15%	38.81%		
	06/01/2010 15:03	301.1	0.5	0.17%					
			0.5	**0.17%**	**0.45**	**0.15%**		**0.02%**	**Direction**
	06/01/2010 15:05	301.5	0.4	0.13%	-0.15	-0.05%	8.35%		
			0.9	**0.30%**	**0.3**	**0.10%**		**0.20%**	**Direction**
17	07/01/2010 10:59	310.55							
	07/01/2010 10:59				-0.04	-0.01%	38.81%		
	07/01/2010 10:59				0	0.00%	8.74%		
	07/01/2010 11:02	309.95	-0.6	-0.19%	-0.15	-0.05%	0.54%		
			-0.6	**-0.19%**	**-0.19**	**-0.06%**		**0.13%**	**Direction**
	07/01/2010 11:05	310	0.05	0.02%					
	07/01/2010 11:08				-0.08	-0.03%	5.57%		
	07/01/2010 11:09	309.75	-0.25	-0.08%					
			-0.8	**-0.26%**	**-0.27**	**-0.09%**		**0.17%**	**Direction**
18	07/01/2010 14:57	315.25							
	07/01/2010 14:59				0.8	0.25%	38.81%		
	07/01/2010 15:00	315.2	-0.05	-0.02%					
			-0.05	**-0.02%**	**0.8**	**0.25%**		**0.27%**	**Incorrect**
	07/01/2010 15:02	315.45	0.25	0.08%					
	07/01/2010 15:03				-0.64	-0.20%	19.41%		
	07/01/2010 15:07	315.85	0.4	0.13%					
			0.6	**0.19%**	**0.16**	**0.05%**		**0.14%**	**Direction**
19	08/01/2010 10:57	323.83							
	08/01/2010 11:00	324.25	0.42	0.13%					
	08/01/2010 11:03	324.1	-0.15	-0.05%					
	08/01/2010 11:04				-0.3	-0.09%	58.22%		

	08/01/2010 11:07	323.4	-0.7	-0.22%					
			-0.43	**-0.13%**	**-0.3**	**-0.09%**		**0.04%**	**Direction**
20	08/01/2010 14:57	320.25							
	08/01/2010 14:59				-0.15	-0.05%	12.05%		
	08/01/2010 15:03				-0.3	-0.09%	17.70%		
	08/01/2010 15:08	321.65	1.4	0.44%					
			1.4	**0.44%**	**-0.45**	**-0.14%**		**0.58%**	**Incorrect**
21	11/01/2010 10:58	321.2							
	11/01/2010 11:01	320.5	-0.7	-0.22%					
	11/01/2010 11:02				-0.1	-0.03%	19.41%		
	11/01/2010 11:05	319.05	-1.45	-0.45%					
			-2.15	**-0.67%**	**-0.1**	**-0.03%**		**0.64%**	**Direction**
	11/01/2010 11:07				0.57	0.18%	19.41%		
	11/01/2010 11:08	318.5	-0.55	-0.17%					
			-2.7	**-0.84%**	**0.47**	**0.15%**		**0.99%**	**Incorrect**
22	11/01/2010 14:57	319.35							
	11/01/2010 14:59				-0.75	-0.23%	19.41%		
	11/01/2010 15:02	319.4	0.05	0.02%					
			0.05	**0.02%**	**-0.75**	**-0.23%**		**0.25%**	**Incorrect**
	11/01/2010 15:03	319.3	-0.1	-0.03%	-0.3	-0.09%	19.41%		
			-0.05	**-0.02%**	**-1.05**	**-0.33%**		**0.31%**	**Direction**
23	12/01/2010 10:58	316.6							
	12/01/2010 11:02	316.6	0	0.00%	0	0.00%	13.78%		
			0	**0.00%**	**0**	**0.00%**		**0.00%**	**Correct**
24	12/01/2010 14:57	314.8							
	12/01/2010 14:59				-0.05	-0.02%	19.41%		
	12/01/2010 15:01				-0.03	-0.01%	6.76%		
	12/01/2010 15:03	315.2	0.4	0.13%					
			0.4	**0.13%**	**-0.08**	**-0.03%**		**0.15%**	**Incorrect**
	12/01/2010 15:05	315.6	0.4	0.13%					
	12/01/2010 15:07				0.6	0.19%	11.05%		
	12/01/2010 15:08	315.95	0.35	0.11%					
			1.15	**0.37%**	**0.52**	**0.17%**		**0.20%**	**Direction**
25	13/01/2010 10:59	312.45							
	13/01/2010 11:03	312.3	-0.15	-0.05%					
	13/01/2010 11:04				0.35	0.11%	77.62%		
	13/01/2010 11:06	312.15	-0.15	-0.05%					
			-0.3	**-0.10%**	**0.35**	**0.11%**		**0.21%**	**Incorrect**
	13/01/2010 11:09	312.4	0.25	0.08%					
	13/01/2010 11:11				-0.55	-0.18%	27.15%		
	13/01/2010 11:13	312.53	0.13	0.04%					
			0.08	**0.03%**	**-0.2**	**-0.06%**		**0.09%**	**Incorrect**
26	13/01/2010 14:56	312.7							
	13/01/2010 15:00				-0.64	-0.20%	38.81%		
	13/01/2010 15:01	311.7	-1	-0.32%					
			-1	**-0.32%**	**-0.64**	**-0.20%**		**0.12%**	**Direction**
27	14/01/2010 10:57	317.7							
	14/01/2010 10:58				0	0.00%	9.33%		
	14/01/2010 11:00	317.85	0.15	0.05%					

Stop	Date	Value	Change Pts	Change %	Predicted Pts	Predicted %	Prob	Variation	Direction
			0.15	**0.05%**	**0**	**0.00%**		**0.05%**	**Marginal**
28	14/01/2010 14:56	319.1							
	14/01/2010 14:56				-0.15	-0.05%	15.00%		
	14/01/2010 14:56				0	0.00%	15.35%		
	14/01/2010 15:01	319.05	-0.05	-0.02%					
			-0.05	**-0.02%**	**-0.15**	**-0.05%**		**0.03%**	**Direction**

	HSBA								
Stop	**Date**	**Value**	**Change**		**Predicted**			**Comparison**	
			Pts	**%**	**Pts**	**%**	**Prob**	**Variation**	**Direction**
1	01/12/2009 10:56	723.5							
	01/12/2009 10:57				0	0.00%	6.43%		
	01/12/2009 11:00	723.4	-0.1	-0.01%					
			-0.1	**-0.01%**	**0**	**0.00%**		**0.01%**	**Marginal**
2	01/12/2009 14:59	725.5							
	01/12/2009 14:59				1.5	0.21%	38.81%		
	01/12/2009 15:01				-0.41	-0.06%	7.38%		
	01/12/2009 15:03	726.1	0.6	0.08%					
			0.6	**0.08%**	**1.09**	**0.15%**		**0.07%**	**Direction**
3	02/12/2009 10:57	716.2							
	02/12/2009 10:59				0	0.00%	6.03%		
	02/12/2009 11:00	716.1	-0.1	-0.01%					
			-0.1	**-0.01%**	**0**	**0.00%**		**0.01%**	**Marginal**
4	02/12/2009 14:59	722.5							
	02/12/2009 14:59				0	0.00%	19.41%		
	02/12/2009 15:01				-0.4	-0.06%	6.79%		
	02/12/2009 15:04	722.7	0.2	0.03%					
			0.2	**0.03%**	**-0.4**	**-0.06%**		**0.08%**	**Incorrect**
5	03/12/2009 10:57	734.6							
	03/12/2009 10:57				1.5	0.20%	19.41%		
	03/12/2009 10:58				0	0.00%	6.71%		
	03/12/2009 11:00	734.7	0.1	0.01%	-0.4	-0.05%	4.56%		
			0.1	**0.01%**	**1.1**	**0.15%**		**0.14%**	**Direction**
	03/12/2009 11:04	734.6	-0.08	-0.01%					
	03/12/2009 11:07	733.9	-0.72	-0.10%					
	03/12/2009 11:09				0	0	5.63%		
	03/12/2009 11:10	731.9	-2	-0.27%					
			-2.7	**-0.37%**	**1.1**	**0.15%**		**0.52%**	**Incorrect**
6	03/12/2009 14:59	729.6							
	03/12/2009 15:00				0	0.00%	8.02%		
	03/12/2009 15:01	728.5	-1.1	-0.15%					
			-1.1	**-0.15%**	**0**	**0.00%**		**0.15%**	**Incorrect**
7	04/12/2009 10:59	714.1							
	04/12/2009 10:59				0	0.00%	8.98%		
	04/12/2009 11:02	714	-0.1	-0.01%					
			-0.1	**-0.01%**	**0**	**0.00%**		**0.01%**	**Marginal**
8	04/12/2009 14:58	730.3							
	04/12/2009 14:58				0.5	0.07%	19.41%		
	04/12/2009 15:00				0	0.00%	8.67%		
	04/12/2009 15:03	731.8	1.5	0.21%					

			1.5	**0.21%**	**0.5**	**0.07%**		**0.14%**	**Direction**
9	08/12/2009 10:57	705.8							
	08/12/2009 10:58				-0.4	-0.06%	5.97%		
	08/12/2009 11:00	705.5	-0.3	-0.04%					
			-0.3	**-0.04%**	**-0.4**	**-0.06%**		**0.01%**	**Direction**
10	08/12/2009 14:59	691.4							
	08/12/2009 15:00	693.3	1.9	0.27%					
	08/12/2009 15:01				-0.4	-0.06%	18.05%		
	08/12/2009 15:04	692.7	-0.6	-0.09%					
			1.3	**0.19%**	**-0.4**	**-0.06%**		**0.25%**	**Incorrect**
11	04/01/2010 10:57								
	04/01/2010 10:57				-0.41		11.21%		
	04/01/2010 10:58				0		5.24%		
	04/01/2010 11:00				-0.3		0.00%		
					-0.71				**Error**
12	04/01/2010 14:59								
	04/01/2010 14:59				-0.41		8.96%		
	04/01/2010 15:00				0		4.63%		
	04/01/2010 15:02				0.2		0.00%		
	04/01/2010 15:10				0		0.12%		
					-0.21				**Error**
13	05/01/2010 10:58								
	05/01/2010 10:58				-0.5		19.41%		
	05/01/2010 10:59				-0.3		19.41%		
	05/01/2010 11:03				0		40.90%		
	05/01/2010 11:09				0.2		0.00%		
					-0.6				**Error**
14	05/01/2010 14:57								
	05/01/2010 14:59				-0.2		56.57%		
					-0.2				**Error**
15	06/01/2010 10:58	736.3							
	06/01/2010 11:05	737.2	0.9	0.12%					
			0.9	**0.12%**					**None**
16	06/01/2010 14:58	737.7							
	06/01/2010 15:03	737.7	0	0.00%					
	06/01/2010 15:05	738	0.3	0.04%					
			0.3	**0.04%**					**None**
17	07/01/2010 10:59	733.7							
	07/01/2010 10:59				0	0.00%	5.20%		
	07/01/2010 11:01				-0.4	-0.05%	2.24%		
	07/01/2010 11:02	732.8	-0.9	-0.12%					
			-0.9	**-0.12%**	**-0.4**	**-0.05%**		**0.07%**	**Direction**
	07/01/2010 11:05	733.1	0.3	0.04%					
	07/01/2010 11:07				0	0.00%	2.11%		
	07/01/2010 11:09	732	-1.1	-0.15%					
			-1.7	**-0.23%**	**-0.4**	**-0.05%**		**0.18%**	**Direction**
18	07/01/2010 14:57	736							
	07/01/2010 14:59				0.2	0.03%	19.41%		
	07/01/2010 15:00	735.7	-0.3	-0.04%					
			-0.3	**-0.04%**	**0.2**	**0.03%**		**0.07%**	**Incorrect**
19	08/01/2010 10:57	738.1							

	08/01/2010 10:57				-0.5	-0.07%	38.81%		
	08/01/2010 10:59				0.2	0.03%	19.41%		
	08/01/2010 11:00	738.9	0.8	0.11%					
			0.8	**0.11%**	**-0.3**	**-0.04%**		**0.15%**	**Incorrect**
	08/01/2010 11:01				0		9.57%		
	08/01/2010 11:03	737.5	-1.4	-0.19%					
			-0.6	**-0.08%**	**-0.3**	**-0.04%**		**0.04%**	**Direction**
20	08/01/2010 14:57	736.3							
	08/01/2010 14:57				0	0.00%	7.36%		
	08/01/2010 15:08	738.1	1.8	0.24%					
			1.8	**0.24%**	**0**	**0.00%**		**0.24%**	**Incorrect**
21	11/01/2010 10:58	736.2							
	11/01/2010 11:00				0.2	0.03%	38.81%		
	11/01/2010 11:01	735.8	-0.4	-0.05%					
			-0.4	**-0.05%**	**0.2**	**0.03%**		**0.08%**	**Incorrect**
	11/01/2010 11:05	735.3	-0.5	-0.07%					
	11/01/2010 11:08	735.5	0.2	0.03%					
	11/01/2010 11:09				0	0.00%	21.68%		
	11/01/2010 11:11	735	-0.5	-0.07%					
			-1.2	**-0.16%**	**0.2**	**0.03%**		**0.19%**	**Incorrect**
22	11/01/2010 14:57	733.2							
	11/01/2010 14:57				0	0.00%	19.41%		
	11/01/2010 15:02	734	0.8	0.11%					
			0.8	**0.11%**	**0**	**0.00%**		**0.11%**	**Incorrect**
23	12/01/2010 10:58	731							
	12/01/2010 11:02	730.7	-0.3	-0.04%					
	12/01/2010 11:05	730	-0.7	-0.10%					
	12/01/2010 11:08	729.8	-0.2	-0.03%	0	0.00%	12.44%		
			-1.2	**-0.16%**	**0**	**0.00%**		**0.16%**	**Incorrect**
24	12/01/2010 14:58	730.1							
	12/01/2010 15:03	729.7	-0.4	-0.05%					
			-0.4	**-0.05%**					**None**
25	13/01/2010 10:59	718.7							
	13/01/2010 10:59				0	0.00%	8.67%		
	13/01/2010 11:01				-0.6	-0.08%	6.71%		
	13/01/2010 11:03	718.1	-0.6	-0.08%					
			-0.6	**-0.08%**	**-0.6**	**-0.08%**		**0.00%**	**Correct**
26	13/01/2010 14:56	719.3							
	13/01/2010 14:58				0	0.00%	8.50%		
	13/01/2010 15:01	718.1	-1.2	-0.17%					
			-1.2	**-0.17%**	**0**	**0.00%**		**0.17%**	**Incorrect**
27	14/01/2010 10:57	717.5							
	14/01/2010 10:57				0	0.00%	8.22%		
	14/01/2010 11:00	718.4	0.9	0.13%					
			0.9	**0.13%**	**0**	**0.00%**		**0.13%**	**Incorrect**
28	14/01/2010 14:56	714.5							
	14/01/2010 14:56				0	0.00%	8.86%		
	14/01/2010 15:00				0.2	0.03%	0.53%		
	14/01/2010 15:01	715	0.5	0.07%					
			0.5	**0.07%**	**0.2**	**0.03%**		**0.04%**	**Direction**
	14/01/2010 15:04	714.7	-0.3	-0.04%					

	14/01/2010 15:06	714.5	-0.2	-0.03%	0	0.00%	0.75%		
			0	**0.00%**	**0.2**	**0.03%**		**0.03%**	**Marginal**

	LLOY								
Stop	**Date**	**Value**	**Change**		**Predicted**			**Comparison**	
			Pts	**%**	**Pts**	**%**	**Prob**	**Variation**	**Direction**
1	01/12/2009 10:56	54.62							
	01/12/2009 10:57				0	0.00%	8.20%		
	01/12/2009 11:00	54.64	0.02	0.04%					
			0.02	**0.04%**	**0**	**0.00%**		**0.04%**	**Marginal**
2	01/12/2009 14:59	54.22							
	01/12/2009 15:01				0	0.00%	6.25%		
	01/12/2009 15:04	54.24	0.02	0.04%					
			0.02	**0.04%**	**0**	**0.00%**		**0.04%**	**Marginal**
3	02/12/2009 10:57	51.5							
	02/12/2009 10:59				0	0.00%	7.38%		
	02/12/2009 11:00	51.23	-0.27	-0.52%					
			-0.27	**-0.52%**	**0**	**0.00%**		**0.52%**	**Incorrect**
4	02/12/2009 14:59	53.12							
	02/12/2009 15:04	53.28	0.16	0.30%					
			0.16	**0.30%**					**None**
5	03/12/2009 10:57	55.34							
	03/12/2009 10:58				0	0.00%	5.97%		
	03/12/2009 11:00	55.45	0.11	0.20%					
			0.11	**0.20%**	**0**	**0.00%**		**0.20%**	**Incorrect**
6	03/12/2009 14:59	55.91							
	03/12/2009 14:59				-0.07	-0.13%	19.41%		
	03/12/2009 15:00				0	0.00%	14.86%		
	03/12/2009 15:01	55.78	-0.13	-0.23%					
			-0.13	**-0.23%**	**-0.07**	**-0.13%**		**0.11%**	**Direction**
7	04/12/2009 10:59	54.05							
	04/12/2009 11:02	53.98	-0.07	-0.13%					
	04/12/2009 11:03				0.26	0.48%	19.41%		
	04/12/2009 11:06	53.8	-0.18	-0.33%					
			-0.25	**-0.46%**	**0.26**	**0.48%**		**0.94%**	**Incorrect**
8	04/12/2009 14:58	56.43							
	04/12/2009 15:02				0.21	0.37%	19.41%		
	04/12/2009 15:03	56.51	0.08	0.14%					
			0.08	**0.14%**	**0.21**	**0.37%**		**0.23%**	**Direction**
9	08/12/2009 10:57	54.18							
	08/12/2009 11:00	54.29	0.11	0.20%					
			0.11	**0.20%**					**None**
10	08/12/2009 14:59	53.25							
	08/12/2009 15:00	52.98	-0.27	-0.51%					
	08/12/2009 15:02				0	0.00%	6.02%		
	08/12/2009 15:04	53.3	0.32	0.60%					
			0.05	**0.09%**	**0**	**0.00%**		**0.09%**	**Marginal**
11	04/01/2010 10:57	51.93							
	04/01/2010 10:59				0	0.00%	6.60%		
	04/01/2010 11:01	51.86	-0.07	-0.13%					

			-0.07	**-0.13%**	**0**	**0.00%**		**0.13%**	**Incorrect**
	04/01/2010 11:03				0.26	0.50%	13.88%		
	04/01/2010 11:04	51.93	0.07	0.13%					
			0	**0.00%**	**0.26**	**0.50%**		**0.50%**	**Incorrect**
12	04/01/2010 14:59	51.95							
	04/01/2010 14:59				0	0.00%	5.00%		
	04/01/2010 15:01	51.94	-0.01	-0.02%					
			-0.01	**-0.02%**	**0**	**0.00%**		**0.02%**	**Marginal**
13	05/01/2010 10:58	54.07							
	05/01/2010 10:59				0.08	0.15%	19.41%		
	05/01/2010 11:01	54.23	0.16	0.30%					
			0.16	**0.30%**	**0.08**	**0.15%**		**0.15%**	**Direction**
14	05/01/2010 14:57	53.89							
	05/01/2010 15:02	53.9	0.01	0.02%					
			0.01	**0.02%**					**None**
15	06/01/2010 10:58	53.5							
	06/01/2010 10:58				-0.07	-0.13%	19.41%		
	06/01/2010 11:02	53.35	-0.15	-0.28%					
			-0.15	**-0.28%**	**-0.07**	**-0.13%**		**0.15%**	**Direction**
16	06/01/2010 14:58	54.31							
	06/01/2010 15:03	54.27	-0.04	-0.07%					
	06/01/2010 15:05	54.34	0.07	0.13%					
	06/01/2010 15:07	54.39	0.05	0.09%					
	06/01/2010 15:10				0	0.00%	2.93%		
	06/01/2010 15:13	54.34	-0.05	-0.09%					
			0.03	**0.06%**	**0**	**0.00%**		**0.06%**	**Marginal**
17	07/01/2010 10:59	55.25							
	07/01/2010 11:02	55.15	-0.1	-0.18%					
	07/01/2010 11:03				0	0.00%	8.67%		
	07/01/2010 11:05	55	-0.15	-0.27%					
			-0.25	**-0.45%**	**0**	**0.00%**		**0.45%**	**Incorrect**
18	07/01/2010 14:57	56.6							
	07/01/2010 15:00	56.64	0.04	0.07%					
			0.04	**0.07%**					**None**
19	08/01/2010 10:57	56.92							
	08/01/2010 10:59				0	0.00%	9.57%		
	08/01/2010 11:00	56.97	0.05	0.09%					
			0.05	**0.09%**	**0**	**0.00%**		**0.09%**	**Marginal**
20	08/01/2010 14:57	56.77							
	08/01/2010 15:08	56.95	0.18	0.32%					
			0.18	**0.32%**					**None**
21	11/01/2010 10:58	56.98							
	11/01/2010 11:01	56.86	-0.12	-0.21%					
			-0.12	**-0.21%**					**None**
22	11/01/2010 14:57	57.02							
	11/01/2010 15:01				-0.05	-0.09%	19.41%		
	11/01/2010 15:02	57	-0.02	-0.04%					
			-0.02	**-0.04%**	**-0.05**	**-0.09%**		**0.05%**	**Direction**
23	12/01/2010 10:58	55.95							
	12/01/2010 10:58				0	0.00%	5.73%		
	12/01/2010 11:02	55.9	-0.05	-0.09%					

			-0.05	**-0.09%**	**0**	**0.00%**		**0.09%**	**Marginal**
24	12/01/2010 14:57	55.31							
	12/01/2010 15:03	55.62	0.31	0.56%					
			0.31	**0.56%**					**None**
25	13/01/2010 10:59	55.47							
	13/01/2010 10:59				0	0.00%	8.79%		
	13/01/2010 11:03	55.44	-0.03	-0.05%					
			-0.03	**-0.05%**	**0**	**0.00%**		**0.05%**	**Marginal**
26	13/01/2010 14:56	55.66							
	13/01/2010 15:01	55.68	0.02	0.04%					
			0.02	**0.04%**					**None**
27	14/01/2010 10:57	57.98							
	14/01/2010 11:00	57.89	-0.09	-0.16%					
	14/01/2010 11:03	57.77	-0.12	-0.21%					
	14/01/2010 11:07	57.7	-0.07	-0.12%					
	14/01/2010 11:10	57.71	0.01	0.02%					
	14/01/2010 11:13	57.73	0.02	0.03%					
	14/01/2010 11:14				0	0.00%	69.30%		
	14/01/2010 11:17	57.85	0.12	0.21%					
			-0.13	**-0.22%**	**0**	**0.00%**		**0.22%**	**Incorrect**
28	14/01/2010 14:56	57.3							
	14/01/2010 14:56				0.19	0.33%	19.41%		
	14/01/2010 14:56				0	0.00%	9.07%		
	14/01/2010 15:01	57.3	0	0.00%					
			0	**0.00%**	**0.19**	**0.33%**		**0.33%**	**Incorrect**

	RBS								
Stop	**Date**	**Value**	**Change**		**Predicted**			**Comparison**	
			Pts	**%**	**Pts**	**%**	**Prob**	**Variation**	**Direction**
1	01/12/2009 10:56	34.07							
	01/12/2009 11:00	34.15	0.08	0.23%	0.12	0.35%	19.41%		
			0.08	**0.23%**	**0.12**	**0.35%**		**0.12%**	**Direction**
2	01/12/2009 14:59	34.09							
	01/12/2009 15:03	34.14	0.05	0.15%					
	01/12/2009 15:04				0.07	0.21%	19.41%		
	01/12/2009 15:06	33.97	-0.17	-0.50%					
			-0.12	**-0.35%**	**0.07**	**0.21%**		**0.56%**	**Incorrect**
3	02/12/2009 10:57	32.2							
	02/12/2009 10:57				0.1	0.31%	38.81%		
	02/12/2009 10:59				0	0.00%	10.24%		
	02/12/2009 11:00	32.07	-0.13	-0.40%					
			-0.13	**-0.40%**	**0.1**	**0.31%**		**0.71%**	**Incorrect**
	02/12/2009 11:01				0.01	0.03%	0.24%		
	02/12/2009 11:03	32.15	0.08	0.25%					
			-0.05	**-0.16%**	**0.11**	**0.34%**		**0.50%**	**Incorrect**
4	02/12/2009 14:59	32.58							
	02/12/2009 14:59				-0.03	-0.09%	19.41%		
	02/12/2009 15:04	32.62	0.04	0.12%					
			0.04	**0.12%**	**-0.03**	**-0.09%**		**0.21%**	**Incorrect**
5	03/12/2009 10:57	34.71							

	03/12/2009 10:58				0	0.00%	11.56%		
	03/12/2009 11:00	34.81	0.1	0.29%					
			0.1	**0.29%**	**0**	**0.00%**		**0.29%**	**Incorrect**
	03/12/2009 11:03				0.03	0.09%	1.34%		
	03/12/2009 11:04	34.74	-0.07	-0.20%					
			0.03	**0.09%**	**0.03**	**0.09%**		**0.00%**	**Correct**
6	03/12/2009 14:59	34.92							
	03/12/2009 15:00				-0.05	-0.14%	19.41%		
	03/12/2009 15:01	34.9	-0.02	-0.06%					
			-0.02	**-0.06%**	**-0.05**	**-0.14%**		**0.09%**	**Direction**
	03/12/2009 15:02	35.07	0.17	0.49%	0	0.00%	14.86%		
	03/12/2009 15:06				0.01	0.03%	4.70%		
	03/12/2009 15:08	34.85	-0.22	-0.63%					
			-0.07	**-0.20%**	**-0.04**	**-0.11%**		**0.09%**	**Direction**
7	04/12/2009 10:59	34.4							
	04/12/2009 10:59				-0.06	-0.17%	19.41%		
	04/12/2009 11:00				0	0.00%	14.11%		
	04/12/2009 11:02	34.35	-0.05	-0.15%					
			-0.05	**-0.15%**	**-0.06**	**-0.17%**		**0.03%**	**Direction**
	04/12/2009 11:05				0.03	0.09%	5.60%		
	04/12/2009 11:06	34.31	-0.04	-0.12%					
			-0.09	**-0.26%**	**-0.03**	**-0.09%**		**0.17%**	**Direction**
8	04/12/2009 14:58	34.99							
	04/12/2009 15:00				0.23	0.66%	19.41%		
	04/12/2009 15:03	34.95	-0.04	-0.11%					
			-0.04	**-0.11%**	**0.23**	**0.66%**		**0.77%**	**Incorrect**
	04/12/2009 15:04	34.89	-0.06	-0.17%	0.12	0.34%	19.41%		
			-0.1	**-0.29%**	**0.35**	**1.00%**		**1.29%**	**Incorrect**
9	08/12/2009 10:57	31.77							
	08/12/2009 11:00	31.86	0.09	0.28%					
			0.09	**0.28%**					**None**
10	08/12/2009 14:59	30.34							
	08/12/2009 15:00	30.42	0.08	0.26%					
	08/12/2009 15:02				0	0.00%	6.02%		
	08/12/2009 15:04	30.77	0.35	1.15%					
			0.43	**1.42%**	**0**	**0.00%**		**1.42%**	**Incorrect**
11	04/01/2010 10:57	30.58							
	04/01/2010 10:57				-0.06	-0.20%	19.41%		
	04/01/2010 11:07	30.62	0.04	0.13%					
			0.04	**0.13%**	**-0.06**	**-0.20%**		**0.33%**	**Incorrect**
12	04/01/2010 14:59	31.65							
	04/01/2010 15:00	31.65	0	0.00%					
	04/01/2010 15:03	31.72	0.07	0.22%	0.03	0.09%	22.76%		
			0.07	**0.22%**	**0.03**	**0.09%**		**0.13%**	**Direction**
	04/01/2010 15:08	31.65	-0.07	-0.22%	0	0.00%	3.63%		
			0	**0.00%**	**0.03**	**0.09%**		**0.09%**	**Marginal**
13	05/01/2010 10:58	34.53							
	05/01/2010 11:00				0.23	0.67%	19.41%		
	05/01/2010 11:01	34.56	0.03	0.09%					
			0.03	**0.09%**	**0.23**	**0.67%**		**0.58%**	**Direction**
14	05/01/2010 14:57	35.1							

	05/01/2010 14:59				0.01	0.03%	19.41%		
	05/01/2010 15:02	35.09	-0.01	-0.03%					
			-0.01	**-0.03%**	**0.01**	**0.03%**		**0.06%**	**Incorrect**
15	06/01/2010 10:58	36.02							
	06/01/2010 10:59				0	0.00%	19.41%		
	06/01/2010 11:02	36.07	0.05	0.14%					
			0.05	**0.14%**	**0**	**0.00%**		**0.14%**	**Incorrect**
16	06/01/2010 14:58	36.36							
	06/01/2010 14:58				0.05	0.14%	24.26%		
	06/01/2010 15:02	36.44	0.08	0.22%					
			0.08	**0.22%**	**0.05**	**0.14%**		**0.08%**	**Direction**
17	07/01/2010 10:59	35.97							
	07/01/2010 10:59				-0.03	-0.08%	19.41%		
	07/01/2010 10:59				0	0.00%	5.24%		
	07/01/2010 11:01				0.01	0.03%	0.71%		
	07/01/2010 11:02	35.89	-0.08	-0.22%					
			-0.08	**-0.22%**	**-0.02**	**-0.06%**		**0.17%**	**Direction**
	07/01/2010 11:05	35.87	-0.02	-0.06%	0.03	0.08%	0.71%		
			-0.1	**-0.28%**	**0.01**	**0.03%**		**0.31%**	**Incorrect**
18	07/01/2010 14:57	36.06							
	07/01/2010 15:00	36	-0.06	-0.17%					
			-0.06	**-0.17%**					**None**
19	08/01/2010 10:57	36.1							
	08/01/2010 10:59				0	0.00%	9.57%		
	08/01/2010 11:00	36.08	-0.02	-0.06%					
			-0.02	**-0.06%**	**0**	**0.00%**		**0.06%**	**Marginal**
20	08/01/2010 14:57	34.91							
	08/01/2010 15:08	35.08	0.17	0.49%					
			0.17	**0.49%**					**None**
21	11/01/2010 10:58	34.36							
	11/01/2010 11:00				0.01	0.03%	19.41%		
	11/01/2010 11:01	34.14	-0.22	-0.64%					
			-0.22	**-0.64%**	**0.01**	**0.03%**		**0.67%**	**Incorrect**
	11/01/2010 11:05	34.05	-0.09	-0.26%					
	11/01/2010 11:08	34.21	0.16	0.47%					
	11/01/2010 11:09				0	0.00%	21.68%		
	11/01/2010 11:11	34.28	0.07	0.20%					
			-0.08	**-0.23%**	**0.01**	**0.03%**		**0.26%**	**Incorrect**
22	11/01/2010 14:57	34.59							
	11/01/2010 15:02	34.7	0.11	0.32%					
			0.11	**0.32%**					**None**
23	12/01/2010 10:58	34.83							
	12/01/2010 10:58				0	0.00%	5.20%		
	12/01/2010 11:00				0.01	0.03%	0.35%		
	12/01/2010 11:02	34.76	-0.07	-0.20%					
			-0.07	**-0.20%**	**0.01**	**0.03%**		**0.23%**	**Incorrect**
	12/01/2010 11:04				0.03	0.09%	0.35%		
	12/01/2010 11:05	34.75	-0.01	-0.03%					
			-0.08	**-0.23%**	**0.04**	**0.11%**		**0.34%**	**Incorrect**
24	12/01/2010 14:57	34.82							
	12/01/2010 14:58				0.09	0.26%	19.41%		

	12/01/2010 15:02	34.88	0.06	0.17%					
			0.06	**0.17%**	**0.09**	**0.26%**		**0.09%**	**Direction**
	12/01/2010 15:05	34.87	-0.01	-0.03%					
	12/01/2010 15:07	34.85	-0.02	-0.06%	0	0	10.59%		
			0.03	**0.09%**	**0.09**	**0.26%**		**0.17%**	**Direction**
25	13/01/2010 10:59	34.73							
	13/01/2010 10:59				0	0.00%	8.42%		
	13/01/2010 10:59				0.02	0.06%	4.96%		
	13/01/2010 11:03	34.72	-0.01	-0.03%					
			-0.01	**-0.03%**	**0.02**	**0.06%**		**0.09%**	**Incorrect**
	13/01/2010 11:06	34.68	-0.04	-0.12%					
	13/01/2010 11:09	34.7	0.02	0.06%	0	0.00%	4.26%		
			-0.03	**-0.09%**	**0.02**	**0.06%**		**0.14%**	**Incorrect**
26	13/01/2010 14:56	35.38							
	13/01/2010 14:58				0.01	0.03%	38.81%		
	13/01/2010 15:01	35.4	0.02	0.06%					
			0.02	**0.06%**	**0.01**	**0.03%**		**0.03%**	**Direction**
	13/01/2010 15:02				0.08	0.23%	16.97%		
	13/01/2010 15:03	35.4	0	0.00%					
			0.02	**0.06%**	**0.09**	**0.25%**		**0.20%**	**Direction**
27	14/01/2010 10:57	35.84							
	14/01/2010 11:00	35.87	0.03	0.08%					
			0.03	**0.08%**					**None**
28	14/01/2010 14:56	35.95							
	14/01/2010 14:56				0	0.00%	9.18%		
	14/01/2010 15:00				0.03	0.08%	3.32%		
	14/01/2010 15:01	35.95	0	0.00%					
			0	**0.00%**	**0.03**	**0.08%**		**0.08%**	**Marginal**

	STAN								
Stop	**Date**	**Value**	**Change**		**Predicted**			**Comparison**	
			Pts	**%**	**Pts**	**%**	**Prob**	**Variation**	**Direction**
1	01/12/2009 10:56	1529.5							
	01/12/2009 10:58				1.5	0.10%	38.81%		
	01/12/2009 11:00	1530.5	1	0.07%					
			1	**0.07%**	**1.5**	**0.10%**		**0.03%**	**Direction**
2	01/12/2009 14:59	1556.5							
	01/12/2009 15:00				0.5	0.03%	38.81%		
	01/12/2009 15:03				1.5	0.10%	10.59%		
	01/12/2009 15:04	1556	-0.5	-0.03%					
			-0.5	**-0.03%**	**2**	**0.13%**		**0.16%**	**Incorrect**
3	02/12/2009 10:57	1542							
	02/12/2009 10:59				4.5	0.29%	38.81%		
	02/12/2009 11:00	1542	0	0.00%					
			0	**0.00%**	**4.5**	**0.29%**		**0.29%**	**Incorrect**
	02/12/2009 11:03				2.5	0.16%	10.59%		
	02/12/2009 11:03	1542	0	0.00%					
			0	**0.00%**	**7**	**0.45%**		**0.45%**	**Incorrect**
4	02/12/2009 14:59	1550							
	02/12/2009 15:01				-3	-0.19%	19.41%		

	02/12/2009 15:04	1553	3	0.19%					
			3	**0.19%**	**-3**	**-0.19%**		**0.39%**	**Incorrect**
5	03/12/2009 10:57	1571							
	03/12/2009 10:58				0.5	0.03%	38.81%		
	03/12/2009 11:00	1572.5	1.5	0.10%					
			1.5	**0.10%**	**0.5**	**0.03%**		**0.06%**	**Direction**
	03/12/2009 11:02				-2	-0.13%	10.59%		
	03/12/2009 11:04	1573	0.5	0.03%					
			2	**0.13%**	**-1.5**	**-0.10%**		**0.22%**	**Incorrect**
6	03/12/2009 14:59	1573							
	03/12/2009 15:01	1570	-3	-0.19%	0	0.00%	58.22%		
			-3	**-0.19%**	**0**	**0.00%**		**0.19%**	**Incorrect**
	03/12/2009 15:03	1570	0						
	03/12/2009 15:04				5.5	0.35%	13.42%		
	03/12/2009 15:08	1571	1	0.06%					
			-2	**-0.13%**	**5.5**	**0.35%**		**0.48%**	**Incorrect**
7	04/12/2009 10:59	1507							
	04/12/2009 10:59				-3	-0.20%	38.81%		
	04/12/2009 11:00				1.5	0.10%	10.59%		
	04/12/2009 11:02	1506	-1	-0.07%					
			-1	**-0.07%**	**-1.5**	**-0.10%**		**0.03%**	**Direction**
	04/12/2009 11:04				2.5	0.17%	6.45%		
	04/12/2009 11:06	1508	2	0.13%					
			1	**0.07%**	**1**	**0.07%**		**0.00%**	**Correct**
8	04/12/2009 14:58	1536							
	04/12/2009 14:59				0	0.00%	19.41%		
	04/12/2009 15:00				1	0.07%	5.30%		
	04/12/2009 15:02	1537	1	0.07%					
			1	**0.07%**	**1**	**0.07%**		**0.00%**	**Correct**
9	08/12/2009 10:57	1506							
	08/12/2009 11:00	1505	-1	-0.07%					
	08/12/2009 11:01				5	0.33%	19.41%		
	08/12/2009 11:04	1504.5	-0.5	-0.03%					
			-1.5	**-0.10%**	**5**	**0.33%**		**0.43%**	**Incorrect**
10	08/12/2009 14:59	1443.5							
	08/12/2009 15:00	1449	5.5	0.38%					
	08/12/2009 15:01				-3	-0.21%	19.41%		
	08/12/2009 15:04	1447.5	-1.5	-0.10%					
			4	**0.28%**	**-3**	**-0.21%**		**0.48%**	**Incorrect**
11	04/01/2010 10:57	1593							
	04/01/2010 10:59				1	0.06%	38.81%		
	04/01/2010 11:01	1592.5	-0.5	-0.03%	-0.5	-0.03%	11.54%		
			-0.5	**-0.03%**	**0.5**	**0.03%**		**0.06%**	**Incorrect**
	04/01/2010 11:04	1592.5	0	0.00%					
	04/01/2010 11:05				0	0.00%	5.06%		
	04/01/2010 11:07	1592.5	0	0.00%					
			-1	**-0.06%**	**1**	**0.06%**		**0.13%**	**Incorrect**
12	04/01/2010 14:59	1604.5							
	04/01/2010 15:01	1604	-0.5	-0.03%	5.5	0.34%	5.92%		
			-0.5	**-0.03%**	**5.5**	**0.34%**		**0.37%**	**Incorrect**
	04/01/2010 15:03	1605	1		0		5.45%		

			0.5	**0.03%**	**5.5**	**0.34%**		**0.31%**	**Direction**
13	05/01/2010 10:58	1630							
	05/01/2010 11:01	1628	-2	-0.12%					
			-2	**-0.12%**					**None**
14	05/01/2010 14:57	1631							
	05/01/2010 14:58				5.5	0.34%	58.22%		
	05/01/2010 15:02	1627.5	-3.5	-0.21%	0	0.00%	29.53%		
			-3.5	**-0.21%**	**5.5**	**0.34%**		**0.55%**	**Incorrect**
	05/01/2010 15:05	1626.5	-1	-0.06%					
	05/01/2010 15:07	1627.5	1	0.06%	5.5	0.34%	8.57%		
			-3.5	**-0.21%**	**11**	**0.67%**		**0.89%**	**Incorrect**
15	06/01/2010 10:58	1615.35							
	06/01/2010 11:02	1613	-2.35	-0.15%					
			-2.35	**-0.15%**					**None**
16	06/01/2010 14:58	1615.5							
	06/01/2010 15:00				-3	-0.19%	19.41%		
	06/01/2010 15:03	1616	0.5	0.03%					
			0.5	**0.03%**	**-3**	**-0.19%**		**0.22%**	**Incorrect**
17	07/01/2010 10:59	1589							
	07/01/2010 11:00				5.5	0.35%	38.81%		
	07/01/2010 11:02	1589	0	0.00%					
			0	**0.00%**	**5.5**	**0.35%**		**0.35%**	**Incorrect**
	07/01/2010 11:04				0		5.97%		
	07/01/2010 11:05	1586	-3						
			-3	**-0.19%**	**5.5**	**0.35%**		**0.53%**	**Incorrect**
18	07/01/2010 14:57	1597.5							
	07/01/2010 14:58				-2	-0.13%	38.81%		
	07/01/2010 15:00	1598	0.5	0.03%					
			0.5	**0.03%**	**-2**	**-0.13%**		**0.16%**	**Incorrect**
	07/01/2010 15:01		1	0.06%	0	0.00%	10.59%		
			2	**0.13%**	**-2**	**-0.13%**		**0.25%**	**Incorrect**
19	08/01/2010 10:57	1610.5							
	08/01/2010 10:59				0.5	0.03%	19.41%		
	08/01/2010 11:00	1615	4.5	0.28%					
			4.5	**0.28%**	**0.5**	**0.03%**		**0.25%**	**Direction**
20	08/01/2010 14:57	1614.5							
	08/01/2010 14:58				0	0.00%	38.81%		
	08/01/2010 15:00				-2	-0.12%	6.71%		
	08/01/2010 15:08	1617	2.5	0.15%					
			2.5	**0.15%**	**-2**	**-0.12%**		**0.28%**	**Incorrect**
21	11/01/2010 10:58	1613.5							
	11/01/2010 11:00				-10	-0.62%	38.81%		
	11/01/2010 11:01	1613.5	0	0.00%					
			0	**0.00%**	**-10**	**-0.62%**		**0.62%**	**Incorrect**
22	11/01/2010 14:57	1601.5							
	11/01/2010 14:57				7	0.44%	19.41%		
	11/01/2010 15:02	1602.5	1	0.06%					
			1	**0.06%**	**7**	**0.44%**		**0.37%**	**Direction**
23	12/01/2010 10:58	1576							
	12/01/2010 10:58				1	0.06%	58.22%		
	12/01/2010 10:59				0	0.00%	24.83%		

	12/01/2010 11:01				-0.5	-0.03%	11.23%		
	12/01/2010 11:02	1576.11	0.11	0.01%					
			0.11	**0.01%**	**-0.5**	**-0.03%**		**0.04%**	**Incorrect**
24	12/01/2010 14:57	1552.5							
	12/01/2010 14:59				0.5	0.03%	19.41%		
	12/01/2010 15:03	1554	1.5	0.10%					
			1.5	**0.10%**	**0.5**	**0.03%**		**0.06%**	**Direction**
25	13/01/2010 10:59	1549.5							
	13/01/2010 11:01				1.5	0.10%	58.22%		
	13/01/2010 11:03	1548.5	-1	-0.06%					
			-1	**-0.06%**	**1.5**	**0.10%**		**0.16%**	**Incorrect**
	13/01/2010 11:04				0	0	19.57%		
	13/01/2010 11:06	1549	0.5	0.03%					
			-0.5	**-0.03%**	**1.5**	**0.10%**		**0.13%**	**Incorrect**
	13/01/2010 11:09	1550	1	0.06%	5.5	0.35%	8.16%		
			0.5	**0.03%**	**7**	**0.45%**		**0.42%**	**Direction**
26	13/01/2010 14:56	1557							
	13/01/2010 14:58				0.62	0.04%	19.41%		
	13/01/2010 15:01	1555	-2	-0.13%					
			-2	**-0.13%**	**0.62**	**0.04%**		**0.17%**	**Incorrect**
	13/01/2010 15:03	1555.5	0.5	0.03%	0	0.00%	10.78%		
			-1.5	**-0.10%**	**0.62**	**0.04%**		**0.14%**	**Incorrect**
27	14/01/2010 10:57	1560							
	14/01/2010 11:00	1559.5	-0.5	-0.03%					
			-0.5	**-0.03%**					**None**
28	14/01/2010 14:56	1565.5							
	14/01/2010 14:57				-0.3	-0.02%	38.81%		
	14/01/2010 15:01	1567	1.5	0.10%					
			1.5	**0.10%**	**-0.3**	**-0.02%**		**0.11%**	**Incorrect**

2. Share and News results

	BARC								
Stop	**Date**	**Value**	**Change**		**Predicted**			**Comparison**	
			Pts	**%**	**Pts**	**%**	**Prob**	**Variation**	**Direction**
1	01/12/2009 10:56	296.6							
	01/12/2009 11:00	295.75	-0.85	-0.29%					
	01/12/2009 11:01				0.57	0.19%	14.93%		
	01/12/2009 11:03	295.55	-0.2	-0.07%					
			-1.05	**-0.35%**	**0.57**	**0.19%**		**0.55%**	**Incorrect**
2	01/12/2009 14:59	299.04							
	01/12/2009 14:59				0	0.00%	11.75%		
	01/12/2009 15:01				-0.08	-0.03%	8.24%		
	01/12/2009 15:03	299.35	0.31	0.10%					
			0.31	**0.10%**	**-0.08**	**-0.03%**		**0.13%**	**Incorrect**
3	02/12/2009 10:57	288.9							
	02/12/2009 10:59				0.6	0.21%	7.46%		
	02/12/2009 11:00	289.15	0.25	0.09%					
			0.25	**0.09%**	**0.6**	**0.21%**		**0.12%**	**Direction**
4	02/12/2009 14:59	292.15							
	02/12/2009 14:59				-0.04	-0.01%	14.93%		
	02/12/2009 15:04	293.8	1.65	0.56%					
			1.65	**0.56%**	**-0.04**	**-0.01%**		**0.58%**	**Incorrect**
5	03/12/2009 10:57	311.6							
	03/12/2009 10:59				0.45	0.14%	7.46%		
	03/12/2009 11:00	311.5	-0.1	-0.03%					
			-0.1	**-0.03%**	**0.45**	**0.14%**		**0.18%**	**Incorrect**
6	03/12/2009 14:59	308.85							
	03/12/2009 14:59				0	0.00%	8.26%		
	03/12/2009 15:01	308.55	-0.3	-0.10%	0.6	0.19%	7.46%		
			-0.3	**-0.10%**	**0.6**	**0.19%**		**0.29%**	**Incorrect**
7	04/12/2009 10:59	299							
	04/12/2009 11:01				-0.75	-0.25%	8.82%		
	04/12/2009 11:02	299.15	0.15	0.05%					
			0.15	**0.05%**	**-0.75**	**-0.25%**		**0.30%**	**Incorrect**
	04/12/2009 11:04				-0.15	-0.05%	7.46%		
	04/12/2009 11:06	299.45	0.45	0.15%					
			0.6	**0.20%**	**-0.9**	**-0.30%**		**0.50%**	**Incorrect**
8	04/12/2009 14:58	309.15							
	04/12/2009 14:58				1.25	0.40%	14.93%		
	04/12/2009 15:00				-0.08	-0.03%	7.46%		
	04/12/2009 15:03	309.95	0.8	0.26%					
			0.8	**0.26%**	**1.17**	**0.38%**		**0.12%**	**Direction**
9	08/12/2009 10:57	293.75							
	08/12/2009 11:00	293.7	-0.05	-0.02%					
			-0.05	**-0.02%**					**None**
10	08/12/2009 14:59	288.6							
	08/12/2009 15:00				-0.35	-0.12%	14.93%		
	08/12/2009 15:05	288.3	-0.3	-0.10%					
			-0.3	**-0.10%**	**-0.35**	**-0.12%**		**0.02%**	**Direction**

11	04/01/2010 10:57	278.84							
	04/01/2010 10:57				0	0.00%	4.55%		
	04/01/2010 10:59				0.6	0.22%	7.46%		
	04/01/2010 11:01	279.1	0.26	0.09%					
			0.26	**0.09%**	**0.6**	**0.22%**		**0.12%**	**Direction**
12	04/01/2010 14:59	279.3							
	04/01/2010 15:01	279.35	0.05	0.02%					
			0.57	**0.20%**					**None**
13	05/01/2010 10:58	294.5							
	05/01/2010 10:58				-0.31	-0.11%	7.46%		
	05/01/2010 11:01	293.85	-0.65	-0.22%					
			-0.65	**-0.22%**	**-0.31**	**-0.11%**		**0.12%**	**Direction**
14	05/01/2010 14:57	296.5							
	05/01/2010 14:59				0.05	0.02%	14.93%		
	05/01/2010 15:02	296.4	-0.1	-0.03%					
			-0.1	**-0.03%**	**0.05**	**0.02%**		**0.05%**	**Marginal**
15	06/01/2010 10:58	296							
	06/01/2010 11:00				0.25	0.08%	29.11%		
	06/01/2010 11:02	296.4	0.4	0.14%					
			0.4	**0.14%**	**0.25**	**0.08%**		**0.05%**	**Direction**
16	06/01/2010 14:58	300.6							
	06/01/2010 15:00				0.25	0.08%	29.11%		
	06/01/2010 15:03	301.1	0.5	0.17%					
			0.5	**0.17%**	**0.25**	**0.08%**		**0.08%**	**Direction**
17	07/01/2010 10:59	310.55							
	07/01/2010 10:59				-0.04	-0.01%	14.93%		
	07/01/2010 10:59				0	0.00%	11.20%		
	07/01/2010 11:02	309.95	-0.6	-0.19%					
			-0.6	**-0.19%**	**-0.04**	**-0.01%**		**0.18%**	**Direction**
18	07/01/2010 14:57	315.25							
	07/01/2010 14:57				-0.5	-0.16%	8.09%		
	07/01/2010 15:00	315.2	-0.05	-0.02%					
			-0.05	**-0.02%**	**-0.5**	**-0.16%**		**0.14%**	**Direction**
19	08/01/2010 10:57	323.83							
	08/01/2010 11:00	324.25	0.42	0.13%					
	08/01/2010 11:01				-0.3	-0.09%	22.39%		
	08/01/2010 11:03	324.1	-0.15	-0.05%					
			0.27	**0.08%**	**-0.3**	**-0.09%**		**0.18%**	**Incorrect**
20	08/01/2010 14:57	320.25							
	08/01/2010 14:59				-0.75	-0.23%	17.64%		
	08/01/2010 15:08	321.65	1.4	0.44%					
			1.4	**0.44%**	**-0.75**	**-0.23%**		**0.67%**	**Incorrect**
21	11/01/2010 10:58	321.2							
	11/01/2010 10:58				0	0.00%	3.85%		
	11/01/2010 11:01	320.5	-0.7	-0.22%					
			-0.7	**-0.22%**	**0**	**0.00%**		**0.22%**	**Incorrect**
	11/01/2010 11:03				0.57	0.18%	7.46%		
	11/01/2010 11:05	319.05	-1.45	-0.45%					
			-2.15	**-0.67%**	**0.57**	**0.18%**		**0.85%**	**Incorrect**

22	11/01/2010 14:57	319.35							
	11/01/2010 14:57				-0.15	-0.05%	9.70%		
	11/01/2010 14:59				-0.75	-0.23%	8.82%		
	11/01/2010 15:02	319.4	0.05	0.02%					
			0.05	**0.02%**	**-0.9**	**-0.28%**		**0.30%**	**Incorrect**
	11/01/2010 15:03	319.3	-0.1	-0.03%	-0.3	-0.09%	7.46%		
			-0.05	**-0.02%**	**-1.2**	**-0.38%**		**0.36%**	**Direction**
23	12/01/2010 10:58	316.6							
	12/01/2010 10:58				0	0.00%	16.52%		
	12/01/2010 11:02	316.6	0	0.00%					
			0	**0.00%**	**0**	**0.00%**		**0.00%**	**Correct**
24	12/01/2010 14:57	314.8							
	12/01/2010 15:03	315.2	0.4	0.13%					
			0.4	**0.13%**					**None**
25	13/01/2010 10:59	312.45							
	13/01/2010 10:59				-0.03	-0.01%	9.70%		
	13/01/2010 11:01				0.2	0.06%	8.09%		
	13/01/2010 11:03	312.3	-0.15	-0.05%					
			-0.15	**-0.05%**	**0.17**	**0.05%**		**0.10%**	**Incorrect**
26	13/01/2010 14:56	312.7							
	13/01/2010 14:56				-0.1	-0.03%	14.93%		
	13/01/2010 14:58				0.6	0.19%	5.60%		
	13/01/2010 15:01	311.7	-1	-0.32%					
			-1	**-0.32%**	**0.5**	**0.16%**		**0.48%**	**Incorrect**
27	14/01/2010 10:57	317.7							
	14/01/2010 11:00	317.85	0.15	0.05%					
	14/01/2010 11:03	317.65	-0.05						
	14/01/2010 11:07	317.35	-0.35						
	14/01/2010 11:10	318.15	0.45						
	14/01/2010 11:11				0	0.00%	35.60%		
	14/01/2010 11:13	317.85	0.15						
			0.1	**0.03%**	**0**	**0.00%**		**0.03%**	**Marginal**
28	14/01/2010 14:56	319.1							
	14/01/2010 15:01	319.05	-0.05	-0.02%					
			-0.05	**-0.02%**					**None**

	HSBA								
Stop	**Date**	**Value**	**Change**		**Predicted**			**Comparison**	
			Pts	**%**	**Pts**	**%**	**Prob**	**Variation**	**Direction**
1	01/12/2009 10:56	723.5							
	01/12/2009 10:56				0	0.00%	8.26%		
	01/12/2009 11:00	723.4	-0.1	-0.01%					
			-0.1	**-0.01%**	**0**	**0.00%**		**0.01%**	**Marginal**
2	01/12/2009 14:59	725.5							
	01/12/2009 14:59				1.5	0.21%	19.41%		
	01/12/2009 15:03	726.1	0.6	0.08%	0	0.00%	8.99%		
			0.6	**0.08%**	**1.5**	**0.21%**		**0.12%**	**Direction**
3	02/12/2009 10:57	716.2							
	02/12/2009 10:57				0	0.00%	4.13%		

	02/12/2009 11:00	716.1	-0.1	-0.01%					
			-0.1	**-0.01%**	**0**	**0.00%**		**0.01%**	**Marginal**
4	02/12/2009 14:59	722.5							
	02/12/2009 14:59				0	0.00%	7.46%		
	02/12/2009 15:04	722.7	0.2	0.03%					
			0.2	**0.03%**	**0**	**0.00%**		**0.03%**	**Marginal**
5	03/12/2009 10:57	734.6							
	03/12/2009 10:57				1.5	0.20%	9.70%		
	03/12/2009 11:00	734.7	0.1	0.01%					
			0.1	**0.01%**	**1.5**	**0.20%**		**0.19%**	**Direction**
6	03/12/2009 14:59	729.6							
	03/12/2009 14:59				0	0.00%	8.26%		
	03/12/2009 15:01	728.5	-1.1	-0.15%					
			-1.1	**-0.15%**	**0**	**0.00%**		**0.15%**	**Incorrect**
7	04/12/2009 10:59	714.1							
	04/12/2009 10:59				0	0.00%	17.64%		
	04/12/2009 11:02	714	-0.1	-0.01%					
			-0.1	**-0.01%**	**0**	**0.00%**		**0.01%**	**Marginal**
8	04/12/2009 14:58	730.3							
	04/12/2009 14:58				0.5	0.07%	8.09%		
	04/12/2009 15:03	731.8	1.5	0.21%					
			1.5	**0.21%**	**0.5**	**0.07%**		**0.14%**	**Direction**
9	08/12/2009 10:57	705.8							
	08/12/2009 11:00	705.5	-0.3	-0.04%					
			-0.3	**-0.04%**					**None**
10	08/12/2009 14:59	691.4							
	08/12/2009 15:00	693.3	1.9	0.27%					
			1.9	**0.27%**					**None**
11	04/01/2010 10:57								
	04/01/2010 10:57				-0.4		14.93%		
	04/01/2010 10:58				0		10.93%		
	04/01/2010 11:00				0.2		0.03%		
12	04/01/2010 14:59								
	04/01/2010 14:59				-0.4		14.93%		
	04/01/2010 15:00				0		15.22%		
	04/01/2010 15:02				0.2		0.00%		
13	05/01/2010 10:58								
	05/01/2010 10:58				-0.4		14.93%		
	05/01/2010 11:19				0.1		79.20%		
14	05/01/2010 14:57								
	05/01/2010 14:57				1.5		19.41%		
15	06/01/2010 10:58	736.3							
	06/01/2010 10:58				0	0.00%	15.63%		
	06/01/2010 11:05	737.2	0.9	0.12%					
			0.9	**0.12%**	**0**	**0.00%**		**0.12%**	**Incorrect**
16	06/01/2010 14:58	737.7							
	06/01/2010 15:03	737.7	0	0.00%					

	06/01/2010 15:05	738	0.3	0.04%						
			0.3	**0.04%**						**None**
17	07/01/2010 10:59	733.7								
	07/01/2010 10:59				0	0.00%	12.38%			
	07/01/2010 11:02	732.8	-0.9	-0.12%						
			-0.9	**-0.12%**	**0**	**0.00%**		**0.12%**		**Incorrect**
18	07/01/2010 14:57	736								
	07/01/2010 14:59				0.2	0.03%	7.46%			
	07/01/2010 15:00	735.7	-0.3	-0.04%						
			-0.3	**-0.04%**	**0.2**	**0.03%**		**0.07%**		**Marginal**
19	08/01/2010 10:57	738.1								
	08/01/2010 10:57				-0.5	-0.07%	16.17%			
	08/01/2010 10:59				0.2	0.03%	7.46%			
	08/01/2010 11:00	738.9	0.8	0.11%						
			0.8	**0.11%**	**-0.3**	**-0.04%**		**0.15%**		**Incorrect**
	08/01/2010 11:03	737.5	-1.4	-0.19%						
	08/01/2010 11:04				0	0.00%	5.51%			
	08/01/2010 11:07	736.7	-0.8	-0.11%						
			-1.4	**-0.19%**	**-0.3**	**-0.04%**		**0.15%**		**Direction**
20	08/01/2010 14:57	736.3								
	08/01/2010 15:01				-0.3	-0.04%	74.64%			
	08/01/2010 15:08	738.1	1.8	0.24%						
			1.8	**0.24%**	**-0.3**	**-0.04%**		**0.29%**		**Incorrect**
21	11/01/2010 10:58	736.2								
	11/01/2010 11:00				0.2	0.03%	14.93%			
	11/01/2010 11:01	735.8	-0.4	-0.05%						
			-0.4	**-0.05%**	**0.2**	**0.03%**		**0.08%**		**Marginal**
22	11/01/2010 14:57	733.2								
	11/01/2010 14:57				0	0.00%	8.82%			
	11/01/2010 15:02	734	0.8	0.11%						
			0.8	**0.11%**	**0**	**0.00%**		**0.11%**		**Incorrect**
23	12/01/2010 10:58	731								
	13/01/2010 10:58				0	0.00%	16.74%			
	12/01/2010 11:02	730.7	-0.3	-0.04%						
			-0.3	**-0.04%**	**0**	**0.00%**		**0.04%**		**Marginal**
24	12/01/2010 14:58	730.1								
	12/01/2010 15:03	729.7	-0.4	-0.05%						
			-0.4	**-0.05%**						**None**
25	13/01/2010 10:59	718.7								
	13/01/2010 10:59				0	0.00%	6.86%			
	13/01/2010 10:59				-0.6	-0.08%	7.46%			
	13/01/2010 10:59				0	0.00%	7.46%			
	13/01/2010 11:00				-0.3	-0.04%	7.46%			
	13/01/2010 11:03	718.1	-0.6	-0.08%						
			-0.6	**-0.08%**	**-0.9**	**-0.13%**		**0.04%**		**Direction**
26	13/01/2010 14:56	719.3								
	13/01/2010 14:56				0	0.00%	4.44%			
	13/01/2010 14:57				-0.3	-0.04%	1.27%			
	13/01/2010 15:01	718.1	-1.2	-0.17%						
			-1.2	**-0.17%**	**-0.3**	**-0.04%**		**0.13%**		**Direction**
	13/01/2010 15:02				0	0.00%	1.66%			

	13/01/2010 15:03	717.5	-0.6						
			-1.8	**-0.25%**	**-0.3**	**-0.04%**		**0.21%**	**Direction**
27	14/01/2010 10:57	717.5							
	14/01/2010 11:00	718.4	0.9	0.13%					
	14/01/2010 11:03	718.3	-0.1						
	14/01/2010 11:07	718	-0.3						
	14/01/2010 11:10	719	1						
	14/01/2010 11:11				0	0.00%	24.07%		
	14/01/2010 11:13	718.7	-0.3						
			1.2	**0.17%**	**0**	**0.00%**		**0.17%**	**Incorrect**
28	14/01/2010 14:56	714.5							
	14/01/2010 15:01	715	0.5	0.07%					
			0.5	**0.07%**					**None**

	LLOY								
Stop	**Date**	**Value**	**Change**		**Predicted**			**Comparison**	
			Pts	**%**	**Pts**	**%**	**Prob**	**Variation**	**Direction**
1	01/12/2009 10:56	54.62							
	01/12/2009 10:56				0	0.00%	8.26%		
	01/12/2009 11:00	54.64	0.02	0.04%					
			0.02	**0.04%**	**0**	**0.00%**		**0.04%**	**Marginal**
2	01/12/2009 14:59	54.22							
	01/12/2009 15:01				0	0.00%	16.51%		
	01/12/2009 15:04	54.24	0.02	0.04%					
			0.02	**0.04%**	**0**	**0.00%**		**0.04%**	**Marginal**
3	02/12/2009 10:57	51.5							
	02/12/2009 10:57				0	0.00%	4.13%		
	02/12/2009 11:00	51.23	-0.27	-0.52%					
			-0.27	**-0.52%**	**0**	**0.00%**		**0.52%**	**Incorrect**
4	02/12/2009 14:59	53.12							
	02/12/2009 15:04	53.28	0.16	0.30%					
			0.16	**0.30%**					**None**
5	03/12/2009 10:57	55.34							
	03/12/2009 10:57				0	0.00%	6.19%		
	03/12/2009 11:00	55.45	0.11	0.20%					
			0.11	**0.20%**	**0**	**0.00%**		**0.20%**	**Incorrect**
6	03/12/2009 14:59	55.91							
	03/12/2009 14:59				-0.07	-0.13%	7.46%		
	03/12/2009 15:01	55.78	-0.13	-0.23%					
			-0.13	**-0.23%**	**-0.07**	**-0.13%**		**0.11%**	**Direction**
7	04/12/2009 10:59	54.05							
	05/12/2009 10:59				0	0.00%	4.88%		
	04/12/2009 11:02	53.98	-0.07	-0.13%					
			-0.07	**-0.13%**	**0**	**0.00%**		**0.13%**	**Incorrect**
8	04/12/2009 14:58	56.43							
	05/12/2009 14:58				0	0.00%	12.39%		
	04/12/2009 15:03	56.51	0.08	0.14%					
			0.08	**0.14%**	**0**	**0.00%**		**0.14%**	**Incorrect**
9	08/12/2009 10:57	54.18							
	08/12/2009 11:00	54.29	0.11	0.20%					

			0.11	**0.20%**					**None**
10	08/12/2009 14:59	53.25							
	08/12/2009 15:00	52.98	-0.27	-0.51%					
			-0.27	**-0.51%**					**None**
11	04/01/2010 10:57	51.93							
	05/01/2010 10:57				0	0.00%	4.55%		
	04/01/2010 11:01	51.86	-0.07	-0.13%					
			-0.07	**-0.13%**	**0**	**0.00%**		**0.13%**	**Incorrect**
12	04/01/2010 14:59	51.95							
	04/01/2010 14:59				0	0.00%	14.66%		
	04/01/2010 15:01	51.94	-0.01	-0.02%					
			-0.01	**-0.02%**	**0**	**0.00%**		**0.02%**	**Marginal**
13	05/01/2010 10:58	54.07							
	05/01/2010 10:59				0.08	0.15%	7.46%		
	05/01/2010 11:01	54.23	0.16	0.30%					
			0.16	**0.30%**	**0.08**	**0.15%**		**0.15%**	**Direction**
14	05/01/2010 14:57	53.89							
	05/01/2010 15:02	53.9	0.01	0.02%	0	0.00%	13.96%		
			0.17	**0.32%**	**0.08**	**0.15%**		**0.17%**	**Direction**
15	06/01/2010 10:58	53.5							
	06/01/2010 10:58				0	0.00%	14.93%		
	06/01/2010 11:02	53.35	-0.15	-0.28%					
			-0.15	**-0.28%**	**0**	**0.00%**		**0.28%**	**Incorrect**
16	06/01/2010 14:58	54.31							
	06/01/2010 15:03	54.27	-0.04	-0.07%					
			-0.04	**-0.07%**					**None**
17	07/01/2010 10:59	55.25							
	07/01/2010 11:02		-0.1	-0.18%					
	07/01/2010 11:03				0	0.00%	8.58%		
	07/01/2010 11:05	55.1	-0.15	-0.27%					
			-0.25	**-0.45%**	**0**	**0.00%**		**0.45%**	**Incorrect**
18	07/01/2010 14:57	56.6							
	07/01/2010 15:00	56.64	0.04	0.07%					
			0.04	**0.07%**					**None**
19	08/01/2010 10:57	56.92							
	08/01/2010 10:57				0	0.00%	3.84%		
	08/01/2010 11:00	56.97	0.05	0.09%					
			0.05	**0.09%**	**0**	**0.00%**		**0.09%**	**Marginal**
20	08/01/2010 14:57	56.77							
	08/01/2010 14:57				0	0.00%	7.70%		
	08/01/2010 15:08	56.95	0.18	0.32%					
			0.18	**0.32%**	**0**	**0.00%**		**0.32%**	**Incorrect**
21	11/01/2010 10:58	56.98							
	11/01/2010 11:01	56.86	-0.12	-0.21%					
			-0.12	**-0.21%**					**None**
22	11/01/2010 14:57	57.02							
	11/01/2010 15:02	57	-0.02	-0.04%					
			-0.02	**-0.04%**					**None**
23	12/01/2010 10:58	55.95							
	12/01/2010 10:58				0	0.00%	15.82%		
	12/01/2010 11:02	55.9	-0.05	-0.09%					

			-0.05	**-0.09%**	**0**	**0.00%**		**0.09%**	**Marginal**
24	12/01/2010 14:57	55.31							
	12/01/2010 15:03	55.62	0.31	0.56%					
			0.31	**0.56%**					**None**
25	13/01/2010 10:59	55.47							
	13/01/2010 10:59				0	0.00%	6.53%		
	13/01/2010 11:03	55.44	-0.03	-0.05%	-0.02	-0.04%	3.57%		
			-0.03	**-0.05%**	**-0.02**	**-0.04%**		**0.02%**	**Direction**
	13/01/2010 11:03				0	0.00%	4.75%		
	13/01/2010 11:04				0.08	0.14%	7.46%		
	13/01/2010 11:06	55.43	-0.01						
			-0.04	**-0.07%**	**0.08**	**0.14%**		**0.22%**	**Incorrect**
26	13/01/2010 14:56	55.66							
	13/01/2010 14:56				0	0.00%	7.26%		
	13/01/2010 14:56				-0.07	-0.13%	1.28%		
	13/01/2010 15:01	55.68	0.02	0.04%	0		0.80%		
			0.02	**0.04%**	**-0.07**	**-0.13%**		**0.16%**	**Incorrect**
27	14/01/2010 10:57	57.98							
	14/01/2010 11:00	57.89	-0.09	-0.16%					
	14/01/2010 11:03	57.77	-0.12	-0.21%					
	14/01/2010 11:07	57.7	-0.07	-0.12%					
	14/01/2010 11:10	57.71	0.01	0.02%					
	14/01/2010 11:11				0	0.00%	66.00%		
	14/01/2010 11:13	57.73	0.02	0.03%					
			-0.25	**-0.43%**	**0**	**0.00%**		**0.43%**	**Incorrect**
28	14/01/2010 14:56	57.3							
	14/01/2010 15:01	57.3	0	0.00%					
			0	**0.00%**					**None**

	RBS								
Stop	**Date**	**Value**	**Change**		**Predicted**			**Comparison**	
			Pts	**%**	**Pts**	**%**	**Prob**	**Variation**	**Direction**
1	01/12/2009 10:56	34.07							
	01/12/2009 11:00	34.15	0.08	0.23%	0.12	0.35%	8.82%		
			0.08	**0.23%**	**0.12**	**0.35%**		**0.12%**	**Direction**
2	01/12/2009 14:59	34.09							
	01/12/2009 14:59				0	0.00%	11.20%		
	01/12/2009 15:01				0.23	0.67%	4.26%		
	01/12/2009 15:03	34.14	0.05	0.15%					
			0.05	**0.15%**	**0.23**	**0.67%**		**0.53%**	**Direction**
3	02/12/2009 10:57	32.2							
	02/12/2009 10:57				-0.01	-0.03%	14.93%		
	02/12/2009 11:00	32.07	-0.13	-0.40%					
			-0.13	**-0.40%**	**-0.01**	**-0.03%**		**0.37%**	**Direction**
4	02/12/2009 14:59	32.58							
	02/12/2009 14:59				-0.03	-0.09%	7.46%		
	02/12/2009 15:04	32.62	0.04	0.12%					
			0.04	**0.12%**	**-0.03**	**-0.09%**		**0.21%**	**Incorrect**
5	03/12/2009 10:57	34.71							
	03/12/2009 10:57				0	0.00%	6.19%		

	03/12/2009 11:00	34.81	0.1	0.29%					
			0.1	**0.29%**	**0**	**0.00%**		**0.29%**	**Incorrect**
6	03/12/2009 14:59	34.92							
	03/12/2009 15:00				-0.05	-0.14%	16.17%		
	03/12/2009 15:01	34.9	-0.02	-0.06%					
			-0.02	**-0.06%**	**-0.05**	**-0.14%**		**0.09%**	**Direction**
7	04/12/2009 10:59	34.4							
	04/12/2009 10:59				-0.06	-0.17%	10.78%		
	04/12/2009 11:02	34.35	-0.05	-0.15%					
			-0.05	**-0.15%**	**-0.06**	**-0.17%**		**0.03%**	**Direction**
8	04/12/2009 14:58	34.99							
	04/12/2009 15:00				0.23	0.66%	7.46%		
	04/12/2009 15:03	34.95	-0.04	-0.11%					
			-0.04	**-0.11%**	**0.23**	**0.66%**		**0.77%**	**Incorrect**
	04/12/2009 15:04	34.89	-0.06	-0.17%	0.12	0.34%	8.82%		
			-0.1	**-0.29%**	**0.35**	**1.00%**		**1.29%**	**Incorrect**
9	08/12/2009 10:57	31.77							
	08/12/2009 11:00	31.86	0.09	0.28%					
			0.09	**0.28%**					**None**
10	08/12/2009 14:59	30.34							
	08/12/2009 15:00	30.42	0.08	0.26%					
	08/12/2009 15:03				0.03	0.10%	17.64%		
	08/12/2009 15:04	30.77	0.35	1.15%					
			0.43	**1.42%**	**0.03**	**0.10%**		**1.32%**	**Direction**
11	04/01/2010 10:57	30.58							
	04/01/2010 10:57				-0.06	-0.20%	10.78%		
	04/01/2010 11:07	30.62	0.04	0.13%					
			0.04	**0.13%**	**-0.06**	**-0.20%**		**0.33%**	**Incorrect**
12	04/01/2010 14:59	31.65							
	04/01/2010 15:00	31.65	0	0.00%					
			0	**0.00%**					**None**
13	05/01/2010 10:58	34.53							
	05/01/2010 11:00				0.23	0.67%	7.46%		
	05/01/2010 11:01	34.56	0.03	0.09%					
			0.03	**0.09%**	**0.23**	**0.67%**		**0.58%**	**Direction**
14	05/01/2010 14:57	35.1							
	05/01/2010 14:57				0.03	0.09%	8.14%		
	05/01/2010 14:57				0	0.00%	7.46%		
	05/01/2010 15:01				0.03	0.09%	8.82%		
	05/01/2010 15:02	35.09	-0.01	-0.03%					
			-0.01	**-0.03%**	**0.06**	**0.17%**		**0.20%**	**Incorrect**
15	06/01/2010 10:58	36.02							
	06/01/2010 11:02	36.07	0.05	0.14%	-0.02	-0.06%	32.34%		
			0.05	**0.14%**	**-0.02**	**-0.06%**		**0.19%**	**Incorrect**
16	06/01/2010 14:58	36.36							
	06/01/2010 14:58				0.05	0.14%	8.09%		
	06/01/2010 15:00				0.1	0.28%	7.46%		
	06/01/2010 15:02	36.44	0.08	0.22%					
			0.08	**0.22%**	**0.15**	**0.41%**		**0.19%**	**Direction**
17	07/01/2010 10:59	35.97							
	07/01/2010 10:59				-0.03	-0.08%	7.46%		

	07/01/2010 10:59				0	0.00%	11.20%		
	07/01/2010 11:02	35.89	-0.08	-0.22%					
			-0.08	**-0.22%**	**-0.03**	**-0.08%**		**0.14%**	**Direction**
	07/01/2010 11:03				0.03	0.08%	8.82%		
	07/01/2010 11:05	35.87	-0.02	-0.06%					
			-0.1	**-0.28%**	**-0.03**	**-0.08%**		**0.19%**	**Direction**
18	07/01/2010 14:57	36.06							
	07/01/2010 15:00	36	-0.06	-0.17%					
			-0.06	**-0.17%**					**None**
19	08/01/2010 10:57	36.1							
	08/01/2010 10:57				0	0.00%	3.84%		
	08/01/2010 11:00	36.08	-0.02	-0.06%					
			-0.02	**-0.06%**	**0**	**0.00%**		**0.06%**	**Marginal**
20	08/01/2010 14:57	34.91							
	08/01/2010 14:57				0	0.00%	7.70%		
	08/01/2010 15:08	35.08	0.17	0.49%					
			0.17	**0.49%**	**0**	**0.00%**		**0.49%**	**Incorrect**
21	11/01/2010 10:58	34.36							
	11/01/2010 10:59				-0.06	-0.17%	19.41%		
	11/01/2010 11:01	34.14	-0.22	-0.64%					
			-0.22	**-0.64%**	**-0.06**	**-0.17%**		**0.47%**	**Direction**
22	11/01/2010 14:57	34.59							
	11/01/2010 15:02	34.7	0.11	0.32%					
			0.11	**0.32%**					**None**
23	12/01/2010 10:58	34.83							
	12/01/2010 10:58				0	0.00%	16.76%		
	12/01/2010 11:02	34.76	-0.07	-0.20%					
			-0.07	**-0.20%**	**0**	**0.00%**		**0.20%**	**Incorrect**
24	12/01/2010 14:57	34.82							
	12/01/2010 15:02	34.88	0.06	0.17%					
			0.06	**0.17%**					**None**
25	13/01/2010 10:59	34.73							
	13/01/2010 11:03	34.72	-0.01	-0.03%	0.03	0.09%	8.82%		
			-0.01	**-0.03%**	**0.03**	**0.09%**		**0.12%**	**Incorrect**
26	13/01/2010 14:56	35.38							
	13/01/2010 14:57				-0.06	-0.17%	19.41%		
	13/01/2010 15:01	35.4	0.02	0.06%	0.03	0.08%	8.82%		
			0.02	**0.06%**	**-0.03**	**-0.08%**		**0.14%**	**Incorrect**
	13/01/2010 15:03		0	0.00%					
	13/01/2010 15:06	35.31	-0.09	-0.25%	0	0.00%	4.66%		
			-0.07	**-0.20%**	**-0.03**	**-0.08%**		**0.11%**	**Direction**
27	14/01/2010 10:57	35.84							
	14/01/2010 10:57				0	0.00%	11.62%		
	14/01/2010 11:00	35.87	0.03	0.08%					
			0.03	**0.08%**	**0**	**0.00%**		**0.08%**	**Marginal**
	14/01/2010 11:01				0.12	0.33%	4.48%		
	14/01/2010 11:03	35.84	-0.03	-0.08%					
			0	**0.00%**	**0.12**	**0.33%**		**0.33%**	**Incorrect**
	14/01/2010 11:07	35.76	-0.08						
	14/01/2010 11:10	35.78	0.02						
	14/01/2010 11:11				0	0.00%	5.48%		

	12/01/2010 11:13				0.08	0.22%	5.75%		
	13/01/2010 11:13				-0.01	-0.03%	6.76%		
	14/01/2010 11:13	35.81	0.03		0	0.00%	9.70%		
			-0.03	**-0.08%**	**0.19**	**0.53%**		**0.61%**	**Incorrect**
28	14/01/2010 14:56	35.95							
	14/01/2010 15:01	35.95	0	0.00%					
			0	**0.00%**					**None**

	STAN								
Stop	**Date**	**Value**	**Change**		**Predicted**			**Comparison**	
			Pts	**%**	**Pts**	**%**	**Prob**	**Variation**	**Direction**
1	01/12/2009 10:56	1529.5							
	01/12/2009 10:58				1.5	0.10%	16.17%		
	01/12/2009 11:00	1530.5	1	0.07%					
			1	**0.07%**	**1.5**	**0.10%**		**0.03%**	**Direction**
2	01/12/2009 14:59	1556.5							
	01/12/2009 14:59				0	0.00%	29.86%		
	01/12/2009 15:04	1556	-0.5	-0.03%					
			-0.5	**-0.03%**	**0**	**0.00%**		**0.03%**	**Marginal**
3	02/12/2009 10:57	1542							
	02/12/2009 10:58				-1.5	-0.10%	22.39%		
	02/12/2009 11:00	1542	0	0.00%					
			0	**0.00%**	**-1.5**	**-0.10%**		**0.10%**	**Marginal**
4	02/12/2009 14:59	1550							
	02/12/2009 15:01				1	0.06%	7.46%		
	02/12/2009 15:04	1553	3	0.19%					
			3	**0.19%**	**1**	**0.06%**		**0.13%**	**Direction**
5	03/12/2009 10:57	1571							
	03/12/2009 10:58				0.5	0.03%	24.26%		
	03/12/2009 11:00	1572.5	1.5	0.10%					
			1.5	**0.10%**	**0.5**	**0.03%**		**0.06%**	**Direction**
6	03/12/2009 14:59	1573							
	03/12/2009 15:01	1570	-3	-0.19%					
	03/12/2009 15:03	1570	0	0.00%	-1.5	-0.10%	19.41%		
			-3	**-0.19%**	**-1.5**	**-0.10%**		**0.10%**	**Direction**
7	04/12/2009 10:59	1507							
	04/12/2009 11:00				0	0.00%	22.39%		
	04/12/2009 11:02	1506	-1	-0.07%					
			-1	**-0.07%**	**0**	**0.00%**		**0.07%**	**Marginal**
8	04/12/2009 14:58	1536							
	04/12/2009 14:59				0.5	0.03%	8.09%		
	04/12/2009 15:02	1537	1	0.07%					
			1	**0.07%**	**0.5**	**0.03%**		**0.03%**	**Direction**
9	08/12/2009 10:57	1506							
	09/12/2009 10:58				1.5	0.10%	14.93%		
	08/12/2009 11:00	1505	-1	-0.07%					
			-1	**-0.07%**	**1.5**	**0.10%**		**0.17%**	**Incorrect**
10	08/12/2009 14:59	1443.5							
	08/12/2009 15:00	1449	5.5	0.38%					
	08/12/2009 15:01				1	0.07%	14.93%		

	08/12/2009 15:04	1447.5	-1.5	-0.10%					
			4	**0.28%**	**1**	**0.07%**		**0.21%**	**Direction**
11	04/01/2010 10:57	1593							
	04/01/2010 10:59				1	0.06%	21.56%		
	04/01/2010 11:01	1592.5	-0.5	-0.03%					
			-0.5	**-0.03%**	**1**	**0.06%**		**0.09%**	**Marginal**
12	04/01/2010 14:59	1604.5							
	04/01/2010 15:01	1604	-0.5	-0.03%					
			-0.5	**-0.03%**					**None**
13	05/01/2010 10:58	1630							
	05/01/2010 11:01	1628	-2	-0.12%					
			-2	**-0.12%**					**None**
14	05/01/2010 14:57	1631							
	05/01/2010 14:58				5.5	0.34%	22.39%		
	05/01/2010 15:02	1627.5	-3.5	-0.21%					
			-3.5	**-0.21%**	**5.5**	**0.34%**		**0.55%**	**Incorrect**
15	06/01/2010 10:58	1615.35							
	06/01/2010 10:59				-0.3	-0.02%	14.93%		
	06/01/2010 11:01				4.5	0.28%	14.93%		
	06/01/2010 11:02	1613	-2.35	-0.15%					
			-2.35	**-0.15%**	**4.2**	**0.26%**		**0.41%**	**Incorrect**
	06/01/2010 11:05	1615	2	0.12%					
	06/01/2010 11:08	1616.5	1.5	0.09%					
	06/01/2010 11:10				0	0.00%	11.20%		
	06/01/2010 11:12	1616	-0.5	-0.03%					
			0.65	**0.04%**	**4.2**	**0.26%**		**0.22%**	**Direction**
16	06/01/2010 14:58	1615.5							
	06/01/2010 14:59				-1	-0.06%	17.64%		
	06/01/2010 15:03	1616	0.5	0.03%					
			0.5	**0.03%**	**-1**	**-0.06%**		**0.09%**	**Marginal**
17	07/01/2010 10:59	1589							
	07/01/2010 11:00				5.5	0.35%	14.93%		
	07/01/2010 11:02	1589	0	0.00%					
			0	**0.00%**	**5.5**	**0.35%**		**0.35%**	**Incorrect**
18	07/01/2010 14:57	1597.5							
	07/01/2010 14:58				-2	-0.13%	16.17%		
	07/01/2010 15:00	1598	0.5	0.03%					
			0.5	**0.03%**	**-2**	**-0.13%**		**0.16%**	**Incorrect**
19	08/01/2010 10:57	1610.5							
	08/01/2010 10:59				0	0.00%	8.09%		
	08/01/2010 11:00	1615	4.5	0.28%					
			4.5	**0.28%**	**0**	**0.00%**		**0.28%**	**Incorrect**
20	08/01/2010 14:57	1614.5							
	08/01/2010 15:01				-0.5	-0.03%	24.26%		
	08/01/2010 15:08	1617	2.5	0.15%					
			2.5	**0.15%**	**-0.5**	**-0.03%**		**0.19%**	**Incorrect**
21	11/01/2010 10:58	1613.5							
	11/01/2010 11:00				0.5	0.03%	7.46%		
	11/01/2010 11:01	1613.5	0	0.00%					
			0	**0.00%**	**0.5**	**0.03%**		**0.03%**	**Marginal**
22	11/01/2010 14:57	1601.5							

	11/01/2010 14:57				7	0.44%	7.46%		
	11/01/2010 15:02	1602.5	1	0.06%					
			1	**0.06%**	**7**	**0.44%**		**0.37%**	**Direction**
23	12/01/2010 10:58	1576							
	12/01/2010 11:01				1.5	0.10%	7.46%		
	12/01/2010 11:02	1576.11	0.11	0.01%					
			0.11	**0.01%**	**1.5**	**0.10%**		**0.09%**	**Direction**
24	12/01/2010 14:57	1552.5							
	12/01/2010 14:58				1.5	0.10%	7.46%		
	12/01/2010 15:03	1554	1.5	0.10%					
			1.5	**0.10%**	**1.5**	**0.10%**		**0.00%**	**Correct**
25	13/01/2010 10:59	1549.5							
	13/01/2010 11:03	1548.5	-1	-0.06%	-0.5	-0.03%	24.26%		
			-1	**-0.06%**	**-0.5**	**-0.03%**		**0.06%**	**Direction**
26	13/01/2010 14:56	1557							
	13/01/2010 14:58				2	0.13%	9.70%		
	13/01/2010 15:01	1555	-2	-0.13%	1.5	0.10%	7.46%		
			-2	**-0.13%**	**3.5**	**0.22%**		**0.35%**	**Incorrect**
27	14/01/2010 10:57	1560							
	14/01/2010 10:58				0	0.00%	14.93%		
	14/01/2010 11:00	1559.5	-0.5	-0.03%					
			-0.5	**-0.03%**	**0**	**0.00%**		**0.03%**	**Marginal**
28	14/01/2010 14:56	1565.5							
	14/01/2010 15:01	1567	1.5	0.10%					
			1.5	**0.10%**					**None**