Modelling and Simulation for the Analysis of Securities Markets

Rui Hu[1,2], Vadim Mazalov[1,3] and Stephen M. Watt[1]

[1]University of Western Ontario, London Ontario, Canada

[2]Quantica Trading, Kitchener Ontario, Canada

[3]Amazon Canada, Toronto, Ontario, Canada

SCSS 2014

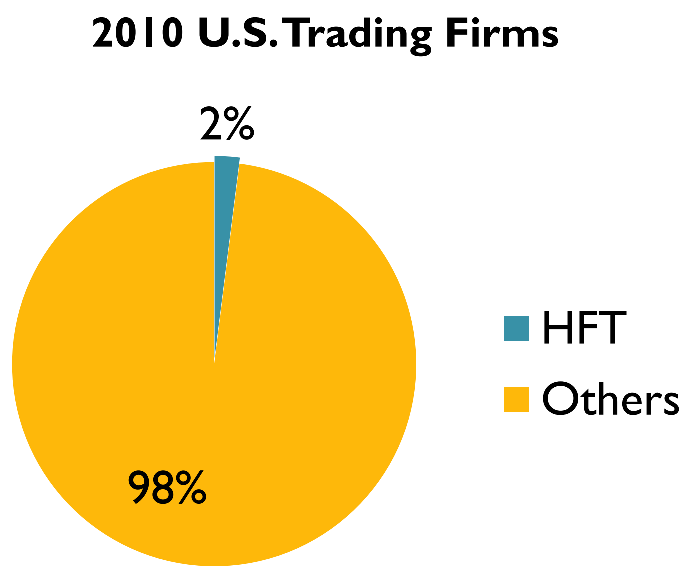
Gammarth, La Marsa, Tunisia, 7 December 2014

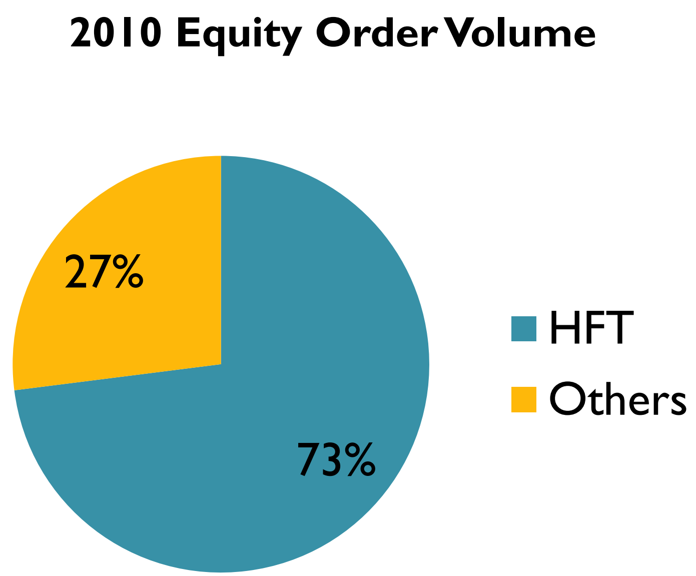
Outline

- Introduction
- A Trading Model
- A Market Simulator
- Conclusion

High-Frequency Trading

- Uses algorithms to automate trading activities
- Has grown rapidly around the globe
- Has dramatically changed how securities are traded





Problems

High-frequency trading still faces many challenges.

We are particularly interested in two sub-problems:

- ***strategy modelling***, where accurate and efficient prediction of securities' price movement is required, and strategy evaluation, and
- ***strategy evaluation***, where strategies must be examined in a variety of market conditions before being launched in real markets.

The Modelling Problem

- Accurate and efficient short term prediction of one change in the price of an asset.

- A number of strategies developed over time, from simple and fast to sophisticated models. These include methods based on time series analysis, support vector machines, hidden Markov models, nearest neighbor classifiers, etc.

Our Approach

- Observations as points in a multi-dimensional space of numerical technical indicators.
- **Clusters of points** representing **price movements up** and **down**.
- Points weighted by distance to cluster centroids.
- Predictions made when classification confidence is high enough.
- Groups of new points added as events occur, updating clusters.

Technical Indicators

- Choose a set of numerical indicators.
  - Too few => loss of precision.
  - Too many => too sparse (since high dimension)

- $ROC_i = w \frac{\frac{b_i}{o_i} - \frac{b_{i-1}}{o_{i-1}}}{t_i - t_{i-1}} + (1-w) ROC_{i-1}$

  b_i bid depth o_i offer depth
- n_{cb} (n_{co}) number of times an exchange/ATS locked the market on the bid (offer)
- n_{lb} (n_{lo}) number of times an exchange/ATS left the NBBO on the bid (offer)
- s_b (s_o) sum of weights of venues with bid (offer) equal to the NBBO bid (offer)

Form a $5E + 2$ dimensional space, with E exchanges.

The Classifier

- In high-frequency setting the classifier should be efficient.
- We compute the feature-weighted distance from a test sample to the centroid of a cluster, since this is one of the least expensive techniques in artificial intelligence.
- Training is performed by collecting points and finding their centroid.
- Classification is done by finding the squared weighted Euclidean distance to the centroid.

Classification vs Prediction

- Classification happens with each quote received – a feature vector is formed and the distances to centroids are evaluated.

- A prediction is made only if the distances between the sample and the centroids satisfy certain criteria, i.e. if the feature point is relatively close to one of the two centroids.

Experimental Settings

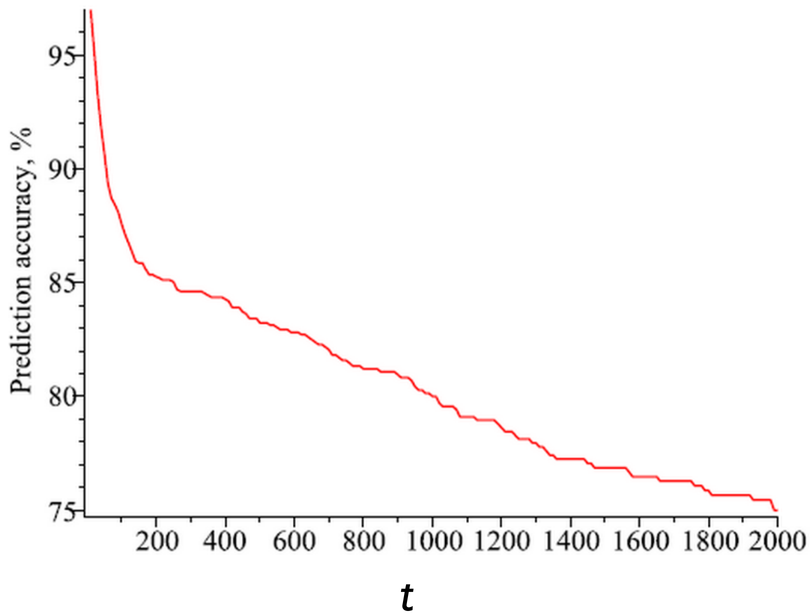
- We collected MSFT (Microsoft) securities, using quotes from 10 leading exchanges.
- The recorded events: change in bid/offer prices and bid/offer depth.
- We recorded several days in Dec 2011 with the total of 9,389,993 quotes and 4,658 price changes.
- Training was performed until both clusters had at least 10 points. The value of the weight in computation of the
- ROC was taken as 0.6.
- After 5 changes in price, parameters of a cluster were recomputed.

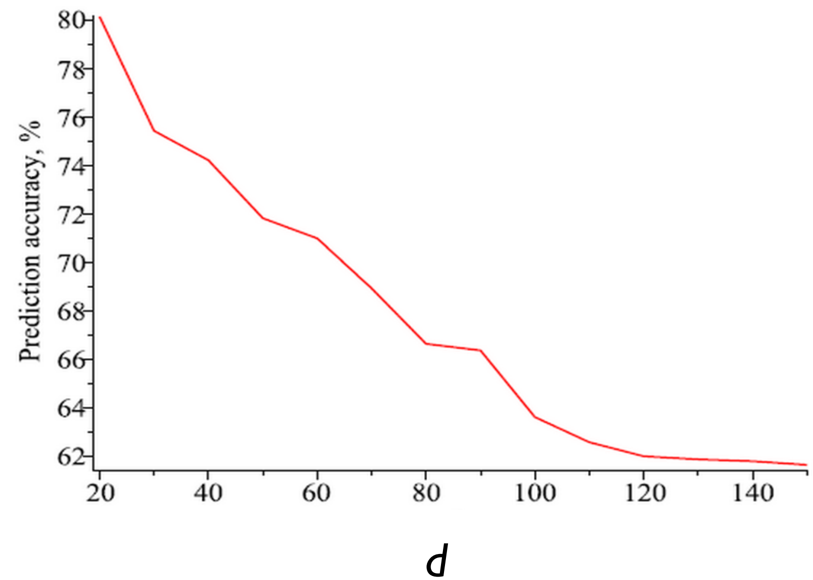
On-Change Accuracy

- The on-change distance was counted as correct if the distance to the centroid of the cluster in the direction of the price change was smaller than the distance to the other cluster.

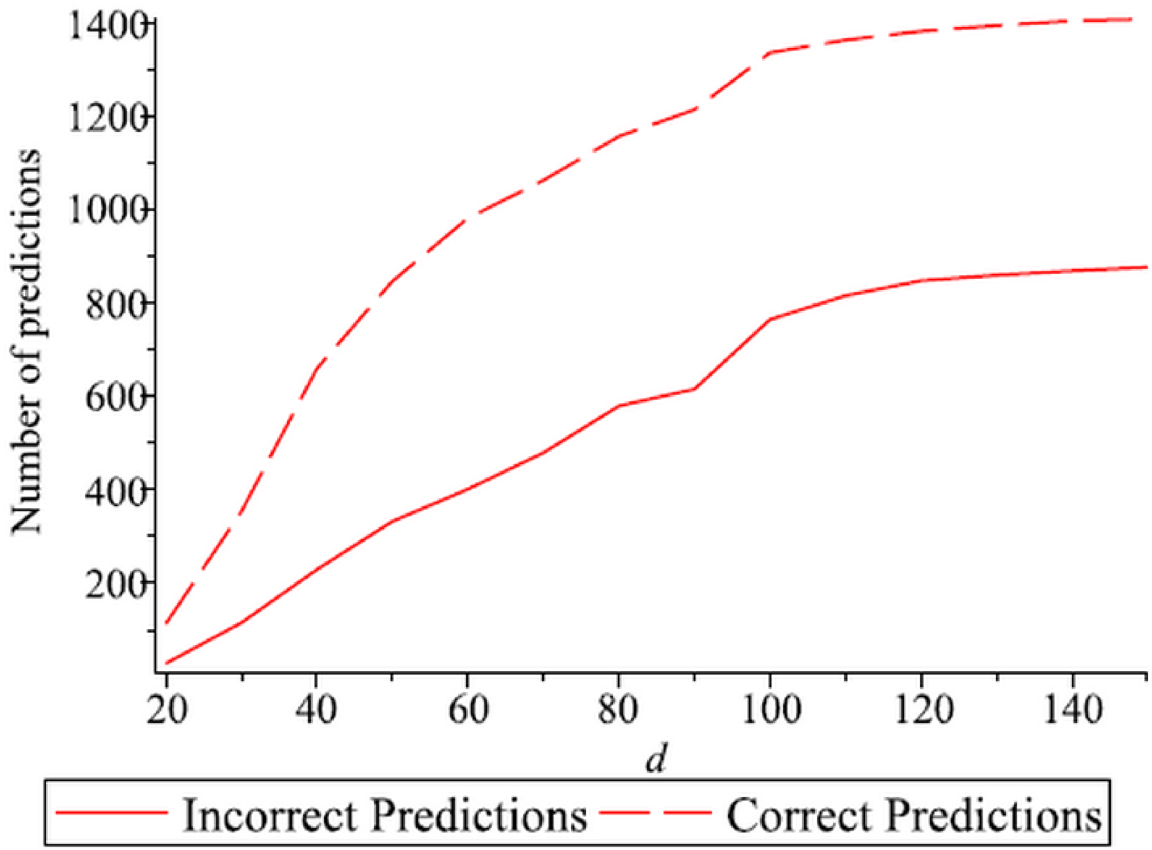
- The on-change accuracy of the model on the recorded data was 96.25%.

The Prediction Accuracy





The Prediction Accuracy



HFT is a double-edge sword

- **Advantages**:
  - Low latency
  - Rapid execution
  - Large input and output
- **Disadvantages**:
  - Any tiny mistake can lead to a tremendous disaster.

    e.g. On Aug. 1, 2012, Knight Capital lost $457.6 million in 15 mins due to a software glitch.

Therefore, trading strategies must be evaluated before launching them in real markets. This is carried out in practice through ***simulators***.

Our Motivations

Existing simulators significantly rely on

- *live market data*, or
- *historical data*

There are a number of limitations:

- live market is not always available
- testing strategies do not have any impact to the market
- back-testing has similar issues and can be over-fitting
- there are no standard protocols for interaction with users

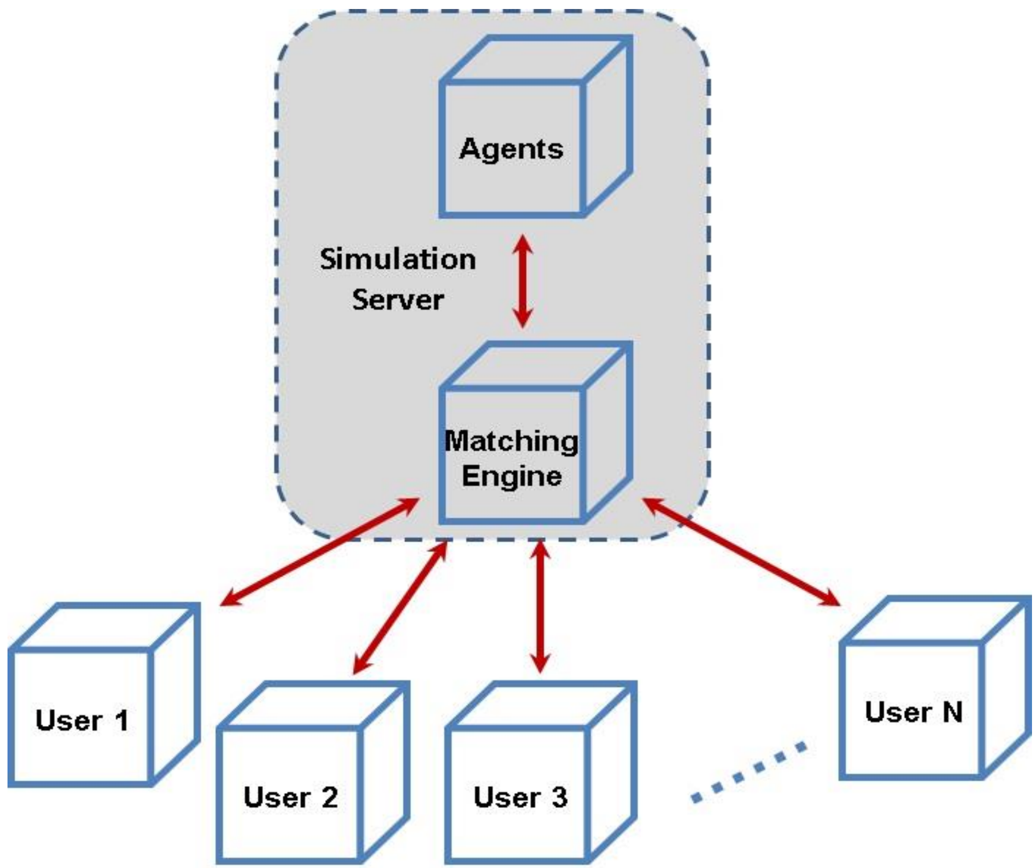
We wish to build a simulator that

- can support market simulation research and
- is suitable for evaluation of algorithmic trading strategies

Simulator Architecture

The simulator consists of

- a communication interface
- a matching engine
- a variety of simulated trading agents



Communication Interface – FIX

- FIX stands for Financial Information eXchange.
- It is a communication standard in global markets.

8=FIX.4.2 SOH 9=201 SOH 35=D SOH 49=Broker_16 SOH 56=MatchingEngine SOH
52=20140301-20:42:37.426 SOH 34=357256 SOH 1=Sim_Account SOH
11=8321660696624948305 SOH 21=1 SOH 55=TD.CA SOH 54=1 SOH 38=31600 SOH 40=2
44=93.77 SOH 60=20140301-20:42:37.426 SOH 59=0 SOH 100=* SOH 167=CS SOH 10=128 SOH

A buy limit order attempting to purchase 31,600 shares of the security TD Canada Trust at or below $93.77

- We use FIX for message delivery, making it easy to integrate the simulator into other systems.

Matching Engine

Accepts orders and maintains a number of order books.

- Buy orders with the highest price are placed at the top.
- Sell orders with the lowest price are placed at the top.

Bid Depth	Bid Price	Ask Price	Ask Depth
2000	$50.00	$50.01	11000
10000	$49.99	$50.02	10000
13000	$49.98	$50.03	14000
15000	$49.97	$50.04	9000

An order book

Simulated Trading Agents

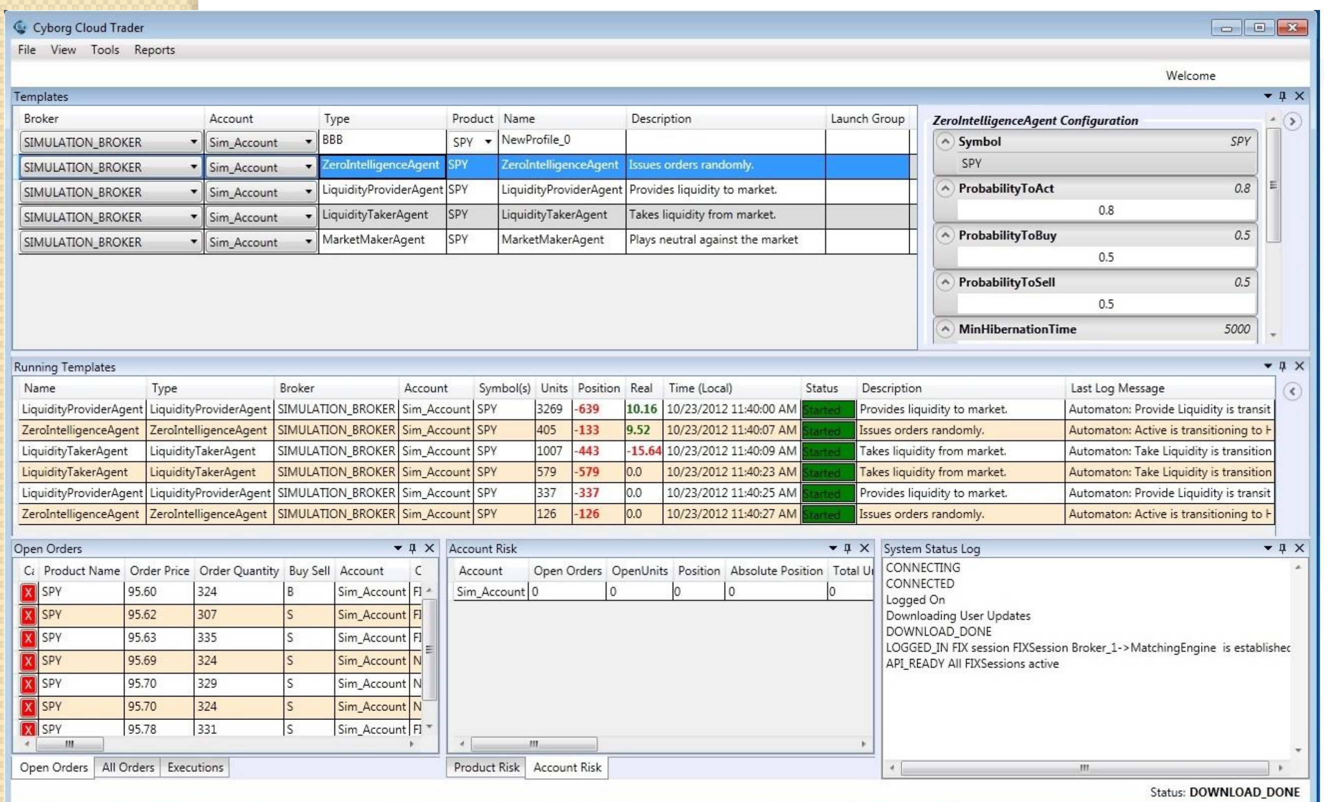
- We used Agent-Based Modelling to simulate a market.
- Each agent represents a human or robotic trader.
- Agents compete with each other, which determines the price of securities and consequently forms a market.
- Trading strategies can be arguably more fully evaluated by this approach than in the back-testing model.
- We have developed five pre-defined types of agents:
  - *Market Maker Agent*
  - *Liquidity Taker Agent*
  - *Liquidity Provider Agent*
  - *Random Agent*
  - *Swift Agent*
  - *Other Custom Agents*

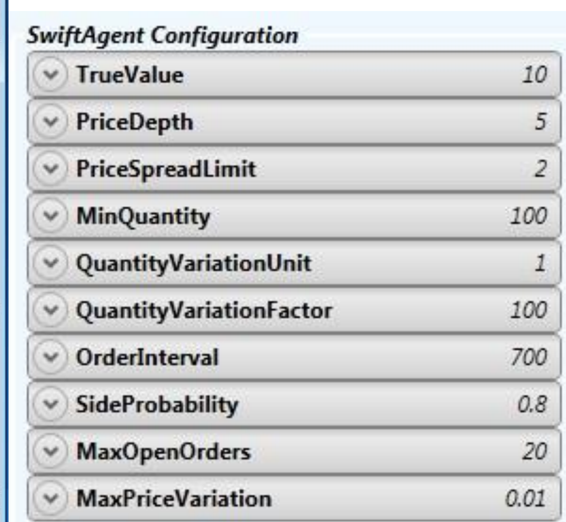
Software Implementation

- The simulator supports a variety of security types, including *equities, futures, foreign exchange* and *options*.
- It may run a market consisting
  - exclusively of *simulated trading agents* (human or robotic) using different trading strategies (*e.g.*, to study algorithms or market effects),
  - or *external participants* (typically human) may log in and interact with the simulated market (*e.g.*, for training).
- Participants can interact with the simulation environment as if they were trading in a real market.
- FIX is adopted to ease integration into other systems.
- The simulator can run in two different settings:
  - Agent-Based Simulation
  - Live-Data Simulation

Agent-Based Simulation

- A number of agents can be launched to simulate a market.
- Agents can perform a variety of actions at the same time.
- The simulator maintains the correct ordering of agents' requests.
- Each agent provides a succinct UI for easy configuration.





Live-Data Simulation

- Orders are matched against the current quoted prices in real market.
- It offers a risk-free environment to test strategies in a realistic setting.

This simulator outperforms others in several different aspects:

- it supports more security types
- provides data with finer granularity

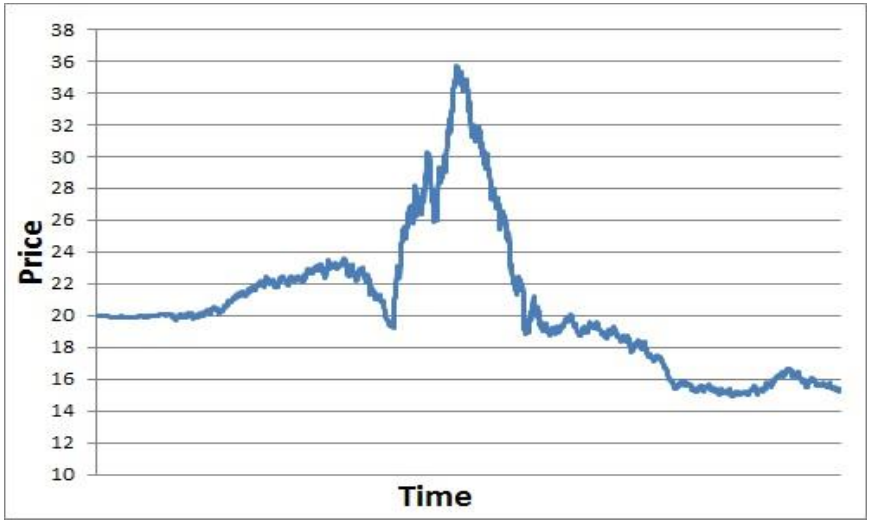
  (tick data vs 3-second market snapshot of Penn Exchange Simulator)
- adopts FIX protocol to ease integration into other systems
- does not require programming skills in specific languages

Useful Scenarios

Both *agent-based* and *live-market-data* simulator have been adopted by *Quantica Inc*, a company located in Kitchener, Canada.

We have found, informally, it to be useful in several scenarios.

- Strategy testing
- Education and training
- Demo
- Evaluating regulatory effects



a "bubble"

Conclusion

We have explored the analysis and simulation of financial markets.

Our analysis has explored short-term price changes of securities using pattern recognition techniques.

We proposed a method which was determined to perform well, even with the simple indicators.

We have also presented a simulator along with several types of simulated trading agents which represent a subset of traders observed in real markets.

We have found that in a corporate setting that our simulator is useful in a number of scenarios.