Teaching Financial Data Mining using Stocks and Futures Contracts

Gary D. BOETTICHER
Department of Computer Science, University of Houston – Clear Lake
Houston, Texas 77058, USA

ABSTRACT

Financial data mining models is considered to be "the hardest way to make easy money." Data miners are certainly motivated by the prospect of discovering a financial "Holy Grail." However, designing and implementing a successful model poses many intellectual challenges. These include securing and cleaning data; acquiring a sufficient amount of financial domain knowledge; bounding the complexity of the problem; and properly validating results. Teaching financial data mining is especially difficult due to the student's limited financial domain knowledge and the relatively short period (one semester) for building financial models. This paper describes an application of a financial data mining term project based on Stock and E-Mini futures contracts and discusses "lessons learned" from assigning similar term projects over six different semesters. Results of each case study results are presented and discussed.

Keywords: Data Mining, Financial Data Mining, Time-Series, EMini, Futures, Stock Market, Machine Learning.

1. INTRODUCTION

During the last fifteen years there has been an explosion of interest in the mining of time-series databases. Numerous research-based applications abound in various domains including weather [3, 12], medicine [8, 13], and Physics [9].

Among all the possible domains available for mining, financial data mining is probably one of the most popular for several reasons. Participants are motivated to learn how to get their money to work for them. In this context, data mining success is easy to measure and easy to understand. There is an abundance of domain specific knowledge in the form of technical indicators. Extensive amounts of data are available in terms of duration (price movements go back several decades); granularity (minute-by-minute price movements); financial alternatives (tens of thousands of stocks, options, or futures contracts). Finally, there are many financial markets available. Thus, financial data mining skills may be applied at a global level.

Mining financial data is a very intellectually challenging problem. It requires a combination of technical and domain knowledge; an understanding of how the financial markets work and are manipulated; the ability to formulate many models; a method of statistically and financially validating the model; and the capability to intelligently assess the results. Building effective models and successfully applying them in "real world" situations may take years of development.

Presenting "real world" financial domain knowledge and corresponding processes in an academic context is especially difficult. Computer Science, Computer Information Systems, and/or Statistical students typically lack the financial domain knowledge. Thus, compressing a vast amount of domain knowledge into one semester is a rather daunting task. Furthermore, there is an additional assumption that the professor possesses adequate financial, technical, and mathematical domain knowledge. Attaining proficiency in these domains may take weeks/months of research.

Despite these difficulties, there are several reasons for offering students the opportunity to mine financial time-series databases. Financial domain knowledge is easier to grasp than other domains. Many financial markets exist throughout the world. Therefore, financial markets are very visible at a global level.

Another motivating factor is that knowledge and skills attained from a financial data mining project may be applied towards personal portfolios. Thus, students may continuously apply "lessons learned" for the remainder of their lives.

This paper describes a financial data mining project, called the **GDB Cup**, that was implemented six times over six different semesters. Inspired by the KDD Cup, which is a data mining competition held in conjunction with the ACM SIGKDD Conference, the GDB Cup (which is the author's initials) provides students with financial data, corresponding domain-based documents, and a tool for validating results. The task is to mine the financial time-series data and produce financially successful models.

This work expands upon [2] in several ways. The original papers presented case studies based on daily stock data (2 semesters) and intra-day futures contracts (1 semester). This paper presents 6 cases studies. The first two case studies are based on stock data. The last four cases are based on intra-day futures contracts. Examining results over 6 different semesters allows for a more extensive analysis of "lessons learned."

The paper is organized as follows: section two discusses related research in the area of financial data mining. Sections three and four describe the financial data along with data cleansing issues. Sections five and six introduce financial domain concepts (trading and technical analysis). Section seven describes The Trade Simulator, which is a tool for validating and assessing financial models. Section eight presents six case studies. Section nine offers a discussion regarding the process and the results. Finally, sections ten and eleven provide a conclusion and future directions.

2. RELATED RESEARCH

There has been a great deal of research in the area of mining financial time-series. Some of the more successful and informative papers are described below.

Dempster[6] uses a Genetic Program to implement a real-time trading system and applies it to the foreign exchange (FOREX) market. Their approach achieves modest profits.

Frick [7] combines Point-And-Figure (PNF) charting and Genetic Algorithms. Given a series of price movements, a

corresponding PNF chart is created. The Genetic Algorithm generates a set of rules to apply to the PNF chart and the results are assessed.

Mizuno et al. [10] build a neural network model based on the Tokyo Stock Exchange Prices Index (TOPIX) which spans a 5-year period (1982-1987). The inputs consist of 4 technical indicators and three outputs (Buy/Hold/Sell). They use an "Equalized Learning Method." for duplicating vectors that contain Buy/Sell output values as a way of redistributing instance counts. They tested their models against the TOPIX from 1986 to 1987. Their best model produces an annual return on Investment (ROI) of 20%. However, the buy-and-hold strategy during this period produced an annual ROI of 21%.

Chenoweth et al. [5] use a Directional Index (Directional Movement divided by True Range). The inputs consist of S&P500; S&P500, lagged 1 day; S&P500, lagged 2 days; U.S. Treasury Rate, lagged 2 months; U.S. Treasury Rate, lagged 3 months; and 30 Year Government bonds. They train on data spanning 1982 through 1988 and test on data from 1989 through 1993. Their best model achieves an annual rate of return of 16.39 percent.

3. FINANCIAL DATA

Financial data assumes many forms including stocks, options, mutual funds, commodities, or futures contracts. Choosing which type data to data mine is a matter of preference. A property of each financial instrument is that it requires a unique validation method. This idea is discussed later in the paper. For simplicity, only one type of data is mined per case (semester).

There are several methods available for populating a financial data repository at a nominal cost. One option is to purchase financial data from a commercial data vendor [1, 11]. Another option is to buy/lease a commercial trading/investment software product. Normally these vendors supply their data in a proprietary format. However, there is usually an export feature to save data in a CSV or TXT format.

The least expensive option is to write a "screen-scraping" program that captures data off of financial web-sites.

Irrespective of the source of data, there are common data design issues including:

- **Time frame granularity (1-minute, 5-minute, etc.).** Depending upon the data source, it may be possible to analyze data on a minute-by-minute basis through a year-by-year basis. Depending upon the creativity of the data miner, he/she may aggregate the data and assess it over multiple time frames.
- **Time series sample size.** It is possible to build a model using a month's worth of data (e.g. 20 end-of-day samples). A minimal of 200 samples is recommended. This allows for the application of certain mathematical algorithms (e.g. moving average) which might need *n* number of samples in order to actually assess the data.
- **Financial symbols distribution.** Considering there are tens of thousands of financial symbols available, it is not feasible to analyze all these symbols due computational and memory overhead. When subsampling the data, it is desirable to have a set of symbols with both positive and negative covariance amongst the symbols.
- **Start/End bounds.** One option when analyzing a time-series is to adopt a "buy-and-hold" strategy. By selecting start and end dates so that start and end values are equal neutralizes the effect of a "buy-and-hold" strategy. For instance, the S&P EMini contract opened at 998.75 on 6/14/02 and closed at 999 on 6/12/03.
- **Market coverage.** A market may behave in one of three ways: go up (bull market), go down (bear market), or go sideways. It is important that the data span all three types of markets. This insures the robustness of the financial model.

The first two cases consist of daily stock data primarily from the S&P market. The remaining four cases are based on Futures contracts using a 5-minute time period. Futures contracts offer tremendous leverage on an investor's equity. For example, an investor could make 25 percent on their equity in one day. The risks are also high for this type of investing.

A typical tuple consists of 5 attributes *Open*, *High*, *Low*, *Close*, and *Volume*. The *Open* is the price of a stock/index at the beginning of a time period (beginning of the day or 5-minute interval). The *High* is the highest price attained by the stock/index during a specific time period. The *Low* is the lowest price attained by the stock/index during a specific time period. The *Close* is the last price the stock/index traded at during a specific time period. Finally, the *Volume* represents the number of shares/contracts that are traded during a specific time period.

4. DATA CLEANSING

The CrispDM[4] lists data preparation, in particular, data cleansing as one of the steps in the data mining process. This step is an important step within the whole data mining process; otherwise resulting models are susceptible to serious flaws.

By its nature, financial data contains "missing values" (e.g. no financial data on holidays) that must be considered. Additional data anomalies are injected into the data set in order to provide students with data cleansing opportunities. This includes dropping some instances from the time-series or setting certain values (e.g. the Low price) to zero.

5. INTRODUCTION TO TRADING

This section presents a cursory view of possible trading strategies in the stock market. A comprehensive examination is beyond the scope of this paper.

There are primarily two trading strategies, going long or selling short. **Going long** means buying a stock/contract at a particular price with the intent that the stock/contract will increase in price. Assuming this occurs, the stock/contract is sold for a profit. When **selling short** a stock/contract is initially sold with the intent that the stock/contract will decrease in price. Assuming this occurs, the stock/contract is bought in order to close the position. For example, a stock sold for $10, then later bought at $8 would result in a $2 profit. Thus, it is possible to make money in either market direction.

When an investor initially buys/sells a stock/contract the price may move in the opposite direction resulting in a losing position. An investor must decide how large a loss he/she is willing to tolerate or whether to risk the situation will turn in their favor. The amount of tolerance for loss (or risk) is called **drawdown**. Normally, this may be a percentage (e.g. 10 percent of the original price) for a stock purchase. For a futures contract, it may be a fixed number of points.

Determining the profit for a stock trade (assuming it is a long position) is equal to the purchase price *P* minus the selling price *S* times the number of shares *N* purchased.

$$Profit(Long\ Trade) = (P - S) * N \quad (1)$$

Futures contracts operate as follows: For a long position an investor may buy one EMini S&P contract (others are available) for $2,000. For each point the S&P goes up, the price of the contract increases by 50 dollars. If a contract is bought at $950, sold at $960, then the investor nets 10 points times $50 for a profit of 500 dollars on an initial investment of $2,000, or 25% increase in just one transaction. This illustrates the tremendous leverage in futures trading. Future contracts may also be shorted.

6. INTRODUCTION TO TECHNICAL ANALYSIS

There are many ways for deciding when to buy/sell a stock/contract. In this context, the process of financial data mining synthesizes technical analysis with machine learning for constructing successful models.

A **model** is defined as a series of trades. Each trade identifies a buy and sell date and time, the stock/contract price upon entering/exiting a trade, and the number of shares or contracts.

Technical analysis involves constructing one or more mathematical models based upon the stock/contract movement or change in *Volume*. One of the keys to effective data mining is acquisition of specific domain knowledge. Possessing domain knowledge speeds up the data mining process because it allows the modeler to discriminate between a multitude of strategies that may be available. Furthermore, possessing domain knowledge helps in recognizing a "good" model. In this case, the **technical indicators** serve as the domain-specific knowledge.

A technical indicator is an algorithm constructed using price or volume parameters. There are more than 100 technical indicators available. Common examples include *moving averages*, *relative strength index*, or *commodity channel index*. One of the challenges in the model formulation process is to focus on only a relatively few technical indicators which appear to be most promising.

Models may be constructed solely with technical indicators. However, students are encouraged to synthesize technical indicators with various machine learners.

7. MODEL VALIDATION: THE TRADE SIMULATOR

Building and assessing financial time-series models is a complex process. Once the data is cleaned, one or models are formulated which contain a series of long/short transactions. These transactions serve as the basis for assessing the models.

A tool is provided to students, called the *Trade Simulator* which automatically assesses the transactions. There are several reasons for providing this tool.

- **It eliminates portfolio management skills.** Students that participate in the GDB Cup normally have a strong Computer Science and/or Statistical background, but a relatively weak economic background. Providing this tool allows students to focus on the time-series modeling issues
- **Consistency in model assessment.** Providing the Trade Simulator guarantees that if one model performs better than a second model, it is due to better time-series analysis, not better portfolio management.
- **Realistic assessment.** It is easy to build a successful theoretical financial model. However, these models do not always fare well when applied to a "real world" situation. Usually any failures are due to a misunderstanding of risk. To mitigate risk, the Trade Simulator diversifies risk through asset allocation (when appropriate), how much risk is tolerated (e.g. acceptable drawdown). Also, the Trade Simulator limits the number of shares/contracts that may be purchased/sold. This prevents the model from becoming either the supply or the demand for a thinly trading stock.

The Trade Simulator validates a financial model both statistically and financially. Providing both methods of validation insures the credibility of a model.

Once a set of transactions are loaded into the Trade Simulator, it statistically analyzes the transactions determining: number of winning trades; number of losing trades; number of neutral trades for both long/short directions; length of each trade (minutes or days); maximum profit of a winning trade; maximum loss of a losing trade; average drawdown; average profit/loss; probability of winning (losing) trade; and Reward/Risk ratio. Figure 1 illustrates a sample screenshot from the statistical analysis.

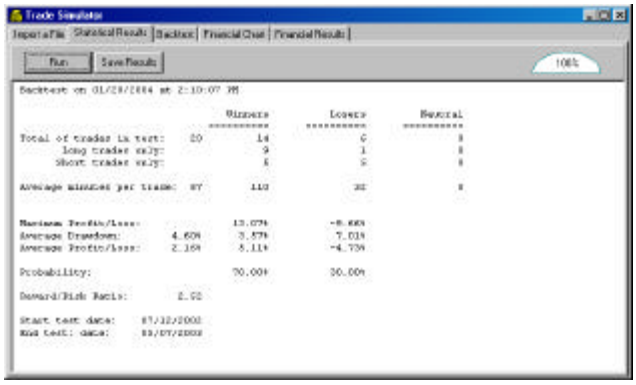

Figure 1: Statistical Analysis from the Trade Simulator

Financial mode that produce very good statistical results (e.g. all winning trades) do not necessarily produce very good financial results. One reason is that the statistical analysis ignores drawdown. Therefore it is imperative to conduct a financial analysis of a model. Usually, this is referred to as "backtesting" a model. The Trade Simulator assumes that a model starts with $100,000 in equity. It proceeds to analyze each trade and buy or sell according to a set of prescribed rules. Rule settings include: base equity ($100,000), minimum and maximum number of shares/contracts that may be purchased, stop limit, commission costs, whether the trader may use margin, and when the buy/sell actually occurs. Figure 2 shows a screenshot from the financial analysis of the Trade Simulator.

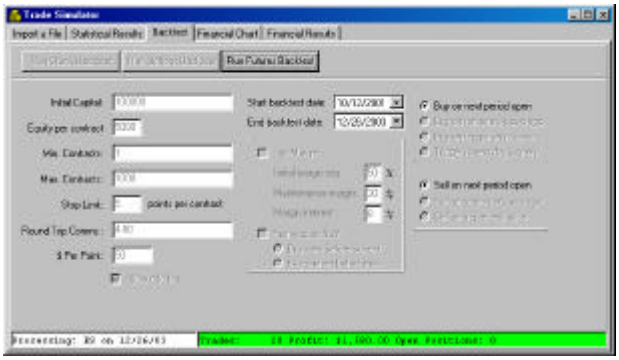

Figure 2: Financial Analysis from the Trade Simulator

As transactions occur, the bottom portion of Figure 2 shows the change in the equity status. After backtesting a financial model the net profit/loss is displayed.

To help the student visually assess the results of a backtest, the Trade Simulator provides an equity chart (see Figure 3). This enables a student to assess the volatility of their model. Ideally, a model produces steady growth with relatively few dips.

Details of all the transactions are also available (Figure 4). This is beneficial in understanding differences between statistical validation and financial validation. For example, there may be several "missed trades." This occurs when a financial model is fully invested; therefore it must pass on one or more transactions. A second common difference is when a model enters into a trade and is "stopped out." This means that a transaction encountered a drawdown limit and had to close the position prematurely.

Figure 3: Equity Analysis Graph from the Trade Simulator

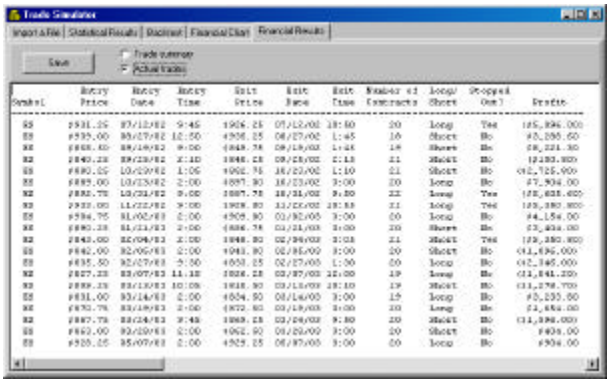

Figure 4: Transaction Details from the Trade Simulator

Finally, a transaction summary is provided.

8. CASE STUDIES

Since the GDB Cup is administered in a course taken by Computer Science, Computer Information Systems, and/or Statistics graduate students, it is assumed that students have little, if any knowledge, regarding the financial markets. Thus, it is necessary to explain many domain-specific terms including: up market, down market, sideways market, Open, High, Low, Close, Volume, Buying Long, Selling Short, Drawdown, and Margin Calls. Students are directed to various tutorials on the Internet where they may acquire additional financial domain knowledge. Furthermore, it is strongly emphasized the trading is a serious and very-risky endeavor.

Besides using technical indicators, it is possible to build models using accounting information (Earnings Per Share, P/E ratio, etc.), or global/financial news. Accounting information was not considered in order to avoid additional modeling complexity. News information (e.g. Reuters) is very difficult to correlate with price movement, therefore it is not considered.

Students have the option of either working individually or in groups for this project. For the first four cases, students could program in any programming language. However, project integration issues led to the requirement that all projects be written in c# for the last two cases. Students that form a group must complete a peer-feedback form at the end of the semester. Each student (or group) needs to decide upon a name for their respective group. Using group names protects a student's privacy when posting results on the Web.

Several progress reports are due throughout the semester. The first report is a "data cleansing" report which identifies data anomalies. The second report, due about two-thirds into the semester, requires students to produce at least one working model. Requiring interim reports prevents procrastination on this assignment. As an incentive, standings are posted on the course website.

The GDB Cup project has been implemented six times. Below is a description of each implementation.

Case 1: Fall, 2002

The financial data for this case study consists of daily stock data (*Open*, *High*, *Low*, *Close*, and *Volume*) from 12/31/1999 through 5/31/2002 for 452 stocks. These stocks were extracted from the S&P 500. Any stock that sold for more than 5 dollars and had a volume exceeding 100,000 was included in the list.

Each student group was encouraged to produce many models and explore the combination of two or more models. At the end of the semester, their models were run through the Trade Simulator. Table 1 shows the results for each group. All simulations start with $100,000 in equity. For this case, and all that follow, the results show the best model produced by each team.

Table 1: Results from the GDB Cup from fall, 2002

NAME	Amount
Mono-Poly	$1,036,137
Compass	$851,283
Saturday	$454,649
Trend Traders	$342,496
Midas Touch Trio	$187,336
Star	$172,635
Money Collector	$165,000
Supersonic Sys	$100,000

Four groups produced models that were all totally valid. Three groups generated models that where some were valid. And one group did not produce any valid models.

A group consisting of one person called *Mono-Poly* produced the best valid model. He leveraged $100,000 into $1,036,137 over a two and a half-year period. This equates to an annual ROI of 270 percent.

Case 2: Spring, 2003

The financial data for the next case also consists of daily stock data (*Open*, *High*, *Low*, *Close*, and *Volume*) ranging from 12/31/1999 through 5/31/2002. However, the stock pool is expanded to include 712 symbols. This allows each group to

split the symbols into two independent groups in order to train and test their models. The stock symbols included the entire S&P 500 plus many other recognizable symbols. Every symbol had to satisfy the criteria where the *Close* price is greater than 5 dollars and the *Volume* exceeds 100,000 shares per day. Considering that the second financial data set was a superset of data set used in the first case study, it was anticipated that the models in this case study would produce better results.

As before, each group was encouraged to produce many models and explore the combination of two or more models. At the end of the semester, their models were run through the Trade Simulator. Table 2 shows the results for each group. All simulations started with $100,000.

Table 2: Results from the GDB Cup from Spring, 2003

NAME	Amount
Stocks R Us	$1,405,760.17
Stock Miner	$1,074,124.63
Tom Dog	$605,763.09
Billionaires	$264,609.14
Affluent Buddies	$207,142.87
Cifey	$137,397.37
Creator	$135,706.63
Stochastinators	$146,908.68
Forecasters	$5,789.71

The best valid model, produced by *Stocks-R-Us*, leveraged $100,000 into $1,405,760 over a two and a half-year period. This equates to an annual ROI of 310 percent.

Case 3: Fall, 2003

The data set from the second case study was unwieldy for students in terms of storage and validating. As a consequence, the remaining cases are based on Futures intra-day data, 5-minute intervals. Thus, one day usually contains 87 data records (8 AM to 3:10 PM Central Time) where each record consists of *Open*, *High*, *Low*, *Close*, and *Volume* attributes.

The data set for the third case uses the S&P EMini Futures (symbol *ES*) which generates 50 dollars profit/loss per point movement. The EMini data spans from 6/14/02 through 6/12/03 (about 22,300 samples). During this time frame the price of the Emini started at 998.75 and ended at 999. Thus, a "buy and hold strategy" would be rendered neutral.

Due to the high leverage capability in trading Futures contracts, tight stop restrictions are imposed in the form of a 5-point drawdown limit upon the financial simulations. This means that the trade simulator exits a trade if it loses 5 or more points. Thus, one of challenges for was to incorporate risk assessment within their models.

Table 3: Results from the GDB Cup from Fall, 2003

NAME	Training Data
Millionaire Club	$852,453
Rainbow	$783,681
Money Tree	$624,417
The Money Maker	$499,154
The Tick	$239,402
Bankrupt By Halloween	$213,199
Wall	$125,655

The *Millionaire Club*, leveraged $100,000 into $852,453 over a one-year period. This equates to an annual ROI of 852 percent.

Case 4: Spring, 2004

The fourth case also uses the S&P EMini Futures symbol, ES. The sample set is expanded from 10/12/01 to 12/26/03 with about 49,000 samples. Once again, a 5-point drawdown restriction is imposed.

In the previous three cases, completed models were validated against the original data set. For this case, completed models are validated against an independent Futures data which ranges from 12/29/04 to 4/16/04 (about 6700 samples). Students never had access to this test set.

Table 4: Results from the GDB Cup from Spring, 2004

Team Name	Training Data	Test Data
Extreme Money Makers	$51,454,740	$270,588
TheStreet.Com	$35,484,449	$814,621
Mining Wizards	$3,643,309	$ 85,268
For Fortune	$1,088,176	$ 75,074
Precious Dreams	$1,085,189	$185,435
Money Miners	$976,923	$120,244
We'll Be Rich	$958,883	$ 99,154
Money Makers	$192,226	$100,000

Table 4 shows the results from this fourth case study. Two teams did extremely well in both the training and test phases. Three teams which did well against the training data, but lost money against the test data. One team, the *Money Makers*, generated zero trades against the test data. This argues for tracking the number of trades a financial model generates.

Case 5: Fall, 2004

The fifth case switches from S&P to NQ (Nasdaq 100) EMini Futures data. Rather than 50 dollars/point leverage it is 20 dollars/point. Since there is a decrease in leverage, the drawdown is extended from 5 points to 15 points. The main reason for the switch was to deter the "reuse" of student work from previous semesters. The dataset ranges from 3/25/02 to 6/9/04 with approximately 49,000 samples. A larger independent test set is used that starts on 9/7/00 and ends on 3/6/02 (about 33,000 samples). It is hoped that the longer duration would compensate for the reduced leverage.

Table 5: Results from the GDB Cup from Fall, 2004

Team Name	Training Data	Test Data
Money Making Machines	$4,817,298	$103,872
Smart Trader	$3,959,017	$3,009,372
Money Learners	$2,116,194	$1,846
The Burgeoning Data Miners	$955,620	$12,402
Prime Timers	$869,113	$1,949
Mining the Future	$527,459	$1,922
Delta	$100,000	$1,926
The Mathematician	$100,000	$1,822

The test data results from this case study are not very good. One group did well, one broke about even, the rest went broke. Two vital lessons are learned from this case study. It is important to stress the significance of validation against the training set. Also, this was very bearish market. Most groups

developed bullish (long) strategies. It is important to consider both market directions when building models.

Case 6: Spring, 2005

The last case study uses NQ EMini Futures training data starting with 9/7/00 to 6/9/04 for about 83,600 samples. The assignment emphasized model validation by extracting two independent time frames for training and testing their models.

The test data ranged from 6/10/04 to 4/26/05 for a total of about 19,600 samples. The market movement of the NQ during this period is primarily sideways. Table 6 shows the results from the training and test experiments.

Table 6: Results from the GDB Cup from Spring, 2005

Team Name	Training Data	Test Data
Winners	$428,468,209	$26,749,509
RSG	$370,177,814	$54,321,076
Fortune Hunters	$9,383,678	$65,708
L and S	$3,745,732	$89,040
Werewolf	$332,067	$177,761
KZW	$216,880	$39,692
The Learners	$1,988	$1,976
Sonics	$1,898	$12,861

Each team had to consider the financial success of their model along with the robustness. Thus, this case study stressed the importance of model consistency.

The first two teams *Winners* and *RSG* did extremely well. Their respective models were scrutinized for modeling flaws. So far, none have been found. *Fortune Hunters*, *L and S*, and *KZW* lost money against the test data set. However, these teams did not go broke (as in the previous case study). The *Werewolf* team created an interesting model. It proved to be very successful, but very slow. Thus, it was only tested against a small portion (one month) of the test data. Finally, the last two teams were unable to build any financial successful models.

9. DISCUSSION (LESSONS LEARNED)

The results from all six case studies are very impressive. The best models from all six case studies produce annual ROI in excess of 250 percent. All these best models exceed the ROI produced by [4, 8] by ten times.

Due to the time constraints (one semester), an initial hurdle students must address is how to select a few key technical indicators from a relatively large pool on which to build models. Some teams are successful in their selection, but other teams divided the coding of technical indicators into without combining their models. This approach does not exploit the potential synergy between technical indicators.

In assessing models statistical success does not equate to financial success. Some models generated a high percentage of winning trades, but ended up losing money. Winning trades may be marginal or may occur at the same time.

Some teams inadvertently build invalid models. For later cases (4 and higher), each team was required to validate their best model in EXCEL. This approach helps identify any flawed reasoning in their approach. A common mistake would be to forget to offset a moving average when comparing to the actual price.

Several approaches combined machine learners with technical indicators. It is observed that neural network and classifier models did not fare well. Genetic Program and Instance Based Learners (IBL) did well. However, IBL models trained very slowly.

Based upon these results, one may consider transitioning from paper trading to actual trading. This raises a whole new set of issues regarding character and emotional strength. Issues certainly beyond the scope of this paper.

Over successive semesters the number of invalid models declined dramatically. One reason for this improvement is the students were educated about peeking into the future and creating transactions based on future price activity.

10. CONCLUSIONS

The results demonstrate that it is possible to successfully mine financial time-series data within one semester (15 weeks). Collectively, 42 out of 48 teams, or 87.5% were able to produce financially successful training models and only about 3 models, or 6.25%, generated financially challenged models. For the test cases, only 37.5 were financially successful, but the average profit was $3,597,630 which suggests the margins of the winning test cases greatly exceeding the losing test cases.

11. FUTURE DIRECTIONS

Different financial investment instruments present their own unique set of issues. One future direction is to mine other financial instruments (e.g. Mutual funds or Options).

12. REFERENCES

[1] Ashkon Technology, 2005, Available at www.ashkon.com

[2] Boetticher, G., "The GDB Cup: Applying "Real World" Financial Data Mining in an Academic Setting," *The 8th World Multi-Conference on Systemics, Cybernetics and Informatics*, Orlando, Fla., 2004.

[3] Calvo, R.A., Navone, H.D., and H. A. Ceccatto, "Neural network analysis of time series: Applications to climatic data" In *Southern Hemisphere Paleo- And Neoclimates: Key Sites, Methods, Data and Models* (Ed. W. Volkheimer and P. Smolka, Springer Verlag, ISBN: 3540 665897, 2000).

[4] Chapman, Pete, Julian Clinton, Thomas Khabaza, Thomas Reinartz, and Rdiger Wirth. The CRISP-DM process model. Technical report, CRISP-DM consortium, March 1999.

[5] Chenoweth, T., Obradovic, Z. and Lee, S., "Embedding Technical Analysis into Neural Network Based Trading Systems." *Applied Artificial Intelligence*, vol 10, no. 6., 1996, Pp. 523-541.

[6] Dempster, M.A., C.M Jones, A Real-Time Adaptive Trading System using Genetic Programming, Quantitative Finance, Vol. 1, 2001, Pp. 397-413.

[7] Frick, A., Herrmann, R., Kreidler, M., and A. Narr, Genetic-Based Trading Rules - A New Tool to Beat the Market With -- First Empirical Results, Proceedings of 6th International AFIR Colloquium, Nürnberg, Oct. 1996, (Ed. P. Albrecht) Verlag Versicherungswirtschaft e.V. Karlsruhe, Volume I/II, Pp. 997–1018.

[8] Hunter, J. & McIntosh, N.. Knowledge-based event detection in complex time series data. *Artificial Intelligence in Medicine*. 1999, pp. 271-280.

[9] Judd, K, A. I. Mees. On selecting models for nonlinear time series, *Physica D*, 82:426-444, 1995.
[10] Mizuno, et al., Application of Neural Network To Technical Analysis of Stock Market Prediction, Studies in Informatics and Control (With Emphasis on Useful Applications of Advanced Technology), 7 2, June 1998.
[11] Nikolenko, Alexandre, www.ANFutures website. Information available at: http://www.anfutures.com/
[12] Sripada, S.G., E. Reiter, J. Hunter, J. Yu, *Segmenting Time Series for Weather Forecasting*. In Ann L. Macintosh, Richard Ellis, and Frans Coenen (Eds) Applications And Innovations in Intelligent Systems X, Springer, London, 2002, pages 193-206.
[13] Zöllei, L., Panych, L., Grimson, E., W.M. Wells III: "Exploratory Identification of Cardiac Noise in MRI Images," *MICCAI 2003*, Montreal, CANADA, LNCS 2878, Pp. 475-483.