Combining machine learning and effective feature selection for real-time stock trading in variable time-frames

A.K.M. Amanat Ullah*

Department of Computer Science,
University of British Columbia, Canada
Email: amanat.ndc@gmail.com
Email: amanat7@mail.ubc.ca
*Corresponding author

Fahim Imtiaz, Miftah Uddin Md Ihsan,
Md. Golam Rabiul Alam and Mahbub Majumdar

Department of Computer Science and Engineering,
BRAC University, Bangladesh
Email: fahim.imtiaz@g.bracu.ac.bd
Email: miftah.uddin.mohammad.ihsan@g.bracu.ac.bd
Email: rabiul.alam@bracu.ac.bd
Email: majumdar@bracu.ac.bd

Abstract: The unpredictability and volatility of the stock market render it challenging to make a substantial profit using any generalised scheme. Many previous studies tried different techniques to build a machine learning model, which can make a significant profit in the US stock market by performing live trading. However, very few studies have focused on the importance of finding the best features for a particular period for trading. Our top approach used the performance to narrow down the features from a total of 148 to about 30. Furthermore, the top 25 features were dynamically selected before each time training our machine learning model. It uses ensemble learning with four classifiers: Gaussian naive Bayes, decision tree, logistic regression with L1 regularisation and stochastic gradient descent, to decide whether to go long or short on a particular stock. Our best model performed daily trade between July 2011 and January 2019, generating 54.35% profit. Finally, our work showcased that mixtures of weighted classifiers perform better than any individual predictor about making trading decisions in the stock market.

Keywords: feature selection; feature extraction; stock trading; ensemble learning.

Reference to this paper should be made as follows: Ullah, A.K.M.A., Imtiaz, F., Ihsan, M.U.M., Alam, M.G.R. and Majumdar, M. (2023) 'Combining machine learning and effective feature selection for real-time stock trading in variable time-frames', *Int. J. Computational Science and Engineering*, Vol. 26, No. 1, pp.28–44.

Biographical notes: A.K.M. Amanat Ullah is pursuing his PhD in Computer Science at the University of British Columbia. He completed in MS and BS from BRAC University.

Fahim Imtiaz graduated BSc in Computer Science, BRAC University, Bangladesh. He is currently working as a Writer at the Daily Star, Bangladesh.

Miftah Uddin Md Ihsan graduated BSc in Computer Science and Engineering, BRAC University. Bangladesh. He is currently working as a Software Engineer in Google, Warsaw, Poland.

Md. Golam Rabiul Alam is an Assistant Professor of Computer Science and Engineering Department of BRAC University. He received his PhD in Computer Engineering from Kyung Hee University, South Korea. He received his BS and MS degrees in Computer Science and Engineering, and Information Technology from Khulna University and University of Dhaka respectively.

Mahbub Majumdar received his PhD in Mathematics University of Cambridge, MS from Stanford University and BS from Institute of Massachusetts Technology. He is currently the Dean and a Professor, School of Data and Sciences, BRAC University, Bangladesh. Computational finance is one of his research interests.

1 Introduction

1.1 Background

Stocks are effectively little parts of a company's ownership, and the stock market functions similarly to an auction, with investors buying and selling stocks. When a shareholder acquires stock, he or she owns a percentage of the firm equal to the number of shares purchased compared to the total number of outstanding shares. For example, if a corporation has 1 million shares and an individual owns 50,000 of them, the individual owns 5% of the company.

1.2 Long-short investment strategy

According to Jacobs and Levy (1993), classic stock investing focuses on finding equities to purchase long that are expected to rise in value. There was little consideration, if any, given to profiting from short-selling expensive equities. When investors started to combine long and short strategies in their investment portfolio, they discovered new advantages and possibilities that were previously inaccessible.

Buying long simply means purchasing a stock that you believe will gain in value and then selling for a profit when the price rises. Assume you purchased 500 shares of a certain company at a price of $10 per share. This is a total of $5,000. The price of an ABC share climbs to $55 after a week. You make $500 when you sell the shares.

Shorting is when you borrow stocks from a broker at a profit and sell them while waiting for the price to decrease. Once the price has dropped significantly, you repay the lender by purchasing the same number of stocks at the reduced price that you borrowed in the first place. The difference in price minus interest and commissions is your profit.

For example, suppose you borrow 100 XYZ shares for $50 each and sell them for $5,000 while waiting for the share price to fall. You acquire 100 shares of XYZ for $4,500 after the price per share of XYZ has plummeted to $45. Return the 100 shares to the lender, and the profit is the difference between the interest and commissions. Your profit in this situation is $500.

1.3 Motivation

"I will tell you how to become rich. Close the doors. Be fearful when others are greedy. Be greedy when others are fearful" (Warren Buffett). The quote suggests that trading decisions need to be made entirely based on logic and not based on human emotions. Oftentimes people cannot control their emotions It is difficult to let out emotion while trading. Effective trading involves making decisions without letting emotions get in the way. The perfect way to solve this problem is to deploy a machine that solely relies on logic to make effective decisions. On another note, current estimates show that automated trading accounts for 50–70% of equities trades in the USA, 40% in Canada, and 35% in London (O'Reilly, 2012; Grant, 2011). Therefore there will come a time where all the trades will be managed by machines. To prepare the world for such a time more research into this field is vital. We are all aware of the unpredictability of the stock market, and how difficult it is to predict because of the noise in the data based on the work by Bloembergen et al. (2015). Some people believe that it is not possible to do so. We believe that with the advancements in machine learning algorithms and artificial intelligence, we can predict stock market trends sufficiently, given we provide sufficient, refined, data to our models. Many previous researchers (Yuan et al., 2020; Tsai and Hsiao, 2010; He et al., 2013) worked with selecting features with different algorithms for maximising profit. However, most of them did not run the final test on an actual stock trading setting. Additionally, none of the research worked on which feature time-frame works best for how many days of trading. Our research aims to explore this research gap.

1.4 Contributions

To our knowledge, our model is the first to recommend which feature time-frame suitable for how many days of trading. To address this problem our novel approach calculated each of the features using different time-variants (default, 1 day, 2 days, 5 days, 22 days) to find out which variant works best for daily trading, weekly trading and monthly trading. Using the recommended features our model further used dynamic feature selection techniques coupled with advanced machine learning algorithms to generate profit on real-time stock data from 1,500 stocks from the US stock market from 2011 to 2019 and generated significant profit which is on par and in some cases better than the state-of-the-art models. The main contributions of this research are as follows:

- a dynamic feature selection mechanism has been proposed to select discriminative features over multiple time-frames for holding long and short positions for effective stock trading
- after the initial feature selection mechanism, the proposed model uses ANOVA for finalising the set of features and uses ensemble of various machine learning algorithms for stock trading that generated 54.35% profit on the initial investment.

1.5 Paper organisation

In the next section of our paper, we review the literature of the previous work on machine learning models in order to predict trends in the stock market. We discuss our overall strategy in Section 3. The dataset analysis is discussed in Section 4. Sections 5, 6, 7 and 8 are on the result analysis. Sections 9 and 10 concludes the paper and talks about the limitations in our thesis and future prospects.

2 Literature review

The prevalence of volatility in the stock market and also other markets [e.g., forex (Tiong et al., 2016) and cryptocurrency (Chen et al., 2020)] makes predicting stock prices anything but simple. Before investing, investors perform two kinds of analysis (Patel et al., 2015). The first of these is fundamental analysis, where investors look into the value of stocks, the industry performance, economic factors, sentiment analysis (Chang, 2020), etc. and decide whether or not to invest. Technical analysis is the second, more advanced, analysis that involves evaluating those stocks through the use of statistics and activity in the current market, such as volume traded and previous price levels (Patel et al., 2015). Technical analysts use charts to recognise patterns and try to predict how a stock price will change. Malkiel and Fama's efficient market hypothesis states that predicting the values of stocks considering financial information is possible because the prices are informationally efficient (Malkiel and Fama, 1970). As many unpredictable variables influence stocks and the stock market in general, it seems logical that factors such as the public image of the company and the political scenario of a country will be reflected in the prices. By sufficiently preprocessing the data obtained from stock prices and the algorithms and their factors are appropriate, it may be possible to predict stock or stock price index.

There were quite a few different implementations of machine learning algorithms for the purposes of making stock market price predictions. Different papers experimented with different machine learning algorithms that they implemented in order to figure out which models produced the best results. Dai et al. (2012) attempted to narrow down the environment by selecting certain criteria. Under these criteria, they were able to achieve a profit of 0.0123, recall 30.05%, with an accuracy of 38.39%, and 55.07% precision, using a logistic regression model, after training the model for an hour. Zheng and Jin observed that when compared with logistic regression, Bayesian network, and a simple neural network, a support vector machine having radial kernel gave them the most satisfactory results (Zheng and Jin, 2017). Due to their limited processing power, they were only able to use a subset of their data for training their model and recommended that a more powerful processor be used to achieve better results. Similar recommendations were made by Chen et al., stating that their preferred model, the long short-term memory (LSTM) (Wu et al., 2021), would have performed better were they able to train the different layers and neurons using higher computing power (Chen et al., 2017). Since the data was non-linear in nature, a recurrent neural network (RNN) would be more suited to the task. Recent researches also showcase the use of transformer networks (Hu, 2021) and fuzzification (Hu, 2021) techniques in stock trading and prediction.

In Hegazy et al. (2014), it was discussed that when performing stock price prediction, it came out to be that ANN the algorithm that was once popular for prediction suffers from overfitting due to large numbers of parameters that it needs to fix (Tao et al., 2004). This is where support vector machine (SVM) came into play and,, it was suggested, that this method could be used as an alternative to avoid such limitations, where according to the VC theory by Vapnik (2013) SVM calculates globally obtained sol unlike the ones obtained through ANN which mostly tend to fall in the local minima. It was seen that using an SVM model the accuracy of the predicted output came out to be around 57% (Kim, 2003). There is one other form of SVM and that is least squared support vector machine (LS-SVM). In Madge and Bhatt (2015), it was mentioned that if the input parameters of LS-SVM is tuned and refined then the output of this classification algorithm boosts even further and shows promise to be a very powerful method to keep an eye out for. SVM being this powerful and popular as is it, is now almost always taken into consideration when it comes to predicting price of a volatile market, and thus we think that incorporating this into our research will boost our chances of getting a positive result.

The study by He et al. (2013) measured twelve technical indicators for further investigation using data from the Shanghai Stock Exchange Composite Index (SSECI) from 24 March 1997 to 23 August 2006. The stock market's input variables were chosen from a total of 12 indicators. SMA, EMA, Alexander's filter (ALF), relative strength, RSI, MFI, percent B indicator, volatility, volatility band, Chaikin oscillator (CHO), moving average convergence-divergence (MACD), percent K indicator, accumulation and distribution (AD) oscillator, and Williams percent R indicator are some of the indicators used. Then, principal component analysis (PCA), genetic algorithm, and sequential forward feature selection methods to select which features for optimal investment. However, the paper did not include any resulting analysis or graphical representations of the results.

Yuan et al. (2020) selected 60 features for their prediction. The data comes from the Chinese A-share market and dates from 1 January 2010 to 1 January 2018. The algorithms used for prediction were SVM, artificial neural networks (ANN) and random forest (RF). For the feature selection, the paper used recursive feature elimination (RFE) and random forest feature selection using the information gain values. The RF for feature selection and RF model for prediction has the greatest annualised return when it picks the top 1% of companies, with a 29.51% annualised return. The RF-RF model's profitability is further investigated using the stratified back-testing technique, and the new long-short portfolio's annualised return from 2011 to 2018 is 21.92%, with a maximum drawdown of just 13.58%. This profit is not substantial for proving the success of their model because better results can be achieved.

To decrease the cost of training time and increase prediction accuracies, the work of Huang and Tsai (2009) combined the support vector regressor (SVR) with the self-organising feature map (SOFM) method and a filter-based feature selection. Thirteen technical indicators were used as input variables to forecast the daily price in the Taiwan index futures (FITX) in order to forecast the

price index for the next day. The SOFM-SVR with feature selection had a mean absolute percentage error (MAPE) of 1.7726%, which is higher than the single SVR with feature selection and the one without feature selection. However, they did not test their strategy in the real stock market which would further evaluate their model's actual performance.

Barak et al. (2015) proposed a hybrid feature selection method using adaptive neural fuzzy inference system (ANFIS) and the imperialist competitive algorithm (ICA) is used to choose the most suitable features. The trading signals generated by the model achieved superior outcomes with 87% prediction accuracy, and the wrapper features selection achieves a 12% increase in predictive performance over the basic research. Furthermore, since wrapper-based feature selection models are much more time-consuming, the results of our wrapper ANFIS-ICA method are better in terms of reducing time and improving prediction accuracy when compared to other algorithms like the wrapper genetic algorithm (GA). However, they worked on only 24 features at max and did not test implement a long-short strategy.

The research by Nti et al. (2019) used RF with an improved leave-one-out cross-validation strategy and a LSTM network to evaluate the degree of importance between various sectors stock-price and MVs and forecasted a 30-day had stock-price. From January 2002 to December 2018, the research dataset was acquired from the GSE official website, and the 42 macroeconomic indicators dataset was collected from the Bank of Ghana (BoG) official website. The LSTM model performed better than the baseline ARIMA model. But real-time trading was not done in this study.

The paper by Gandhmal and Kumar (2021) used 12 technical features. The features are selected using decision tree algorithm based on wrapper feature selection. The paper uses the chronological penguin Levenberg-Marquardt-based nonlinear autoregressive network (CPLM-based NARX) for prediction. The suggested paper showed that CPLM-based NARX outperformed the competition in terms of MAPE and RMSE, with values of 0.96 and 0.805, respectively in comparison with the regression model, deep belief network (DBN), and neuro fuzzy-neural network. This study does not analyse between the different timeframes of each technical feature and only uses 12 technical features.

PCA, genetic algorithms (GA), and decision trees (CART) are all compared in the research article by Tsai and Hsiao (2010). It examines their prediction accuracy and mistakes by combining them using union, intersection, and multi intersection methods. The findings of the experiments indicate that integrating several feature selection techniques may improve prediction performance over single feature selection methods. The intersection of PCA and GA, as well as the multi-intersection of PCA, GA, and CART, perform the best, with accuracy rates of 79% and 78.98%, respectively.

The causal feature selection (CFS) method is proposed in the research by Zhang et al. (2014), to choose more representative features for improved stock prediction modelling. Comparative tests were performed between CFS and three well-known feature selection methods, namely PCA, decision trees (DT; CART), and the least absolute shrinkage and selection operator, using 13-year data from the Shanghai Stock Exchanges (LASSO). When coupled with each of the seven baseline models, CFS performs best in terms of accuracy and precision in most instances and finds 18 key consistent characteristics out the 50 initial input features given.

Ensemble methods such as random forests help to reduce the probability of the data overfitting. RFs use decision trees and majority voting to obtain reliable results. In order to perform an analysis on stock returns, Lin et al. tested a prediction model that used the classifier ensemble method Tsai et al. (2011) and took bagging and majority voting methods into consideration. It was found that models using single classifiers under-performed compared to the ones using multiple classifiers, in regards to ROI and accuracy when the performances of those using an ensemble of several classifiers and those using single baseline classifiers were compared Patel et al. (2015). An SVM ensemble-based financial distress prediction (FDP) was a new method proposed by Sun and Li (2012). Both individual performance and diversity analysis were used in selecting the base classifiers from potential candidates for the SVM ensemble. The SVM ensemble produced superior results when compared to the individual SVM classifier. A sum of ten data mining techniques, some of which included KNN, naive Bayes using kernel estimation, linear discriminant analysis (LDA), LS-SVM, was used by Ou and Wang (2009) to try and forecast price fluctuations in the stock market of Hong Kong. The SVM and LS-SVM were shown to produce better predictions compared to the other models.

3 Real-time stock trading strategy

The diagram in the figure system depicts our model's whole process. Selecting stocks, feature selection, data pre-processing, training and creating predictions using machine learning algorithms, and making adjustments to the portfolio based on the forecasts are all part of the process.

The plan used the Quantopian algorithm feature, which included numerous machine learning algorithms. We began by utilising the Q1500US algorithm supplied by Quantopian to gather data on 1,500 of the market's most popular stocks. Following that, we imported all of the factors supplied, as well as TA-LIB factors (a significant financial factor provider). Some custom factors have to be implemented by us as well. There are 148 features in all, including asset growth 3M, asset to equity ratio, capex to cash flows, EBIT to assets, EBITDA yield, earnings quality, MACD signal line, mean reversion 1M, AD, ADX, APO, ATR, BETA, MFI, and others.

However, because of the market's volatility, just putting all of the inputs into the ML algorithm will not provide a particularly consistent result overall, since a specific component might have both a positive and negative

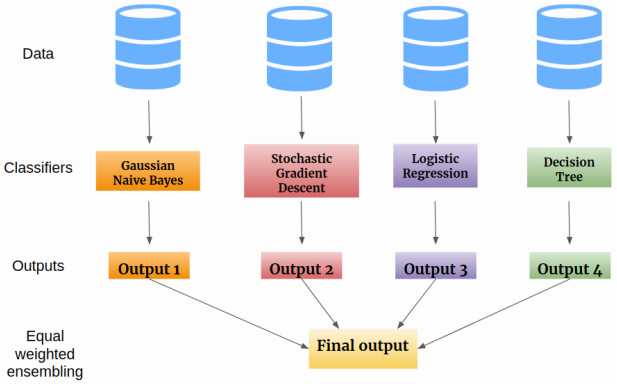
influence on the forecast at various times in the market. To solve this difficulty, we have to use dynamic feature reduction, which is covered in Section 4. We choose the top 25 features based on the F-value of ANOVA after the first feature selection.

We specified the number of stocks we wanted to trade, the machine learning window length, the N^{th} forward day we wanted to forecast, which in this instance was set to 5, and the trading frequency, which is the number of days after which we wanted to begin the trade, after collecting the characteristics. We sort and trade on two separate quantiles, higher 30% and lower 30%, from the 1,500 stock data that we imported before. We conduct this slicing to ensure that we are not trading on companies with a highly stable rate of change in their price, but rather on stocks that are put higher and lower down the ladder on which we might go long (30%) and short (30%) and have a substantial success rate. We put the top 30% to 1, indicating that we are long, the lower 30% to –1, indicating that we are short, and the remaining 40% to 0, indicating that we do not trade on them. When these higher and lower quantiles are added together, we get 500 stocks, which is the number we selected previously. We had to remove the label (returns) from the zipline and run it through a five-day calculation. Because there are no five days in forwarding time data at that specific time, the T – 5 days data had to be deleted, resulting in NAN labels, and was therefore removed from the zipline data frame. To provide the Label column to the ML algorithm, it has to be maintained separately. We had to sort everything outside of Quantopian since it does not enable machine learning or data preparation within the pipeline.

After the data has been preprocessed, we create a new column named ML in the pipeline and use the machine learning function to populate it for each and every stock for that day. The universe and all the columns of the pipeline, i.e., the factors and labels that we determined, are the parameters of the ML function. This is where we use the factor reduction technique we discussed before. This approach is repeated dynamically throughout the training process, i.e., we only train the algorithm with the top 25 features using the SelectKBest feature selection technique every time we train it.

Our ensemble learning model is shown in Figure 1. In this case, four machine learning algorithms each provide a hypothesis and an output. To obtain the final result, equal-weighted ensembling is applied to these four outputs. We ran into TLE when we attempted to combine three or more high complexity classifiers (SVM, AdaBoost, etc.) with dynamic feature selection while building ML algorithms. However, we chose algorithms with a very short runtime, typically under 2–3 seconds to test and train, which is critical in a live trading algorithm.

Figure 1 Structure of the machine learning model used (see online version for colours)



4 Proposed feature selection model

The paper analyses a total of 148 features based on four criteria.

- returns analysis
- information coefficient analysis
- turnover analysis
- grouped analysis.

For the analysis we used Alphalens on the stock data from start_date = '2011-03-06' and end_date = '2012-03-06'. The feature selection method is showed in Figure 2. The method went on long and short positions on the top and bottom quantile or reverse if the feature is negative. The trading is done for 1D (1-day hold period), 5D (5-day hold period) and 22D (22-day hold period). The features must have 'mean return' greater than 0.05% or 0.5 basis points for both the long and short positions selected to trade for that certain time period. The information coefficient must be greater than 0.005. In the turnover analysis, the mean turnover must be greater than 0.25. The stocks satisfying these criteria will be initially selected. In addition, Sklearn's SelectKbest method is used to select the best features out of the selected feature. For the hyper-parameter, 'f_classif' is selected, which ranks the features using the T-scores from ANOVA.

The stock values were divided into three equal quantiles. The lower quantile, the middle quantile and the upper quantile. Each quantile had about 33.33% of the values. Table 1 shows the selected features for daily, weekly and monthly trading.

Figure 2 Proposed feature selection model (see online version for colours)

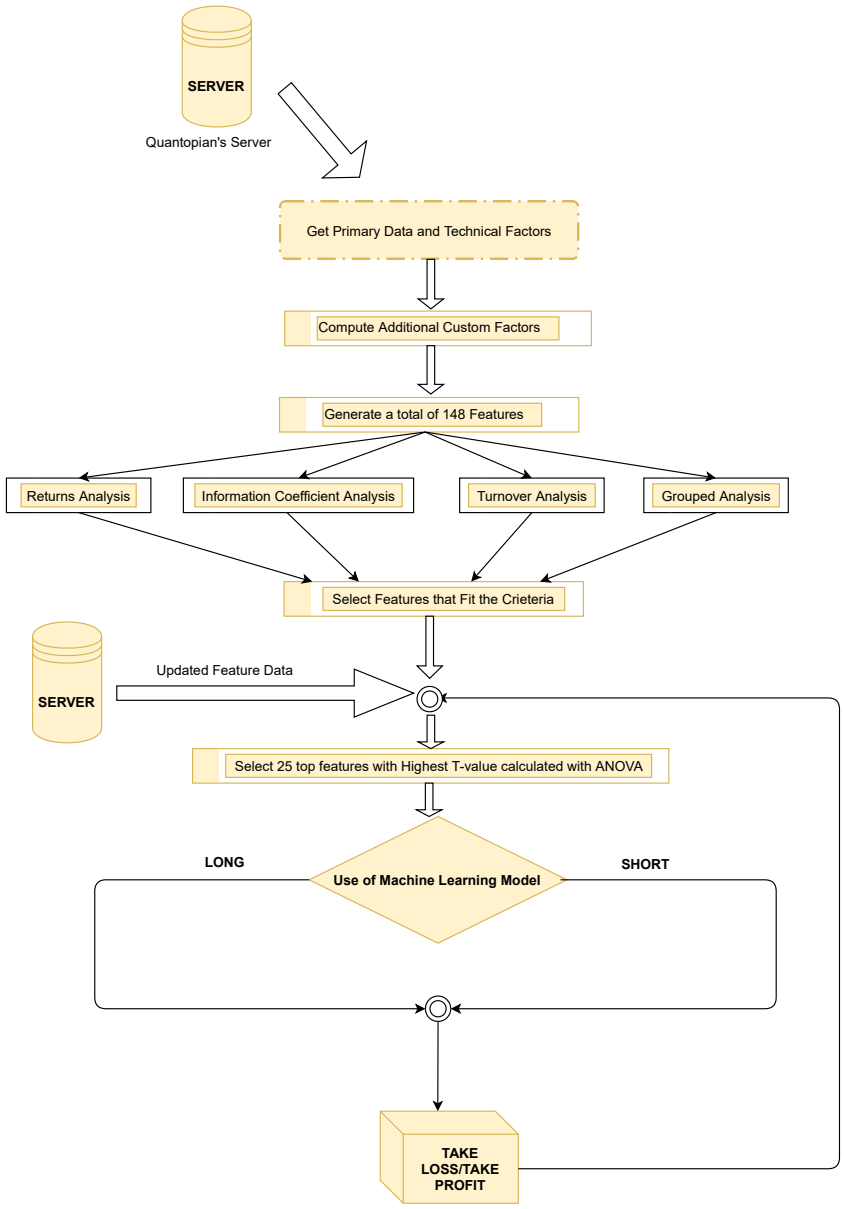


Table 1 Selected features for trading based on feature analysis for different timeframe

Feature name	*Trading position*			*Feature name*	*Trading position*		
	1D	*1W*	*1M*		*1D*	*1W*	*1M*
Asset_Growth_5D				MACD_Signal_10d	✓	✓	✓
Asset_Growth_2D	✓	✓	✓	MACD_Signal_1d			
Asset_Growth_5D				MACD_Signal_2d			
Asset_Growth_22D				MACD_Signal_5d			
Asset_To_Equity_Ratio	✓	✓	✓	MACD_Signal_22d			
Capex_To_Cashflows	✓	✓	✓	AD_14D			
EBITDA_Yield		✓		AD_1D	✓		
EBIT_To_Assets	✓	✓	✓	AD_2D			
Net_Income_Margin	✓	✓	✓	AD_5D		✓	
Return_On_Invest_Capital	✓	✓	✓	AD_22D			✓
Mean_Reversion_1M			✓	ADX_29D	✓		✓
Mean_Reversion_2D	✓			ADX_1D			
Mean_Reversion_5D				ADX_2D			
Mean_Reversion_6D		✓		ADX_5D			

Table 1 Selected features for trading based on feature analysis for different timeframe (continued)

Feature name	*Trading position*			*Feature name*	*Trading position*		
	1D	*1W*	*1M*		*1D*	*1W*	*1M*
ADX_22D				PLUS_DI_22D		✓	
APO_12D_26D				PLUS_DM_15D			
ATR_15D		✓	✓	PLUS_DM_1D			✓
ATR_2D	✓			PLUS_DM_2D			
ATR_5D				PLUS_DM_5D			
ATR_22D				PLUS_DM_22D			
BETA_6D				PPO_12D_26D	✓		
BETA_1D				PPO_8D_13D		✓	
BETA_2D				PPO_1D_3D			
BETA_5D				PPO_24D_50D			✓
BETA_22D	✓	✓	✓	STDDEV			
RSI_10D			✓	TRANGE_2D		✓	✓
BOP	✓			TRANGE_1D			
CCI_14D			✓	TRANGE_5D	✓		
CCI_1D				TRANGE_22D			
CCI_2D	✓			TYPPRICE_1D			
CCI_5D	✓	✓		TYPPRICE_2D			
CCI_22D				TYPPRICE_5D			
CMO_15D			✓	TYPPRICE_22D			
CMO_5D				Earnings_Quality			
CMO_2D				WILLR_14D			✓
CMO_22D		✓		WILLR_1D	✓		
DX_15D				WILLR_2D			
DX_22D		✓	✓	WILLR_5D			
DX_2D	✓			WILLR_22D		✓	
DX_5D				Average_Dollar_Volume	✓	✓	✓
MAX				Moneyflow_Volume_5D			
MAXINDEX				Moneyflow_Volume_1D			✓
MEDPRICE_1D	✓	✓	✓	Moneyflow_Volume_2D		✓	
MEDPRICE_2D				Moneyflow_Volume_22D	✓		
MEDPRICE_5D				Annualised_Volatility	✓	✓	✓
MEDPRICE_22D				Operating_Cashflows_To_Assets	✓	✓	✓
MFI_15D				Price_Momentum_3M	✓		✓
MFI_1D				Price_Oscillator_20D			
MFI_2D			✓	Price_Oscillator_1D	✓		
MFI_5D				Price_Oscillator_2D			
MFI_22D	✓	✓		Price_Oscillator_5D			
MIDPOINT				Price_Oscillator_22D			
MIN				Returns_215D		✓	✓
MININDEX				Returns_190D			
MINUS_DI_15D			✓	Returns_160D			
MINUS_DI_1D				Returns_100D	✓		
MINUS_DI_2D				Returns_50D			
MINUS_DI_5D				Returns_25D			
MINUS_DI_22D				Trendline_252D	✓	✓	✓
MINUS_DM_15D				Trendline_25D			
MINUS_DM_1D				Trendline_50D			
MINUS_DM_2D	✓			Trendline_100D			
MINUS_DM_5D				Trendline_150D			
MINUS_DM_22D				Vol_3M			
PLUS_DI_15D				Vol_1D			
PLUS_DI_1D				Vol_2D			
PLUS_DI_2D	✓			Vol_5D	✓		
PLUS_DI_5D		✓	✓	Vol_22D		✓	✓

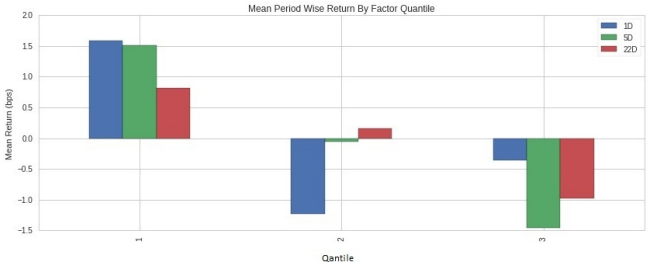
4.1 Feature evaluation example for WILLR 14 day

The same evaluation was done for all the features. For demonstrating purposes we only show the graphical results of the feature WILLR_14D.

1 *Mean period wise return by factor quartile*

The total return is shown in Figure 3 as a function of graph height. Long is represented by a positive graph, whereas short is represented by a negative graph. In our scenario, we're separating three quartiles into three different days (1D 5D 22D), which we may trade in the Quantopian environment. Figure 3 shows that the factor chosen works best with 5D trading since we get a fair amount of return for both long and short, however for 1d trading, we can see that long produces a good result for the first quartile but is not the best for going short as seen in the third quartile.

Figure 3 Mean period wise return by factor quartile (see online version for colours)



2 *Factor weighted long-short portfolio cumulative return*

This graph depicts the position of the trader's portfolio if the individual only traded using the experimented factor. This indicates the trader's total returns on his portfolio.
Figure 4 shows distinct portfolio positions for three different trading frequencies (1D, 5D, 22D) as measured by quartile deceleration.

3 *Period wise return by factor quantile*

When the median value is not a dependable alternative for judging the status of the data being experimented on, this graph, also known as the violin graph, comes in help.

When comparing the summary statistics of the range of quartiles, this graph comes in handy. This graph's format is fairly similar to that in Figure 5, but this time we can see the density of where our returns are focused for each time period.

4 *Cumulative returns by quantile*

Each time period's cumulative quantiles are obtained and averaged across the trading time period. The major goal of this curve is to check whether the quartiles can be separated as much as feasible. The greater the distance between them, the better. The third quantile is obviously higher than the first quartile, and this becomes more apparent as time passes. The fewer graphs that overlap one other, the better.

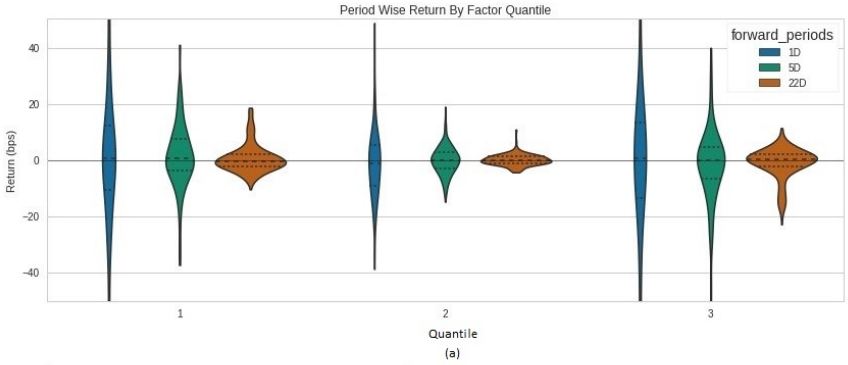
This is estimated for the three distinct quartiles for the time period shown in Figure 6.

5 *Top minus bottom quantile mean*

To smooth out the findings for the provided trading time period, this graph in Figure 7 subtracts the top quantile from the bottom quantile and gets the mean of the response. The higher the positive graph plot, the higher the return throughout the transaction period.

Figure 4 Factor weighted long-short portfolio cumulative return, (a) 1D (b) 5D (c) 22D (see online version for colours)

Figure 5 Period wise return by factor quantile (see online version for colours)



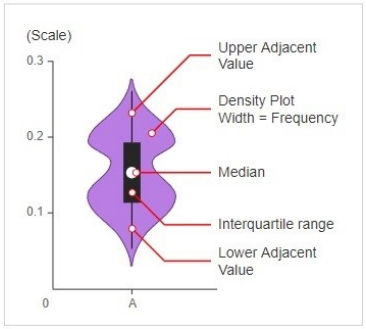


Figure 6 Cumulative returns by quantile, (a) 1D (b) 5D (c) 22D (see online version for colours)

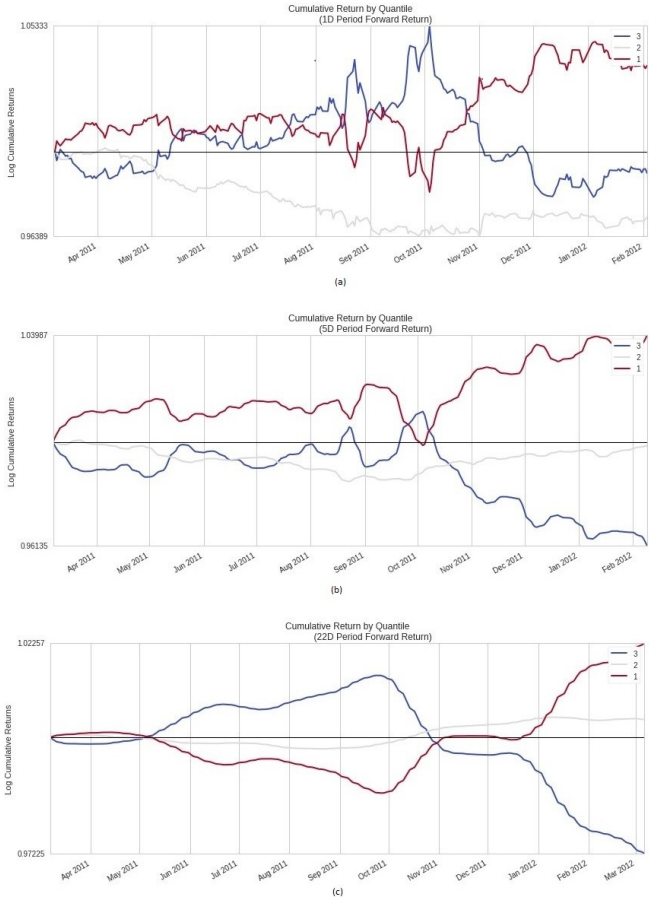


Figure 7 Top minus bottom quantile mean, (a) 1D (b) 5D (c) 22D (see online version for colours)

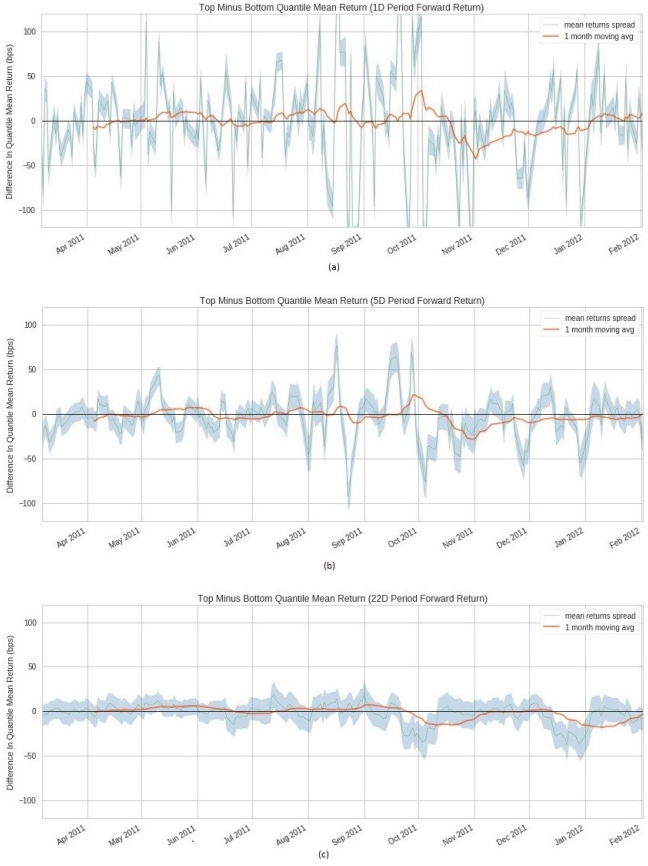
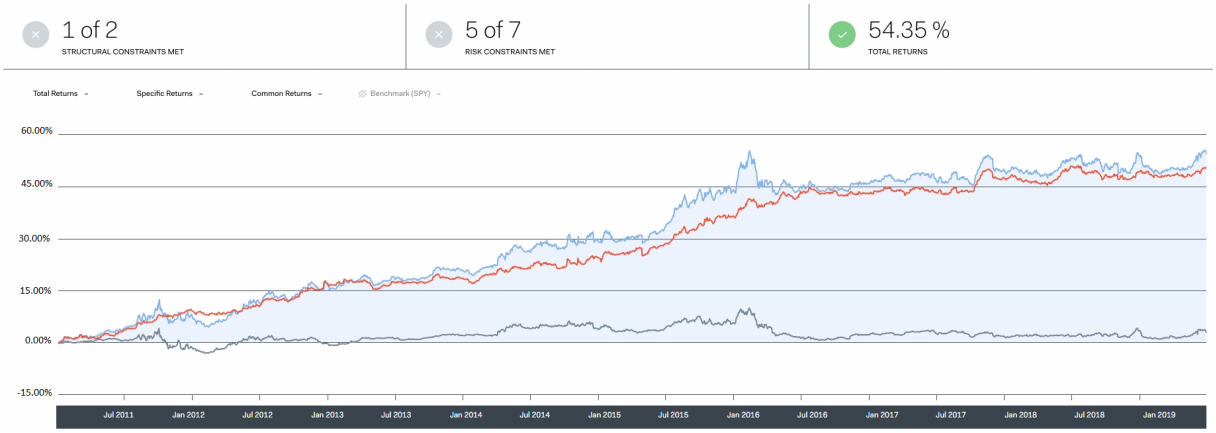


Figure 8 Top result daily trade by ensembling GaussianNB, LogisticRegression, DTC and SGDC (best classifier) (see online version for colours)



5 Live trading results

We used all the seven classifiers discussed to start to perform calculations on data from 2011-03-06 to 2011-09-7. We split by 80:20 ratio to form the train set and test set. Table 2 shows that ensemble methods work far better in this case. However, for ensemble methods, we only predicted the top and bottom values, as in real life we do not need to trade all the 1,500 stocks. The ensemble 1 model is showing an accuracy of 99.25% included LR, Gaussian_NB, Bernoulli_NB and SGDC whereas the ensemble 2 showing an accuracy of 74.23% consisted of LR_L1Regress, LR_L2Regress, Gaussian_NB and Bernoulli_NB.

Table 2 Accuracy test on data from 1,500 US stocks 2011-03-06 to 2011-09-7

Name of the algorithm	*Test accuracy*
Naive Bayes (NB)	51.21%
Logistic regression (LR)	51.77%
Stochastic gradient descent (SGDC)	50.56%
Support vector machine (SVM)	54.06%
Adaboost	53.29 %
Random forest	52.43%
Ensemble 1 (predict top and bottom)	99.25%
Ensemble 2 (predict top and bottom)	74.23%

5.1 *Day trading*

- *RandomForest:* using random forest algorithm and daily trading we get a return of 18.08% with a Sharpe ratio of 0.77.
- *AdaBoost:* using AdaBoost classifier in the mix we get a return of 11.69% with a sharpe ratio of 0.49.
- *Ensemble 1 classifiers:*

  1 GaussianNB
  2 LogisticRegression
  3 BernoulliNB
  4 Sgdc.

  Using the mixed classifiers of all these algorithms together we get a return of 34.99% with a Sharpe ratio of 0.67. Time complexity of all these algorithms combined is very less and thus is very feasible for our purpose.
- *Best classifiers:*

  1 GaussianNB
  2 LogisticRegression
  3 DTC
  4 Sgdc.

  Figure 8 shows, using decision tree classifiers in the mix we get a return of 54.63% with a sharpe ratio of 1.16%.

  1 GaussianNB
  2 LogisticRegression
  3 AdaBoostClassifier
  4 Sgdc.

5.2 *Weekly trading*

- *AdaBoostClassifier:* using AdaBoost for weekly trading we get a return of 5.25%.
- *Decision tree:* decision tree for weekly trading we get a total return of 10.23%.
- *Random forest:* using random forest we get a total return of 7.86%.

5.3 *Monthly trading*

- *AdaBoostClassifier:* using AdaBoost for weekly trading we get a return of 6.16%.

- *SVM:* Using AdaBoost for weekly trading we get a return of 13.05%.
- *Random forest:* Using AdaBoost for weekly trading we get a return of 4.05%.

6 Performance evaluation and risk evaluation

Our best algorithm from all of the above was the ensemble learning algorithm which incorporated Gaussian naive Bayes classifier, logistic regression, decision tree classifier and stochastic gradient descent classifier. The training day for each decision was set to be 200 days prior to that day and trading was done daily. Below are the few results that are got by running the algorithm from the date 01/04/2011 to 07/05/2019 with a capital of 10,000,000 USD.

Table 3 Performance of the system's best model

Total returns	54.35%
Specific returns	50.60%
Common returns	2.71%
Sharp	1.16%
Max draw down	–8.31%
Volatility	0.05%

Table 3 depicts that returns calculated from the initial investment were 54.35% on the total capital. The average Sharpe ratio is 1.16 and the average volatility is 0.05 and the final max drawdown was –8.31. These values indicate that our model returns a portfolio that has a low level of risk.

Total returns: it is the sum of an investment's returns over a certain time period. This reflects two distinct investment types:

1 fixed income investment

2 distribution and capital appreciation

Common returns: it is a measure of how much of your overall returns can be ascribed to common risk variables like market beta, sectors, momentum, mean reversion, volatility, size, and value, as estimated by Quantopian. If all of your results are the same, your algorithm is not doing anything special and hence is not really useful. 2.71% of frequent results are shown in Table 3.

Specific returns: it is an excess return that we get from an asset that is independent of specific returns of other assets. Table 3 shows 50.60% of common returns.

Sharpe ratio: it is the measure of performance measure of investment by risk adjustment. It measures the excess returns for every unit deviation of a trade. Our approach had a 1.16% Sharpe ratio which is decent shown in Table 3.

$$\text{Sharpe Ratio} = \frac{E_p - E_f}{\sigma_p} \tag{1}$$

where

E_p return of portfolio

E_f risk-free rate

σ_p portfolio additional return's standard deviation.

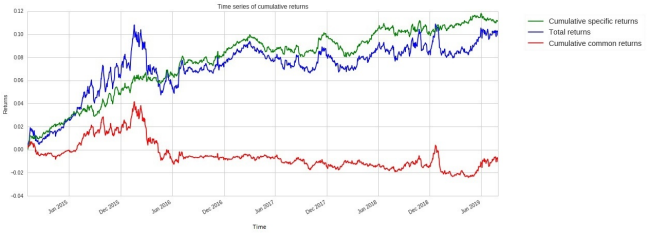
Max drawdown: it is the biggest loss recorded between the graph's highest and smallest observed points. This is used to determine a stock strategy's relative risk.

$$MDD = \frac{\text{Trough value} - \text{Peak value}}{\text{Peak value}} \tag{2}$$

Volatility: it is the measure of risk and it shows how much the portfolio fluctuates over time.

Cumulative specific and total returns: cumulative returns are independent of the time period and us the total amount of profit or loss from a particular investment. The common returns are very low which is a good sign for the model as it means that our algorithm has a low beta and performs well irrespective of whether the stock prices rise or fall. Which made the specific return very high (50.60%) as shown in Figure 9.

Figure 9 Cumulative specific and total returns (see online version for colours)

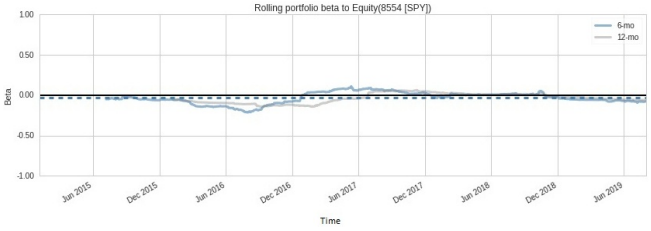


Returns over time: returns are gains or losses made by a particular investment. Returns can be expressed as the percentage increase or decrease in a particular investment or it can be quantified in a particular currency. Figure 10 shows that the returns are mostly positive.

Figure 10 Returns over time (see online version for colours)

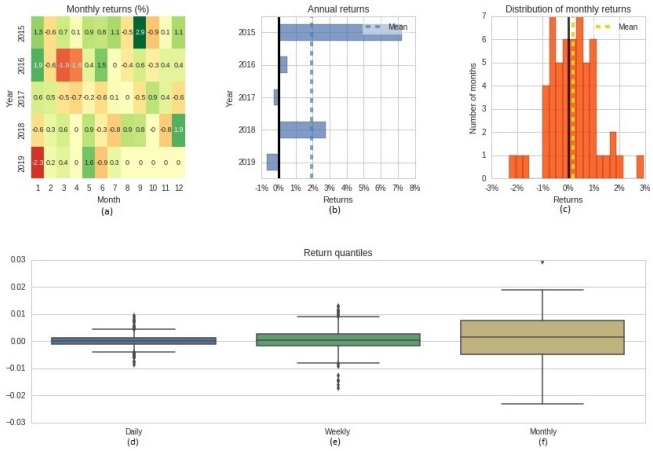
Rolling portfolio beta to equity: this is shown in Figure 11. The beta is the risk that can be attributed to the movement of the market. A beta having the value 1 signifies that a portfolio follows the trend of the market precisely. Whereas, a beta having a lower value than 1 means that a portfolio is less correlated with the overall market. A low beta value incorporated with a high Alpha value will mean that the portfolio will make a profit irrespective of the market movement.

Figure 11 Rolling portfolio beta to equity (see online version for colours)



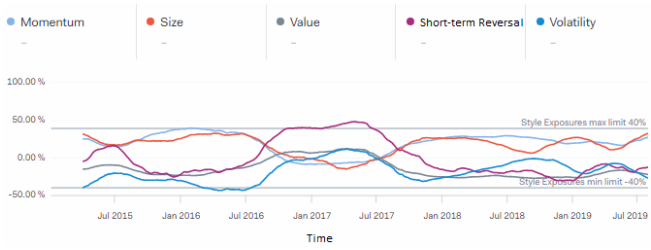
Daily weekly and Monthly returns: Figure 12 illustrates returns over the daily, weekly and monthly periods are indicated in the above figure. Each figure gives how much profit was made in a particular period. The daily, annual and monthly

Figure 12 (a) Daily returns (b) Weekly returns (c) Monthly returns (d) Daily quantiles (e) Weekly quantiles (f) Monthly quantiles (see online version for colours)



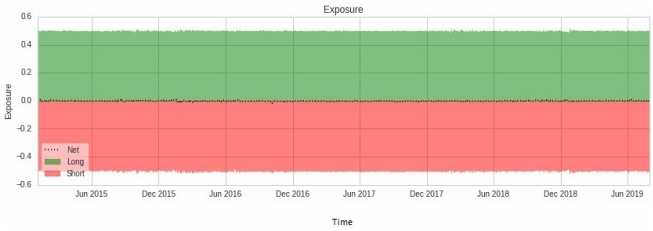
Style exposure: Figure 13 shows, exposure to various investing styles. The values displayed are the rolling 63-day mean. The relevant styles are described below:

Figure 13 Expose to momentum, size, value, short-term reversal and volatility (see online version for colours)



Ratio of long and short position: we implemented an equal amount of long and short position strategy as shown in Figure 14. So at a time, we went long on 250 stocks and short on 250 stocks. This made our model perform well both on a bull market and a bear market.

Figure 14 Ratio of long and short position (see online version for colours)

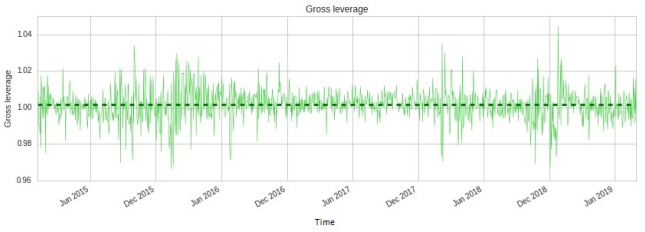


Daily holdings: from Figure 15, we see the total daily holdings of our portfolio which never exceeds 500. As we set our maximum holding limit in our portfolio to be 500.

Figure 15 Daily holdings (see online version for colours)

Gross leverage: Figure 16 shows, we kept our leverage at max 1.05 and at least 0.96 so that our money would be utilised but avoided the risk of being liquidated.

Figure 16 Gross leverage (see online version for colours)



All the features that we calculated were later filtered out and grouped out into their specific dates for trading where they perform the best. The three categories are weekly trading, monthly trading and daily trading. We then used specific different algorithms to trade in order to compare their performance.

6.1 *Time complexity analysis of the model*

In Table 4, n is the number of training examples and p is the number of features.

The proposed feature selection takes a lot of time as it uses four different types of analysis. The other feature reduction algorithm and the other standard machine learning are also mentioned in the table. In the ensemble learning method, each algorithm had to be trained individually before the final result. Therefore the combination of the most efficient algorithm generated the best result as in the Quantopian platform time is an important factor.

Figure 17 Descriptive statistics of the synthetic data

	WILLR1	CCI_5D	TRANGE_5D	Price Oscillator1D	BOP	MFI2D	Earnings Quality	ATR2	PLUS_DI2	CMO2D	MEDPRICE
count	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000
mean	44.005103	3.516222	28.767159	0.495170	48.485151	36.918114	2.039838	5.980287	14.066248	35.192657	8.408863
std	21.577075	1.446376	16.026859	0.289945	13.483475	20.696564	1.147104	1.743082	5.751792	16.711553	4.339233
min	7.002618	1.000060	1.001068	0.000130	25.012245	1.005475	0.003682	3.000239	4.000696	6.004086	1.012838
25%	25.024638	2.262077	15.015692	0.243277	36.903169	19.160635	1.042545	4.453646	9.139007	20.585906	4.678866
50%	44.221253	3.529280	28.812670	0.492920	48.342528	36.547465	2.058611	5.992757	14.080048	35.295630	8.333083
75%	62.710065	4.772728	42.653096	0.750678	59.947694	55.220304	3.038629	7.513247	19.137138	49.952540	12.074140
max	80.986973	5.999387	55.995911	0.999993	71.994185	71.975806	3.999677	8.996947	23.958355	63.992037	15.995171

Table 4 Time-complexity of the proposed methods and machine learning algorithms

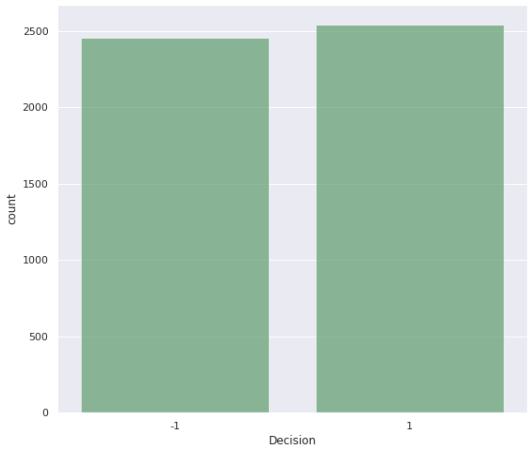
Algorithm name	*Time complexity*
Proposed feature selection	$O(pn^3)$
Feature reduction (ANOVA)	$O(pn \log n)$
Naive Bayes	$O(np)$
Logistic regression	$O(p^2n + p^3)$
Stochastic gradient descent	$O(pn^2)$
Support vector machine	$O(n^2p + n^3)$
AdaBoost	$O(np^2)$
Random forest	$O(n^2pntrees)$
Decision tree	$O(n^2p)$

7 Evaluation on synthetic dataset

To further evaluate our model we created two synthetic datasets. In order to test if our model works both for normally distributed dataset and non-Gaussian dataset we use two different types of data generation techniques. All of the 148 data were used in both the datasets. After running our feature selection model 25 of the features were selected for final decision making.

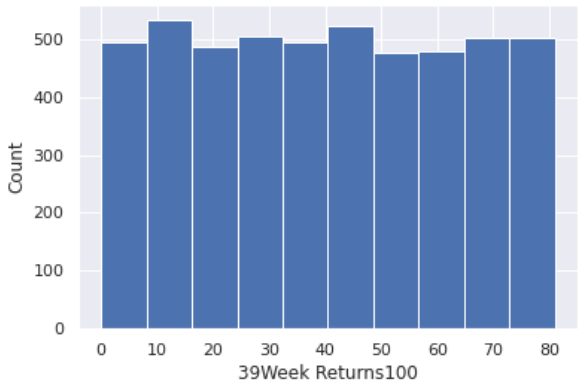
Figure 17 shows mean, standard deviation, minimum value, maximum value and specific percentile scores that give a basic idea about the synthetic datasets.

Figure 18 Labels of dataset 1 (uniform) (see online version for colours)



In Figure 18, the labels are for long and short are represented by 1 and –1 respectively. The number of each label is close to 2,500 adding up to a total of 5,000 instances.

Figure 19 Distribution of features in dataset 1 (uniform) (see online version for colours)



In Figure 19 the distribution of one of the features of the dataset 1 is shown. The min and max value of the data was required to generate this non-Gaussian distribution.

The correlation of the final 25 features is shown in Figure 20. The correlation of the features is important in determining the relationship between the features. Only one feature should out of two if they are highly co-related.

Figure 21 shows the boxplot of the 25 selected features. The values are scaled from 0 to 1. The middle line shows the mean of the feature. The distribution of the feature values can be visualised from Figure 21.

Figure 22 is the confusion matrix generated by the proposed best model. Our model achieves 84.20% accuracy in the non-Gaussian dataset. Using the values from the confusion matrix the recall is 82.46%, precision is 85.26% and the F1 score is 83.84%.

In Figure 23, the labels are for long and short are represented by 1 and –1 respectively. The number of each label is close to 2,500 adding up to a total of 5,000 instances.

Figure 20 Correlation heatmap of dataset 1 (uniform) (see online version for colours)

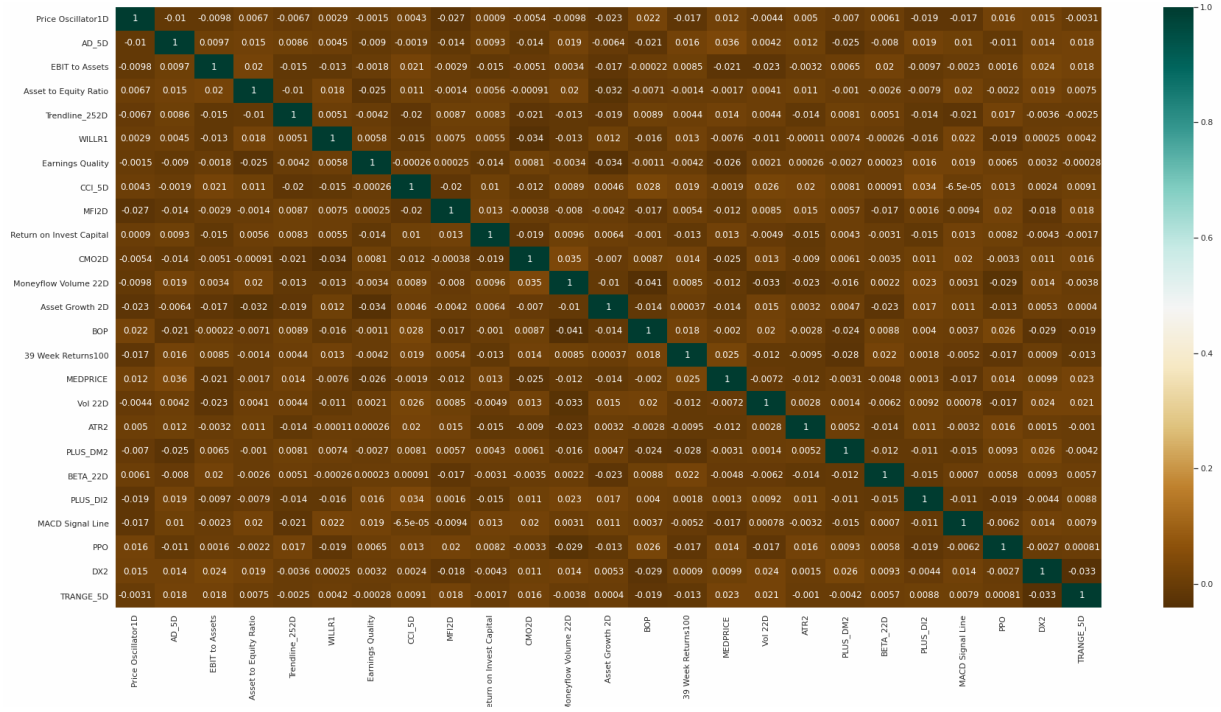


Figure 21 BoxPlot of the 25 selected features in dataset 1 (uniform) (see online version for colours)

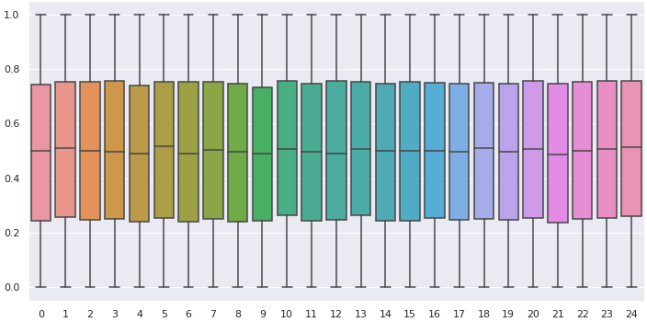


Figure 22 Confusion matrix from the ensemble method on dataset 1 (uniform) (see online version for colours)

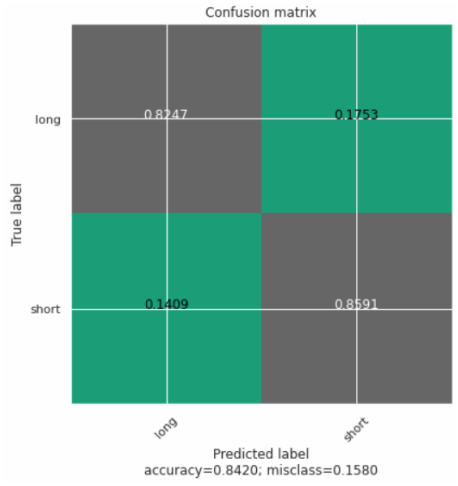


Figure 23 Labels of dataset 2 (Gaussian) (see online version for colours)

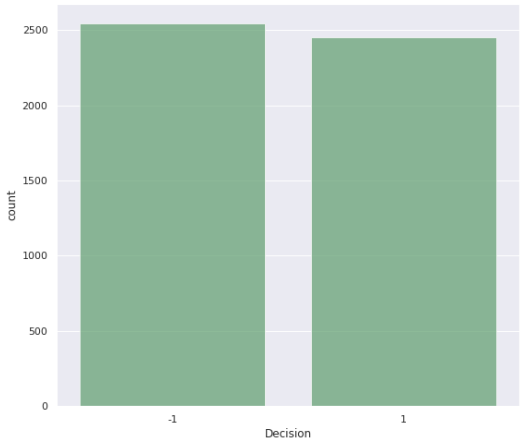


Figure 24 Distribution of features in dataset 2 (Gaussian) (see online version for colours)

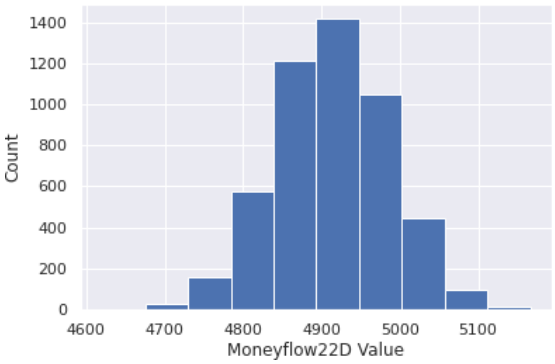
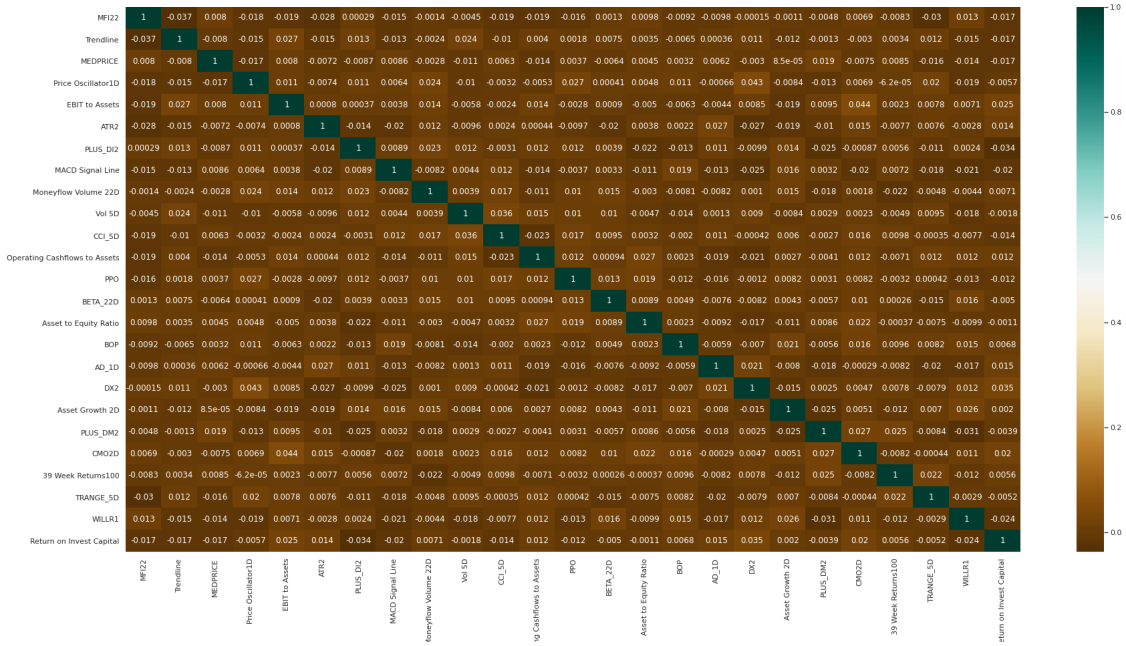


Figure 25 Correlation heatmap of dataset 2 (Gaussian)



In Figure 24 the distribution of one of the features of the dataset 2 is shown. The dataset is created using the mean value and standard deviation of the original data.

The correlation of the final 25 features is shown in Figure 25. The correlation of the features is important in determining the relationship between the features. Only one feature should out of two if they are highly co-related.

Figure 26 BoxPlot of the 25 selected features in dataset 2 (Gaussian) (see online version for colours)

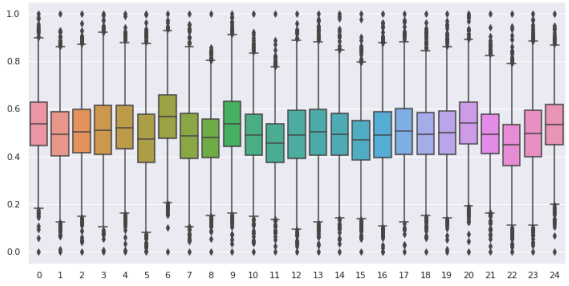


Figure 26 shows the boxplot of the 25 selected features. The values are scaled from 0 to 1. The middle line shows the mean of the feature. The distribution of the feature values can be visualised from Figure 21.

Figure 27 is the confusion matrix generated by the proposed best model. Our model achieves 88.30% accuracy in the non-Gaussian dataset. Using the values from the confusion matrix the recall is 89.70%, precision is 87.36% and the F1 score is 88.51%.

A greater X-axis value in a ROC curve implies a larger number of false positives than true negatives. While a higher Y-axis value implies a greater number of true positives than false negatives, a lower Y-axis value suggests a lower number of true positives. As a result, the threshold is determined by the capacity to balance false positives and false negatives.

Figure 27 Confusion matrix from the ensemble method on dataset 2 (Gaussian) (see online version for colours)

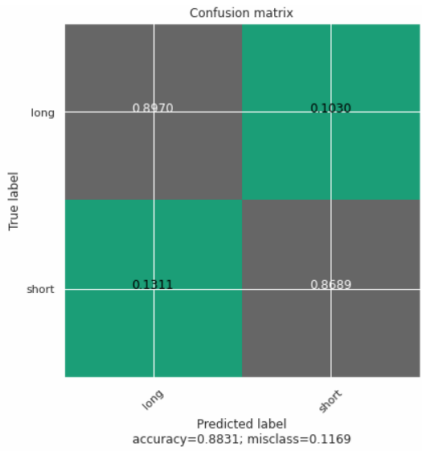
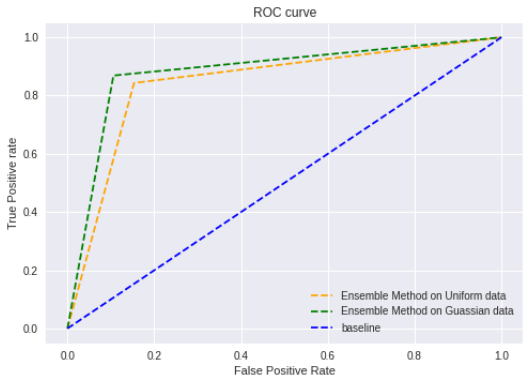


Figure 28 ROC curve of the synthetic datasets (see online version for colours)



It is evident from Figure 28 that the AUC for the ensemble model on Gaussian data ROC curve is higher than that for the non-Gaussian data ROC curve. Therefore, we can say

that our model does a better job of classifying the positive class in the normally distributed dataset. The AUC value of our model in the normally distributed dataset is 0.8814 and the AUC value of our model with uniformly distributed dataset is 0.8449.

Table 5 Proposed model performance on synthetic data

Type of dataset	*Accuracy*	*Precision*	*Recall*	*F1-score*	*AUC-score*
Normally distributed	88.30%	87.36%	89.70%	88.51%	0.8814
Uniformly distributed	84.20%	85.26%	82.46%	83.84%	0.8449

Table 5 shows that the proposed model works better when dataset is normally distributed. As most stock data is normally distributed that is why the proposed model is finely tuned to work better with normally distributed data. However, the model also performs quite well achieving 84.20% accuracy and quite good in other performance matrices as shown in Table 5.

8 Comparison with other models

The most important part of our model is our novel feature calculation and selection method. For this reason even with the huge drawbacks of the Quantopian platform our model performs on par with the state-of-the-art models that perform quantitative Trading as can be seen from Table 6. The biggest advantage of our proposed model is that the feature selection method can be added to any decision making model to make better predictions.

Table 6 Comparison of proposed model with state-of-the-art models

Author	*Time period*	*Returns*	*Trading frequency*	*Method used*
Dai et al. (2012)	5 years	30.66%	30 days	SVM and logistic regression
Chen et al. (2017)	7 years	103%	30 days	LSTM
Vo et al. (2019)	3 years	50.78%	1 year	Reinforcement learning
Proposed model	8 years	54.35%	1 day	Ensemble learning with feature selection

Table 6 shows that the proposed model performs better than the model of Dai and Chen. Moreover, the feature extraction and selection method can select the best feature for trading in any time-period. Therefore, this model can be incorporated with any model to significantly improve the quality of the features.

9 Findings and research challenges

In the instance of stock market trading, it is evident that ensemble learning provided a superior outcome than employing a single algorithm, as shown through experiments. Furthermore, it became evident that the feature extraction portion of a stock trading algorithm is the most crucial aspect. Because of the high quality of the features employed in 1 day trading, the algorithm produced 54.35% profit over the period of 8 years. The weekly and monthly algorithms, on the other hand, did not perform as well owing to their characteristics. Our most important contribution is that we identified which characteristics should operate well for which time range using statistical metrics. Over a one-day period, the model can clearly represent the market's tendency.

The Quantopian platform does not allow users to download their data; therefore, the model was restricted to the limitations of Quantopian. The highest data look back for daily trading was 200 days before the current trading day. In the case of weekly and monthly trading, the days were 150 days and 100 days, respectively. Due to these constraints, the weekly and the monthly trading algorithm could not perform as well as the daily trading algorithm.

We were unable to utilise Pipeline to train our model due to a lack of resources, and as a result, we were unable to train our models without exceeding the time limit for various methods. High frequency trading, such as hourly trading, is especially difficult to execute since Quantopian lacks functionality for hourly trading. We also could not use any form of neural network since Quantopian would not let us use Keras for tensor flow. Furthermore, we would have been able to run our own neural network over the data for better results if we had greater access to trade data, but we were unable to do so since Quantopian does not allow downloads of their datasets. We could have produced better and more accurate outcomes if we had greater resources and access to trade data.

10 Conclusions and future works

This thesis applied novel methods on the stock market of the USA and validated the data in an external synthetic dataset. In this research, we demonstrated a new feature selection method for trading with different time-frames. The results reflect success of the model in a live trading environment. For future implementation purposes, we intend to design our own reinforcement learning algorithm that will be specifically tailored for this purpose. In order to get better results, we would like to try high-frequency trading, preferably minutely and hourly. There is also a new platform called QuantConnect which offers more flexibility than Quantopain where we can do our future work without the stated limitations.

Information sharing statement

In order to ensure reproducibility of our research we published our entire work at: https://github.com/amanat9/QuantopianThesis.

References

Barak, S., Dahooie, J.H. and Tichỳ, T. (2015) 'Wrapper ANFIS-ICA method to do stock market timing and feature selection on the basis of Japanese candlestick', *Expert Systems with Applications*, Vol. 42, No. 23, pp.9221–9235.

Bloembergen, D., Hennes, D., McBurney, P. and Tuyls, K. (2015) 'Trading in markets with noisy information: an evolutionary analysis', *Connection Science*, Vol. 27, No. 3, pp.253–268.

Chang, Q. (2020) 'The sentiments of open financial information, public mood and stock returns: an empirical study on chinese growth enterprise market', *International Journal of Computational Science and Engineering*, Vol. 23, No. 2, pp.103–114.

Chen, G., Chen, Y. and Fushimi, T. (2017) *Application of Deep Learning to Algorithmic Trading* Technical report, Stanford University [online] http://cs229.stanford.edu/proj2017/final-reports/5241098.pdf (accessed 7 January 2019).

Chen, W., Zheng, Z., Ma, M., Wu, J., Zhou, Y. and Yao, J. (2020) 'Dependence structure between bitcoin price and its influence factors', *International Journal of Computational Science and Engineering*, Vol. 21, No. 3, pp.334–345.

Dai, T., Shah, A. and Zhong, H. (2012) *Automated Stock Trading Using Machine Learning Algorithms*, Technical report, Stanford University [online] https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.278.5891&rep=rep1&type=pdf (accessed 7 January 2019).

Gandhmal, D.P. and Kumar, K. (2021) 'Wrapper-enabled feature selection and cplm-based narx model for stock market prediction', *The Computer Journal*, Vol. 64, No. 2, pp.169–184.

Grant, J. (2011) 'High-frequency boom time hits slowdown', *Financial Times*, p.12.

He, Y., Fataliyev, K. and Wang, L. (2013) 'Feature selection for stock market analysis', *International Conference on Neural Information Processing*, Springer, pp.737–744.

Hegazy, O., Soliman, O.S. and Salam, M.A. (2014) *A Machine Learning Model for Stock Market Prediction*, arXiv preprint arXiv:1402.7351 [online] https://arxiv.org/ftp/arxiv/papers/1402/1402.7351.pdf (accessed 7 January 2019).

Hu, J. (2021) 'Local-constraint transformer network for stock movement prediction', *International Journal of Computational Science and Engineering*, Vol. 24, No. 4, pp.429–437.

Huang, C-L. and Tsai, C-Y. (2009) 'A hybrid SOFM-SVR with a filter-based feature selection for stock market forecasting', *Expert Systems with Applications*, Vol. 36, No. 2, pp.1529–1539.

Jacobs, B.I. and Levy, K.N. (1993) 'Long/short equity investing', *Journal of Portfolio Management*, Vol. 20, No. 1, p.52 [online] https://jlem.com/documents/FG/jlem/articles/580182_LongShortEquityInvesting.pdf.

Kim, K-J. (2003) 'Financial time series forecasting using support vector machines', *Neurocomputing*, Vol. 55, Nos. 1–2, pp.307–319.

Madge, S. and Bhatt, S. (2015) *Predicting Stock Price Direction Using Support Vector Machines*, Independent Work Report Spring [online] https://www.cs.princeton.edu/sites/default/files/uploads/saahil_madge.pdf (accessed 7 January 2019).

Malkiel, B.G. and Fama, E.F. (1970) 'Efficient capital markets: a review of theory and empirical work', *The Journal of Finance*, Vol. 25, No. 2, pp.383–417.

Nti, K.O., Adekoya, A. and Weyori, B. (2019) 'Random forest based feature selection of macroeconomic variables for stock market prediction', *American Journal of Applied Sciences*, Vol. 16, No. 7, pp.200–212.

O'Reilly, R. (2012) 'High frequency trading: are our vital capital markets at risk from a rampant form of trading that ignored business fundamentals?', *The Analyst*.

Ou, P. and Wang, H. (2009) 'Prediction of stock market index movement by ten data mining techniques', *Modern Applied Science*, Vol. 3, No. 12, pp.28–42 [online] https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.906.2882&rep=rep1&type=pdf (accessed 7 January 2019).

Patel, J., Shah, S., Thakkar, P. and Kotecha, K. (2015) 'Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques', *Expert Systems with Applications*, Vol. 42, No. 1, pp.259–268.

Sun, J. and Li, H. (2012) 'Financial distress prediction using support vector machines: Ensemble vs. individual', *Applied Soft Computing*, Vol. 12, No. 8, pp.2254–2265.

Tao, X., Renmu, H., Peng, W. and Dongjie, X. (2004) 'Input dimension reduction for load forecasting based on support vector machines', *2004 IEEE International Conference on Electric Utility Deregulation, Restructuring and Power Technologies: Proceedings*, IEEE, Vol. 2, pp.510–514.

Tiong, L.C., Ngo, D.C. and Lee, Y. (2016) 'Forex prediction engine: framework, modelling techniques and implementations', *International Journal of Computational Science and Engineering*, Vol. 13, No. 4, pp.364–377.

Tsai, C-F. and Hsiao, Y-C. (2010) 'Combining multiple feature selection methods for stock prediction: Union, intersection, and multi-intersection approaches', *Decision Support Systems*, Vol. 50, No. 1, pp.258–269.

Tsai, C-F., Lin, Y-C., Yen, D.C. and Chen, Y-M. (2011) 'Predicting stock returns by classifier ensembles', *Applied Soft Computing*, Vol. 11, No. 2, pp.2452–2459.

Vapnik, V. (2013) *The Nature of Statistical Learning Theory*, Springer Science & Business Media [online] http://bit.ly/statisticalLearningTheory (accessed 7 January 2019).

Wu, S., Liu, Y., Zou, Z. and Weng, T-H. (2021) 'S_I_LSTM: stock price prediction based on multiple data sources and sentiment analysis', *Connection Science*, pp.1–19.

Yuan, X., Yuan, J., Jiang, T. and Ain, Q.U. (2020) 'Integrated long-term stock selection models based on feature selection and machine learning algorithms for China stock market', *IEEE Access*, Vol. 8, pp.22672–22685 [online] https://ieeexplore.ieee.org/abstract/document/8968561.

Zhang, X., Hu, Y., Xie, K., Wang, S., Ngai, E. and Liu, M. (2014) 'A causal feature selection algorithm for stock prediction modeling', *Neurocomputing*, Vol. 142, pp.48–59 [online] https://www.sciencedirect.com/science/article/pii/S0925231214005359.

Zheng, A. and Jin, J. (2017) *Using AI to Make Predictions on Stock Market*, Technical report, Stanford University [online] http://cs229.stanford.edu/proj2017/final-reports/5212256.pdf (accessed 7 January 2019).