Capturing dynamics of post-earnings-announcement drift using genetic algorithm-optimized supervised learnings

Zhengxin Joseph Ye,[1] Thomas Heinis,[1] and Pantelis J. Beaghton[1]
Department of Computing, Imperial College

(Dated: 28 July 2019)

Post-Earnings-Announcement Drift (PEAD) is a stock market phenomenon when a stock's cumulative abnormal returns have a tendency to drift in the direction of an earnings surprise for weeks following an earnings announcement. Although it is one of the most studied stock market anomalies and its existence is well understood, the current literature is limited in explaining this phenomenon by a small number of factors using simpler regression methods and hasn't been able to accurately perform prediction on it using known factors. In this paper we aim to use supervised learning models to capture the dynamics of stock' quarterly PEAD using a wider range of both fundamental and technical factors. We test a deep neural network (Multiplayer Perceptron), an extreme gradient boosting model (XGBoost) as well as support vector machines (SVM) with different kernels on earnings announcement data from 386 S&P500 companies between 1996 and 2018. Our experiments show that XGBoost performs better in PEAD predictions of 3035 out-of-sample test stocks than MLP and SVM. More significantly, since we put the focus of our experiments and analysis at the portfolio level, for the first time in the literature we've produced post-earnings stock return predictions that are strongly correlated with return of portfolios consisting of out-of-sample test stocks. Our methods and results consistently show that we can form the highest-returning portfolios out of the sample population for buy-and-hold trading strategies and the lowest-returning portfolios for short selling strategies.

I. INTRODUCTION

The stock market is characterized by nonlinearities, discontinuities, and multi-polynomial components because it continuously interacts with many factors such as individual company's news, political events, macro economic conditions, and general supply and demand, etc [1]. The non-stationary nature of the stock market is supported by a widely believed but still hotly contested economic theory Efficient Market Hypothesis which states that asset prices fully reflect all available information and the market only moves by reacting to new information. Such a theory implies that the stock market behaves like a martingale and knowledge of all past prices is not informative regarding the expectation of future prices.

Ball and Brown [2] were the first to note that after earnings are announced, estimated cumulative abnormal returns continue to drift up for firms that are perceived to have reported good financial results for the preceding quarter and drift down for firms whose results have turned out worse than the market had expected. The discovery of Post Earnings Announcement Drift, which is a violation of semi-strong Efficient Market Hypothesis, seems to suggest that while stock markets are generally efficient, there may be information leakages around the announcement dates, coupled with post-earnings drift, resulting in price movement anomalies. It also seems to suggest that past stock price information or other past economic or financial information can potentially be used to predict price movement following an earnings announcement.

We have noticed that a lot of researches on PEAD came out in the late 1980s and 1990s. Fama and French [3] shows that average stock returns co-vary with three factors, namely, the market risk factor, the book-to-market factor, and the size factor. Bhushan suggests that the existence of sophisticated and unsophisticated investors, transaction costs and economies of scale in managing money can explain the market's delayed response to earnings [4]. Nearly all previous research has pooled companies with negative and positive earnings surprises when measuring the effect of earnings surprises on abnormal returns and regress the absolute value of earnings surprise as well as other factors against the absolute value of abnormal return [5]. However, we believe that stock markets don't react symmetrically to negative and positive earnings surprises and there are a lot more factors in play that drive the near term risk adjusted returns of a stock following an earnings release.

By using machine learning models we have manged to leap straight to the more important goal of predicting PEAD. In this process we've overcome a number of constraints commonly seen in previous researches: we are including a much wider range of factors including both fundamental and technical/momentum factors; we achieve a higher level of generality without having to pre-group companies by the value of their earnings surprises or other attributes prior to the analysis or prediction. Additionally we've chosen 386 S&P500 stocks (and increasing) that existed as a component of the index over at least half of the chosen test time period between 1996 and 2018. Our selection includes companies that either went bankrupt or dropped out of S&P500, significantly reducing survivorship bias in our training data. This company population is larger than a lot of previous studies of similar nature. For example Beyaz and co only chose 140 stocks from S&P500 when they attempted to forecast stock prices both six months and a year out based on fundamental analysis and technical analysis [6] and Bradbury used a sample of only 172 firms to research the relationships among voluntary semi-annual earnings disclosures, earnings volatility, unexpected earnings, and firm size [7].

Recognising the highly nonlinear nature of stock price movements, we've chosen a variety of supervised learning models in search of one which can best work through the high noises embedded in the price data. We've experimented with a deep neural network, a varieties of support vector machines with different kernels, and an Extreme Gradient Boosting (XGB) model. More conclusion on individual model's results here.

In our experiments with all these models, we divide the training data into an in-sample and out-of-sample period of varying lengths and use the in-sample data set to tune a model's hyperparameters. Our early experiments show that a traditional grid search way of finding optimal parameter set is incomplete and very slow. Instead we've chosen to use the highly adaptable Genetic Algorithm to tune our models. Additionally, as the search range and granularity of each model's tunable hyperparameters examined by the Genetic Algorithm directly determines the complexity of the resulting model and is sometimes not known beforehand, they must be chosen sensibly. Searching for a limited number of parameter combinations will result in an inappropriate model that is not able to fit the essential structure of the training set while too many combinations reduce the whole search and fitting process' efficiency. To address these concerns, we employ a 5-fold cross validation (CV) within each Genetic Algorithm iteration for estimating the optimal combination of each model's hyperparameters.

II. RELATED WORK

Since the discovery of Post Earnings Announcement Drift as a stock market anomaly by Ball and Brown [2] who documented the return predictability for up to two months after the annual earnings announcements, an extensive research has been carried out in literature though with varying results. For example Foster, Olsen and Shevlin [8] found systematic post-announcement drifts in security returns are only found for a subset of earnings expectations models when testing drifts in the [+1, +60] trading day period. In recent years the literature has become less limited to the specific study of PEAD and instead put more focus on the direct predictions of stock price movement using stocks' fundamental and/or technical information, again with varying rate of success. Malkiel studied the impact of price/earnings (P/E) ratios and dividend yields on stock prices using the Campbell-Shiller model. He conceded his work demonstrated that exploitable arbitrage didn't exist for investors to earn excess risk-adjusted returns and he could not find a market timing strategy capable of producing returns exceeding buying and hold a broad market index [9]. Olson and Mossman not only showed that artificial neural network outperforms traditional regression based methods when forecasting 12-month returns by examining 61 financial ratios for 2352 Canadian stocks but more importantly shows that by using fundamental metrics sourced from financial reports they were able to achieve excessive risk-adjusted returns [10].

Other authors went beyond metrics from earnings reports and attempted stock forecast using both fundamental and technical analysis. Sheta, et al. explored the use of ANN, SVM and Multiple Linear Regression for prediction of S&P500 market index. They selected 27 technical indicators as well as macro economic indicators and reported that SVM contributes to better predictions than the other models tested [11]. Hafezi et all considered both fundamental and technical analyses in a novel model called Bat-neural Network Multi-agent System when forecasting stock returns. The resulted MAPE statistic showed that the new model performed better than typical Neural Network coupled with Genetic Algorithm [12].

When it comes to selecting machine learning models for event driven stock price forecast the literature has looked a lot at Support Vector Machines. Zhang constructed a novel ensemble method integrated with AdaBoost algorithm, probabilistic Support Vector Machine and Genetic Algorithm and verified its performance over 20 shares from the SZSE and 16 stocks from NASDAQ. He showed the new ensemble method achieved preferable profit in simulation of stock investment [13]. Madge used daily closing prices for 34 technology stocks on a SVM model with radial kernel to calculate price volatility and momentum for individual stocks and for the overall sector. The model attempts to predict whether a stock price sometime in the future will be higher or lower than it is on a given day. They found little predictive ability in the short-run but definite predictive ability in the long-run [14]. We have not found any creditable research on stock forecast using XGBoost and we are contributing to the literature for that.

Joseph's note to tutors (to be removed): I have conducted literature review on other individual technical topics such as Genetic Algorithm and Cross Validation as well as stock price forecasting under other settings and scenarios. I will be able to extend the literature review section into these areas in the real paper if necessary.

III. MODEL FEATURES GENERATION

We have chosen in total 386 S&P500 companies for analysis. The chosen time frame was between the fourth financial quarter of 1996 (1996 Q4) and the second financial quarter of 2018 (2018 Q2). The chosen companies were in continuous operation as well as being a constituent of the S&P500 index during at least half of the chosen time frame.

While the output of the learning model are risk-adjusted near term stock movements (%change) of individual companies following each quarter's earnings announcement, the input to our models consists of the following sets of data which we've sourced from Bloomberg:

- Financial statements data
- Earnings Surprise data
- Price movements data
- Momentum indicator data

Although we've sourced totally 32973 quarterly financial statements from our chosen companies over the test time frame, the test population eventually comes down to 19492 data points which we use for training and prediction. There are a number of reasons for the reduction: (a) not every chosen feature exists for all the companies and over all the historical quarters and this is particularly the case with the Earnings data; (b) we've discarded certain companies in certain historical quarters when the earnings reports suffered badly from missing data; (c) We've been very careful with whether an earnings report was released before market opened or after market closed as such a difference is significant but Bloomberg is missing such an information for some earlier earnings quarters and we've discarded those quarters.

A. Financial Statements data

Table 1 shows 24 metrics from earnings reports have been chosen to create training data.

Cash (or equivalent) account	Short Term Debt
Total Equity	Long Term Debt
Total Liability	Long Term Debt / Capital
Net Income	Long Term Debt / Equity
Net Income Available to Common	Long Term Debt / Total Asset
Operating Income	Total Debt / Capital
Operating Margin	Total Debt / Equity
Pre tax Income	Total Debt / Total Asset
Profit Margin	Return on Asset
Revenue	Return on Common Equity
Common Equity / Total Asset	Earnings per share (adjusted and diluted)
Net Debt / EBIT	Income Tax Expense

TABLE I. Earnings report metrics chosen as input features

These metrics will be pre-processed in order for them to make better sense to the learning models and new features will be created through feature engineering. Specially: (a) we've turned income statement items and cashflow statement items into percentage of the same quarter's revenue level and turned balance sheet items into percentage of the same quarter's total asset level. Such a transformation is done according to the principle of Vertical Analysis [15] which determines the relative weight of each item and its share in asset resources or revenue generation. (b) Instead of using the reported financial metrics directly as model inputs, we've calculated the simple arithmetic difference of each of all the 24 report metrics (except for revenue which is calculated as %change) from value of the same metric the quarter before (quarterly change) and the year before (yearly change). Through this pre-processing step we've effectively engineered 48 features out of the chosen earnings report metrics.

Joseph's note to tutors (to be removed): Initially a larger set of financial report metrics had been chosen and used. This included other commonly known metrics such as Price/EPS ratio, Price/Book ratio, etc. However, some metrics were dropped because they had suffered from missing data. Handling missing data is on its own a technical subject to be considered and possibly explored

B. Earnings Surprise data

Earnings Surprise represents how much a company's actual reported Earnings Per Share (EPS) is more (or less) than the average of a selected group of stock analysts' estimates on that quarter's EPS. We are not calculating Earnings Surprise as a %change between the reported EPS and market estimated EPS because (a) %change is too volatile as a very small change when the actual EPS levels is close to zero will lead to a misleading large %change, and (b) we would like to avoid the change of signs problem when EPS turns from negative to positive or vice versa.

Joseph's note for tutors (to be removed): From a more mathematically rigorous point of view, we calculate the metric value difference of two adjacent quarters as absolute difference if we think values of this metric follow a normal distribution whereas we should calculate the difference as %change instead if the metric values follow lognormal distribution. However I don't think such a consideration matters a lot to the machine learning model and instead has chosen to pay more attention the more practical problems such as the two points outlined above.

We've subsequently engineered the following three features related to Earnings Surprise:

- Current quarter's Earnings Surprise (reported EPS market estimated EPS);
- Difference between current quarter's Earnings Surprise and that of the previous quarter;
- Difference between current quarter's Earnings Surprise and the average Earnings surprise of the preceding three quarters;

Joseph's note to tutors (to be removed): The student has also prepared for Guidance data which represents how a company's management would expect the company to perform in the coming quarters. Guidance data is currently not used as model input because they are too sparse.

C. Price Movement data

The following data related to recent stock movements have been engineered as the model input features:

- %change of S&P500 index's price from 20 day prior to the release of a company's quarterly earnings to the index's price level at the close of trading the day before the announcement;

In our experiments we forecast a company's PEAD over 30 days but we do that on a number of different starting point, such as +1 or +2 days following an announcement. When the forecast starting point is not a stock's closing price the day prior to the announcement we also engineer the following additional features:

- %change from a stock's closing price a day prior to announcement to chosen starting point of the forecast;

- %change from the S&P500 index's closing price a day prior to a stock's quarterly earnings announcement to chosen starting point of the forecast;

We are including stock movements around the time as well as shortly after the release of earnings data so as to provide information to the learning models about the market's immediate reactions to the data release. We are including movement of the broad market (proxied by the S&P500 index) around the time as well as shortly after the release of earnings data of an individual company is to help paint a picture to the models on the health of the general market as well as the prevailing collective mood of the investor population.

It's important to note that a company's earnings announcement can be made before market opens, after market closes, or even during the day. Since the default starting point of our forecast on 30 day PEAD is a stock's closing price the day before the announcement, we have been very careful in deciding which day is the correct one day before' by using the announcement time information.

D. Momentum Indicators

We've chosen the following technical/momentum indicator values calculated on the same day an individual company's quarterly earnings data was released:

- 9-day Relative Strength Index (RSI)

- 30-day Relative Strength Index

- 5-day Moving Average / 50-day Moving Average

- 5-day Moving Average /200-day Moving Average

- 50-day Moving Average / 200-day Moving Average

We believe all these indicators should in a way measure how a stock's recent short term movements compare to its historical movements further back in time. The inclusion of momentum indicators is to allow the prediction process of future stock movements to take into account a stock's recent movement trend.

IV. DATA PRE-PROCESSING

We have by now finished collecting training data for each of the 386 companies at each of the historical earnings reporting quarters between 1996 and 2018. In order for them to be understood by the models we put them into a matrix-like data structure where each row represents an n-dimensional training data point, indexed by a company name and a historical quarter name, and each column holds data of the same feature from all the data points.

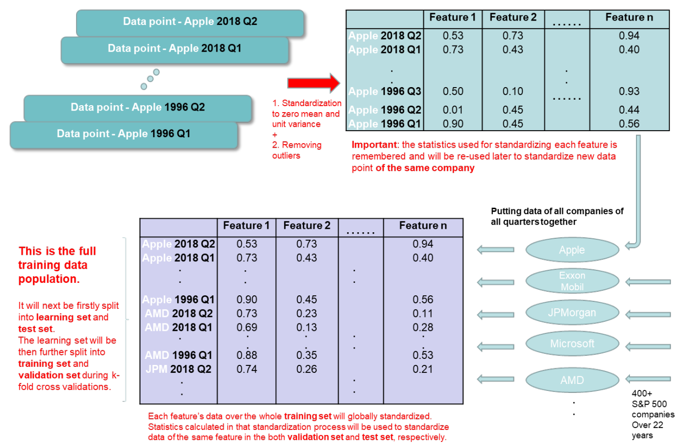

FIG. 1. Steps of Data Pre-processing

Before we put the data of all the companies and of all the quarters into a matrix, we pre-process each company's data to deal with outliers and to standardize data of every company. Firstly, we employ Winsorization [16] to reduce the number of outliers present in the input features. This is carried out on the feature data of each individual company. Secondly, we standardize a selective group of features of each company. Every company's standardized features will then be stacked back into a full training data set. The rationale of standardizing the features at the company level can be explained in the following example. Certain blue chip large cap companies, or certain companies from non-cyclical sectors may have little variation in some of their financial metrics from quarter to quarter whereas a lot of mid cap growth companies may see more volatile movements in the value of the same financial metrics. Both cases are considered norm for respective companies but when we put data of these companies together and put them under the same feature column, we can make them more comparable from the models' perspective once different company's data

Deep Neural Network	**XGBoost**	**Support Vector Machine**
Number of epochs	*Max depth*	*Kernel method*
Hidden layer neuron count	*Sub sample*	*Gamma*
Dropout rate	*Column sample by tree*	*C (model's penalty parameter)*
Regularization Lambda	*Gamma*	*Epsilon*
Learning rate	*Learning Rate*	
Hidden layer count	*Minimum child weight*	

TABLE II. Model hyperparameters optimized by GA + CV

have been standardized. The pre-processing process is illustrated in Figure 1.

V. MODELS AND METHODS

In order to forecast 30 day post earnings drift we've chosen to use a deep neural network, an Extreme Gradient Boosting mdoel, and support vector machines with different kernels.

Joseph's note to tutors: Mathematical descriptions of all these three models and their functioning principles will be given in the real paper if required. Also, a more in depth description of Genetic Algorithm will also be given in the real paper if required.

The whole training data population is split into training set and test set with the former consisting of data of each company's financial quarters up to Q4 2015 and the latter consisting of data between Q1 2016 to Q2 2018. The training set is used to tune the models with the help of Genetic Algorithm (GA) and cross validation (CV). Model tuning is one of the most important steps in ensuring the predicted outputs can meaningfully capture the underlying dynamics of the dependent variable. We have experimented a more straightforward approach of grid search on optimal hyperparameter sets and have found it less effective in its performance and incomplete in the search results. Genetic Algorithm as an adaptable and easily extensible heuristic optimization method has been chosen to perform model tuning on all the models under experiment. Table II gives the list of hyperparameters of every model that we've put through GA for tuning:

We would like to note that researchers in the literature typically focus on one or two kernel methods to go with the Support Vector Machine models. For instance Tay and Gao chosen Gaussian kernel with SVM to forecast financial time series [17] and Madge used Radial basis function (RBF) kernel in his attempt to forecast stock price movement [14]. Instead we've chosen 7 different kernels (including RBF, Sigmoid, Linear, and Polynomial of degrees 2 to 5) and use GA to optimize SVM's output accuracy out of all these kernels. This ensures we are not limited to a small number of common kernels like we've seen in the literature and instead we take full advantage of GA's optimization prowess to help us identify the best kernel and its accompanying model parameters for our SVM model. Similarly, when using multi-layer Neural Network researchers in the literature typically pre-fix the number of hidden layers or the number of neurons in each hidden layers for their models and only carry out model tuning on common hyperparameters. Again this practice can be subjected to sub-optimal model accuracy as the modeller has not included the model structure as part of the model optimization process and instead only focus on the hyper parameters. Recognising the deficiency of this model calibration process we are including the number of hidden layers and the number of neurons in each hidden layer as our tuning parameters effectively tuning both the Neural Network model structure as well as model hyperparameters.

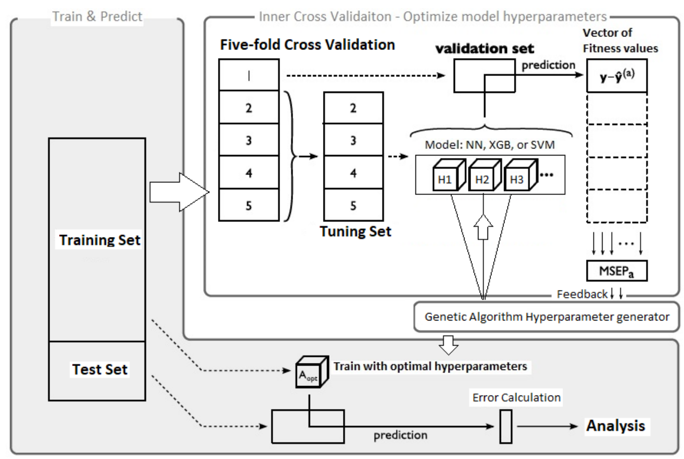

FIG. 2. Hyperparameter Tuning using GA + CV

When tuning a model each of the model's chosen parameters is randomly initialised according to the parameter's valid range of values. This initialization is repeated 40 times so that we have 40 combinations of randomly initialised model parameters to start the GA process. Each of the combinations is called a population and each hyperparameter within a combination is called a chromosome. All the 40 populations together are considered to be part of the current generation. The GA process carries out a 5-fold cross-validation routine using the hyperparameter values in each of the 40 populations and when finished, keep the 20 populations that have produced the smallest fitness value and these 20 populations are considered to have performed better in post-announcement drift prediction using the current model than the 20 discarded ones. The remaining 20 populations are then used to cross-breed into 20 new populations and in this process mutation is allowed to happen to the new populations, i.e. chromosomes in the 20 new populations are allowed to randomly change value. At the end of this process we have produced a new and better set of 40 populations and we call them the new generation. They are then

	MAE	RMSE
XGB	6.94%	1.03%
MLP	7.85%	1.23%
SVM	9.69%	1.74%

TABLE III. Prediction error metrics by models

	Classification Success Rate
XGB	58%
MLP	52%
SVM	50%

TABLE IV. Average classification success rate by models

fed through a second iteration of the GA process until eventually the minimum fitness value no longer changes within tolerance and at this point we've arrived at the optimal population which produces the smallest fitness value using the current model.

Figure 2 shows how Genetic Algorithm and Cross Validation work together to produce the set of hyperparameters of each model which result in the highest prediction accuracy on the validation set.

VI. RESULTS

Having prepared and engineered a wide range of feature data, we have carefully tuned a deep Neural Network (multilayer perceptron, MLP), an Extreme Gradient Boosting model, as well as Support Vector Machine (SVM) models of different kernels. We have chosen 16457 data points from Q1 1996 to Q4 2015 as our training set and carried out 30 day post-earnings stock price drift predictions on 3035 quarters belonging to 386 S&P500 companies from Q1 2015 to Q2 2018. Our results show that XGBoost has significantly outperformed both MLP and SVM both in the point-by-point prediction accuracy as well as in capturing significant risk-adjusted returns from the out-of-sample data. Table III shows the Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) on the 3035 out-of-sample data points by the three models. As our main benchmark, MAE by XGBoost is 12% better than that of MLP and 28% better than that of SVM on a RBF kernel (which is the best performing SVM scheme selected by the Genetic Algorithm scheme).

We've also examined the classification results by checking how successfully the models have predicted a positive/negative 30 day post-announcement return comparing against the actual stock return of the same data point. Similarly the XGBoost has produced better results by having a 58% success rate compared against MLP's 52%. SVM has fared worst with a 50% success rate. See table IV.

Joseph's note to tutors: I would like to say that 58% classification success rate is already very good in stock forecast but I don't know how to put this statement out appropriately. This is not helped by the presence in the literature of some lower quality paper which purport to have achieved 70%+ success rate but I find those very implausible.

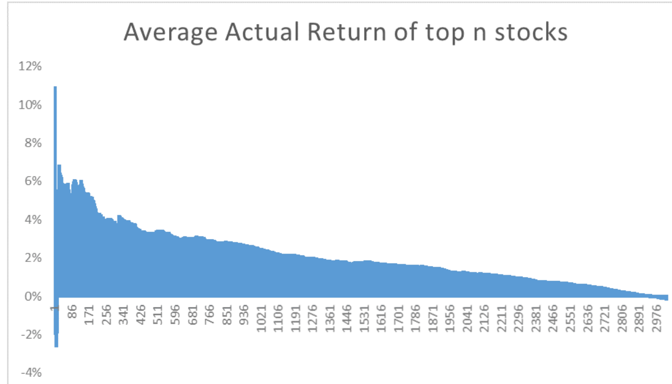

FIG. 3. Average Actual Return of the top n stocks

A 58% classification success rate would allow us to build a profitable portfolio of stocks using the buy-and-hold scheme [17]. However we would like not to focus on the profitability test and instead put our attention on a more important discovery from our experiments. Despite XGBoost's more superior performance against other tested models and a financially viable classification success rate of 58%, we've recognised that on a purely point-by-point basis, a mean absolute error of 6.9% in regression results can hardly be replied upon to perform accurate near term post-announcement price forecast on a single stock. On the other hand we've also recognised that since MAE is the mean of absolute errors it may not be the best error measure on the returns of an entire portfolio of stocks since movement of component stocks can offset each other within a portfolio. (*Joseph's note: I don't know if it's appropriate to include the two preceding sentences*) By analysing the average return of groups of test stocks, We've subsequently discovered that by using predicted return of the 3035 individual test stocks between Q1 2016 and Q2 2018 as a proxy, we've been able to rank the test stocks in a way that the average actual return of the top n stocks ($n \in [2, 3035]$) are almost always descending, and this is simply achieved by ranking the test stocks by their predicted returns from high to low. This phenomenon is best illustrated by Figure 3. In Figure 3, each point in the descending curve represents the average actual return of all the stocks up to the current point with the x-axis being index of the stocks, after all the test stocks have been ranked from high to low by the predicted 30 day post-earnings-announcement returns. This discovery indicates that we've successfully captured the relative positioning of individual stocks within the distribution of the 30 day post-announcement drifts from our test stocks. *(Joseph's note: is the preceding sentence an accurate way of describing this discovery?)* Although we have not yet been able to rank individual test stock's return from high to low, we've been able to rank these stocks in a way that the average actual return of the top n stocks are almost always going from high to low as n increases.

Another way to look at how the test stocks are ranked by predicted returns is by looking at the average return of moving portfolios of 100 stocks. Again the test stocks have now been ranked from high to low according to their predicted returns. On Figure 4 each point on the graph represents the average actual return of the preceding 100 stocks and the graph is trending downward. This observation again shows that the predicted returns have correctly captured the rankings within the 3035 test stocks. This observation is further backed up by how correlated the test stocks' predicted returns are with the average actual return of the top n test stocks (once they are ranked by the predicted returns). We've run 20 rounds of predictions using XGBoost on the same training and test set and these two series of data have achieved an average R-square value as high as 0.7. This result consistently shows that we can form the highest-returning portfolios out of the sample population for buy-and-hold trading strategies and the lowest-returning portfolios for short selling strategies.

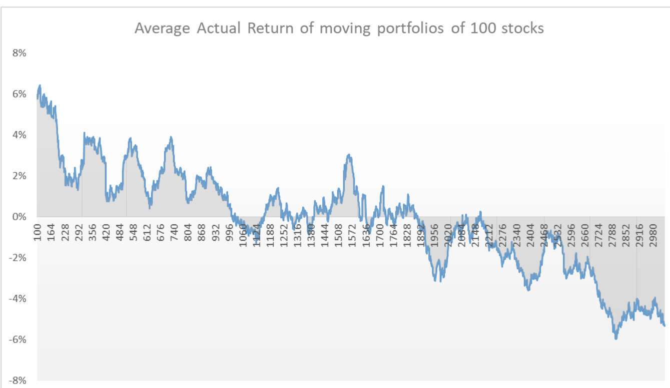

FIG. 4. Average Actual Return of moving portfolios of 100 stocks

Further more, we've recognised that from practicality's point of view one can not form a 30 day buy-and-hold strategy on stocks that are available to buy only at different years, i.e. despite the seemingly significant results from our 3035 test stocks, these test stocks were sourced over two-and-a-half-year time horizon between Q1 2016 and Q2 2018 and hence we would not be able to practically form a portfolio from the top 100 stocks that is shown to have significantly higher risk-adjusted 30 day post-announcement returns than the bottom 100. As a result we've also looked at the average stock returns of only those test points from Q2 2018 after the stocks have been ranked by their predicted returns from high to low in Figure 5. The results are fairly consistent with those from the entire test set.

Lastly, in an attempt to explain our observations, we've looked at the test stocks more closely. We've discovered that although XGBoost's overall classification success rate on positive/negative returns stands at 58%,

FIG. 5. 100 stock window size - Q2 2018 only

the success rate for test stocks whose actual 30 day returns are positive is as high as 72%. This new information indicates that when we rank the stocks by their predicted returns, we've more correctly placed positive-returning stocks at the upper end of the ranked series than the lower end and that echos with our earlier observations. However this observation doesn't explain why XGBoost has been able to 'rank' the stocks given that there is a clear descending trend in the graph with average returns of moving portfolios of 100 stocks.

VII. CONCLUSION

To be written. There are more work to do and the eventual conclusions may well change.

BIBLIOGRAPHY

[1] Gocken et al. "Integrating metaheuristics and artificial neural networks for improved stock rice prediction". In: *Expert Systems With Applications* 44 (2016).

[2] R. Ball and P Brown. "An Empirical Evaluation of Accounting Income Numbers". In: *Journal of Accounting Research* (1968), pp. 159–78.

[3] E. Fama and K. French. "Common Risk Factors in the Returns on Stocks and Bonds". In: *Journal of Financial Economics* 22 (1993).

[4] R. Bhushan. "An informational efficiency perspective on the post-eanings announcement drift". In: *Journal o f Accounting and Economics* 18 (1994), pp. 45–65.

[5] Luke Qiu. *Earnings Announcement and Abnormal Return of S&P 500 Companies*. 2014. DOI: `https://openscholarship.wustl.edu/wushta_spr2014/74`.

[6] Erhan Beyaz et al. "Comparing Technical and Fundamental indicators in stock price forecasting". In: *IEEE 4th International Conference on Data Science and Systems* (June 2018), pp. 1607–1613.

[7] Michael E. Bradbury. "Voluntary Semiannual Earnings Disclosures, Earnings Volatility, Unexpected Earnings, and Firm Size". In: *Journal o f Accounting Research* 30 (Spring 1992).

[8] G. Foster, C. Olsen, and T. Shevlin. “Earnings releases, anomalies, and the behavior of security returns”. In: *The Accounting Review* 59 (1984).

[9] Burton G. Malkiel. “Models of stock market predictability”. In: *Journal of Financial Research* 27.4 (2004), pp. 449–459.

[10] Dennis Olson and Charles Mossman. “Neural network forecasts of Canadian stock returns using accounting ratios”. In: *International Journal of Forecasting* 19(3) (2003), pp. 453–465.

[11] Alaa F. Sheta, Sara Elsir M. Ahmed, and Hossam Faris. “A comparison between regression, artificial neural networks and support vector machines for predicting stock market index”. In: *Soft Computing* 8 (2015).

[12] Reza Hafezi, Jamal Shahrabi, and Esmaeil Hadavandi. “A batneural network multi-agent system (BNNMAS) for stock price prediction: Case study of DAX stock price”. In: *Applied Soft Computing* 29 (2015), pp. 196–210.

[13] Zhang X., Li A., and Pan R. “Stock trend prediction based on a new status box method and AdaBoost probabilistic support vector machine”. In: *Applied Soft Computing* 49 (2016), pp. 385–398.

[14] S. Madge. *Predicting Stock Price Direction using Support Vector Machines*. Independent Work Report. 2015.

[15] Berislav Bolfek, Milan Stanifffdfffd, and Sanja Knefffdfffdevifffdfffd. “Vertical and Horizontal Financial Statement Analysis”. In: *Ekonomski Vjesnik* XXV.1 (2012).

[16] Bin Duan and William P. Dunlap. *The Robustness of Trimming and Winsorization When the Population Distribution Is Skewed*. ProQuest Dissertations and Theses. 1998.

[17] Jui-Sheng Chou and Thi-Kha Nguyen. “Forward Forecast of Stock Price Using Sliding-Window Metaheuristic-Optimized Machine-Learning Regression”. In: *IEEE Transactions on Industrial Informatics* 14 (July 2018), No.7.