Deep Learning Post-Earnings-Announcement Drift

Zhengxin Joseph Ye
GLAM, Department of Computing
Imperial College London
London, UK
z.ye18@imperial.ac.uk

Björn W. Schuller
GLAM, Department of Computing
Imperial College London
London, UK
bjoern.schuller@imperial.ac.uk

***Abstract*—Post-Earnings-Announcement Drift (PEAD) has traditionally been studied using regression models in the literature which often involve smaller data sets and smaller groups of factors whose analysis results tend to be more linear in nature. In this paper, we explore using machine learning models to overcome those limitations and aim to find an optimal supervised model in forecasting drift direction following an earnings release. We test a deep neural network (DNN), an extreme gradient boosting model (XGB) as well as support vector machines (SVM) with different kernels and use a long list of carefully prepared and engineered input features including data from quarterly earnings reports from 1 106 companies in the Russell 1000 index between 1997 and 2018. We find that XGB performs marginally better than the considered DNN and both are significantly better than the SVM variants. We use both Cochran's Q Test and McNemar's Test to prove that our findings are statistically meaningful. We also find that movement of stocks in different industrial sectors respond differently to the same factors when using the same models and provided analysis on that.**

I. Introduction

Post-Earnings-Announcement Drift, or PEAD, is a market phenomenon that was first studied by Ball and Brown [1] who noted that a stock's abnormal returns would drift in the direction as indicated by the perceived quality of reported financial results of the company. While PEAD has garnered much attention in earlier literature, we see limitations in regression based approaches widely adopted in those studies, especially in the way companies are pre-selected by *a priori* conditions. When measuring the effect of stock earnings on abnormal returns, Qiu [2] would separately pool companies with positive and negative earnings surprises before conducting regression analysis on multiple factors against the level of returns. Similarly Kim [3] sought to pre-group different portfolios by different characteristics of the factors under analysis and tried to analyse and determine the relationship between respective portfolios' return and the corresponding economic factors that segregated the portfolios. We believe that stock markets do not just react symmetrically to negative and positive earnings surprises nor linearly to a static set of factors. We further believe it is important that we study the combination of a large set of drivers that may impact the near term excess returns of a stock following an earnings event. To achieve that, we introduce machine learning to the study of PEAD dynamics and aim to capture those nonlinear combination of drivers that were not well studied in regression methods.

To begin with, we attempt to overcome constraints commonly seen in earlier researches on PEAD: we adopt a much larger input feature space which includes financial report data, earnings surprise data, near term momentum indicator data and short interest data and both raw data and engineered data are considered; we do not segregate companies by a static list of pre-determined attributes (*subsample analysis*) [4]; our company universe includes every company that was a constituent of the Russell 1000 index between 1997 and 2018 and is a larger set than those in most prior PEAD studies of similar nature.

To deploy machine learning to serve the research goals, we have chosen three types of supervised learning methods to work through the high noises embedded in the price and signal data: a deep neural network (DNN), Support Vector Machines (SVMs) with different kernels, and an eXtreme Gradient Boosting (XGB) model. We chronically separate the raw data into in-sample and out-of-sample time frames whose lengths vary depending on the particular test scenario. We use the in-sample training data to optimise a model's hyperparameters. Having tried and ruled out the grid search method as inexhaustive and slow, we have chosen to use the highly adaptable *Genetic Algorithm* (GA) to tune our models as seen in the work of Deng et al. [5], and use a broad value range and a small granular step for each of the hyperparameters. We employ a 5x2-fold cross validation (CV) within each GA iteration for estimating the optimal combination of each model's hyperparameters. Our results identify both XGB and DNNs as capable of producing meaningful accuracy in forecasting the direction of 30-day cumulative abnormal stock movement following an earnings release when we examine their performances every year between 2015 and 2018 with XGB being marginally better than a DNN. SVM, however, perform significantly worse on the same tasks. In our studies, we not only collectively look at all the stocks in each test time framework, we also delve into stocks that belong to specific industrial sectors. We discover that, while working with the same feature space, the same models produce different levels of accuracy for different sectors. This can be seen as a proof that movement of stocks from different industries is subject to different drivers which we have produced an analysis of. Finally, we carry out the Cochran's Q test and McNemar's test to reject the null hypothesis and to prove that the differences in model performance as we identified

are statistically meaningful.

II. Related Work

Ball and Brown [1] first studied Post-Earnings-Announcement Drift as a stock market anomaly and observed the possibility to predict return for up to two months following the event of annual earnings announcements. Bhushan suggests that the market's delayed response to earnings can be explained by the existence of sophisticated and unsophisticated investors, transaction costs, and economies of scale in money management [6]. Ayes, Li, and Yeung [7] examined attributions to distinct PEADs on the same news and found via regression analysis that distinct behaviours by traders of different levels of sophistication as well as different trade sizes could explain distinct drift movements observed in response to the same earning results. The work by Bhushan and Ayes et al. in particular has painted a picture of inherent complexity in the way markets interpret earning results, adding to the challenges of trading on such an economic event and prompting us to use supervised learning models as nonlinear tools to assist in our analysis.

The literature has in recent years seen a lot of focus on stock price forecasting by machine learning with some paying special attention to fundamental metrics sourced from earnings reports. Olson and Mossman [8] studied 2 352 Canadian stocks and used 61 financial ratios with their models. In the task of forecasting 12-month returns, they observed that a artificial neural network outperformed traditional regression methods. They showed that fundamental metrics helped them achieve excessive risk-adjusted returns. Other studies went beyond fundamental metrics and involved more financial data. Working with 578 NASDAQ stocks, Namdari and Li [9] picked 12 financial metrics via feature selection and a group of technical signals as inputs to a Multi Layer Peceptron model. They were able to achieve the best accuracy of 65.87 % in predicting stock movement directions when combining fundamental and technical data in the model input space.

In addition to MLP, SVM and decision tree based boosting models such as XGB are also popular choices in the literature. Zhang [10] constructed a novel ensemble method integrated with the AdaBoost algorithm, probabilistic SVM and GA. He showed the new ensemble method achieved preferable profit in the simulation of stock investment using 20 shares from the SZSE and 16 stocks from NASDAQ. Madge [12] used the daily closing price 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. When attempting to predict future stock price movements, they found little predictive ability in the short run but definite predictive prowess in the long-run. Part of the methods by Chatzis et al. [13] to evaluate the possibility of a future global market crash is by forecasting 1-day and 20-day stock market returns. A vast set of data from global stock, bond, and FX markets were used. They tested Logistic Regression, SVMs, Random Forest, DNNs, and XGB and declared the superiority of XGB over others by examining the forecast results through a list of statistical measurement metrics.

The abundance of model choices makes it imperative to evaluate and compare the performance of models. Sheta, et al. [14] studied the performance of ANN, SVM, and Multiple Linear Regression in the prediction of the S&P500 market index. Using macro economic indicators and 27 technical indicators in their experiments, they observed that SVM generated comparatively better results than the other models tested. Hsu et al. used both ANN and SVM to study the influence on the predictability of a financial market and the feasibility of profitable model-based trading by the maturity of the market, the forecasting method employed, the horizon for which it generates predictions, and the methodology used to assess the model and simulate model-based trading [15].

Finally it is worth pointing out that the proliferation of machine learning in financial studies did not always come with enough attention on the generalisation quality of data. For example, Bradbury [16] used a small group of 172 firms to research the relationships among unexpected earnings, earnings volatility, firm size, and voluntary semi-annual earnings disclosures. Beyaz et al. [17] used both fundamental analysis and technical analysis to forecast stock prices six months and a year into the future using data from 140 S&P500 companies. Such limitations have been well addressed in this paper given the large amount of company data we are using.

III. Feature Generation and Data Pre-processing

The input feature space to our models consists of four groups of data all of which are sourced from Bloomberg: *Financial statements data, Earnings Surprise data, Momentum Indicator data, Short Interest data.*

Each quarterly earnings publication of a company is considered as a data point. We only include those data points where the full set of data in the chosen feature space is available on Bloomberg and discarded data points which suffered badly with missing data in their earnings reports. Since we needed to know if an earnings report was published before market open, after market close or during trading hours, we also discarded the data points which did not have explicit information on the publication time of day. This leaves us with close to 50 000 data points in our final population.

Table I lists 29 metrics from financial reports chosen by us. We have further created engineered features based on quarterly change and yearly change of each earnings metric.

New features are engineered using Earnings Surprise data, as we wanted to develop more granular measurement of the impact by Earnings Per Share (EPS). Similarly, new features are created using the raw RSI and Moving Average indicators so that we allow the models to account for any stock's recent movement momentum. This is, because information leakage could happen before earnings come out and could have been traded on in the markets. Finally short interest ratio is included as it provides good indication of how heavily shorted a stock is compared to its trading volume prior to an earnings event.

Cash	Operating Margin
Cash from Operating Activities	Price to Book Ratios
Cost of Revenue	Price to Cashflow Ratios
Current Ratio	Price to Sales Ratios
Dividend Payout Ratio	Quick Ratio
Dividend Yield	Return On Assets
Free Cash Flow	Return On Common Equity
Gross Profit	Revenue
Income from Continued Operations	Short Term Debt
Inventory Turnover	Total Asset
Net Debt to EBIT	Total Asset
Net Income	Total Debt to Total Assets
Operating Expenses	Total Debt to Total Equity
Operating Income	Total Inventory
	Total Liabilities

TABLE I
FINANCIAL REPORT METRICS CHOSEN AS INPUT FEATURES WHOSE QUARTERLY AND YEARLY PERCENTAGE CHANGES ARE ALSO INCLUDED IN THE INPUT SPACE

DNN	**XGB**	**SVM**
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)
Regularisation Lambda	Gamma	Epsilon
Learning rate	Learning Rate	
Hidden layer count	Minimum child weight	

TABLE II
MODEL HYPERPARAMETERS OPTIMISED BY GA + 5X2 CV FOR DNN, XGB AND XVM

Next we perform data pre-processing on the raw data to minimise the impact of outliers as well as to standardise data across the companies. First, Winsorization [18] is employed for outlier reduction. In this process, we dynamically set the upper and lower bounds for each company so as to preserve the original data structure as much as possible. Second, a selective group of features are standardised. Standardisation is only carried out on the training data set, and the statistics are re-applied to the testing set.

IV. MODELS AND METHODS

As outlined, our model candidates to forecast post earnings price drift include a DNN, an XGB, and SVMs with different kernels. Deep learning has been a cornerstone in machine learning for the last decade and we have chosen it as the benchmark model for comparison. Training a neural network is a convex optimisation problem with the loss function defined as:

$$L(\omega) \triangleq \sum_{i=1}^{M} L_i(\omega) \tag{1}$$

Where $L_i(\omega)$ is a loss function for data point $i \in \{1, 2, ..., M\}$ and ω are the model weights being optimised. A neural network is a nonlinear system due to the presence of an activation function on each neuron (except for the output neuron as it can be linear). An activation function $\sigma(x)$ can take a lot of forms such as sigmoid, tanh, or Rectified Linear Unit (ReLU), and it makes the output of a neuron appear as $f(x) = \sigma(\omega^T X + b)$, with X being inputs to the neuron and b being the bias. To minimise the loss function, we use batch Gradient Descend to optimise model weights iteratively by using the learning rate α and the Jacobian matrix of derivatives of the loss function with respect to all the model weights $\nabla L(\omega) = \frac{\partial L}{\partial \omega_1}, \frac{\partial L}{\partial \omega_2}, ..., \frac{\partial}{\partial \omega_N}$:

$$\omega_i := \omega_i - \alpha \nabla L(\omega). \tag{2}$$

SVMs were first invented by Vladimir Vapnik and his colleagues in 1963 with its current standard form ϵ-SVM proposed by Cortes and Vapnik in 1995 [19]. Unlike regression based methods which aim at minimising the error function, SVM finds a hypothesis function $f(x)$, which represents a hyperplane in the input feature space whose prediction output $\widehat{y_i}$ deviates away from the actually observed value y_i by at most ϵ. In our experiments we have tested the linear kernel, the sigmoid kernel, the Radial Basis Function (RBF) kernel as well as the polynomial kernels with degrees of 2, 3, 4 and 5.

Extreme Gradient Boosting or XGBoost, invented by Tianqi Chen [22], is a scalable tree boosting supervised learning model. A number of features such as cache awareness, or weight awareness and sparsity awareness [20] [21] give it standout performance in many applications. The former provides fast access to gradient statistics in memory and the latter two allows the algorithms to handle weighted data and sparse data very well [22]. A regularised learning objective is introduced within a tree structure to reduce overfitting in the process of branch pruning and tree splitting.

XGB utilises independent regression trees represented by hypothesis f_k. The leaf scores by each tree structure help form the decision rules so as to classify each set of inputs x_i into leaves and the final predicted output is calculated by summing up the scores in the leaves concerned:

$$\widetilde{y_i} = \phi(x_i) = \sum_{k=1}^{K} f_k(x_i), f_k \in \mathbb{F}, \tag{3}$$

where x_i is a data set in matrix form $A \in M_{m \times n}(\mathbb{R})$ with m data points and n features and $\mathbb{F}$ is the space of regression trees. The scoring function for leaf splitting is

$$L_{split} = \frac{1}{2}\left[\frac{(\sum_{i \in I_L} g_i)^2}{\sum_{i \in I_L} h_i + \lambda} + \frac{(\sum_{i \in I_R} g_i)^2}{\sum_{i \in I_R} h_i + \lambda} + \frac{(\sum_{i \in I} g_i)^2}{\sum_{i \in I} h_i + \lambda}\right] - \gamma. \tag{4}$$

A method called the *exact greedy algorithm* uses the scores to enumerate all the possible splits for continuous features, optimising each level of a tree and minimising the overall loss function along the way.

V. HYPERPARAMETER OPTIMISATION

GA as an adaptable and easily extensible heuristic optimisation method has been chosen to perform model tuning on

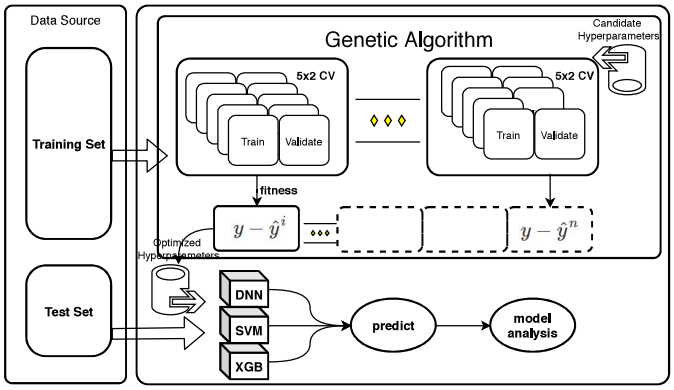

Fig. 1. Hyperparameter Tuning using GA + CV

All Sectors (accuracy%)	2018	2017	2016	2015
XGB	56.75	58.32	61.09	59.46
SVM	50.43	50.86	53.25	53.82
DNN	56.69	57.63	60.02	58.31

TABLE III
AVERAGED CLASSIFICATION SUCCESS RATES OF 100 RUNS FROM 2015 TO 2018 USING GA OPTIMISED DNN, XGB AND SVM. THE SUCCESS RATE IS BASED ON THE ACCURACY OF A MODEL CORRECTLY PREDICTING THE MOVEMENT DIRECTION (UP OR DOWN) OF A STOCK'S 30-DAY PEAD FOLLOWING AN EARNING EVENT.

all the selected models under experiment. For SVM we see researchers in the literature typically go with a small handful of kernel methods. For instance Tay and Gao chose a Gaussian kernel with SVM to forecast financial time series [11] and Madge used Radial basis function (RBF) kernel in his attempt to forecast stock price movement [12]. Instead we have chosen 7 different kernels (RBF, Sigmoid, linear, and polynomial of degrees 2, 3, 4 and 5) and use GA to optimise the SVMs' output accuracy out of all these kernels. This ensures we are not limited to a small number of common kernels as seen in the literature and instead take full advantage of GA's optimisation prowess to help us identify the best kernel and its accompanying model parameters for our SVM model. Similarly, when working with DNN, researchers in the literature often prefix the number of hidden layers or the number of neurons in each hidden layer for their models and only carry out model tuning on common hyperparameters such as dropout rate and learning rate. 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 optimisation process and instead only focuses on the hyper parameters of a predefined structure. 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 optimisation hyperparameters, effectively tuning the neural network model structure. We have given both the hidden layer count and neuron count in each layer a large enough range so that a DNN can go deeper if the GA optimisation finds it necessary. Table II gives the list of hyperparameters of every model that we have put through a GA for tuning.

Forty groups of initial values, which together is considered as a *generation*, are randomly generated for each model's hyperparameter set. Each group is called a *population* and each hyperparameter within a group is a *chromosome*. The GA process fits a model using each *population* of hyperparameters and validates the quality of a fitted model by calculating a fitness value. Populations which produce high fitness values are replaced with new populations of values through methods of *cross-breading* and *mutation*. This process is iterated through 20 generations and the final hyperparameter values are chosen as those that yield the lowest fitness value. The most important component of the GA process is the 5x2-fold cross validation which has been chosen over a simple k-fold CV as recommended by Dietterich [23]. This is to ensure each data point appears only in the training or validation dataset for a single estimate of fitness and hence, the two datasets are totally independent. To do it, we randomly split the data set reserved for hyperparameter tuning into two halves. Data in the first half is standardised before being used to train the model. Data in the second half is used for validation and is also standardised, but by using the same statistics generated in the standardisation process of the the first half's data. Such a train-and-validate process is repeated 5 times and the fitness value is the average of the 5 iterations. Figure 1 illustrates how GAs works together with cross validation to produce the best set of hyperparameters which results in the highest classification rate (smallest fitness value) over the validation set.

VI. RESULTS

A. Predicting Drift Direction

We focus our study on a stock's 30-day post earnings *Cumulative Abnormal Return* (CAR). We set our model output as the drift direction turning our research into a classification problem. The cumulative abnormal return from T_1 to T_n for stock i, is defined below.

$$CAR_i(T_1, T_n) = \sum_{t=T_1}^{T_n} (AR_i(t)) = \sum_{t=T_1}^{T_n} (r_i(t) - E(r_i(t))), \tag{5}$$

where $r_i(t)$ is the actual one-day stock return, $E(r_i(t))$ is the expected one-day return of stock i. Each stock's expected return is calculated using the Capital Asset Pricing Model (CAPM) model [24]:

$$E(r_i) = r_f + \beta_i(E(r_m) - r_f), \tag{6}$$

where r_f is the risk free rate, $beta_i$ is a company's systematic risk, and $E(r_m)$ is the market expected return. Historical 10Y U. S. Treasury Yield which estimates r_f, S&P500 index return which estimates $E(r_m)$, as well as each company's historical beta at the time of each earnings reporting are all sourced from Bloomberg.

In the first group of tests, we perform forecast on PEAD direction following all the earnings events in every year between 2015 and 2018. Hyperparameters are optimised for

	2018			2017			2016			2015			Average		
(accuracy%)	XGB	SVM	DNN	XGB	SVM	DNN	XGB	SVM	DNN	XGB	SVM	DNN	XGB	SVM	DNN
Industrial	62.92	52.09	61.14	58.51	48.62	57.20	59.04	59.72	60.03	62.27	57.17	60.12	60.69	54.40	59.62
Basic Materials	56.69	51.01	54.91	59.91	57.04	56.76	54.95	57.14	54.21	61.06	52.14	57.09	58.15	54.33	55.74
Cosumer Cyclical	56.87	51.23	58.31	60.30	54.49	58.83	58.39	49.84	55.74	59.06	49.84	60.67	58.65	52.53	58.39
Consumer Non-Cyclical	57.43	53.00	58.41	62.20	50.91	60.67	60.28	50.91	58.72	59.79	50.91	60.62	59.93	53.82	59.61
Financial	55.37	48.46	53.52	53.32	51.01	53.30	60.12	53.73	59.14	56.34	48.69	51.82	56.29	50.47	54.45
Technology	55.91	55.60	58.54	54.83	55.74	56.35	58.00	59.64	60.78	61.01	45.51	59.13	57.44	54.12	58.70
Communications	53.22	47.92	51.78	53.87	53.81	47.71	57.50	51.95	54.30	57.41	46.35	50.56	55.50	50.01	51.08
Energy	53.61	44.50	48.45	51.01	47.51	47.00	52.49	52.35	46.96	55.49	45.66	51.37	53.15	47.50	48.45
Utilities	49.92	47.47	49.15	49.95	51.77	42.06	46.74	46.78	49.70	46.19	42.73	47.12	48.20	47.19	47.01

TABLE IV

AVERAGED CLASSIFICATION SUCCESS RATES OF 100 RUNS FROM 2015 TO 2018 USING GA OPTIMISED XGB, SVM, AND DNN ON STOCKS FROM INDIVIDUAL INDUSTRIAL SECTORS. THE SUCCESS RATE IS BASED ON THE ACCURACY OF A MODEL CORRECTLY PREDICTING THE MOVEMENT DIRECTION (UP OR DOWN) OF A STOCK'S 30-DAY PEAD FOLLOWING AN EARNING EVENT.

2018			2017			2016			2015			Average		
XGB	SVM	DNN	XGB	SVM	DNN	XGB	SVM	DNN	XGB	SVM	DNN	XGB	SVM	DNN
6	0	3	7	1	1	5	1	3	6	0	3	8	0	1

TABLE V

NUMBER OF SECTORS WHERE THE HIGHEST OUT-OF-SAMPLE PREDICTION ACCURACY IS OBSERVED UNDER A MODEL

Test Year	Q value	P value
2015	27.52	1.05E-06
2016	48.22	3.38E-11
2017	88.21	7.00E-20
2018	56.69	4.88E-13

TABLE VI

TEST STATISTICS FROM COCHRAN'S Q TEST ON WHETHER THERE IS STATISTICAL DIFFERENCE AMONG THE CLASSIFICATION RESULTS BY DNN, XGB AND SVM. THE RESULTS SHOW THAT IN EVERY ONE OF THE CHOSEN TEST YEARS, THE CALCULATED P VALUE IS MUCH SMALLER THAN A CHOSEN ALPHA VALUE OF 0.05 WHICH DEFINES THE SIGNIFICANCE LEVEL, ALLOWING US TO REJECT THE NULL HYPOTHESIS AND PROVING THERE IS STATISTICAL SIGNIFICANCE IN THE DIFFERENCES BETWEEN TEST RESULTS BY THE THREE MODELS.

Test Year	DNN vs XGB	DNN vs SVM	XGB vs SVM
2015	0.60854777	6.59E-05	7.00E-05
2016	0.43330609	3.86E-09	1.57E-06
2017	0.67178157	1.25E-14	3.03E-12
2018	0.13876126	8.11E-09	3.85E-09

TABLE VII

TEST STATISTICS FROM MCNEMAR'S TEST. USING A SIGNIFICANCE LEVEL OF 0.05 (ALPHA), THE CALCULATED STATISTICS IN DNN VS SVM AND XGB VS SVM ARE MUCH SMALLER THAN THE ALPHA VALUE, ALLOWING US TO REJECT THE NULL HYPOTHESIS AND HENCE PROVING THAT THE UNDERPERFORMANCE OF SVM IN THESE TASKS AGAINST XGB AND DNN IS STATISTICALLY SIGNIFICANT. HOWEVER, WE ARE NOT ABLE TO REJECT THE NULL HYPOTHESIS IN DNN VS XGB AND HENCE CAN NOT STATISTICALLY DISTINGUISH THE PERFORMANCE BETWEEN DNN AND XGB.

each of the DNN, SVM, and XGB models using data preceding the year under test. Each testing set contains all the relevant companies that filed for quarterly earnings with U. S. Securities and Exchange Commission (SEC) in the testing year. Most companies filed four times and each is considered an independent data point in the data set. All the earnings reported prior to the test year are included in the training set.

We adjust for imbalanced classes for all the models before using the data. Our model output is a binary movement, i. e., positive or negative cumulative abnormal stock return 30 days after an earnings release. The number of ups and downs are not equal in our data set, and hence, the need to adjust for this imbalance arises. Under each test scenario, sample weight adjustments are first calculated for the DNN and XGB based on the full training data set which is then used for class imbalance adjustment during testing. For SVM, no prior weight adjustment calculation is needed, because the SVM library can internally 'balance' the given data when instructed.

In each experiment, 100 tests are run using the same training and testing set and the average classification accuracy on the out-of-sample test set is calculated. Table III presents the averaged prediction accuracy of the three models when including stocks from all the sectors. As evidenced by this result, XGB and DNN outperform the SVM by a meaningful margin. XGB performs better than the DNN most of the time, although the margin of difference is a lot smaller than that between SVM and DNN or XGB.

To ensure our findings are not by statistical chance, we compute the statistical significance of our results. Since we have three candidate models, we are conducting a *multiple hypotheses testing*. Inspired by the work by Raschka [25], our first step is to conduct an omnibus test under the null hypothesis that there is no statistical difference between the classification accuracies. Omnibus tests are statistical tests designed to check whether random samples depart from a null hypothesis. We have chosen the non-parametric Cochran's Q test for this task, since it works well with related categories where the response is binary. As in [25], we produce an $n \times M$ matrix, where n is the number of test examples, and M the number of classifiers. The entry *ij*th of such a matrix is 0 if a data example x_i by the classifier C_j is incorrect, otherwise 1. Our results are presented in table VI. We use a significance level of $\alpha = 0.05$ and successfully reject the null hypothesis that the proportion of 'successes' is the same in all groups.

Next, we conduct pairwise tests using *McNemar's test* which was proven by Dietterich [23] to perform well in

comparing two models on an independent test set with binary response which produces one of the lowest type I error. We use the mlxtend library written by Raschka [26] with results given in table VII. Again, using a 0.05 significance level, we are able to reject the null hypothesis between XGB and SVM as well as DNN vs SVM, proving that the underperformance of SVM in these tasks against XGB and DNN is statistically significant. However, we are not able to reject the null hypothesis in DNN vs XGB.

B. Drift Analysis at Sector Level

We run the same tests on stocks that belong to each of the nine industrial sectors *Industrial, Basic Materials, Consumer Cyclical, Consumer Non-Cyclical, Financial, Technology, Communications, Energy, and Utilities* as seen on Bloomberg, so as to give a more granular view on each model's performance on data sets of diffing underlying characteristics.

Table IV presents the grid of results under the three models on each sector between 2015 and 2018. Each number on the grid is the classification success rate on the drift direction of out-of-sample stocks in a particular sector. Table V summarises the number of sectors, where the highest out-of-sample prediction accuracy is observed under a model. Consistent with the results presented in the previous section, we see XGB produces the highest classification success rates, whereas SVM is considerably poorer in this task. This is especially the case in sectors where the models generally have higher success rates such as in the Consumer Cyclical and Consumer Non-Cyclical sectors.

It is interesting to observe that the same models produce excellent results with out-of-sample stocks from most industrial sectors, but not with the likes of *Energy* and *Utilities*. This is clear evidence that our data is more impactful to some sectors than others. We provide two explanations for this observation. First, it is evident that other forms of data would have produced meaningful impact on stock's post-earnings-release movements should we have included them in our feature space. We have identified a variety of data which includes management's guidance, recent revisions of analysts' price fore-cast, other text information carried in financial reports, and meeting minutes with analysts amongst others, as further data to consider in future studies. Second, there is not a static combination of features driving the movement of stocks from different sectors following their earnings release. The driving forces are dynamic, and the implicit weighting of the same group of features are evidently different for different sectors. XGB and in a way DNN have demonstrated their prowess in our findings to produce a high degree of accuracy in forecasting PEAD direction when the feature data carries enough signals in them, despite the dynamism in the driving factors, which prompts us to believe that more consistent results would be achievable across all the sectors should we improve upon signal extraction from the features available. In that sense, combining our chosen supervised learning methods with representation learning [27] or data augmentation appears promising in the next stage's research.

VII. Conclusion

Post-Earnings-Announcement Drift (PEAD) as a stock market anomaly has traditionally been studied using regression based approaches which often focus on a limited list of factors under study. Results of such studies are linear and tend to focus on explaining the impactfulness of individual factors. Also, most studies do not tend to use a large data set and sometimes use *a priori* assumptions to segregate companies into groups prior to analysis. Attempting to plug this gap in the literature, we use supervised learning models to provide a nonlinear analysis on PEAD and prove effectiveness of machine learning models in predicting the direction of Cumulative Abnormal Returns (CAR) following an earnings event. In this process, we selected to use a much bigger set of numerical features including financial report data, earnings surprise data, near term momentum data, and short interest ratios with some features specifically engineered, all of which have been sourced over a long time frame of twenty-one years. Our results are two fold. First, we demonstrated that when properly configured using a Genetic Algorithm, supervised learning models combined with our selected input features are genuinely capturing the dynamics that drive the direction of PEAD with an out-of-sample classification success rate up to 62.9 % depending on the test scenarios. Second, we studied some of the most popular supervised models, DNN, XGB, and SVM in the same series of studies. Backed up by results by Cochran's Q Test and McNemar's Test, we illustrated how significantly better XGB and DNN have performed than SVM did when meansured by the classification accuracy, although the slightly superior results of XGB over DNN as we observed are not understood as statistically significant enough. We further observed that the current set of features provided a varying degree of driving force over the PEAD movement of stocks from different industrial sectors and at a different time.

This gives us a future research direction where one should aim to put more emphasis on both capturing and taking advantage of more forms of drivers in the feature space as well as working with representation learning and data augmentation techniques.

References

[1] Ray Ball and Philip Brown, "An Empirical Evaluation of Accounting Income Numbers", Journal of Accounting Research , vol.6, 1968, pp 159–78.

[2] Luke Qiu, "Earnings Announcement and Abnormal Return of S&P 500 Companies", 2014.

[3] Dongcheol Kim and Myungsun Kim, "A Multifactor Explanation of Post-Earnings Announcement Drift", The Journal of Financial and Quantitative Analysis, vol.38, 2003, pp.383–398.

[4] H. Kent Baker, Yang Ni, Samir Saadi, and Hui Zhu. "Competitive earnings news and post-earnings announcement drift", International Review of Financial Analysis, vol.63, 2016, pp.331–343.

[5] Shangkun Deng, Kazuki Yoshiyama, Takashi Mitsubuchi, andAkito Sakurai, "Hybrid Method of Multiple Kernel Learning and Genetic Algorithm for Forecasting Short-Term Foreign Exchange Rates", Computational Economics, vol.45.1, 2015, pp.49–89.

[6] Ravi Bhushan, "An informational efficiency perspective on the post-eanings announcement drift", Journal o f Accounting and Economics, vol.18, 1994, pp.45–65.

[7] Benjamin Ayers, Li Oliver Zhen, and Eric Yeung, "Investor Trading and the Post Earnings Announcement Drift", The Accounting Review, vol.86, Aug.2010, No Pagination.

[8] Dennis Olson and Charles Mossman, "Neural network forecasts of Canadian stock returns using accounting ratios", International Journal of Forecasting, vol.19.3, 2003, pp.453–465.

[9] Alireza Namdari and Zhaojun Li, "Integrating Fundamental and Technical Analysis of Stock Market through Multi-layer Perceptron", Proceedings of 2018 IEEE Technology and Engineering Management Conference (TEMSCON), June 2018, pp.1–6.

[10] Xiao-Dan Zhang, Ang Li, and Ran Pan, "Stock trend prediction based on a new status box method and AdaBoost probabilistic support vector machine", Applied Soft Computing, vol49, 2016, pp.385–398.

[11] Francis Tay and Lijuan Cao, Application of support vector machines in financial time series forecasting, vol.29, pp.309-317, 2001.

[12] S. Madge, "Predicting Stock Price Direction using Support Vector Machines", Independent Work Report, 2015.

[13] Sotirios P. Chatzis, Vassilis Siakoulis, Anastasios Petropoulos, Evangelos Stavroulakis, and Nikos Vlachogiannakis, "Forecasting stock market crisis events using deep and statistical machine learning techniques", Expert Systems with Applications, vol.112, 2018, pp.353–371.

[14] Alaa Sheta, Sara Ahmed, and Hossam Faris, "A comparison between regression, artificial neural networks and support vector machines for predicting stock market index", International Journal of Advanced Research in Artificial Intelligence, vol.4, July 2015, pp.55–63.

[15] Ming-Wei Hsu, Stefan Lessmann, Ming-Chien Sung, TiejunMa, and Johnnie Johnson, "Bridging the Divide in Financial Market Forecasting: Machine Learners vs. Financial Economists", Expert Systems with Applications, vol.61, May 2016, pp.215–234.

[16] Michael E. Bradbury, "Voluntary Semiannual Earnings Disclosures, Earnings Volatility, Unexpected Earnings, and Firm Size", Journal of Accounting Research, vol.30.1, Spring 1992, pp.137–145.

[17] Erhan Beyaz, Firat Tekiner, Xiao-Jun Zeng, and John Keane, "Comparing Technical and Fundamental indicators in stock price forecasting", Proceedings of IEEE 4th International Conference on Data Science and Systems, June 2018, pp.1607–1613.

[18] Bin Duan and William P. Dunlap, "The Robustness of Trimming and Winsorization When the Population Distribution Is Skewed", Pro Quest Dissertations and Theses, 1998.

[19] Corinna Cortes and Vladimir Vapnik, "Support-Vector Networks", Machine Learning, vol.20.3, Sep 1995, pp.273–297.

[20] Stephen Tyree, Kilian Weinberger, Kunal Agrawal, and Jennifer Paykin, "Parallel boosted regression trees for web search ranking", Proceedings of the 20th international conferenceon World wide web, 2011, pp.387–396.

[21] Jerry Ye, Jyh-Herng Chow, Jiang Chen, and Zhaohui Zheng, "Stochastic gradient boosted distributed decision trees", Proceedings of the 18th ACM conference on Information and knowledge management, Jan. 2009, pp.2061–2064.

[22] Tianqi Chen and Carlos Guestrin, "XGBoost: A Scalable Tree Boosting System", Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp.785–794.

[23] Thomas G. Dietterich, "Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms", Neural Computation, vol.10, 1996, pp.1895–1923.

[24] William Sharpe, "Capital Asset Prices: A Theory of Market Equilibrium Under Conditions of Risk", The Journal of Finance1, vol.9, 1964, pp.425–442.

[25] Sebastian Raschka, "Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning", ArXivabs/1811.12808, 2018.

[26] Sebastian Raschka, "MLxtend: Providing machine learning and data science utilities and extensions to Python's scientific computing stack", The Journal of Open Source Software, vol.3.24, Apr 2018.

[27] Jianwen Xie, Ruiqi Gao, Erik Nijkamp, Song Zhu, and Ying-nian Wu, "Representation Learning: A Statistical Perspective", Annual Review of Statistics and Its Application, vol.7, Mar 2020.