

Predicting Equity Market Price Impact with Performance Weighted Ensembles of Random Forests

Ash Booth

Institute for Complex System Simulation
University of Southampton
United Kingdom
Email: ash.booth@soton.ac.uk

Enrico Gerding

Electronics and Computer Science
University of Southampton
United Kingdom
Email: eg@ecs.soton.ac.uk

Frank McGroarty

Management School
University of Southampton
United Kingdom
Email: f.j.mcgroarty@soton.ac.uk

Abstract—For many players in financial markets, the price impact of their trading activity represents a large proportion of their transaction costs. This paper proposes a novel machine learning method for predicting the price impact of order book events. Specifically, we introduce a prediction system based on performance weighted ensembles of random forests. The system’s performance is benchmarked using ensembles of other popular regression algorithms including: liner regression, neural networks and support vector regression using depth-of-book data from the BATS Chi-X exchange. The results show that recency-weighted ensembles of random forests produce over 15% greater prediction accuracy on out-of-sample data, for 5 out of 6 timeframes studied, compared with all benchmarks.

I. INTRODUCTION

Over the last two decades almost all stock and derivatives exchanges across the globe have transitioned to electronic trading using limit orderbooks (LOBs), creating the need for a new breed of quantitative models to describe such markets. In particular, for large players that regularly transact in these markets, the price impact of their trading activity represents a large proportion of their transaction costs which are in turn considered a substantial determinant of investment performance [1]. As well as affecting the performance of an active investment strategy, they also affect how rapidly assets may be converted into cash. As such, it is vital that firms are able to quantify and predict the potential impact of their trading activity.

To address this challenge, this paper presents an empirical model for predicting the short term price impact in a limit orderbook of events that alter the best available prices in the book. Such events include any market orders or limit orders at the current best prices, as well as cancelations that remove all volume at the best quoted price. Henceforth, we refer to such situations simply as “events”. Specifically, we develop a model, based on performance weighted ensembles of random forests, to forecast the relative change in price 1, 5, 10, 60 and 600 seconds after an event. The model is trained and tested on 100 days of full depth of book data from the BATS Chi-X exchange.

While other studies have investigated the predictive power of more tradition regression techniques, currently, no study has explored the use of performance weighted ensemble to

predict the short term price impact of order book events. We demonstrate that an ensemble of recency-biased, performance-weighted random forests is able to predict the price impact of events more consistently and with greater accuracy than linear regression, neural networks, support vector regression and a single algorithms or combines as ensembles.

The paper is structured as follows. Section II gives an overview of the relevant literature. Section III describes the features and the feature selection algorithm used to refine the inputs to the model while Section IV describes the prediction algorithm itself. In Sections V and VI the experiments are described and results summarised while Section VII gives concluding remarks and discussed potential future work.

II. LITERATURE REVIEW

One of the dominant questions of market microstructure analysis regards how trading activity leads to price changes. The impact of incoming orders is an integral part of the price formation process as it translates changes in supply and demand into changes in price. The early market microstructure literature describes this concept with a focus on specialist markets. In such markets, prices are quoted by a centralised market-maker who receives orders from brokers and updates her quoted prices as according to the incoming order-flow that she witnesses. From the viewpoint of the broker, the price impact of his orders is a cost paid to the market-maker for her continued obligation to accept his orders [2], i.e. a cost for immediacy. From the viewpoint of the market-maker, some information about the future prices of assets is inferred from the order-flow of the brokers. This information is then captured in the market-maker’s quotes in a process reflected by the *permanent market impact* [3]. The difference between the price that an order obtains and the best prevailing quote is termed the *immediate price impact* and is an increasing function of order size. The *temporary price impact* is then defined as the difference between the immediate and permanent impact of an order.

Although specialist markets have mainly been replaced with electronic LOBs, the same price impact terminology is still used. However, due to the decentralised nature of LOBs, it is much harder to disentangle temporary and permanent price impact. Some efforts have been made to describe price

impact in LOBs with a theoretical economic model. This approach requires complete knowledge of all traders' utilities and strategies in order to derive order-flow dynamics. Even though this may be possible in some highly stylised markets, such models commonly involve many free parameters and tend to be intractable, limiting their practical applicability [4].

Given the shortcomings of theoretical market microstructure models, statistical price models have gained significant interest [5]. Such statistical approaches are made possible by the availability of detailed orderbook data. As such, empirical studies of price impact are often differentiated according to the type of data used: proprietary data from a single firm regarding its own orders, or public exchange data that contains anonymised orders from all market participants.

Many academic studies have worked only with proprietary datasets that contain the orders of, typically, just one firm (see [6]). Such datasets are information rich, containing detailed information on each order including the parent trade that each order is part of. With this information, smaller orders, that were submitted to an exchange, can be aggregated into larger trades that describe a firm's intentions. Also, this kind of data allows one to model parent order transaction cost or impact as a function of its size, as well as other parameters that are not included in publicly available exchange data. While most studies using proprietary data agree that the price impact of an order is a monotonically increasing function of its size (e.g., see [6], [7]), their reliance on such data is prone to several criticisms. Firstly, there are a number of factors that affect price impact but are difficult to control in a proprietary dataset, including the trading venues and algorithms used to execute a trade [8]. Second, the trader of the orders is able to stop an execution during unfavourable circumstances, introducing significant selection bias [9]. Thus, the ability to model the price impact of parent orders is important, but the criticisms mentioned above make it difficult to draw generalisable conclusions based on proprietary data.

An alternative approach is to use anonymised publicly available data from exchanges that includes all of the orders placed by all traders. However, until a short time ago, almost all studies focussed exclusively on marketable orders, analysing correlations between various order parameters and succeeding price movement [10], [11]. These studies found the same conclusions as those that used proprietary data: the temporary impact of a marketable order is an increasing concave function of order size. However, a major problem exists with the use of only marketable orders in the study of price impact. It has been reported that marketable orders account for only 10% of orders that reach exchanges [12], likely due to fast algorithmic trading strategies using passive limit orders to reduce costs [13]. Thus, a model that describes only the impact of marketable orders gives a somewhat limited description of the price formation process. More recent studies have considered the impact of marketable and passive orders, and relate their impact to a variety of factors including standing limit orders, order arrival rate, daily volume and cancellation rate [14], [15]. The statistical models used in these studies reveal complex interactions between the dynamics of order flows in a limit orderbook and the resulting high level price dynamics.

Correspondingly, modern machine learning techniques lend

themselves perfectly to capturing complex interactions in data. As a result, many such techniques have been applied to prediction with financial data including: neural networks [16], support vector machines [17] and evolutionary learning [18]. However, many of these methods have been shown to be prone to overfitting the training data.

Random forests, on the other hand, have been shown to always converge such that over-fitting does not occur [19] and by their very nature have proved successful across a range of variety of domains including: image classification [20], ecological prediction [21], gene selection [22] and finance [23]. A method that shows particular promise is the online generation of performance-weighted random forests, which was shown to perform consistently well in out of sample test when predicting the daily return of stock prices [24].

Against this background, we propose an adaptation to the system specified in [24] for the application of a performance weighted ensemble of random forests to predicting price impact. Specifically, we address the potential overfitting problem common to financial data by using random forests and tackle the non-stationary element of data by generating random forest in an online fashion.

III. FEATURE SELECTION

In order to make meaningful predictions we wish to capture the state of the LOB at a given time with a number of features. These features then form the input to the prediction system. Given the richness of information available in depth of book data, we propose the exploration of more than 100 features that can be broadly divided into three categories:

- Price* These features provide information about the log-normalised best bid/ask price series. This includes various technical analysis indicators such as: moving averages of the last n prices before a trade, price oscillators and the relative strength index (RSI) over the last N price changes.
- Spread* These features aim to provide information about relative changes in the bid/ask spread. Again, many technical analysis indicators are used here, with time series data normalised by the minimum allowable price increment for a particular stock.
- Liquidity* These features contain information about the apparent liquidity of the book. E.g. relative depth of each side of the book, order arrival and cancelation rates, and modal order price relative to best bid/ask price.

Clearly, not all features will have the same predictive power. Thus, in the interests of dimensionality reduction and computation, we wish to eliminate those features that have little or no impact on the performance of our random forest based prediction model. One simple method for selecting features is to choose the subset of features that is most highly correlated to the target variable. However, this is likely to generate a highly collinear feature space and impair the performance of the learning algorithm. As a result, we use a method of feature importance ranking, first suggested by Breiman [19], to eliminate irrelevant variables.

In more detail, to rank the features, a single random forest is first trained on the training data. While doing so, the root mean squared error (RMSE) is recorded for each tree, on the out-of-bag portion of the data¹. The same is done after the values of the features are randomly permuted. The difference between the errors are then averaged across all trees and normalised by the standard deviation of the differences. This represents the signal-to-noise ratio of the difference between the RMSE before and after permutation of the features. Specifically, the importance of feature j , \mathcal{VI}_j , is calculated as:

$$\mathcal{VI}_j = \frac{\sum_{\theta=1}^{\Theta} (e_{\theta,j} - e_{\theta,\pi j})}{\Theta \cdot \hat{\sigma}} \quad (1)$$

where Θ is the number of trees in the forest, $e_{\theta,j}$ is the RMSE of tree θ without permuting j , $e_{\theta,\pi j}$ is the RMSE of that tree after permutation of j , and $\hat{\sigma}$ is the standard deviation of the differences between $e_{\theta,j}$ and $e_{\theta,\pi j}$. Features that produce larger \mathcal{VI} values are considered more important than features that produce smaller values.

For feature selection, we propose a backwards elimination method as it has been shown that such methods provide a stronger variable subset than similar alternatives [25]. Our feature selection algorithm is described fully below:

- 1: Train a single random forest on the training data using all J features
- 2: Compute average RMSE of model on cross validation (CV) data
- 3: Rank features according to performance as defined by equation 1
- 4: **for** each subset of variables $J_i = J - 1, J - 2, \dots, 1$ **do**
- 5: Train a new forest using J_i features with highest \mathcal{VI}
- 6: Compute average RMSE of model on CV set
- 7: Rerank the features
- 8: **end for**
- 9: Determine which J_i yielded the smallest RMSE

This algorithm was applied to the training and CV data and a plot of the RMSE at each stage of the elimination is shown in Figure 1. As this figure shows, there is a slight initial increase in performance (decrease in RMSE) as features are eliminated. Following this, there is a swift decline in performance as features that are essential for prediction are eliminated (from right to left). The algorithm yields the optimal set of 76 features listed in Table II in the appendix.

IV. THE MODEL

This section describes an adaptation of a predictive system first proposed in [24] for predicting daily changes in stock prices. Our modified system consists of an ensemble of models, that is used to predict the price impact of events that alter the best bid/ask prices. The price impact is measured by taking the price changes at 1, 5, 10, 60 and 600 seconds after the event normalised by the pre-event price. A separate ensemble is used for the prediction of each time interval. An overview of the model structure is given in Figure 2.

¹As each tree in a random forest is constructed using its own bootstrap sample of about 60% of the data, the remaining data may be used to generate an unbiased error estimate.

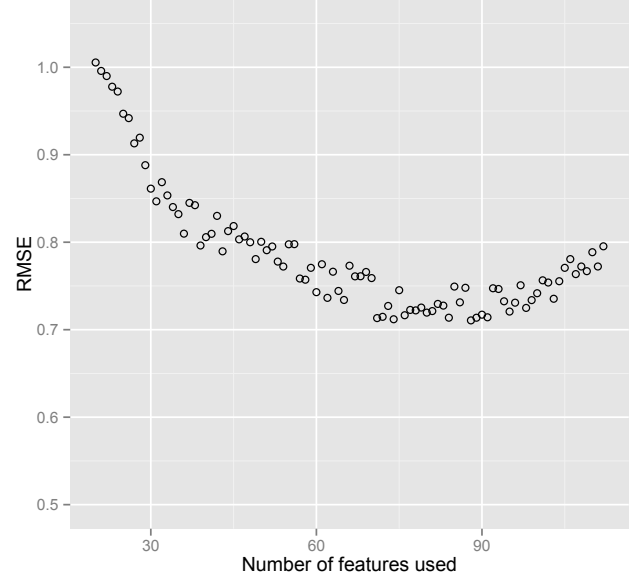


Fig. 1. Plot of RMSE for each round of the feature elimination algorithm. It can be seen that, as features are removed, there is an initial, slight improvement in performance before a rapid decline.

To that end, the prediction system described in this section is based on an ensemble of random forest regressors that will, from here on, be referred to as experts. Specifically, every fifteen minutes, three new experts are trained on the previous 1, 2 and 3 hours of data and added to the ensemble in an online fashion. That is, new experts are trained and introduced into the ensemble continuously, even during out of sample tests. The ensemble is capped to a maximum number of experts, E_{max} , and once the maximum number is reached the poorest performing expert is dropped upon each new addition.

A. The Base Learner

Each expert in the ensemble is represented by a random forest regression algorithm. In this paper, a random forest is an ensemble of many regression trees designed to produce accurate predictions without overfitting the training data [19]. Regression trees use their tree structure to recursively subdivide the features space such that the subsets of the feature space are manageable enough to fit simple models to them. Thus, a tree model has two portions: the recursive subdivisions and a simple model for each partition. Each of these partitions is represented by a terminal node (leaf) of a tree, and is attached to a simple model which applies only to that partition. An example of a decision tree trained on the inputs listed in Table II is given in Figure 3. To generate a prediction for a new data point we begin at the root node of the tree (EMA book vol.) and answer a series of questions about the feature values. For example, a data point with an *EMA book vol.* of 0.2 and an *EMA spread* of -0.6 would generate a prediction of 0.14 for the normalised price impact.

To train a random forest, bootstrap samples are drawn from the training data in order to construct multiple trees. Importantly, each tree is constructed using a randomised subset of features. In detail, we produce forests of regression trees

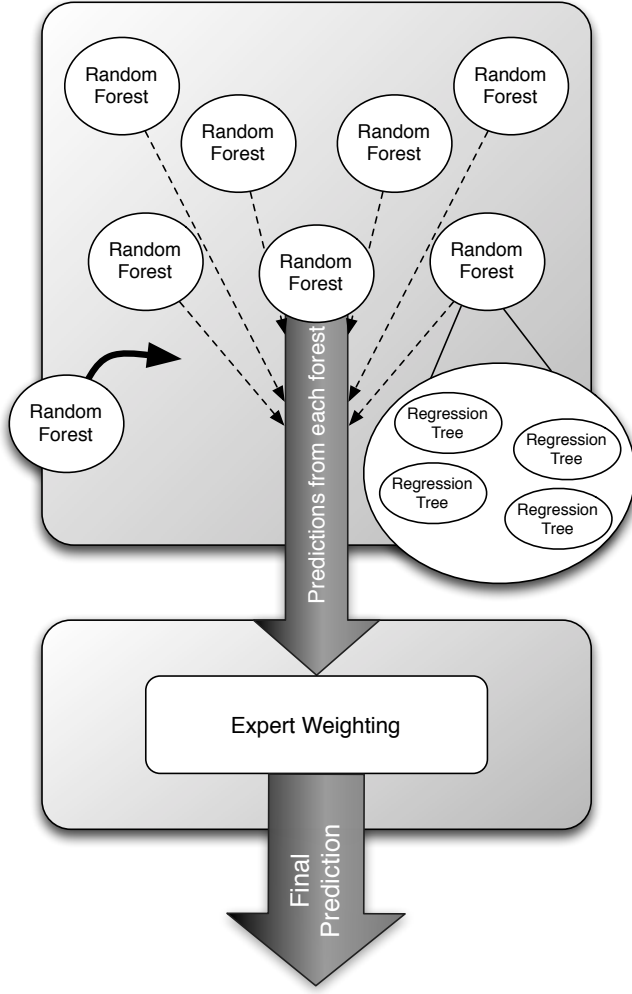


Fig. 2. An overview of the prediction system. The inputs given in table Table II are used to train three new random forests (experts) every fifteen minutes. Each expert makes a prediction about the price change after an event and the expert weighting layer aggregates these predictions using the method described in Section IV-B.

using the procedure below, where the training set is defined as $D = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$ with the aim to find a function $f: X \rightarrow Y$, where X is the space of features and Y is the output. Further, let M denote the number of features.

- 1) Randomly select n observations from D with replacement to form a bootstrap sample.
- 2) If there are M features, select $m \ll M$ features such that, at each node, m features are selected at random and the best split (measured using Gini impurity) on these m is used to split the node.
- 3) Grow each tree to the largest possible extent without pruning.

By using a random subset of features among which to select the best split and growing the tree without pruning, random forests are able to maintain high prediction accuracy while encouraging diversity among trees [19]. Moreover, a random selection of features reduces the potential for correlation between tree predictions, keeping overall model bias low. An

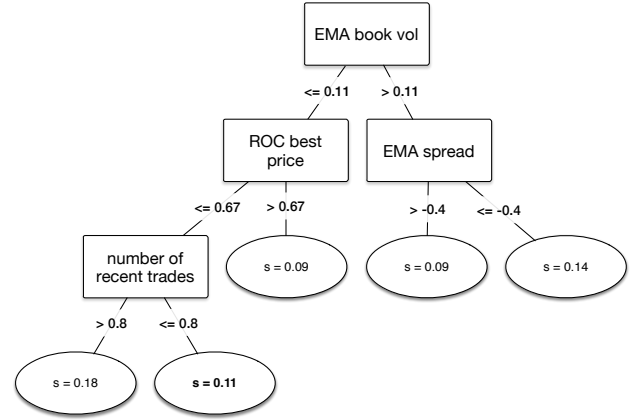


Fig. 3. Example of a simplified regression tree for predicting the price impact generated using the features in Table II.

ensemble of such trees seeks to also reduce variance.

On presentation with unseen observations, the random forest produces a prediction by averaging the predictions of all the individual trees in the forest. In the rest of this paper, this prediction is denoted by $S_{i,t}$, where i is the random forest, and t is the time of the observation.

The features given in Table II form the input to the random forest prediction algorithm. In our study, the goal of the random forest is to generate a function of the features given in Table II that predicts the relative price change at particular intervals after an order book event that alters the best bid/ask price (as described in Section I).

B. An Ensemble of Ensembles

The previous section describes the process of training and predicting with a single (random forest) expert. In order to improve performance of the prediction system, we propose an ensemble of such random forests that is generated in an online fashion. In this study, three experts are trained every fifteen minutes on a moving window of 1, 2 and 3 hours of training data to generate an ensemble of experts. The outputs of all of the experts are then combined using an expert weighting algorithm to generate a prediction about the price change after an event.

The expert weighting algorithm described in this study is based on an algorithm suggested by Creamer and Freund [26] that produces a prediction by taking an average of the predictions of all the experts weighted by their training error. The formula that they use for calculating weights is derived from a weighted majority algorithm first introduced in [27]. We explore a modification of Creamer and Freund's algorithm that, instead of using the complete historical performance of each expert, uses a recency-biased performance measure.

Specifically, we denote the historical performance of expert i at time t by $k_{i,t}$, giving more weight to experts with larger values of $k_{i,t}$. We define $k_{i,t}$ as:

$$k_{i,t} = \lambda/\eta_{i,t-1} + (1 - \lambda)k_{i,t-1} \quad (2)$$

where $\eta_{i,t-1}$ is the RMSE of the last prediction made by expert i at time $t-1$ and λ is a smoothing parameter that allows us to control the recency weighting of the performance measure.

As mentioned previously, a greater weight is given to experts that have a higher $k_{i,t}$ as this represents better historical performance. As such, we use the exponential weighting algorithm described in [26] for generating weights for each expert. Specifically, the weight of the expert trained first is given by:

$$w_{1,t} = \exp\left(\frac{k_{1,t-1}}{\sqrt{t}}\right) \quad (3)$$

The weight of all following experts is then defined as:

$$w_{i,t} = I_i \cdot \text{ramp}(t - t_i) \cdot \exp\left(\frac{k_{i,t-1}}{\sqrt{t - t_i}}\right) \quad (4)$$

where I_i is the weight assigned to a newly generated expert and is the mean of all current experts' weights; $\text{ramp}(t - t_i) = \min\left(\frac{t - t_i}{t_{i+1} - t_i}, 1\right)$ allows new experts to be brought in cautiously; and t_{i+1} is the time at which the next expert will be added.

The combined output of the ensemble of random forests at time t is then a weighted average of the predictions of all expert at time t :

$$P_{g,t} = \frac{\sum_i S_{i,t} w_{i,t}}{n} \quad (5)$$

where $S_{i,t}$ is the prediction of expert i at time t , n is the current total number of experts and $P_{g,t}$ is a prediction of the relative price change.

V. EXPERIMENTS

The analysis in this paper is based on historical depth-of-book data from the BATS Chi-X exchange using the 25 most actively traded stocks over 100 days of trading from 12th February 2013 to 3rd July 2013. This raw data contains details on all events in the book, that is: order arrivals, executions, modifications and cancellations, with each item timestamped to the millisecond. We consider only the regular trading time between 9:30 - 16:00, and all other periods are discarded. Prior to conducting experiments, the data is split into training, cross-validation (CV) and test sets as follows: 60 days training, 20 days CV and 20 days test. It should be noted that no parameters are tuned using the test set. This set is used only to report the performance of the model found to perform best on the CV data.

The experiments reported in Section VI were conducted according to the following procedure:

- 1: **for all** parameter settings **do**
- 2: **for all** stocks **do**
- 3: Generate features from data. As in Section III.
- 4: Iterate through training set generating three new experts every 15 minutes with a moving windows of data sized 1, 2 and 3 hours.

- 5: Monitor the prediction performance of each expert, updating the weights after every event according to Equation (4).
- 6: Upon reaching the CV data, an ensemble prediction is made for each event using Equation 5.
- 7: At the end of the CV period, store the average RMSE of prediction across the CV set for current stock.
- 8: **end for**
- 9: CV performance is reported as the average RMSE of the model across all 25 stocks.
- 10: **end for**
- 11: The model with the best CV performance for each stock is kept and run on the corresponding stock over the test set.

The experimental procedure outlined above is used to determine the optimal parameterisation of our model, as well as the benchmarks, using only the training and CV data sets. Once the optimal parameterisations are found, the performance on test set is reported. For the random forests themselves, there are only two parameters of interest: the number of trees grown in each forest (n_trees) and the number of features randomly chosen when splitting a node (m). For n_trees , the larger the better. We find no improvement in performance above $n_trees = 250$ and thus use this value throughout our experiments. Similarly we find no improvement in performance over a maximum number of experts, E_{max} , greater than 500. It is generally thought that lower values of m lead to a reduction of variance but also an increase in bias. We thus use cross validation to find optimal values for this parameter.

In addition to finding optimal values for m , cross validation is used to explore values of the recency bias, λ , for the performance metric described in Equation 2. To assess the performance of these variables we perform a two dimensional grid-search of the parameter space. The experimental procedure above is performed for each point in the parameter space to generate an average RMSE across all stocks for each parameterisation.

As the benefits of this method of expert weighting have been demonstrated in [24] we explore the effectiveness of random forests as the basis of the prediction system. To this end, we compare the predictive performance of the system described above with a multitude of base learners including: random forests, ordinary least squares regression, multi-layer feed-forward neural networks with backpropagation and support vector regression with Gaussian kernels. The parameters of each of the base learning algorithms optimised using multi-dimensional grid search on the CV data sets. Results are reported as the performance on the previously unused test set.

VI. RESULTS

The results of the parameter grid-search for our model are shown in Figure VI. It can be seen that the best parametrisation for the recency-weighted model lies in the vicinity of $\lambda = 0.85$ and $m = 18$. This value for m is equivalent to using all possible features for each node split, and it has been shown that such a parametrisation is often optimal for random forest regression [28]. A value of $\lambda = 0.88$ is also interesting as it is a very commonly used value for exponential moving averages and is very close to the value used by the financial risk management

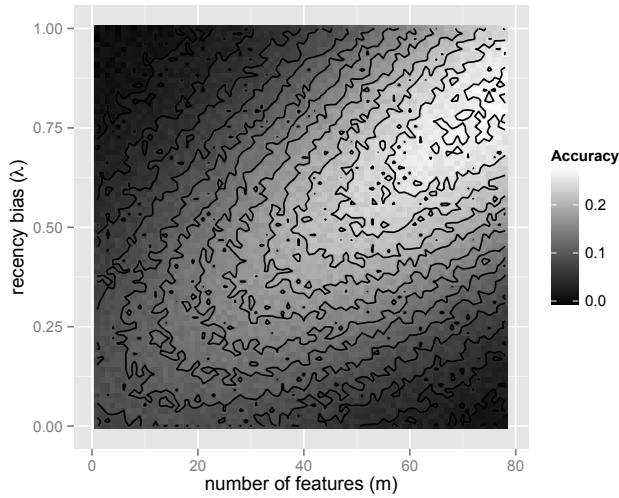


Fig. 4. Heat map showing results performance of our ensemble model over the gridsearch space. Higher values represent better performance.

TABLE I. A COMPARISON OF THE PERFORMANCE OF VARIOUS ENSEMBLES OF REGRESSION MODELS AND A SINGLE RANDOM FORESTS ALGORITHM IN PREDICTING THE RELATIVE CHANGE IN ASSET PRICE $t = 1, 5, 10, 60$ AND 600 AFTER AN EVENT. ASTERISKS SIGNIFY A STATISTICAL SIGNIFICANCE COMPARED TO OUR MODEL OF $p < 0.05$. OUTPUTS OF ALL REGRESSION ALGORITHMS ARE COMBINES AS DESCRIBED IN SECTION IV-B

Model	RMSE at $t =$				
	1	5	10	60	600
Training Phase					
Single random forest	0.22*	0.28*	0.29	0.33	0.43
Linear regression ensemble	0.18*	0.22*	0.26*	0.34*	0.49*
Neural Network ensemble	0.13*	0.19*	0.23*	0.31*	0.41*
SVR ensemble	0.13*	0.18*	0.20*	0.31*	0.40*
Random forest ensemble	0.15	0.20	0.26	0.32	0.40
CV Phase					
Single random forest	0.23*	0.28*	0.31*	0.35	0.44
Linear regression ensemble	0.21*	0.26*	0.32*	0.38*	0.50*
Neural network ensemble	0.22*	0.29*	0.31*	0.40*	0.45*
SVR ensemble	0.24*	0.29*	0.35*	0.36*	0.42*
Random forest ensemble	0.15	0.22	0.25	0.33	0.40
Test Phase					
Single random forest	0.19*	0.30*	0.31*	0.35	0.46*
Linear regression ensemble	0.24*	0.27*	0.31*	0.42*	0.59*
Neural network ensemble	0.24*	0.28*	0.31*	0.37*	0.50*
SVR ensemble	0.40*	0.39*	0.39*	0.45*	0.47*
Random forest ensemble	0.15	0.23	0.24	0.34	0.40

firm RiskMetricsTM for their moving average estimation of volatility [29].

Table I shows the results of the comparative performance of the various base learners outlined above in predicting the price of stocks at various time point after best-price-altering events. The inputs for all algorithms are those listed in Table II. Experiments were performed as described above with results averaged across all stocks.

It can be seen that our recency-weighted random forest ensemble outperforms all other models on out-of-sample data.

While the random forest based ensemble produces higher error values on the training data, the proven ability of random forests to avoid over fitting allows the ensemble of random forests to produce significantly superior out of sample results to all other models on 4 out of 5 time intervals. A distinct advantage of online ensemble generation can be seen when comparing the results of a single random forest to the ensemble of random forest. Introducing online training of experts improves the performance of the model across all 5 time periods, with 4 of 5 results proving significant at $p = 0.05$

VII. CONCLUSION

In this paper we apply, for the first time, performance weighted ensembles of random forests to the prediction of the price impact of order book events. In more detail, random forest regressors are continually generated, at fixed time intervals, and added to an ensemble of such experts. Upon the occurrence of an event that alters the best prices of the orderbook (limit order at the best price, market order, cancellation or modification), each expert makes a prediction and an expert weighting system averages the predictions of all experts weighted by their recent performance. A separate ensemble is used to predict the price change at $t = 1, 5, 10$, and 60 seconds after an event. This performance of random forest ensembles is then benchmarked against ensembles of linear regression, neural networks and support vector models as well as single random forests. The models are trained using 60 days of data, cross-validated using 20 and tested using 20. Reported results are an average over the 25 most liquid stocks from the BATS Chi-X exchange.

In out-of-sample trading simulations, the performance-weighted random forest ensemble outperforms all other models across all time intervals. Specifically, our model outperforms all other models by around 20% when predicting the price change at 1, 5, and 10 seconds. Above this, performance of all models fall substantially. This demonstrates the ability of the online-generated ensemble to make predictions in the highly non-stationary orderbook environment. In addition to the increased prediction accuracy, the continual addition of new experts to the random forest ensemble has a number of other advantages. Firstly, new experts are trained separately to the ensemble and added seamlessly without the need to stop the system or close out any open trades, a necessity for a quantitative trading applications. Secondly, each random forest, and thus the entire ensemble, is composed of human readable and visually representable decision trees. Thus, there is no ‘black box’ and, as such, simple visual analysis can give managers and regulators insight into the risks involved in such a trading strategy.

In future work we intend incorporate this approach into optimal trade scheduling algorithms and test its performance across a wider range of securities.

ACKNOWLEDGMENT

The authors are extremely thankful for the depth-of-book data that was provided by *G-Research* and the insightful comments provided by the reviewers.

REFERENCES

- [1] R. Guobuzaitė, K. Byrne, and A. Freyre-Sanders, “A Review of Trading Cost Models,” *The Journal of Investing*, vol. 13, pp. 93–115, 2004.

- [2] H. Demsetz, "The Cost of Transacting," *Quarterly Journal of Economics*, vol. 82, pp. 33–53, 1968.
- [3] A. S. Kyle, "Continuous Auctions and Insider Trading," *Econometrica*, vol. 53, p. 1315, 1985.
- [4] C. C. A. Parlour and D. J. D. Seppi, "Limit Order Markets : A Survey," *Handbook of Financial Intermediation and Banking*, vol. 5, pp. 63–96, 2008.
- [5] R. Kissell, M. Glantz, and R. Malamut, "A practical framework for estimating transaction costs and developing optimal trading strategies to achieve best execution," *Finance Research Letters*, vol. 1, no. 1, pp. 35–46, Mar. 2004.
- [6] R. Almgren, C. Thum, E. Hauptmann, and H. Li, "Direct Estimation of Equity Market Impact," *Risk*, vol. 18, no. 7, pp. 58–62, 2005.
- [7] B. Tóth, Y. Lempérière, C. Deremble, J. de Lataillade, J. Kockelkoren, and J.-P. Bouchaud, "Anomalous Price Impact and the Critical Nature of Liquidity in Financial Markets," *Phys. Rev. X*, vol. 1, no. 2, Oct. 2011.
- [8] Y. Brandes, I. Domowitz, B. Jiu, and H. Yegerman, "Algorithms, Trading Costs, and Order Size," *Trade*, 2007.
- [9] A. A. Obizhaeva, "Liquidity Estimates and Selection Bias," 2012.
- [10] J. Hasbrouck, "Measuring the information content of stock trades," *The Journal of Finance*, vol. 46, pp. 179–207, 1991.
- [11] X. Gabaix, P. Gopikrishnan, V. Plerou, and H. E. Stanley, "A theory of power-law distributions in financial market fluctuations," *Nature*, vol. 423, pp. 267–270, 2003.
- [12] J. Hasbrouck and G. Saar, "Low-latency trading," *Journal of Financial Markets*, vol. 16, no. 4, pp. 646–679, 2013.
- [13] E. Fishler, "The Difficulty of Trading "Ultra-Liquid" Stocks," *Pragma Trading*, Tech. Rep. 4, 2012.
- [14] F. Lillo, J. D. Farmer, and R. N. Mantegna, "Master curve for price impact function," *Nature*, vol. 421, pp. 129–130, 2003.
- [15] Z. Eisler, J.-P. Bouchaud, and J. Kockelkoren, "The price impact of order book events: market orders, limit orders and cancellations," *Quantitative Finance*, vol. 12, no. 9, pp. 1395–1419, 2012.
- [16] D. Enke and S. Thawornwong, "The Use of Data Mining and Neural Networks for Forecasting Stock Market Returns," *Expert Systems with Applications*, vol. 29, no. 4, pp. 927–940, Nov. 2005.
- [17] F. E. H. Tay and L. J. Cao, "Modified Support Vector Machines in Financial Time Series Forecasting," *Neurocomputing*, vol. 48, no. 1, pp. 847–861, 2002.
- [18] B. Lebaron, *An Evolutionary Bootstrap Method for Selecting Dynamic Trading Strategies*. Norwell, MA: Social Systems Research Institute, University of Wisconsin, 1998.
- [19] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [20] A. Bosch, A. Zisserman, and X. Munoz, "Image classification using random forests and ferns," in *IEEE 11th International Conference on Computer Vision*. Ieee, 2007, pp. 1–8.
- [21] A. M. Prasad, L. R. Iverson, and A. Liaw, "Newer Classification and Regression Tree Techniques: Bagging and Random Forests for Ecological Prediction," *Ecosystems*, vol. 9, no. 2, pp. 181–199, 2006.
- [22] R. Díaz-Uriarte and S. Alvarez De Andrés, "Gene selection and classification of microarray data using random forest," *BMC Bioinformatics*, vol. 7, no. 1, p. 3, 2006.
- [23] B. Larivière and D. Vandenpoel, "Predicting customer retention and profitability by using random forests and regression forests techniques," *Expert Systems with Applications*, vol. 29, no. 2, pp. 472–484, 2005.
- [24] A. Booth, E. Gerding, and F. McGroarty, "Automated trading with performance weighted random forests and seasonality," *Expert Systems with Applications*, vol. 41, no. 8, pp. 3651–3661, 2014.
- [25] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.
- [26] G. Creamer and Y. Freund, "Automated Trading with Boosting and Expert Weighting," *Quantitative Finance*, vol. 4, no. 10, pp. 401–420, 2010.
- [27] N. Littlestone and M. K. Warmuth, "The weighted majority algorithm," in *30th Annual Symposium on Foundations of Computer Science*, 1989, pp. 256–261.
- [28] A. Liaw and M. Wiener, "Classification and Regression by randomForest," *R news*, vol. 2, no. 3, pp. 18–22, 2002.
- [29] J.P.Morgan and Reuters, "RiskMetrics Technical Document," New York, Tech. Rep., 1996.

APPENDIX

TABLE II. TABLE DESCRIBING FEATURES USED FOR THE MODEL ENSEMBLE OF RANDOM FORESTS MODEL. PRICE FEATURES ARE ALL NORMALISED BY THE PRICE PRECEDING AN EVENT, WHILE SPREAD FEATURES ARE NORMALISED BY THE MINIMUM PRICE INCREMENT ALLOWABLE IN THE BOOK.

Indicator	Parameters
Price features	
<i>Exponential moving average</i> of the last n observations of best prices	$n = 16$
<i>Bollinger bands</i> of the last n observations of best prices	$n = 32$
<i>Momentum</i> of the best prices over the last n observations	$n = 12$, and 24
<i>Acceleration</i> of the best prices over the last n observations	$n = 18$
<i>The rate of change</i> of the best prices over the last n observations	$n = 22$
<i>The MACD</i> of the best prices	$f = 12$, $s = 24$
<i>The relative strength index</i> of the best prices over the last n observations	$n = 20$ and 32
<i>The fast stochastic K</i> of the best prices over the last n observations	$n = 12$ and 18
<i>The Chaikin volatility</i> of the best prices over the last n observations	$n = 10$
<i>The accumulation/distribution line</i>	-
<i>The Chaikin oscillator</i>	$n_1 = 3$, and $n_2 = 10$
Spread features	
<i>Exponential moving average</i> of the last n observations of spread	$n = 10$
<i>Momentum</i> of the spread over the last n observations	18
<i>The rate of change</i> of the spread over the last n observations	$n = 10, 16$ and 22
<i>The MACD</i> of the spread	$f = 12$, $s = 30$
<i>The relative strength index</i> of the spread over the last n observations	$n = 14$
<i>The fast stochastic K</i> of the spread over the last n observations	$n = 12$ and 24
Liquidity features	
<i>Exponential moving average</i> of bid/ask book volume over the last n observations	$n = 22$
<i>Exponential moving average</i> of volume at best bid/ask price over the last n observations	$n = 12$ and 36
<i>Momentum</i> of bid/ask book volume over the last n observations	$n = 12, 24$ and 36
Number of price improvements in the last n observations	$n = 25$, and 50
Number of trades in the last n observations	$n = 50$
Number of bid/ask quotes arrived in the last n observations	$n = 50$
Number of bid/ask cancellations in the last n observations	$n = 50$
Current modal bid/ask price relative to best bid/ask price	-
Current mean price increment between order prices	-