Application of Conformal Prediction Interval Estimations to Market Makers' Net Positions

Wojciech Wisniewski WOJCIECH.WISNIEWSKI.2019@LIVE.RHUL.AC.UK
Department of Computer Science
Royal Holloway, University of London,
Egham, United Kingdom

David Lindsay DAVID@ALGOLABS.COM
Siân Lindsay SIAN@ALGOLABS.COM
Algorithmic Laboratories
Bracknell, United Kingdom

Editor: Alexander Gammerman, Vladimir Vovk, Zhiyuan Luo, Evgueni Smirnov and Giovanni Cherubin

Abstract

In this study we focus on the application of Conformal Prediction (CP) interval estimations to provide financial Market Makers (MMs) with some "meaningful" forecasts relating to their future short-term position in a given financial market. The idea is that using these market position forecasts, MMs can deploy proactive risk management strategies with a given degree of confidence. We make use of a novel financial time series dataset that comprises the net positions of a given MM over a three year period for trades pertaining to the top-traded Foreign Exchange (FX) symbols. This dataset - NetPositionTimeSeries - is noisy and complex. The net positions within it are generated from the trades of tens of thousands of clients trading in different directions (buy or sell) and over many different time horizons. We approached the problem of predicting future net position not as one that required an accurate point estimate as this is impossible. Rather we sought to gain a meaningful range of possible position bounds which would nonetheless be invaluable. In this study we tested a range of predictive Machine Learning (ML) techniques. We compared the CP framework to benchmark methods like moving average (MA) and quantile regression (QR). We demonstrate how application of the CP framework gives well calibrated region bounds on the MM net position forecasts.

Keywords: Prediction Intervals, Conformal Predictors, Time Series, Uncertainty, Net Position Forecast, Market Makers, Finance, Risk Management, Foreign Exchange

1. Introduction

Financial market makers (or MMs), are a collective term describing companies that quote both buy (bid) and sell (ask) prices in financial instruments, such as Foreign Exchange (FX). MMs quote prices to many different market participants, from individuals trading at home or on their mobile devices, to brokerages who forward the prices onto thousands of their clients, to larger professional financial institutions such as investment banks and hedge funds.

When a market participant accepts a quoted buy or sell price from the MM, the MM immediately places a corresponding sell or buy order from its own inventory or *net position* to fulfil the order.

The MM has two linked objectives: 1) to make some profit from the bid-ask spread, which is the difference between the price the MM quotes and the price it pays from its own inventory, the former of which is always higher, and 2) to manage its market exposure or risk by maintaining its position within some predefined bounds. Ideally the MM would prefer to hold a neutral (or flat) position for most of the time. This is because whilst the MM holds some amount of inventory, it remains at the mercy of any unfavourable price movements in the wider market to the extent that unloading its inventory is no longer profitable and may incur a significant loss. To help achieve the objectives of making profit and reducing risk, MMs would benefit from being informed about what their market positions might be at some point in the future. To the best of our knowledge, predictive models that provide insight into future net position movements *with a high degree of confidence* are a relatively uncharted area of research. This is the focus of the study presented here.

In this paper we apply the conformal prediction framework to predict a MM's net position within the next hour. Conformal prediction (see Vovk et al. (2005), Gammerman et al. (1998)) is a technique with rigorous performance guarantees and has been used successfully for classification and regression problems. Applying it to time series data is problematic since it violates the main assumption of exchangeability. The latter states that the join distribution of a sequence does not change under any permutation which is clearly not the case for time series data such as the MM position dataset tested in this study.

To deal with this problem, modifications to the conformal framework were proposed by Balasubramanian et al. (2014, pp. 183–184), where one assumes that a sample only depends on time window W. Other applications can be found as below (note this is a non-exhaustive list):

- Dashevskiy and Luo (2011) where Prediction with Expert Advice and Conformal Predictors (CP) is performed and despite the violation of exchangeability, empirically valid intervals were obtained.

- Chernozhukov et al. (2019) where an extension of the applicability of conformal inference to time series data is performed within the framework of randomisation inference. The authors provide approximate validity under weak assumptions on the conformity score when the exchangeability condition is violated (ex. i.i.d residuals will suffice).

- Kath and Ziel (2019) applied CP framework to short-term electricity price forecasting in different markets. They found CP framework to give reliable results, moreover they a path dependent evaluation study of key aspects of CP was conducted.

In this study we will use an Inductive conformal prediction (Johansson et al. (2014)) using the window approach which is also recently proposed in Kath and Ziel (2019). Moreover we will compare a number of machine learning algorithms combined with CP to the respective quantile regression models and a simple benchmark, a moving average where the confidence bounds are proportional to moving standard deviation.

We also make use of several performance measures and interval forecast statistical tests in order to assess efficiency and validity of the prediction intervals.

2. Conformal Prediction

For our task the Inductive version of CP (ICP) is used and what follows is a brief description of the main principles and assumptions of ICP. Conformal prediction yields valid prediction intervals that meet the designated confidence level $1-\alpha$. We assume that there is no long-range dependence between the observations in order to fit CP to a time series framework. Conformal Prediction is very versatile with applications in regression, classification or an online or batch setting. It adds an interval estimate to an existing point forecasting model with the help of a non-conformity score λ which determines how uncommon an observation is in comparison to the real value. A fuller discussion of this is provided in Vovk et al. (2005).

Suppose we have:

- The out of rolling sample prediction scheme (see Figure 1) on dataset $\mathcal{D}$, where a model is re-trained, re-calibrated and re-tested on rolling dataset $\mathcal{D}_h$, where the entire dataset is defined as $\mathcal{D} := \cup_h \mathcal{D}_h$.
- A dataset $\mathcal{D}_h = \{(\mathbf{x}_{1,h}, y_{1,h}), ..., (\mathbf{x}_{L,h}, y_{L,h}), (\mathbf{x}_{L+1,h}, y_{L+1,h})\}$ that we split into:
  1. A training set
     $\mathcal{D}_{\text{train},h} = \{(\mathbf{x}_{1,h}, y_{1,h}), ..., (\mathbf{x}_{M,h}, y_{M,h})\}$
  2. A calibration set
     $\mathcal{D}_{\text{calib},h} = \{(\mathbf{x}_{M+1,h}, y_{M+1,h}), ..., (\mathbf{x}_{L,h}, y_{L,h})\}$.
  3. A test set
     $\mathcal{D}_{\text{test},h} = \{(\mathbf{x}_{L+1,h}, y_{L+1,h})\}$.
- A model that exploits $\mathcal{D}_{\text{train},h}$ for training and yields estimate $\hat{y}_i$. We train on $\mathcal{D}_{\text{train},h}$ and supply it with the data of $\mathcal{D}_{\text{calib},h}$ to obtain unbiased out-of-sample alike estimates $\hat{y}_{M+1,h}, ..., \hat{y}_{L,h}$,
- The simplest non-conformity score $\lambda_{i,h} = |y_{i,h} - \hat{y}_{i,h}|$, only applied on the estimates in $\mathcal{D}_{\text{calib},h}$.

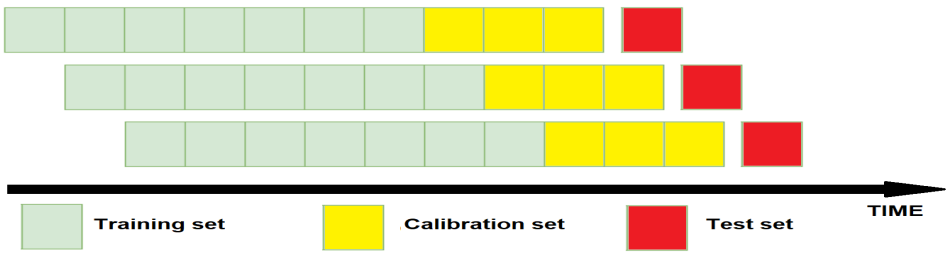

Figure 1: Out-of-sample rolling estimation scheme. The windows slides and new $\mathcal{D}_{\text{train},h}$, $\mathcal{D}_{\text{calib},h}$ and $\mathcal{D}_{\text{test},h}$ are constructed and fed to the conformal prediction algorithm.

The division into training and calibration is essential since we explicitly fit a model on $\mathcal{D}_{\text{train}}$ and exploit $\mathcal{D}_{\text{calib}}$ in an out-of-sample context. The point forecast model is trained to minimize the error made for $\mathcal{D}_{\text{train}}$. For the determination of non conformity scores the threshold value $\lambda^{\alpha}_{L+1,h}$ is identified by the equation (based on Johansson et al. (2014))

$$\lambda^{\alpha}_{L+1,h} := \min_{x \in \{\lambda_{M+1,h},\ldots,\lambda_{L,h}\}} \left\{ x : \frac{|\{i = M+1,..,L : \lambda_{i,h} < x\}| + 1}{|\mathcal{D}_{\text{calib},h}| + 1} \geq 1 - \alpha \right\}. \tag{1}$$

The following symmetric interval comprises the true net positions with confidence $1-\alpha$ under exchangeability in the underlying dataset

$$\hat{y}_{L+1,h} \pm \lambda^{\alpha}_{L+1,h}. \tag{2}$$

It is worth mentionning that this approach does not take into account the possible issue of heteroscedasticity. Since net position manifests volatility clustering we also use the approach referred to as Normalized Conformal Prediction that overcomes this issue. The non-conformity score now becomes

$$\lambda_{i,h} = \frac{|y_{i,h} - \hat{y}_{i,h}|}{\hat{\varepsilon}_{i,h}}, \tag{3}$$

with $\hat{\varepsilon}_{i,h}$ is the estimation of accuracy/error or the corresponding prediction. A separate model is used that predicts those errors in parallel. The interval forecast is now given by

$$\hat{y}_{L+1,h} \pm (\lambda^{\alpha}_{L+1,h} \hat{\varepsilon}_{L+1,h}). \tag{4}$$

3. Proposed Approach

In this section we briefly present an overview of the dataset used in this study. We then describe pre-prossessing and feature engineering of the data and then move onto presenting the machine learning models that were applied.

3.1. The NetPositionTimeSeries Dataset

The publicly available NetPositionTimeSeries dataset (Lindsay (2020)) comprises the historic net positions accumulated from a sample of a MM's client flow during January 2014 to January 2017. Positions are based on the MM's 5 most liquid currency pairs during that time (e.g. EUR/USD). Net positions are measured hourly and in US dollars (USD). Figure 2 illustrates how the MM's position changes in size and direction over time and according to the type (buy or sell), size (amount) and sequence of orders made by the MM's clients. Essentially the MM position changes are equal and opposite to those of its client order fills.

Figure 3 provides three perspectives of the NetPositionTimeSeries dataset. Assessing each plot in turn (from left to right), we can ascertain that in this dataset:

1. Net position across the three years is volatile and fluctuates between 40 mio[1] USD short and 120 mio USD long. In 2014 net positions are mostly long, whereas from

1. mio: an abbreviation of million(s) .

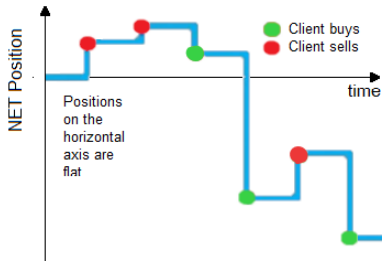

Figure 2: How a market maker's net position changes over time. The MM's position is categorised as being FLAT, LONG or SHORT if the position size is on, above or below the horizontal axis, respectively.

mid 2014 onwards they tend to be short and of smaller magnitude (+ 20 to - 40 mio USD);

2. The volatility of the position changes each hour varies through time too, for example in early 2014 we observe jumps of +/- 70 mio USD, which then settle down thereafter typically +/- 15 mio USD;
3. The autocorrelation of the time series is significant up until a window of approximately 60 days perhaps due to that being the maximum holding period of most of the MM's clients.

It is important to remember that this complexity in the data arises from the fact that the net positions are an aggregation of thousands of different clients trading many different markets with different amounts of capital, over different time horizons and with different motivations.

In this study we attempt to apply various regression models to this NetPosition time series data, however the effective usage of these algorithms require stable variance within the data, to combat this we apply the following so called Yeo and Johnson transformation to standardised Net position data. (see Yeo and Johnson (2000)).

$$YeoJohnsonTransform(\beta, y_{h,t}) = \begin{cases} \frac{((1+y_{h,t})^{\beta}-1)}{\beta} & \text{if } \beta \neq 0, y_{h,t} \geq 0 \\ \log(1+y_{h,t}) & \text{if } \beta = 0, y_{h,t} \geq 0 \\ \frac{((1-y_{h,t})^{2-\beta}-1)}{(\beta-2)} & \text{if } \beta \neq 2, y_{h,t} < 0 \\ -\log(1-y_{h,t}) & \text{if } \beta = 2, y_{h,t} < 0. \end{cases} \tag{5}$$

Note that β is estimated and $y_{h,t}$ is standardised net position data.

3.2. Feature engineering and estimation scheme

As previously stated, this study will apply several machine learning (ML) techniques to the NetPositionTimeSeries dataset. This requires us to consider the features used by the ML algorithms in order to augment their performance. The dataset is relatively noisy as

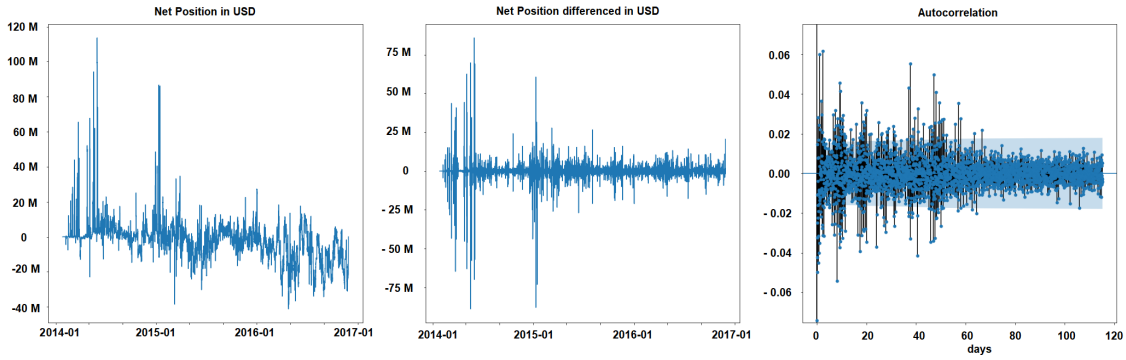

Figure 3: Visualisation of the NetPositionTimeSeries dataset. The left plot shows the net position time series that we are trying to forecast in this study. The first order difference in the time series is shown in the middle plot. The right plot shows the auto-correlation of the time series to itself given window sizes up to 120 days.

described above, therefore we chose to generate a few features for the ML's to work with. For instance, analysis of the average net position by hour over the course of a week (Figure 4) shows obvious seasonality with respect to variance. One can ascertain that changes in position are linked to time of day and day of week. In the currency markets worldwide, there are three main trading sessions which are reflected in the weekday plots in Figure 4. We can observe:

- a gradual increase in activity as Asia trading hours commence (22:00 previous day to 06:00 next day UTC);
- a noticeable accumulation in activity as the trading day in London / Europe commences (06:00 - 16:00 UTC);
- activity becomes busier still as trading in the USA commences (16:00 - 22:00).

Taking into account the nature of NetPositionTimeSeries discussed above, we included the following extra derived features (which are also supplied in Lindsay (2020)):

- Hour;
- Weekday;
- Four binary variables denoting if a given market is open or closed;
- Time until opening and closure of every market, and
- Lagged net position for the last 24 hours.

Note that the latter features are discarded for the LSTM model (which is introduced later), since it can learn dependencies ranging over long time intervals.

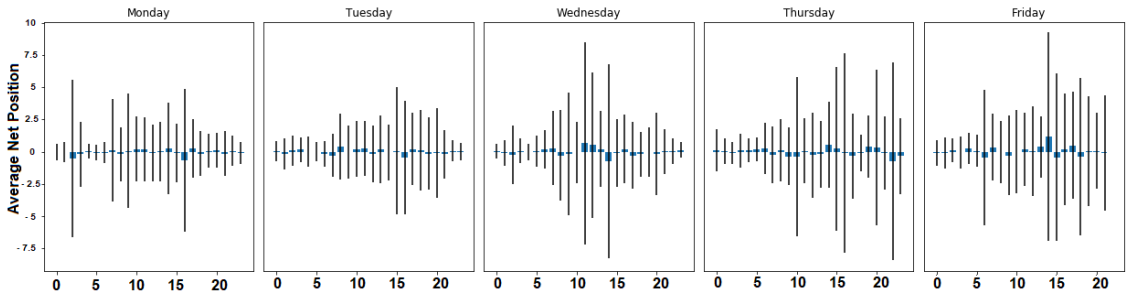

Figure 4: Average Net Position in millions of USD on the hourly basis for every day of the week. The vertical bars accounts for one standard deviation which is an evidence of presence of important noise in the dataset.

We split the resulting dataset into training and testing with a ratio of 80/20. We apply the out of rolling sample prediction scheme (see Figure 1) where a model is re-trained and re-calibrated every 5 days. On each epoch the training dataset spans over 14k hours and is used for prediction every hour for the next 5 days. Moreover we tuned the hyper-parameters using walk forward cross-validation with random grid search method. Parameters which minimised so called Quality-Driven Loss Function (Pearce et al. (2018)) for each p-value α were chosen:

$$Loss_{QD} = MPIW_{capt} + \frac{n}{\alpha(1-\alpha)} max(0, (1-\alpha) - PICP)^2$$

where:

$$\begin{aligned}
MPIW_{capt} &= \frac{1}{c}\sum_{t\in\mathcal{D}_{train}}(\hat{U}_{t,h}(\alpha) - \hat{L}_{t,h}(\alpha)) \cdot k_i \\
PICP &= \frac{c}{|(\mathcal{D}_{train}|} \\
c &= \sum_{t\in\mathcal{D}_{train}} k_t \\
k_i &= \mathbb{1}_{\{\hat{L}_{t,h}(\alpha) \le y_{t,h} \le \hat{U}_{t,h}(\alpha)\}} \\
\hat{L}_{t,h}(\alpha) &: \text{ predicted lower confidence interval} \\
\hat{U}_{t,h}(\alpha) &: \text{ predicted upper confidence interval}
\end{aligned}$$

This loss was designed to minimise the width of the confidence intervals subject to capturing a desired proportion of observations (i.e. $1-\alpha$).

3.3. Prediction models

This section gives a brief overview of the diverse ML models tested in our study. Implementations were carried out in Python using scipy and keras libraries, applying the Conformal Prediction (CP), Normalised Conformal Prediction (NCP) and Quantile Regression (Q) versions of the following ML models:

- K-nearest neighbours: simply outputs the average of the values of k nearest neighbors with respect to a given distance metric.

- Lasso linear regression: is a regularised version of a regression model. It can reduce model complexity and prevent over-fitting by adding a constraint (l1 loss) on the weights therefore penalising it. It is known feature is that it regularises and performs variable selection.
- Gradient Boosting: is an ensemble technique in which the predictors are trained sequentially (the error of one stage is passed as input into the next stage).
- Random forest: another ensemble technique where the prediction in the average output of many regression trees.
- LSTM (Long Term Short Term Memory): is a Recurrent Neural Networks (RNN) designed to handle sequence dependencies, therefore it a popular choice for time series prediction. A LSTM network enables learning long term dependencies thanks to its so called "gates" that decide which information to remember and which to forget. A bidirectional LSTM (BiLSTM) is an extention where two LSTMs are applied in each direction, i.e. the algorithm learns on the input sequence and its reverse. In our experiments we use a one layer BiLSTM with tanh activation functions followed up by a dropout and dense layer.

3.4. Benchmark

Moving averages (MA) are widely used in financial forecasting (more commonly to predict prices) with the full understanding of their limitations of being a crude lagging indicator of movement. The prediction for the next time epoch is the rolling mean and the confidence interval is constructed by adding and subtracting the respective rolling standard deviation multiplied by a constant. We assume that these forecasts are normal α quantiles i.e. 95% is 2 standard deviations (stdev) away from the mean, 99.7% is 3 stdev etc. We can expect that the position in the very distant past is less relevant than the position one hour ago. Moreover as demonstrated earlier in Figure 3 there are regime shifts in volatility therefore a well chosen window size is key for the moving average. A badly chosen window would negatively impact the interval estimates and one would expect a trade-off between mean/width and desired coverage rate of estimated intervals. Figure 8 provides an illustration of this trade-off where the performance of MA with respect to various window sizes is tested. We see that, at some point, width and standard deviation become too significant hence for benchmark purposes a window of 100 hours was selected.

Figure 5 shows the net position confidence predictions (upper and lower bound predictions at a 95% confidence level) from May 2016 to Dec 2016 for Linear Regression along with its meta-learning overlays: Q (shown in red), CP (grey) and NCP (green). Also shown is the best benchmark 100-hour MA (blue). As expected it is apparent that the MA predictions lag behind the actual net position (indicated by the thick black line). Briefly we also observe variability in the various implementations of the Linear Regression model. CP predictions (grey) are wider than the Q predictions (red), the latter being a little too tight to the true position (black) which deviates outside of this.

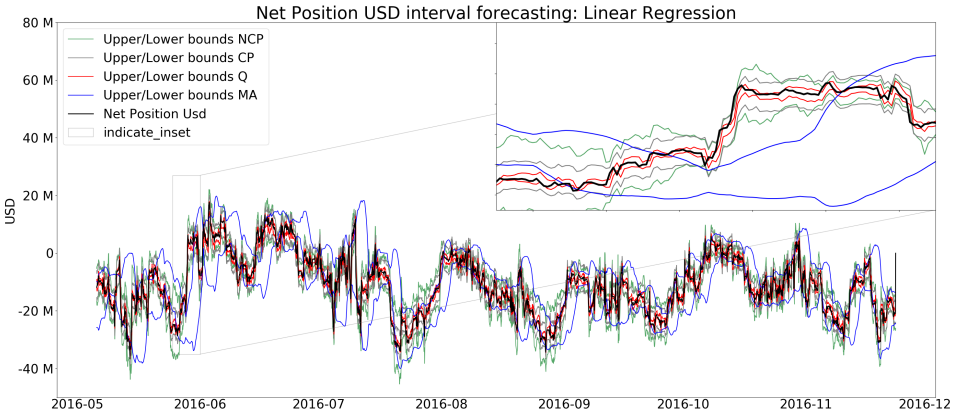

Figure 5: Example of both Linear Regression (with Q, CP and NCP meta-learning overlays) and the benchmark 100 hour Moving Average upper- and lower-bound predictions at a 95% confidence level, for the period May 2016 to Dec 2016. The black line represents the actual net position. For readability, a magnified portion of the plot is shown inset for predictions for late May 2016

3.5. Performance measures

In this section we will present the measures, statistical tests and losses which were used to assess forecasting performance. Losses are commonly used in interval forecasting or in back-testing Value-At-Risk models. Intuitively, one would consider width histogram or coverage rate as appropriate measures of performance. We define coverage rate as follows:

$$Coverage = \frac{\sum_{t \in \mathcal{D}_{test}} \mathbb{1}_{\{\hat{L}_{t,h}(\alpha) \leq y_{t,h} \leq \hat{U}_{t,h}(\alpha)\}}}{|\mathcal{D}_{test}|}$$

where the true value $y_{t,h} \in \mathcal{D}_{test}$, $\hat{L}_{t,h}(\alpha)$ is the lower limit and $\hat{U}_{t,h}(\alpha)$ is the upper limit of a forecasting interval. The higher the coverage rate, the more desirable the prediction model.

The width of an interval forecast $W_{t,h}$ is defined as follows:

$$W_{t,h} = \hat{U}_{t,h}(\alpha) - \hat{L}_{t,h}(\alpha)$$

In practice, the coverage rate and width interval measures may lead to controversial evaluations since a good coverage rate does not indicate if the width of confidence intervals is optimised. Therefore the Winkler score (Winkler (1994)) is preferred. For $(1-\alpha)100\%$ prediction interval, it is defined as follows:

$$WinklerScore = \begin{cases} W_{t,h} & \text{if } \hat{L}_{t,h}(\alpha) \leq y_{t,h} \leq \hat{U}_{t,h}(\alpha) \\ W_{t,h} + 2 \cdot \frac{\hat{L}_{t,h}(\alpha) - y_{t,h}}{\alpha}, & \text{if } \hat{L}_{t,h}(\alpha) \geq y_{t,h} \\ W_{t,h} + 2 \cdot \frac{y_{t,h} - \hat{U}_{t,h}(\alpha)}{\alpha}, & \text{if } \hat{U}_{t,h}(\alpha) \leq y_{t,h} \end{cases}.$$

The Winkler score gives a penalty if $y_{t,h}$ is outside the prediction interval and a lower score indicates a better prediction interval.

A wide variety of tests have been proposed to evaluate the performance of VaR models, which will be useful in our case. These VaR performance tests can be classified into two groups, those based on any statistical test and those based on the loss function. The former only show whether VaR estimates are accurate hence are limited in use because they do not allow for comparison of models.

A well-known tool, that estimates the conditional quantile, is the so-called pinball loss. It is usually used in quantile regression and is quantile specific. We average pinball losses together (for quantiles $\frac{\alpha}{2}$ and $1 - \frac{\alpha}{2}$ and taking the average of the latter as the final result) in order to calculate the loss for a prediction interval. The pinball loss is defined as follows:

$$Loss^{\alpha}_{pinball}(\hat{q}_{y_{t,h}}(\alpha), y_{t,h}) = \begin{cases} (1-\alpha)(\hat{q}_{y_{t,h}}(\alpha) - y_{t,h}) & \text{for } y_{t,h} < \hat{q}_{y_{t,h}}(\alpha) \\ \alpha(y_{t,h} - \hat{q}_{y_{t,h}}(\alpha)) & \text{for } y_{t,h} \geq \hat{q}_{y_{t,h}}(\alpha), \end{cases}$$

where $\hat{q}_{y_{t,h}}(\alpha)$ is the α-th estimated quantile of the hourly net position time series $y_{t,h}$. Note to optimise the pinball loss, the actual value series needs to be less than quantile prediction 100α percent of the time.

The unconditional coverage test, the conditional coverage test and the independence test of Christoffersen (1998) are the most common backtesting procedures for VaR model evaluation. In order to implement these statistical tests a so-called exception binary variable has to be defined:

$$I(\hat{L}_{t,h}(\alpha), \hat{U}_{t,h}(\alpha), y_{t,h}) = \begin{cases} 0 & \text{for } \hat{L}_{t,h}(\alpha) \leq y_{t,h} \leq \hat{U}_{t,h}(\alpha) \\ 1 & \text{otherwise} \end{cases}$$

We have an exception when the observation is outside the predicted interval.

The unconditional coverage test assumes that an accurate interval provides an unconditional coverage (UC), i.e.

$$\mathbb{P}[I(\hat{L}_{t,h}(\alpha), \hat{U}_{t,h}(\alpha), y_{t,h}) = 1] = \alpha$$

The conditional coverage(CC) test proposed by Christoffersen (1998) jointly examines whether the model generates a correct proportion of failures and whether the exceptions are statistically independent of one another i.e.

$$\mathbb{P}[I(\hat{L}_{t,h}(\alpha), \hat{U}_{t,h}(\alpha), y_{t,h}) = 1 | \mathcal{F}_{t-1})] = \alpha$$

where $\mathcal{F}_{t-1}$ denotes the set of information available at time $t-1$. The independence property means that past exceptions should not be informative about current and future exceptions.

4. Experiments and Results

4.1. Calibration of prediction intervals

As previously stated, the CP framework provides guarantees for a given error rate α under the exchangeability assumption. These calibration charts in Figure 6 give a good overview of whether the models tested are well calibrated as well as the effectiveness of their predictions. We not only want the predictions to be well calibrated but as tight as possible (i.e. the upper and lower bounds of the prediction intervals as small as possible). For each of the models, we evaluated their prediction intervals using 19 different confidence levels (5% to 95%). Unsurprisingly, MA (top left of Figure 6) was not calibrated for any confidence level, and had the widest prediction intervals with 15mio USD at 95% confidence. As shown earlier in Figure 5, the Linear Regression models tended to predict very tight prediction intervals $< 2.5\%$ at 95% confidence for the Q implementation, this results in the error rate deviating above the calibration line. Application of CP does make Linear Regression results calibrated yet at the cost of widening the prediction intervals to 10mio USD.

With the K-Nearest Neighbours and Decision Tree results we see the best calibration with Q overlay, with CP/NCP showing slight deviation above the diagonal line of calibration. Interestingly the region widths are wider than that of the benchmark MA model, however the Decision Tree predictions are reasonably tight apart from the NCP Decision Tree result of 19mio USD at 95% confidence. The best results are demonstrated with the more sophisticated ML techniques - Gradient Boosting and LSTM - both of which show good calibration for all meta learning overlays as well as tight region widths $<$ 10mio USD up to 95% confidence.

Further insight into the comparison of the prediction intervals of each ML method is provided by the box plots in Figure 7. These offer us a view of the standard deviation of the prediction intervals, rather than just the means as shown in Figure 6. With few exceptions the benchmark 100 hour MA under-performs and most algorithms have significantly better average and standard deviation of forecast interval widths. We notice that for all ML models except for K-Nearest Neighbours, the NCP overlay gives the largest standard deviation. We can see most of the box plots for NCP and Q meta learners show outliers on the larger region widths blowing out to 40mio USD.

4.2. Reliability of prediction intervals

Table 1 captures all considered performance measures at significance level $\alpha = 0.05$. The table is sorted with respect to the average width of prediction intervals. The top three models for each performance measure are highlighted in bold. It appears that the two models that come up on top are Q Gradient Boosting and CP Linear Regression (as seen in Figure 5). The CP Linear Regression manifests the lowest standard deviation which suggests that the other models are over-fitting. We believe this is due to the high noise level in the data therefore it makes sense that the model with the lowest complexity generalises the best. Moreover the CP framework yields narrower prediction interval which is probably why the NCP normalisation models overestimate the uncertainty of the prediction error. For comparison between all algorithms with respect to the Winkler score and pinball loss for each ML model at different confidence levels, see Figure 11 and Figure 10.

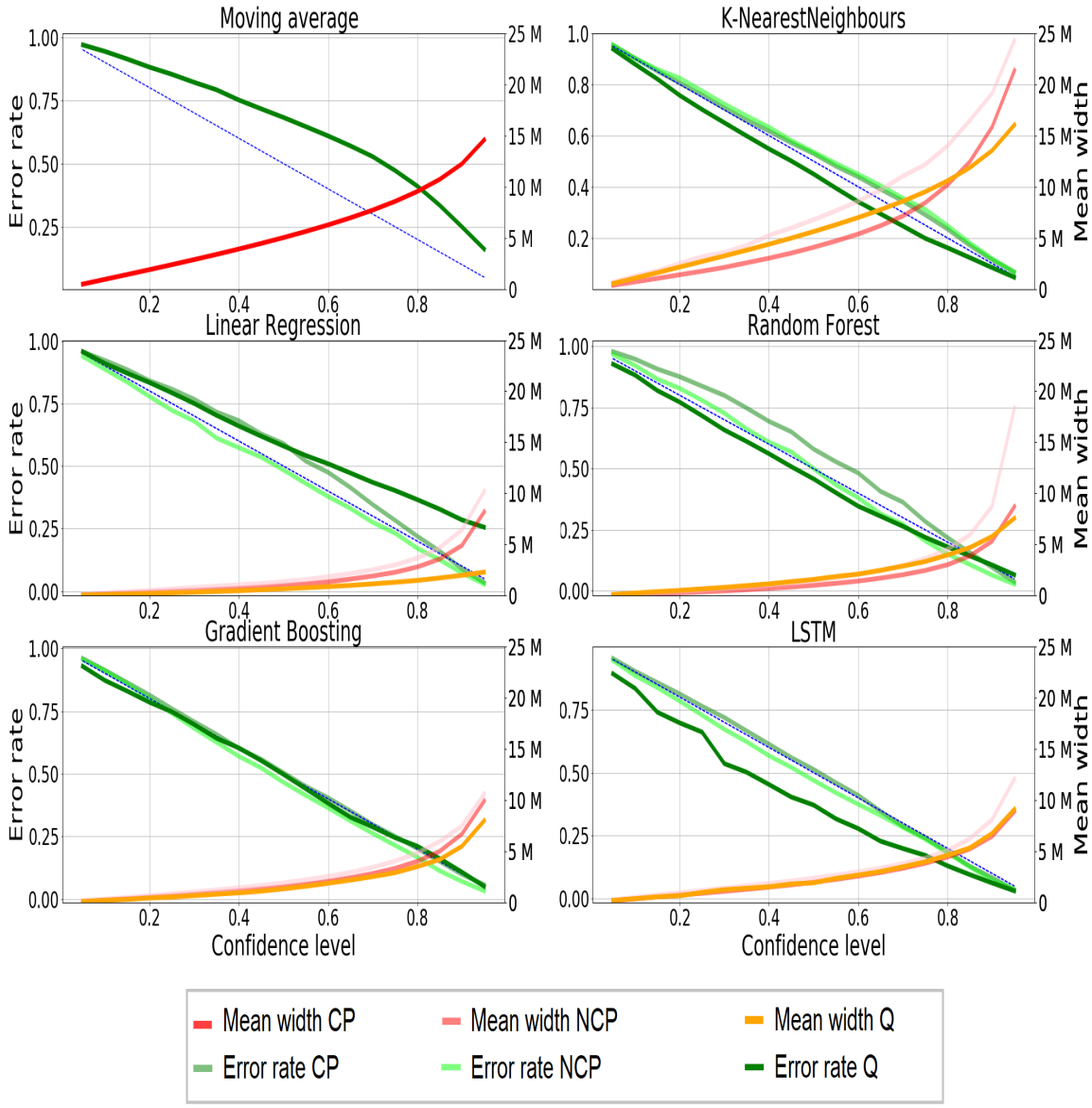

Figure 6: Plots charting the error rate (left vertical) and mean region prediction width (right vertical) for various confidence levels (horizontal axes) for each of the ML models (with different meta-learning overlays) tested. The dashed diagonal line shows the ideal level of calibration: the goal is for the actual error rate (green lines) to be below this. The orange lines show the average width of the region predictions at each confidence level; intuitively higher confidence requires larger confidence regions. The 100 hour MA benchmark is shown in the top left.

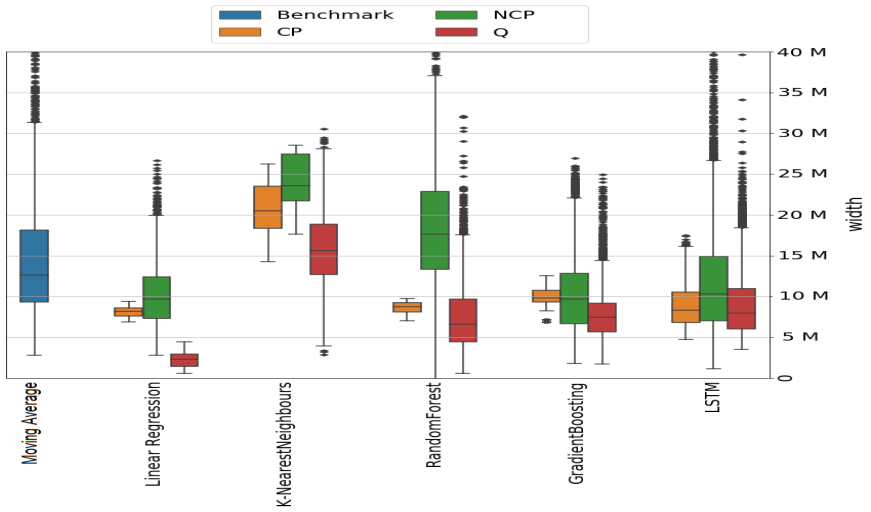

Figure 7: Box plots of the prediction interval width distributions for the various ML techniques tests. The 100 hour MA benchmark is shown in blue (leftmost), with all ML models, CP, NCP and Q versions are coloured orange, green and red respectively.

Table 1: Table summarising the experimental results for 95% theoretical confidence level for all ML models and their meta-learning overlays. Coverage scores marked in red denotes predictors with empirical coverage lower than 95% (i.e. not calibrated). Scores in bold font indicate top three predictors for a given performance measure.

Predictor	Coverage	Mean width	Std width	Winkler	Pinball	Type
Linear Regression	74%	$\mathbf{2.30{\cdot}10^6}$	$\mathbf{8.25{\cdot}10^5}$	$1.58{\cdot}10^7$	$1.97{\cdot}10^5$	Q
RandomForest	93%	$\mathbf{7.57{\cdot}10^6}$	$4.20{\cdot}10^6$	$1.22{\cdot}10^7$	$1.52{\cdot}10^5$	Q
GradientBoosting	95%	$\mathbf{8.00{\cdot}10^6}$	$3.45{\cdot}10^6$	$\mathbf{1.13{\cdot}10^7}$	$\mathbf{1.41{\cdot}10^5}$	Q
Linear Regression	97%	$8.20{\cdot}10^6$	$\mathbf{6.58{\cdot}10^5}$	$\mathbf{1.13{\cdot}10^7}$	$\mathbf{1.42{\cdot}10^5}$	CP
RandomForest	96%	$8.72{\cdot}10^6$	$\mathbf{6.76{\cdot}10^5}$	$1.21{\cdot}10^7$	$1.51{\cdot}10^5$	CP
LSTM	97%	$8.89{\cdot}10^6$	$2.59{\cdot}10^6$	$1.19{\cdot}10^7$	$1.49{\cdot}10^5$	CP
LSTM	97%	$9.12{\cdot}10^6$	$4.26{\cdot}10^6$	$\mathbf{1.18{\cdot}10^7}$	$\mathbf{1.48{\cdot}10^5}$	Q
GradientBoosting	94%	$9.93{\cdot}10^6$	$1.27{\cdot}10^6$	$1.44{\cdot}10^7$	$1.80{\cdot}10^5$	CP
Linear Regression	97%	$1.02{\cdot}10^7$	$3.94{\cdot}10^6$	$1.23{\cdot}10^7$	$1.54{\cdot}10^5$	NCP
GradientBoosting	96%	$1.06{\cdot}10^7$	$5.22{\cdot}10^6$	$1.30{\cdot}10^7$	$1.63{\cdot}10^5$	NCP
LSTM	96%	$1.21{\cdot}10^7$	$7.39{\cdot}10^6$	$1.46{\cdot}10^7$	$1.83{\cdot}10^5$	NCP
MA	84%	$1.46{\cdot}10^7$	$7.66{\cdot}10^6$	$2.64{\cdot}10^7$	$3.29{\cdot}10^5$	Benchmark
K-NearestNeighbours	95%	$1.60{\cdot}10^7$	$4.33{\cdot}10^6$	$1.93{\cdot}10^7$	$2.42{\cdot}10^5$	Q
RandomForest	97%	$1.83{\cdot}10^7$	$7.56{\cdot}10^6$	$1.96{\cdot}10^7$	$2.45{\cdot}10^5$	NCP
K-NearestNeighbours	93%	$2.14{\cdot}10^7$	$3.47{\cdot}10^6$	$2.81{\cdot}10^7$	$3.52{\cdot}10^5$	CP
K-NearestNeighbours	93%	$2.43{\cdot}10^7$	$3.18{\cdot}10^6$	$3.07{\cdot}10^7$	$3.83{\cdot}10^5$	NCP

4.3. Statistical evaluation of interval forecasts

We apply Christoffersen (1998) tests in order to examine unconditional coverage (UC), and conditional coverage (CC). The results are reported in the Table 2. UC tests for the true coverage while CC takes clustering effect into account. It is worth mentioning that CC is a joined unconditional coverage test and independence test that simultaneously checks if the percentage of exceptions is correct and if the exceptions are independent.

We remark that across all significance levels, CP, LSTM and Gradient Boosting have a high pass rate ($> 70\%$) of the UC test, which suggests their exact calibration (see Figure 9 to inspect deviation from the significance levels). For NCP the pass rates are significantly lower i.e. in the $10 - 20\%$ range. For Q, only Gradient Boosting has a high pass rate. On the other hand, CC tests are widely rejected across all significance levels. This suggests the presence of clustering of errors. This could be explained by high volatility and unpredictability of the net position caused by periodic high client trading activity as previously described.

Table 2: Statistical coverage test results. The percentage score is the average success rate of passed tests across all considered significance levels (19 levels from 0.05 to 0.95).

Predictor	Type	UC	CC
Gradient Boosting	CP	79%	0%
Gradient Boosting	NCP	21%	5%
Gradient Boosting	Q	68%	0%
K-Nearest Neighbours	CP	0%	0%
K-Nearest Neighbours	NCP	0%	0%
K-Nearest Neighbours	Q	5%	0%
LSTM	CP	74%	0%
LSTM	NCP	11%	0%
LSTM	Q	0%	0%
Linear Regression	CP	11%	0%
Linear Regression	NCP	11%	0%
Linear Regression	Q	11%	0%
MA	Benchmark	0%	0%
RandomForest	CP	11%	0%
RandomForest	NCP	21%	0%
RandomForest	Q	11%	0%

5. Conclusion and perspectives

In this paper we applied Conformal Prediction (CP) to a financial time series problem. We explained the practical rationale underpinning our study which is the value that a market maker (MM) places in being able to estimate its future net position with some degree of confidence. Forecasting net positions has significant implications for a MM being able to both maximise its profit and mitigate risk. We used a novel time series dataset (NetPositionTimeSeries - Lindsay (2020)) which consists of hour-by-hour net positions of a MM over a three-year period starting from Jan 2014. To forecast net positions, we tested

a range of ML algorithms where CP works like a secondary layer and outputs confidence intervals to the forecast. Through comparison with a basic benchmark (rolling mean) and quantile regression we showed that CP, in most cases, is well calibrated throughout all significance levels and lives up to the expectations of delivering valid prediction intervals. In terms of performance measures such as mean and standard deviation of the width of prediction intervals, Winkler score and pinball loss, CP yields comparable or slightly better results than quantile regression.

It is worth pointing out that on average quantile regression yielded narrower prediction intervals than NCP (see Figure 7) hence ideally one would investigate combining quantile prediction with CP framework. A recent study by Romano et al. (2019) proposed this idea, however when we carried out preliminary implementation of this so called conformalised quantile regression with our data we did not observe any significant improvement in predictive performance.

In carrying out this work we have along the way identified several relevant questions which future research could help in answering:

- What is the impact of altering the size of the training and calibration set on predictive performance?
- How does the performance of the ML forecasts compare to those of classical time series models?
- What would be the effect of drilling down into the problem, e.g. can we consider prediction of long and short positions separately?
- Can we add relevant exogenous data in order to improve the accuracy of prediction intervals and therefore detect big jumps of the net position more precisely?

Acknowledgments

We would like to give special thanks to Bibi-Rehana Lindsay and the team at AlgoLabs and Equiti Capital UK for their support and guidance throughout this project.

References

V.N. Balasubramanian, S-S. Ho, and V. Vovk. *Conformal Prediction for Reliable Machine Learning: Theory, Adaptations and Applications - Chapter 9: Other Adaptations*. Morgan Kaufmann Publishers Inc., San Francisco, CA, 1st edition, 2014. ISBN 0123985374.

V. Chernozhukov, K. Wuthrich, and Y. Zhu. An exact and robust conformal inference method for counterfactual and synthetic controls. Technical report, Massachusetts Institute of Technology, 2019. URL https://arxiv.org/pdf/1712.09089v7.pdf.

P.F. Christoffersen. Evaluating interval forecasts. *International Economic Review*, 39(4): 841–862, 1998.

M. Dashevskiy and Z. Luo. Time series prediction with performance guarantee. *IET Communications*, 5:1044–1051, May 2011.

A. Gammerman, V. Vovk, and V. Vapnik. Learning by transduction. *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 148 – 155, 1998.

U. Johansson, H. Boström, T. Löfström, and H. Linusson. Regression conformal prediction with random forests. *Machine Learning*, 97:1–22, October 2014.

C. Kath and F. Ziel. Conformal prediction interval estimations with an application to day-ahead and intraday power markets. Technical report, University Duisburg-Essen, 2019. URL https://arxiv.org/pdf/1905.07886.pdf.

D. Lindsay. Net position time series dataset. *Kaggle*, July 2020. URL https://www.kaggle.com/davidlindsay1979/net-position-time-series-dataset.

T. Pearce, A. Brintrup, M. Zaki, and A. Neely. High-quality prediction intervals for deep learning: A distribution-free, ensembled approach. *Proceedings of the 35th International Conference on Machine Learning*, pages 4075–4084, 2018.

Y. Romano, E. Patterson, and E.J. Candès. Conformalized quantile regression. *33rd Conference on Neural Information Processing Systems*, 2019. URL https://papers.nips.cc/paper/8613-conformalized-quantile-regression.pdf.

V. Vovk, A. Gammerman, and G. Shafer. *Algorithmic Learning in a Random World*. Springer, Boston, MA, 1st edition, 2005. ISBN 9780387250618.

R.L. Winkler. Evaluating probabilities: Asymmetric scoring rules. *Management Science*, 40:1395–1405, November 1994.

I-K. Yeo and R. Johnson. A new family of power transformations to improve normality or symmetry. *Biometrika*, 87:954–959, December 2000.

Appendix

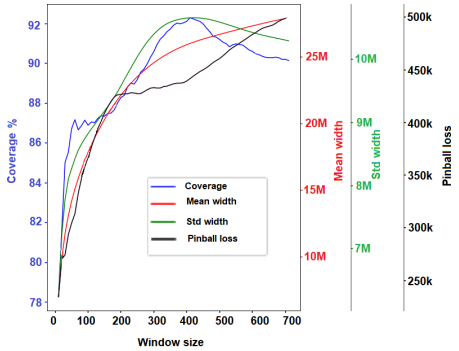

Figure 8: Performance of the moving average

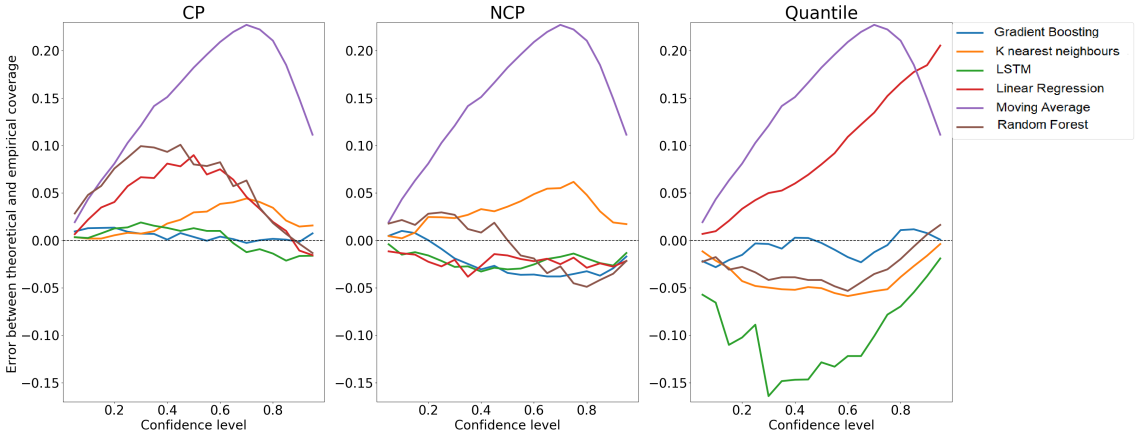

Figure 9: Difference between theoretical and empirical coverage across all confidence levels for all interval prediction methods.

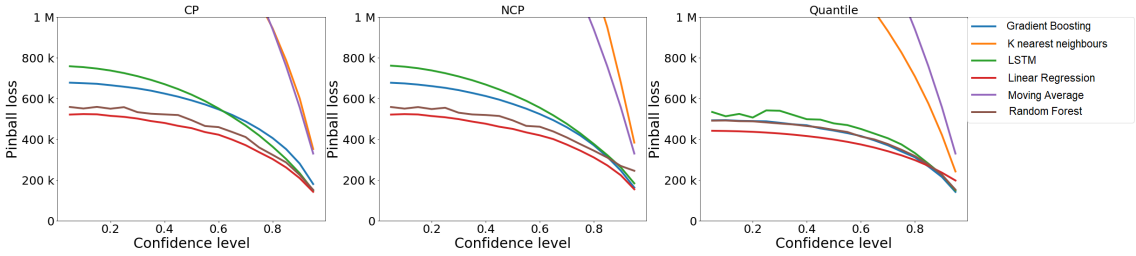

Figure 10: Pinball loss

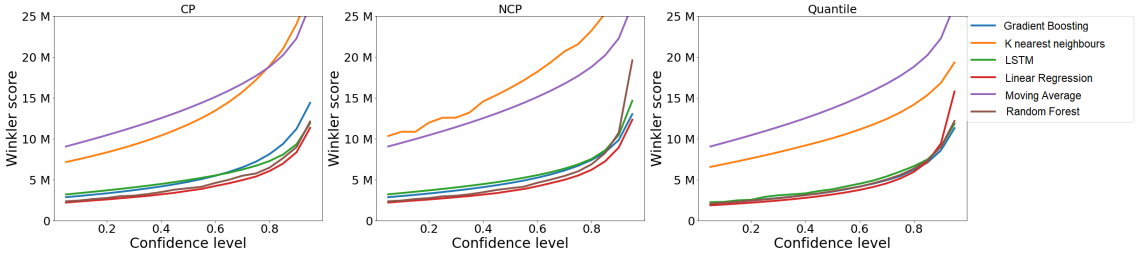

Figure 11: Winkler score