

Peeking inside FFORMS: Feature-based FOfRecast Model Selection

Abstract

Features of time series are useful in identifying suitable models for forecasting. Talagala, Hyndman & Athanasopoulos (2018) proposed a classification framework, labelled FFORMS (Feature-based FOfRecast Model Selection), which selects forecast models based on features calculated from the time series. The FFORMS framework builds a mapping that relates the features of a time series to the “best” forecast model using a random forest. In this paper we explore what is happening under the hood of the FFORMS framework. This is accomplished using model-agnostic machine learning interpretability approaches. The analysis provides a valuable insight into how different features and their interactions affect the choice of forecast model.

Keywords: forecasting, time series, machine learning interpretability, black-box models, LIME

1 Introduction

The field of time series forecasting has been evolving for a few decades now and has introduced a wide variety of models for forecasting. However, for a given time series the selection of an appropriate forecast-model among many possibilities is not straight forward. This selection is one of the most difficult tasks as each method perform best for some but not all tasks. The features of a time series are considered to be an important factor in identifying suitable forecasting models (Collopy & Armstrong 1992; Meade 2000; Makridakis & Hibon 2000; Wang, Smith-Miles & Hyndman 2009). However, a comprehensive description of the relationship between the features and the performance of algorithms is rarely discussed.

There have been several recent studies on the use of meta-learning approaches to automate forecast-model selection based on features computed from the time series (Shah 1997; Prudêncio & Ludermir 2004; Lemke & Gabrys 2010; Kück, Crone & Freitag 2016). A meta-learning approach provides a systematic guidance on model selection based on knowledge acquired from historical

data sets. The key idea is, forecast-model selection is posed as a supervised learning task. Each time series in the meta-data set is represented as a vector of features and labelled according to the “best” forecast-model (i.e. for example model with lowest MASE over a test set, etc.). Then a meta-learner is trained to identify a suitable forecast-model (usually a machine learning algorithm is used). With these existing frameworks, the question that must be raised is: “Can we trust the Machine Learning based frameworks if we don’t know how it works?” In the era of big data, such an automated model selection process is necessary because the cost of invoking all possible forecast-models is prohibitive. However, the existing literature suffers from the limitation of providing answers to questions such as: i) How features are related to the property being modeled?; ii) How features interact with each other to identify a suitable forecast-model?; iii) Which features contribute most to the classification process? etc. Addressing such questions can enhance the understanding of the relations between features and model selection outcomes. To the best of our knowledge, a very limited effort has been taken to understand how the meta-learners are making its decisions and what is really happening inside these complex model structures. Providing transparency will result, building trust in the prediction results of the meta-learner.

Furthermore, besides the goal of developing an automated forecast-model selection framework very few researchers have made an attempt to provide a description of the relationship between the features and the choice of different forecast-models (Schnaars 1984; Wang, Smith-Miles & Hyndman 2009; Lemke & Gabrys 2010; Petropoulos et al. 2014, are among some exceptions). These studies are limited by the scale of problem instances used, the diversity of forecast-models implemented, and the limited number of features considered to identify the relationship between features and forecast-model performance.

To fill this gap, this paper makes a first step towards providing a comprehensive analysis of the relationship between time series features and forecast-model selection using machine learning interpretability techniques. This paper builds on the method from our previous work Talagala, Hyndman & Athanasopoulos (2018), in which we introduced the FFORMS (Feature-based FOfRecast Model Selection) framework. A random forest is used to model the relationship between features and “best” performing forecast-model. A large collection of time series is used to train the meta-learner.

In this article, we make the following contributions:

1. We extend the FFORMS framework to handle weekly, daily and hourly series. We also extend the diversity of forecast-models used as class labels. The contribution of this paper

differs from previously published work related to meta-learning (Prudêncio & Ludermir 2004; Lemke & Gabrys 2010; Kück, Crone & Freitag 2016) in three ways: i) a more extensive collection of features is used (features to handle multiple seasonality in hourly and daily series), ii) the diversity of forecast-models considered as class labels, and iii) capability of handling high frequency data;

2. We analyse the application of the FFORMS framework to the M4-competition data. We generated point forecasts and prediction intervals for the M4-competition time series data, and is shown to yield accurate forecast comparable to several benchmarks and other commonly used automated approaches of time series forecasting. Our approach achieved a high accuracy rate based on individual forecast-model selection rule.
3. The main contribution of the paper is to explore what is happening under the hood of the FFORMS framework and thereby gain an understanding of the relationship between features of time series and the choice of forecast-model selection using the FFORMS framework. We call it “Peeking inside FFORMS”. We try to answer the following questions:
 - i) **Which** features are the most important?
 - ii) **Where** (overall classification or only within specific classes) are they important? ?
 - iii) **How** are they important?
 - iv) **When** and **how** are features linked with the prediction outcome?
 - v) **When** and **how strongly** do features interacts with other features?

The remainder of the paper is structured as follows. The [Section 2](#) outlines the extended FFORMS framework. [Section 3](#) through [Section 5](#) explain the three main components of the extended FFORMS framework. Results are discussed in [Section 6](#) and [Section 7](#). [Section 8](#) concludes.

2 Methodological framework

[Figure 1](#) shows the extended FFORMS framework with all additional components highlighted in yellow. There are three main components of this extended FFORMS framework. They are,

1. offline phase (blue);, the development of a meta-learner
2. online phase (red): use the pre-trained meta-learner to identify the “best” forecast-model.
3. peeking inside FFORMS (yellow): gain insights into what is happening under the hood of the FFORMS framework.

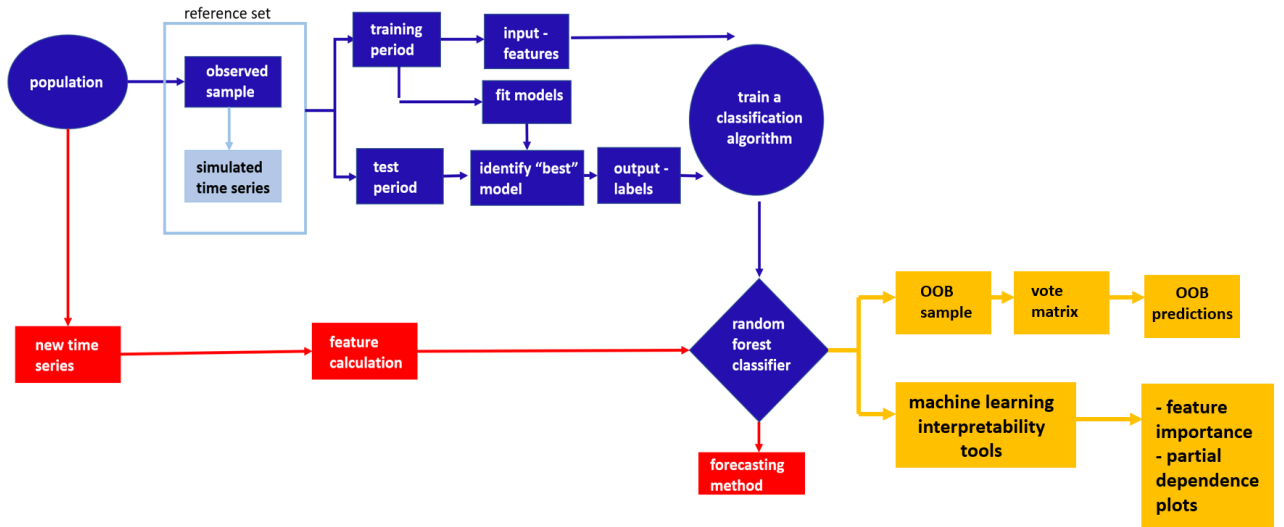


Figure 1: extended FFORMS (Feature-based FOREcast-Model Selection) framework. The offline phase is shown in blue, the online phase by red and the peeking inside FFORMS is shown in yellow.

3 Offline phase

We now explain application of the FFORMS framework to the M4-competition data. We analyse yearly, quarterly, monthly, weekly, daily and hourly series separately due to their differences in frequencies and hence the appropriateness of different features and forecast-models we considered as class labels.

3.1 Reference set

We call the collection of time series used for training the meta-learner as the “reference set”. The reference set consist of two sets of time series: i) observed sample and ii) simulated time series.

3.1.1 Observed sample

We use the time series from the M1, M3 and M4 competitions as the observed sample. Note that from the M4 competition a randomly selected subset of time series are used for the observed sample. The rest is used as a validation set to evaluate the performance of the meta-learner. Note that, we did not use the test period of the each series of the M4 competition to our observed sample or validation set as it was not available to the competitors during the running of the

M4-competition.

3.1.2 Simulated time series

As described in Talagala, Hyndman & Athanasopoulos (2018), we augment the reference set by adding multiple time series simulated based on each series in the M4 competition. We use several automated algorithms to simulate multiple time series. The `ets()` and `auto.arima()` functions available in the forecast package in R (Hyndman et al. 2018) are used to simulate yearly, quarterly and monthly data from exponential smoothing and ARIMA models. The `stlf()` function also available in the forecast package is used to simulate multiple time series based on multiple seasonal decomposition approach. Using the above functions, we fit models to each time series in the M4 database and then simulate multiple time series from the selected models. In this experiment the length of the simulated time series is set to be equal to: length of the training period specified in the M4 competition + length of the forecast horizon specified in the competition. For example, the series with id “Y13190” contains a training period of length 835. The length of the simulated series generated based on this series is equals to 841 (835+6). Before simulating time series from daily and hourly series, we convert the time series into multiple seasonal time series (msts) objects. For daily time series with length less than 366, the frequency set to 7 and longer time series are converted to multiple seasonal time series objects with frequencies set to 7 and 365.25. For hourly series, we set the frequencies to 24 and 168 to handle multiple frequencies corresponds to time-of-day pattern and time-of-week pattern respectively. We should re-emphasize that all the observed time series and the simulated time series form the reference set to build our meta-learner. Once we create the reference set for random forest training we split each time series in the reference set into training period and test period.

3.2 Input: features

The FFORMS framework operates on features calculated from the time series. For each time series in the reference set, features are calculated based on the training period of the time series.

The description of the features calculated under each frequency category is shown in Table 1. A comprehensive description of the features used in the experiment is given in Talagala, Hyndman & Athanasopoulos (2018).

Table 1: *Time series features*

	Feature	Description	Y	Q/M	W	D/H
1	T	length of time series	✓	✓	✓	✓
2	trend	strength of trend	✓	✓	✓	✓
3	seasonality_q	strength of quarterly seasonality	-	✓	-	-
4	seasonality_m	strength of monthly seasonality	-	✓	-	-
5	seasonality_w	strength of weekly seasonality	-	✓	-	-
6	seasonality_d	strength of daily seasonality	-	-	-	✓
7	seasonality_h	strength of hourly seasonality	-	-	-	✓
8	linearity	linearity	✓	✓	✓	✓
9	curvature	curvature	✓	✓	✓	✓
10	spikiness	spikiness	✓	✓	✓	✓
11	e_acf1	first ACF value of remainder series	✓	✓	✓	✓
12	stability	stability	✓	✓	✓	✓
13	lumpiness	lumpiness	✓	✓	✓	✓
14	entropy	spectral entropy	✓	✓	✓	✓
15	hurst	Hurst exponent	✓	✓	✓	✓
16	nonlinearity	nonlinearity	✓	✓	✓	✓
17	alpha	ETS(A,A,N) $\hat{\alpha}$	✓	✓	✓	-
18	beta	ETS(A,A,N) $\hat{\beta}$	✓	✓	✓	-
19	hwalpha	ETS(A,A,A) $\hat{\alpha}$	-	✓	-	-
20	hwbeta	ETS(A,A,A) $\hat{\beta}$	-	✓	-	-
21	hwgamma	ETS(A,A,A) $\hat{\gamma}$	-	✓	-	-
22	ur_pp	test statistic based on Phillips-Perron test	✓	-	-	-
23	ur_kpss	test statistic based on KPSS test	✓	-	-	-
24	y_acf1	first ACF value of the original series	✓	✓	✓	✓
25	diff1y_acf1	first ACF value of the differenced series	✓	✓	✓	✓
26	diff2y_acf1	first ACF value of the twice-differenced series	✓	✓	✓	✓
27	y_acf5	sum of squares of first 5 ACF values of original series	✓	✓	✓	✓
28	diff1y_acf5	sum of squares of first 5 ACF values of differenced series	✓	✓	✓	✓
29	diff2y_acf5	sum of squares of first 5 ACF values of twice-differenced series	✓	✓	✓	✓
30	sediff_seacf1	ACF value at the first seasonal lag of seasonally-differenced series	-	✓	✓	✓
31	sediff_acf5	sum of squares of first 5 autocorrelation coefficients of seasonally-differenced series	-	✓	✓	✓
32	seas_pacf	partial autocorrelation coefficient at first seasonal lag	-	✓	✓	✓
33	lmres_acf1	first ACF value of residual series of linear trend model	✓	-	-	-
34	y_pacf5	sum of squares of first 5 PACF values of original series	✓	✓	✓	✓
35	diff1y_pacf5	sum of squares of first 5 PACF values of differenced series	✓	✓	✓	✓
36	diff2y_pacf5	sum of squares of first 5 PACF values of twice-differenced series	✓	✓	✓	✓

3.3 Output: class-labels

The description of class labels considered under each frequency is shown in Table 2. Note that these added to Talagala, Hyndman & Athanasopoulos (2018). Most of the labels given in Table 2 are self-explanatory labels. In STL-AR, mstlets, and mstlarima, first an STL decomposition is applied to the time series and then seasonal naive is used to forecast the seasonal component. Then, AR, ETS and ARIMA models are used to forecast the seasonally adjusted data respectively. We fit the corresponding models outlined in Table 2 to each series in the reference set. The models are estimated using the training period for each series, and forecasts are produced for the test periods.

According to the *M4 Competitor's Guide* (2018), in the M4-competition the forecast accuracy is

Table 2: *Class labels*

class label	Description	Y	Q/M	W	D/H
WN	white noise process	✓	✓	✓	✓
AR/MA/ARMA	AR, MA, ARMA processes	✓	✓	✓	-
ARIMA	ARIMA process	✓	✓	✓	-
SARIMA	seasonal ARIMA	✓	✓	✓	-
RWD	random walk with drift	✓	✓	✓	✓
RW	random walk	✓	✓	✓	✓
Theta	standard theta method	✓	✓	✓	✓
STL-AR		-	✓	✓	✓
ETS-notrendnoseasonal	ETS without trend and seasonal components	✓	✓	✓	-
ETStrendonly	ETS with trend component and without seasonal component	✓	✓	✓	-
ETSdampedtrend	ETS with damped trend component and without seasonal component	✓	✓	-	-
ETStrendseasonal	ETS with trend and seasonal components	-	✓	-	-
ETSdampedtrendseasonal	ETS with damped trend and seasonal components	-	✓	-	-
ETSseasonalonly	ETS with seasonal components and without trend component	-	✓	-	-
snaive	seasonal naive method	✓	✓	✓	✓
tbats	TBATS forecasting	-	✓	✓	✓
nn	neural network time series forecasts	✓	✓	✓	✓
mstlets		-	-	✓	✓
mstlarima		-	-	-	✓

evaluated based on the mean Absolute Scaled Error (MASE) and the symmetric Mean Absolute Percentage Error (MAPE). Hence, in order to identify the “best” forecast-model for each time series in the reference set we combine MASE and the symmetric MAPE calculated over the test set. More specifically, for each series both forecast error measures MASE and sMAPE are calculated for each of the forecast models. Each of these is respectively standardized by the median MASE and median sMAPE calculated across the forecast-models. The model with the lowest average value of the scaled MASE and scaled sMAPE is selected as the output class-label. The last step of offline phase of the framewok is to train a meta-learner. A random forest algorithm is used to train the meta-learner.

4 Online phase

The online phase of the algorithm involves generating point forecasts and 95% prediction intervals for the new series or the future values observed time series. We calculate point forecasts and prediction intervals over the test period of M4-competition data. First, the corresponding features are calculated based on the full length of the training period provided by the M4 competition. Second, point forecasts and 95% prediction intervals are calculated based on the predicted class labels. We should note that all negative forecasts are set to zero.

5 Peeking inside FFORMS

The main objective of this paper is to explore the nature of the relationship between features and forecast-model selection learned by the FFORMS framework. More specifically, to identify which of the features are important for model predictions and how different features and their interactions led to the different choices. First, we explore the global role of features in the choice of different forecast-model selection. Global interpretability evaluates the behavior of a model on entire data set. Global perspective of model interpretation helps users to understand the overall modeled relationship between features and the FFORMS outcome. In the following subsections, we provide a description of tools we use to explore the global perspective of the FFORMS meta-learners.

5.1 Visualise patterns learned by the meta-learner

A useful by-product of random forest is Out-Of-Bag (OOB) observations. The observations not included in building a given tree is called OOB observations. In general, each bagged tree does not make use of around one third of observations to construct a tree. Furthermore, each tree is grown based on different bootstrap samples hence, each tree has different set of OOB observations. We use a vote matrix calculated based on OOB observations to visualise patterns learned by the random forest. The vote matrix ($N \times P$; N is total number of observations, P is number of classes) contains the proportion of times each observation was classified to each class based on OOB observations.

5.2 Feature importance

Jiang & Owen (2002) explains variable importance under three different views: i) causality: change in the value of Y for an increase or decrease in the value of x , ii) contribution of X based on out-of-sample prediction accuracy and iii) face value of X on prediction function g , for example in linear regression model estimated coefficients of each predictor can be considered as a measure of variable importance. See Jiang & Owen (2002) for comparable face value interpretation for machine learning models. In this paper we use the first two notions of variable importance. Partial dependency functions and individual conditional expectation curves are used to explore the “causality” notion of variable importance while Mean decrease in Gini coefficient and Permutation-based variable importance are used to capture the second notion of variable importance-features contribution to the predictive accuracy (Zhao & Hastie (2017)). We will introduce each of these variable importance measures below.

Let $\mathcal{P} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^N$ be the historical data set we use to train a classifier. Consider a p -dimensional feature vector $X = (X_1, X_2, \dots, X_p)$ and a dependent variable, the best forecasting method for each series Y . Let \mathcal{G} be the unknown relationship between X and Y . Zhao & Hastie (2017) term this as “law of nature”. Inside the FFORMS framework, random forest algorithm tries to learn this relationship using the historical data we provided. We denote the predicted function as g .

5.2.1 Mean decrease in Gini coefficient

Mean decrease in Gini coefficient is a measure of how each feature contributes to the homogeneity of the nodes and leaves in the resulting random forest proposed by Breiman (2001).

5.2.2 Permutation-based variable importance measure

The permutation-based variable importance introduced by Breiman (2001) measures the prediction strength of each feature. This measure is calculated based on the out-of-bag (OOB) observations. The calculation of variable importance is formalized as follow: Let $\mathcal{B}^{(k)}$ be the OOB sample for a tree k , with $k \in \{1, \dots, ntree\}$, where $ntree$ is the number of trees in the random forest. Then the variable importance of variable X_j in k^{th} tree is:

$$VI^{(k)}(X_j) = \frac{\sum_{i \in \mathcal{B}^{(k)}} I(\gamma_i = \gamma_{i, \pi_j}^k)}{|\mathcal{B}^{(k)}|} - \frac{\sum_{i \in \mathcal{B}^{(k)}} I(\gamma_i = \gamma_i^k)}{|\mathcal{B}^{(k)}|},$$

where γ_i^k denotes the predicted class for the i^{th} observation before permuting the values of X_j and γ_{i, π_j}^k is the predicted class for the i^{th} observation after permuting the values of X_j . The overall variable importance score is calculated as:

$$VI(X_j) = \frac{\sum_{t=1}^{ntree} VI^{(t)}(x_j)}{ntree}.$$

Permutation-based variable importance measures provide a useful starting point for identifying relative influence of features on the predicted outcome. However, they provide a little indication of the nature of the relationship between the features and model outcome. To gain further insights into the role of features inside the FFORMS framework we use partial dependence plot (PDP) introduced by Friedman, Popescu, et al. (2008).

5.2.3 Partial dependence plot (PDP) and Variable importance measure based on PDP

Partial dependence plot can be used to graphically examine how each feature is related to the model prediction while accounting for the average effect of other features in the model. Let X_s be the subset of features we want examine partial dependencies for and X_c be the remaining set of features in X . Then g_s , the partial dependence function on X_s is defines as

$$g_s(X_s) = E_{x_c}[g(x_s, X_c)] = \int g(x_s, x_c) dP(x_c).$$

In practice, PDP can be estimated from a training data set as

$$\bar{g}_s(x_s) = \frac{1}{n} \sum_{i=1}^n g(x_s, X_{iC}),$$

where n is the number of observations in the training data set. Partial dependency curve can be created by plotting the pairs of $\{(x_s^k, \bar{g}_s(x_{sk}))\}_{k=1}^m$ defined on grid of points $\{x_{s1}, x_{s2}, \dots, x_{sm}\}$ based on X_s . FFORMS framework has treated the forecast-model selection problem as a classification problem. Hence, in this paper partial dependency functions display the probability of certain class occurring given different values of the feature X_s .

Greenwell, Boehmke & McCarthy (2018) introduce a variable importance measure based on the partial dependency curves. The idea is to measure the “flatness” of partial dependence curves for each feature. A feature whose PDP curve is flat, relative to the other features, indicates that the feature does not have much influence on the predicted value as it changes while taking into account the average effect of the other features in the model. The flatness of the curve is measured using the standard deviation of the values $\{\bar{g}_s(x_{sk})\}_{k=1}^m$.

5.2.4 Individual conditional expectation (ICE) curves Variable importance measure based on ICE curves

While partial dependency curves are useful in understanding the estimated relationship between features and the predicted outcome in the presence of substantial interaction between features, it can be misleading. Goldstein et al. (2015) propose the Individual Conditional Expectation (ICE) curves to address this issue. Instead of averaging $g(x_s, X_{iC})$ over all observations in the training data, ICE plots the individual response curves by plotting the pairs $\{(x_s^k, g(x_{sk}, X_{iC}))\}_{k=1}^m$ defined on grid of points $\{x_{s1}, x_{s2}, \dots, x_{sm}\}$ based on X_s . In other words, partial dependency curve is simply the average of all the ICE curves.

This method is similar to the PDP-based variable importance scores above, but are based on measuring the “flatness” of the individual conditional expectation curves. We calculated standard deviations of each ICE curves. We then computed an ICE based variable importance score – simply the average of all the standard deviations. A higher value indicates a higher degree of interactivity with other features.

5.2.5 Ranking of features based on feature importance measures

To identify class-specific important features we rank features in three different ways: i) based on permutation-based variable importance, ii) based on partial dependence functions and iii) based on ICE-curves. We consider 25 features for yearly data. The feature that shows the highest importance is ranked 25, the second best is ranked 24, and so on. Finally, for each feature, a mean rank is calculated based on the rankings of the three measures. Similarly, the overall feature importance is evaluated based on the permutation-based variable importance measure and the Gini coefficient-based feature importance measure.

5.3 Relationship between most important features and the choice of forecast-model selection

The partial-dependence curves along with their confidence intervals are used to visualize the relationship between top features are the choice of forecast model selection.

5.4 Assessment of interaction effect

Friedman’s H-statistic (Friedman, Popescu, et al. (2008)) is used to test the presence of interaction between all possible pairs of features. This statistic is computed based on the partial dependence functions. For two-way interaction between two specific variable x_j and x_k , Friedman’s H-statistic is defined as follow,

$$H_{jk}^2 = \sum_{i=1}^n [\bar{g}_s(x_{ij}, x_{jk}) - \bar{g}_s(x_{ij}) - \bar{g}_s(x_{ik})]^2 / \sum_{i=1}^n \bar{g}_s^2(x_{ij}, x_{jk}).$$

The Friedman’s H-statistic measures the fraction of variance of two-variable partial dependency, $\bar{g}_s(x_{ij}, x_{jk})$ not captured by sum of the respective individual partial dependencies, $\bar{g}_s(x_{ij}) + \bar{g}_s(x_{ik})$. In addition to Friedman’s H-statistic we also use the PDP of two variables to visualize the interaction effects. Similarly, the Friedman’s H-statistic for testing whether a specific feature interacts with any other feature can be estimated by

$$H_j^2 = \sum_{i=1}^n [g(\mathbf{x}_i) - \bar{g}_s(x_{ij}) - \bar{g}_s(x_{i/j})]^2 / \sum_{i=1}^n g^2(\mathbf{x}_i).$$

Note that the, PD plots, ICE curves and PD-, ICE-associated measures and Friedman’s H-statistic are computationally intensive to compute, especially when there are large number of observations in the training set. Hence, in our experiments ICE and PDP-based variable importance measures are computed based on the subset of randomly selected training examples.

5.5 Local Interpretable Model-agnostic Explanations (LIME)

Global interpretations help us to understand the entire modelled relationship. Local interpretations help us to understand the predictions of the model for a single instance or a group of similar instances. In other words, this allows users to zoom into a particular instance or a subset and explore how different features affect the resulting prediction. We use Local Interpretable Model-agnostic Explanations (LIME) approach introduced by Ribeiro, Singh & Guestrin (2016) for explaining individual predictions which relies on the assumption that “every complex model is linear on a local scale”. This is accomplished by locally approximating the complex black-box model with a simple interpretable model. Ribeiro, Singh & Guestrin (2016) highlighted features that are globally important may not be important in the local context and vice versa. The algorithm steps can be summarized as follow:

1. Select an observation of interest which we need to have explanations for its black-box prediction.
2. Create a permuted data set based on the selected observation. Permuted data set is created by making slight modifications to the features of selected observations.
3. Obtain similarity scores by calculating distance between permuted data and selected observation.
4. Obtain predicted outcomes for all permuted data using the black-box model.
5. Select m number of features best describing the black-box model outcome. This can be accomplished by applying feature selection algorithms such as ridge regression, lasso, etc.
6. Fit a simple linear model to the permuted data based on m selected features, similarity scores in step 3 as weights and complex model prediction outcomes in step 4 as response variable.
7. Use the estimated coefficients of simple linear model to explain the local behaviour corresponds to the selected observation in step 1.

An alternative for explaining local behaviour of complex models is proposed by Lundberg & Lee (2017) based on game theory named “Shapley values”.

6 Evaluation based on the M4 competition data

Table 3 shows the performance of FFORMS approach on the M4 competition data. The point forecasts and prediction intervals are evaluated based on the test period of each series. The point forecasts are evaluated based on MASE computed for each forecast horizon, and then by averaging the MASE errors across all series corresponds to each frequency category. Similarly, MSIS is used to evaluate the prediction intervals. The results are compared against several bench marks and top three (3) winners of the M4 competition. The top ranking methods of M4 competition are based on some kind of combination approach. The 1st ranking method is based on hybrid approach which produce forecasts based on both ETS and Machine-Learning approach. The 2nd and 3rd places were based on combination of seven-eight different forecast-models. The 2nd and 3rd approaches are time consuming because forecasts from all candidate models must be computed. The results of FFORMS approach is based on individual forecasts. According to the Table 3, we can see FFORMS approach achieved comparable performance in much more cost-effective and time-effective way. Based on FFORMS approach we provided an entry to the M4 competition. After submission we found a bug in our implementation for hourly data, hence only for hourly data results are different with the published results of the M4 competition. Results of yearly, quarterly, monthly, weekly and daily data are same as the published results. This completes the evaluation of FFORMS framework. The main question we have now is “Can we trust the FFORMS framework if we do not know how it works?” In the next section we present results of what is happening inside the FFORMS framework?

7 What is happening under the hood of FFORMS?

7.1 Yearly series

This vote-matrix information for the yearly data is presented in Figure 2. Figure 2 indicates three primary patterns: i) the distributions of correctly classified classes dominate, indicating a good classification of the meta-learner, the associated outliers of the boxplots indicate some series are correctly classified with high probability ii) irrespective of the class labels the random walk with drift has a high chance of getting selected for all series, (the results if Section 6 also show

Table 3: The performance of FFORMS on the M4 competition data based on point forecasts (based on MASE) and prediction intervals (based on MSIS)

Point Forecasts (Mean Absolute Scaled Error (MASE))						
	Yearly	Quarterly	Monthly	Weekly	Daily	Hourly
FFORMS	3.17	1.20	0.98	2.31	3.57	0.84
auto.arima	3.40	1.17	0.93	2.55	-	-
ets	3.44	1.16	0.95	-	-	-
theta	3.37	1.24	0.97	2.64	3.33	1.59
rwd	3.07	1.33	1.18	2.68	3.25	11.45
rw	3.97	1.48	1.21	2.78	3.27	11.60
nn	4.06	1.55	1.14	4.04	3.90	1.09
stlar	-	2.02	1.33	3.15	4.49	1.49
snaive	-	1.66	1.26	2.78	24.46	2.86
tbats	-	1.19	1.05	2.49	3.27	1.30
wn	13.42	6.50	4.11	49.91	38.07	11.68
mstlarima	-	-	-	-	3.84	1.12
mstlets	-	-	-	-	3.73	1.23
combination (mean)	4.09	1.58	1.16	6.96	7.94	3.93
M4-1st	2.98	1.12	0.88	2.36	3.45	0.89
M4-2nd	3.06	1.11	0.89	2.11	3.34	0.81
M4-3rd	3.13	1.23	0.95	2.16	2.64	0.87
Prediction Intervals (Mean Scaled Interval Score (MSIS))						
FFORMS	39.79	11.24	9.82	20.84	36.36	8.07
M4-1st	23.89	8.55	7.20	22.03	26.28	7.92
M4-2nd	27.47	9.38	8.65	21.53	34.37	18.50
M4-3rd			not submitted			
naive	56.55	14.07	12.30	26.35	32.55	71.24

random walk with drift give very high prediction accuracy with yearly series), iii) FFORMS framework successfully learned the similarities and dissimilarities between the classes itself. For example, within ETS models with a trend component (ETS_T, 3rd panel), disparities between forecast-models with trend components and forecast-models without trend component (wn, ARMA, ETS_NTNS) are immediately visible.

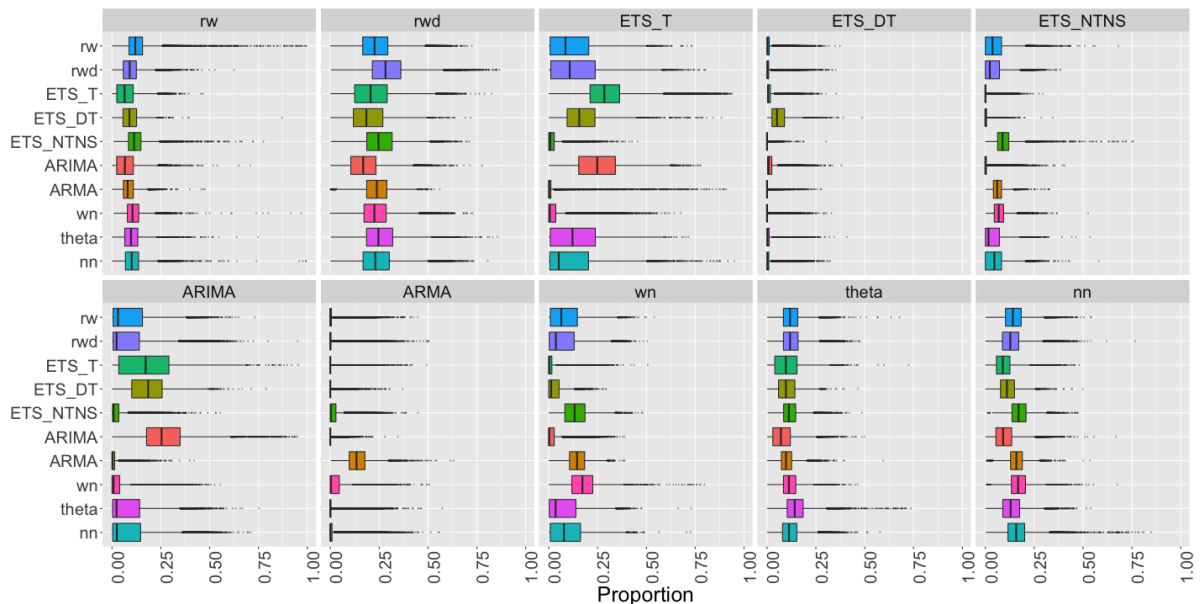


Figure 2: Visualization of the vote matrix based on OOB sample for yearly random forest. Each panel shows the predicted class from the random forest. The X-axis denotes the proportion of times each time series was classified to each class. The colours of boxplots correspond to class label of the “best” forecast-model identified based on MASE and sMAPE. On each row, distribution of correctly classified class dominates, indicating a good classification of the meta-learner.

Figure 3 shows how important each one of the features we considered in the overall classification mechanism of the FFORMS as well as within each class. The first panel gives the ranks of overall variable importance for all features and the other panels shows the class-specific feature rankings. The main point here is strength of trend and test statistic based on Phillips-Perron unit root test, first ACF value of the differenced series, and linearity are the most important features across all categories. In addition to that within first ACF value of residual series of linear trend model appear to be important within all classes. In addition to linearity, the other features related to different types of trend (damped trend: measured by beta, exponential trend: measured by curvature) are assigned a very high importance with in ETS_T, ETS_DT and ETS_NTNS which handles different versions of trend within ETS family. Spikiness appear to be an important feature in the overall classification, even though it does not appear to be among top five feature, spikiness has been assigned a relatively high importance within all categories. The reason for the high importance of spikiness within overall classification is due to the high Mean Decrease in Gini Coefficient. This suggests contribution of spikiness to the homogeneity of the nodes and leaves in the resulting random forest make it important rather than its effect of causality. The length of time series (N) is assigned a very low importance within all categories.

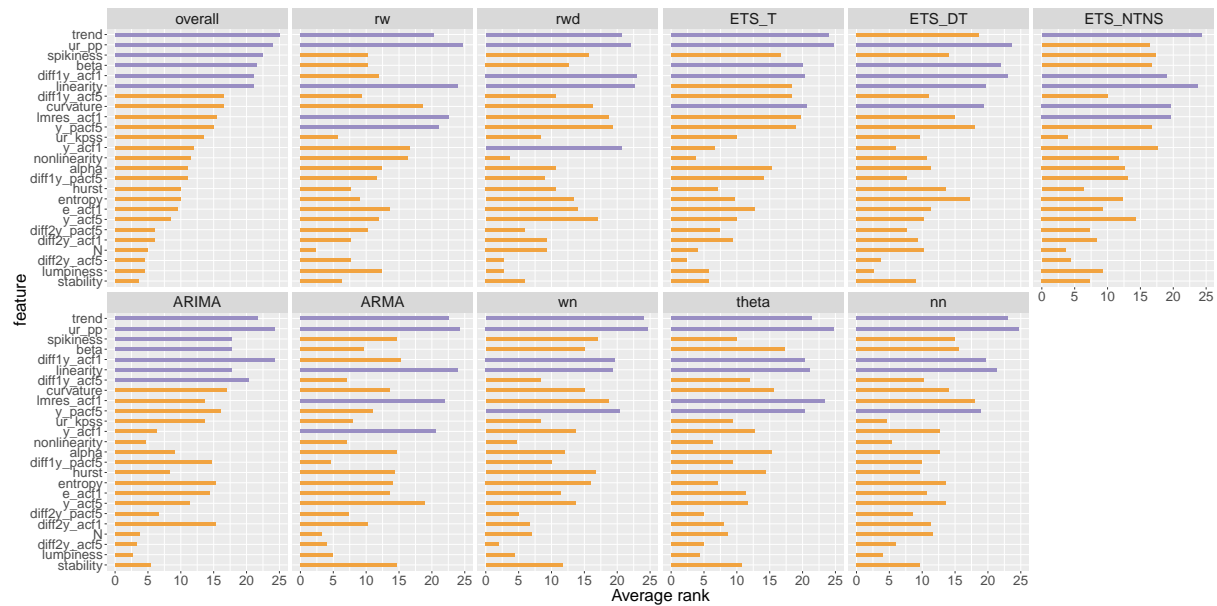


Figure 3: Feature importance plot for yearly series. Overall feature importance (top left plot) is evaluated based on two measures: i) Permutation-based variable importance measure and ii) mean decrease in Gini coefficients. Class-specific feature importance is evaluated based on three measures: i) permutation-based variable importance, PD-based variable importance measure, and ICE-based variable importance measure. Longer bars indicate more important features. Top 5 overall features are highlighted in purple. Strength of trend appears to be the most important feature.

Figure 4 to Figure 6 show the partial dependency curves, and associated confidence intervals of the top-three features that get selected most and top within each class. In all three cases, within

each class the displayed relationship consistent with the theoretical expectations. For example, a high negative value of Phillips-Perron test statistic indicate a stationarity of the series while a positive value of the test statistic indicate nonstationarity of the series. According to [Figure 4](#) we can see probability of selecting ETS model with trend component and ARIMA models increases steadily as ur_pp increases while the opposite relationship could be observed for ARMA and WN. According to [Figure 5](#) we can see the information about whether the strength of trend in the series is extremely high or not matters a lot in selecting a forecast-model. The probability of selecting ETS model without trend components (ETS_NTNS), WN and ARMA decreases when the strength of trend is extremely high while the opposite relationship could be observed for other classes. According to the Friedman's H-statistic $diff1y_acf1$ and $lmres_acf1$ show high level of two-way interactivity. The accosiated two-way partial dependency plots are shown in [Figure 7](#). The classes rw , rwd , ETS_NTNS, ARIMA and nn shows interactivity. The plots suggests the probability of selecting the corresponding model changes according to the changes in both $diff1y_acf1$ and $lmres_acf1$. Within neural network class we can see when $lmres_acf1$ get closer to 1, $lmres_acf1$ does not interact with $diff1y_acf1$. ETS_T class show low interactivity, irrespective of the value of $diff1y_acf1$ series with low value of $lmres_acf1$ have high chance of selecting ETS_T model. Within ETS_T the colour of the bands are constant throughout the domain. However, for rw , rwd , ARIMA, theta and neural network class strength of the colour band varies according to the levels of both $diff1y_acf1$ and $lmres_acf1$. The interactivity pattern within wn class is not visible 0-0.21 scale but it is visible within 0-1 scale. This suggests two-way interactivity easily separete wn from the rest.

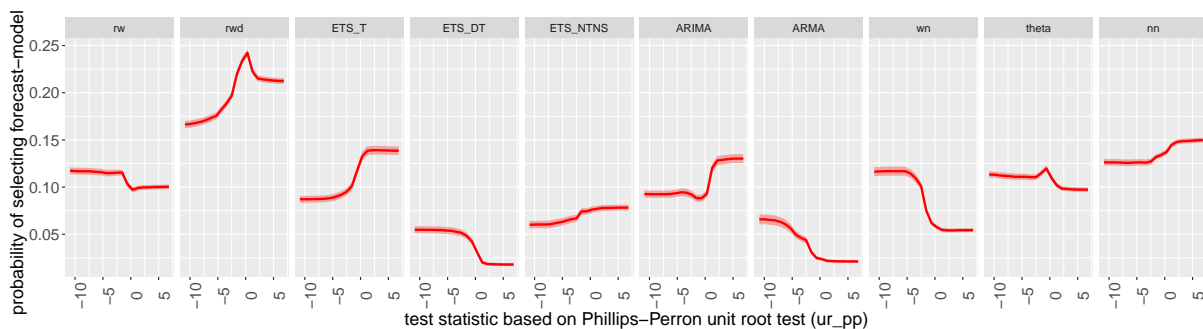


Figure 4: Partial dependence plots for ur_pp . The shading shows the 95% confidence intervals. Y-axis denotes the probability of belong to corresponding class. All classes show a turning point in the relationship around zero.



Figure 5: Partial dependence plots for trend. The shading shows the 95% confidence intervals. Y-axis denotes the probability of belong to corresponding class. Probability of selecting ETS models without a trend component and stationary models (WN and ARMA) decrease for an extremely high value of trend.

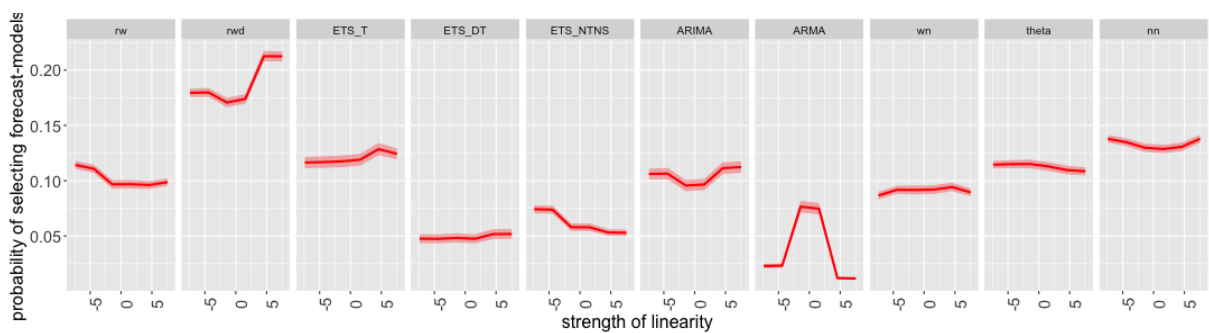


Figure 6: Partial dependence plots for linearity. The shading shows the 95% confidence intervals. Y-axis denotes the probability of belong to corresponding class. Random walk with drift and ARMA classes are highly sensitive to the value of linearity around 0.

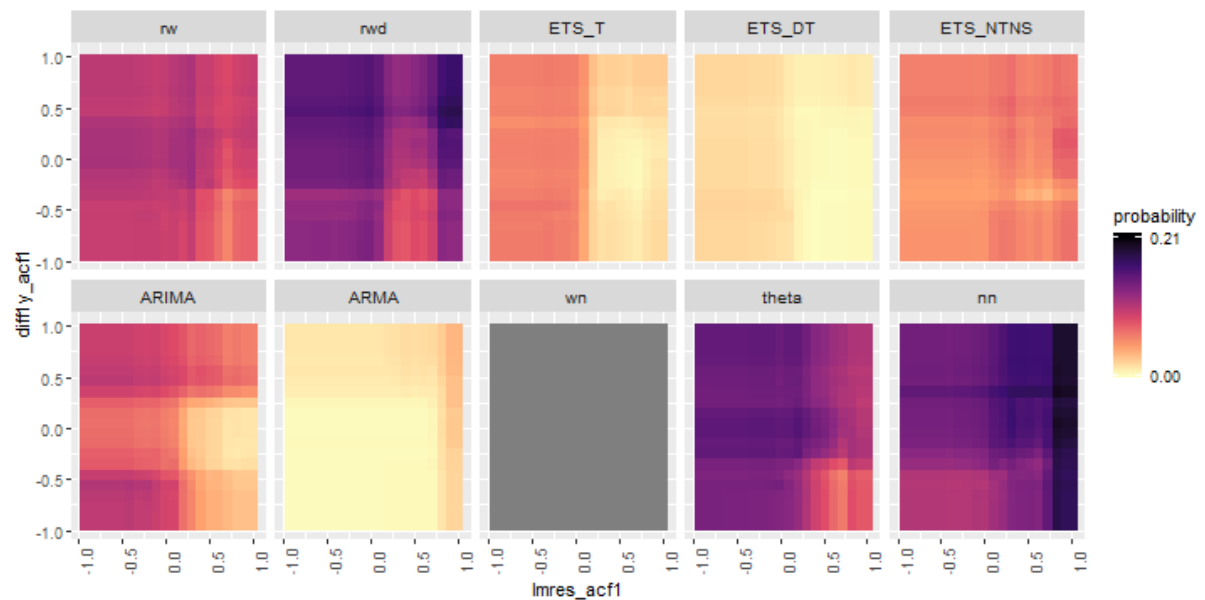


Figure 7: Partial dependence plot of model selection probability and the interaction of linearity and diff1y_acf1 for yearly data. Dark regions show the high probability of belonging to the corresponding class shown in the plot title. Random walk and random walk with drift class show opposite pattern of interactivity between sediff_seacf1 and linearity.

7.2 Quarterly series

Figure 8 shows the vote-matrix of the random forests for quarterly based on OOB observations. The disparities between dominating class and non-dominating classes are immediately apparent within ETS_DT, ETS_T, ETS_DTS, ETS_TS, ETS_S and ARMA classes. Within each panel, the outliers associated with dominating distribution indicate some series are correctly classified with very high probability. Furthermore, except ETS_DTS, ETS_TS, ETS_S and SARIMA all other series has a high chance of classified into random walk with drift class. Within the theta, and nn all distributions level at similar proportionalities as they both represent a general class of forecast-models. Overall random forest successfully learnt the similarities and dissimilarities between different forecast-models. For example, within each panel, series with a label which involves a seasonal parameter have a high chance of getting selected a seasonal forecast models such as SARIMA, ETS models with seasonal components, stlar, and tbats.



Figure 8: Visualization of the vote matrix based on OOB sample for quarterly random forest. Each panel shows the predicted class from the random forest. The X-axis denotes the proportion of times each time series was classified to each class. The colours of boxplots correspond to class label of the “best” forecast-model identified based on MASE and sMAPE. The disparities between dominating class and nondominating classes are immediately apparent within ETS model classes.

Figure 9 shows the feature importance plots for the quarterly. The features strength of seasonality, trend, diff1y_pacf5, linearity and spikiness are the most important features across all categories. Even though the lumpiness does not appear as a top five feature within classes it is appeared to be an important feature in the overall classification process and a relatively high ranks are assigned within many classes. Except, ETS_DT, ETS_DTS and ETS_TS the features, nonlinearity, hwalpha and hwbeta are assigned a very low importance within all other classes.

Figure 10 to Figure 12 show the partial dependency curves, and associated confidence intervals

of the top-three features that get selected most in each class. According to [Figure 10](#), it is immediately apparent, that the probability of selecting a seasonal forecast-model increases as the strength of seasonality increases. According to [Figure 10](#) and [Figure 11](#) probability of selecting SARIMA models drastically changes according to the change of seasonality, and `diff1y_pacf5`. The partial dependence plot of trend is very similar to the observed patterns in yearly data.

According to the Friedman's H-statistic (>0.5) curvature and `e_acf1` is the most commonly appeared two-way interactivity. The associated partial dependency plots are shown in [Figure 18](#). The interactivity pattern of `wn` class is visible only according to the 0-1 scale, which suggest the two-way interactivity easily separate the `wn` class from the rest. on the otherhand the interactivity pattern with `ETS_NTNS`, `ETS_DT`, `ETS_T`, `ARMA`, `ARIMA` is not visible within 0-0.125 scale. The reason could be these model does not have a parameter to handle seasonal component. Hence, the probability selecting these series are relatively low compared to other models. Hence, the pattern of interactivity is not visible. According to the figure [Figure 18](#) we can see that the pattern of interactivity does not completely change the individual relationship of the feature with the predicted outcome, but interactivity influence the probability of selecting the corresponding forecast-model. For example, ingeneral `stlar` models has a high change of getting selected with series with curvature value. However, this probability changes according to the values of `e_acf1`. Within `ETS_S` when curvature > 0 , the two features show very low interactivity.

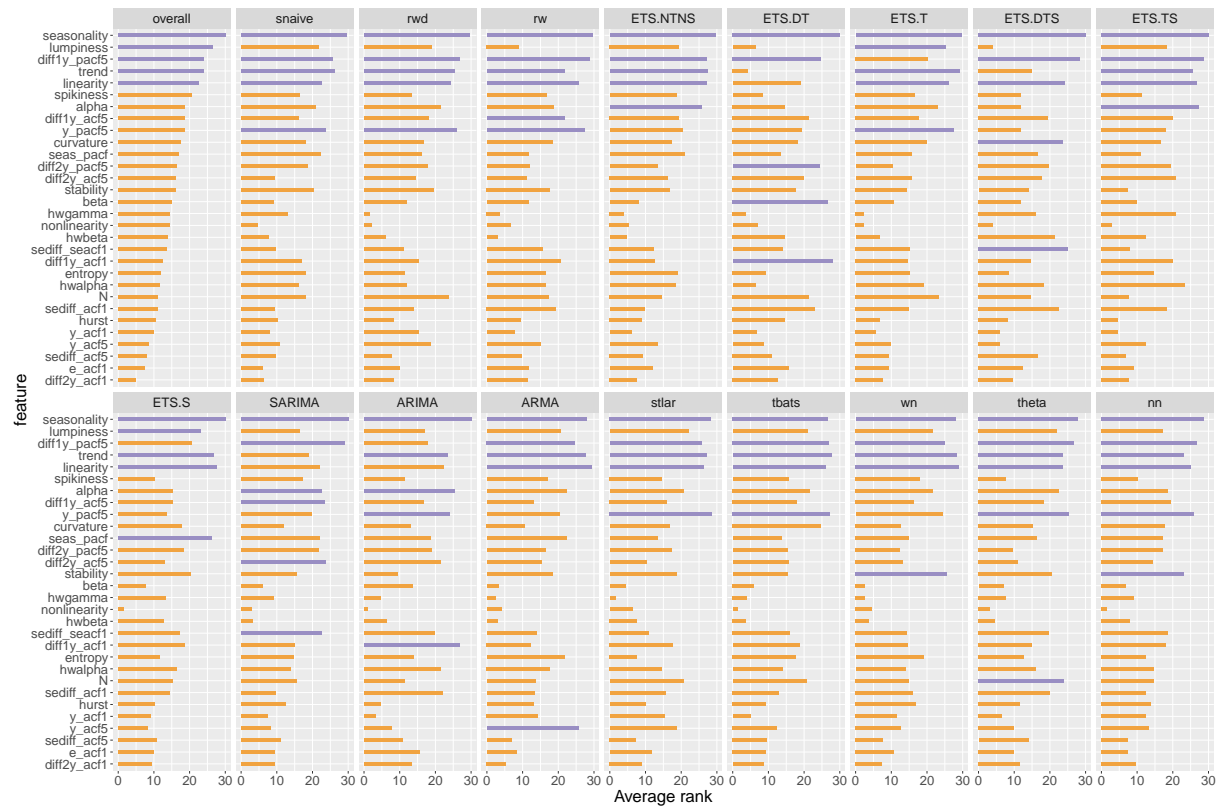


Figure 9: Feature importance plot for quarterly series. Overall feature importance (top left plot) is evaluated based on two measures: i) Permutation-based variable importance measure and ii) mean decrease in Gini coefficients are used to evaluate shown in the top left plot. Class-specific feature importance is evaluated based on three measures: i) permutation-based variable importance, PD-based variable importance measure, and ICE-based variable importance measure. Longer bars indicate more important features. Top 5 overall features are highlighted in purple. Strength of seasonality appears to be the most important feature.

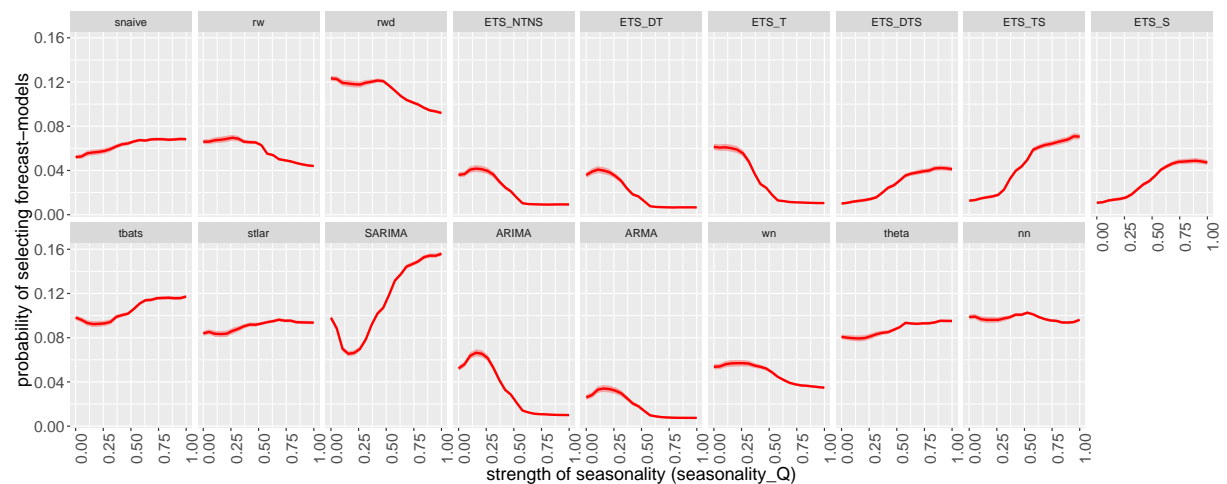


Figure 10: Partial dependence plots for strength of seasonality (seasonality_Q). The shading shows the 95% confidence intervals. Y-axis denotes the probability of belong to corresponding class. Probability of selecting forecast-models with seasonal components increases as seasonality increases.

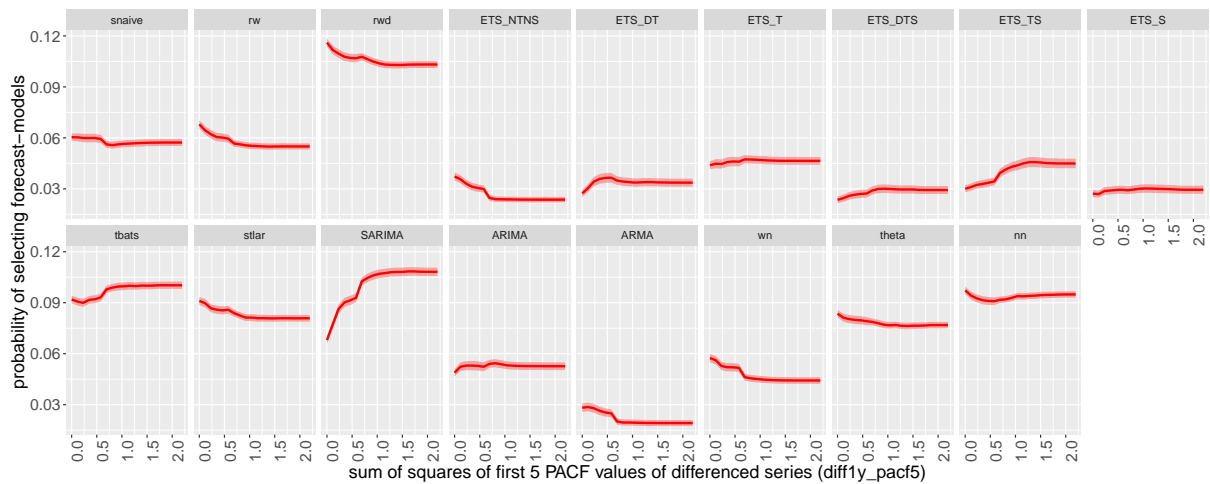


Figure 11: Partial dependence plots for sum of squares of first 5 PACF values of differenced series (diff1y_pacf5). The shading shows the 95% confidence intervals. Y-axis denotes the probability of belong to corresponding class. Probability of selecting *SARIMA* models drastically increases as diff1y_pacf5 increases

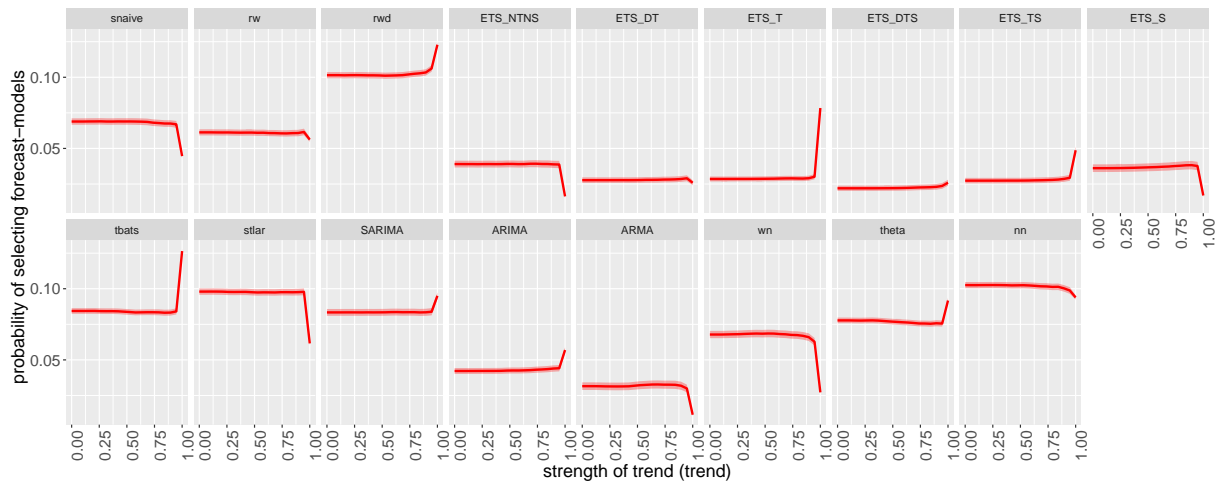


Figure 12: Partial dependence plots for strength of trend (trend). The shading shows the 95% confidence intervals. Y-axis denotes the probability of belong to corresponding class. The information about whether a series has an extremely high trend value or not drastically influence the decision of selecting a forecast-model.

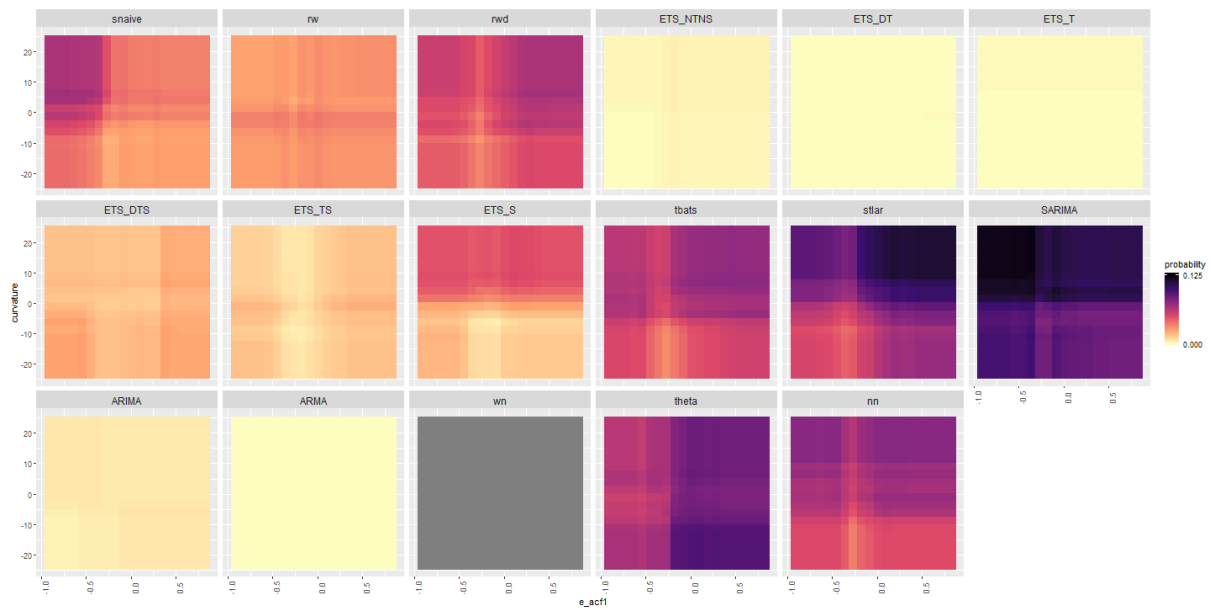


Figure 13: Partial dependence plot of model selection probability and the interaction of curvature and e_acf1 for yearly data. Dark regions show the high probability of belonging to the corresponding class shown in the plot title.

7.3 Monthly series

Figure 14 shows the vote-matrix information for monthly series. Vote matrix information for monthly data depicted similar pattern to quarterly data. For quarterly and monthly data, the same set of features and the class-labels are used to train the model. Hence, this consistency between the results of the quarterly and the monthly series would provide evidence in support of the validity and trustability of the model.

Figure 15 shows the feature importance plots for monthly data. For both quarterly and monthly data strength of seasonality, trend, linearity and spikiness are the most important features across all categories. In the case of yearly series low variable importance is assigned to both stability and length of the series. However, within quarterly and monthly data a high variable importance is assigned to length of the series and stability. In addition to the strength of seasonality, the models handling seasonal components (snaive, SARIMA, all ETS models with seasonal component) assigned a high importance to the additional features related to seasonality such as ACF, PACF-based features related to seasonal lag or seasonally differenced series. Furthermore, as expected features calculated based on parameter estimated of ETS(A, A, A) ranked as important for the choice of ETS with damped trend and seasonal component and ETS with trend and seasonal component. One notable difference between the quarterly series and the monthly series is, for monthly data length of the series is ranked among the top five, specially in random walk with drift, random walk, ETS with seasonal and trend component, ETS_seasonal, SARIMA and

ARIMA classes.

The partial dependency curves of seasonality and trend of monthly data are very similar to the corresponding plots of quarterly data. Hence, these partial dependency curves are not repeated here. [Figure 16](#) partial dependency plot of linearity for monthly data. For non-seasonal forecast models the relationships are very similar to the corresponding observed curves in yearly data. Partial dependency curves of [Figure 17](#) reveals highly parameterized models and models involves seasonal components such as (ETS_TS, ETS_DTS, ETS_S, SARIMA, tbats) shows an increasing relationship with N after some point. According to figure [Figure 36](#) we can see number of features show high level(>0.5) of interactivity within classes are relatively low. ?? shows partial dependency plots corresponds to the interaction between hwalpha and sediff_seacf1. The Friedman's H-statistic (0.78) show very high level of interactivity between the features hwalpha and sediff_seacf1 with neural network class. Inadditin to wn class the interaction between the two features are more useful in separating out the rwd and stlar from the rest, as they show high probability of selecting the corresponding models thoroughpout the region. The tbats class show low interactivity between the two features. The main effect of hwalpha dominates within tbats due to the constant colour horizontal bands. The classes, snaive, rw, rwd, ETS_NYNS, stlar, theta and nn show varying degree of colour strength bands indigating the probability of selecting the models changes according to the levels of both hwalpha and sediff_acf1.

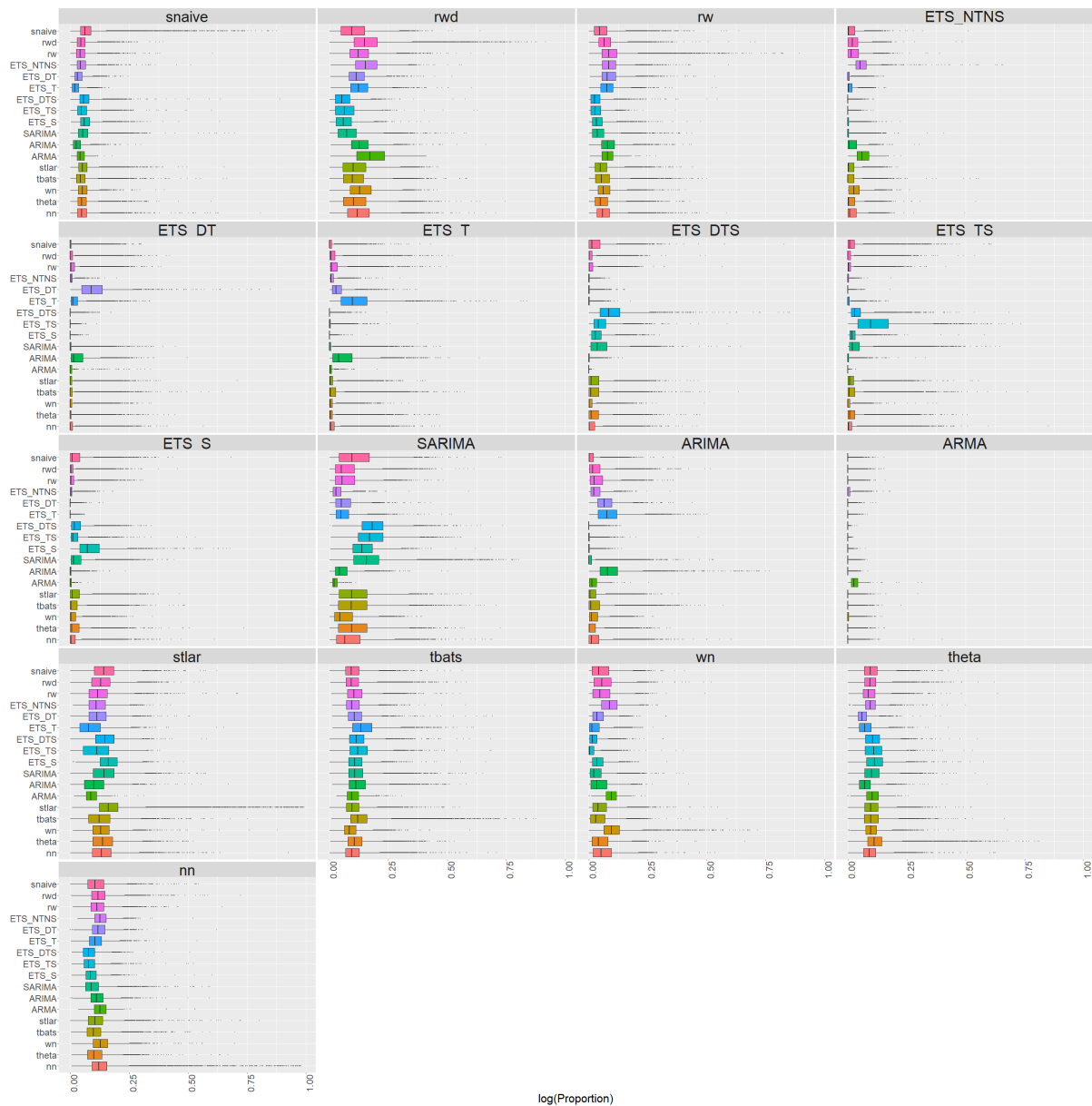


Figure 14: Visualization of the vote matrix based on OOB sample for quarterly random forest. Each panel shows the predicted class from the random forest. The X-axis denotes the proportion of times each time series was classified to each class. The colours of boxplots correspond to class label of the “best” forecast-model identified based on MASE and sMAPE. On each row, distribution of correctly classified class dominates, indicating a good classification of the meta-learner.

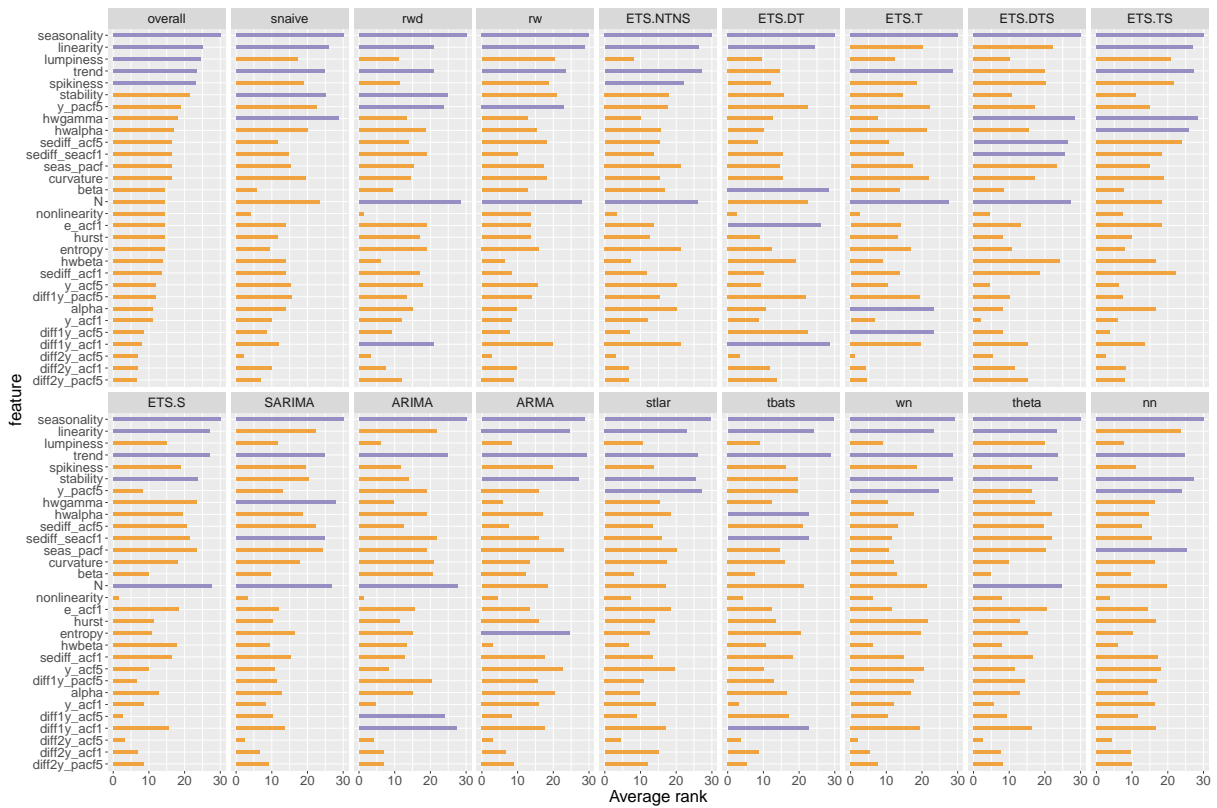


Figure 15: Feature importance plot for monthly series. Overall feature importance (top left plot) is evaluated based on two measures: i) Permutation-based variable importance measure and ii) mean decrease in Gini coefficients are used to evaluate shown in the top left plot. Class-specific feature importance is evaluated based on three measures: i) permutation-based variable importance, PD-based variable importance measure, and ICE-based variable importance measure. Longer bars indicate more important features. Top 5 overall features are highlighted in purple. Strength of seasonality appears to be the most important feature.

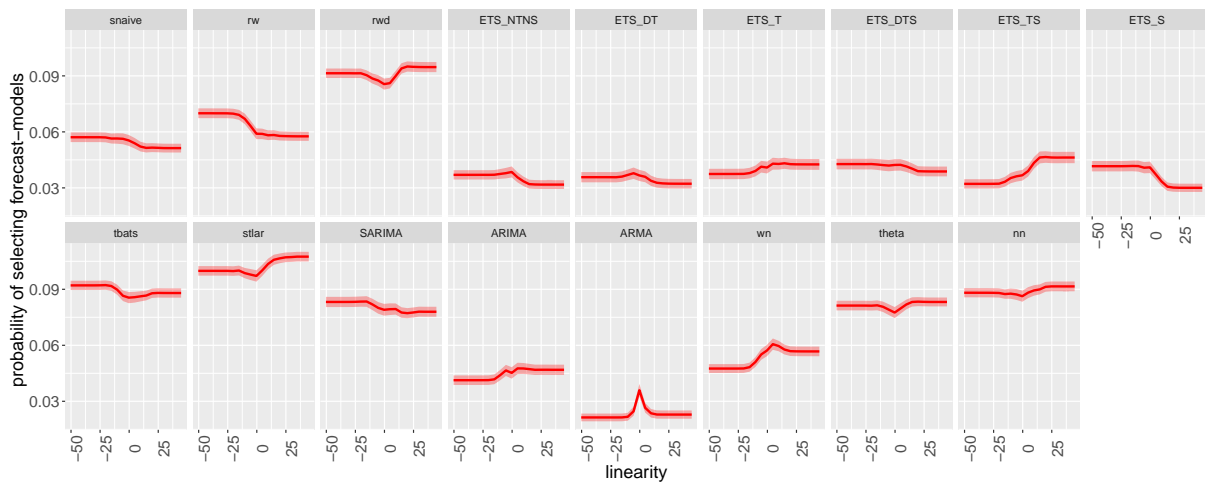


Figure 16: Partial dependence plots for linearity. The shading shows the 95% confidence intervals. Y-axis denotes the probability of belong to corresponding class. All curves show an turning point in the relationship around zero.

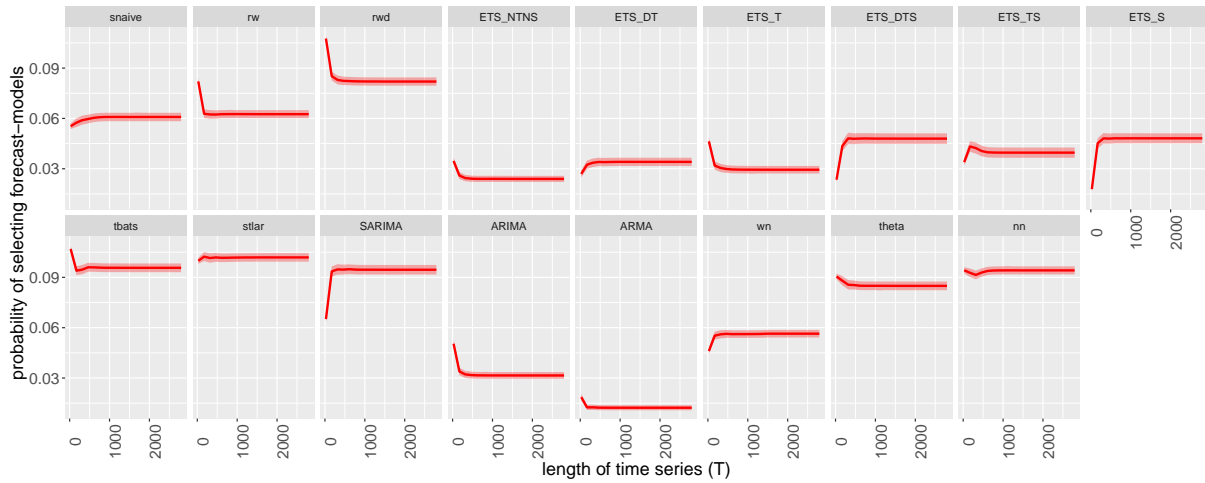


Figure 17: Partial dependence plots for length of time series (T). The shading shows the 95% confidence intervals. Y-axis denotes the probability of belong to corresponding class. Probability of selecting *rw* and *rwd* decreases as the length > 500 .

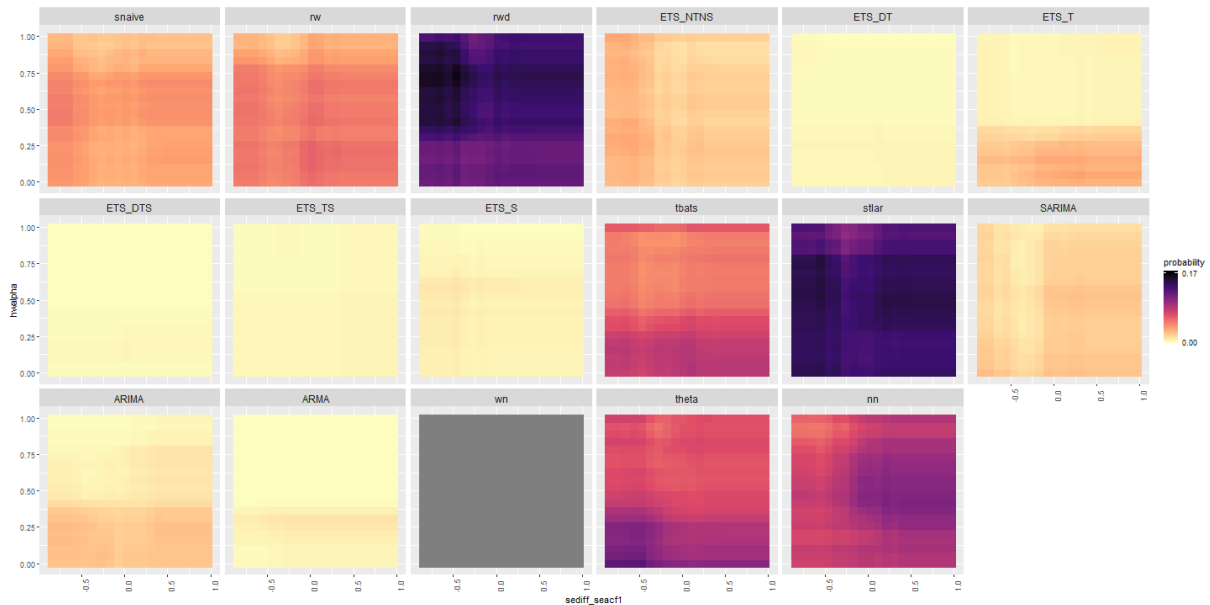


Figure 18: Partial dependence plot of model selection probability and the interaction of linearity and curvature for yearly data. Dark regions show the high probability of belonging to the corresponding class shown in the plot title.

7.4 Weekly series

Figure 19 shows the vote matrix information for weekly data. Unlike, yearly, quarterly and monthly data *theta* method has a low chance of getting selected. The random walk with drift, *tbats* models and *nn* have a high chance of getting selected. *ARMA* class shows some unusual behaviour within some categories due to class imbalance ratio, *ARMA* class contains fewer number of observations in the training set.

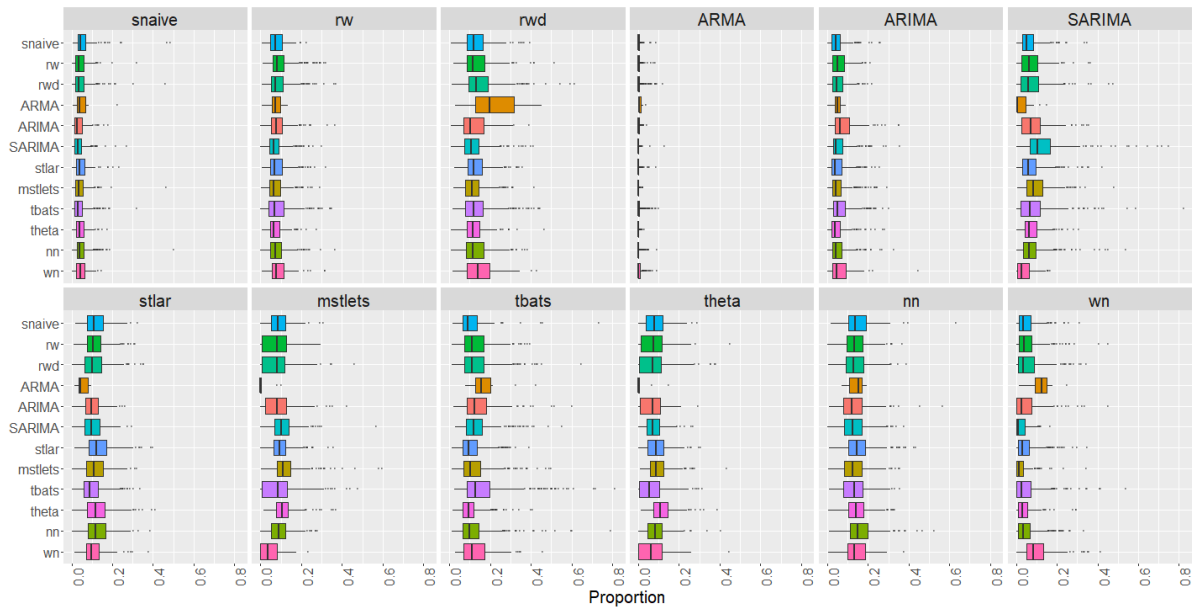


Figure 19: Visualization of the vote matrix based on OOB sample for weekly random forest. The Y-axis denotes what was predicted from the random forest. The X-axis denotes the proportion of times each time series was classified to each class. The colours of boxplots corresponds to class label of the “best” forecast-model identified based on MASE and sMAPE. The models *rwd*, *tbats*, *nn* have a high chance of getting selected.

According to the results of [Figure 20](#) spikiness, linearity, trend, strength of seasonality, stability and lumpiness have been assigned a high importance. This is similar to the results of yearly, quarterly and monthly data. The length of series has been selected among top 5 by *mstlets*, *tbats*, *theta* and neural network models. The partial dependence plots for seasonality, trend and linearity is very similar to the patterns observed in quarterly and monthly data. Hence, we have not repeated the results. Instead, we have plotted the partial dependence plot for spikiness. According to [Figure 21](#) the probability of selecting *snaive*, random walk, neural network and white noise increases as the spikiness increases while other forecast-model classes show the opposite. According to the Friedman’s H-statistic curvature and linearity is the most commonly appeared two-way interactivity. The interactivity pattern of ARMA is not visible between 0-0.15 scale as ARMA models has low chance of getting selected with weekly series. A unique pattern of interactivity could be observed within *snaive*, *rw*, *SARIMA*, *stlar*, *mstlets*, *tbats*, *theta* and *nn*. Within *rw* class we can see the main effect of linearity dominates, however, the strength of colour vary according to the level of curvature.

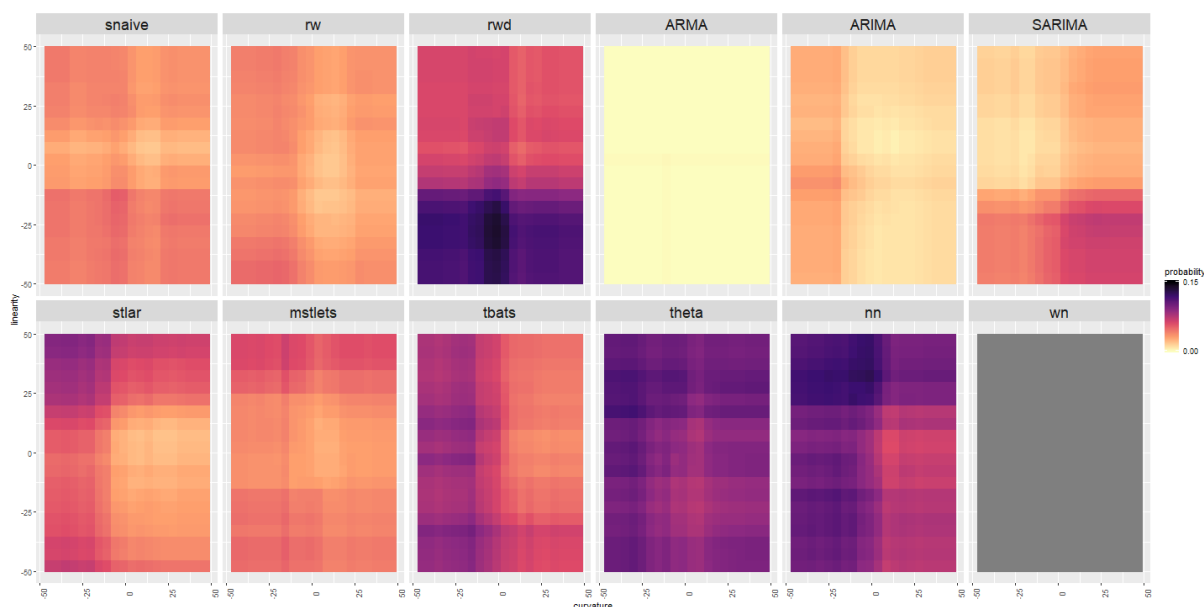


Figure 22: Partial dependence plot of model selection probability and the interaction of curvature and linearity for yearly data. Dark regions show the high probability of belonging to the corresponding class shown in the plot title.

7.5 Daily series

Figure 23 shows the vote matrix information for hourly data. According to Figure 23 the distributions corresponds to observations that have been correctly classified dominate the top for daily data. However, within daily series there are few observations that have been incorrectly classified to `tbats` class with very high probabilities. In general, neural network models have a higher chance of getting selected for daily time series.

Variable importance graph for daily data is shown in Figure 24. The most important features for daily time series are, strength of seasonality corresponds to the weekly seasonality (7, measured by `seasonal_strength1`), stability, trend, lumpiness and linearity. Furthermore, length of the series is important in determining random walk, random walk with drift, `mstlarima`, `mstlets`, `stlar`, `theta` and `nn` classes.

Figure 25 we can see probability of selecting white noise models increases as stability increases. According to the partial dependency curves of `length(N)` Figure 25 shorter series tends to select random walk with drift models while probability of selecting `snaive`, `mstlarima` and `mstlets` models increases as the length of series increases. Neural network models show a non-monotonic relationship with length of the series (`N`). The `theta` models tend to be selected for series with high annual seasonality but very low weekly seasonality. For daily time series, stability and strength of seasonality corresponds to weekly pattern shows high level of two-way interactivity within many classes. The classes `theta` and `nn` shows high level of interactivity

between seasonal_W and stability. Within theta and nn easy separation of quadrants are visible. For example, with low value of seasonal_W and low stability, nn models have high chance of getting selected, while with high value of seasonal_W and stability has low chance of selecting nn. The opposite relation can be observed within rw, mstlarima, tbats and theta classes.

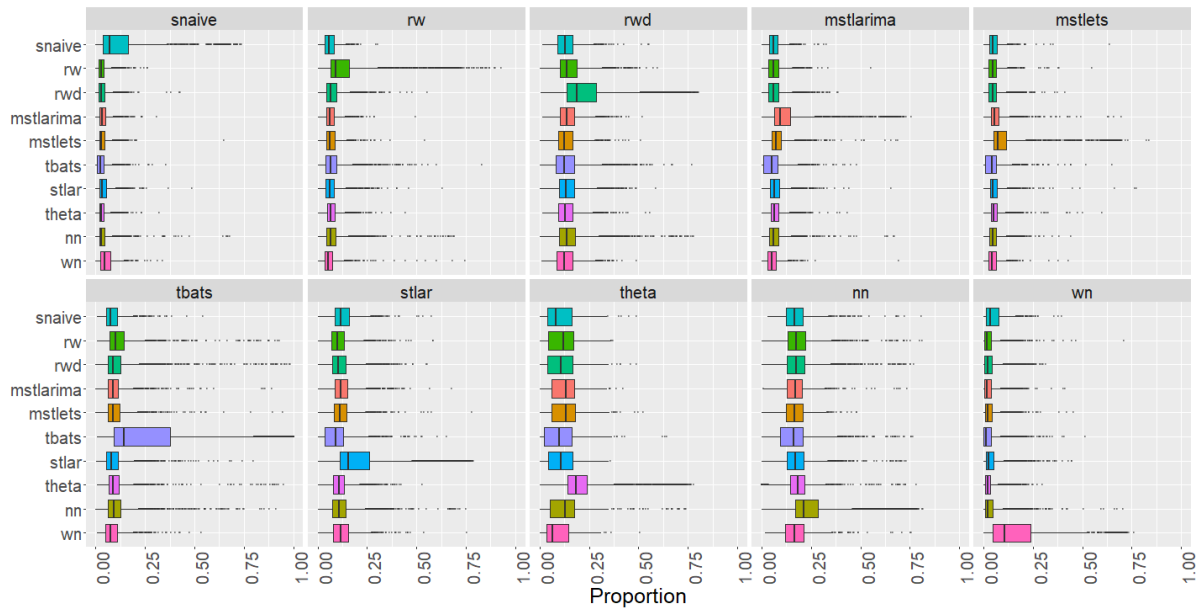


Figure 23: Visualization of the vote matrix based on OOB sample for daily random forest. The Y-axis denotes what was predicted from the random forest. The X-axis denotes the proportion of times each time series was classified to each class. The colours of boxplots corresponds to class label of the “best” forecast-model identified based on MASE and sMAPE. The models rwd, tbats, nn have a high chance of getting selected.

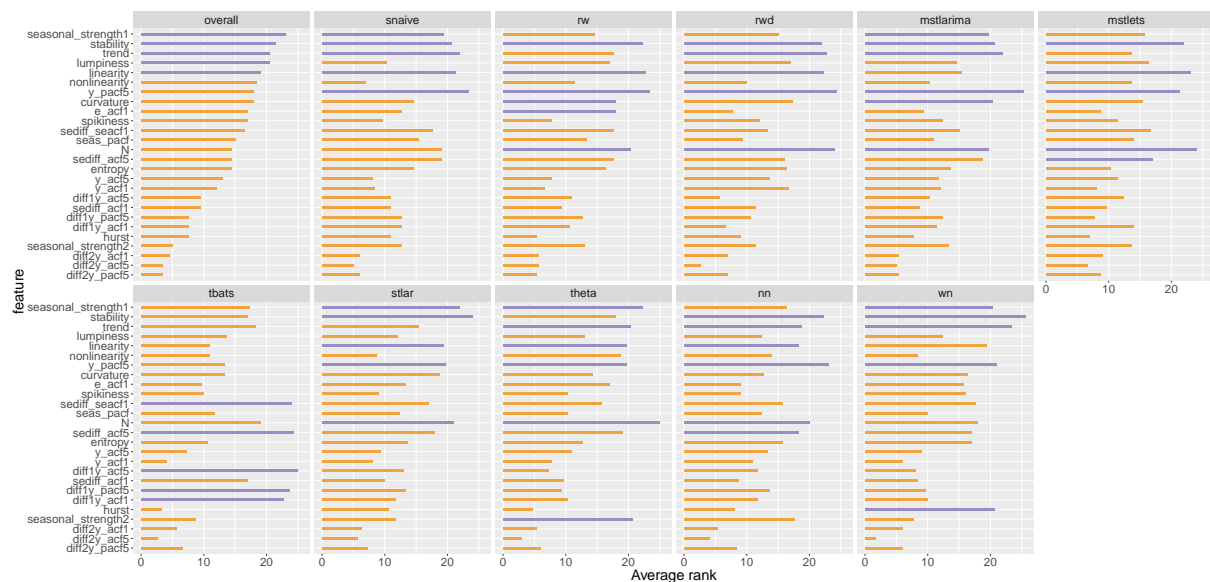


Figure 24: Feature importance plot for daily data. Permutation-based VI measure and mean decrease in Gini coefficients are used to evaluate overall feature importance. Class-specific feature importance is evaluated based on the three measures: permutation-based VI, PD-based VI measure, and ICE-based VI measure. Longer bars indicate more important features. Top 5 features are highlighted in red.

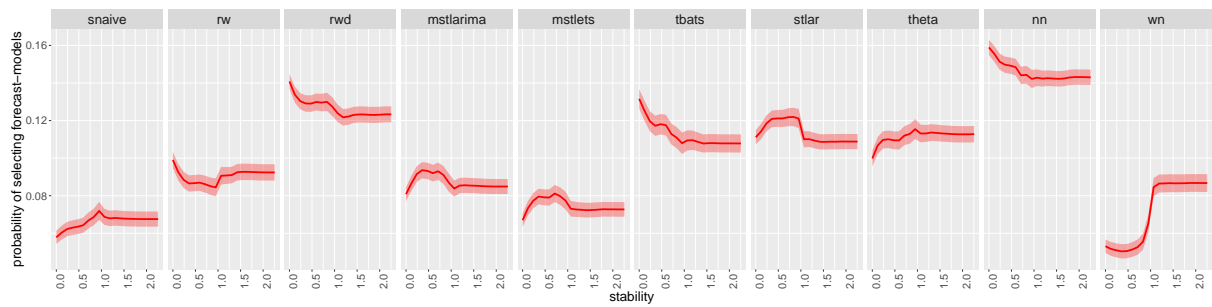


Figure 25: Partial dependence plots for stability. The shading shows the 95% confidence intervals. Y-axis denotes the probability of belonging to corresponding class.



Figure 26: Partial dependence plots for stability. The shading shows the 95% confidence intervals. Y-axis denotes the probability of belonging to corresponding class.

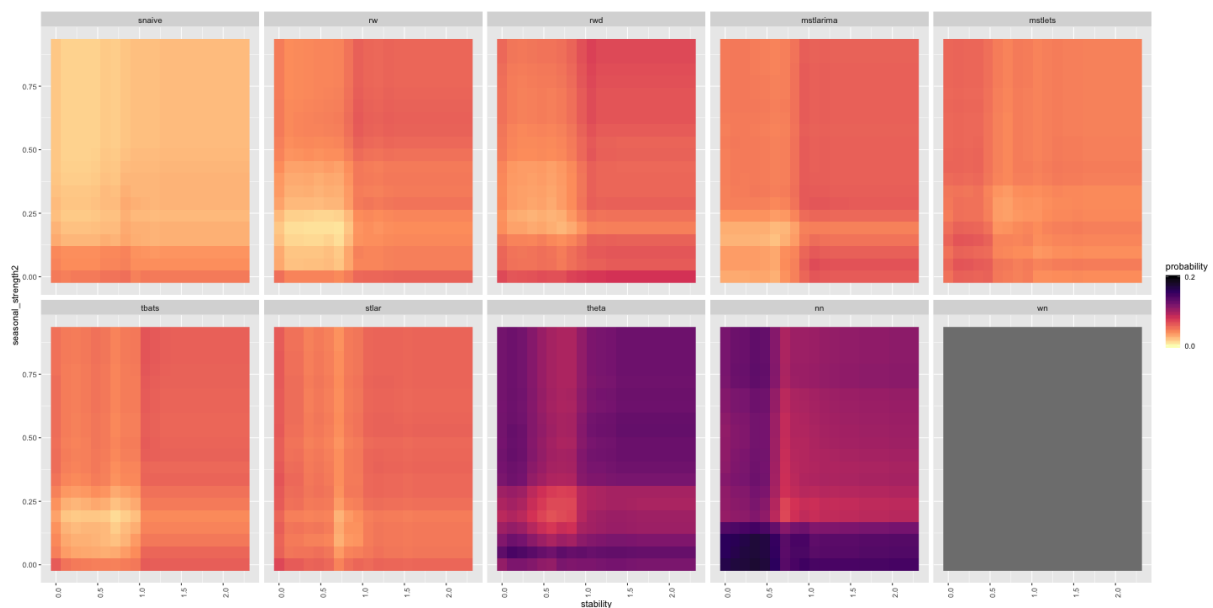


Figure 27: Partial dependence plot of model selection probability and the interaction of seasonal_W and stability for daily data. Dark regions show the high probability of belonging to the corresponding class shown in the plot title. Random walk and random walk with drift class show opposite pattern of interactivity between *sediff_seacf1* and linearity.

7.6 Hourly series

Figure 28 shows the vote matrix information for hourly data. Overall, for hourly series random walk with drift models, tbats and neural network models have a high chance of getting selected. Furthermore, it is important to note that within nn panels all distributions located away from series which indicates all hourly series have been assigned a non-zero probability of getting selected to neural network class.

Variable importance graph for hourly data is shown in Figure 29. According to Figure 29, the strength of daily seasonality (period=24, measured by seasonal_strength1) appear to be more important than the strength of weekly seasonality (period=168, measured by seasonal_strength2). However, the strength of weekly seasonality also seems to be one of the most important features for classes snaive, random walk, mstlarima, and tbats. Furthermore, entropy, linearity, sum of squares of first 5 coefficients of PACF, curvature, trend, spikiness and stability were found to be the most important features in determining best forecasting method for hourly time series. Only snaive category ranked N among top 5 for hourly time series.

The partial dependency plots of the strength of seasonalities are shown in Figure 30. According to Figure 30 the probability of selecting random walk, random walk with drift, theta model and white noise process decrease with higher value of strength of daily seasonality. On the other hand, probability of selecting random walk model increase as the strength of weekly seasonality increases. The partial dependency plots of entropy are shown in Figure 31. The entropy measures forecastability of time series. We can see the probability of selecting random walk with drift and tbats decreases as the entropy increases. This is consistent with our theoretical expectations. In other words, if the series has a clear trend or a clear trigonometric pattern then the forecastability of the time series is high and the entropy is low. For a clear trend, random walk with drift model is suitable while for a series with trigonometric pattern tbats model is suitable.

Friedman's H-Statistic shows high level of interactivity between sediff_seacf1 and linearity across all classes. The associated two-way partial dependency plot is shown in Figure 32. According to the Figure 32 we can see there is a unique pattern of interactivity exist within each class. Within rw, rwd and mstlarima we can see a separation between lower half and upper half due to the effect of sediff_seacf1. However, the colours of the bands is not uniform across the bands, which indicates the probability of selecting the models changes according to the levels of linearity. The two-way partial dependency plots show rwd has a high chance of getting selected with series with low sediff_seacf1, this is due to the overdifferencing of the series.

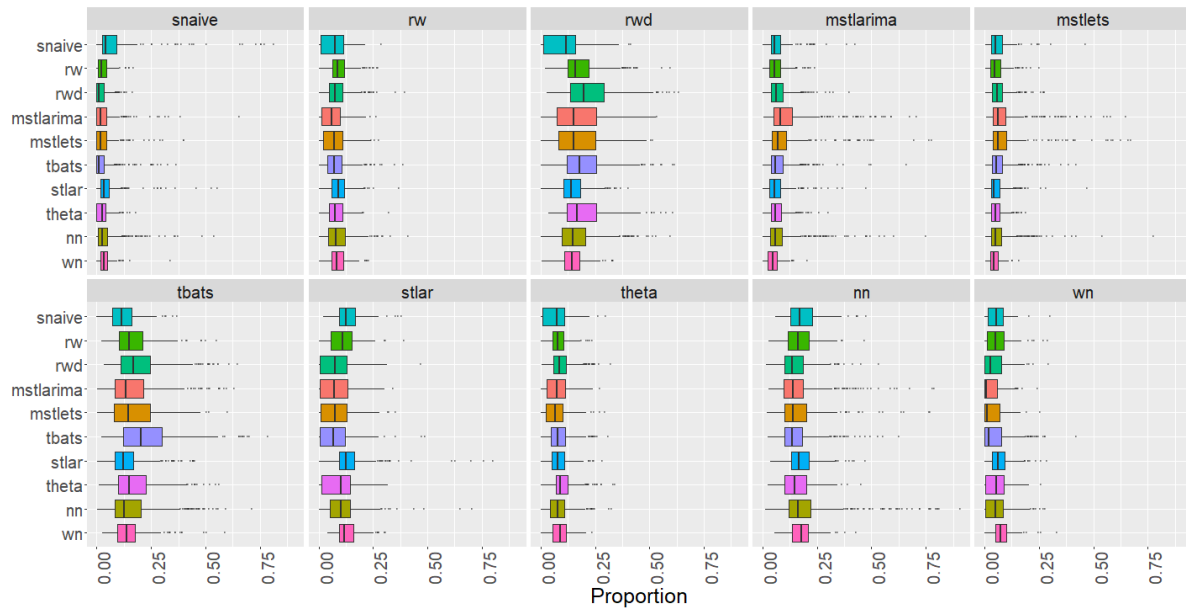


Figure 28: Visualization of the vote matrix based on OOB sample for hourly random forest. The Y-axis denotes what was predicted from the random forest. The X-axis denotes the proportion of times each time series was classified to each class. The colours of boxplots corresponds to class label of the “best” forecast-model identified based on MASE and sMAPE. The models rwd, tbats, nn have a high chance of getting selected.

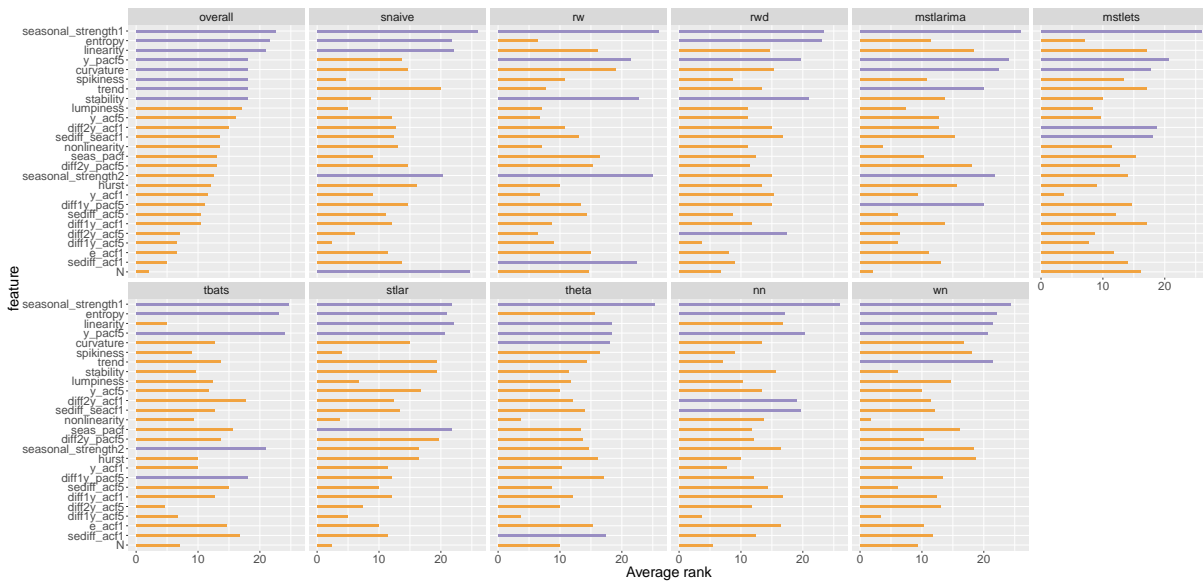


Figure 29: Feature importance plot hourly series. Permutation-based VI measure and mean decrease in Gini coefficients are used to evaluate overall feature importance. Class-specific feature importance is evaluated based on the three measures: permutation-based VI, PD-based VI measure, and ICE-based VI measure. Longer bars indicate more important features. Top 5 features are highlighted in red.

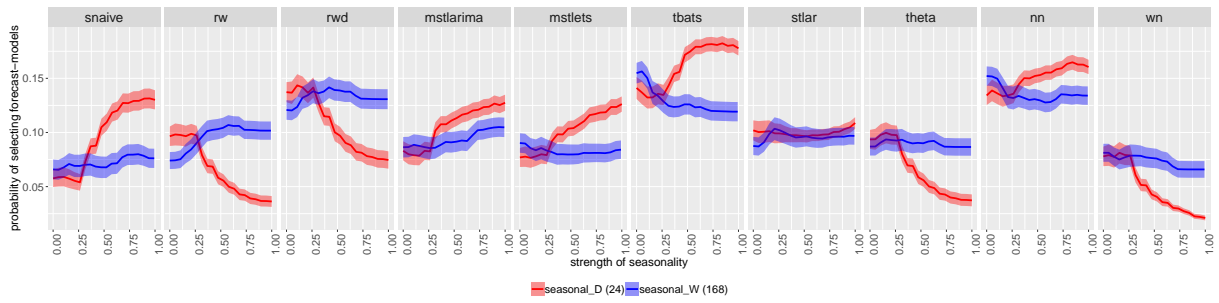


Figure 30: Partial dependence plots for strength of seasonality. The shading shows the 95% confidence intervals. Y-axis denotes the probability of belonging to corresponding class.

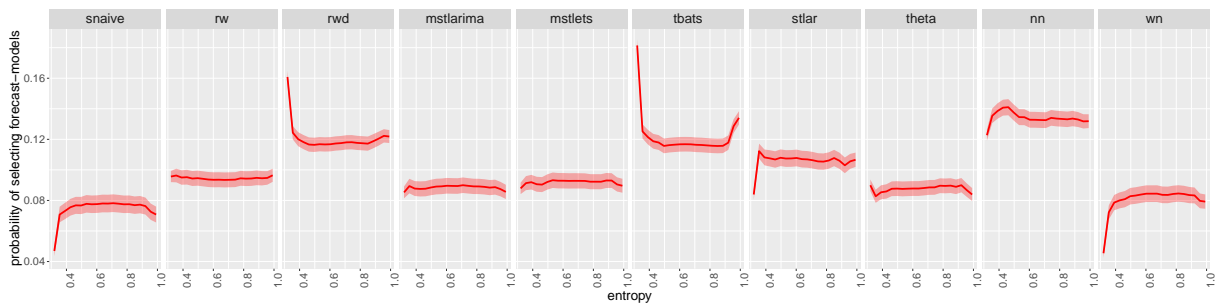


Figure 31: Partial dependence plots for entropy. The shading shows the 95% confidence intervals. Y-axis denotes the probability of belonging to corresponding class.

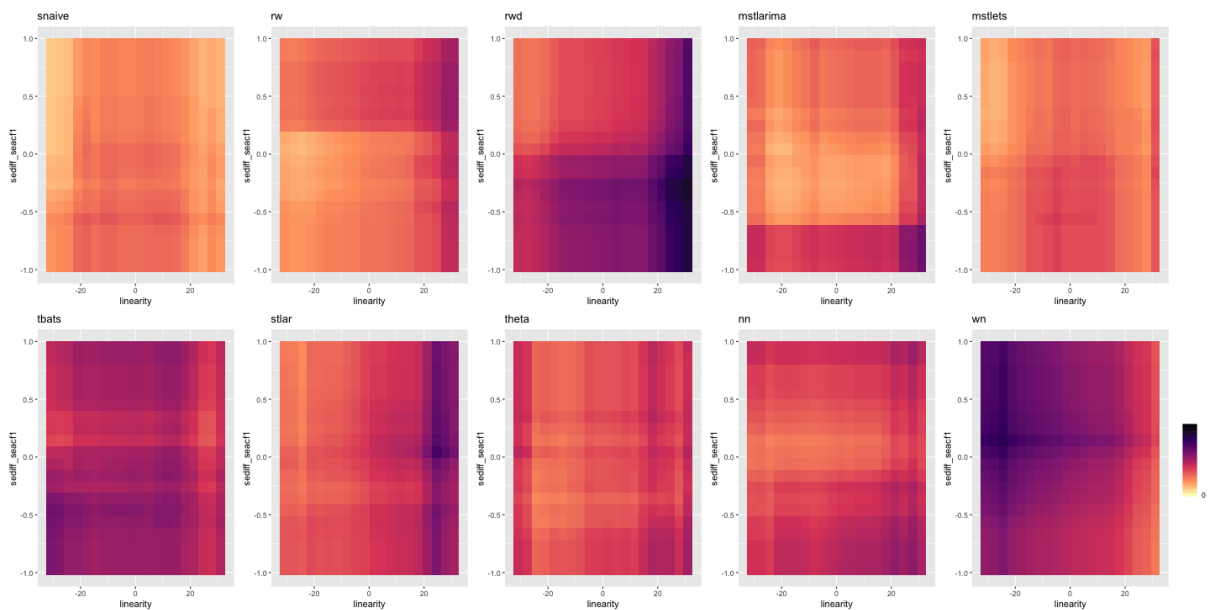


Figure 32: Partial dependence plot of model selection probability and the interaction of `sediffseacf1` and `linearity` for hourly data. Dark regions show the high probability of belonging to the corresponding class shown in the plot title. Random walk and random walk with drift class show opposite pattern of interactivity between `sediffseacf1` and `linearity`.

7.7 Local Interpretable Model-agnostic Explanations

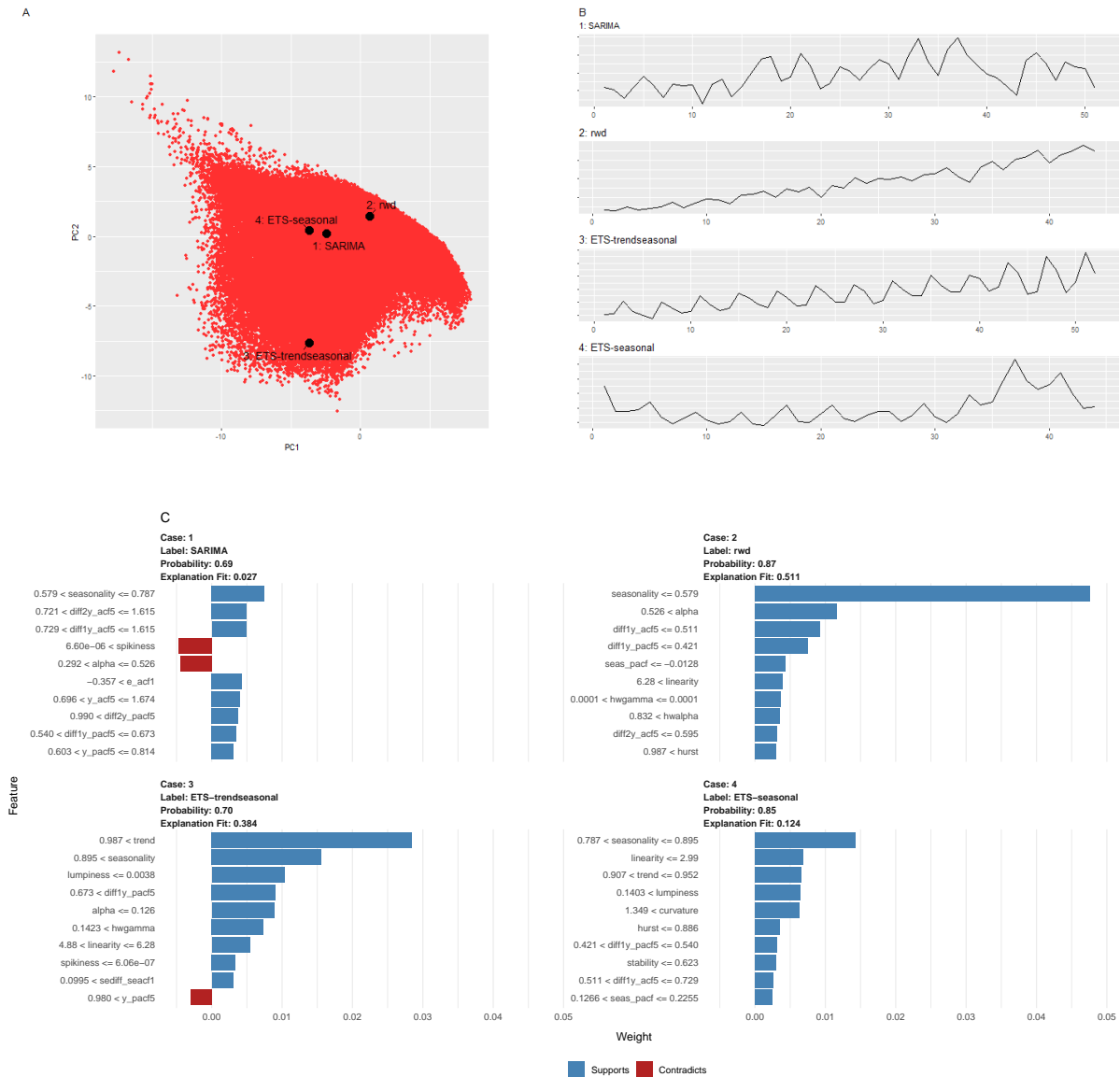


Figure 33: Panel A: Distribution of quarterly time series in the PCA space. Panel B: Time series corresponds to the highlighted points in the PCA space. Panel C: Local interpretable Model-agnostic explanations for four selected quarterly time series. Features denoted with green colour are supporting features for an outcome label and length of the bar is proportional to the weight of a feature.

We now illustrate how LIME approach can be used to zoom into local regions of the data to identify which features, contribute most to classify a specific instance. For the illustration we select four different time series classified with high probability. Figure 33 shows the feature contribution for the instances highlighted on the PCA-space of quarterly series. We can see how the strength of seasonality influences the FFORMS framework to select different types of seasonal forecast-models. For example, SARIMA model is selected when the seasonality varies between 0.579 and 0.787 (case 1), ETS-seasonal model is selected when the strength of seasonality is greater than 0.787 (case 4), random walk with drift when the seasonality is lower

than 0.579 (case 2) and for the highly trended and seasonal series (strength of seasonality > 0.895) ETS model with a trend and seasonal component is selected (case 3). Further, high value of `diff1y_acf5` in supports the selection of SARIMA for case 1 while, moderate value of `diff1y_acf5` supports the selection of ETS-seasonal for the case 4. Similarly, we can explore the reasons for other instances in all frequency categories. From LIME approach we can gain insight into the local neighbourhood characteristics which lead to the choice of a particular neighbourhood over alternative destinations.

8 Discussion and Conclusions

Features of time series are useful in identifying suitable models for forecasting. This paper explores the role of features in forecast-model selection under the hood of FFORMS framework. First, we evaluate the FFORMS prediction performance using the M4-Competition data. Then, we explore the dissimilarities and similarities of forecast-models (class-labels) learned by FFORMS framework based on vote-matrix information of the random forest. Next we explore **which** features are the most important for the for the choice of FFORMS framework. Further, we explore **where** they get most important: for the overall classification process or within a specific class of forecast-models or a set of multiple classes of forecast-models. Then partial dependency plots are used to visualize **when** and **how** these features linked with the prediction outcome of the FFORMS framework. Finally, we explore **when** and **how strongly** each feature interact with other features using Friedman's H-Statistic.

The features such as the strength of trend and strength of seasonality rank top within each class across all frequency categories. In addition to that, linearity, curvature, spikiness also rank among top 5 within most of the classes. Within most of the classes at least one feature related to autocorrelation and partial autocorrelation coefficients rank among the top 5. This confirms that the information regarding the correlation structure of the time series is an essential information for the choice of model selection. The length of time series (T) also appeared to be important in selecting simple forecast-models such as snaive, naive and random walk with drift. Partial dependency plots of `length(T)` shows that short time series tends to select simple forecast models such as snaive, naive and random walk with drift while more-parameterized forecast-models (SARIMA, ETS with trend and seasonal component, neural network approach, forecast models handle multiple seasonal components) often get selected with lengthy series. For all features, the displayed relationships of partial dependency plots consistent with the domain knowledge expectations. This is an important aspect in getting humans trust and use

the proposed framework effectively. Further, LIME approach is explore the reasons behind each individual prediction. This also helps to increase trust on framework because if the humans understand the reasons behind the results, they can use their prior knowledge about the application domain, which they can use to decide whether to accept (trust) or reject prediction outcome. The results of this helps to get a more refined picture of the relationship between features and the choice of forecast-model which is particularly valuable for ongoing research in the field of feature-based time series analysis.

Exploration of conditions learnt by the FFORMS framework also support practitioners to make a good educated guess on suitable forecast-model for a given problem. Further the results of this study is useful in identifying new ways to improve forecasting accuracy by capturing different features of time series.

Appendix: Interaction effects

[Figure 34-Figure 39](#) show how strongly each feature interact with any other feature in the forest for yearly, quarterly, monthly, weekly, daily and hourly series respectively. On each panel top 5 interacting features are highlighted in purple.

According [Figure 3](#) the variable importance plots for yearly data the feature `linearity` is assigned a very high variable importance within `rwd` and `ARMA`, but according to [Figure 34](#) `linearity` show very low interactivity within `rw` and `rwd`. This means that `linearity` is more important on its own rather than its interaction effect. [Figure 6](#) further confirms this as the partial dependency curves corresponds to `rwd` and `ARMA` drastically changes as `linearity` varies. According to [Figure 35](#) for quarterly data strength of seasonality show relatively low interactivity, indicating the importance of main effect of seasonality for the classification process. Furthermore, length of the time series show a high level of interactivity within random walk class. According to [Figure 36](#) and [Figure 37](#), in the process of classifying monthly and weekly series the number of features show high level (>0.5) of interactivity within classes are relatively low. This reflects the importance of individual effect of features when selecting forecast-models for monthly and weekly series. Weekly data FFORMS framework show high level of interactivity of features within `ARMA`. This due to the class imbalance. Weekly series reference data set contains very small number of `ARMA` labeled series. Hence, features interact highly to separate `ARMA` from the rest. Similar to the results of weekly and monthly data, according to [Figure 38](#) the number of features that show high level of interactivity (>0.7) with other features are relatively low. For

hourly time series, according to Figure 39 shows high level of interactivity within many classes.

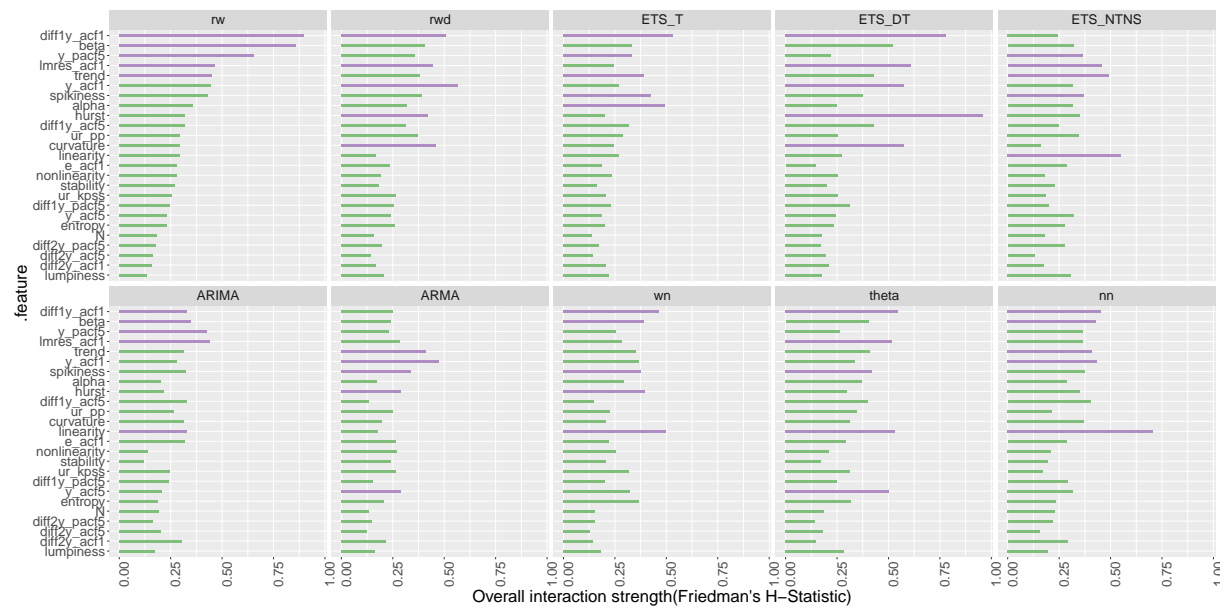


Figure 34: The intection strength (Friedman's H-Statistic) for each feature with any other feature for yearly forest. The Hurst exponent within ETS_DT has the highest interaction effect.



Figure 35: The intection strength (Friedman's H-Statistic) for each feature with any other feature for quarterly forest. Top 5 interacting features are highlighted in purple. Strength of seasonality shows an extremely low interactivity within SARIMA, ETS_DT and ETS_DTS.



Figure 36: The interaction strength (Friedman's H-Statistic) for each feature with any other feature for quarterly forest. Top 5 interacting features are highlighted in purple. Curvature shows highest level of interactivity within ETS_DTS

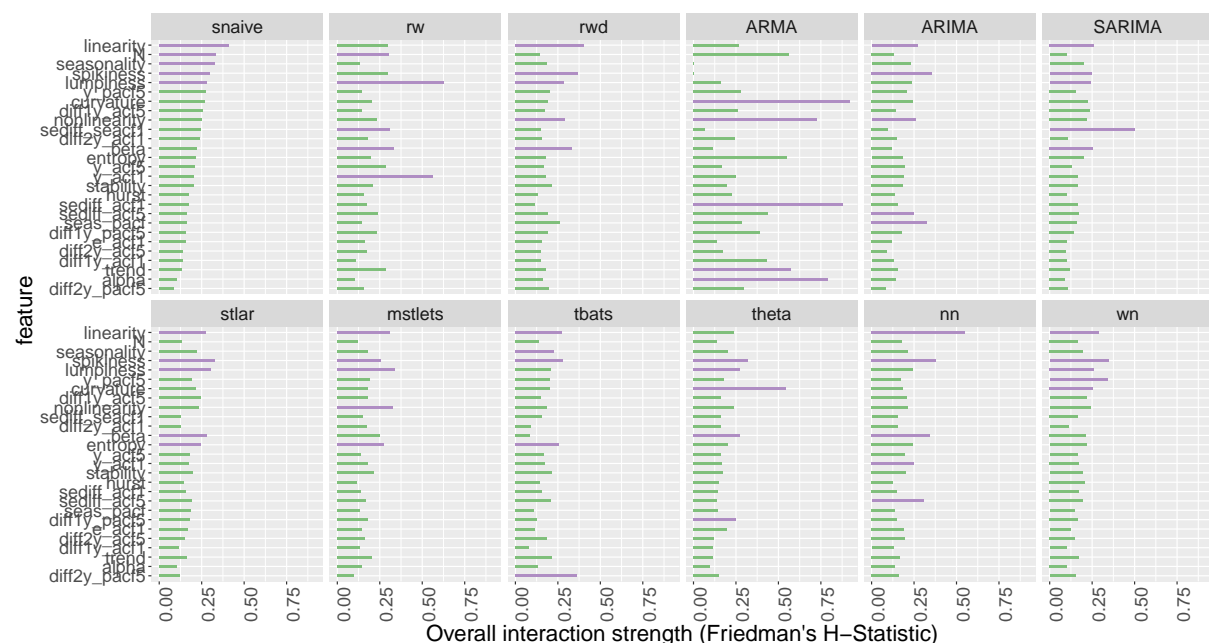


Figure 37: The interaction strength (Friedman's H-Statistic) for each feature with any other feature for weekly forest. Overall, the interaction effects between the features are very weak.

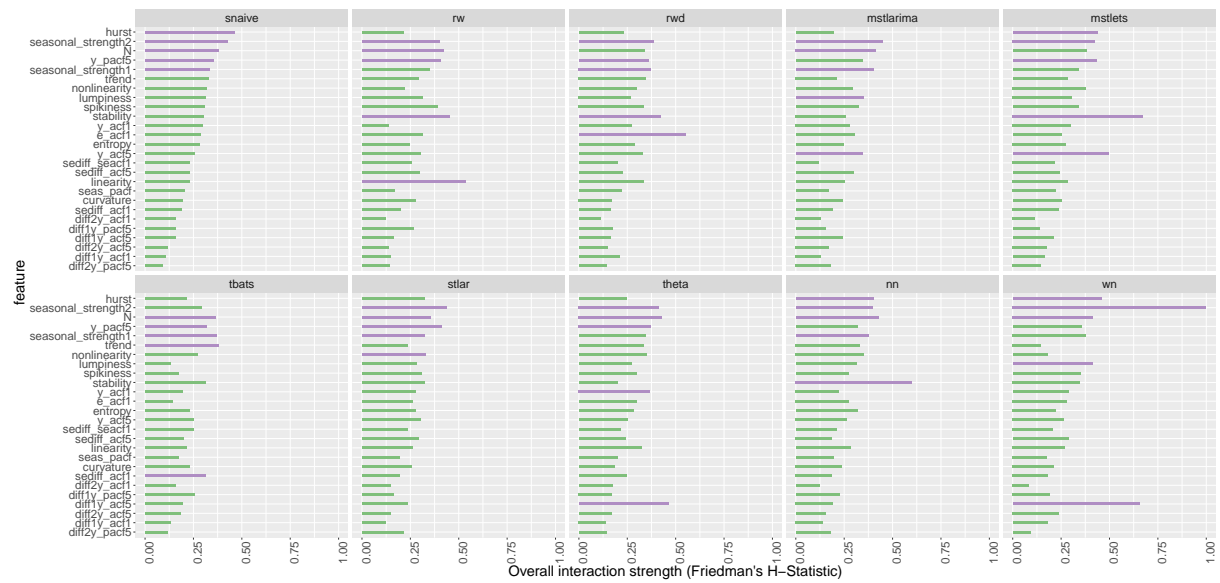


Figure 38: The interaction strength (Friedman's H-Statistic) for each feature with any other feature for daily forest. Strength of trend shows highest level of interactivity within white noise class.

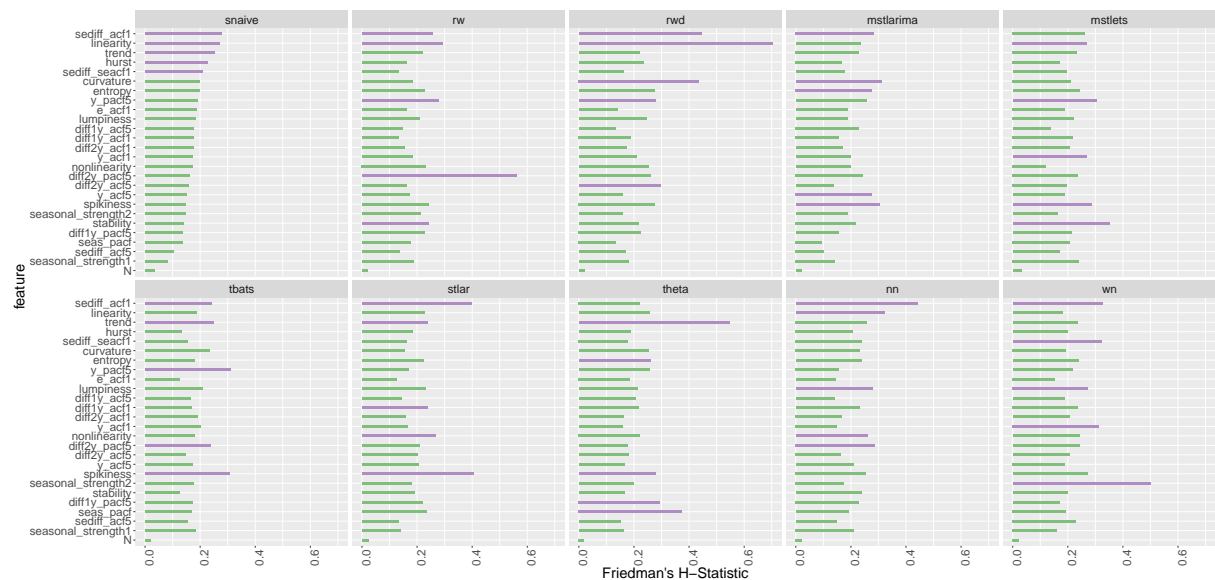
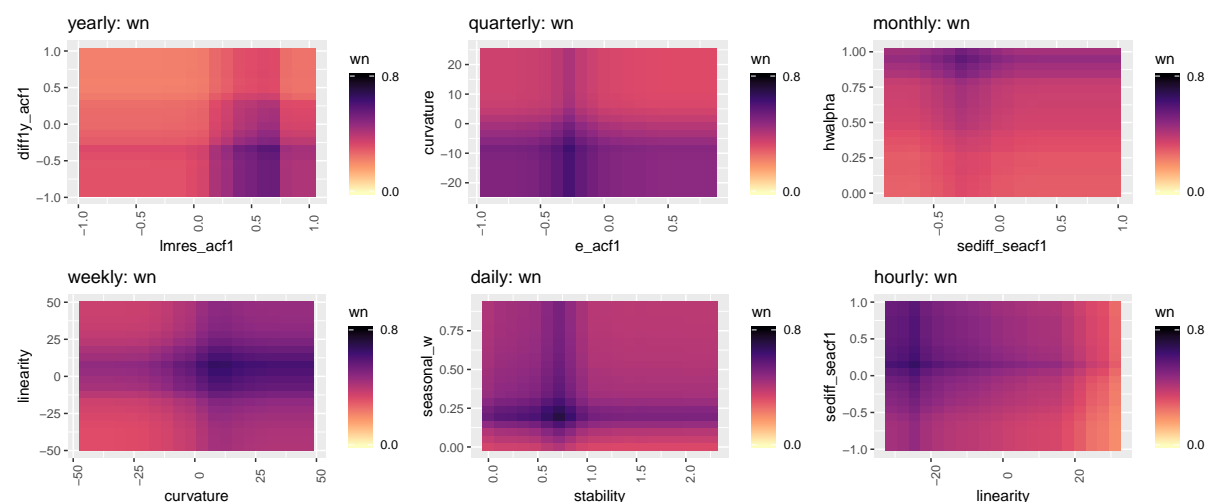


Figure 39: The interaction strength (Friedman's H-Statistic) for each feature with any other feature for hourly forest. Linearity shows high level of interactivity within random walk with drift.



References

- Breiman, L (2001). Random forests. *Machine Learning* **45**(1), 5–32.
- Collopy, F & JS Armstrong (1992). Rule-based forecasting: development and validation of an expert systems approach to combining time series extrapolations. *Management Science* **38**(10), 1394–1414.
- Friedman, JH, BE Popescu, et al. (2008). Predictive learning via rule ensembles. *The Annals of Applied Statistics* **2**(3), 916–954.
- Goldstein, A, A Kapelner, J Bleich & E Pitkin (2015). Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *Journal of Computational and Graphical Statistics* **24**(1), 44–65.
- Greenwell, BM, BC Boehmke & AJ McCarthy (2018). A Simple and effective model-based variable importance measure. *arXiv preprint arXiv: 1805.04755*.
- Hyndman, R, G Athanasopoulos, C Bergmeir, G Caceres, L Chhay, M O’Hara-Wild, F Petropoulos, S Razbash, E Wang & F Yasmeeen (2018). *forecast: Forecasting functions for time series and linear models*. R package version 8.5. <http://pkg.robjhyndman.com/forecast>.
- Jiang, T & AB Owen (2002). Quasi-regression for visualization and interpretation of black box functions. *Technical Report, Stanford University*.
- Kück, M, SF Crone & M Freitag (2016). Meta-learning with neural networks and landmarking for forecasting model selection an empirical evaluation of different feature sets applied to industry data. In: *Neural Networks (IJCNN), 2016 International Joint Conference on*, pp.1499–1506.
- Lemke, C & B Gabrys (2010). Meta-learning for time series forecasting and forecast combination. *Neurocomputing* **73**(10), 2006–2016.
- Lundberg, SM & SI Lee (2017). A unified approach to interpreting model predictions. In: pp.4765–4774.
- M4 Competitor’s Guide (2018). <https://www.m4.unic.ac.cy/wp-content/uploads/2018/03/M4-Competitors-Guide.pdf>. Accessed: 2018-09-26.
- Makridakis, S & M Hibon (2000). The M3-Competition: results, conclusions and implications. *International Journal of Forecasting* **16**(4), 451–476.
- Meade, N (2000). Evidence for the selection of forecasting methods. *Journal of forecasting* **19**(6), 515–535.
- Petropoulos, F, S Makridakis, V Assimakopoulos & K Nikolopoulos (2014). ‘Horses for Courses’ in demand forecasting. *European Journal of Operational Research* **237**(1), 152–163.

- Prudêncio, RB & TB Ludermir (2004). Meta-learning approaches to selecting time series models. *Neurocomputing* **61**, 121–137.
- Ribeiro, MT, S Singh & C Guestrin (2016). Why should I trust you?: Explaining the predictions of any classifier. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'16)*. ACM, New York, NY, USA, pp.1135–1144.
- Schnaars, SP (1984). Situational factors affecting forecast accuracy. *Journal of Marketing Research*, 290–297.
- Shah, C (1997). Model selection in univariate time series forecasting using discriminant analysis. *International Journal of Forecasting* **13**(4), 489–500.
- Talagala, TS, RJ Hyndman & G Athanasopoulos (2018). Meta-learning how to forecast time series. *Technical Report 6/18, Monash University, Department of Econometrics and Business Statistics*.
- Wang, X, K Smith-Miles & RJ Hyndman (2009). Rule induction for forecasting method selection: meta-learning the characteristics of univariate time series. *Neurocomputing* **72**(10), 2581–2594.
- Zhao, Q & T Hastie (2017). Causal interpretations of black-box models. *Journal of Business & Economic Statistics*.