

Peeking inside FFORMS: Feature-based FOfecast 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 FOfecast 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). 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 modelled?; 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 FOREcast 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 (35 different feature types are used which are simple and easy to compute), 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 the relationship between features of time series and the choice of forecast-model selection using the FFORMS framework. We explore the role of features in two different perspectives: i) Global perspective of feature contribution: the overall role of features in the choice of different forecast-models and ii) Local perspective of feature contribution: we zoom into local regions of the data to identify which features contribute most to classify a specific instance. In this category we make five fundamental contributions. They are,
 - i) visualise patterns learned by the meta-learner
 - ii) rank features according to their importance in the choice of forecast-model selection and identify which features are contributing mostly to the predictive mechanism of the fitted model.
 - iii) visualise learned relationships between features and the choice of forecast-model selection using partial dependence plots introduced by Friedman, Popescu, et al. (2008).
 - iv) Assessment of interaction effect between features
 - v) visualise regions in the feature-space in which good or exceptional performance is expected from different forecast-models.

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. In [Section 6](#) we discuss the results. [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 what is happening under the hood of the FFORMS framework.

We now explain application of the extended 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.

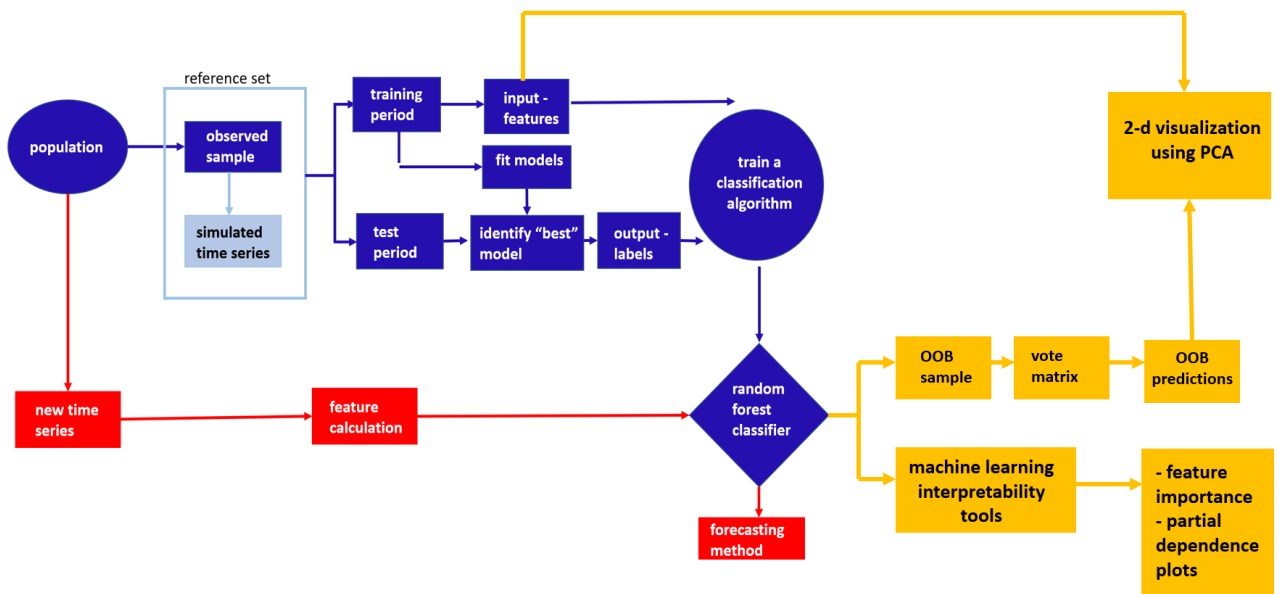


Figure 1: extended FFORMS (Feature-based FOREcast-Model Selection) framework. The offline phase is shown in blue, the online phase by red and the peeking inside FFORMS is shown in yellow.

3 Offline phase

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. Table 1 summarizes the number of time series in the observed sample. Note that from the M4 competition a randomly selected subset of time series are used for the observed sample. The rest shown by the column labelled “Test set - M4” are used as a validation set to evaluate the performance of the meta-learner.

Table 1: *Composition of the time series in the observed sample and the test set*

Frequency	Observed Sample			Test set
	M1	M3	M4	M4
Yearly	181	645	22000	1000
Quarterly	203	756	23000	1000
Monthly	617	1428	47000	1000
Weekly	-	-	259	100
Daily	-	-	4001	226
Hourly	-	-	350	64

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. Table 2 shows the data generating algorithms used for each frequency. 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.

Table 2: *Automatic forecasting algorithms used to simulate time series*

Algorithm	Y	Q	M	W	D	H
ets()	✓	✓	✓			
auto.arima()	✓	✓	✓			
stlf()				✓	✓	✓

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 3. A comprehensive description of the features used in the experiment is given in Talagala, Hyndman & Athanasopoulos (2018).

3.3 Output: class-labels

The description of class labels considered under each frequency is shown in Table 4. Note that these added to Talagala, Hyndman & Athanasopoulos (2018). Most of the labels given in Table 4 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 4 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 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.

Table 3: *Time series features*

	Feature	Description	Y	Q/M	W	D/H
1	T	length of time series	✓	✓	✓	✓
2	trend	strength of trend	✓	✓	✓	✓
3	seasonality 1	strength of seasonality corresponds to frequency 1	-	✓	✓	✓
4	seasonality 2	strength of seasonality corresponds to frequency 2	-	-	-	✓
5	linearity	linearity	✓	✓	✓	✓
6	curvature	curvature	✓	✓	✓	✓
7	spikiness	spikiness	✓	✓	✓	✓
8	e_acf1	first ACF value of remainder series	✓	✓	✓	✓
9	stability	stability	✓	✓	✓	✓
10	lumpiness	lumpiness	✓	✓	✓	✓
11	entropy	spectral entropy	✓	✓	✓	✓
12	hurst	Hurst exponent	✓	✓	✓	✓
13	nonlinearity	nonlinearity	✓	✓	✓	✓
14	alpha	ETS(A,A,N) $\hat{\alpha}$	✓	✓	✓	-
15	beta	ETS(A,A,N) $\hat{\beta}$	✓	✓	✓	-
16	hwalpha	ETS(A,A,A) $\hat{\alpha}$	-	✓	-	-
17	hwbeta	ETS(A,A,A) $\hat{\beta}$	-	✓	-	-
18	hwgamma	ETS(A,A,A) $\hat{\gamma}$	-	✓	-	-
19	ur_pp	test statistic based on Phillips-Perron test	✓	-	-	-
20	ur_kpss	test statistic based on KPSS test	✓	-	-	-
21	y_acf1	first ACF value of the original series	✓	✓	✓	✓
22	diff1y_acf1	first ACF value of the differenced series	✓	✓	✓	✓
23	diff2y_acf1	first ACF value of the twice-differenced series	✓	✓	✓	✓
24	y_acf5	sum of squares of first 5 ACF values of original series	✓	✓	✓	✓
25	diff1y_acf5	sum of squares of first 5 ACF values of differenced series	✓	✓	✓	✓
26	diff2y_acf5	sum of squares of first 5 ACF values of twice-differenced series	✓	✓	✓	✓
27	seas_acf1	autocorrelation coefficient at first seasonal lag	-	✓	✓	✓
28	sediff_acf1	first ACF value of seasonally-differenced series	-	✓	✓	✓
29	sediff_seacf1	ACF value at the first seasonal lag of seasonally-differenced series	-	✓	✓	✓
30	sediff_acf5	sum of squares of first 5 autocorrelation coefficients of seasonally-differenced series	-	✓	✓	✓
31	seas_pacf	partial autocorrelation coefficient at first seasonal lag	-	✓	✓	✓
32	lmres_acf1	first ACF value of residual series of linear trend model	✓	-	-	-
33	y_pacf5	sum of squares of first 5 PACF values of original series	✓	✓	✓	✓
34	diff1y_pacf5	sum of squares of first 5 PACF values of differenced series	✓	✓	✓	✓
35	diff2y_pacf5	sum of squares of first 5 PACF values of twice-differenced series	✓	✓	✓	✓

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. 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.

Table 4: *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		-	-	-	✓

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. Global interpretability evaluates the behavior of a model on entire data set. Global perspective of model interpretation helps users to understand the overall modelled 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

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 sample.

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 $\tilde{\mathcal{B}}^{(k)}$ 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:

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

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.

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 Results:

7 Results: Peeking inside FFORMS

7.1 Yearly data

This vote-matrix information for the yearly data is presented in Figure 2. Figure 2 indicates three main types of information. They are, i) the distributions of correctly classified classes dominate, indicating a good classification of the meta-learner, ii) the random walk with drift has a high chance of getting selected with yearly time series and the results of M4-competition also show the random walk with drift perform well with yearly time 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.

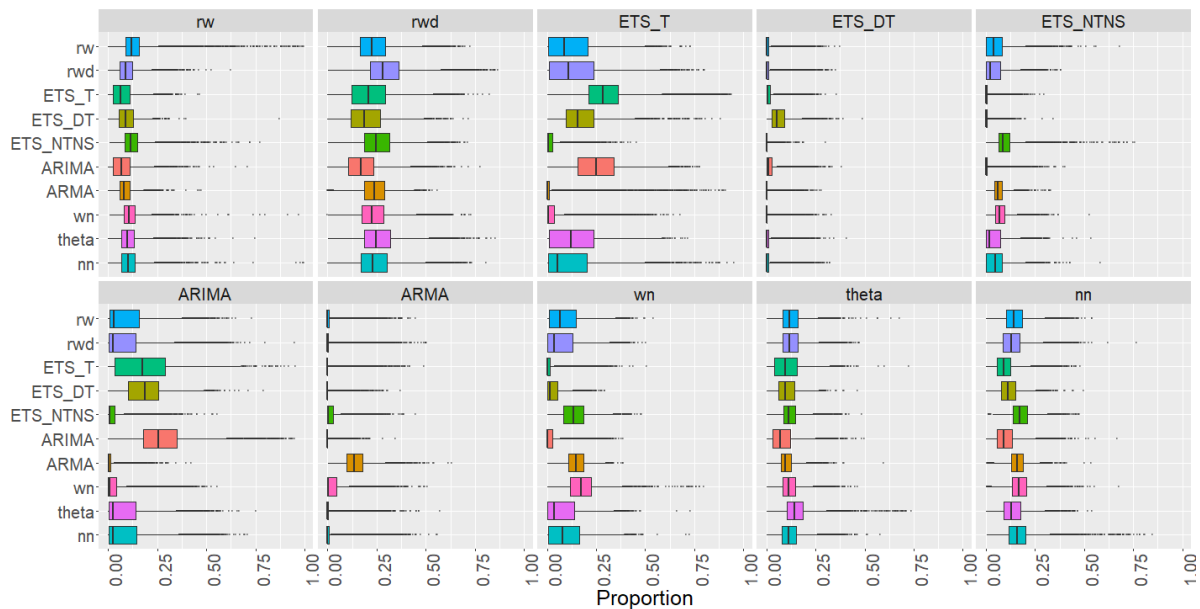


Figure 2: Visualization of the vote matrix based on OOB sample for yearly random forest. Each panel shows the predicted class from the random forest. The X-axis denotes the proportion of times each time series was classified to each class. The colours of boxplots correspond to class label of the “best” 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 within each class as well as in the overall classification mechanism of the FFORMS. The strength of trend, 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. The first ACF value of the original series appears to be important with in rwd and ARMA class. The reason could be, the first ACF value can easily separate very clear stationary and nonstationary models. The features related to

different types of trend (linear trend: measured by linearity, 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. The features related to randomness of the series such as spikiness and `lmres_acf` are also assigned a high importance. The length of time series (N) is assigned a very low importance within random walk class.

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

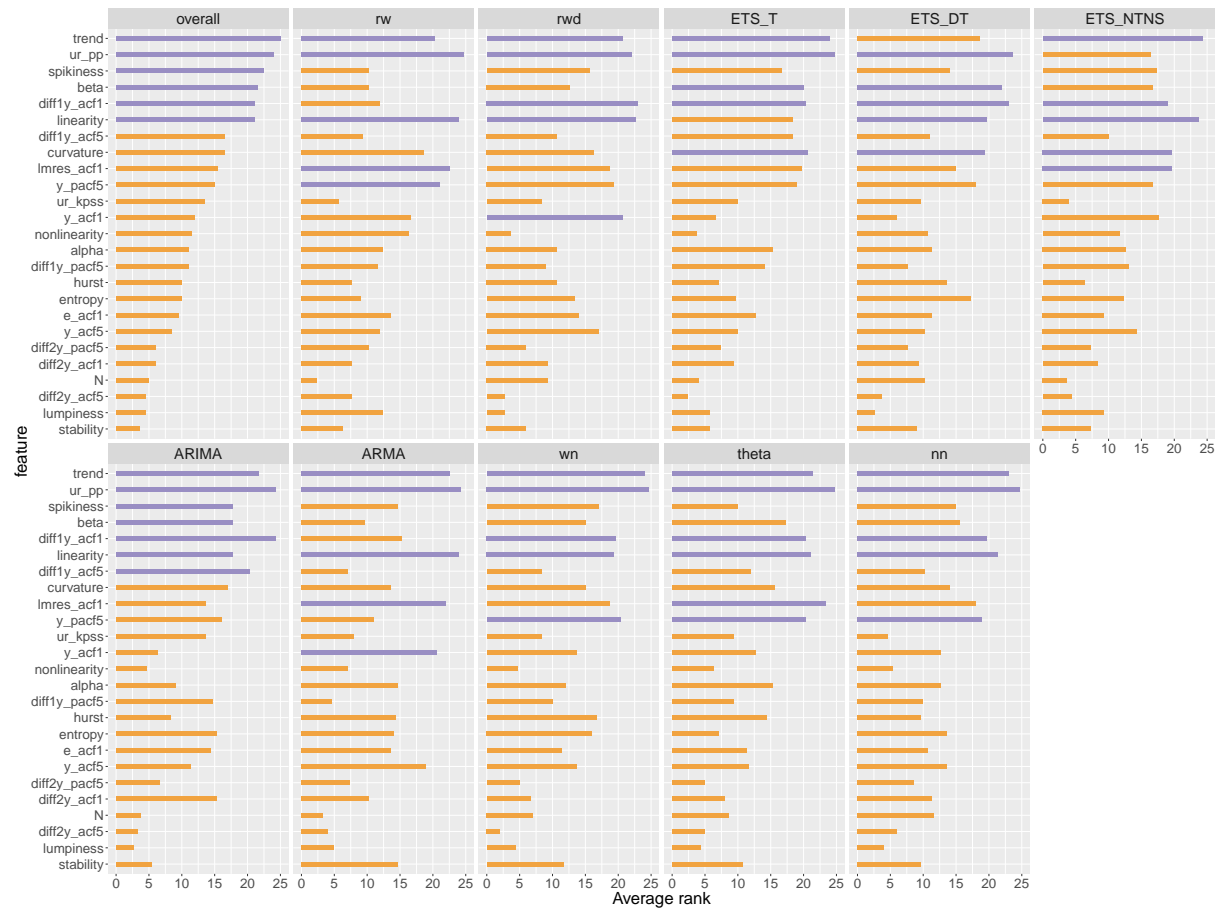


Figure 3: Feature importance plot for yearly series. Overall feature importance (top left plot) is evaluated based on two measures: i) Permutation-based variable importance measure and ii) mean decrease in Gini coefficients are used to evaluate shown in the top left plot. Class-specific feature importance 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 trend appears to be the most important feature.

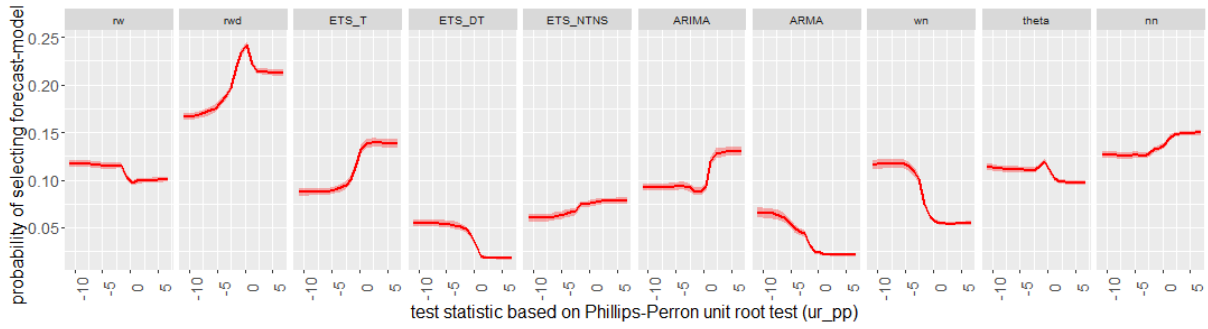


Figure 4: Partial dependence plots for ur_pp . The shading shows the 95% confidence intervals. Y-axis denotes the probability of belong to corresponding class. All features show a nonlinear relationship with predicted probabilities.

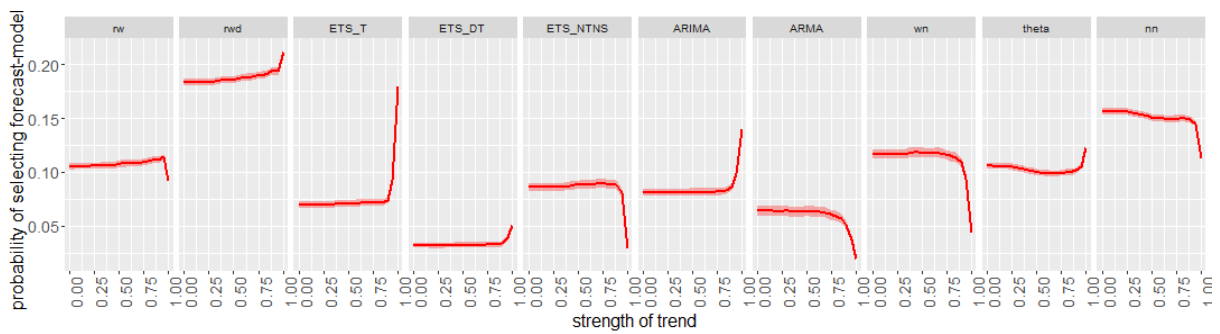


Figure 5: Partial dependence plots for trend. The shading shows the 95% confidence intervals. Y-axis denotes the probability of belong to corresponding class. All features show a nonlinear relationship with predicted probabilities.

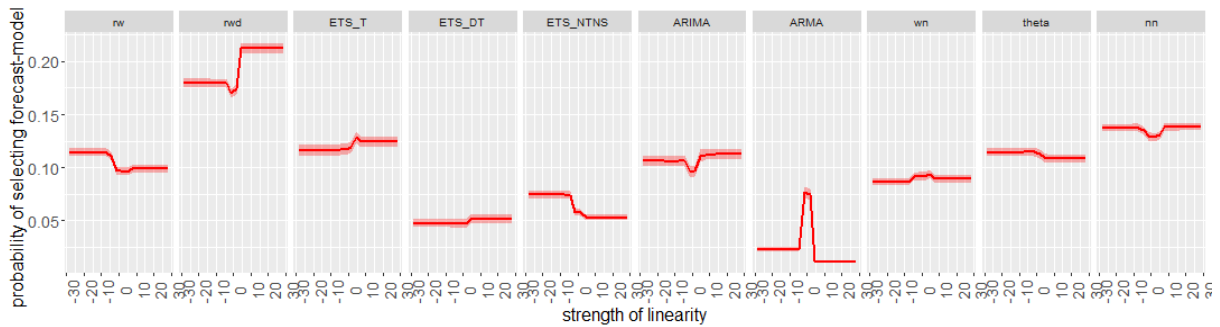


Figure 6: Partial dependence plots for linearity. The shading shows the 95% confidence intervals. Y-axis denotes the probability of belong to corresponding class. All features show a nonlinear relationship with predicted probabilities.

7.2 Quarterly

Figure 7 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 tbats panel all distributions located further away from zero, indicating all time series selected that model at least once from the individual trees in the forest. Within the panels theta, and nn all distributions level at similar proportionalities.



Figure 7: 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 8 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.

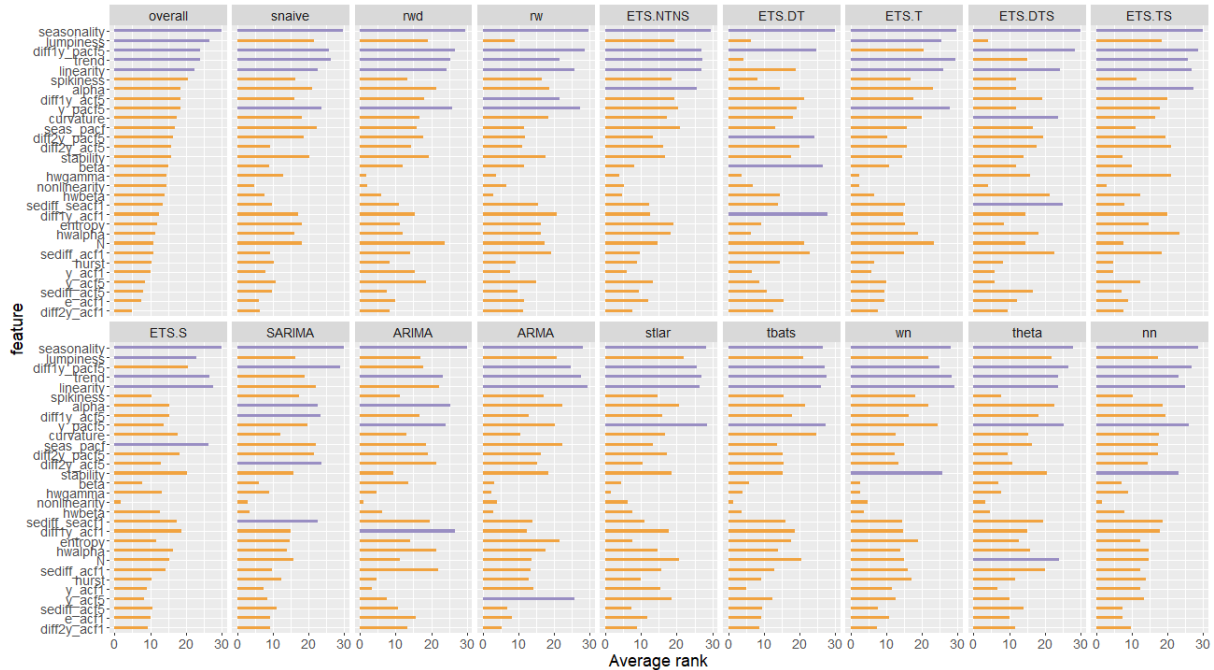


Figure 8: Feature importance plot for quarterly series. Overall feature importance (top left plot) is evaluated based on two measures: i) Permutation-based variable importance measure and ii) mean decrease in Gini coefficients are used to evaluate shown in the top left plot. Class-specific feature importance 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.

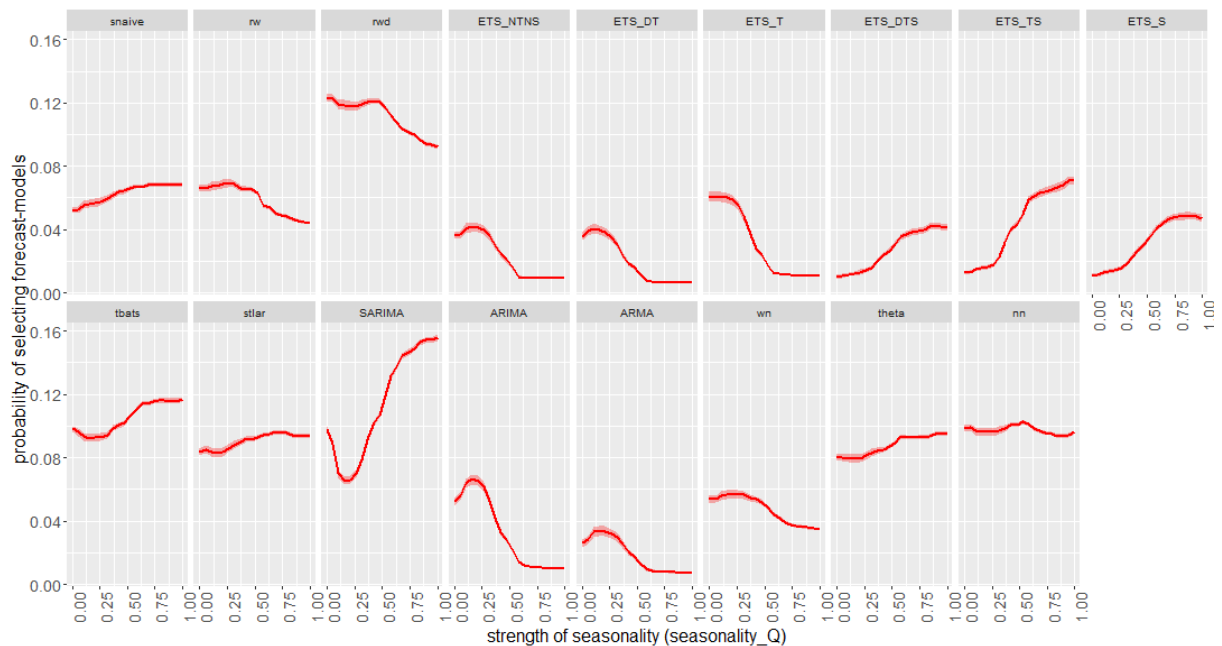


Figure 9: Partial dependence plots for strength of seasonality (seasonality_Q). The shading shows the 95% confidence intervals. Y-axis denotes the probability of belong to corresponding class. All features show a nonlinear relationship with predicted probabilities.

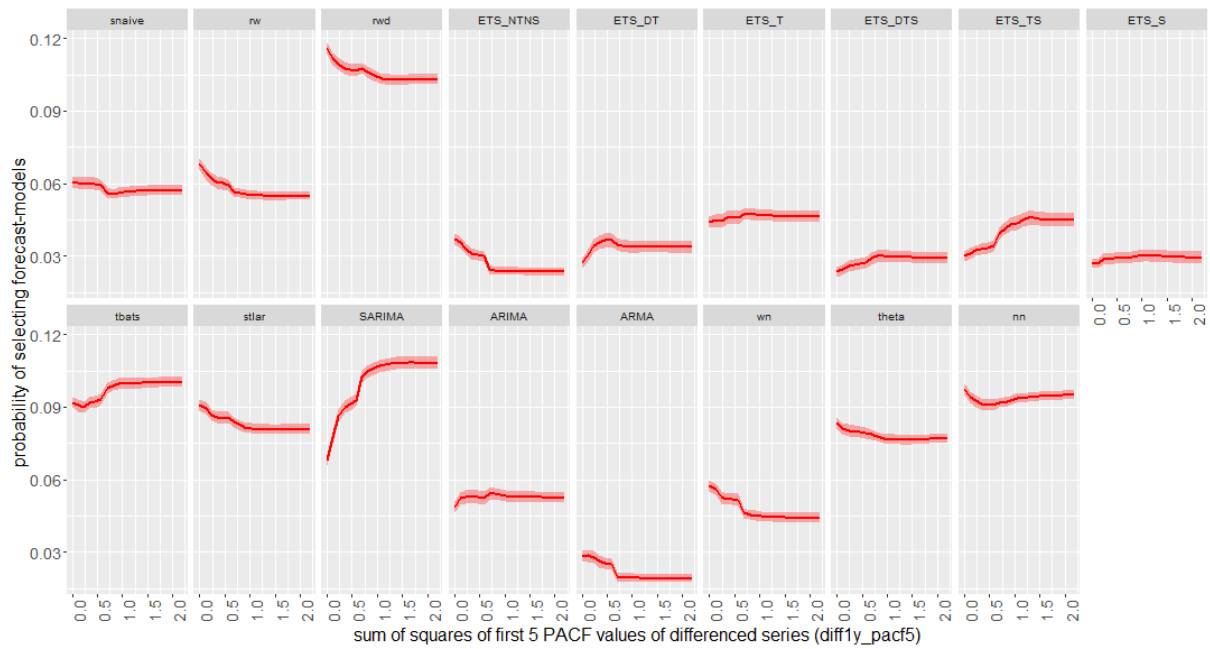


Figure 10: Partial dependence plots for sum of squares of first 5 PACF values of differenced series (*diff1y_pacf5*). The shading shows the 95% confidence intervals. Y-axis denotes the probability of belong to corresponding class. All features show a nonlinear relationship with predicted probabilities.

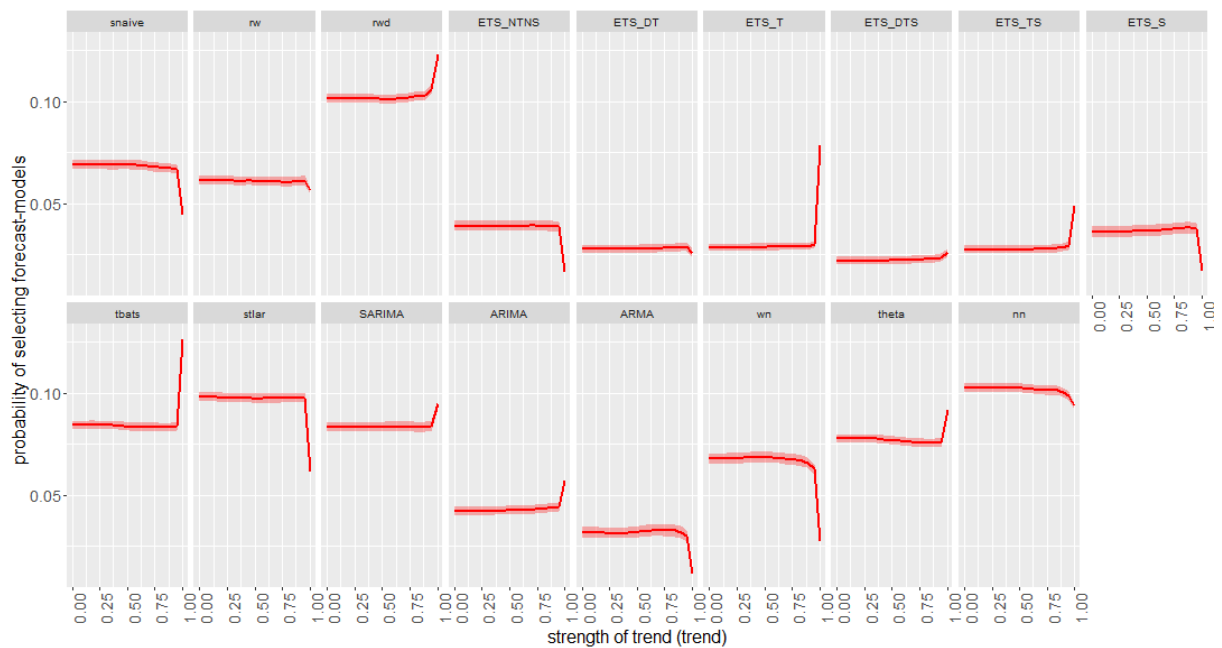


Figure 11: Partial dependence plots for strength of trend (*trend*). The shading shows the 95% confidence intervals. Y-axis denotes the probability of belong to corresponding class. All features show a nonlinear relationship with predicted probabilities.

Figure 9 to Figure 11 show the partial dependency curves, and associated confidence intervals of the top-three features that get selected most in each class. According to Figure 9, it is immediately apparent, that the probability of selecting a seasonal forecast-model increases as the strength of seasonality increases. According to Figure 9 and Figure 10 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.

7.3 Monthly

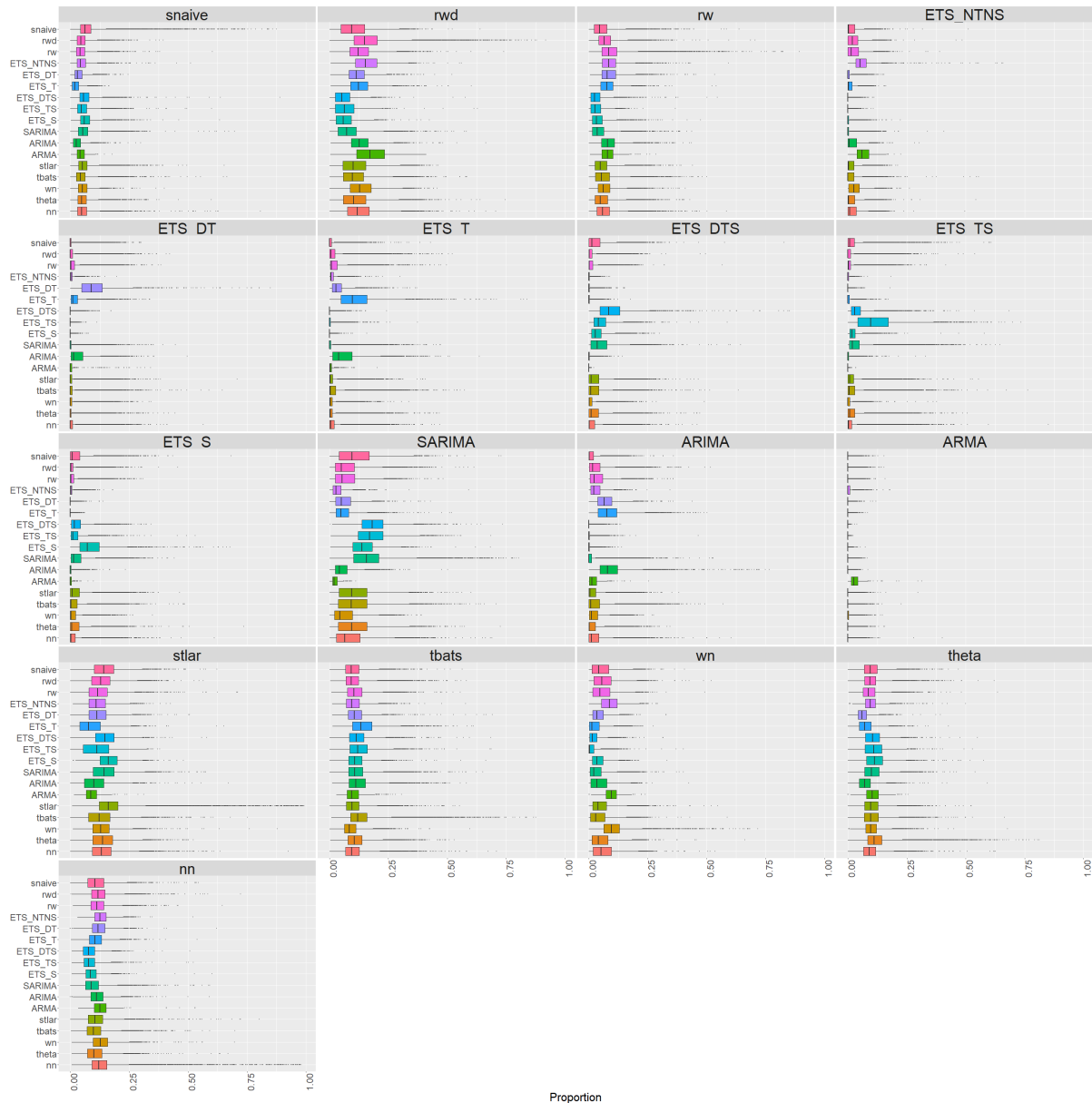


Figure 12: Visualization of the vote matrix based on OOB sample for quarterly random forest. Each panel shows the predicted class from the random forest. The X-axis denotes the proportion of times each time series was classified to each class. The colours of boxplots correspond to class label of the “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 7 shows the vote-matrix information for monthly series. Figure 7 and Figure 12 depicted similar patterns across classes. 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.

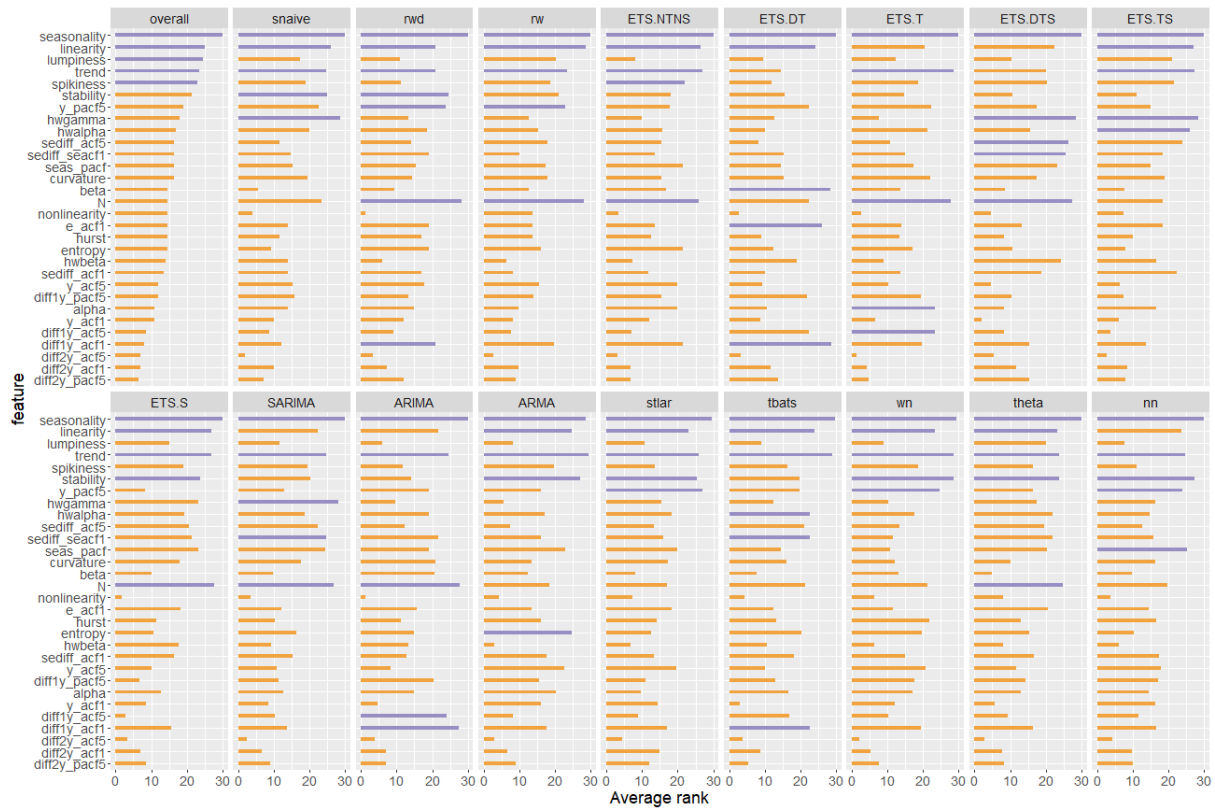


Figure 13: Feature importance plot for monthly series. Overall feature importance (top left plot) is evaluated based on two measures: i) Permutation-based variable importance measure and ii) mean decrease in Gini coefficients are used to evaluate shown in the top left plot. Class-specific feature importance 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 13 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 data 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.

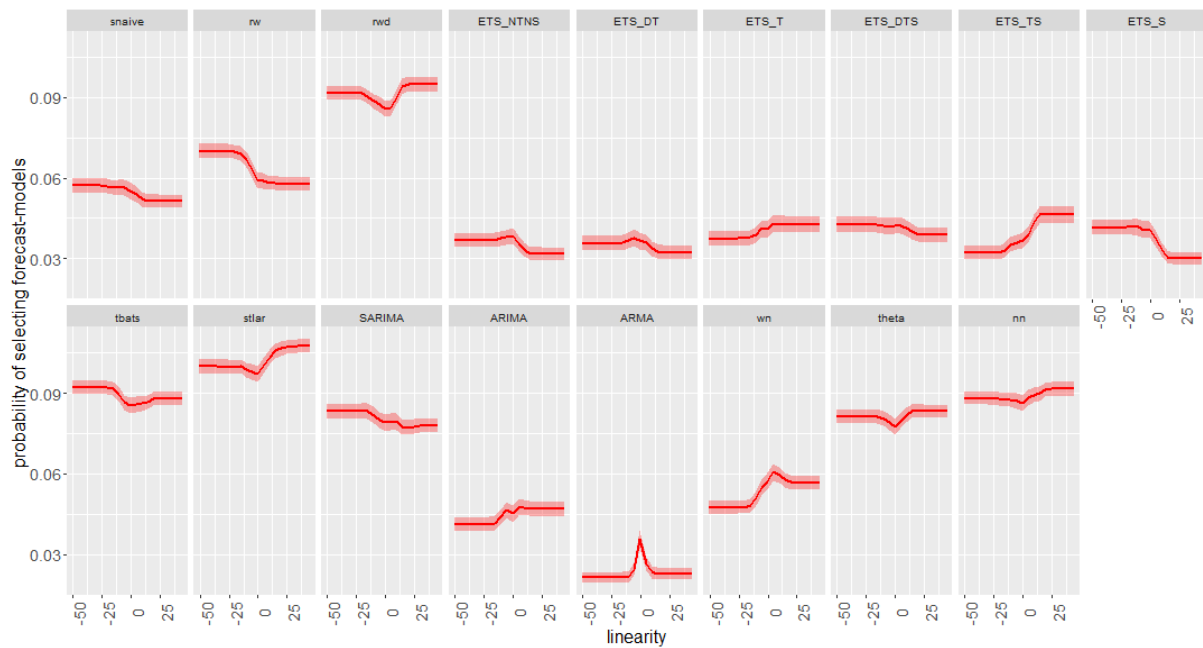


Figure 14: Partial dependence plots for linearity. The shading shows the 95% confidence intervals. Y-axis denotes the probability of belong to corresponding class. All features show a nonlinear relationship with predicted probabilities.

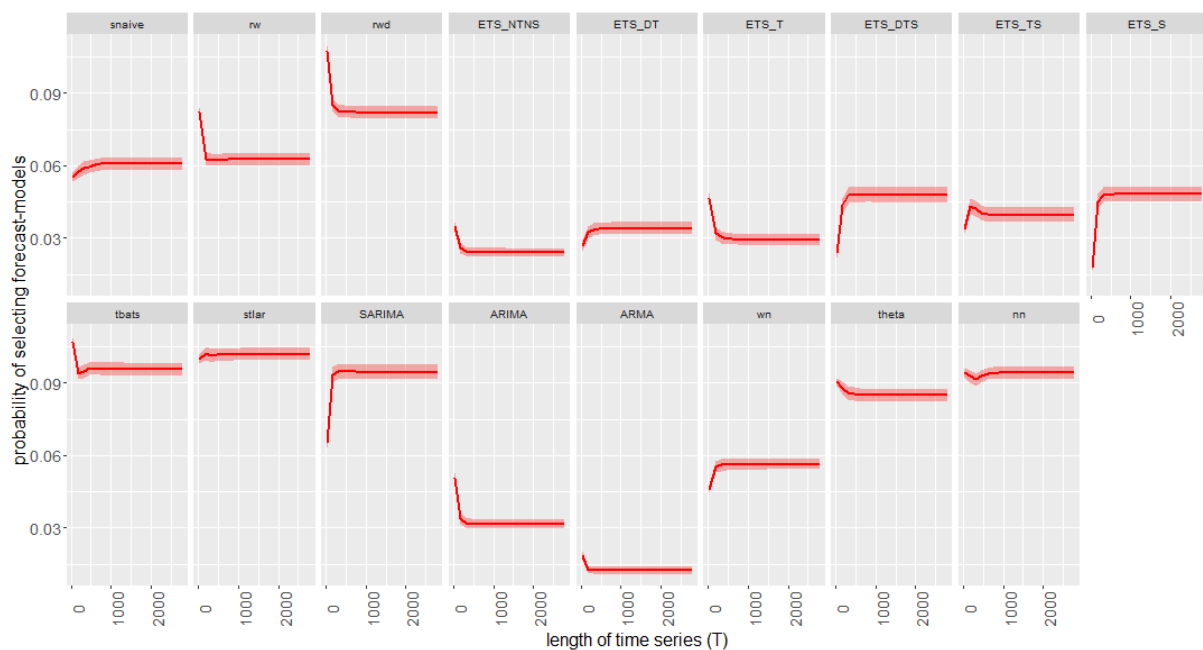


Figure 15: Partial dependence plots for length of time series (T). The shading shows the 95% confidence intervals. Y-axis denotes the probability of belong to corresponding class. All features show a nonlinear relationship with predicted probabilities.

7.4 Weekly

Figure 16 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.

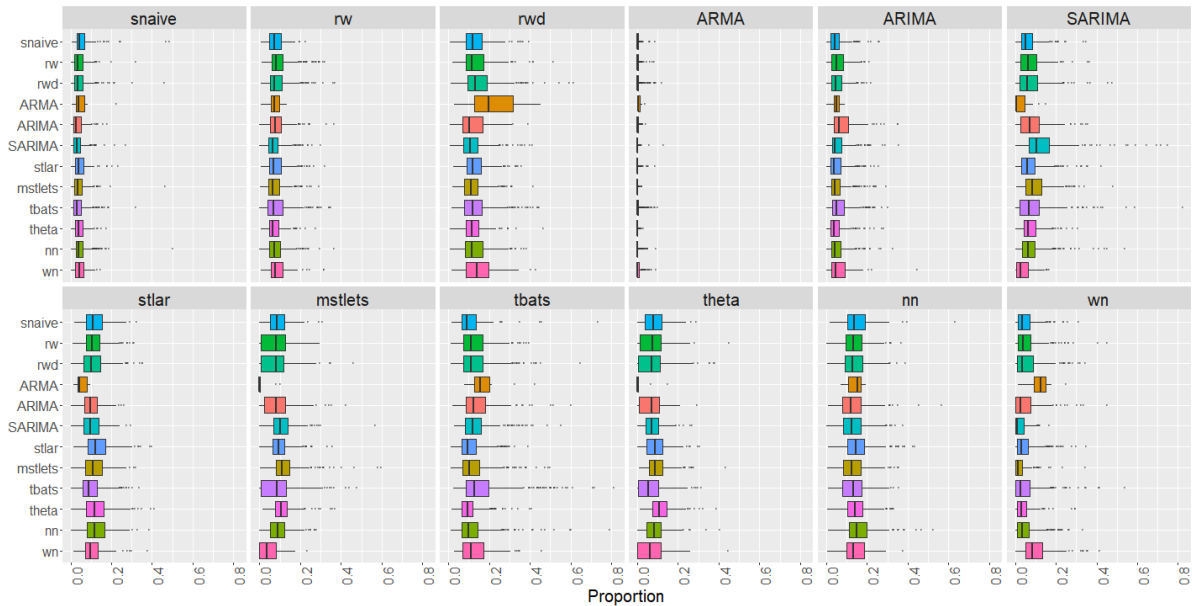


Figure 16: Visualization of the vote matrix based on OOB sample for weekly random forest. The Y-axis denotes what was predicted from the random forest. The X-axis denotes the proportion of times each time series was classified to each class. The colours of boxplots corresponds to class label of the “best” 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 17 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 18 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.

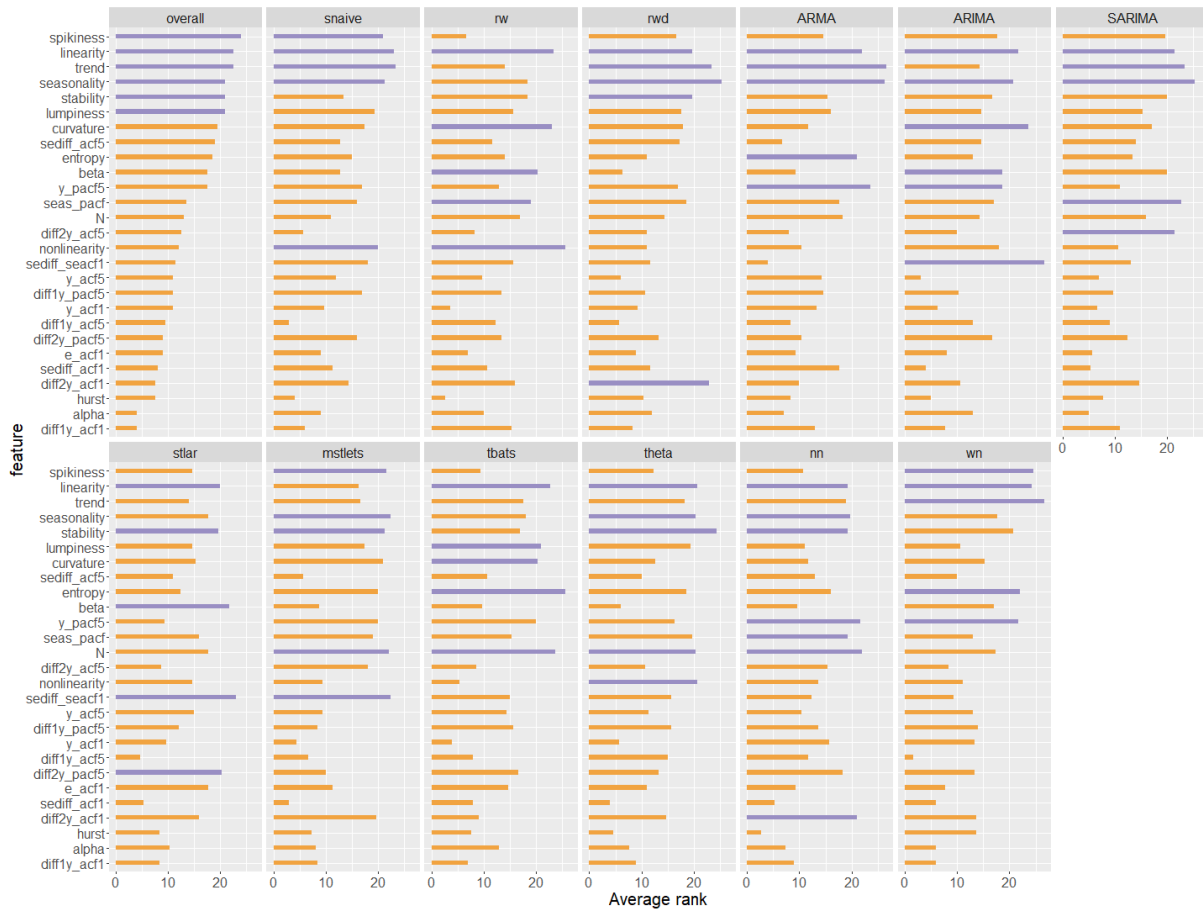


Figure 17: Feature importance plot for weekly data. Permutation-based VI measure and mean decrease in Gini coefficient is used to evaluate overall feature importance. Class-specific feature importance is evaluated based on the three measures: permutation-based VI, PD-based VI measure, and ICE-based VI measure. Longer bars indicate more important features. Top 5 features are highlighted in red.

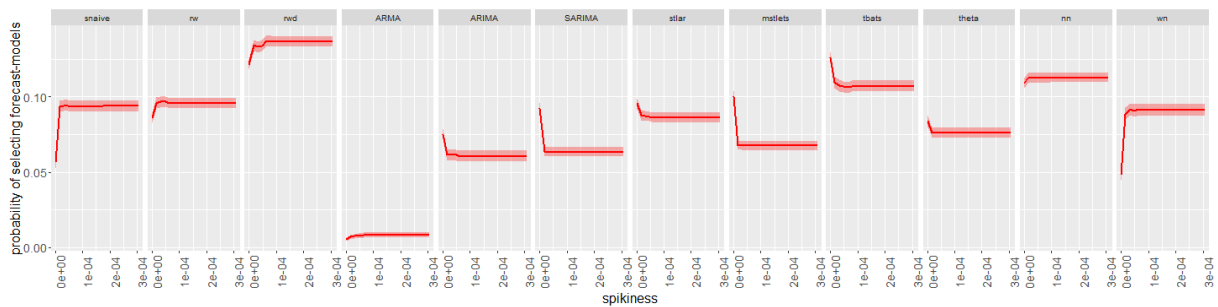


Figure 18: Partial dependence plots for the top ranked features from variable importance measures (weekly series). The shading shows the 95% confidence intervals. Y-axis denotes the probability of belonging to corresponding class.

7.5 Hourly

Figure 19 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.

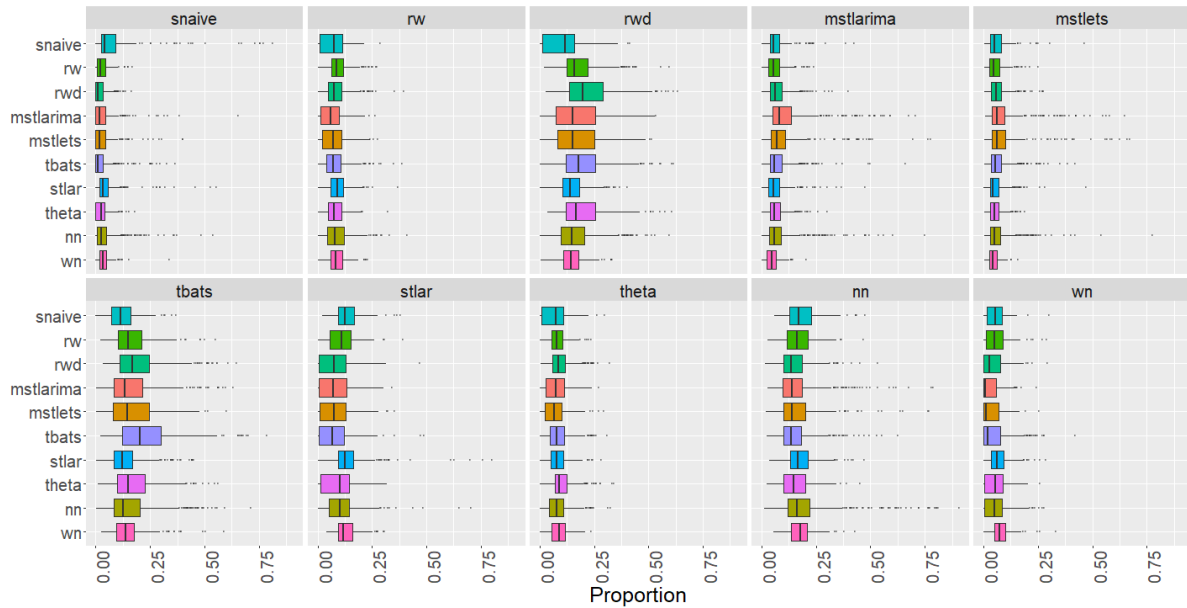


Figure 19: Visualization of the vote matrix based on OOB sample for hourly random forest. The Y-axis denotes what was predicted from the random forest. The X-axis denotes the proportion of times each time series was classified to each class. The colours of boxplots corresponds to class label of the “best”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

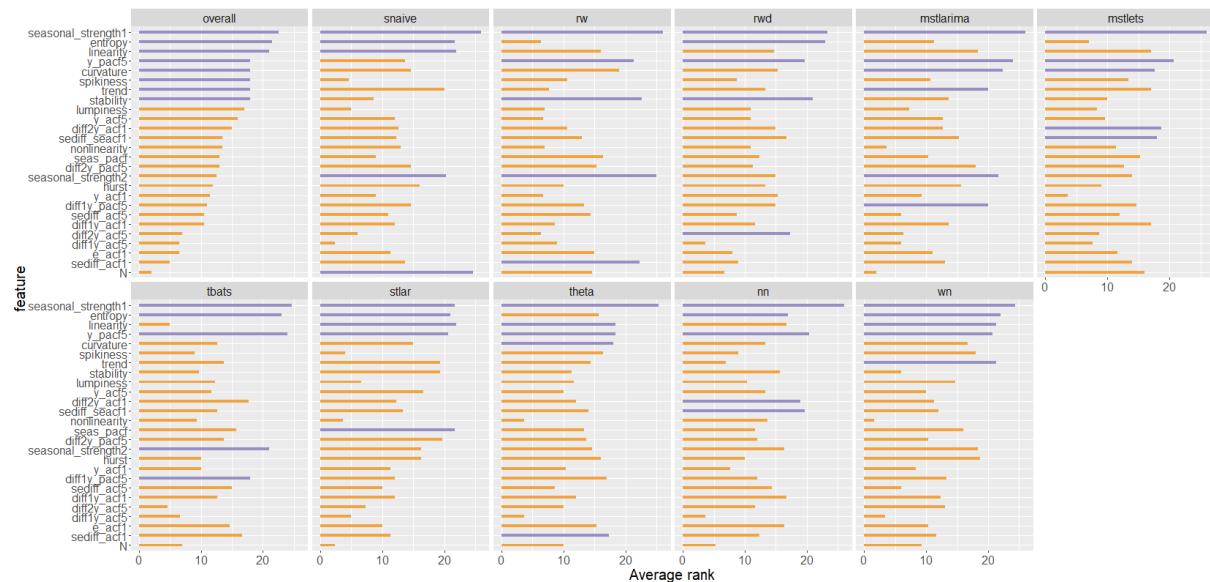


Figure 20: Feature importance plot hourly series. Permutation-based VI measure and mean decrease in Gini coefficients are used to evaluate overall feature importance. Class-specific feature importance is evaluated based on the three measures: permutation-based VI, PD-based VI measure, and ICE-based VI measure. Longer bars indicate more important features. Top 5 features are highlighted in red.

Variable importance graph for hourly data is shown in Figure 20. According to Figure 20, 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.

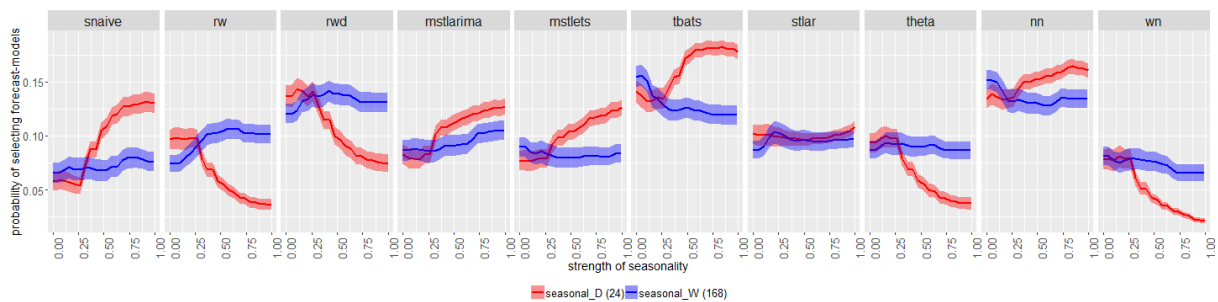


Figure 21: Partial dependence plots for strength of seasonality. The shading shows the 95% confidence intervals. Y-axis denotes the probability of belonging to corresponding class.

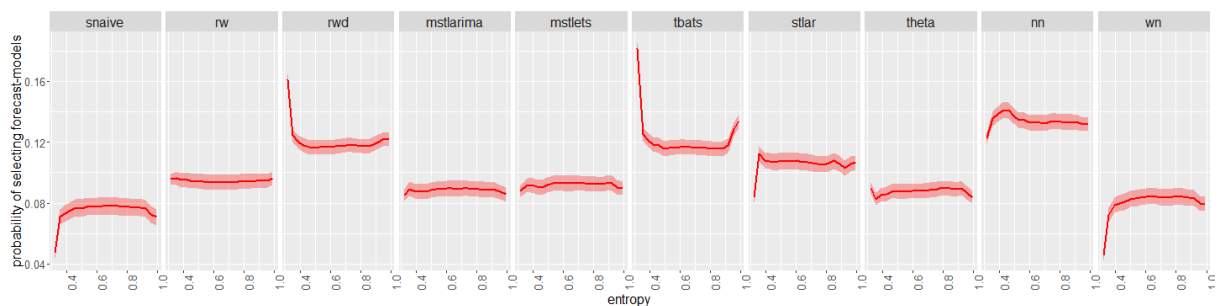


Figure 22: Partial dependence plots for entropy. The shading shows the 95% confidence intervals. Y-axis denotes the probability of belonging to corresponding class.

The partial dependency plots of the strength of seasonalities are shown in [Figure 21](#). According to [Figure 21](#) 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 22](#). 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.

7.6 Daily

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.

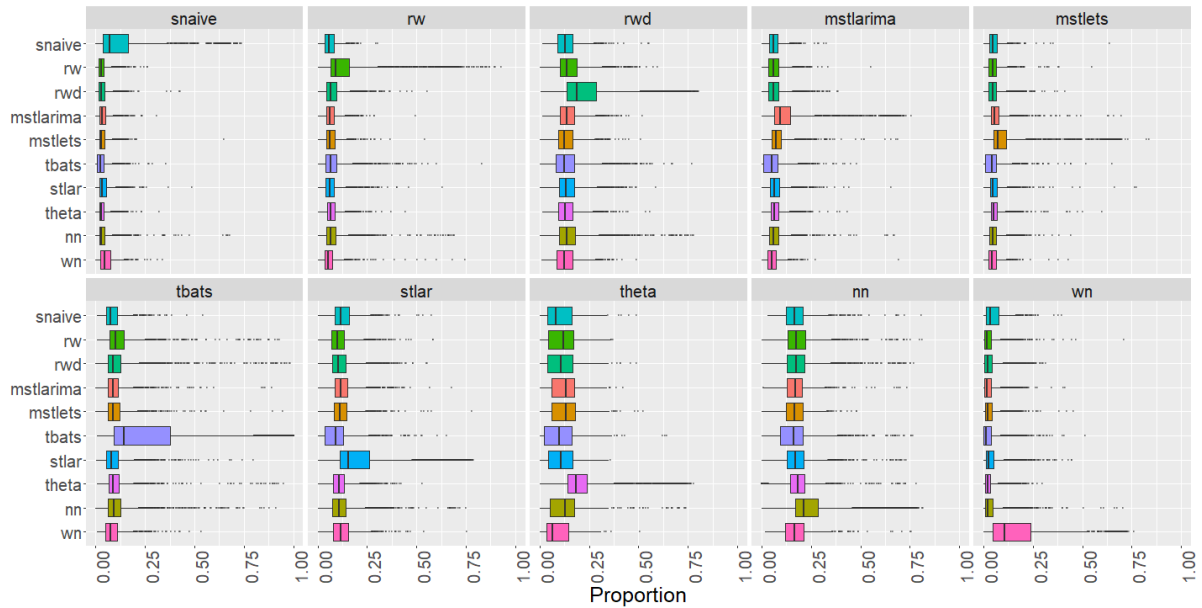


Figure 23: Visualization of the vote matrix based on OOB sample for daily random forest. The Y-axis denotes what was predicted from the random forest. The X-axis denotes the proportion of times each time series was classified to each class. The colours of boxplots corresponds to class label of the “best”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

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.

?? shows the partial dependency plots of the top 3 features for daily series. According to the results of ?? 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.

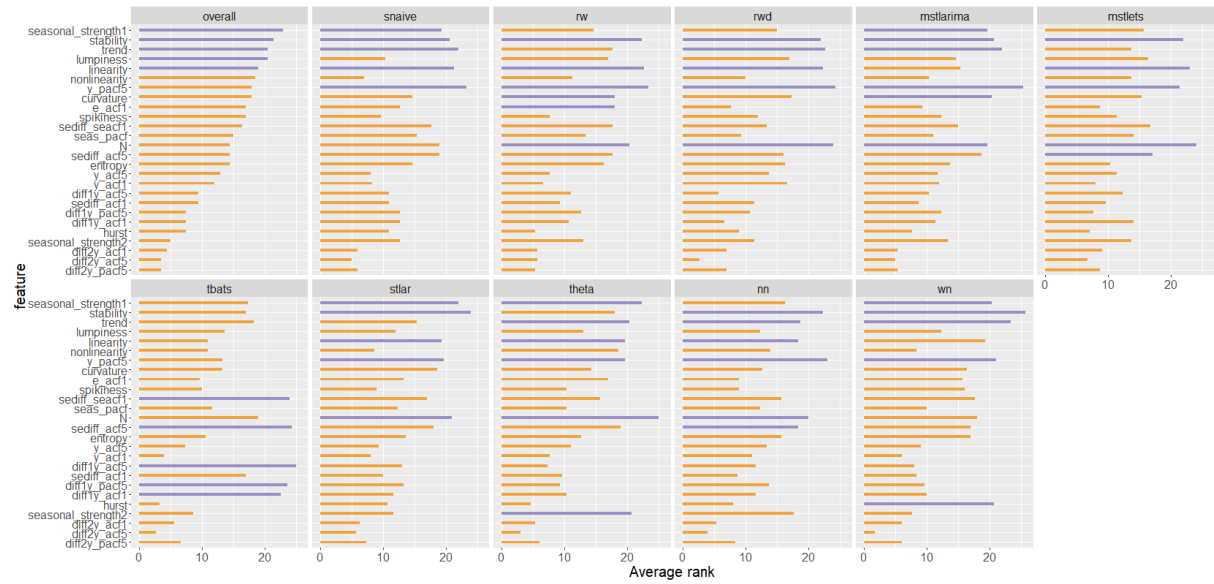


Figure 24: Feature importance plot for daily data. Permutation-based VI measure and mean decrease in Gini coefficients are used to evaluate overall feature importance. Class-specific feature importance is evaluated based on the three measures: permutation-based VI, PD-based VI measure, and ICE-based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

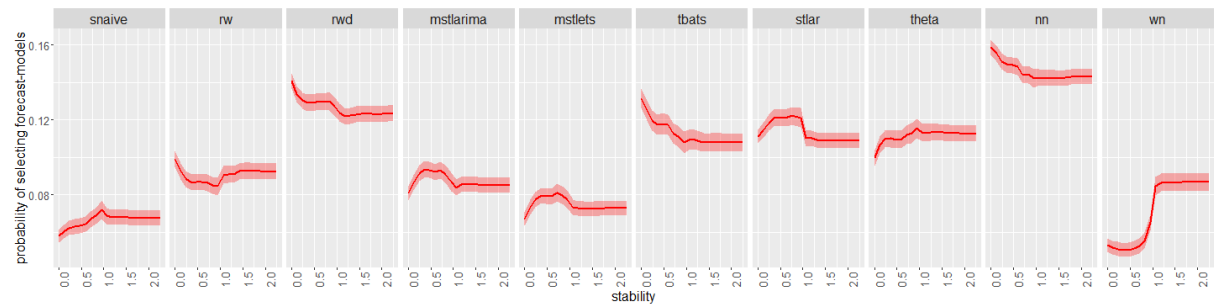


Figure 25: Partial dependence plots for stability. The shading shows the 95% confidence intervals. Y-axis denotes the probability of belonging to corresponding class.



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

7.7 Assessment of interaction effect

Figure 27 to Figure 32 show partial dependency plot for most frequently appeared two-way interacting feature combinations. Most frequently appeared two-way interacting feature combi-

nations were identified based on the Friedman's H-Statistics calculations.

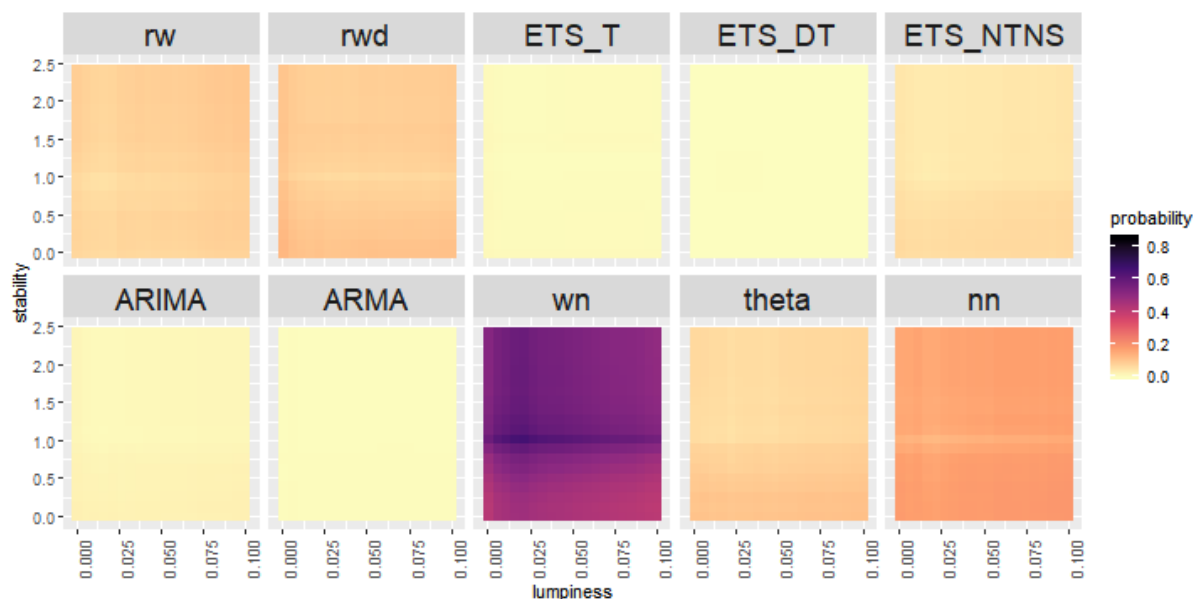


Figure 27: Two-way partial dependency plot for interaction between stability and lumpiness for yearly data.

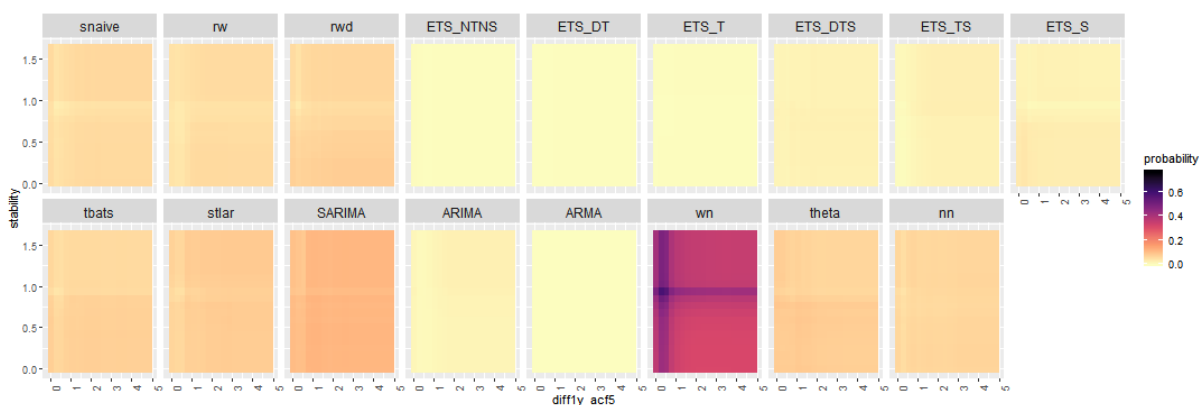


Figure 28: Two-way partial dependency plot for interaction between stability and diff1y_acf5 for quarterly data.

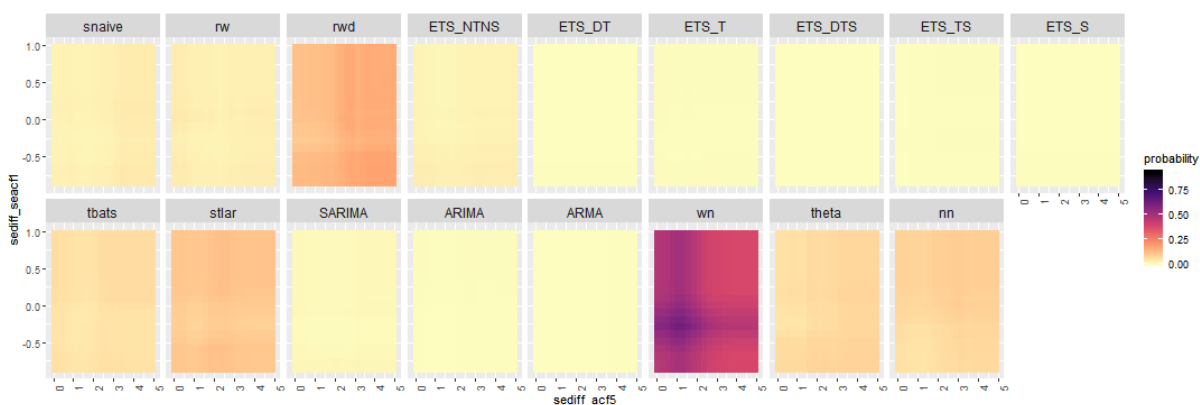


Figure 29: Two-way partial dependency plot for interaction between sediff_acf5 and sediff_seacf1 for monthly data.

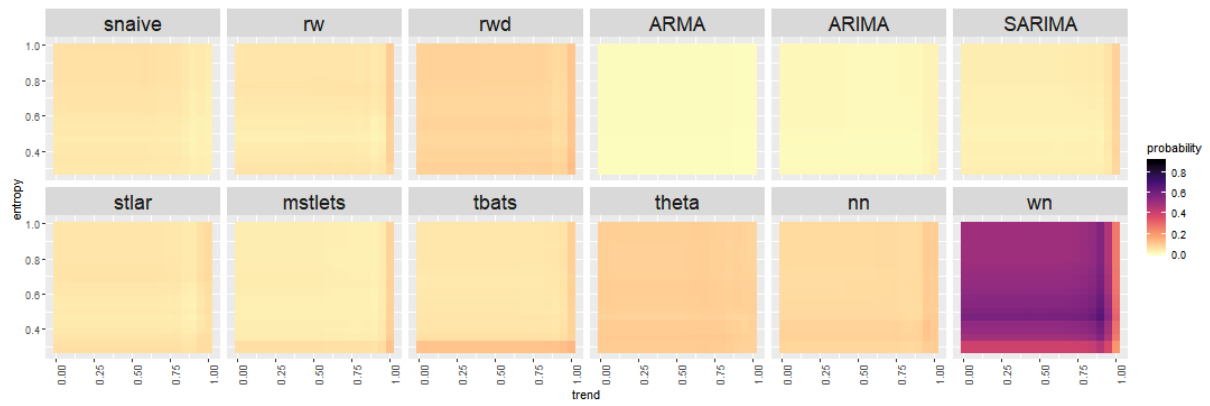


Figure 30: Two-way partial dependency plot for interaction between trend and entropy for weekly data.

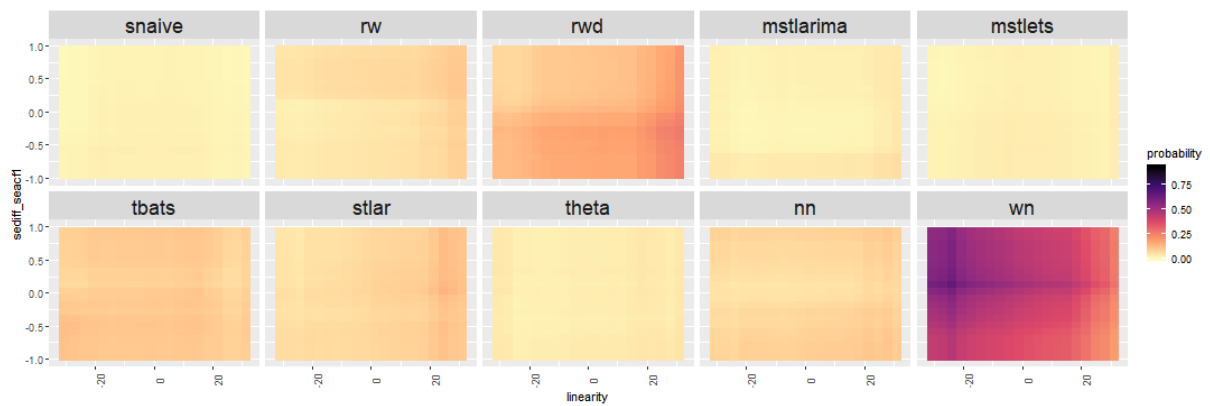


Figure 31: Two-way partial dependency plot for interaction between sediff_seacf1 and linearity for hourly data.

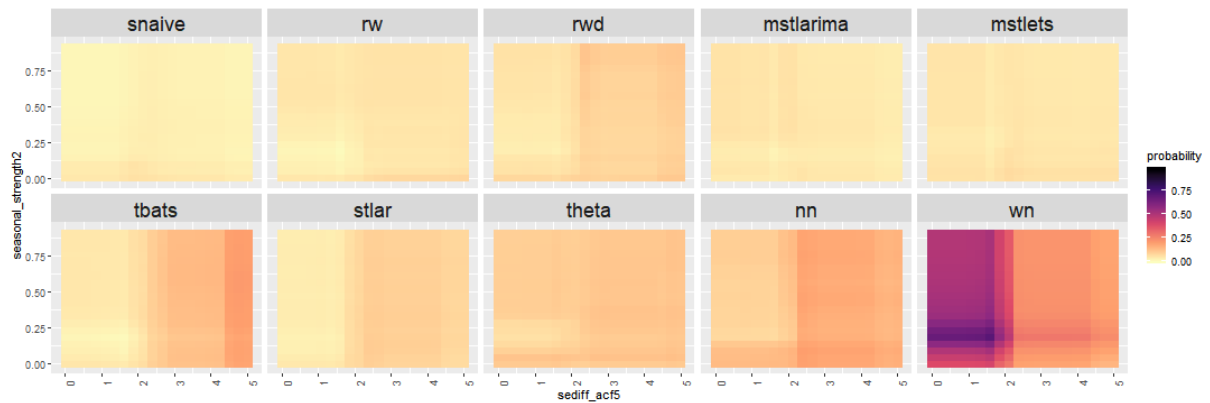


Figure 32: Two-way partial dependency plot for interaction between sediff_acf5 and seasonality_W for daily data.

7.8 Local Interpretable Model-agnostic Explanations

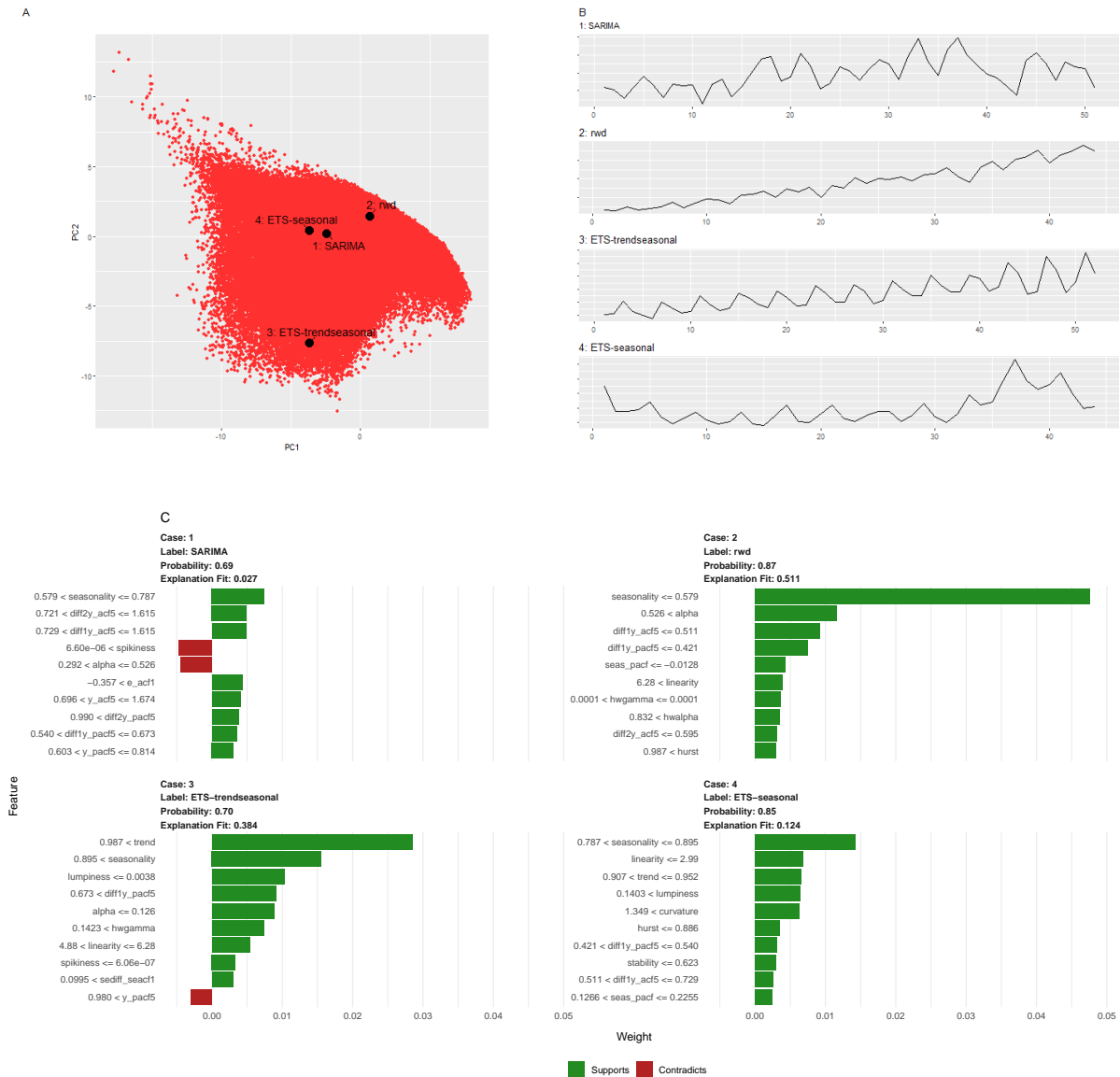


Figure 33: Panel A: Distribution of quarterly time series in the PCA space. Panel B: Time series corresponds to the highlighted points in the PCA space. Panel C: Local interpretable Model-agnostic explanations for four selected quarterly time series. Features denoted with green colour are supporting 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

Forecast-model selection is both time and cost intensive process. Consequently, the application of machine-learning approaches to predict suitable forecast-model from a large number of potentially relevant time series features is a topic growing popularity in the field of time series forecasting. In this paper we use model-agnostic machine learning interpretability tools to explore what is happening under the hood of FFORMS framework and to gain an understanding of what features led to the choices of FFORMS framework. The results we present here are a novel application of machine learning interpretability methods to visualize and explore the role of features in forecast-model selection. Furthermore, explaining predictions is an important aspect in getting humans trust and use the proposed framework effectively, if the explanations are faithful and intelligible. Humans usually have prior knowledge about the application domain, which they can use to accept (trust) or reject prediction if they understand the reasons behind it.

Wickham, Cook & Hofmann (2015) explain the importance of displaying the “model in the data space (m-in-ds)” and “data in the model space (d-in-ms)”. Displaying the data in the model space (d-in-ms) is the most commonly used approach for model-diagnostics. For example, plot of fitted values versus residuals (Wickham, Cook & Hofmann (2015)). D-in-MS is a visualization of embedding high-dimensional data into a low-dimensional space generated from the model. Visualization of D-in-MS do not help to gain an understanding of the nature of the relationship between features predicted outcome. In order to address this issue Wickham, Cook & Hofmann (2015) and Silva, Cook & Lee (2017) have highlighted the importance of visualizations of model in the data space. In the context of classification, representation of m-in-ds could be achieved by first, projecting the training data set into meaningful low-dimensional feature space and then visualize the complete prediction regions or their boundaries. In other words, this can be considered as the visualization of predictor space in the context of the data space. See Wickham,

Cook & Hofmann (2015) for visualization method of this kind and Silva, Cook & Lee (2017) for comparable method for random forest algorithm.

We explore the role of features in two different perspectives: i) individual effect, and ii) interaction effect. The features, strength of trend, strength of seasonality, linearity, spikiness and curvature rank among the top 10 within each frequency category. Lemke & Gabrys (2010) also pointed out features related to nonstationarity and seasonality of a series are important factors for choosing a forecast-models. Partial dependency plots are used to visualize the learned relationship between features and the model predictions. The displayed relationships confirm to domain knowledge expectations. However, since several numbers of features are used to build the framework with comparable contributions, and thus, all individual contributions are small. Furthermore, our results show that the performance of various methods depend upon the length of the time series. Short time series tends to select simple methods such as random walk models, snaive, etc. ETS models with both trend and seasonal components, SARIMA models, mstl models tend to provide accurate forecasts with longer time series as these are more parameterized models.

As FFORMS framework is developed on top of random forest algorithm, it takes into account every possible interaction. It was apparent from the heat matrices of Friedman's H-Statistic that a substantial interaction effect exists between the features. The strength of trend show a less interactivity in yearly series data, reflecting that this feature is more important on its own. The features involved in interaction and their strength of interaction effect differ across the different frequency categories (yearly, quarterly, monthly, weekly, daily and hourly) as well as forecast-models (random walk, ETS models, etc.). However, it is interesting to note that within each frequency category all or a subset of ACF/PACF-based features interact with each other. This confirms that the information regarding the correlation structure of the time series is an essential information for the choice of model selection.

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.

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.
- Silva, N da, D Cook & EK Lee (2017). Interactive graphics for visually diagnosing forest classifiers in R. *arXiv preprint arXiv:1704.02502*.
- 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.
- Wickham, H, D Cook & H Hofmann (2015). Visualizing statistical models: removing the blind-fold. *Statistical Analysis and Data Mining: The ASA Data Science Journal* **8**(4), 203–225.
- Zhao, Q & T Hastie (2017). Causal interpretations of black-box models. *Journal of Business & Economic Statistics*.