# Peeking inside FFORMS: Feature-based FORecast Model Selection

**Abstract**

This study investigates the relationship between features of time series and the choice of forecast-model selection using FFORMS framework. FFORMS framework build a mapping that relates the features of time series to the "best" forecast-model using random forest algorithm. In this paper we make a novel use of FFORMS framework to explore the nature of the role of features to the predictive mechanism. We explore the role of features in two different perspectives: i) Global contribution of features: overall role of features in the choice of different forecast model selection and ii) Local contribution of features: what features are most influential for the prediction of a specific time series?. This is accomplished by using model-agnostic machine learning interpretability approaches. Partial-dependency plots are used to visualize both main and interaction effects of features. The results of this study provide a more refine 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.

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

## 1 Introduction

The time series forecasting field has been evolving for a long time and has introduced a wide variety of forecasting methods. However, for a given time series the selection of an appropriate forecasting-method 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 time series are considered to be an important factor in identifying suitable forecasting models (Meade 2000; Makridakis & Hibon 2000). However, a description of relationship between the features and the performance of algorithms is rarely discussed in the field of forecasting.

There have been several recent studies on use of supervised learning algorithms to automate the forecast model selection based on the features computed from the time series (Shah 1997; Prudêncio & Ludermir 2004; Lemke & Gabrys 2010; Kück, Crone & Freitag 2016). Meta-learning approach provides a systematic guidance on model selection based on knowledge acquire from historical data set, in our case historical collection of time series. The key idea behind these framework 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 labeled according to the "best" forecasting method(i.e. lowest MASE, etc.). Then a meta-learner is trained to identify suitable forecasting models. With the era of big data, such an automated model selection process is necessary because the cost of invoking all possible forecasting method is prohibitive. However, these work suffer from the limitation of providing meaningful interpretations that can enhance understanding of relations between features and model outcomes. To best of our knowledge, very limited efforts have been taken to understand how the models are making its decisions and what is really happening inside these complex model structures. This results in less transparency of the model which lead to the questions of

1. How features are related to the property being modeled?
2. How features interact with each other to identify the suitable forecasting method?
3. Why certain features were responsible in driving certain decisions?
4. Which features contribute the most to classify a specific instance?

On the other hand, aside from the goal of developing automated forecast-model selection framework few researchers have made an attempt to provide a description of relationship between the features and the performance of algorithms (Schnaars 1984; Wang, Smith-Miles & Hyndman 2009; Lemke & Gabrys 2010; Petropoulos et al. 2014). However, these studies are limited by the scale of problem instances used, diversity of forecasting-methods used, quality of features considered, and modelling approached used to identify the relationship between features and forecast model performance. Most of these studies are typically restricted to simple statistical techniques such as simple linear models, decision trees, etc.

To fill this gap, this paper makes a first step towards providing a comprehensive explanation 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 ref, in which we introduced the FFORMS (Feature-based FORecast Model Selection) framework. The random forest algorithm is used to model the relationship between features and "best" performing forecast-model. A large collection of time series is used to train the model. We use 35 features to capture different characteristics of time series. The reason for the choice of random forest algorithm is its ability to model complex variable interactions. One noticeable significance of our approach is this can be parallized to for any given computing budget and time. Although the prediction accuracy of random forest algorithm is high, it is not easy to interpret what is happening inside the forest because of the two-step randomization. In this work we aim at providing a deeper understanding of the underlying mechanism and influence of features in forecast-model selection.

[What is the usefulness of analyzing the relationship between features and model performance?] Understanding the role of features is worthwhile even if producing an accurate and generalizable model is the only objective of the modelling. This is because the less transparency of the model may be distrusted regardless of their predictive performance. The methodology we propose here is a novel application of machine learning interpretability methods to visualize and explore the role of features in forecast-model selection.

This paper proceeds as follows. In Section 2 we describe the application of FFORMS framework to M4competition data. The main contribution of this from our previous work Talagala, Hyndman & Athanasopoulos (2018) is we extend the FFORMS framework to model weekly, daily and hourly series. Section 3 gives background on machine learning interpretability techniques that are used to identify role of features in forecast model selection. In Section 4 we discuss the results. Section 5 concludes.

# 2 FFORMS Application to M4 competition data

The FFORMS framework consists of two main components: i) *offline phase*, which includes the development of a classification model and ii) *online phase*, use the classification model developed in the offline phase to identify "best" forecast-model. We develop separate classifiers for yearly, monthly, quarterly, weekly, daily and hourly series.

## 2.1 FFORMS framework: offline phase

### 2.1.1 observed sample

We split the time series in the M4 competition into training set and test set. The time series in the training set are used as the set of observed time series. The time series in the test set are used to evaluate the classification models. Further, for yearly, quarterly and monthly time series in addition to the time series provided in the M4 competition we used the time series of M1 and M3 competitions. Table 1 summarizes the number of time series in the observed sample and the test set in each frequency category.

### 2.1.2 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

**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 |

standard automatic forecasting algorithms to simulate multiple time series from each series. Table 2 shows the different automatic forecasting algorithms used under each frequency category. The automated ETS and ARIMA are implemented using `ets` and `auto.arima` functions available in the forecast package in R (Hyndman et al. 2018). The `stlf` function in the forecast package (Hyndman et al. 2018) is used to simulate multiple time series based on multiple seasonal decomposition approach. As shown in Table 2 we fit models to each time series in the M4 competition database from the corresponding algorithm and then simulate multiple time series from the selected models. 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 366 the frequency is set to 7 and if the time series is long enough to take more than a year (length > 366), the series is converted to a multiple seasonal time series objects with frequencies 7 and 365.25. For hourly series, if the series length is shorter than 168, frequency is set to 24, if the length of the series is greater than 168 and less than or equals to 8766 only daily and weekly seasonality are allowed setting the frequencies to 24 and 168. 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).

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

| Algorithm | Y | Q | M | W | D | H |
|---|---|---|---|---|---|---|
| automated ETS | ✓ | ✓ | ✓ | | | |
| automated ARIMA | ✓ | ✓ | ✓ | | | |
| forecast based on multiple seasonal decomposition | | | | ✓ | ✓ | ✓ |

As illustrated in Talagala, Hyndman & Athanasopoulos (2018), the observed time series and the simulated time series form the reference to build our classification algorithm. 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.

### 2.1.3 Input: features

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

**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 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).

### 2.1.4 Output: class-labels

In addition to the class labels used by Talagala, Hyndman & Athanasopoulos (2018) we include some more class labels when applying the FFORMS framework to the M4 competition time series. The description of class labels considered under each frequency is shown in Table 4. 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.

**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 | | - | - | - | ✓ |

The `auto.arima` and `ets` functions in the forecast package are used to identify the suitable (S)ARIMA and ETS models. In order to identify the “best” forecast-model for each time series in the reference set we combine the mean Absolute Scaled Error (MASE) and the symmetric Mean Absolute Percentage Error (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 methods. The model with the lowest average value of the scaled MASE and scaled sMAPE is selected as the output class-label. Most of the labels given in Table 4 are self-explanatory labels. In STL-AR, mstlets, and mstlarima, first STL decomposition method applied to the time series and then seasonal naive method is used to forecast the seasonal component. Finally, AR, ETS and ARIMA models are used to forecast seasonally adjusted data respectively.

### 2.1.5 Train a random forest classifier

A random forest with class priors is used to develop the classifier. We build separate random forest classifiers for yearly, quarterly, monthly, weekly, daily and hourly time series. The wrapper function called `build_rf` in the `seer` package enables the training of a random forest and returns class labels(“best” forecast-model) for each time series.

### 2.2 FFORMS framework: online phase

The online phase of the algorithm involves generating point forecasts and 95% prediction intervals for the 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, in this case forecast-models. Finally, all negative values are set to zero.

## 3 Peeking inside FFORMS

The main theme of this paper is to explore the nature of the relationship between features and forecast-model selection. We use both model-diagnostic approaches and machine learning interpretability approaches.

### 3.1 Model-diagnostics

Model-diagnostic is an important aspect in evaluating the accuracy of the model's predictions as well as the model's understanding of the nature of the relationship between features and predicted outcome.

#### 3.1.1 Out-of-bag(OOB) error and uncertainty measure for each observation

It is argued in order to estimate the test error of a bagged model it is not necessary to perform cross-validation approach, because each tree is grown using different bootstrap samples from the training set and a part of training data is not used in the tree construction (Breiman (2001); Chen, Liaw & Breiman (2004)). In general, each bagged tree does not make use of around one third of observations to construct the decision tree. These observations are referred to as the out-of-bag(OOB) observations. Each tree is grown based on different bootstrap samples hence, each tree has different set of OOB observations. These OOB samples can be used to calculate internal estimation of the test set error.

#### 3.1.2 Representation of model in the data space and data in the model space (d-in-ms)

Wickham, Cook & Hofmann (2015) explains 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 visualisation of embedding high-dimensional data into a low-dimensional space generated from the model. Visualisation 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 visualisations 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 visualisation method of this kind and Silva, Cook & Lee (2017) for comparable method for random forest algorithms.

### 3.2 Machine Learning Interpretability

In recent years, there have been a growing interest for interpretability of machine learning algorithms with European General Data Protection Regulation (GDPR) stipulates the explainability of all automatically made decision concerning individuals. We explore the role of features in two different perspectives: i) global explanation of feature contribution: overall role of features in the choice of different forecast model selection, and ii) local explanation of feature contribution: nature of the contributions features make for a prediction of a specific instance. We will introduce each of these ideas briefly below. Model-diagnostics tools are used.

### 3.3 General Notation

Let $\mathcal{P} = \{(\mathbf{x^{(i)}}, y^{(i)})\}_{i=1}^{N}$ be the historical data set we use to train the classifier. Consider a p-dimensional feature vector $X = (X_1, X_2, ..., X_p)$ and a dependent variable, best forecasting method for each series $Y$. Let $\mathcal{G}$ be the unknown relationship between $X$ and $Y$. Zhao & Hastie (n.d.) 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 $g$.

### 3.4 Global Interpretability Methods

Global interpretability evaluate 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 model outcome. For example, which features are contribute mostly to the predictive mechanism of the fitted model, complex interactions between features, etc. In the following subsections we provide a description of tools we use to explore the global perspective of the model.

### 3.5 Analysis of Feature contribution

Jiang & Owen (2002) explains variable importance under three different views: i) causality: change in the value of Y for a 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 (n.d.)). We will introduce each of these variable importance measures below.

### 3.5.1 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).

### 3.5.2 Permutation-based variable importance measure

The permutation-based variable importance introduced by Breiman (2001) measures the 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 $\bar{\mathcal{B}}^{(k)}$ be the OOB sample for a tree $k$, with $k \in \{1, ..., ntree\}$, where $ntree$ is the number of trees in the random forest. Then the variable importance of variable $X_j$ in $k^{th}$ tree is:

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

where $\gamma^k_i$ denotes the predicted class for the $i^{th}$ observation before permuting the values of $X_j$ and $\gamma^k_{i,\pi_j}$ is the predicted class for the $i^{th}$ observation after permuting the values of $X_j$. The overall variable importance score is calculated as:

$$VI(X_j) = \frac{\sum_1^{ntree} VI^{(t)}(x_j)}{ntree}.$$

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).

### 3.5.3 Partial dependence plot (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 feature 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 have treated the forecast-model selection problem as a classification problem. Hence, in this paper partial dependency functions displays the probability of certain class occurring given different values of feature $X_s$.

### 3.5.4 Variable importance measure based on PDP

Greenwell, Boehmke & McCarthy (2018) introduced 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 $\{\bar{g}_s(x_{sk})\}_{k=1}^{m}$.

### 3.5.5 Individual Conditional Expectation (ICE) curves

While partial dependency curves are useful in understanding the estimated relationship between feature and predicted outcome in the presence of substantial interaction between features, it can be misleading. Goldstein et al. (2015) proposed 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.

#### 3.5.6 Variable importance measure based on ICE curves

This method is similar to the PDP-based VI scores above, but are based on measuring the "flatness" of the individual conditional expectation curves. We calculated standard deviations of each ICE curve. We then computed a 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.

### 3.6 Assessment of Interaction Effect

Friedman's H-statistic (Friedman, Popescu, et al. (2008)) is use to test the presence of interaction between all possible pair 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 effect.

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 are measured based on the subset of randomly selected training examples.

### 3.7 Local Interpretable Model-agnostic Explanations (LIME)

Global interpretations help us to understand entire modeled 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 introduce 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 in the local context and vice verse. 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, ridge regression, etc.
6. Fit a simple linear model to the permuted data based on $m$ selected features and 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”.

# 4 Results

## 4.1 Yearly data

### Model diagnostic: FFORMS framework,Yearly series

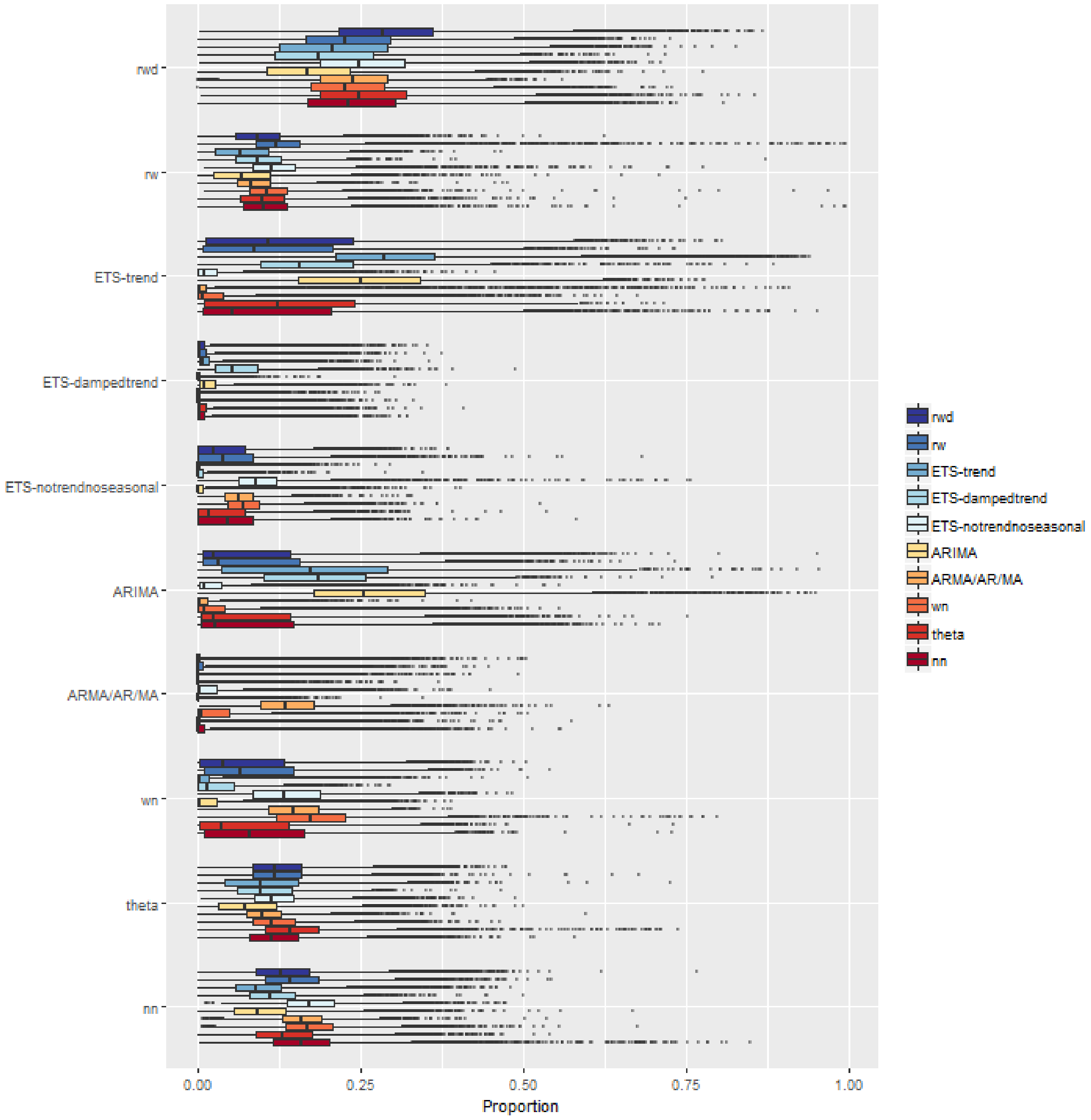

**Figure 1:** *Distribution of proportion of times each yearly time series was assigned to each class in the forest. Each row represent the predicted class label and colours of boxplots corresponds to true class label. There are ten rows in the plot corresponds to each predicted class represented by Y-axis. X-axis denotes the proportion of times a time series is classified in each class. On each row, distribution of correctly classified class dominated the top, indicating a fairly good classification of the model fitted.*

Figure 1 shows the distribution of proportion of times each observation (in our case each time series) was assigned to each class based on OOB sample. Each row represent the **predicted class label** and colour of boxplots corresponds to **true class label**. The proportion 1 indicates, that the time series was always predicted to the corresponding class and 0 being never. This is an alternative way of visualizing the vote-matrix information in the random forest model. The other way of representing vote matrix involves ternary plot (Sutherland et al. (2000)) and jittered side-by-side dotplot (Ehrlinger 2015; Silva, Cook & Lee 2017). To over come the problem of overlapping classes arises due to the scale of the training data set, number of classes and similar types of classes of data, boxplot diagrams are used. On each row of Figure 1 the distribution of proportions corresponds to the time series in which the predicted class label and true class label are the same dominates the top indicating a fairly good classification of the model. In addition to that, in each row the distributions corresponds to the time series labeled similar to the predicted class label also dominate the others. For example, within ETS-trend predicted class, the distributions correspond to the true class labels, ETS-damped trend, ARIMA, were also assigned with high probability and less values were assigned to ARMA/AR/MA, White noise process and ETS (ANN)/ ETS(MNN). This confirms that our FFORMS framework successfully learnt the similarities and dissimilarities between the classes itself. Irrespective of the class-labels yearly series have a high chance of selecting random walk with drift models. One reason for this is as shown in Kang, Hyndman, Li, et al. (2018) yearly series of M1, M3 and M4 are generally trended. (Results of M3 and M4 competition) This diagram helps to evaluate the model performance in the data space (model-in-the-data-space) (Silva, Cook & Lee (2017)).

**Feature importance and main effects**

Permutation-based variable importance and Gini feature importance measure are used to evaluate the overall feature importance. Moreover, most important features for each class is identified based on three measures: i) permutation-based variable importance, ii) partial dependence functions based variable importance and iii) ICE-curves based variable importance measure. The one that shows the highest importance is ranked 25, the second best is ranked 24, and so on. Finally, for each category, an average rank for each feature is computed based on the mean value of all rankings across all the feature importance metrics considered. The corresponding results are shown in Figure 2. The features, strength of trend and test statistic of Phillips–Perron(PP) unit root test, linearity, first autocorrelation coefficient of the differenced series are appear to be most important features in each class. On the other hand, sum of squares of first five autocorrelation coefficients of the twice-difference series and lumpiness show lowest contribution across many classes. The length of time series (N) is assigned a high importance in neural-network class compared to others. This could be due to neural network approach may be beneficial in forecasting time series with long history of observations. Further, first correlation coefficient of the twice-differenced series is appear to be most important in ARIMA class as this category contains the higher order differenced series. Hurst exponent and entropy appear to be equally important in stationary classes. Within ETS-damped trend category beta and curvature ranked as important features.

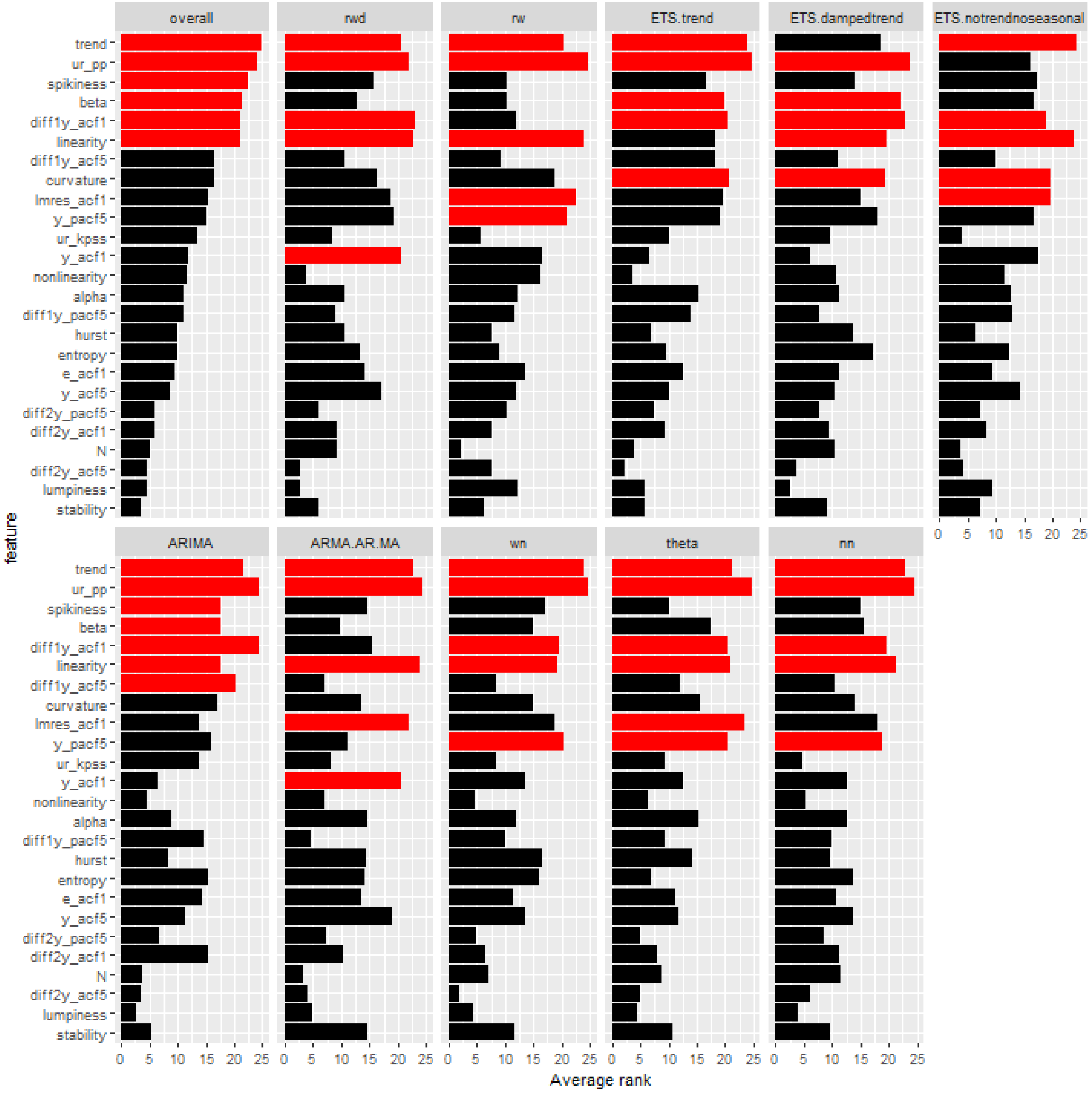

**Figure 2:** *Feature importance plot for yearly series. Longer bars indicate more important features. Top 5 features are highlighted in red.*

### Partial dependency curves

Partial dependency, and associated confidence intervals corresponds to top-three features of each category are plotted to explore the relationship between features and predicted outcome. Confidence intervals also facilitate the indication of interaction effects. Top three features of each category shows a non-linear relationship with predicted outcome. ETS (AAN, MAN, ANN, MNN), ARIMA, ARMA/AR/MA, white noise process and neural network classes show a monotonically increasing or decreasing relationship with trend while theta class has a parabolic relationship with trend. First correlation of the residual series of linear model shows a monotonically increasing relationship with ETS(ANN, MNN) class, whereas the random walk with drift shows an opposite relationship.

### Two-way interaction between features

The relative strength of two-way interaction between features were determined using formulae developed by Friedman, Popescu, et al. (2008), which is implemented in the iml (Molnar, Casalicchio & Bischl (2018),) package in R. Figure 4 shows the heat maps of relative strength of all possible pairwise interactions for each class. The test statistic of Phillip-Perron test, strength of trend, and linearity show a weak interaction with other features in all the class. However, except ETS-trend class, trend and ur_pp shows an high level of interactivity. These two features are appear to be among top 5 in all the classes according to the variable importance measures. Further, narrow confidence bands corresponds to these features in the partial dependency plots also confirms the less interactivity. In almost all the cases partial correlation and auto-correlation based features are heavily interacting. However, the first correlation coefficient of the difference series do not interact with other features heavily in the case of ARIMA class. Further, almost all pair of features appear to be interacting with in neural network category. Figure 5 and Figure 6 communicate more information about the nature of two-way feature interaction effect.

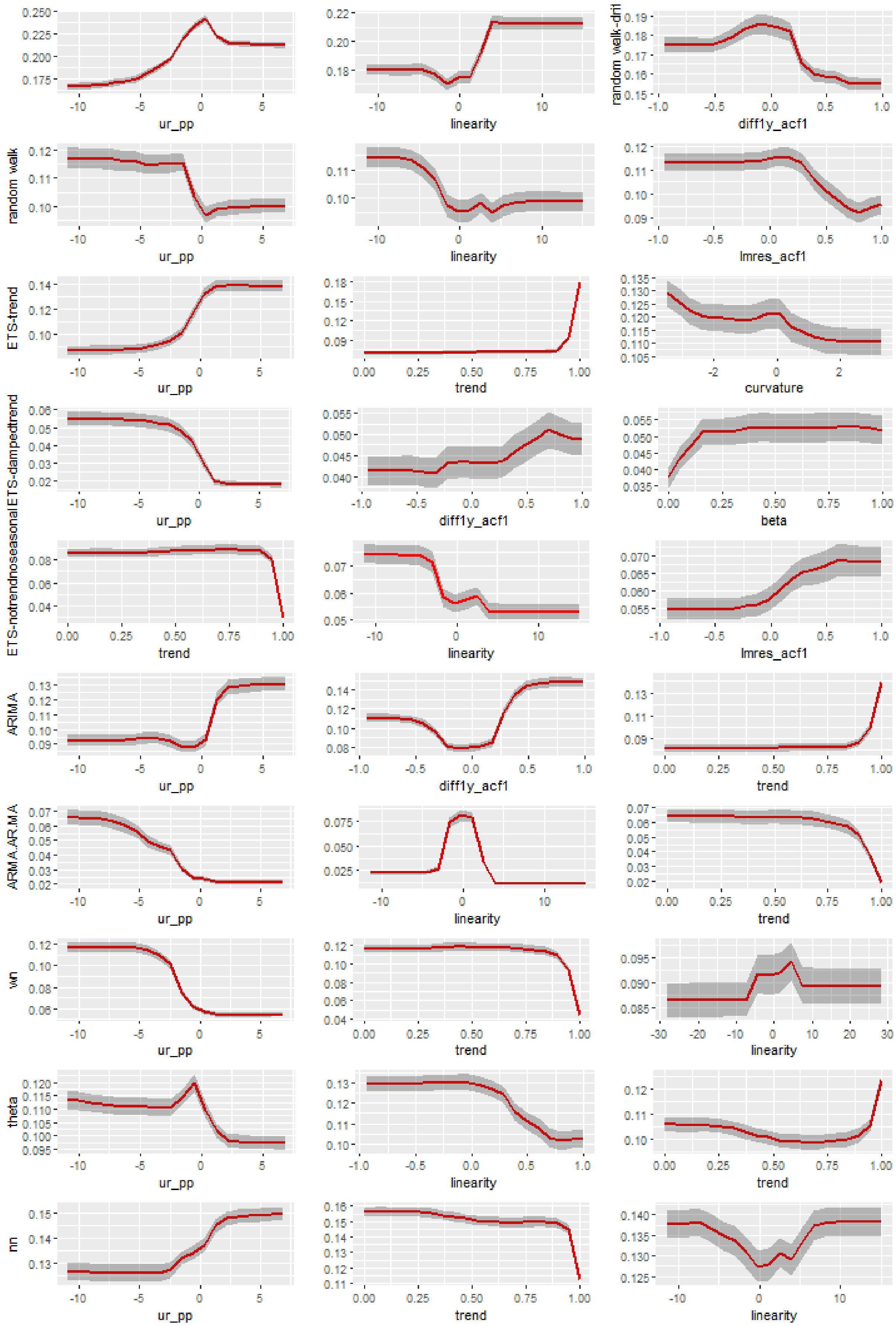

**Figure 3:** *Partial dependence plots for the top ranked features from variable importance measures. The shading shows the 95% confidence intervals. Y-axis denotes the probability of belong to corresponding class.*

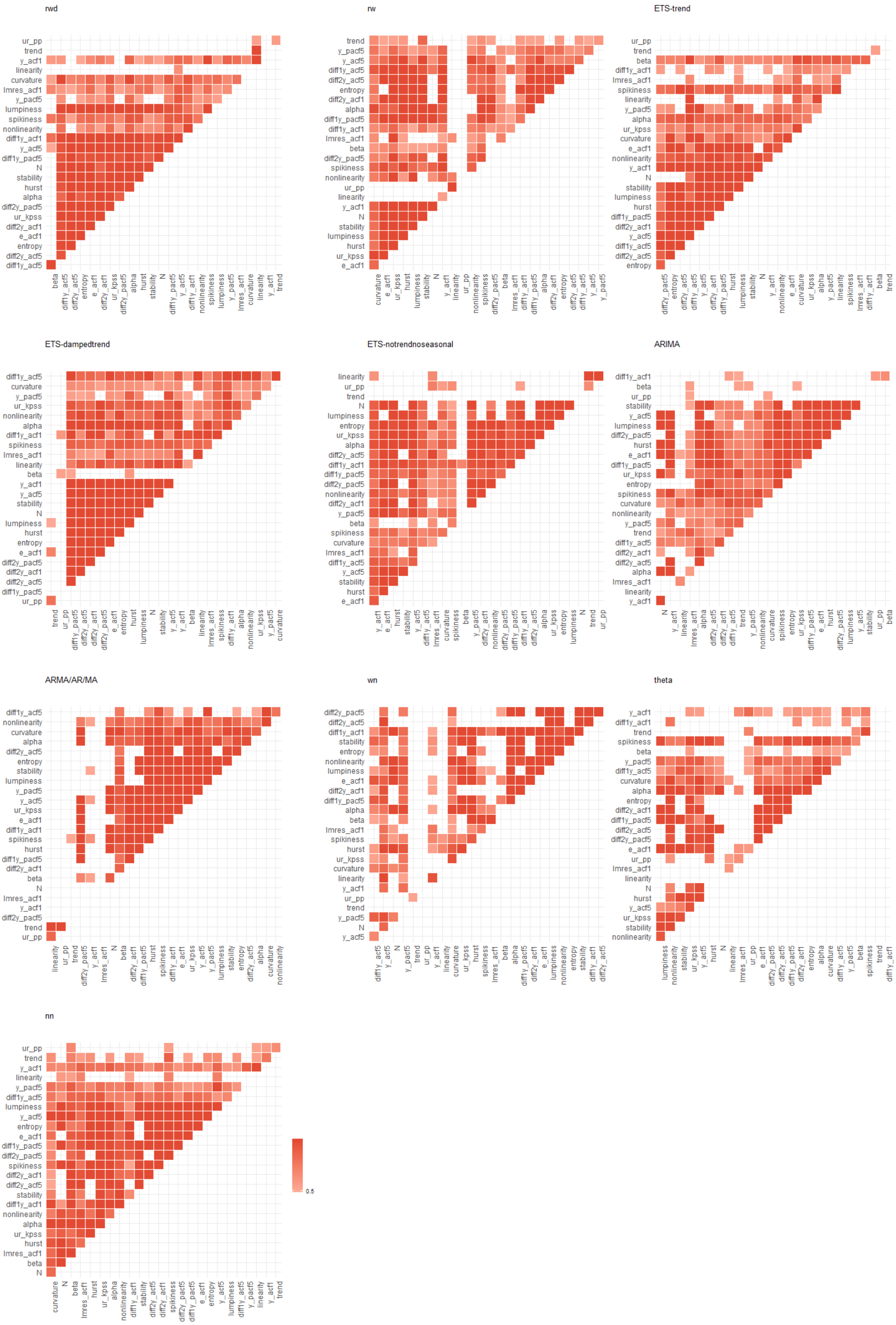

**Figure 4:** *Heat maps of relative strength of all possible pairwise interactions calclated based on Friedman's H-statistic*

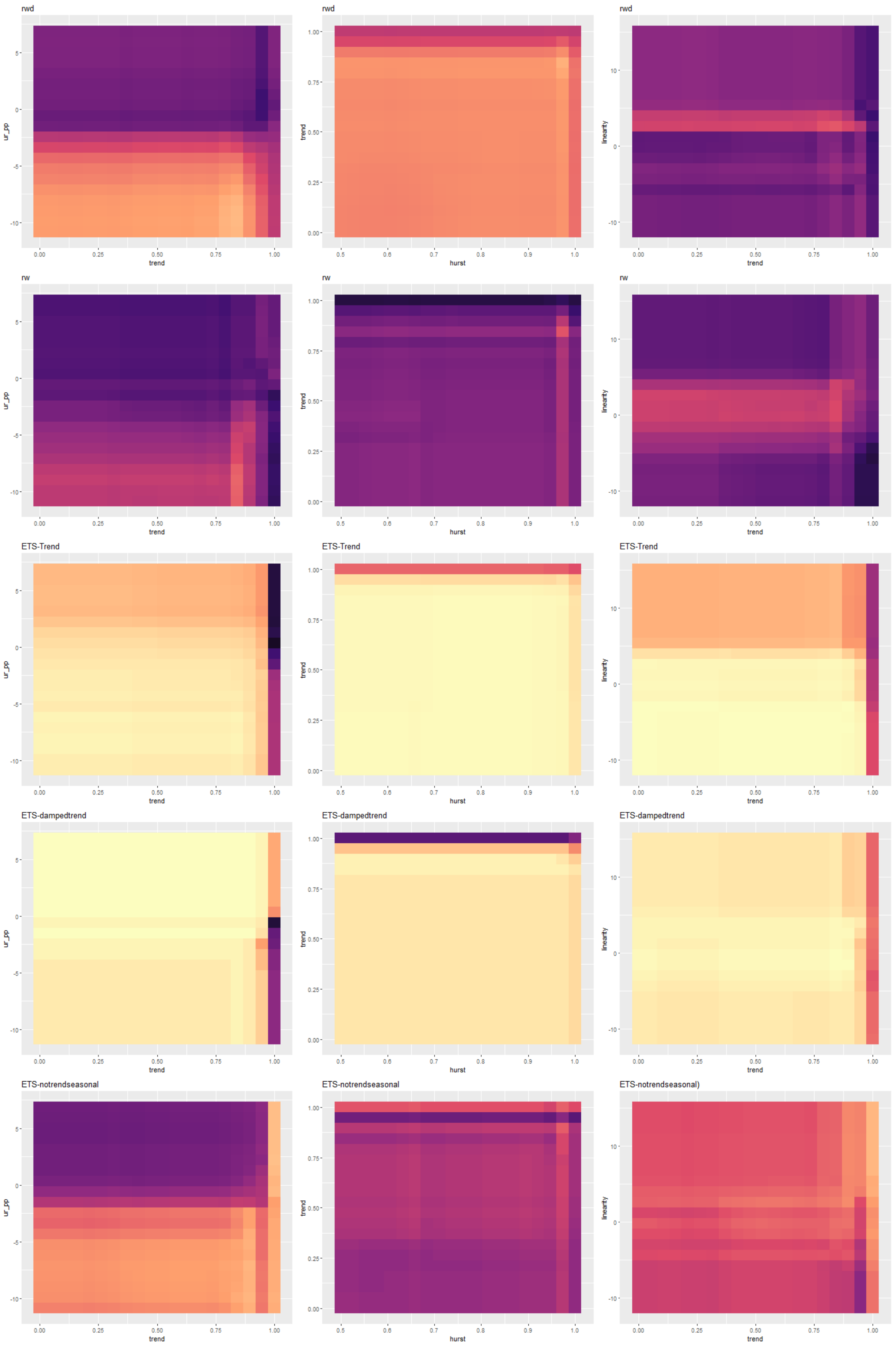

**Figure 5:** *Visualisation of two-way interactions using partial-dependence plots for yearly data.*

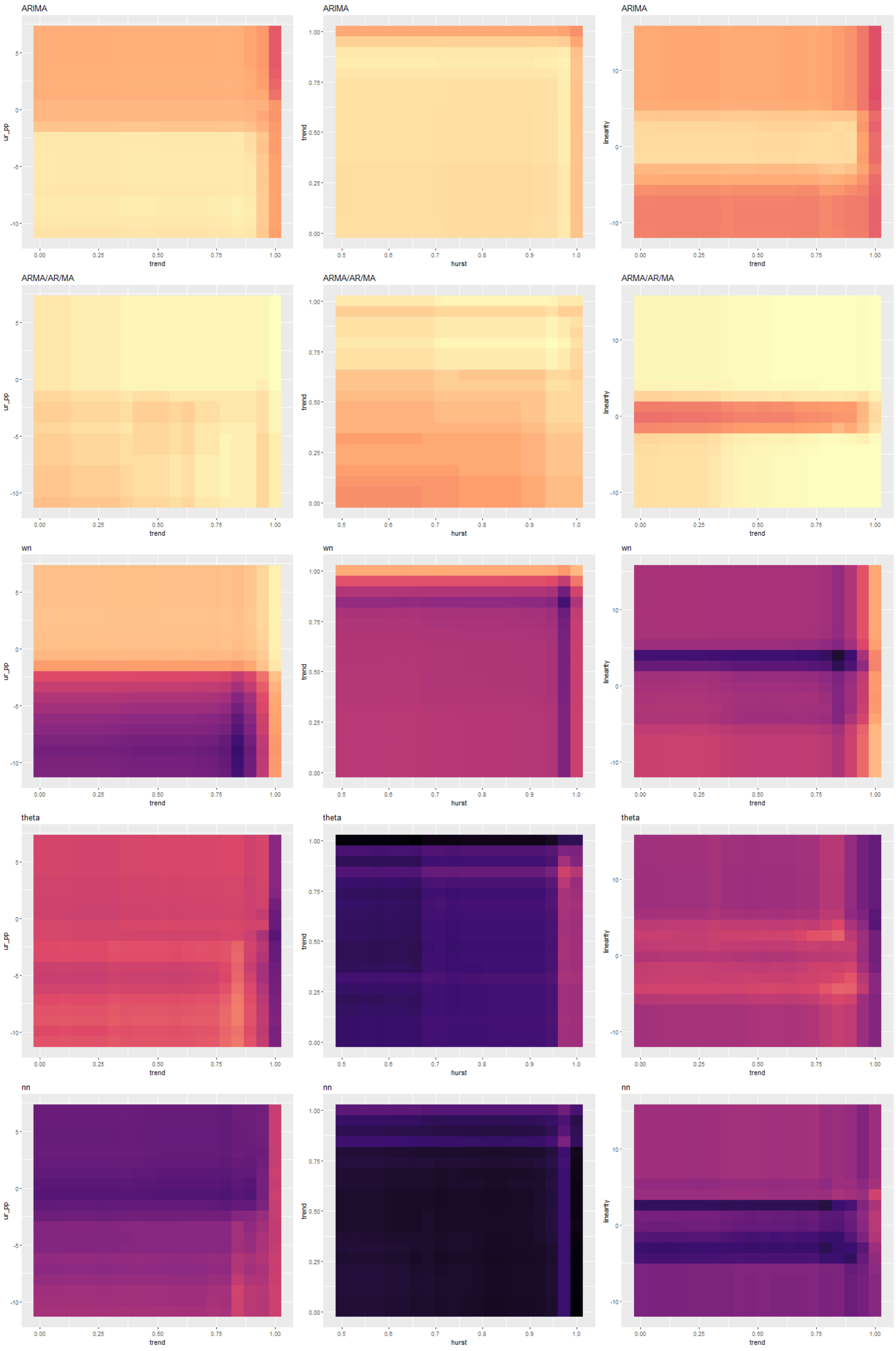

**Figure 6:** *Visualisation of two-way interactions using partial-dependence plots for yearly data.*

### Local Interpretable Model-agnostic Explanations: yearly series

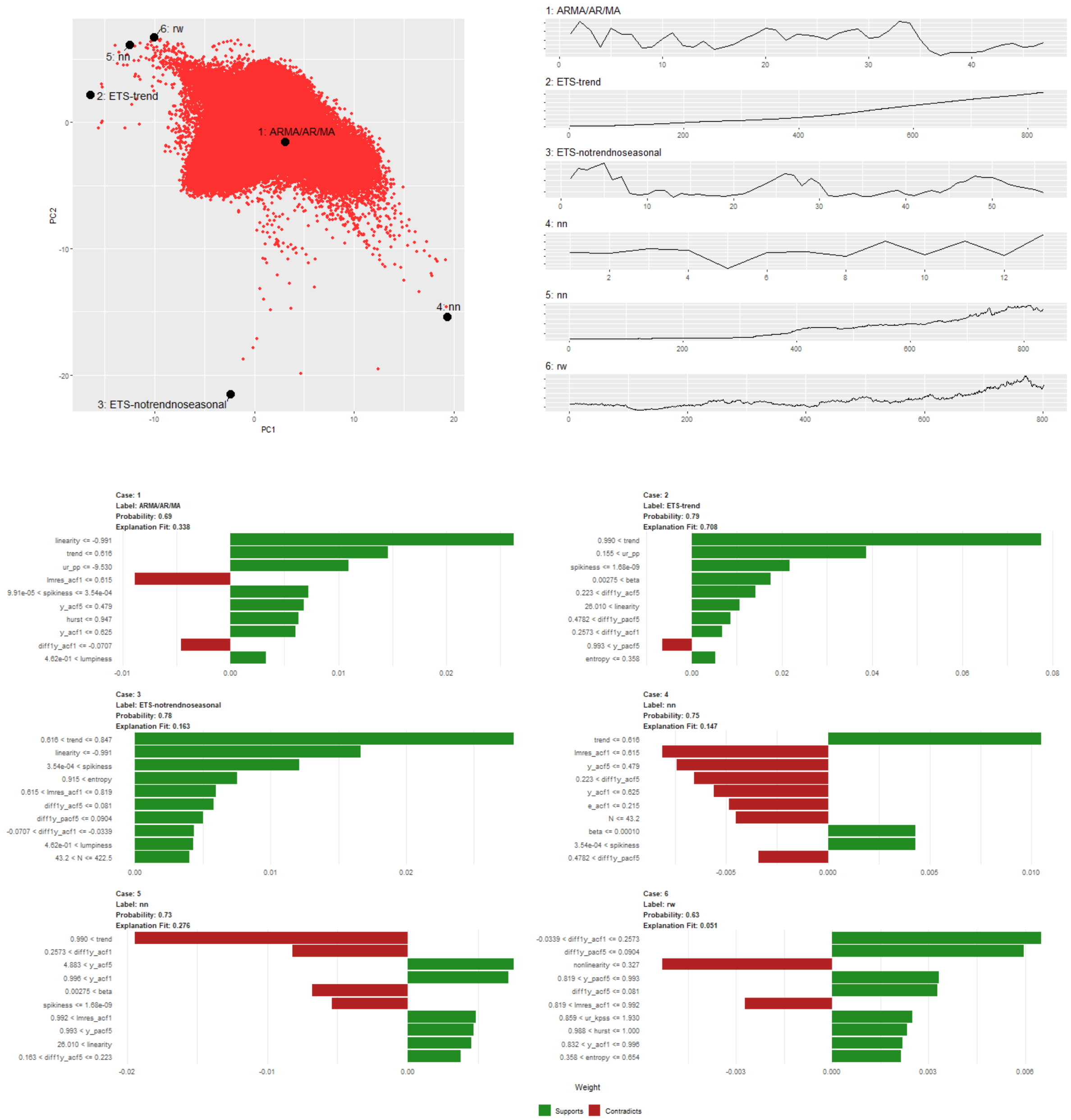

**Figure 7:** *Local interpretable Model-agnostic explanations for six selected yearly time series. Features denoted with green colour are supporting features for an outcome label and length of the bar is proportional to the weight of a feature.*

Figure 7 shows the feature contribution for the instances highlighted in **??**. The low linearity, with a trend value of less than 0.616, low values for Phillip-Perron test-statistic, Hurst exponent, autocorrelation coefficients of the time series and spikiness of the series causes the FFORMS framework to classify the first series with ARMA/AR/MA. On the other hand, LIME indicated the low value of first autocorrelation coefficients of the original series and the residual series of linear regression model contradicts to this decision. The reason might be these properties of autocorrelation coefficients native to the random walk models not to the stationary ARMA/AR/MA models. It is interesting to note that even though 2, 5 and 6 series located close proximity to each other in the PCA space their, varying degree of trend, spikiness, first autocorrelation coefficient of the difference series, entropy, led them to classify as ETS-trend, neural-network and random walk respectively.

## 4.2 Quarterly data

Figure 8 shows proportion of times each series was classified to each class by true class labels. For all ETS models the proportion of times the true class label was identified was higher than other categories. Within snaive, stlar, tbats and white noise classes ETS-seasonal, SARIMA, ETS-trend and ARMA/AR/MA slightly dominates respectively. It could be due to the high-similarity to one another. Further, withing the neural network category ETS (ANN/MNN), ARMA/AR/MA and white noise processes dominates equally. This indicates neural network models share similar type of features with those three classes. Furthermore, except the time series labeled as a ETS models with seasonal component or SARIMA models all other time series have a high probability of being in random walk with drift class irrespective of their true class labels. These results are consisted with the random walk model fitted to yearly data. Except the time series labeled as ARMA/AR/MA all other quarterly time series have a very low chance of being to ARMA/AR/MA class. Further, all distributions corresponds to the tbats row located further away from zero. This indicates all time series select tbats model at least once from the individual trees in the forest. Except few outliars, distributions within neural network category also show a slight upward deviation from zero. However, the upper boundary of these distributions do not surpass the upper boundaries of dominating box plots in the random walk with drift class and SARIMA class. These types of diversity in the distributions indicates the the appropriateness of using combination forecasting. Further, these information are useful in identifying potential time series models for combination forecast and improve the existing combination approaches proposed in the M4-competition.

**Variable importance: quarterly data**

Strength of seasonality appear to be the most important feature across all categories. Similar to the results of yearly time series trend and linearity also listed among the top five features in each category. In the case of yearly data low variable importance is assigned to both stability and length of the series, however with quarterly time series within most categories stability and length of the series are assigned a high importance. In addition to the strength of seasonality, models with seasonal components assigned high importance to additional features related to seasonality such as ACF or PACF feature related to seasonal lag or seasonally-differenced series, for example, snaive and ETS-seasonal with partial autocorrelation coefficient at first seasonal lag, etc. Furthermore, as expected features related to parameter estimates ETS (A, A, A) were selected as important features by ETS with damped trend and seasonal component and ETS with trend and seasonal component models.

**Partial dependency curves: quarterly data**

Figure 10 shows the partial dependency functions for top three features with in each category. Seasonality shows a linear relationship with probability of being to snaive class. Random walk, random walk with drift, all ETS models without seasonal component, ARIMA and white noise process show a similar pattern of relationship with seasonality with vary degree of width of confidence band while all ETS models with seasonal components and SARIMA hold the opposite patter. SARIMA and ARIMA classes shows a non-monotonic relationship with seasonality. Neural network class shows a non-linear relationship with seasonality, and wide confidence bands indicated seasonality interact with other features when deciding probability of being into neural network class. According to the partial dependency functions corresponds to the rows ETS-NTNS, ETS-DT, ETS-T and ETS-TS confirms how well the model's understanding of feature behaviour matches the domain's expert knowledge which gives rise to the trustability of the FFORMS framework. For example, highly trended and low seasonality time series have a high chance of being classified to ETS-Trend class while the opposite relationship can be seen in ETS-seasonal category with low trended and highly seasonally oscillated time series have a high chance of being classified to ETS-seasonal. Even though, two different portfolio of time series are used to train two different classifiers for yearly and quarterly data partial dependency functions show a consistency between the results of yearly framework and quarterly framework. For example, partial dependency function of linearity within ARMA/AR/MA class, partial dependency functions corresponds to trend, etc. Figure 12 shows heat maps of relative strength of all possible pairwise interactions calculated based on Friedman's H-statistic. Within all classes strength of seasonality shows less interactivity with other features. Except ETS-damped trend, SARIMA and ARIMA classes trend interact highly with other features. Further, Friedman's H statistic reflect high interactivity between autocorrelation and partial autocorrelation in SARIMA, ARIMA and ARMA/AR/MA classes. The features alpha and curvature shows very high interactivity with other features within ETS-damped trend, ETS-trend and neural network classes. Length of the time series show very low interactivity with others within snaive class.

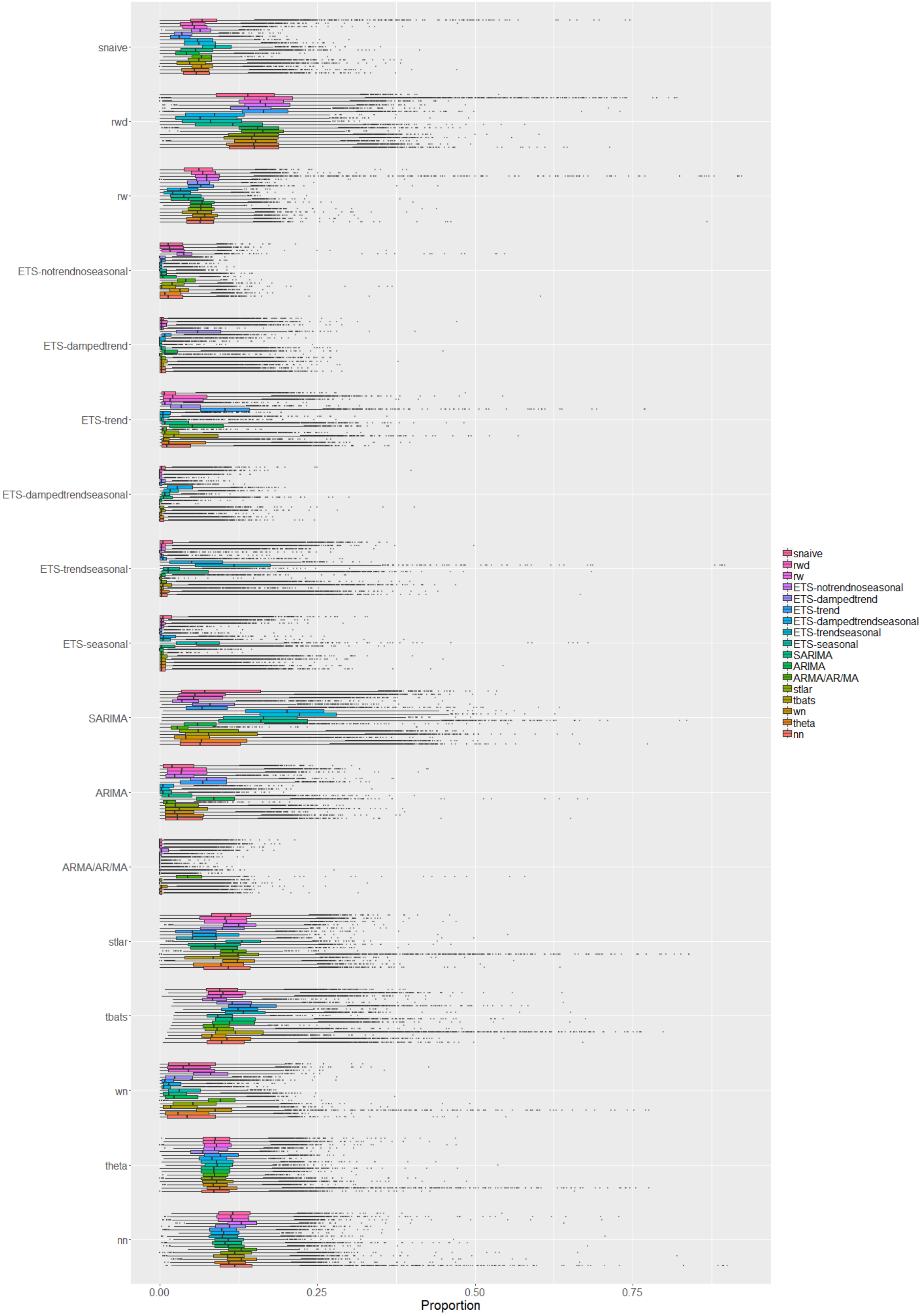

**Figure 8:** *Distribution of proportion of times each quarterly time series was assigned to each class in the forest. Each row represent the predicted class label and colour of boxplots corresponds to true class label. X-axis denotes the proportion of times a time series is classified in each class.*

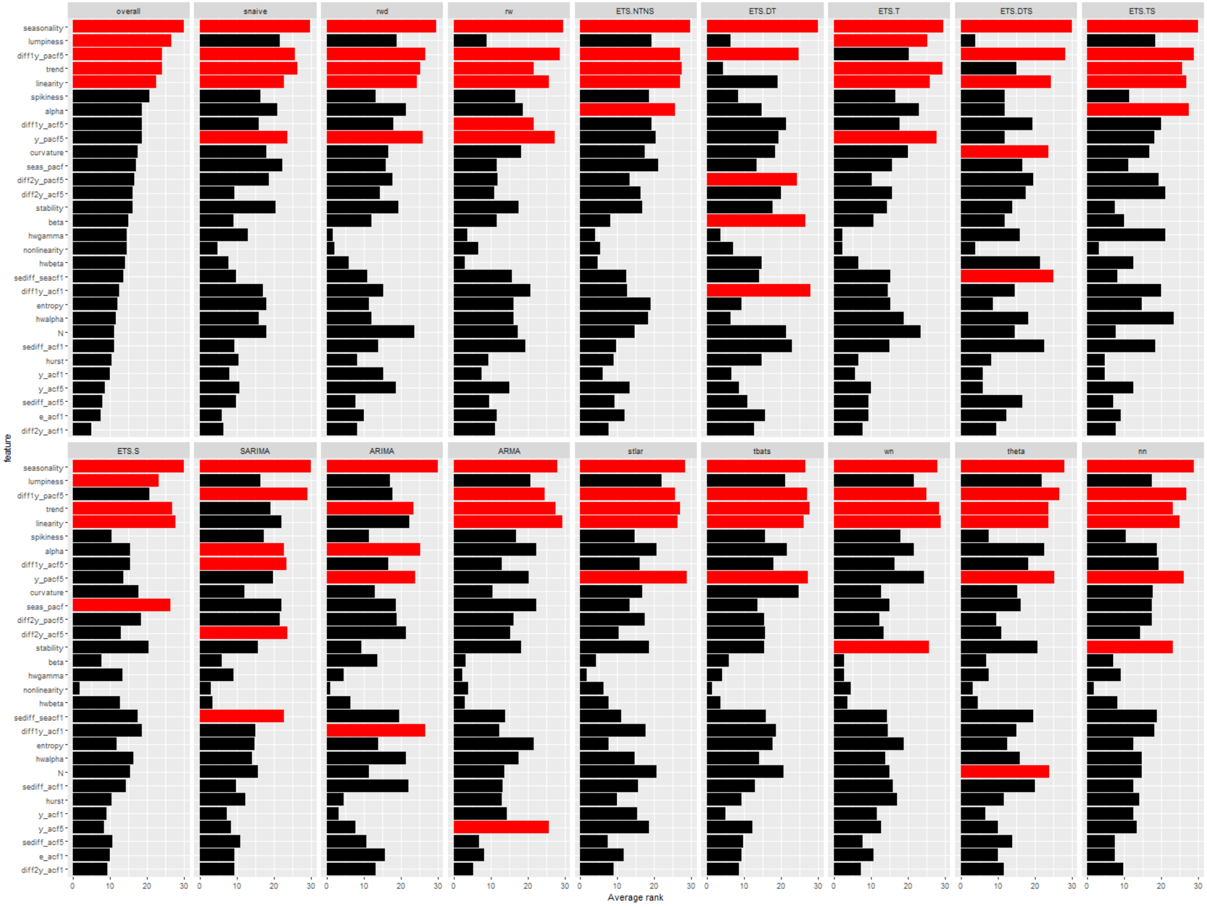

**Figure 9:** *Feature importance plot for quarterly series. Longer bars indicate more important features. Top 5 features are highlighted in red.*

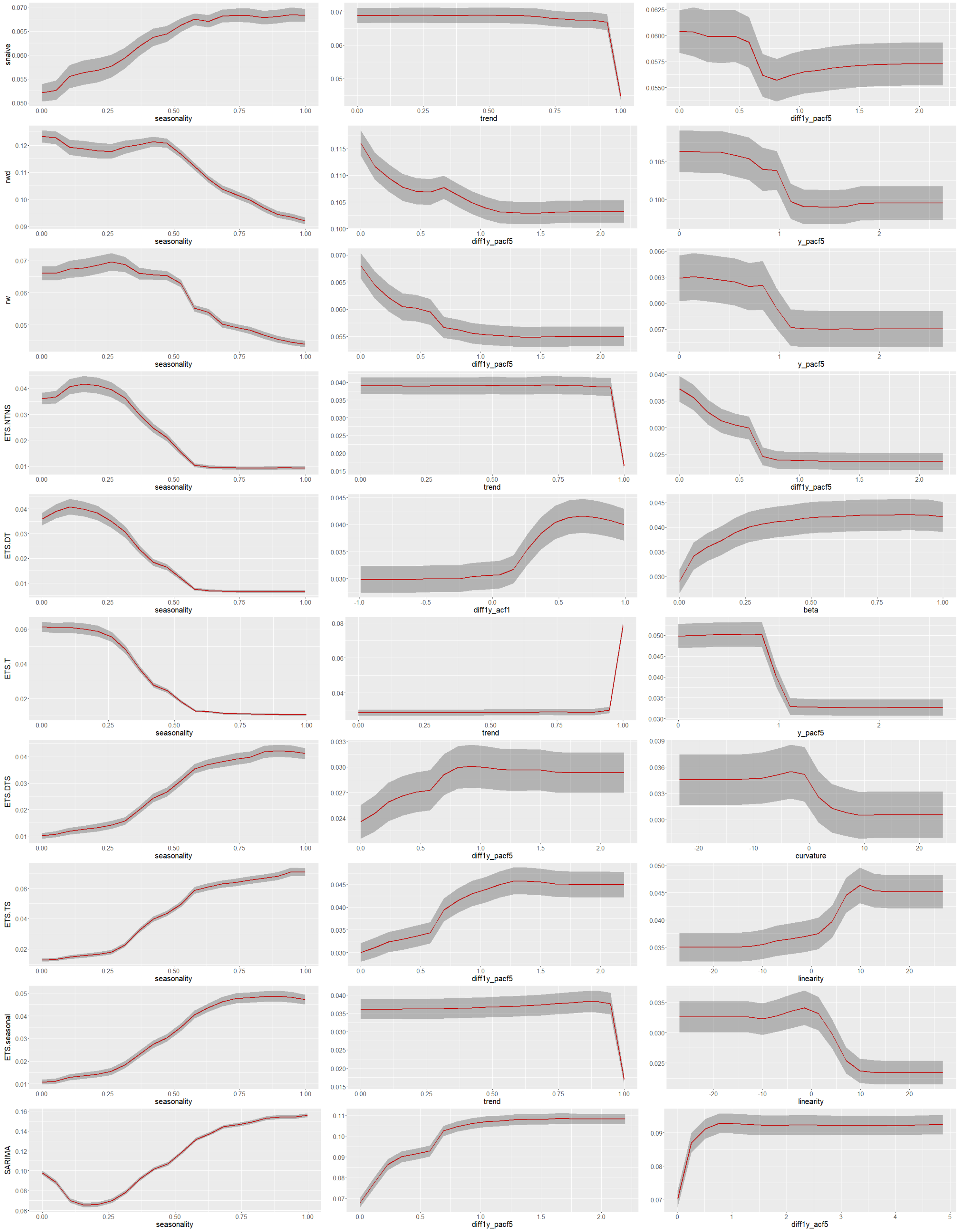

**Figure 10:** *Partial dependence plots for top three features in each category (Quarterly)*

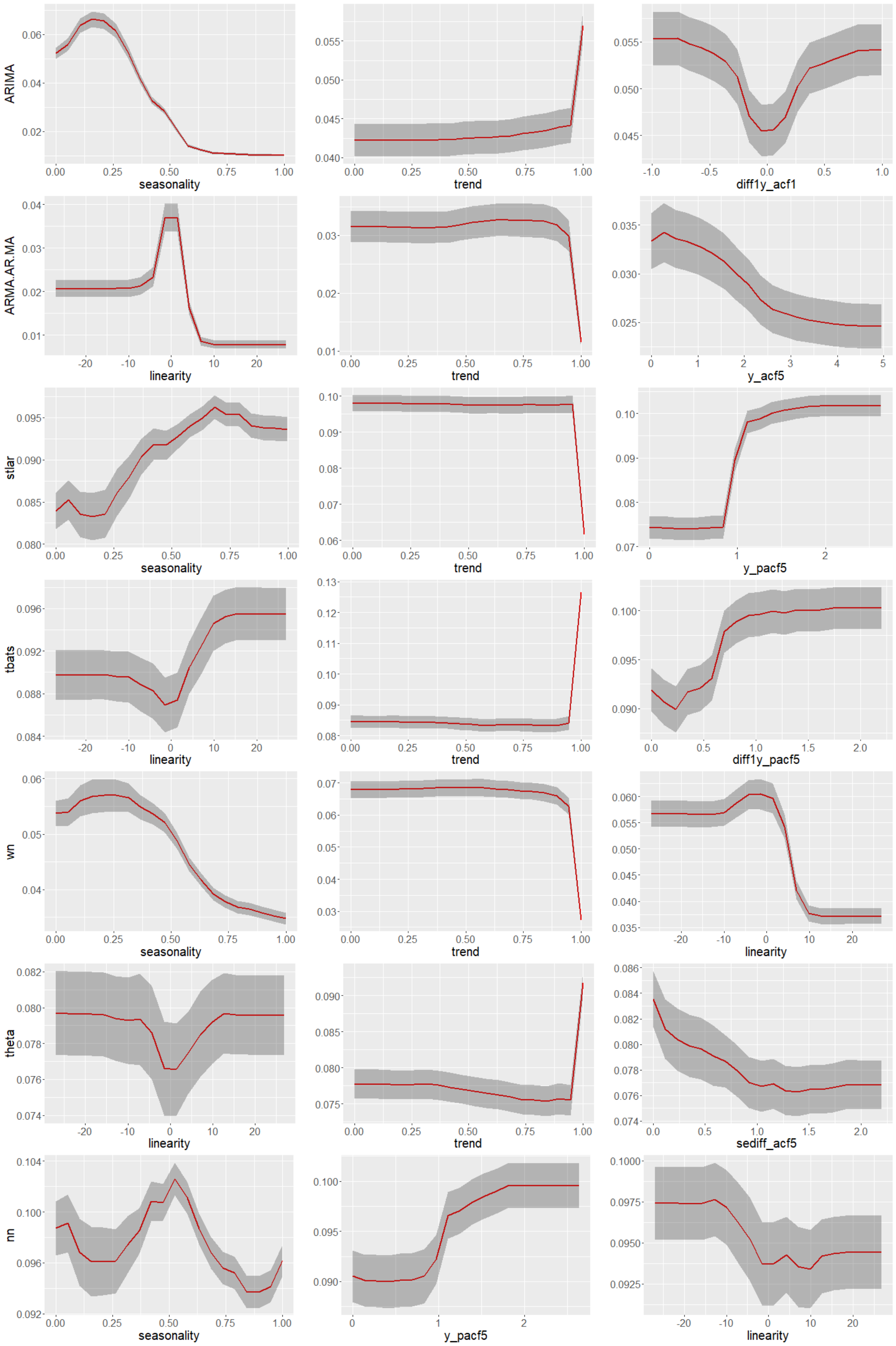

**Figure 11:** *Partial dependence plots for top three features in each category (Quarterly)*

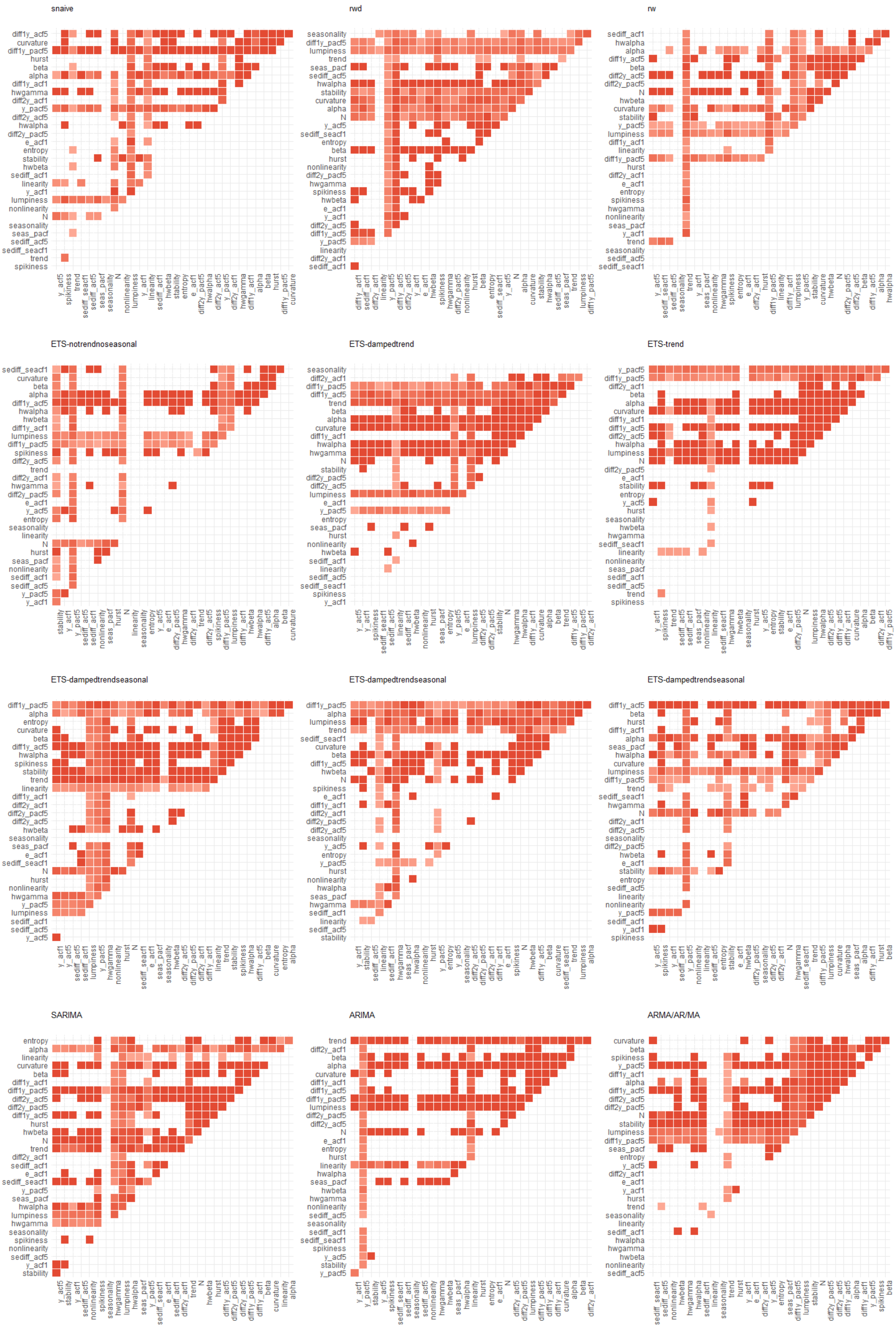

**Figure 12:** *Heat maps of relative strength of all possible pairwise interactions calclated based on Friedman's H-statistic for quarterly data.*

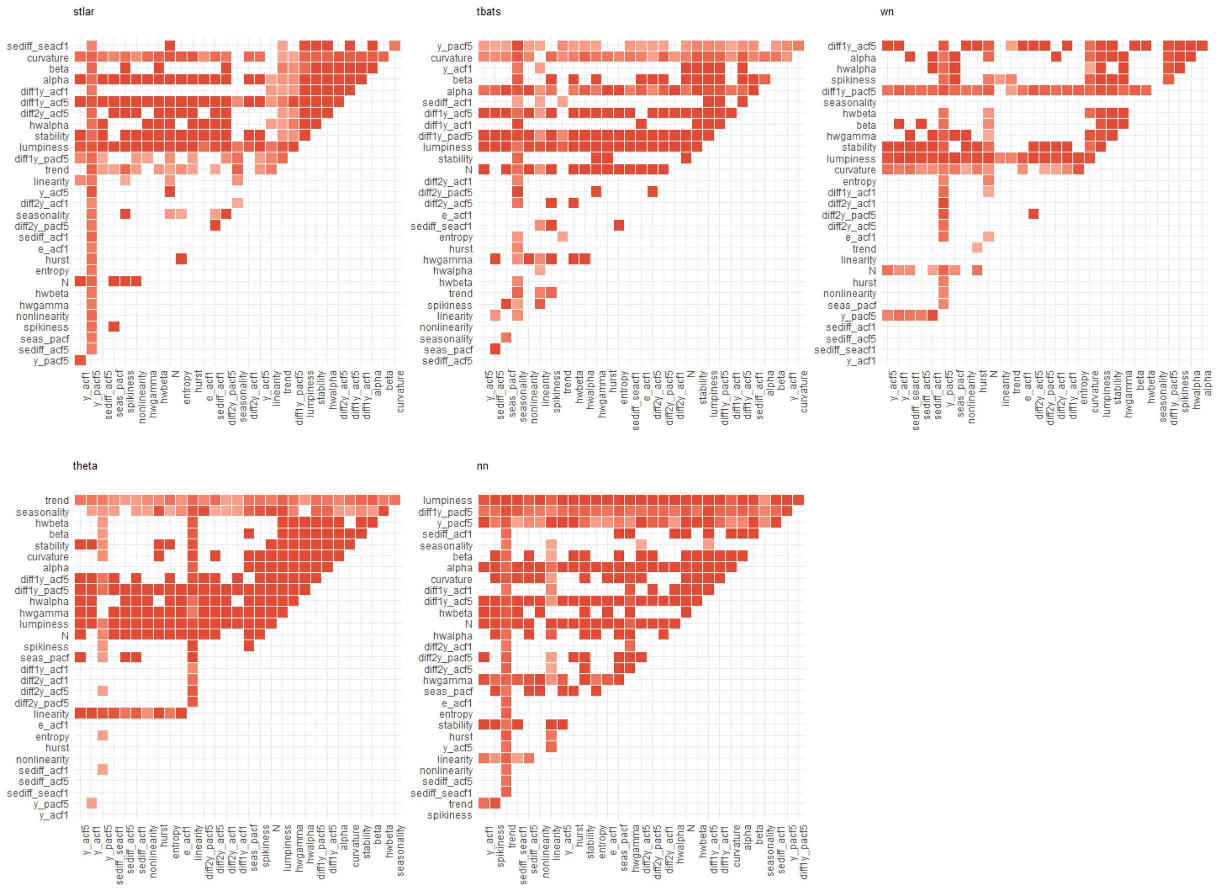

**Figure 13:** *Heat maps of relative strength of all possible pairwise interactions calclated based on Friedman's H-statistic for quarterly data(cont).*

### Local Interpretable Model-agnostic Explanations: quarterly series

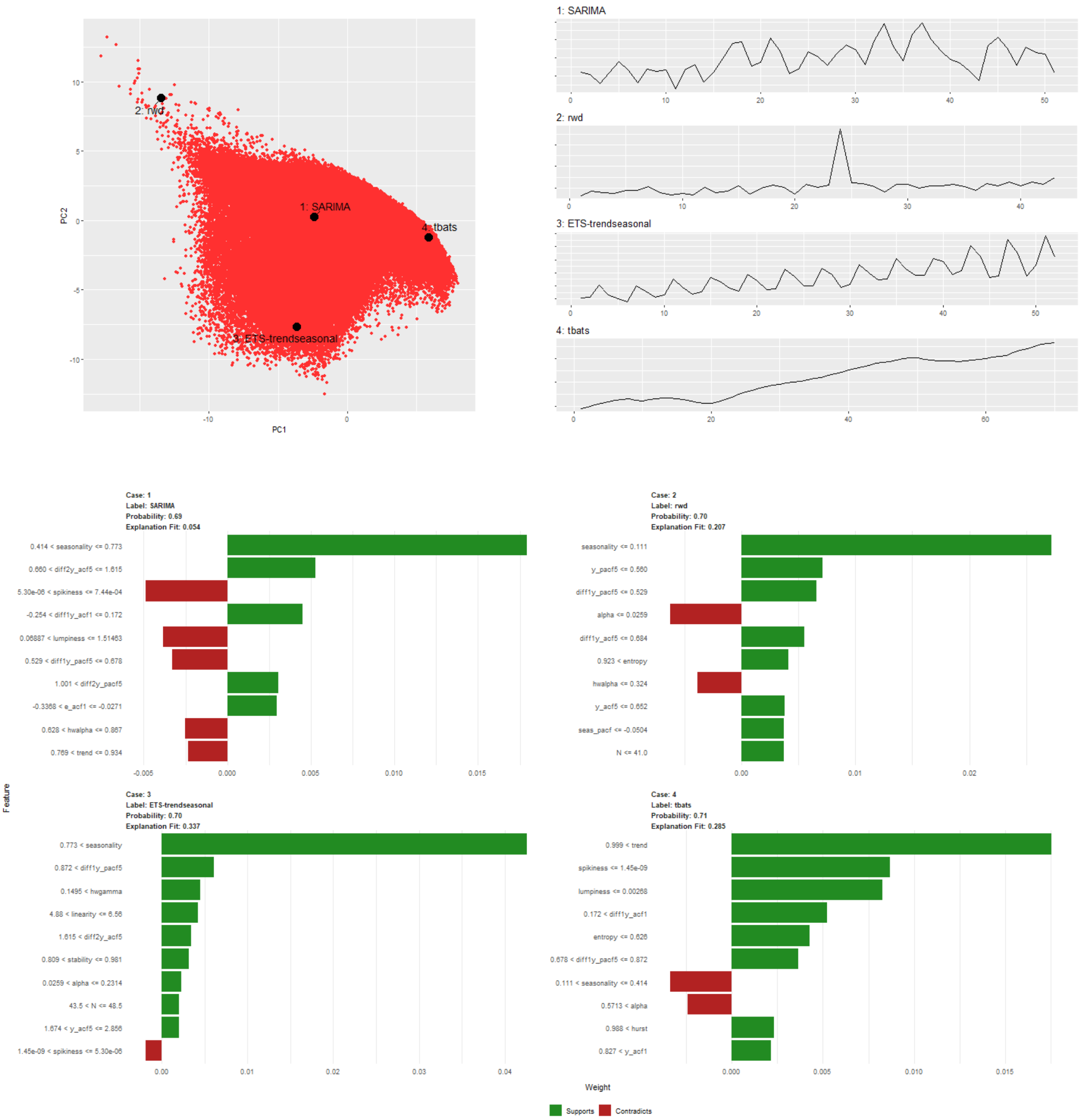

**Figure 14:** *Local interpretable Model-agnostic explanations for four selected quarterly time series. Features denoted with green colour are supporting features for an outcome label and length of the bar is proportional to the weight of a feature.*

## 4.3 Monthly

Figure 15 can be interpreted similar to the results of monthly data. For quarterly and monthly data same set of features and class-labels are used to training the model. Hence, this consistency between the results of quarterly and monthly series would provide evidence in support of the validity and trustability of the model. Seasonality, trend, spikiness and linearity appear to be most important features across all categories. Further, features calculated based on parameter estimates of ETS(A,A,A), ACF and PACF-based features related to seasonal lags and seasonally differenced series were assigned higher importance than the results of quarterly series. One notable difference between quarterly series and monthly series is, length of the series was assigned a very high importance specially in random walk with drift, random walk, ETS with out seasonal and trend components, ETS with seasonal components, SARIMA and ARMA/AR/MA classes. Partial dependence plots of the top 3 features of each class is represented in Figure 17 and Figure 18. Probability of selecting a model with a parameter to capture the seasonal effect (snaive, all ETS models with seasonal component, SARIMA, tbats, theta, stlar) increases as the seasonality increases. Probability of selecting a neural network approach decreases as seasonality increases but this probability increases as stability increases. PPD plots also reveals FFORMS framework select ETS-trend models for short, less seasonally and highly trended (strength of trend close to 1) time series while random walk with drift and random walk series are suitable for short, less seasonally series with trend vary between 0.75 to 0.85. According to Figure 19 and Figure 20 within random walk and ARMA/AR/MA classes trend, seasonality, stability and length of the time series show less interactivity with other features. Trend shows high interactivity with other features within random walk, all classes with ETS model with damped trend component and theta . In general, spikiness, also shows interactivity with other features within many classes.

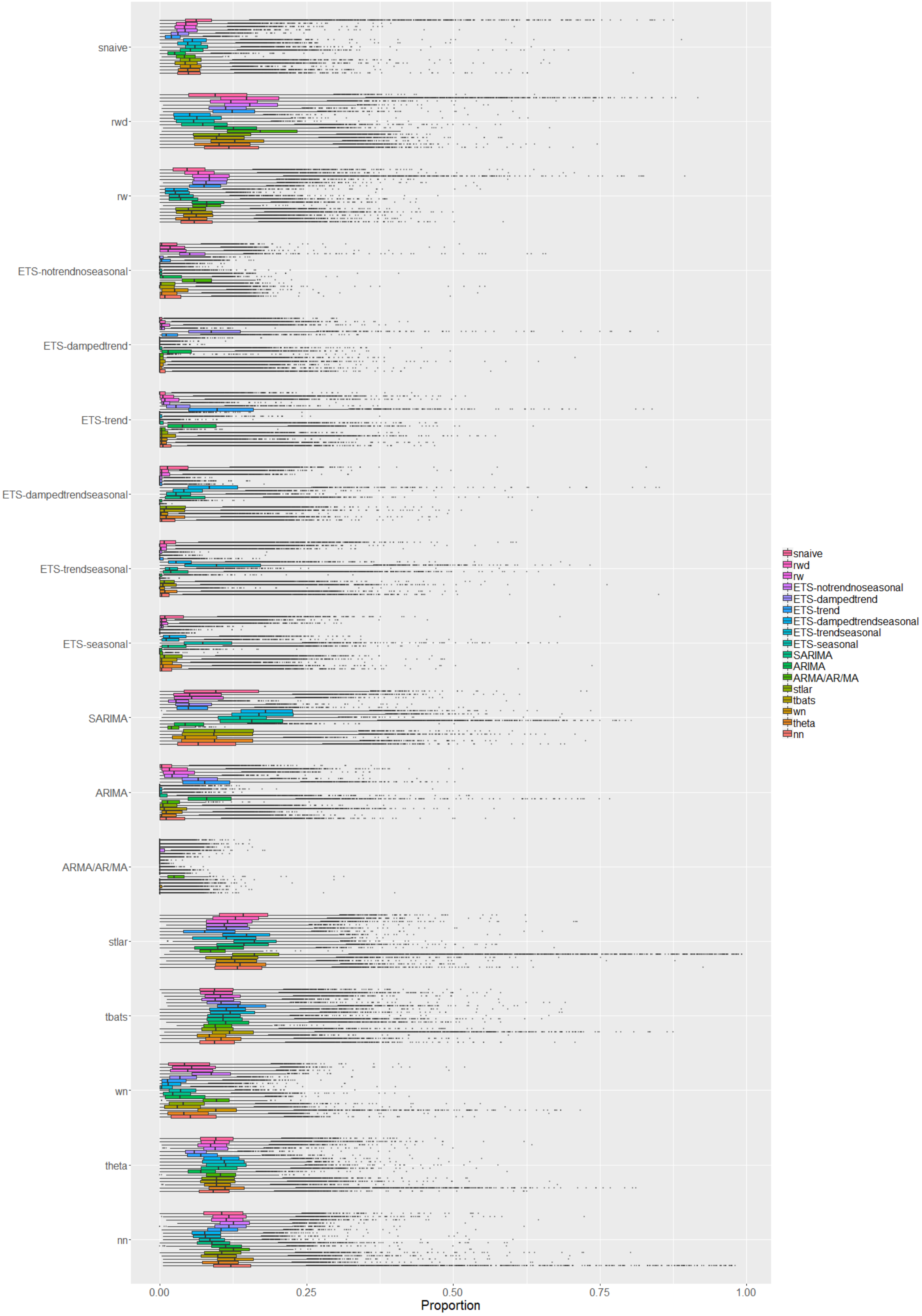

**Figure 15:** *Distribution of proportion of times each monthly time series was assigned to each class in the forest. Each row represent the predicted class label and colour of boxplots corresponds to true class label. X-axis denotes the proportion of times a time series is classified in each class.*

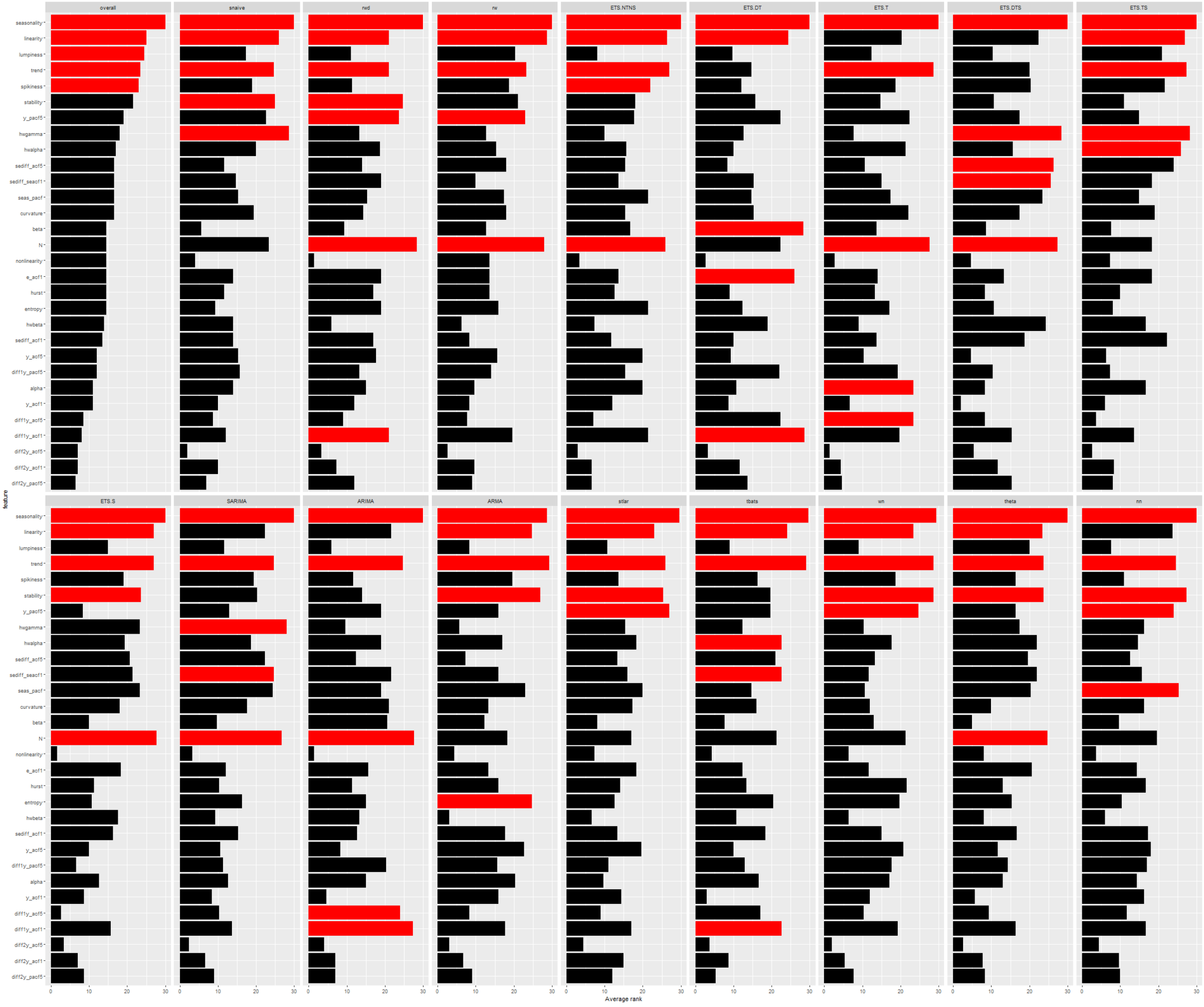

**Figure 16:** *Feature importance plot for monthly series. Longer bars indicate more important features. Top 5 features are highlighted in red.*

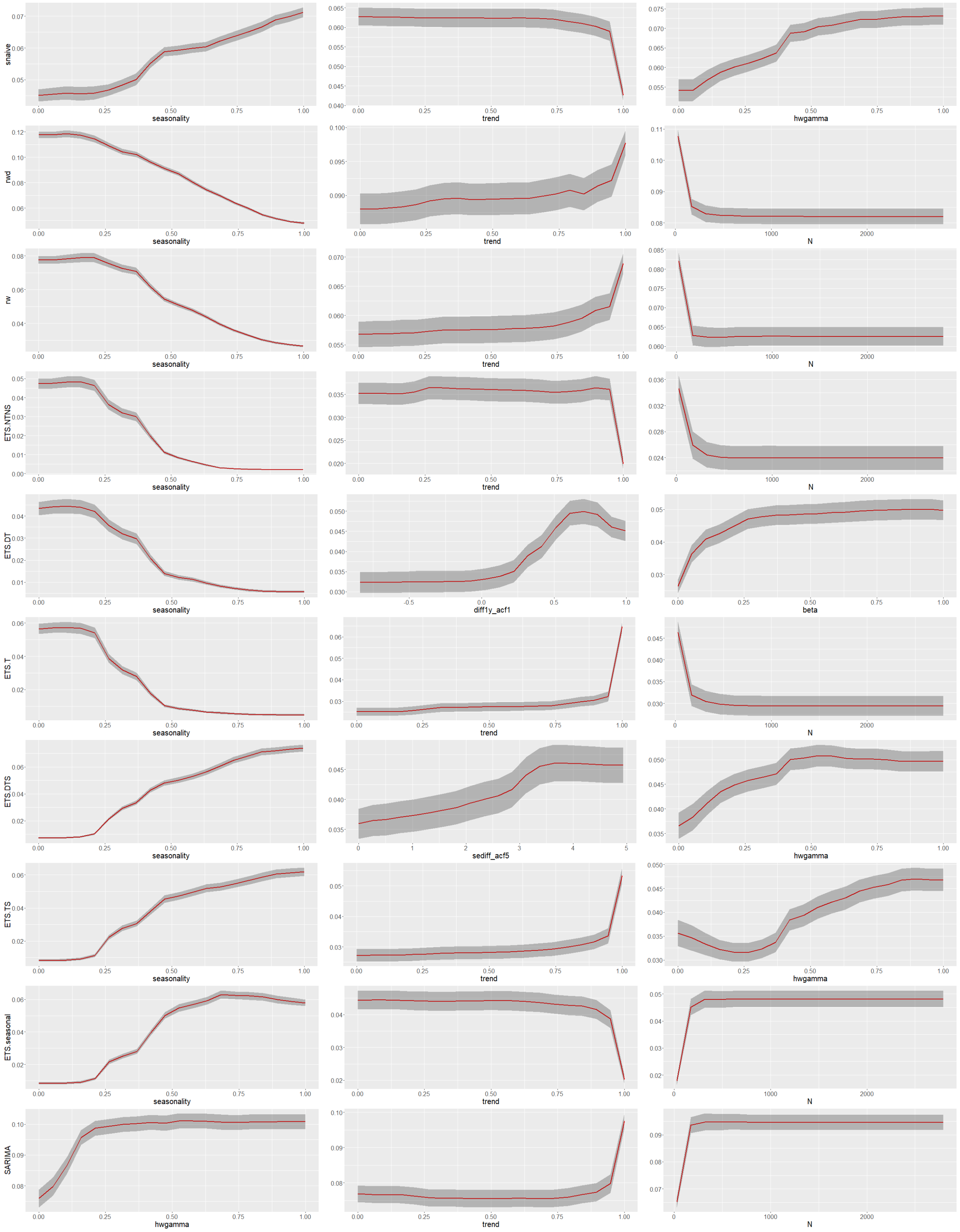

**Figure 17:** *Partial dependence plots for top three features in each category (Monthly)*

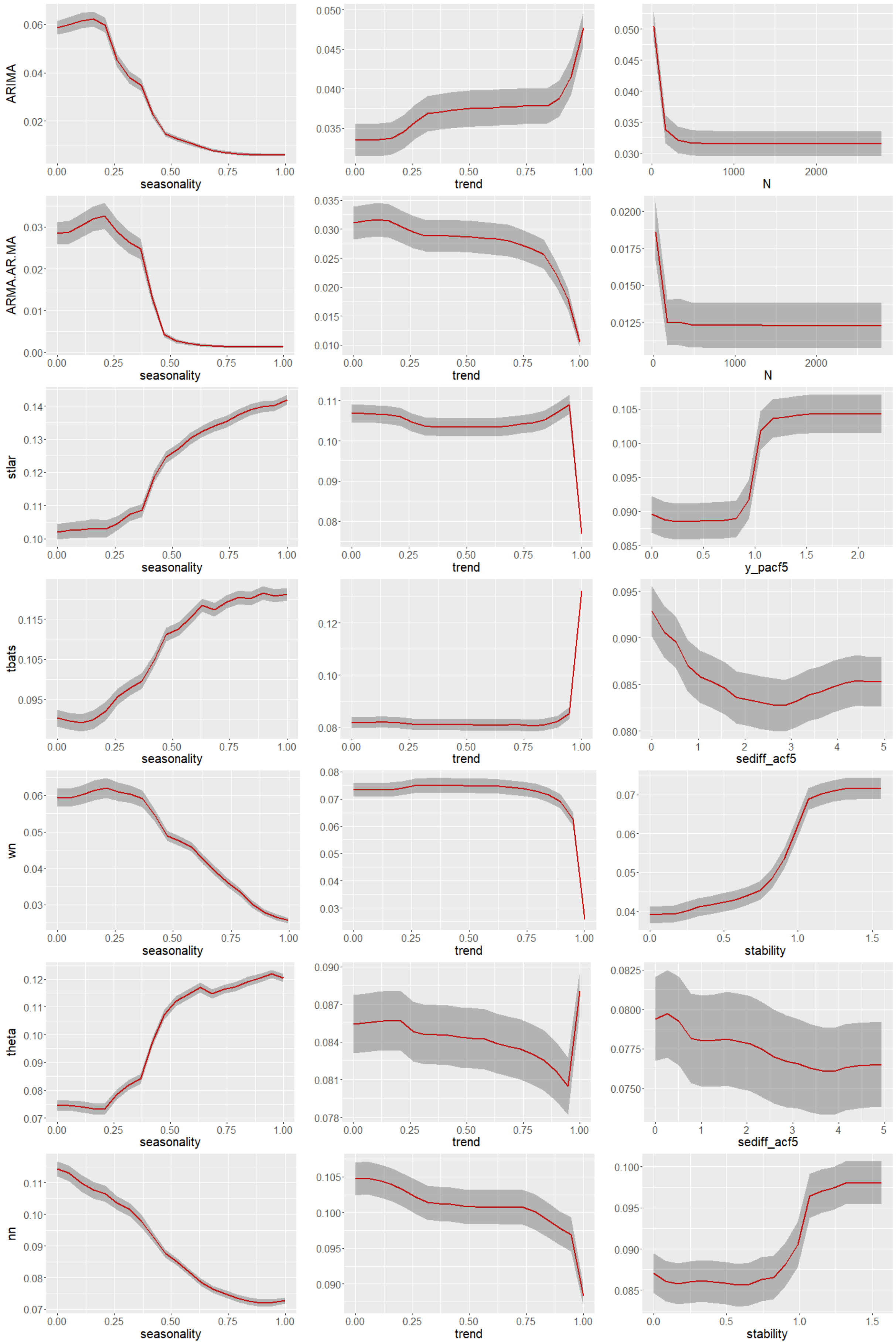

**Figure 18:** *Partial dependence plots for top three features in each category (Monthly)*

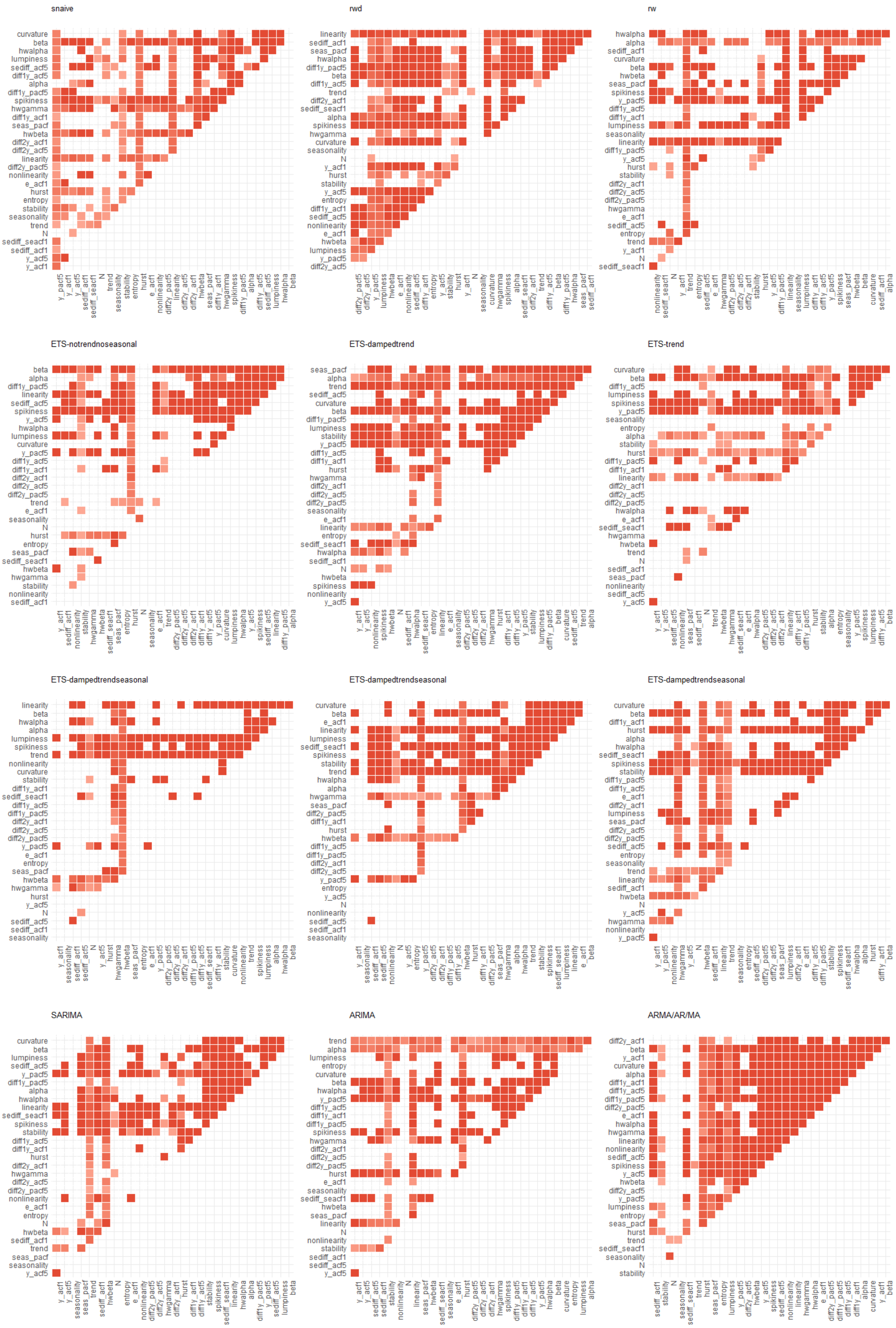

**Figure 19:** *Heat maps of relative strength of all possible pairwise interactions calclated based on Friedman's H-statistic for monthly data.*

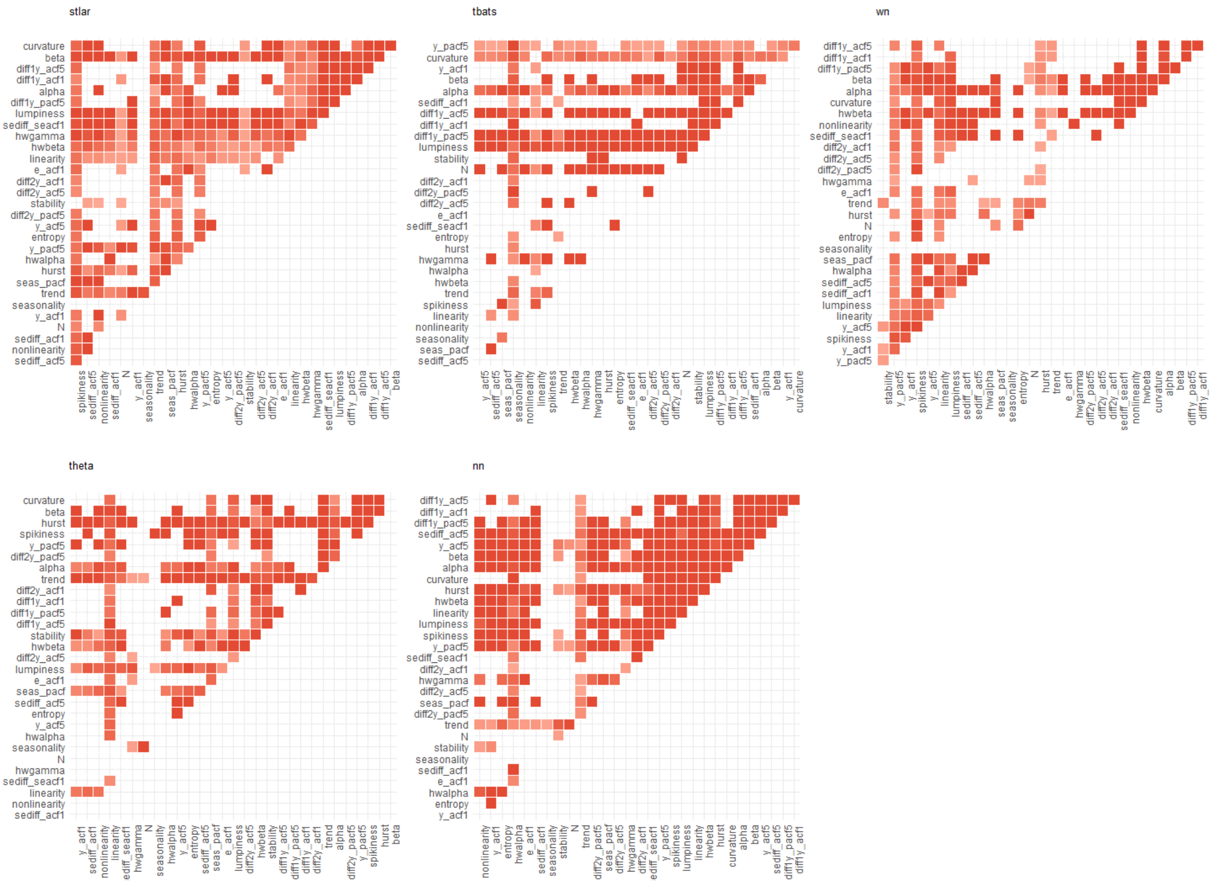

**Figure 20:** *Heat maps of relative strength of all possible pairwise interactions calclated based on Friedman's H-statistic for monthly data.*

### 4.4 Weekly

Figure 21 proportion of times each time series was classified to each class. Unlike, yearly, quarterly and monthly data theta models have low chance of being selected to forecast weekly data. The random walk with drift, tbat models and theta have higher chance of being selected. Except ARMA/AR/MA class the distribution corresponds to the true class label dominates others. ARMA/AR/MA class shows some unusual behaviour within some categories due to class imbalance ratio, ARMA/AR/MA class contains fewer number of observations in the training set. According to the results of Figure 22 spikiness, linearity, trend, strength of seasonality, stability and lumpiness has 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. According to the results of Figure 23 for mstlets models probability of being selected increases as the length of series increases while the opposite relationship was observed at neural network models. The tabts models show a non-monotonic relationship with the length of series. It is surprising to observe that for mstlets models probability of being selected decreases an seasonality increases. This could be due to the interaction effect of seasonality with others. For weekly data, the number of pairs showing a particular degree of interaction strength is relatively low compared to other yearly, quarterly and monthly data. In tbats class, trend, linearity, spikiness, seasonality and first autocorrelation coefficient of the seasonally differenced series interact heavily with other features.

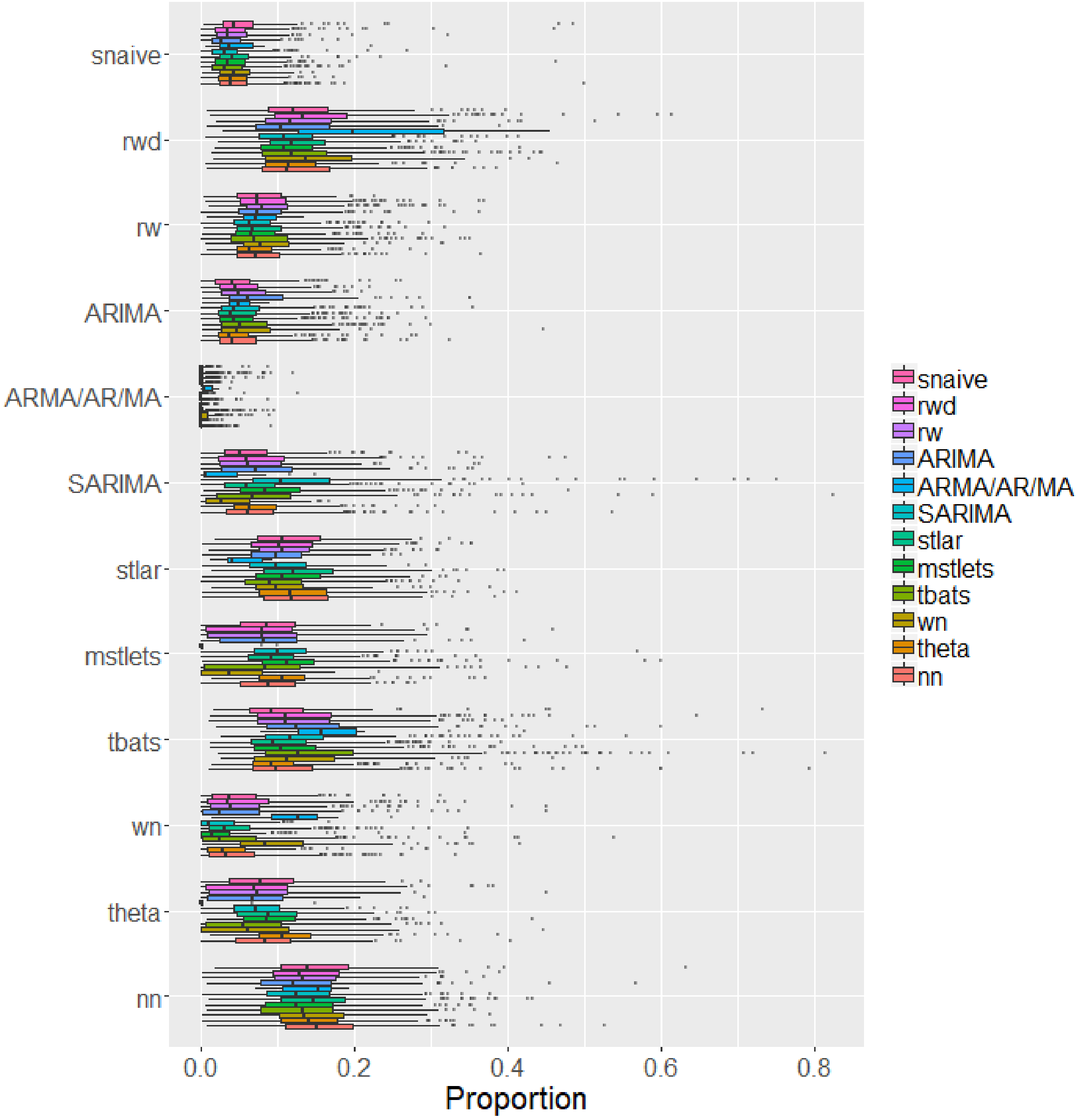

**Figure 21:** *Distribution of proportion of times each weekly time series was classified to each class in the forest. Each row represent the predicted class label and colour of boxplots corresponds to true class label. There are ten rows in the plot corresponds to each predicted class represented by Y-axis. X-axis denotes the proportion of times a time series is classified in each class.*

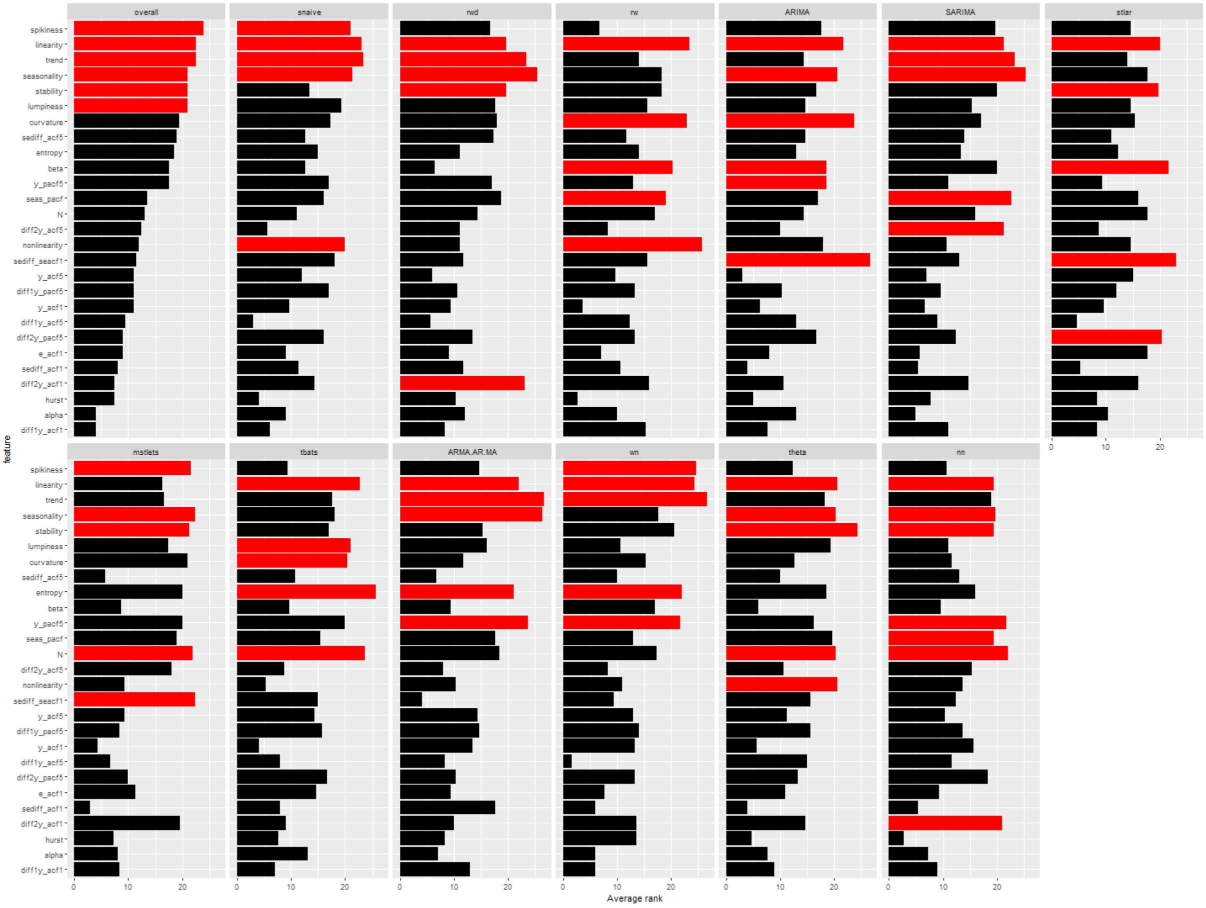

**Figure 22:** *Feature importance plot for weekly series. Longer bars indicate more important features. Top 5 features are highlighted in red.*

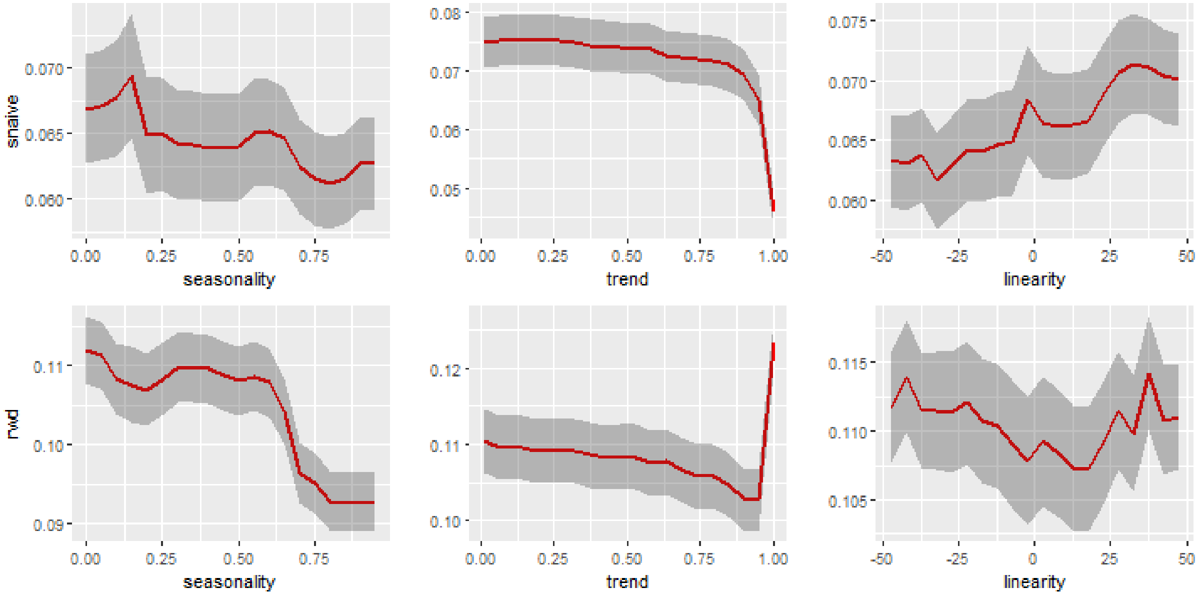

**Figure 23:** *Partial dependence plots for the top ranked features from variable importance measures(weekly series). The shading shows the 95% confidence intervals. Y-axis denotes the probability of belong to corresponding class.*

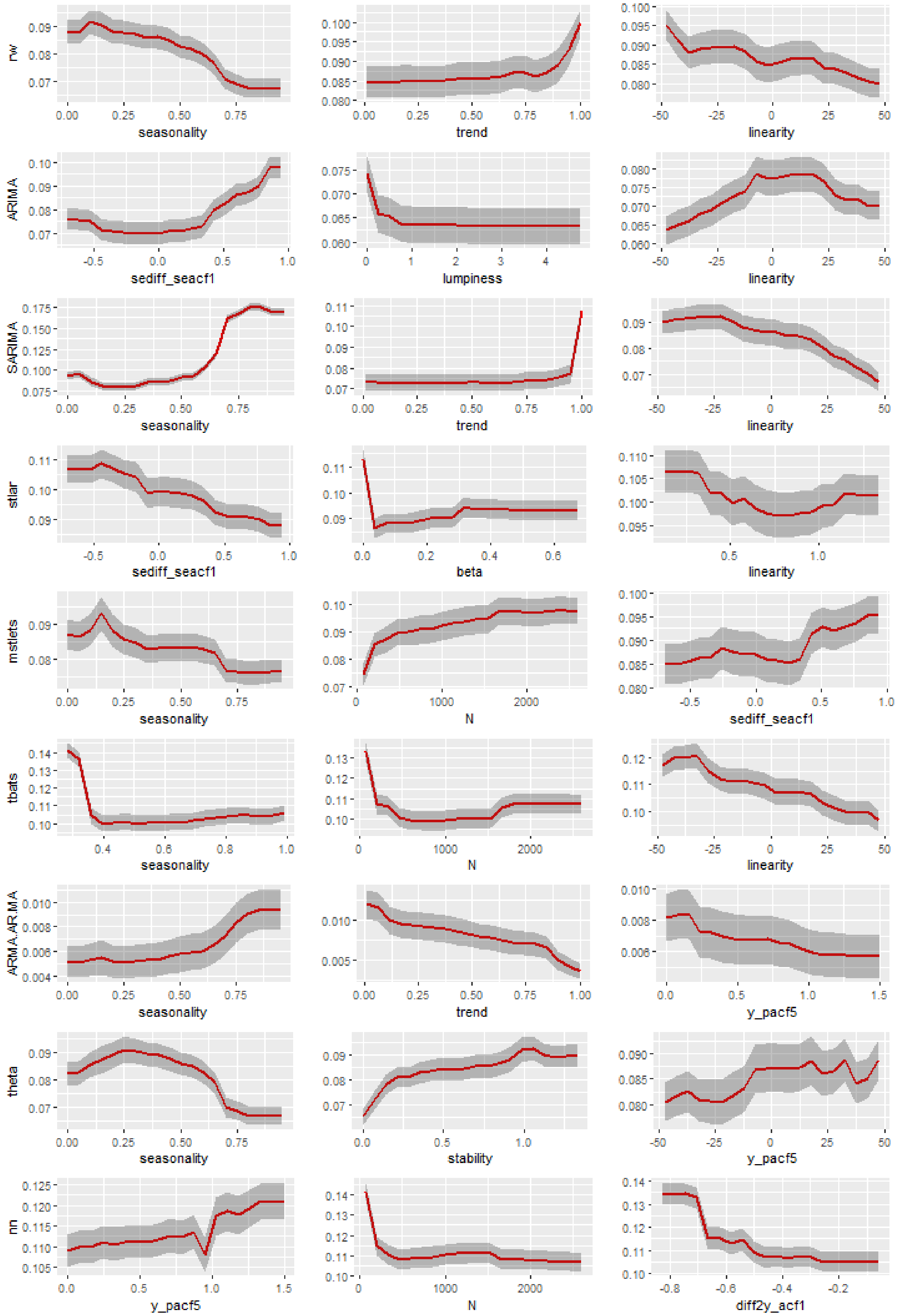

**Figure 24:** *Partial dependence plots for the top ranked features from variable importance measures(weekly series). The shading shows the 95% confidence intervals. Y-axis denotes the probability of belong to corresponding class(cont.).*

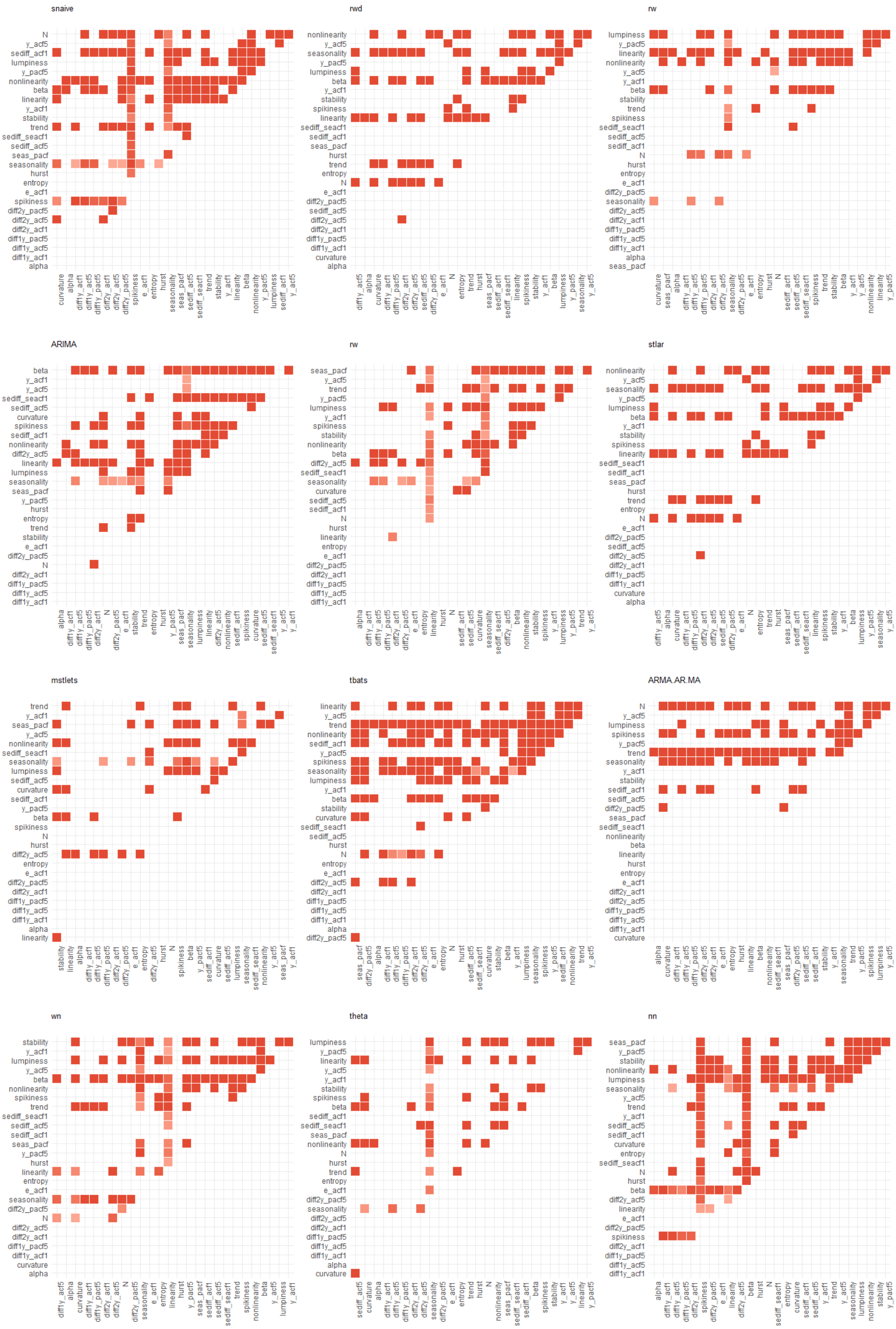

**Figure 25:** *Heat maps of relative strength of all possible pairwise interactions calclated based on Friedman's H-statistic for weekly data.*

### 4.5 Daily

According to Figure 26 the distributions corresponds to observations that have been correctly classified dominate the top, indicating a fairly good classification from FFORMS framework. However, there are few observations that have been incorrectly classified to tbats class with very high probabilities. In general, neural network models have higher chance of being selected for daily time series.

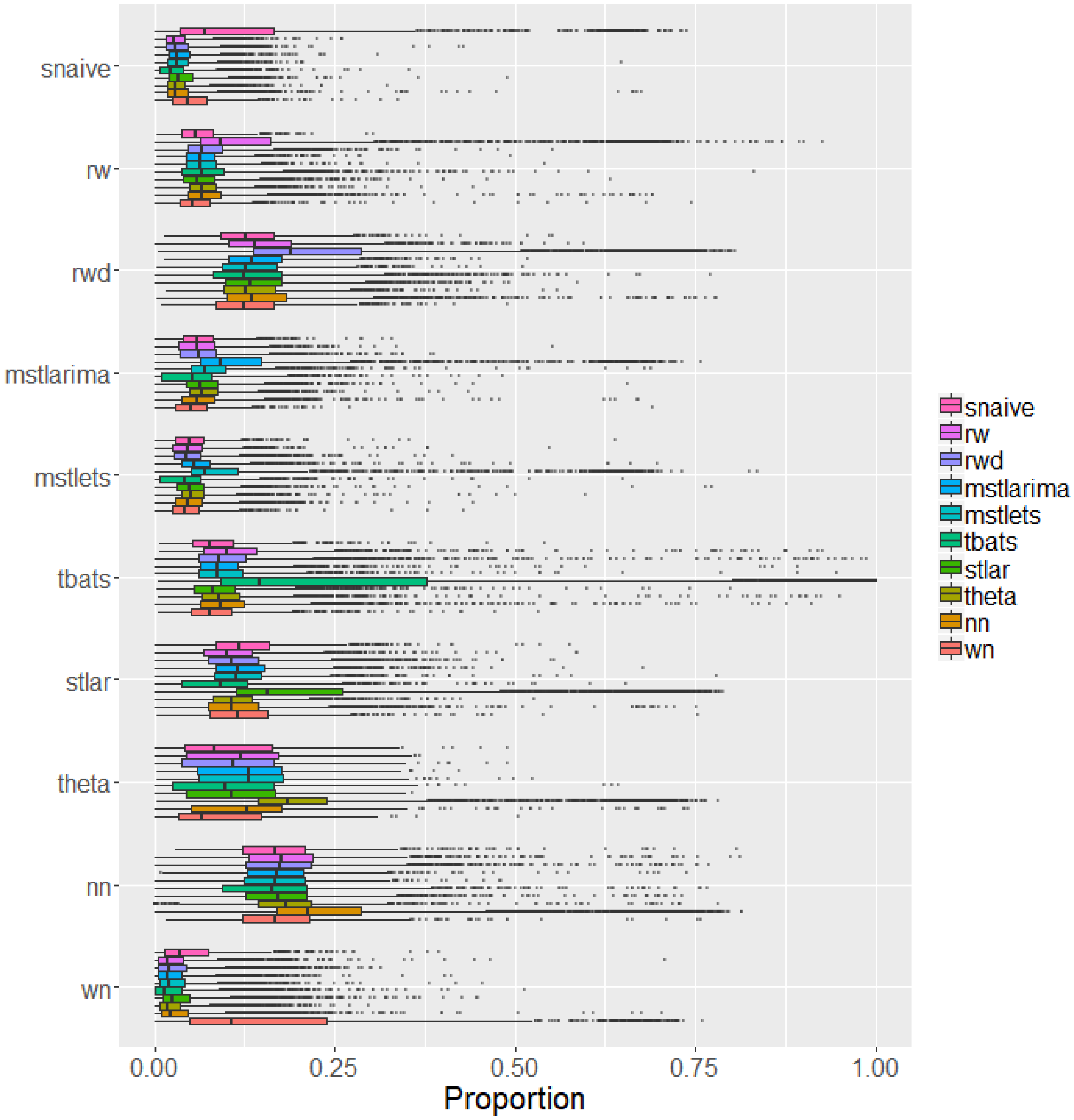

**Figure 26:** *Distribution of proportion of times each daily time series was assigned to each class in the forest. Each row represent the predicted class label and colour of boxplots corresponds to true class label. There are ten rows in the plot corresponds to each predicted class represented by Y-axis. X-axis denotes the proportion of times a time series is classified in each class. On each row, distribution of correctly classified class dominated the top, indicating a fairly good classification of the model fitted.*

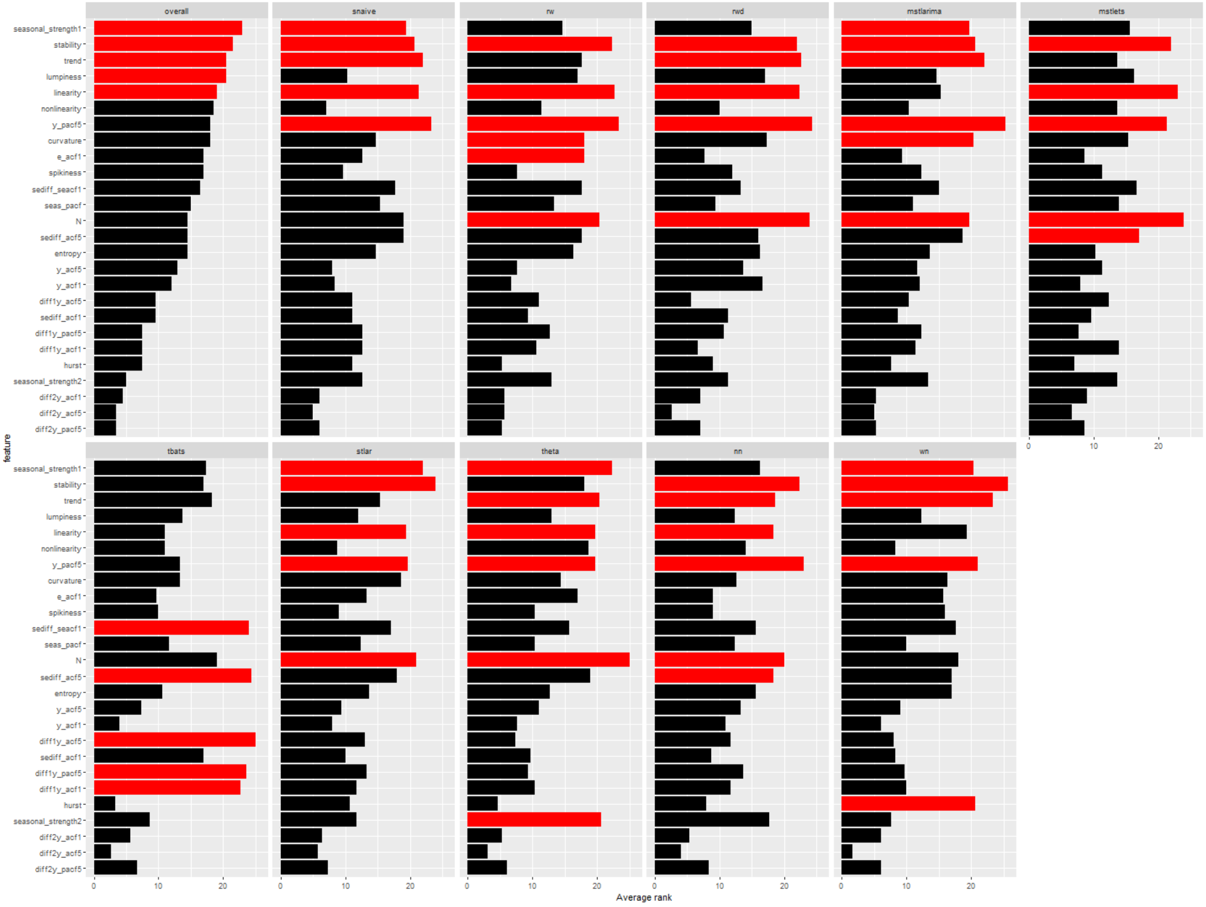

**Figure 27:** *Feature importance plot for daily series. Longer bars indicate more important features. Top 5 features are highlighted in red.*

Variable importance graph for daily data is shown in Figure 27. Overall, similar to the previous results most important features for determining suitable forecast-models for daily time series are, strength of seasonality corresponds to the weekly seasonality (7), stability, trend, lumpiness and linearity. Further more, length of the series is important in determining random walk, random walk with drift, mstlarima, mstlets, stlar, theta and nn classes. Strength of seasonality corresponds to the annual seasonality (365.25) is appeared to be important in determining the selection of theta models. Figure 28 shows the partial dependency plots of the top 3 features from the FFORMS framework. According to the results of Figure 28 shorter series tends to select random walk with drift models while probability of selecting tbats, and theta models increases as the length of series increases. Neural network models shows a non-monotonic relationship with length of the series (N). The theta models tends to be selected for series with high annual seasonality but very low weekly seasonality.

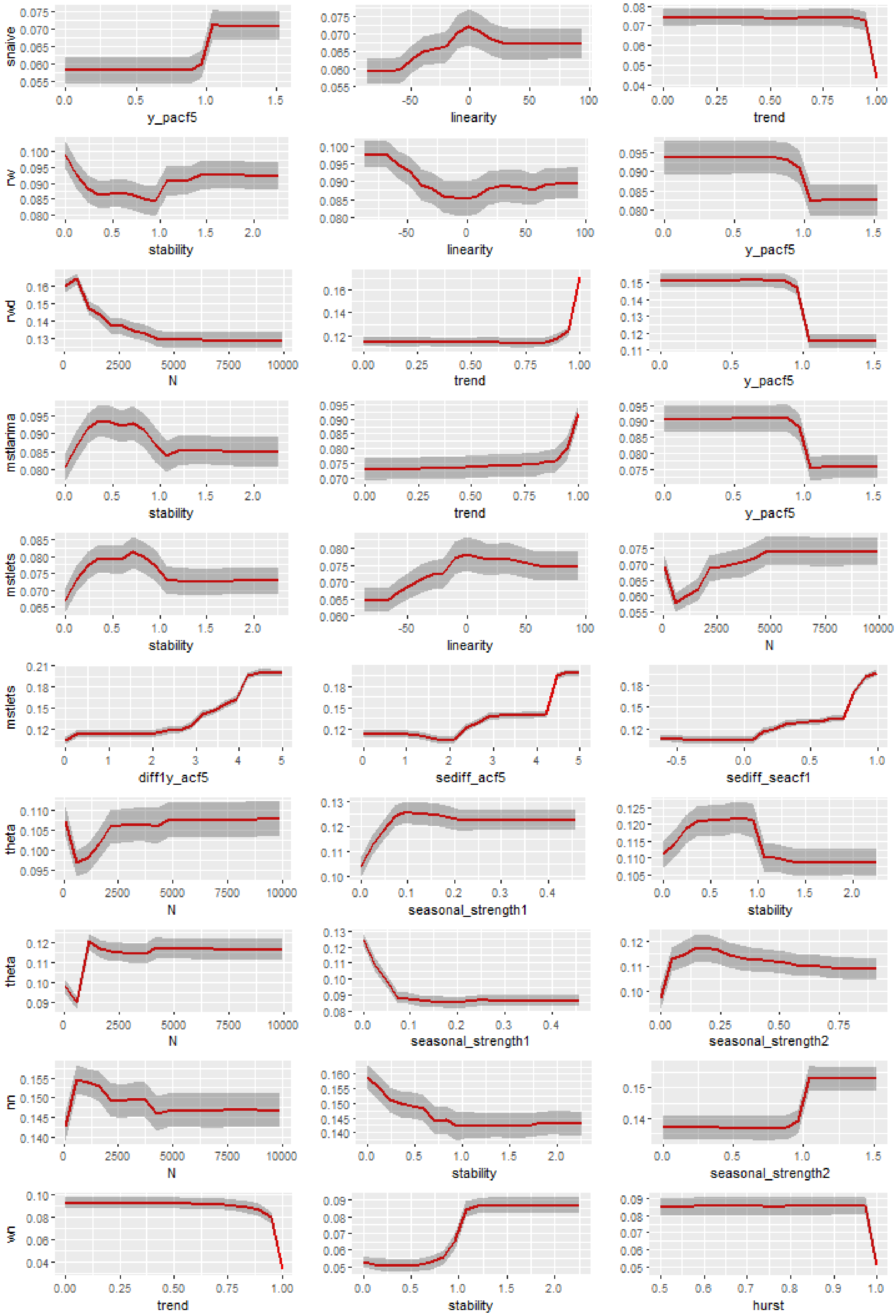

**Figure 28:** *Partial dependence plots for the top ranked features from variable importance measures(daily series). The shading shows the 95% confidence intervals. Y-axis denotes the probability of belong to corresponding class.*

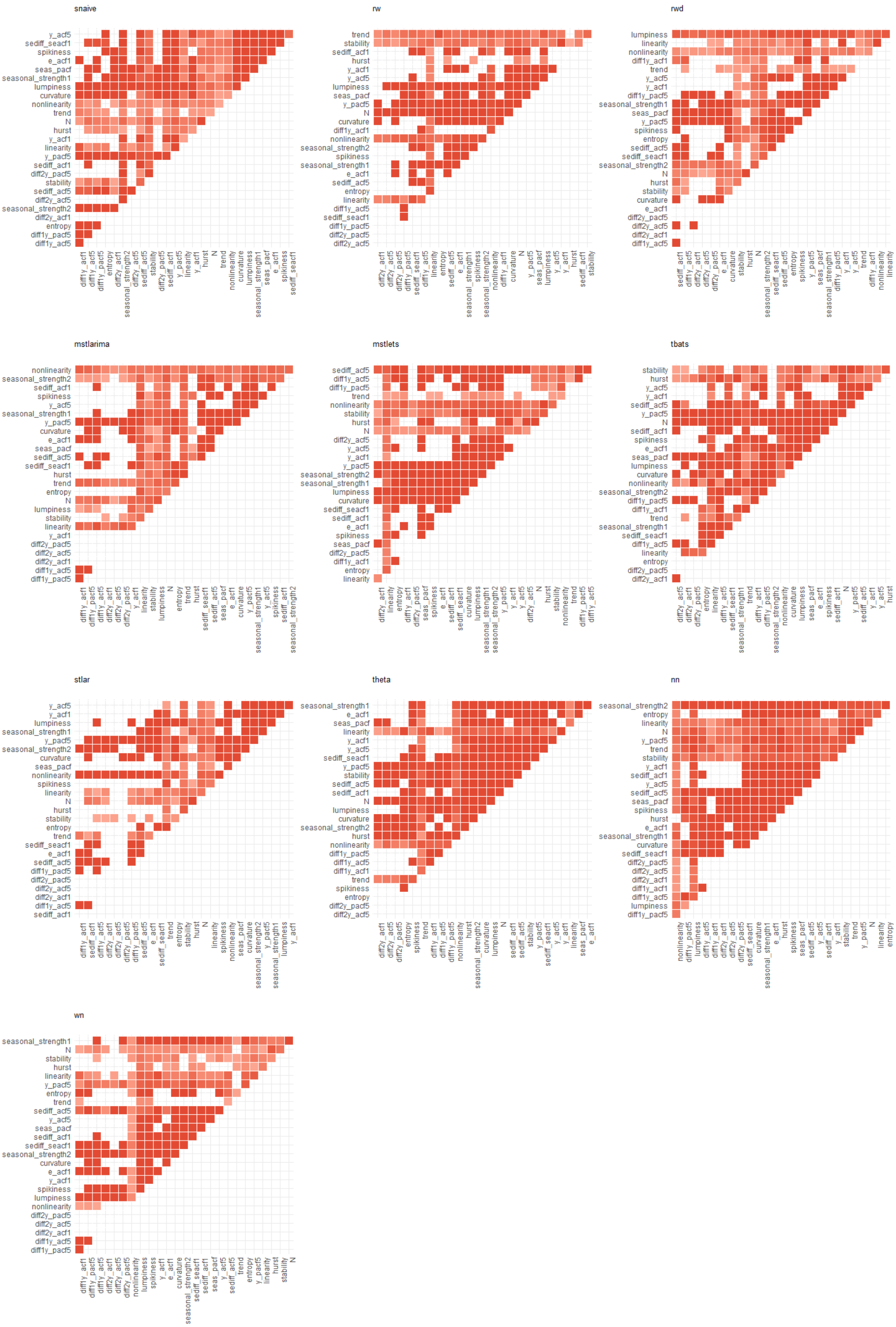

**Figure 29:** *Heat maps of relative strength of all possible pairwise interactions calclated based on Friedman's H-statistic for daily data.*

### 4.6 Hourly

The proportion of times each time series was classified to each class for hourly series are shown in Figure 30. Except neural network class, the distribution corresponds to the correct classification dominated the top. It is important to note that, in neural network category, distribution corresponds to snaive class slightly dominated the distribution corresponds to neural network category. Overall, for hourly series random walk with drift models, tbats and neural network models have high chance of being selected. Furthermore, all distributions corresponds to the neural network category located further away from zero with respect to x-axis indicates, every time series selects neural network models at least once from the trees in the random forest. In general, strength of daily seasonality (period=24) appear to be more important than than the strength of weekly seasonality (period=168). 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. Although the length of the series found to be one of the most important feature from previous results, only snaive category ranked it among top 5 for hourly time series. The strength of seasonality corresponds to weekly seasonality also seems to be one of the most important feature for the classes snaive, random walk, mstlarima, and tbats. The partial dependency plots shows that probability of selecting random walk, random walk with drift, theta model and white noise process decreased with higher strength of daily seasonality, while the opposite relationship hold for other classes.On the other hand, probability of selecting random walk model increased with the increase in strength of weekly seasonality.

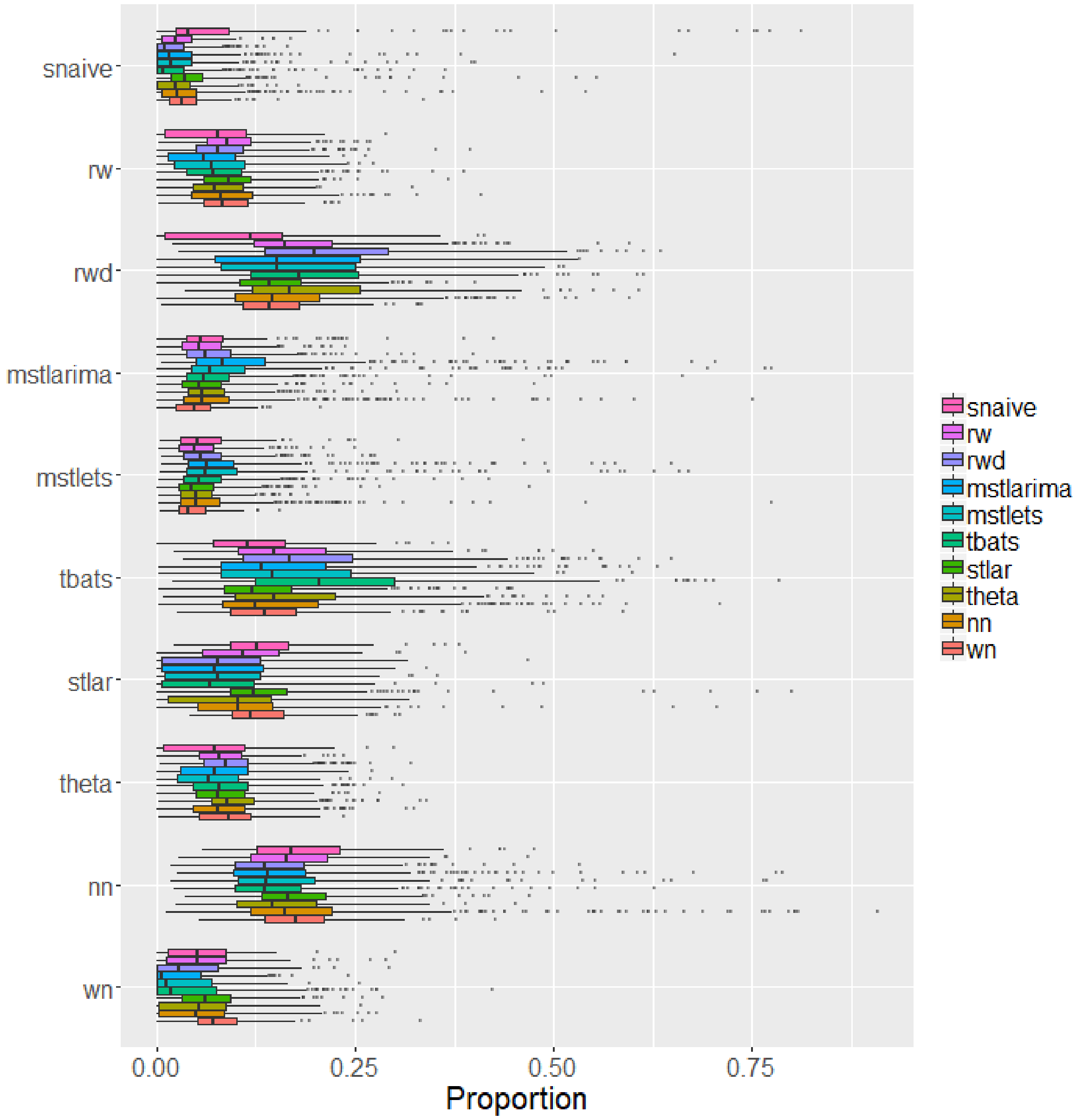

**Figure 30:** *Distribution of proportion of times each hourly time series was classified to each class in the forest. Each row represent the predicted class label and colour of boxplots corresponds to true class label. There are ten rows in the plot corresponds to each predicted class represented by Y-axis. X-axis denotes the proportion of times a time series is classified in each class. On each row, the distribution of correct classification dominated the top.*

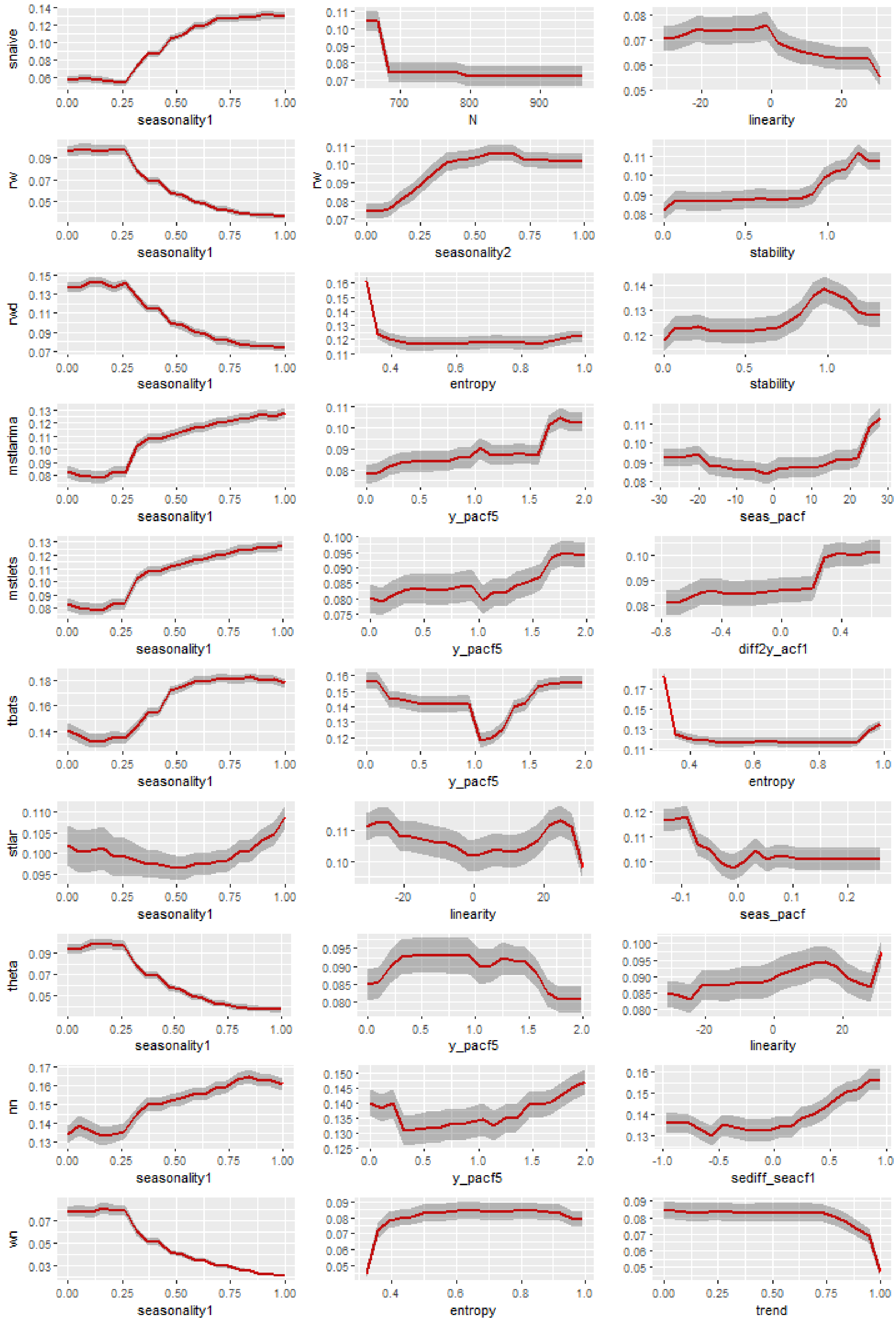

**Figure 31:** *Partial dependence plots for the top ranked features from variable importance measures(hourly series). The shading shows the 95% confidence intervals. Y-axis denotes the probability of belong to corresponding class.*

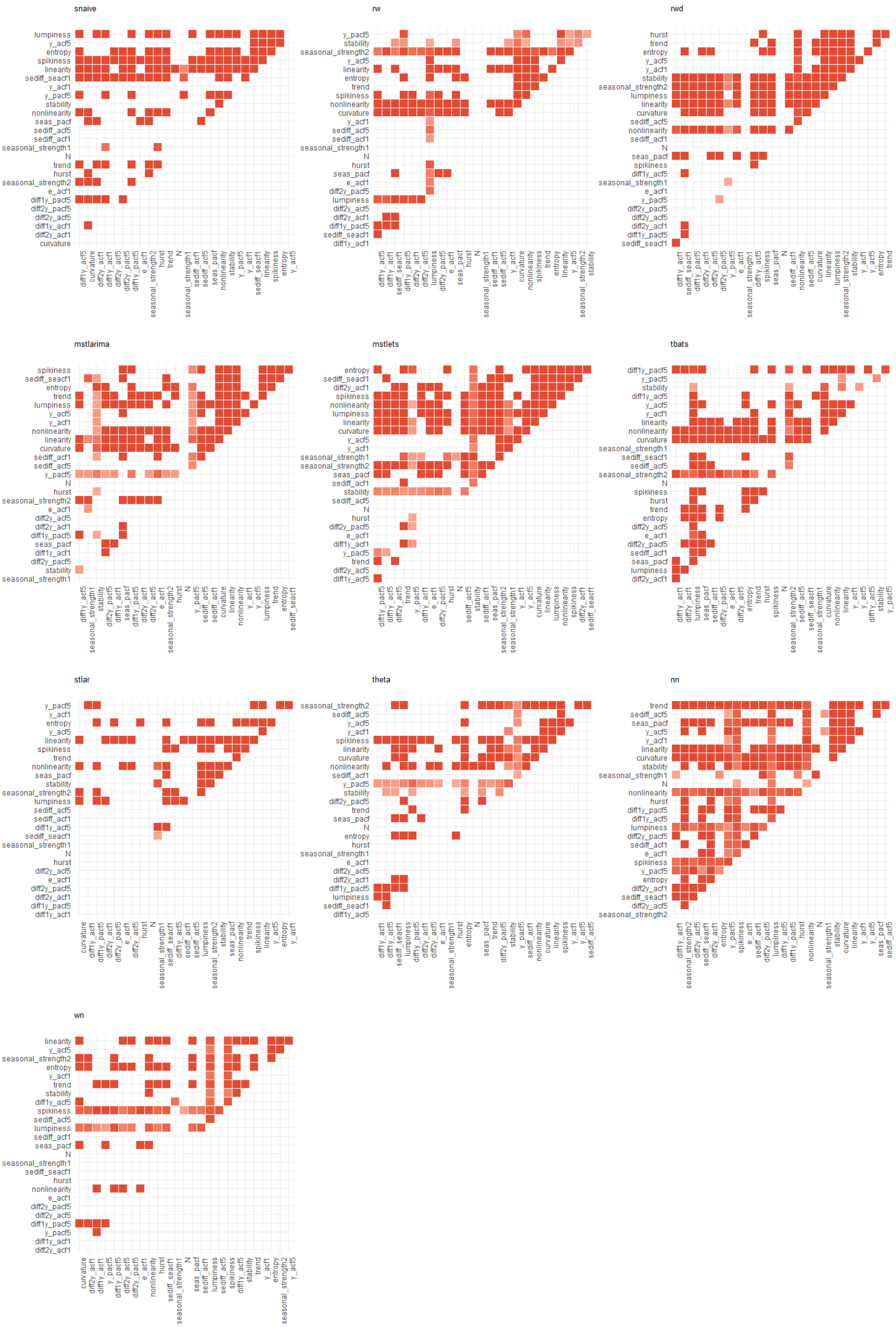

**Figure 32:** *Heat maps of relative strength of all possible pairwise interactions calclated based on Friedman's H-statistic for hourly data.*

# 5 Discussion and Conclusions

# Appendix

## PCA space

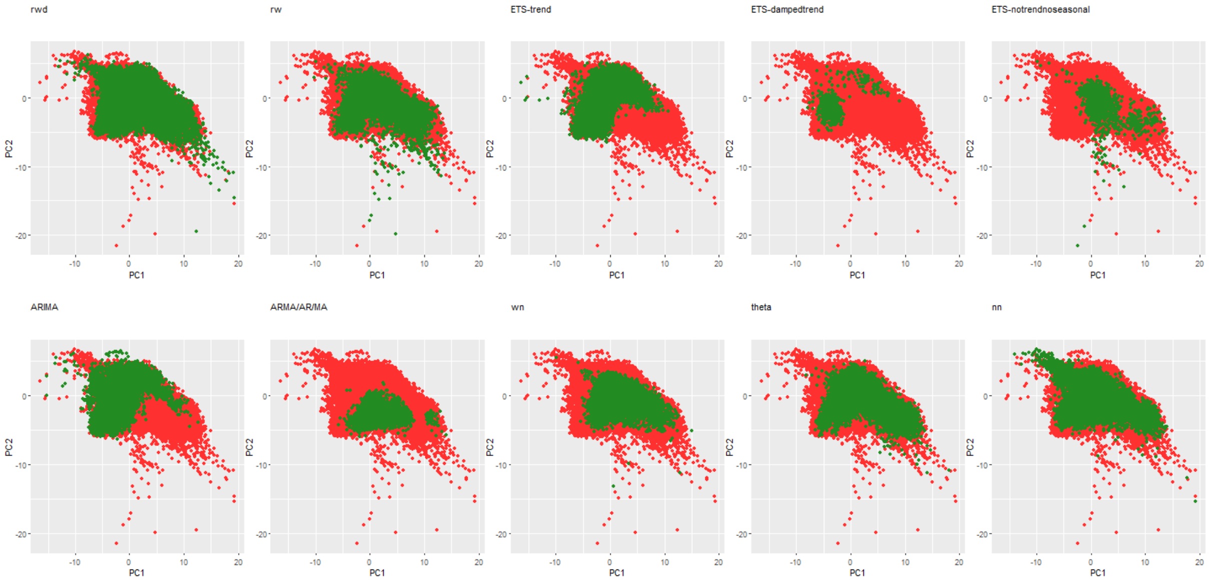

**Figure 33:** *Distribution of prediction space across instance space for yearly time series. PC1 and PC2 are the first two principal components.*

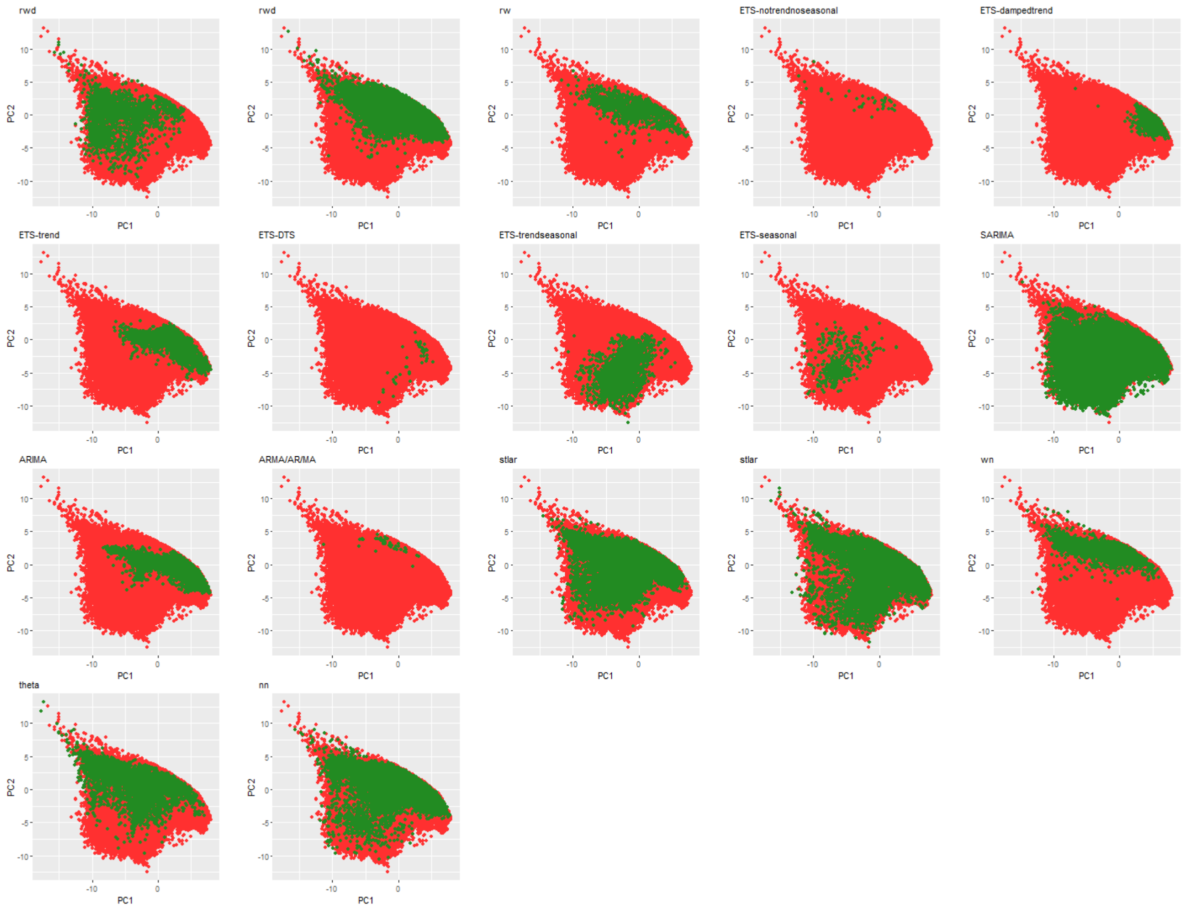

**Figure 34:** *Distribution of prediction space across instance space for quarterly time series. PC1 and PC2 are the first two principal components.*

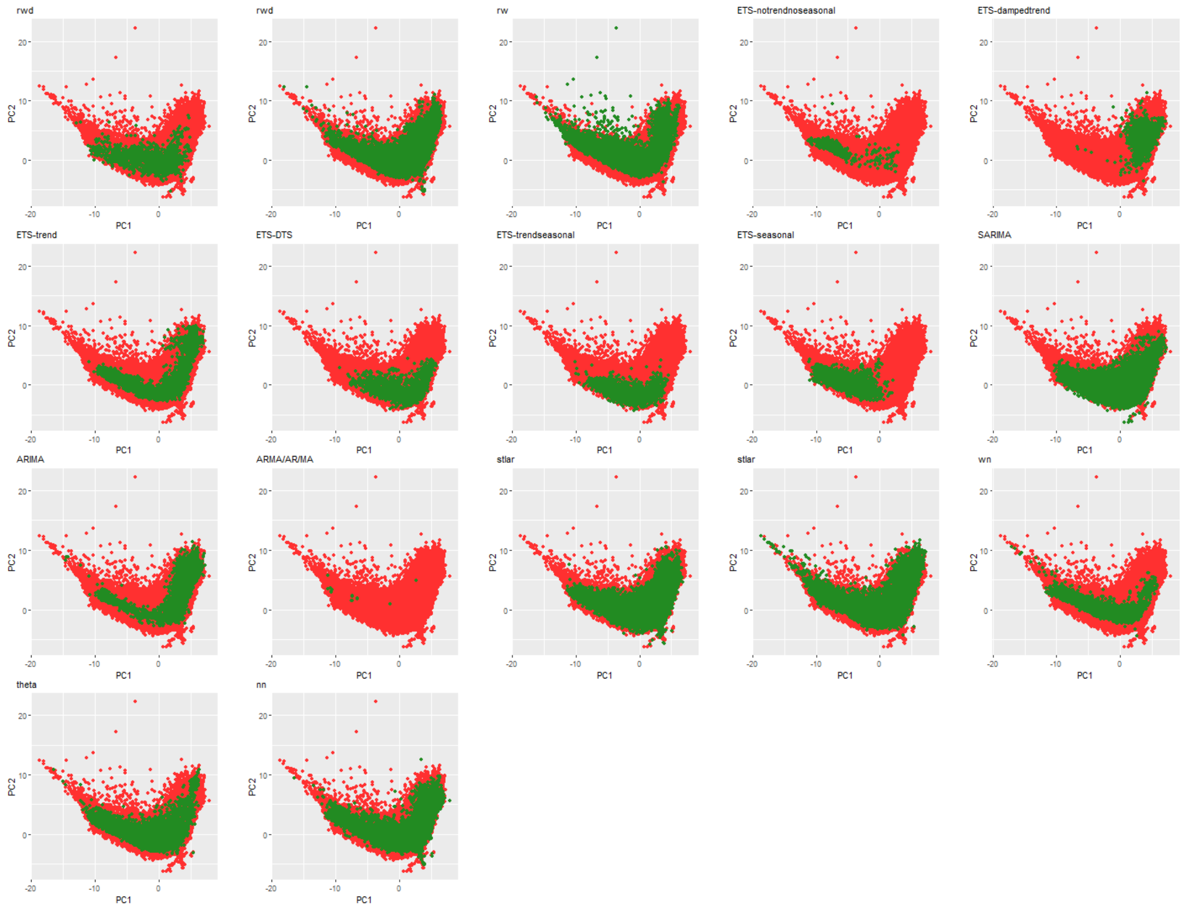

**Figure 35:** *Distribution of prediction space across instance space for monthly time series. PC1 and PC2 are the first two principal components.*

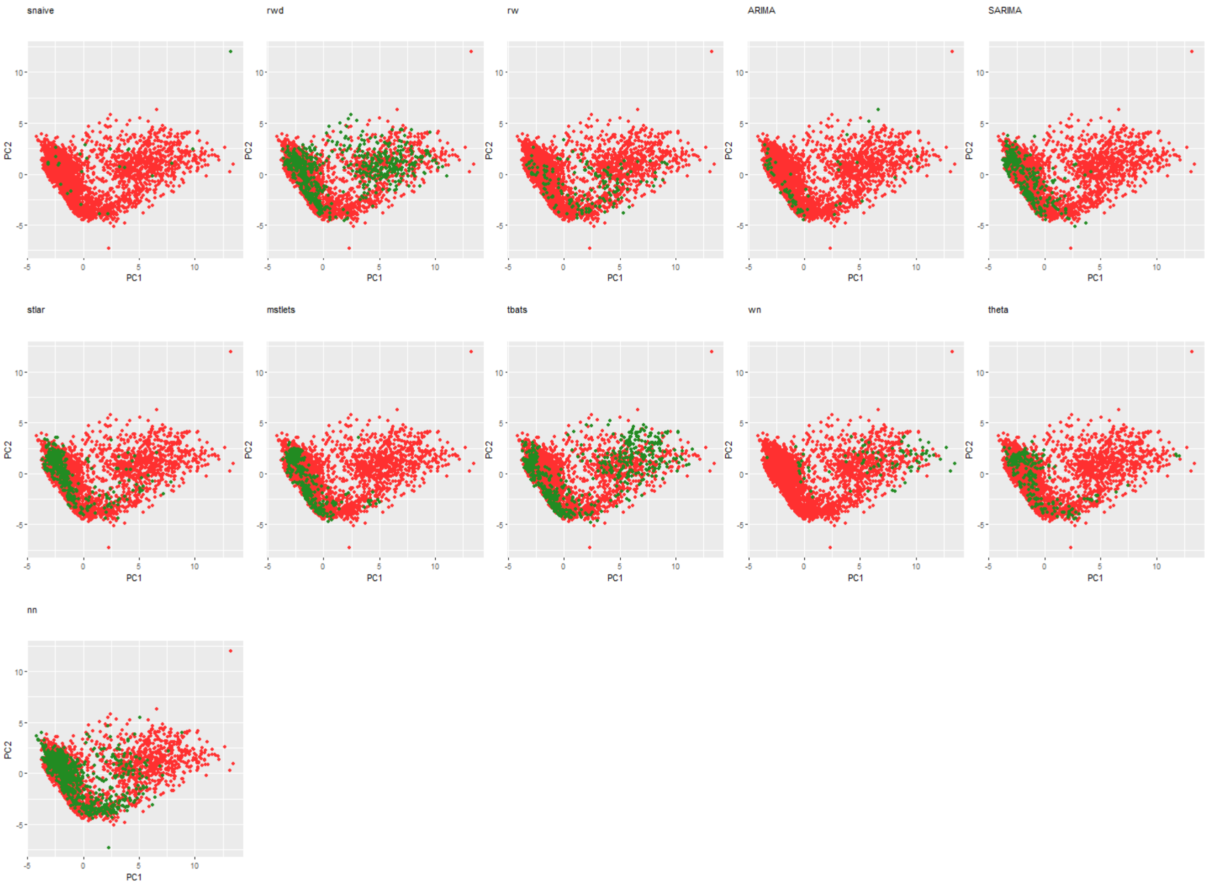

**Figure 36:** *Distribution of prediction space across instance space for weekly time series. PC1 and PC2 are the first two principal components.*

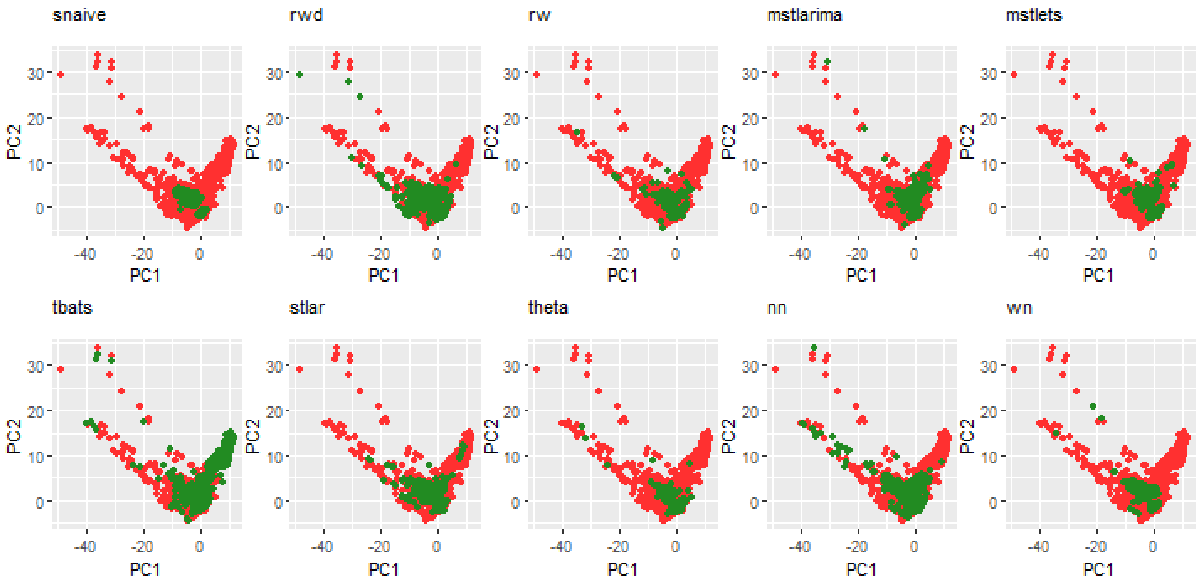

**Figure 37:** *Distribution of prediction space across instance space for daily time series. PC1 and PC2 are the first two principal components.*

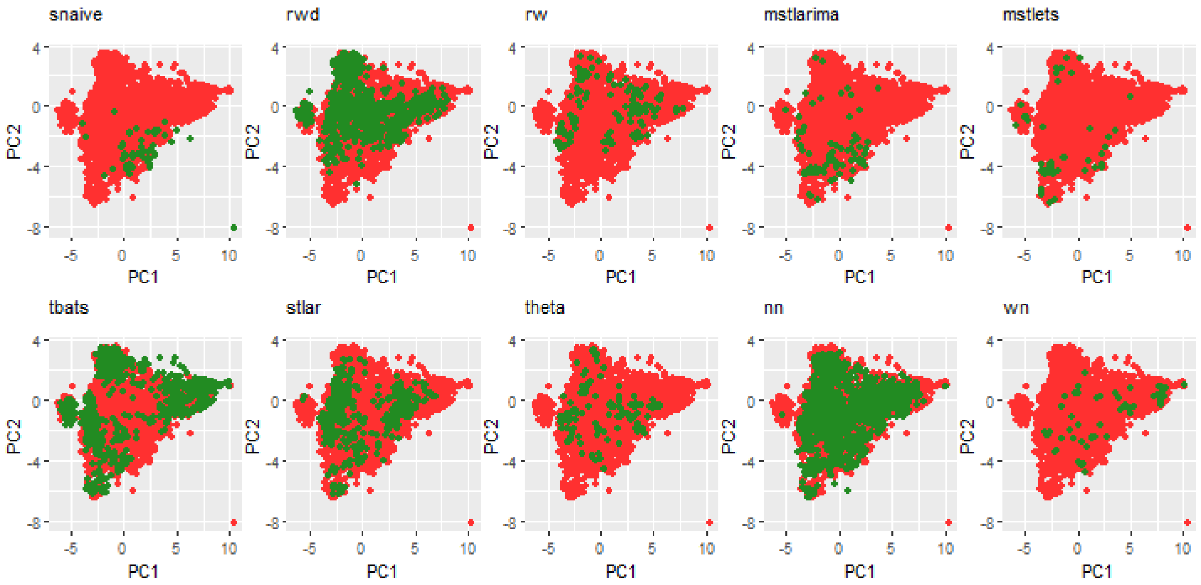

**Figure 38:** *Distribution of prediction space across instance space for hourly time series. PC1 and PC2 are the first two principal components.*

## References

Breiman, L (2001). Random forests. *Machine Learning* **45**(1), 5–32.

Chen, C, A Liaw & L Breiman (2004). *Using random forest to learn imbalanced data*. Tech. rep. University of California, Berkeley. http://statistics.berkeley.edu/sites/default/files/tech-reports/666.pdf.

Ehrlinger, J (2015). ggRandomForests: Visually Exploring a Random Forest for Regression. *arXiv preprint arXiv:1501.07196*.

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.

Hyndman, R, G Athanasopoulos, C Bergmeir, G Caceres, L Chhay, M O'Hara-Wild, F Petropoulos, S Razbash, E Wang & F Yasmeen (2018). *forecast: Forecasting functions for time series and linear models*. R package version 8.3. http://pkg.robjhyndman.com/forecast.

Jiang, T & AB Owen (2002). Quasi-regression for visualization and interpretation of black box functions. *Technical Report, Stanford University.*

Kang, Y, RJ Hyndman, F Li, et al. (2018). *Efficient generation of time series with diverse and controllable characteristics*. Tech. rep. Monash University, Department of Econometrics and Business Statistics.

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*. IEEE, 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: *Advances in Neural Information Processing Systems*, pp.4765–4774.

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.

Molnar, C, G Casalicchio & B Bischl (2018). Iml: An R package for interpretable machine learning. *The Journal of Open Source Software* **3**(786), 10–21105.

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*. ACM, pp.1135–1144. https://doi.org/10.1145/2939672.2939778.

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.*

Sutherland, P, A Rossini, T Lumley, N Lewin-Koh, J Dickerson, Z Cox & D Cook (2000). Orca: A visualization toolkit for high-dimensional data. *Journal of Computational and Graphical Statistics* **9**(3), 509–529.

Talagala, TS, RJ Hyndman & G Athanasopoulos (2018). Meta-learning how to forecast time series. *Technical Report 6/18, Monash University.*

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 blindfold. *Statistical Analysis and Data Mining: The ASA Data Science Journal* **8**(4), 203–225.

Zhao, Q & T Hastie (n.d.). Causal Interpretations of black-box models ().