Investigating the Application of Machine Learning to Forecast Industrial Univariate Time Series using Meta Learning

Author: Gabriel Dalforno Silvestre
Advisor: André C. P. L. F. de Carvalho

Institute of Mathematics and Computer Science
University of São Paulo

April 2021

Abstract

The usage of machine learning models to forecast time series has been a subject of great interest in the artificial intelligence community in the past few years, however it is not known yet in which conditions a machine learning approach can be better than the classical statistical modelling. The goal of this work is to investigate if there are features, in the industrial time series from the M4 competition, that can tell us whether a machine learning model has high probability of outperform or be outperformed by a statistical approach using meta learning tools, more specifically an algorithm recommendation system. The results showed that the time series considered have not much information that allows the system to distinguish with high confidence the difference between machine learning and statistical approaches.

keywords: Meta Learning, Machine Learning, Time Series Forecasting, Industrial Data, Data Analysis, M4 Competition.

1 Introduction

There is a controversy within the artificial intelligence community about the application of machine learning (ML) approaches to forecast time series data, since it is well known that the latter does not usually satisfy the necessary conditions imposed by the statistical learning theory to guarantee "learning" [25]. However, the same theory states that we cannot ensure the learning is not happening as well, this gap gave to the researches and practitioners a hope that machine learning techniques can be incorporated into the time series forecasting toolkit [3]. In the last few years, several works that modify or design ML algorithms for time series modelling purposes have been developed, for instance, [20] modified the original K-nearest-neighbors algorithm to forecast time series data using complexity measures; [1] explored the application of the Long-Short-Term-Memory to forecast groups of similar time series and [14] applied Support-Vector-Machines to forecast financial data.

Furthermore, other works aimed to compare ML and statistical approaches to model time series data [22, 8, 6, 16], but still there is no consensus about whether a ML model should be applied instead of a classical ARIMA or ETS model [13], for instance. This issue leads practitioners to brute force approaches to selected the models with the highest probability of success, which consume a lot of time and computational resources.

Recent works have used the meta learning (MTL) paradigm to deal with the problem of model selection in time series forecasting, [15] used a convolutional neural network to find the best combination of models to forecast retail sales data; [7] explored the usage of arbitrating in the forecasting task and the second place of the M4 competition [17] is also based in a MTL approach that assign weights to various models in order to minimize their average error [19]. Those techniques have showed themselves as powerful tools since they accumulate experience from previous tasks to improve the results when applied to new ones.

In this work, we will apply MTL to recommend either a group of statistical or a group of ML models to forecast univariate industrial time series from the M4 competition. The goal is to investigate if there are features in those datasets that can give us a clue of when a ML model will be better to forecast determined time series, in comparison with a statistical algorithm. It is expected that the information learned can be used to answer the question: What are the characteristics we need to look for in a time series to be confident that a ML model is the best option to predict future observations?

We used the predictions from 12 out of the 61 submitted to the M4 Competition: the 6 submissions classified as "pure ML models" by the organizers and 6 statistical benchmarks: Theta, ARIMA, Damped, ETS, Holt, SES. All the data used is available at the public repository [10]. To be consistent with the competition, we used the same metrics to compare the approaches: SMAPE (Symmetric Mean Absolute Percentage Error), MASE (Mean Absolute Scaled Error) and the OWA (Overall Weighted Average). The time series features were extracted using [24] and the evaluation of the meta learner was done by a stratified holdout with 10 folds, finally, if a test was necessary to tell if two samples are statistically different from each other we used the Kolmogorov-Smirnov test on 2 samples.

The work is organized as follows: Section 2 discuss the M4 Competition; Section 3 describes the MTL approach applied; Section 4 presents the experimental setup that leads to the results in Section 5; The conclusions and discussions are set on the Section 6 and future work on the Section 7.

2 M4 Competition

The M4 Competition [17] took place in 2018, it brought some changes in comparison with the previous ones: increased number of time series and forecasting methods, and the addition of prediction intervals along the point forecasts. The first two, in particular, makes it a rich environment to apply MTL and other global techniques. The time series in the competition are organized as follows:

Interval	**Micro**	**Industry**	**Macro**	**Finance**	**Demog**	**Other**	**Total**
Yearly	6538	3716	3903	6519	1088	1236	23000
Quarterly	6020	4637	5315	5305	1858	865	24000
Monthly	10975	10017	10016	10987	5728	277	48000
Weekly	112	6	41	164	24	12	359
Daily	1476	422	127	1559	10	633	4227
Hourly	0	0	0	0	0	414	414
Total	25121	18798	19402	24534	8708	3437	100000

Table 1: M4 Competition time series

When it comes to forecasting methods, 61 different techniques were considered, 12 benchmarks and models for standard comparison, and 49 valid

submissions. Furthermore, they were divided into 4 categories: hybrid, statistical, ML and combination. Even though it is not clear how that distinction was done [2], for our purposes it is safe to assume it like so.

Among the findings of this competition, we are particularly interested in the poor performance of pure ML methods, in the overall comparison, none of the methods performed better than the Comb benchmark (simple arithmetic average of SES, Holt and Damped exponential smoothing) and only one method was more accurate than the Naïve 2 method (Random Walk with seasonality adjusted data). Several reasons were brought up to explain these results, for instance, limited usage of cross-learning [23], limited sample sizes, effect of over-fitting and non-stationarity of the time series. However, they are not enough to generalize as a rule to decide if a ML based method will perform poorly or not, what is needed, in fact, is a general characterization of the time series in which the a ML model has a high probability to have a good performance and then try to overcome the others by designing new algorithms or combining the existing ones with statistical methods. The latter showed up as the best approach in the M4 competition, i.e., the top submissions were combinations of ML and statistical models, then understanding the weak points of those methods can help us to decide the best way to combine them.

The problem of characterization will be addressed using a MTL model recommendation system.

3 Meta Learning

The MTL paradigm can be summarized as the "learning to learning" process, in other words, it accumulates experience from previous tasks in order to recommend the best way to sort out a new one [4].

Recall the classical supervised learning setup, it is fixed an input space $\mathcal{X}$, an output space $\mathcal{Y}$ and a probability measure μ on $\mathcal{X} \times \mathcal{Y}$, then we would like to find a function $f \in \mathcal{H} \subset \{f | f : \mathcal{X} \to \mathcal{Y}\}$ such that $f(x) \approx y, \forall (x, y) \sim \mu$. If one fix a loss function L that characterizes mathematically the relation "$\approx$", the problem can be reformulated as shown in the Equation 1.

$$\inf_{f \in \mathcal{H}} \mathbb{E}_{(x,y)\sim\mu}[L(y, f(x))] = \inf_{f \in \mathcal{H}} \int_{\mathcal{X}\times\mathcal{Y}} L(y, f(x)) d\mu(x, y) \tag{1}$$

Since μ is unknown the above problem is impossible to treat, though by

the empirical risk minimization principle [25] given a dataset, denoted as $\mathcal{D} = \{(x_i, y_i)|(x_i, y_i) \sim \mu, i = 1, \ldots, n\}$, it is possible to approximate Equation 1 using Equation 2, which is now treatable computationally speaking.

$$\inf_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(x_i)) \tag{2}$$

Our goal is to re-frame the model selection problem into this setting and treat it as a supervised binary classification framework. In this work, we will perform group model selection, which instead of recommending a single approach to forecast a time series, it outputs a set of approaches that can be used for the same purpose.

Given a set of tasks $\mathcal{T} = \{\tau_i | i = 1, \ldots, m\}$, here each task represents a time series, and two groups of methods $M = \{\mathcal{A}_j^M | j = 1, \ldots, k\}$ and $S = \{\mathcal{A}_j^S | j = 1, \ldots, k\}$ we want to output for each task τ_i the group that contains the best average model to forecast it. This process requires two steps: the characterization of each task using meta features and the definition of the meta target.

3.1 Meta Features

A meta feature [21] is the output of any function $g : \tau \to \mathbb{R}$ that characterizes a task in terms of a meaningful real number, there are many different types of meta features for time series data [13], for instance:

- **Statistical features:** any numerical summarization as mean, variance, maximum, minimum, quantiles, etc;
- **ACF features:** information about the coefficients of autocorrelation from the original data or even the detrend and deseasonalized versions;
- **STL features:** characterizes the time series using information present in the output from the STL decomposition [9], for example, the strength of trend and seasonality;
- **Statistical tests features:** Statistics and p-values of common statistical tests applied to time series data as Box-Pierce, Ljung-Box, Kwiatkowski-Phillips-Schmidt-Shin (KPSS), among others;

- **Model based features:** Learned parameters from time series forecasting models, like the ETS and ARIMA.

Once the choice of a set of meta features is done we can define a meta feature extractor (MFE) $\mathcal{F} : \tau \to \mathcal{X} = \mathbb{R}^d$, where d is the number of meta features considered, defined as in Equation 3:

$$\mathcal{F}(\tau) = (g_1(\tau), g_2(\tau), \ldots, g_d(\tau)) \tag{3}$$

The last element necessary to construct the dataset $\mathcal{D}$ is the output space $\mathcal{Y}$ characterized by the meta target.

3.2 Meta Target

Within the context of group model selection, the meta target is given simply by the group that has the best average model. Firstly, fix a task τ, then define an error measure $\mathcal{E}$ that tells how far the values forecasted by a single algorithm $\mathcal{A}$ are from the actual values, the error for the group M presented in this section for the task τ is defined in the Equation 4:

$$\mathcal{E}_\tau^M = \frac{1}{k} \sum_{j=1}^{k} \mathcal{E}(\mathcal{A}_j^M, \tau) \tag{4}$$

Similarly, one defines the error for the group S with respect to the task τ, $\mathcal{E}_\tau^S$, finally the meta target Y_τ is given by the Equation 5:

$$Y_\tau = \begin{cases} 0, & \text{if } \mathcal{E}_\tau^M < \mathcal{E}_\tau^S \\ 1, & \text{otherwise} \end{cases} \tag{5}$$

The final step is to assume the existence of a fixed, although unknown, probability measure μ over $\mathcal{X} \times \mathcal{Y} = \mathbb{R}^d \times \{0, 1\}$ and we have re-framed the MTL problem as a binary classification framework as desired.

4 Experimental Setup

We collected the industrial time series to run the experiments from the M4 competition shown in the Table 1, the weekly, daily and hourly intervals were not considered due to the lack of tasks to train a binary classifier.

Those time series were already split into training and testing sets using a temporal holdout cross-validation by the organizers of the competition, so we took advantage of it to extract the meta features only from the training set using all the features available at [24], moreover the summarization measures used were the mean and the standard deviation.

The elements of the groups M and S will be referred as base learners, therefore the submissions for the competition that we considered as base learners are described in the Table 2, the first 6 correspond to the group M and the last 6 compose the group S.

Author	**Affiliation**
Trotta	Individual
Alves Santos Junior	Individual
Mukhopadhyay	University of Texas
Pełka	Czestochowa University of Technology
RNN	Benchmark
MLP	Benchmark
Theta	Benchmark
ARIMA	Standard for comparison
Damped	Benchmark
ETS	Standard for comparison
Holt	Benchmark
SES	Benchmark

Table 2: Base learners

Their point forecasts were collected from the M4 competition public repository [10] and to be consistent with the results from the latter, the same metrics were used to evaluate them on the testing set:

- **SMAPE:** Gives the deviation of the forecasted values from the actual ones in terms of percentages, it is easy to interpret despite its disadvantages [12], it can be calculated as in Equation 6.

$$\frac{2}{h}\sum_{t=n+1}^{n+h}\frac{|y_t-\hat{y}_h|}{|y_t|+|\hat{y}_t|}*100\% \tag{6}$$

- **MASE:** A scale-independ error that compares the forecasted values from any model against the one step seasonal naive forecaster using

Equation 7.

$$\frac{1}{h}\frac{\sum_{t=n+1}^{n+h}|y_t-\hat{y}_h|}{\frac{1}{n-m}\sum_{t=m+1}^{n}|y_t-y_{t-m}|} \tag{7}$$

- **OWA:** Combines both the SMAPE and the MASE into a single metric by averaging the relative SMAPE and the relative MASE with respect to the Naïve 2 method. Its calculation can be done by Equation 8.

$$\frac{1}{2}\left(\frac{\text{SMAPE}}{\text{SMAPE-Naïve2}}+\frac{\text{MASE}}{\text{MASE-Naïve2}}\right) \tag{8}$$

In the Equations 6 and 7, n is the number of observations in the training set, m is the seasonal period, h is the forecasting horizon, y_t is the actual observation and $\hat{y}_t$ is the value predicted.

We set $\mathcal{E}$ to be the OWA metric, finally after merging the meta features extracted and the meta target we have the dataset that will be used to train and validate our classifier, that will be addressed as meta learner from now on. A total of 3 meta learners will be trained, one for each interval considered, i.e, yearly, monthly and quarterly; moreover the following cleaning steps were applied to the datasets:

1. Remove features with more than 50 missing values;
2. Remove the samples with at least 1 missing value;
3. Remove constant features;
4. Remove high correlated features using Pearson rank.

In order to train the meta learner and make predictions, it was applied a stratified 10-fold cross validation and the classifier chosen was a Random Forest (RF) whose implementation is available at [5]. The RF is an ensemble model that uses the bagging technique, which consists in training each weak learner, in this case decision trees (DT), using a subset of the data containing observations sampled with replacement and features without replacement, it leads to a significant reduce in the variance of the overall classifier and the final predictions are given by voting. Moreover, a DT is a simple classifier that divides the data space recursively into hyper-cubes using the gini impurity as the criteria to find the best threshold to perform the cut, once the

tree is fully grown the predictions are given by the mean of the leaf in which the new query instance lies in [11]. Those models have an embedded feature selection which give us a numerical value that represents the importance of each feature, the number represents the frequency in which the feature was used as a threshold, this fact along with the capacity of learning non-linear functions are the reasons why the RF was chosen to be the meta learner.

We will analyse the meta learner capacity of distinguish between the two groups given the meta features using a confusion matrix and if being stuck with the meta learner recommendation is better than applying the average model of the groups for every industrial univariate time series. If the meta learner succeed we will be able to tell apart the time series in which the ML models tend to fail by analysing the most important features according to the RF over all 10 iterations of the cross validation.

The Algorithms 1 and 2 illustrate the steps presented in this section, the RF hyperparameters chosen to train the model in the RECOMMEND procedure were: "number of estimators" set to 100, "max depth" to 5 and "min sample split" to 5, moreover the model used balanced class weights.

Algorithm 1 Constructing the data to train the meta learner

procedure MAKEDATASET(M4, M, S, $\mathcal{F}$, t)
    $\mathcal{T} \leftarrow filter(\text{M4}, \text{“industry”}, t)$ ▷ Select the M4 industry data from t interval
    $\mathcal{E} \leftarrow$ OWA
    $\mathcal{D} \leftarrow \emptyset$
    for τ **in** $\mathcal{T}$ **do**
        $X_\tau \leftarrow$ Extracted meta features using $\mathcal{F}$ as in Equation 3
        $\mathcal{E}_\tau^M, \mathcal{E}_\tau^S \leftarrow$ Calculates the erros as in Equation 4
        $Y_\tau \leftarrow$ Calculates the Meta target as in Equation 5
        $\mathcal{D} \leftarrow \mathcal{D} \cup \{(X_\tau, Y_\tau)\}$
    return $\mathcal{D}$

Algorithm 2 Recommending the best group of techniques

procedure RECOMMEND($\mathcal{D}$)
  $\mathcal{D}' \leftarrow$ Dataset after applying cleaning steps 1-4
  Folds $\leftarrow$ Stratified 10-Folds on $\mathcal{D}'$
  $\mathcal{R} \leftarrow \emptyset$ ▷ Meta Learner Recommendations
  $\mathcal{I} \leftarrow \emptyset$ ▷ Feature Importances
  for $\{\mathcal{D}_{train}, \mathcal{D}_{test}\}$ **in** Folds **do**
    RF $\leftarrow$ Trained Random Forest on $\mathcal{D}_{train}$
    $\mathcal{I} \leftarrow \mathcal{I} \cup \{$RF learned feature importances$\}$
    $Y \leftarrow$ RF predictions on $\mathcal{D}_{test}$
    for y **in** Y **do**
      if $y = 0$ **then**
        $\mathcal{R} \leftarrow \mathcal{R} \cup \{M\}$
      else
        $\mathcal{R} \leftarrow \mathcal{R} \cup \{S\}$
  return $\mathcal{R}, \mathcal{I}$

5 Experimental Results

The group S had a lower error across the majority of the time series considered as shown in the Figure 1, with more than 80% in all three periods as expected due to the findings of the M4 Competition.

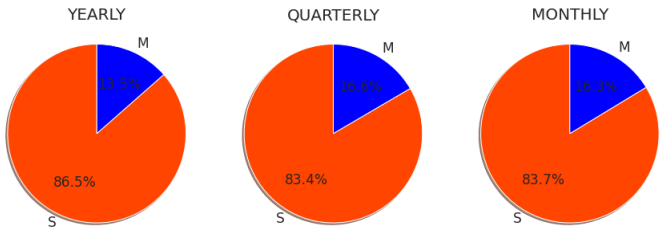

Figure 1: Meta Target Distribution

The projection of the standardized data onto its 2 principal components shows that the data is quite noise with the presence of some outliers and more, there are no clusters separating the classes that can be spotted visually. The time series lied onto a single cluster regarding the meta target as the Figure

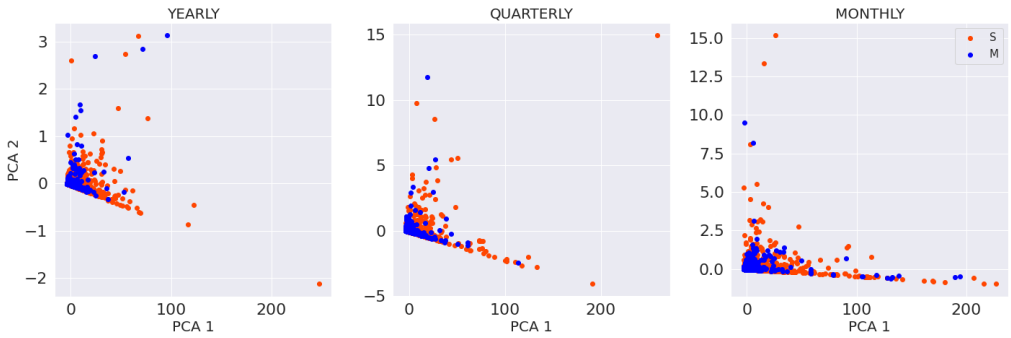

Figure 2: Meta Features Principal Components

2 shows.

After comparing the meta learner recommendations with the true labels, the confusion matrix in the Figure 3 shows that it had a hard time telling the classes apart, even forcing the RF to not ignore the minority class we can see the most common mistake committed was to output the label S when the true label is actually M in all cases. Moreover, although with a small margin, the yearly was the only period in which the number of true positives were greater than the number of false negatives, despite that, the framework behaved similarly regarding the period considered. The respective balanced accuracy scores for yearly, quarterly and monthly periods were 65.13%, 53.23% and 53.94% which shows that the meta learner recommendations were a little better than guessing the label.

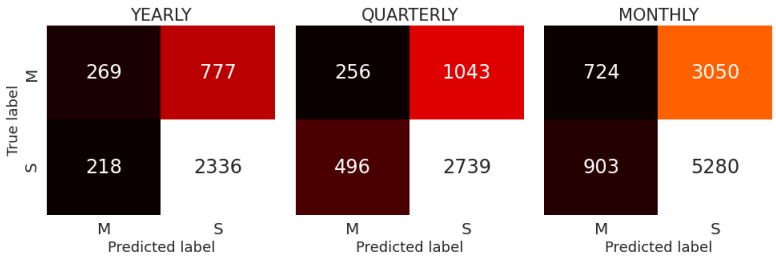

Figure 3: Recommendations Confusion Matrix

The Figure 4 shows the first 10 most important features to make the decisions according to the RF algorithm, the scores are the mean over all

10 iterations of the stratified k-fold. Even though the meta learner results were similar for all cases the most relevant features in each period changed considerably, in the yearly period the general and randomized groups of features dominated the top 10, moreover in the quarterly we can see that the autocorrelation along with the global statistics were more relevant, finally in the monthly data the general characteristics and the local statistics had a bigger importance. The detailed description of the meta features is available at [24].

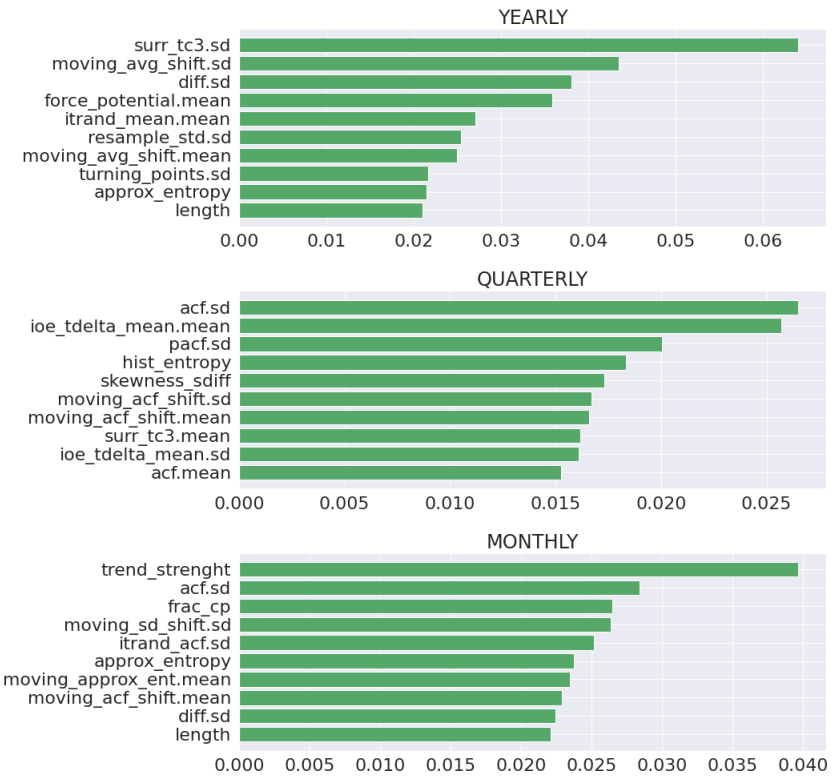

Figure 4: Meta Feature Importances according to the RF

The errors from the group recommended by the meta learner were compared with the errors given by the application of each group 100% of the time despite the time series, the Table 3 shows the average errors over all time series in each interval of time. We can see that the statistical models outperformed the meta learner recommendations in roughly all cases and the

performance of the pure machine learning models were the worst confirming one of the findings of the M4 Competition for the case of industrial time series as well.

	YEARLY			QUARTERLY			MONTHLY		
Group	**SMAPE**	**MASE**	**OWA**	**SMAPE**	**MASE**	**OWA**	**SMAPE**	**MASE**	**OWA**
M	0.29	5.58	2.41	0.13	1.71	1.80	0.18	1.50	1.74
S	**0.19**	**3.50**	**1.11**	**0.10**	**1.17**	**1.02**	**0.14**	**1.02**	**1.00**
RF	0.21	3.77	1.30	**0.10**	1.27	1.18	0.15	1.13	1.14

Table 3: Average Errors over all Time Series

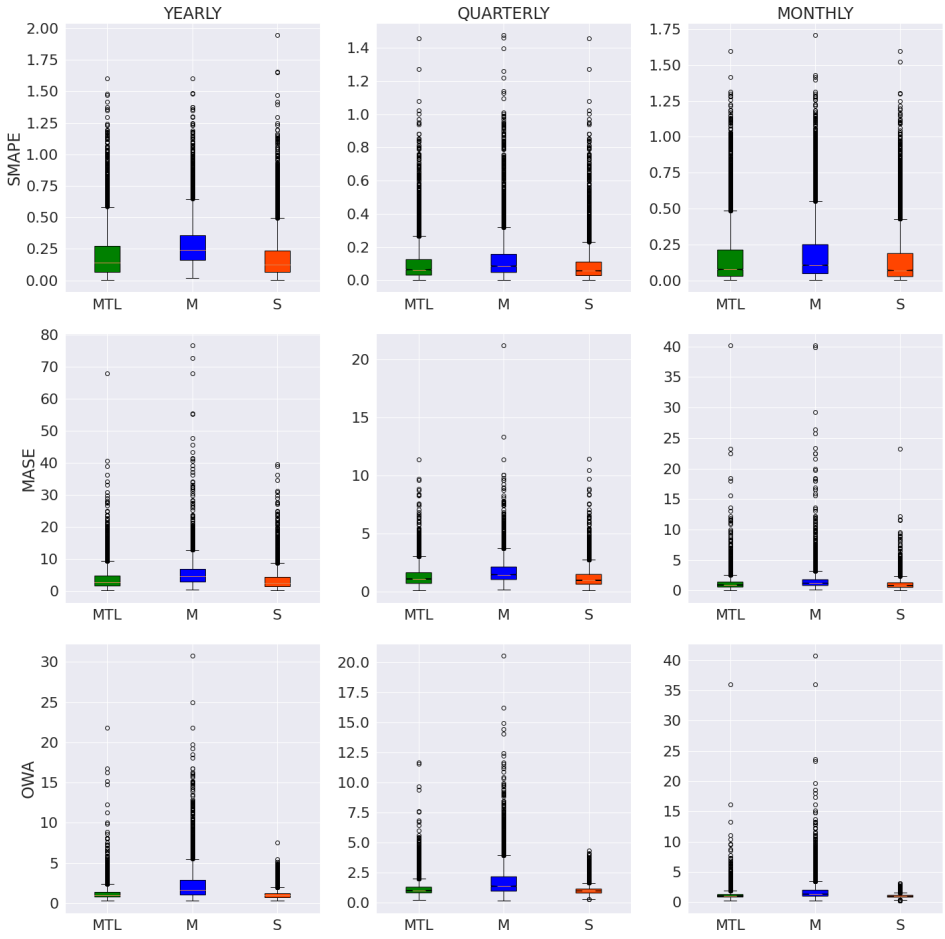

Figure 5: Distribution of the Metrics for each period

Furthermore, the Figure 5 shows the distribution of each metric considered for all 3 periods, it is interesting to note that when it comes to the SMAPE and MASE the box plots from the RF and the group M are quite similar for all periods but it changes abruptly when one considers the OWA metrics, which was the one used to define the meta target rule. The Kolmogorov-Smirnov (KS) statistic on 2 samples was chosen to tell if the errors from 2 techniques are drawn from the same continuous distribution, the fact that it does not make any assumption about the underlying distribution turns it into a quite reliable test for our purposes. The approaches will be compared pairwise for all metrics and all time intervals, in total we will have $\#approaches \times \#metrics \times \#time\ intervals$=27 comparisons. For instance, we could choose the yearly interval, the SMAPE metric and the pair (M, RF) then the first sample will be composed by the SMAPE's of the group M for all time series in the yearly subset and the second sample by the ones from the RF, finally we apply the KS test to check if those approaches output similar errors for that particular metric and time interval using a confidence interval of 5%. At the end, we could reject the null hypothesis, 2 independent samples are drawn from the same continuous distribution, for all cases.

6 Discussion and Conclusion

The results from the previous section permit us to conclude that the meta learner failed to distinguish the time series better forecasted by pure ML models and the ones in which the statistical based models had a lower error. We believe that the industrial time series from the M4 Competition have not enough information to reach our goals, i.e., they resemble no variety when it comes to the relationship between the characterization using meta features and the meta target rule. This fact along with the huge class unbalanced led us to poor results since the best possible scenario was to output the group S 100% of the time regardless of the time series considered.

This work confirmed some of the findings of the M4 Competition, when it comes to the application of pure ML models to forecast univariate time series, that in general they tend to be outperformed by a classical statistical model. However, this does not rule out the fact that a ML model can be, indeed, the best option in some cases, the M5 Competition [18], that was not considered here since it was based on only one domain, showed that pure ML models outperformed classical statistical approaches to forecast groups

of intermittent retail sales data.

Those facts reinforce another finding of the M4 which claims that combining both statistical and ML models into a single one is a safer approach, because it minimizes the chances of choosing the worst model to forecast a time series and more, they avoid the question of whether a model is better since they take into account the information given by all the methods. Thus, the development of new techniques to combine the strengths of both ML and statistical models will be essential to the evolution of the time series forecasting task in the industrial setting.

7 Future Work

During the development of this project, we tested several ML approaches to forecast univariate time series as alternatives to the ones submitted to the M4 Competition, a particular approach based on the decomposition of the time series using STL which sums up to the modelling of the seasonal adjusted data using a ML model and the seasonal component using the Seasonal Naive method led us to a paper submission, named "Seasonal-Trend decomposition based on Loess + Machine Learning: Hybrid Forecasting for Monthly Univariate Time Series", to the International Joint Conference on Neural Network (IJCNN) which is, by the time this report is being written, under analysis.

The results showed that the STL as a preprocessing step tends to boost the predictive performance of a classical ML model when the decomposition gives uncorrelated residuals and the time series considered has a strong seasonal pattern. In future works, we want to develop and analyse other techniques based on ML applied to time series data, and hopefully design a general framework for the application of any ML model to forecast time series that is competitive against the statistical state-of-art techniques highly used in the industry.

Acknowledgments

I want to thank Moisés Rocha (Institute of Mathematics and Computer Science - University of São Paulo) for providing an immense support throughout this work and the FAFQ (Fundação de Apoio à Física e à Química) for pro-

viding financial aid.

References

[1] Kasun Bandara, Christoph Bergmeir, and Slawek Smyl. Forecasting across time series databases using long short-term memory networks on groups of similar series. 10 2017.

[2] Jocelyn Barker. Machine learning in m4: What makes a good unstructured model? *International Journal of Forecasting*, 36(1):150–155, 2020. M4 Competition.

[3] Gianluca Bontempi, Souhaib Ben Taieb, and Yann-Aël Le Borgne. *Machine Learning Strategies for Time Series Forecasting*, volume 138. 01 2013.

[4] Pavel Brazdil, Christophe Giraud-Carrier, Carlos Soares, and Ricardo Vilalta. *Metalearning - Applications to Data Mining.* 01 2009.

[5] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

[6] Alket Cecaj, Marco Lippi, Marco Mamei, and Franco Zambonelli. Comparing deep learning and statistical methods in forecasting crowd distribution from aggregated mobile phone data. *Applied Sciences*, 10(18), 2020.

[7] Vitor Cerqueira, Luís Torgo, Fábio Pinto, and Carlos Soares. Arbitrage of forecasting experts. *Mach. Learn.*, 108(6):913–944, June 2019.

[8] Vitor Cerqueira, Luís Torgo, and Carlos Soares. Machine learning vs statistical methods for time series forecasting: Size matters. 09 2019.

[9] Robert B Cleveland, William S Cleveland, and Irma McRae, Jean E adn Terpenning. Stl: A seasonal-trend decomposition procedure based on loess. *Journal of Official Statistics*, 6(1):3–73, 1990.

[10] M Forecasting Competitions and Electra Skepetari. M4-methods. https://github.com/Mcompetitions/M4-methods, 2018.

[11] Katti Faceli, Ana Carolina Lorena, João Gama, and André Carlos Ponce de Leon Ferreira de Carvalho. *Inteligência artificial: uma abordagem de aprendizado de máquina*. LTC, 2011.

[12] Rob Hyndman. Another look at forecast accuracy metrics for intermittent demand. *Foresight: The International Journal of Applied Forecasting*, 4:43–46, 01 2006.

[13] Robin John Hyndman and George Athanasopoulos. *Forecasting: Principles and Practice*. OTexts, Melbourne, Australia, 3rd edition, 2021.

[14] Kyoung-jae Kim. Financial time series forecasting using support vector machines. *Neurocomputing*, 55:307–319, 09 2003.

[15] Shaohui Ma and Robert Fildes. Retail sales forecasting with meta-learning. *European Journal of Operational Research*, 288(1):111–128, 2021.

[16] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. Statistical and machine learning forecasting methods: Concerns and ways forward. *PLOS ONE*, 13(3):1–26, 03 2018.

[17] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. The m4 competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting*, 36(1):54–74, 2020. M4 Competition.

[18] Spyros Makridakis, Evangelos Spiliotis, and Vassilis Assimakopoulos. The m5 accuracy competition: Results, findings and conclusions. 10 2020.

[19] Pablo Montero-Manso, George Athanasopoulos, Rob J. Hyndman, and Thiyanga S. Talagala. Fforma: Feature-based forecast model averaging. *International Journal of Forecasting*, 36(1):86–92, 2020. M4 Competition.

[20] Antonio Parmezan and Gustavo Batista. A study of the use of complexity measures in the similarity search process adopted by knn algorithm for time series prediction. pages 45–51, 12 2015.

[21] Antonio Parmezan, Huei Lee, and Feng Wu. Metalearning for choosing feature selection algorithms in data mining: Proposal of a new framework. *Expert Systems with Applications*, 75, 01 2017.

[22] Antonio Rafael Sabino Parmezan, Vinicius M.A. Souza, and Gustavo E.A.P.A. Batista. Evaluation of statistical and machine learning models for time series prediction: Identifying the state-of-the-art and the best conditions for the use of each model. *Information Sciences*, 484:302–337, 2019.

[23] Artemios-Anargyros Semenoglou, Evangelos Spiliotis, Spyros Makridakis, and Vassilios Assimakopoulos. Investigating the accuracy of cross-learning time series forecasting methods. *International Journal of Forecasting*, 2020.

[24] Felipe Siqueira. Estrutura unificada para extração de meta-características de séries temporais unidimensionais, 2020. Monografia final de conclusão de curso (Bacharel em Computação), Universidade de São Paulo, ICMC, São Carlos, Brasil.

[25] V. N. Vapnik. An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, 10(5):988–999, 1999.