Journal of Physics: Conference Series

PAPER • OPEN ACCESS

Short-Term Load Forecasting in a microgrid environment: Investigating the series-specific and cross-learning forecasting methods

Short-Term Load Forecasting in a microgrid environment: Investigating the series-specific and cross-learning forecasting methods

Evgenii Genov[1], Stefanos Petridis[2], Petros Iliadis[2], Nikos Nikopoulos[2], Thierry Coosemans[1], Maarten Messagie[1], Luis Ramirez Camargo[1]

[1]Electric Vehicle and Energy Research Group (EVERGI), Mobility, Logistics and Automotive Technology Research Centre (MOBI), Department of Electrical Engineering and Energy Technology, Vrije Universiteit Brussel, Brussels, Belgium;
[2]Centre for Research and Technology Hellas - CERTH: Athens, GR

E-mail: `evgenii.genov@vub.be`

Abstract. A reliable and accurate load forecasting method is key to successful energy management of smart grids. Due to the non-linear relations in data generating process and data availability issues, load forecasting remains a challenging task. Here, we investigate the application of feed forward artificial neural networks, recurrent neural networks and cross-learning methods for day-ahead and three days-ahead load forecasting. The effectiveness of the proposed methods is evaluated against a statistical benchmark, using multiple accuracy metrics. The test data sets are high resolution multi-seasonal time series of electricity demand of buildings in Belgium, Canada and the UK from private measurements and open access sources. Both FFNN and RNN methods show competitive results on benchmarking datasets. Best method varies depending on the accuracy metric selected. The use of cross-learning in fitting a global RNN model has an improvement on the final accuracy.

1. Introduction

A reliable and accurate demand forecast can enhance the operation, scheduling, management and control of energy assets on a microgrid. The forecast enables matching the electricity supply with demand more efficiently with an outlook for the future consumption. In the context of a microgrid, the output of the forecast facilitates new business opportunities, such as demand response and peer-to-peer trading. On the individual building level, the non-aggregated user behaviour shows a highly dynamic and stochastic nature making forecasting a challenging task. Feed Forward neural networks (FFNN) are widely utilized in load forecasting, with ensemble and cascade models particularly finding success (see e.g. [1]). Similarly, Deep learning (DL) methods gain popularity to address this challenge due to its ability to identify relevant features in a high-dimensional feature space. Despite of being computationally intensive, DL methods, like recurrent neural networks (RNN), specifically Long Short-term memory (LSTM) typologies, are becoming more accessible and gaining more attention among forecast professionals. RNN architectures use cases exist e.g. for day-ahead forecast on real datasets both for aggregate loads [2] and individual households [3]. In this paper we compare FFNN and RNN approaches for short-term load forecasting (STLF) within a 3-day and day-ahead horizons on the individual building level. Furthermore, this paper studies the ability of networks at cross-learning (CL). The term describes models' ability to learn from training across multiple time series in an entire data set. This allows to reduce the effect of uncertainty observed for individual users [4]. In the context of a microgrid consumption time series may be available in a limited amount, e.g. new buildings or newly-connected digital meters. In this study, we explore the benefits of CL approach. A global model is trained across a concatenated input of entire data set and evaluated on individual time series independently.

2. Methodology

2.1. Data sets

In this study electric load readings sourced from three locations are used. Two of the datasets, London and British Columbia, are accessed from open access sources. The third set, Green Energy park (GEP), is collected from private sources. A summary of key characteristics is presented in Table 1.

Dataset	London smart meter	British Columbia	Green Energy Park (GEP)
Country	UK	Canada	Belgium
No. of time series	90	28	2
Frequency	30 min	1 h	15 min
Min.Length	5737	4175	15984
Max.Length	19667	30240	44448
missing values [%]	0 - 1.72 %	0 - 1.51 %, 11.93% *	0.01 - 6 %

* Building B16 has an anomaly number of missing values

Table 1: Data Description

- London smart meter data [5]: A group of 90 buildings was selected from a bigger data set of 5,567 London households with the corresponding records of energy consumption. The original set contains information on social-economic demographic, conveyed with CACI ACORN classification [6]. For overall representation of the final selection, 5 time series were picked within each of 18 ACORN groups.
- British Columbia [7]: Hourly dataset, representing 27 residential houses in British Columbia, Canada.
- Green Energy Park (GEP): These time series correspond to multi-yearly electricity consumption records for 2 commercial buildings in a living-lab test site in Zelik, Brussels.

2.2. Pre-processing

Before using data in the training process, we apply a suitable pre-processing and feature generation. The type of pre-processing applied varies slightly between the algorithms. However, there are ideas common to both ANN and RNN approaches.

Initially, the reported missing values were filled using linear interpolation. In order to evaluate the networks on the same temporal granularity, the measurements of non-hourly frequency were re-sampled to hourly intervals with a summation (all values in kWh). Then, standard min-max normalization was utilized to scale the input and output data. An advantage of min-max scaler is that it provides uniform [0,1] scale across all features. The last step includes the train-test split. As time-series samples are not independent, the data cannot be split randomly. The data split follows the sequence logic. From the end of the series, half a year (= 4380 hours) is reserved for test set. In time-series that hold less than a full year span of data, 20% of samples from the end of the series are reserved for test set.

2.3. Implementation

As the problem at hand is a multi-step ahead forecasting problem, a Multiple-Input Multiple-Output (MIMO) strategy is employed. An advantage of such approach is incorporating the inter-dependencies between time steps within the output window. Furthermore, MIMO strategy makes it possible to avoid potential error build-up that can occur when employing the recursive strategy, which is a sequence of one-step-ahead forecasts.

Adding calendar features boosts the network ability to learn seasonal patterns. Although some studies take an additional step in pre-processing and deseasonalize the time series data, according to [8], learning the seasonal patterns from calendar features becomes viable within the network for homogeneous series which is the case of electric load of individual buildings. The following external features were engineered:

(i) Time of day, day of year as continuous features. The features are first converted to a fractional form in $[1, 0]$. Sin and Cos transformations were applied in order to preserve cyclical properties of the features.

(ii) Day of week, one-hot encoded.

(iii) Working day ticker. This is a binary feature indicating bank holidays and weekends in the countries of interest.

All models were developed using Tensorflow version 2.4.0, an open-source deep-learning platform. All necessary scripts are programmed in Python. Two lagged features are added to the input: consumption in the 72 hours and consumption in previous week for the forecasted times.

In order to limit model overfitting, we used an early stopping technique. The models are trained until the maximum number of training epochs is reached or the validation error grows. A delay in number of epochs which have the validation error growing is a patience parameter. The best epoch with smallest validation error is saved and the model is retrained on full training data with no validation split. This is done to address an issue with split into validation and train data, as highlighted in [9].

Non-model parameters set in the design of the architecture are referred as hyperparameters. The hyperparameter configuration includes the following parameters: number of layers, number of cells, learning rate, batch size and dropout rate (applicable if more than one layer). These parameters can be tuned, to achieve the best possible result. For tuning the hyperparameters of the two-stage FFNN model, an open-source Python library, keras-tuner, was used. Each of the six networks included in the model were trained and tuned separately from one another. Since the RNN training process is computationally costly, the hyperparameter optimization was run only for GEP dataset. Therefore, the final hyperparameter configuration is sub-optimal and based on a series of trials and domain knowledge. The loss function used for fitting the model on the training data optimizes the mean squared error (MSE) metric. The resulted hyperparameter configuration includes a hidden layer of 64 cells, a learning rate of 1e-3 with 1500 batch size and no dropout rate.

2.4. Evaluation

The efficacy of the proposed algorithms needs to be assessed by comparing their results with those of an already implemented forecasting technique. For this purpose, an Auto Regressive Integrated Moving Average (ARIMA) with exogenous parameters (ARIMAX) model is created, to provide the benchmark results.

ARIMA models have been used extensively in various forecasting scenarios, not only as a benchmark, but also, as a well-established statistical method to address time series analysis problems. Adding exogenous parameters to the original ARIMA algorithm produces more accurate results, since the time series' behavior is highly dependent on external factors such as the day of the week or if it is a working day or not.

The performance is evaluated using the Root Mean Squared Error (RMSE) and the Mean Average Percentage Error (MAPE), described by the following equations:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=0}^{N-1}\frac{1}{n_O}\sum_{t=0}^{n_O-1}(\hat{y}_i[t]-y_i[t])^2} \tag{1}$$

$$\mathrm{MAPE} = \frac{100\%}{N}\sum_{i=0}^{N-1}\frac{1}{n_O}\sum_{t=0}^{n_O-1}\left|\frac{y_t-\hat{y}_t}{y_t}\right| \tag{2}$$

where N is the number of samples in test set, n_0 is the size of the horizon, i.e. number of steps to be forecasted in each sample, and y_i, $\hat{y}_i$ the actual and forecasted value, respectively.

3. Results

The time-series of electrical load in individual households show a high degree of uncertainty, non-linearity and anomaly days. The LSTM RNN model was trained both using individual buildings data and globally, using the concatenated input from the entire dataset.

Figure 1 shows a small sample of the time series produced by the models in contrast with the actual values for day-ahead and 3 days-ahead forecasting. It is observed that the forecasted values follow the measured ones, producing fairly accurate results. ARIMAX outputs the same results on the both day-ahead and three-days ahead horizons. Other models show a better fit at day-ahead forecasting, since there is less uncertainty and more recent lags accessible. The underestimation of peaks is attributed to regularization to avoid overfitting and the use of mean squared error as a loss function. The forecast example shows that within the observed interval RNN models output a smoother time series.

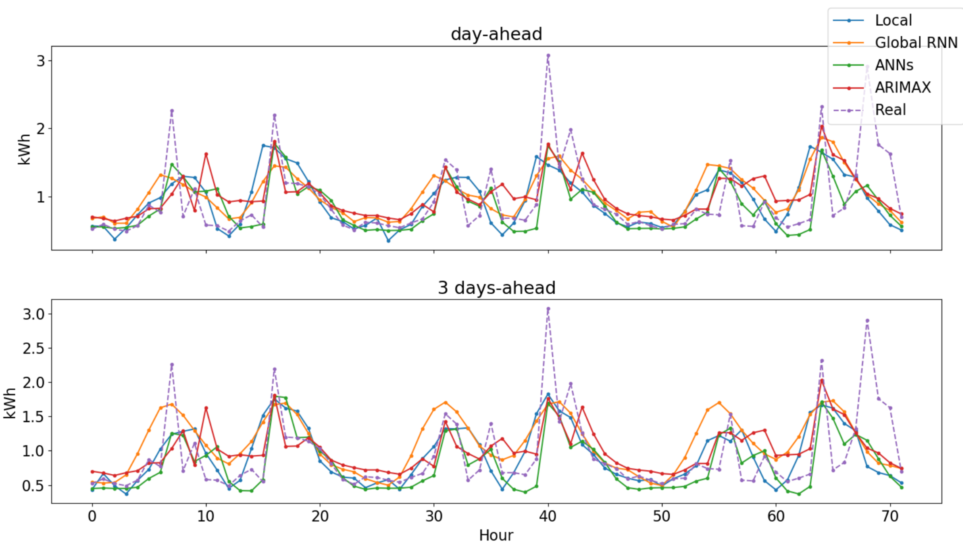

Figure 1. Time series produced by the models for day-ahead(up) and 3 days-ahead(down) forecasting

Table 2 summarizes the accuracy metrics for forecasts produced at day-ahead and 3 days-ahead horizon. The results are highlighted with a gradient colour heat map. The gradient ranges from green (least error) to red (biggest error). From the results we can conclude that both FFNN and LSTM models show superiority over the ARIMAX benchmark in London smart meter and British Columbia sets. In GEP dataset, consisting of only two buildings, the global model has the highest error on 3 days-ahead scale. There, the FFNN models outperform the RNN models on day-ahead forecast. RNN trained in a series-by-series fashion provide a better accuracy forecast on 3 days-ahead horizon.

We see that the accuracy evaluation presents no clear-cut winner regarding accuracy of the forecast. In terms of RMSE, RNN Global and series-specific RNN match or slightly overperform the forecast produced by FFNN on both British Columbia and London sets. However, RNN scores lower MAPE only on British Columbia dataset, while on London set FFNN shows better accuracy in terms of MAPE. As noted in [10], the MAPE metric has a disadvantage of penalizing negative errors more than positive. Due to this fact, we can conclude that RNN models tend to underestimate more than overestimate the actual readings.

There is indication that the LSTM models benefit from learning cross-series information. Globally trained RNN models outperform the individual RNN models, particularly in terms of the MAPE metric. It can be seen on London Smart Meter and British Columbia datasets. For instance, global RNN on average has a 10% better MAPE than RNN trained locally across entire dataset. It is evident that there are private cases where the generalization power of the global model hinders the forecasting performance. For example, the global model gives the lowest accuracy for an ACORN-P segment, which, according to CACI [6], represents 'struggling neighbourhoods'.

		day-ahead							
		RMSE				MAPE			
		RNN	RNN Global	FFNN	ARIMAX	RNN	RNN Global	FFNN	ARIMAX
London Smart meter	ACORN-A	0.46	0.434	0.484	0.601	69,542	54,776	68,492	123,287
	ACORN-B	0.512	0.517	0.535	0.548	64,494	61,127	52,679	78,338
	ACORN-C	0.269	0.273	0.288	0.394	85,621	84,365	64,112	117,567
	ACORN-D	0.46	0.471	0.463	0.483	62,720	57,429	50,121	72,448
	ACORN-E	0.242	0.238	0.243	0.252	134,755	120,856	114,627	152,151
	ACORN-F	0.224	0.224	0.229	0.235	56,098	57,438	54,230	70,755
	ACORN-G	0.41	0.411	0.41	0.436	92,078	78,589	67,330	121,116
	ACORN-H	0.384	0.378	0.374	0.394	105,685	103,952	90,553	124,893
	ACORN-I	0.354	0.346	0.36	0.38	66,234	56,590	56,396	74,337
	ACORN-J	0.168	0.168	0.163	0.195	65,361	62,499	49,599	83,308
	ACORN-K	0.35	0.352	0.348	0.428	66,159	57,969	57,462	124,808
	ACORN-L	0.23	0.23	0.232	0.235	65,721	58,537	50,735	68,487
	ACORN-M	0.401	0.411	0.407	0.413	78,781	75,555	68,316	85,020
	ACORN-N	0.217	0.22	0.219	0.35	68,134	66,366	67,561	142,309
	ACORN-O	0.264	0.265	0.276	0.329	69,375	70,508	67,832	121,908
	ACORN-P	0.332	0.339	0.312	0.306	255,648	277,169	215,068	254,254
	ACORN-Q	0.218	0.223	0.223	0.224	67,121	67,015	54,930	59,106
	ACORN-U	0.31	0.312	0.341	0.322	76,031	69,075	85,255	93,171
	Average	0.322	0.323	0.328	0.362	86,087	82,212	74,183	109,292
British Columbia	B1	0.537	0.543	0.575	0.598	28,466	30,774	25,155	49,939
	B2	0.295	0.3	0.288	0.399	51,693	46,267	40,881	137,909
	B3	0.524	0.558	0.529	0.57	43,790	42,311	42,879	57,288
	B4	0.543	0.548	0.603	0.824	31,571	29,344	29,530	41,363
	B5	0.677	0.672	0.753	0.725	91,217	77,645	67,909	109,097
	B6	0.226	0.233	0.234	0.248	53,328	56,322	52,627	80,159
	B7	0.431	0.436	0.428	0.539	47,339	46,453	36,821	50,280
	B8	0.539	0.51	0.564	0.578	62,857	48,842	55,861	78,052
	B9	0.448	0.45	0.511	0.541	48,990	46,874	40,929	84,668
	B10	0.507	0.515	0.514	0.587	56,100	50,983	44,010	48,972
	B11	0.398	0.435	0.42	0.497	57,736	63,460	52,784	119,812
	B12	0.256	0.264	0.266	0.275	33,790	35,486	35,714	49,430
	B13	0.647	0.664	0.668	1,043	37,319	37,095	27,642	71,450
	B14	0.697	0.708	0.694	0.749	28,952	30,530	25,515	29,997
	B15	1,390	1,478	1,220	1,329	118,479	106,533	73,226	116,123
	B16	0.47	0.45	0.547	0.516	66,276	62,981	137,745	170,478
	B17	0.426	0.42	0.503	0.469	150,107	172,407	256,805	202,169
	B18	1,803	1,887	2,138	1,940	64,130	64,477	70,532	136,026
	B19	0.49	0.502	0.483	0.583	17,052	16,772	16,194	19,746
	B20	0.53	0.523	0.543	0.628	28,309	27,285	23,725	32,079
	B21	0.226	0.213	0.382	0.431	74,589	55,685	188,879	235,076
	B22	0.51	0.519	0.431	0.517	136,012	118,405	63,356	129,826
	B23	0.619	0.638	0.601	0.725	99,425	107,071	90,924	117,232
	B24	0.472	0.468	0.487	0.481	69,367	66,483	50,522	70,497
	B25	0.604	0.617	0.605	0.612	48,483	52,421	39,353	53,868
	B26	0.202	0.187	0.179	0.218	34,522	28,511	34,576	48,189
	B27	1,127	0.962	0.694	1,018	173,482	164,678	166,569	243,689
	Average	0.578	0.582	0.587	0.653	64,940	62,448	66,321	95,682
Green Energy Park	GEP1	18,344	13,871	13,670	13,785	46,633	28,052	32,677	32,052
	GEP4	9,383	8,782	5,008	6,049	23,898	20,762	16,681	22,430
	Average	13,864	11,327	9,339	9,917	35,265	24,407	24,679	27,241

		3 days ahead							
		RMSE				MAPE			
		RNN	RNN Global	FFNN	ARIMAX	RNN	RNN Global	FFNN	ARIMAX
London Smart meter	ACORN-A	0.489	0.457	0.495	0.601	78,295	55,474	72,831	123,287
	ACORN-B	0.519	0.529	0.55	0.548	69,365	63,919	54,822	78,338
	ACORN-C	0.284	0.279	0.286	0.394	99,675	89,678	62,092	117,567
	ACORN-D	0.474	0.478	0.473	0.483	65,181	57,558	50,283	72,448
	ACORN-E	0.245	0.244	0.247	0.252	138,631	119,470	117,558	152,151
	ACORN-F	0.224	0.227	0.234	0.235	59,095	53,106	55,536	70,755
	ACORN-G	0.411	0.412	0.414	0.436	91,122	79,842	68,152	121,116
	ACORN-H	0.394	0.395	0.379	0.394	110,059	103,905	90,675	124,893
	ACORN-I	0.359	0.35	0.365	0.38	67,137	58,984	57,828	74,337
	ACORN-J	0.168	0.166	0.164	0.195	63,159	58,791	49,821	83,308
	ACORN-K	0.356	0.352	0.348	0.428	69,696	58,715	56,760	124,808
	ACORN-L	0.232	0.228	0.232	0.235	67,332	55,894	50,466	68,487
	ACORN-M	0.412	0.41	0.409	0.413	82,520	72,857	68,003	85,020
	ACORN-N	0.22	0.217	0.22	0.35	69,795	63,862	69,212	142,309
	ACORN-O	0.272	0.262	0.277	0.329	73,467	67,797	67,600	121,908
	ACORN-P	0.366	0.34	0.328	0.306	274,547	252,460	234,322	254,254
	ACORN-Q	0.221	0.218	0.224	0.224	69,072	62,089	55,234	59,106
	ACORN-U	0.318	0.322	0.342	0.322	79,077	74,652	88,458	93,171
	Average	0.331	0.327	0.333	0.362	90,401	80,503	76,092	109,292
British Columbia	B1	0.547	0.546	0.578	0.598	29,287	32,219	25,740	49,939
	B2	0.296	0.301	0.29	0.399	49,849	48,749	41,849	137,909
	B3	0.534	0.572	0.542	0.57	45,763	48,570	45,077	57,288
	B4	0.557	0.548	0.61	0.824	31,934	30,945	30,462	41,363
	B5	0.683	0.674	0.76	0.725	80,301	85,171	68,514	109,097
	B6	0.233	0.237	0.245	0.248	55,274	58,252	62,753	80,159
	B7	0.428	0.434	0.427	0.539	45,479	47,443	35,978	50,280
	B8	0.565	0.524	0.56	0.578	68,682	53,419	55,959	78,052
	B9	0.479	0.495	0.562	0.541	54,905	52,795	44,601	84,668
	B10	0.507	0.525	0.516	0.587	56,803	57,328	44,427	48,972
	B11	0.406	0.436	0.424	0.497	54,223	65,170	53,565	119,812
	B12	0.279	0.279	0.301	0.275	38,089	38,666	48,762	49,430
	B13	0.65	0.665	0.677	1,043	37,530	40,597	27,846	71,450
	B14	0.701	0.707	0.71	0.749	28,032	31,534	26,001	29,997
	B15	1,470	1,546	1,276	1,329	141,745	126,514	80,671	116,123
	B16	0.567	0.45	0.58	0.516	80,659	65,516	160,827	170,478
	B17	0.43	0.418	0.49	0.469	166,083	199,280	276,603	202,169
	B18	1,830	1,848	2,188	1,940	66,948	67,184	72,927	136,026
	B19	0.504	0.512	0.477	0.583	17,845	17,297	16,023	19,746
	B20	0.554	0.525	0.555	0.628	29,089	28,088	25,075	32,079
	B21	0.237	0.22	0.386	0.431	92,754	58,963	195,625	235,076
	B22	0.508	0.525	0.444	0.517	135,704	140,458	61,703	129,826
	B23	0.645	0.648	0.609	0.725	121,718	115,795	94,348	117,232
	B24	0.475	0.474	0.485	0.481	70,332	73,655	50,904	70,497
	B25	0.604	0.614	0.605	0.612	51,116	53,102	38,763	53,868
	B26	0.19	0.186	0.179	0.218	25,828	29,464	34,983	48,189
	B27	1,113	1,009	0.712	1,018	166,847	189,332	169,454	243,689
	Average	0.592	0.59	0.6	0.653	68,253	68,722	69,979	95,682
Green Energy Park	GEP1	16,561	13,306	13,670	13,785	44,642	29,362	32,677	32,052
	GEP4	8,813	9,042	5,008	6,049	23,784	23,616	16,681	22,430
	Average	12,687	11,174	9,339	9,917	34,213	26,489	24,679	27,241

Figure 2. mean RMSE and MAPE results

4. Conclusions

From our experiments we conclude that there is a notable difference in performance of models between day-ahead and 3 days-ahead predicting horizons. FFNN results come near to metrics of forecast produced by individual and global RNN, while outscoring them in terms of MAPE.

It was observed that CL method utilizing a global RNN model show competitive results. CL methods are advantageous for applications in microgrids, as the new assets and new measuring units can be integrated into the forecasting framework without a drop in performance. Additionally, the training of such networks requires less time than training multiple models in a series-by-series fashion.

The generalized global parameters may not successfully be applied to individual requirements of outlier time series. Therefore, hybrid and ensemble type architectures show promise and need further research for demand forecasting applications.

Acknowledgments

This research was partly funded by the H2020 RENAISSANCE project (grant number 824342) as well as by VLAIO project MAMUET (grant number HBC.2018.0529)

References

[1] Sina Ardabili, Amir Mosavi, and Annamária R. Várkonyi-Kóczy. Advances in Machine Learning Modeling Reviewing Hybrid and Ensemble Methods. *Lecture Notes in Networks and Systems*, 101:215–227, 2020. ISSN 23673389. doi: 10.1007/978-3-030-36841-8_21. URL www.preprints.org.

[2] Filippo Maria Bianchi, Enrico Maiorino, Michael C Kampffmeyer, Antonello Rizzi, and Robert Jenssen. An overview and comparative analysis of recurrent neural networks for short term load forecasting. *arXiv preprint arXiv:1705.04378*, 2017.

[3] Kasun Bandara, Christoph Bergmeir, and Slawek Smyl. Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach. *Expert Systems with Applications*, 140, 2020. ISSN 09574174. doi: 10.1016/j.eswa.2019.112896.

[4] Artemios Anargyros Semenoglou, Evangelos Spiliotis, Spyros Makridakis, and Vassilios Assimakopoulos. Investigating the accuracy of cross-learning time series forecasting methods. *International Journal of Forecasting*, 2021. ISSN 01692070. doi: 10.1016/j.ijforecast.2020.11.009.

[5] UK Power Networks. SmartMeter Energy Consumption Data in London Households, 2015.

[6] CACI Limited. The Acorn User Guide. *The Consumer Classification*, 2014.

[7] Stephen Makonin. Hue: The hourly usage of energy dataset for buildings in british columbia. Technical report, Simon Fraser University, 2018.

[8] Kasun Bandara, Christoph Bergmeir, and Slawek Smyl. Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach. *arXiv*, 2017. ISSN 23318422.

[9] A Suilin. Kaggle-web-traffic. https://github.com/Arturus/kaggle-web-traffic/, 2017. Accessed: 2021-04-30.

[10] Rob J. Hyndman and Anne B. Koehler. Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4):679–688, 2006. ISSN 01692070. doi: 10.1016/j.ijforecast.2006.03.001.