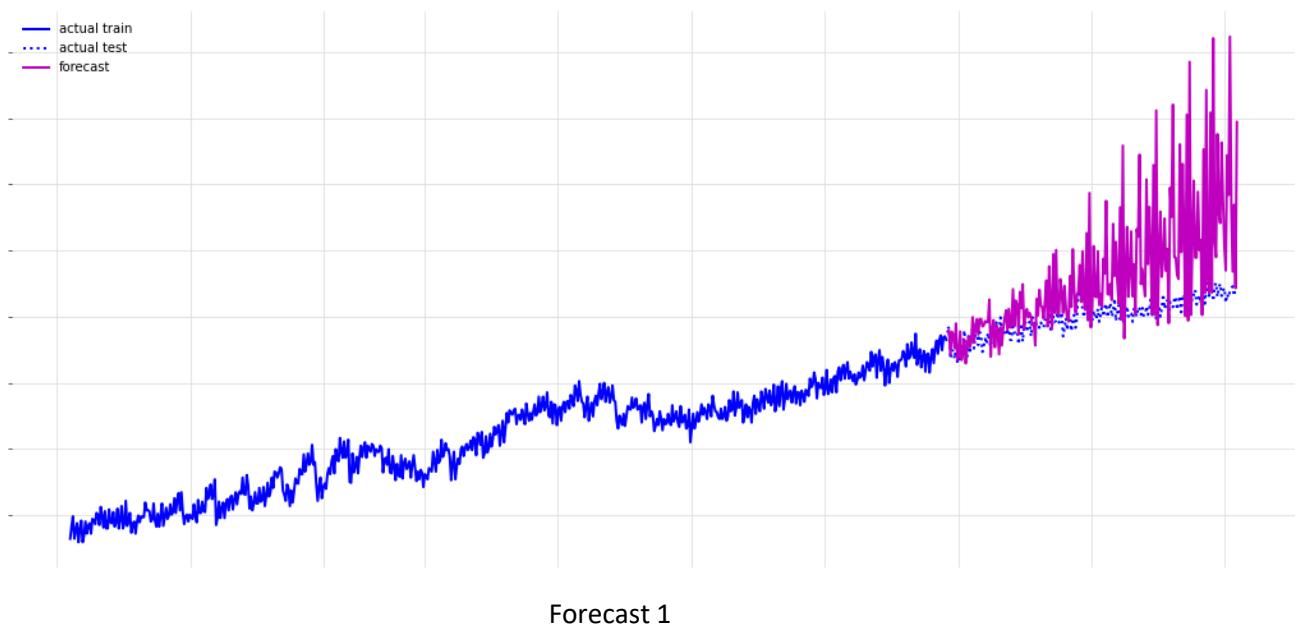
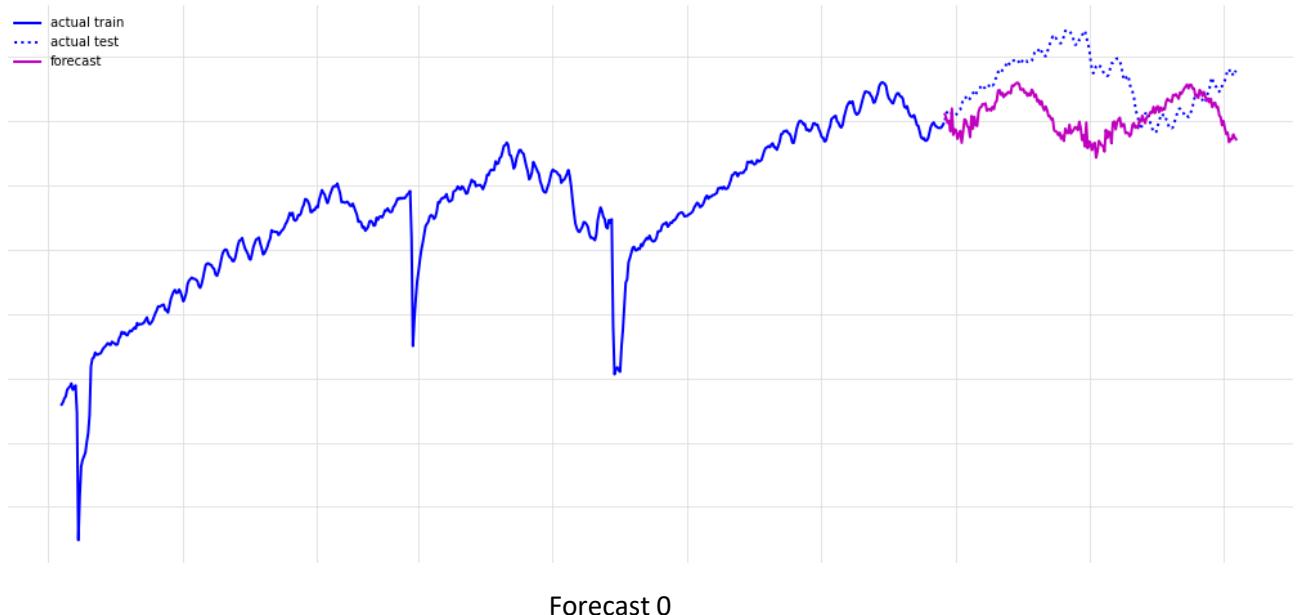
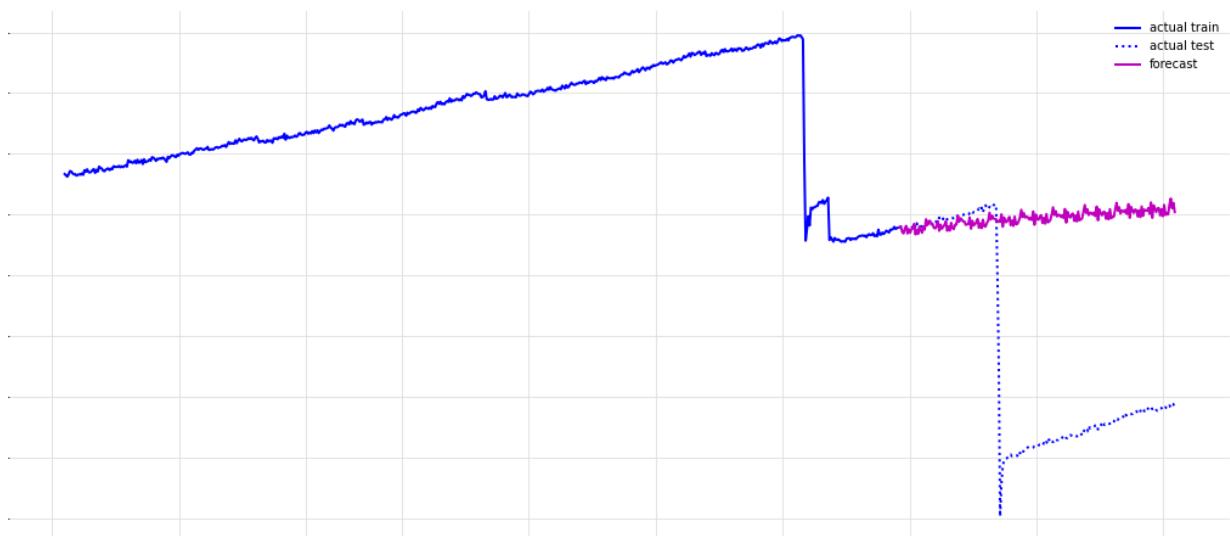


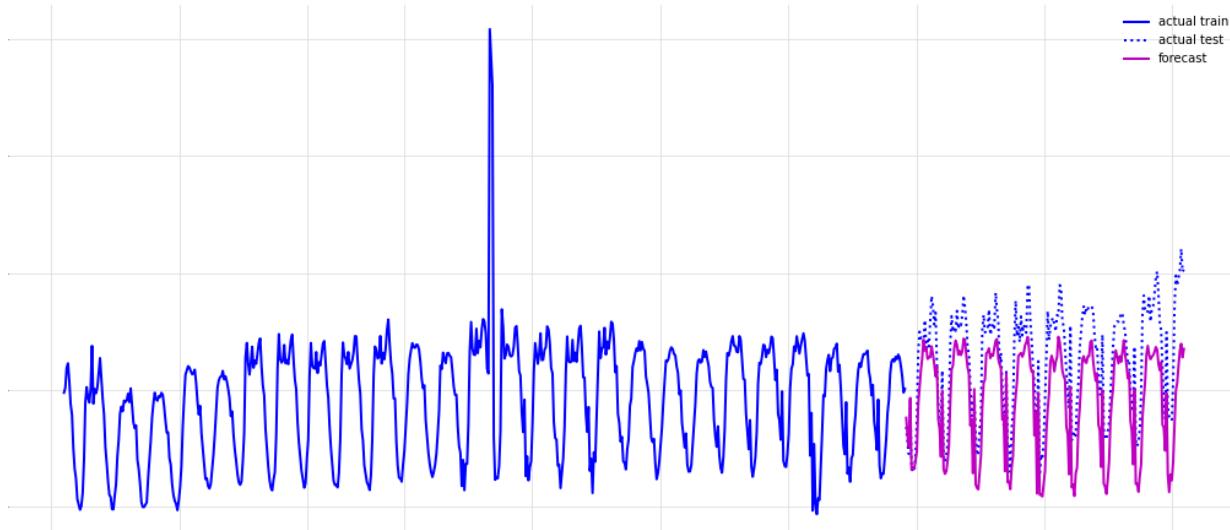
Machine Learning Research Engineer Assignment
Bhavesh Uppaluri (ubhavesh2908@gmail.com)
Applied for – ML Intern

Forecast Plots

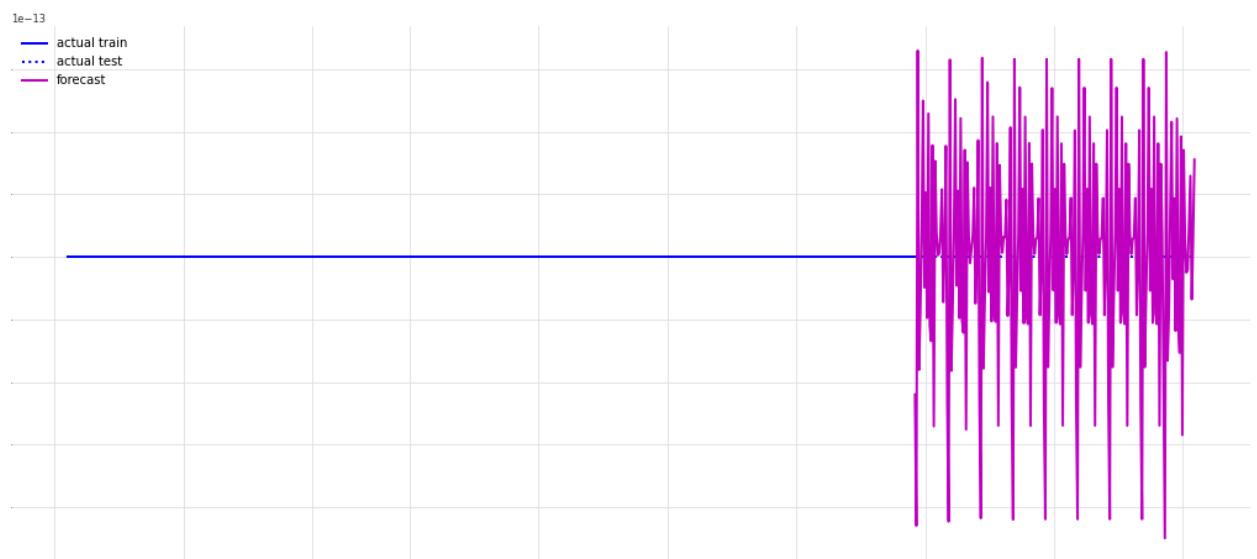




Forecast 2



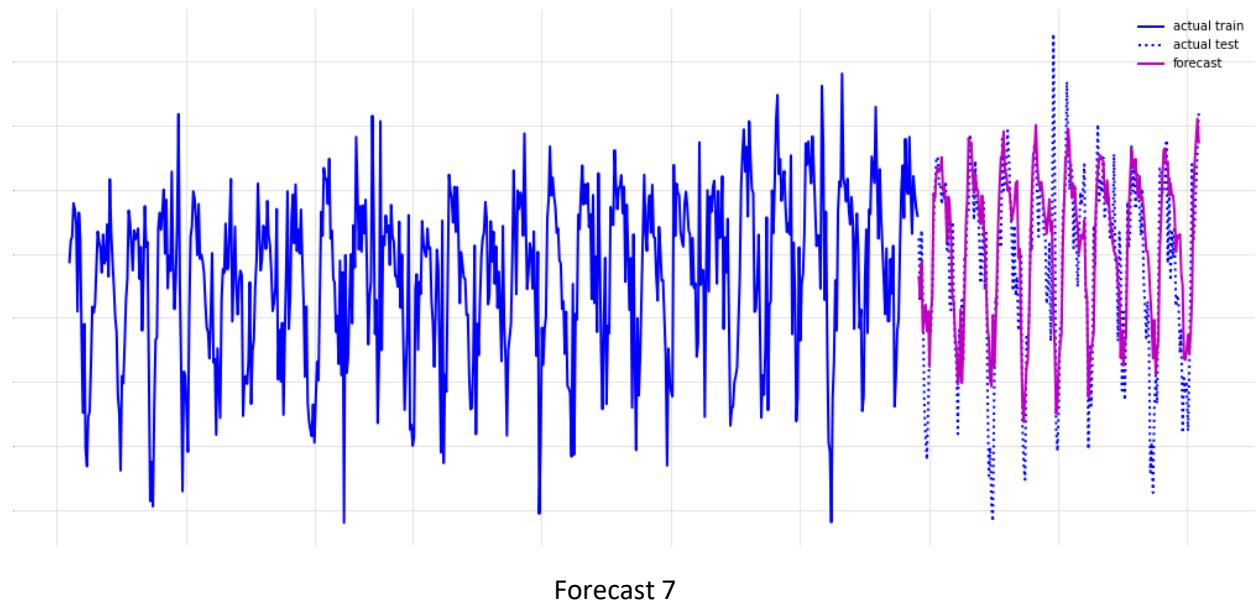
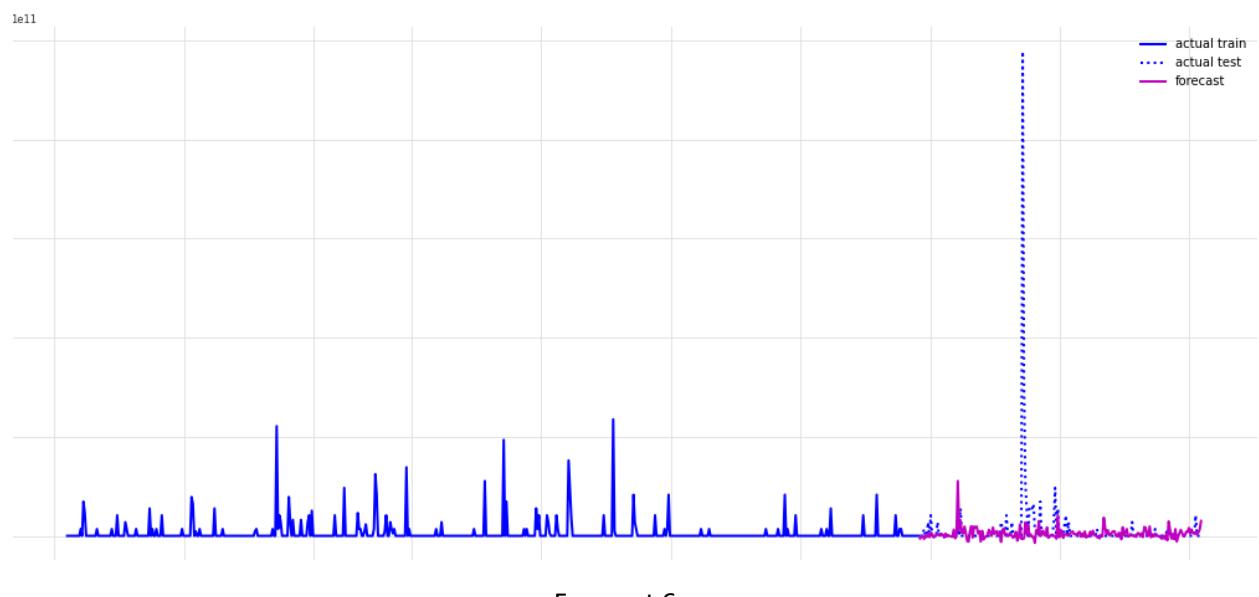
Forecast 3

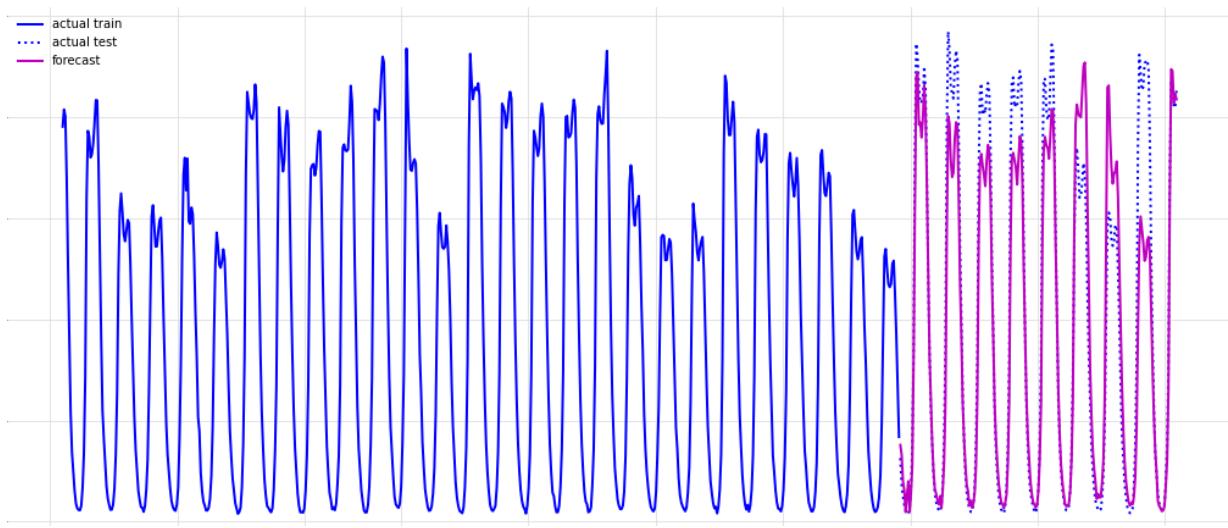


Forecast 4

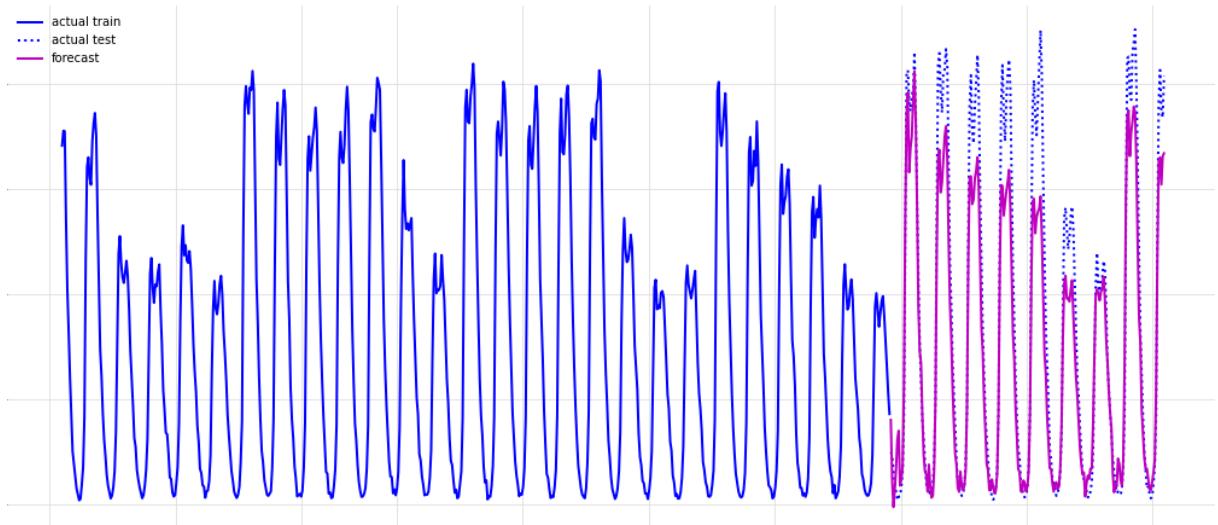


Forecast 5

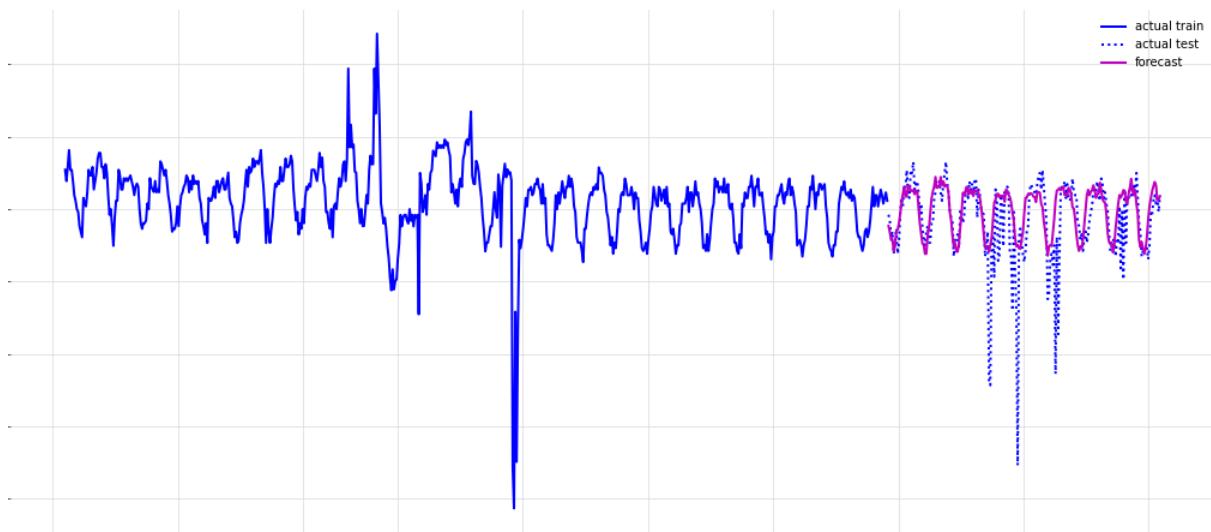




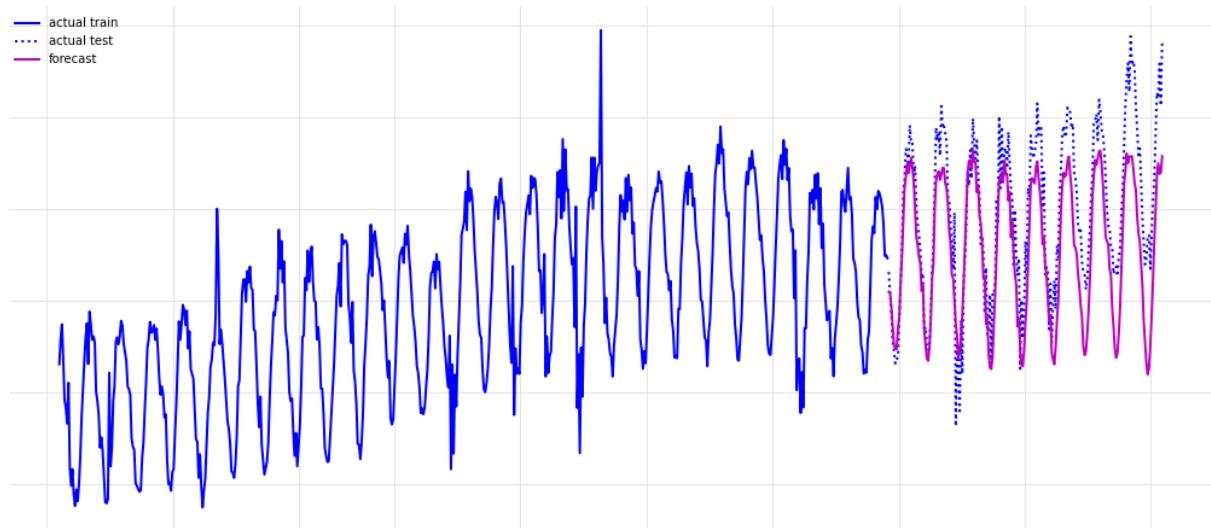
Forecast 8



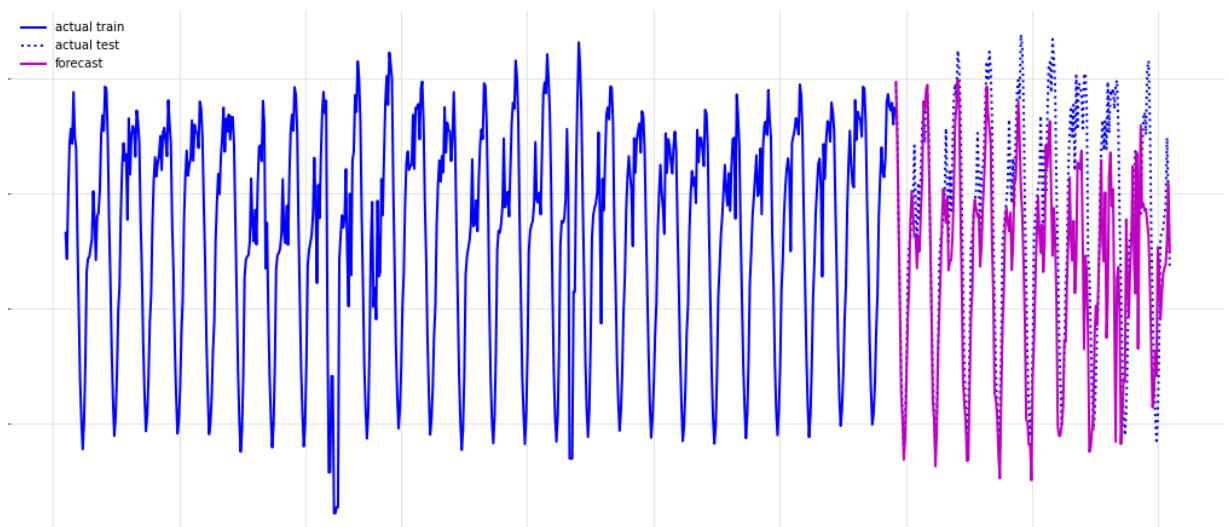
Forecast 9



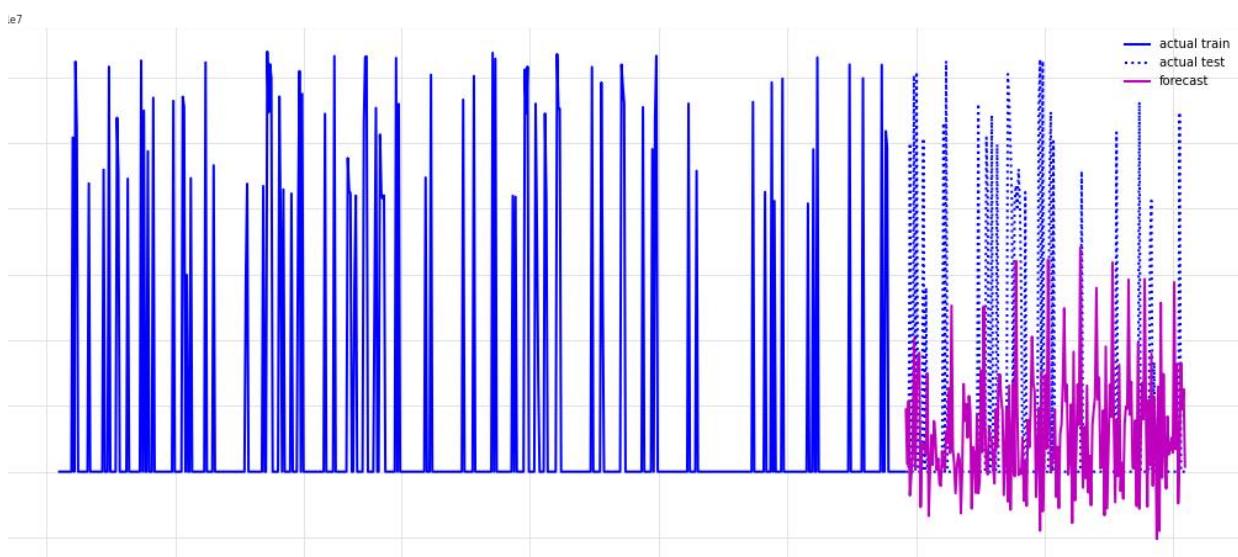
Forecast 10



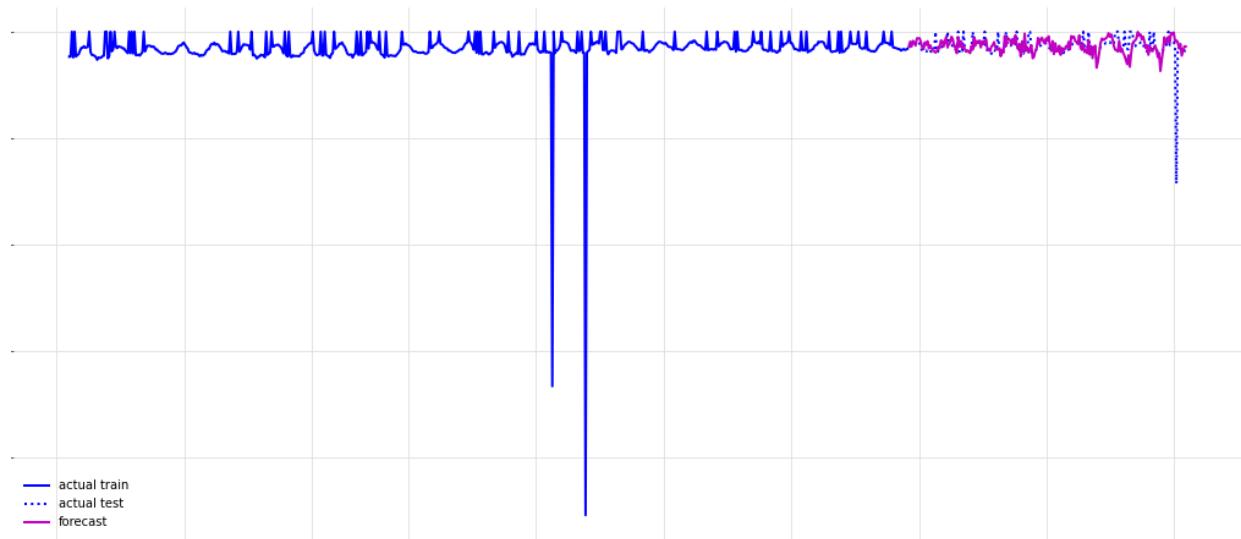
Forecast 11



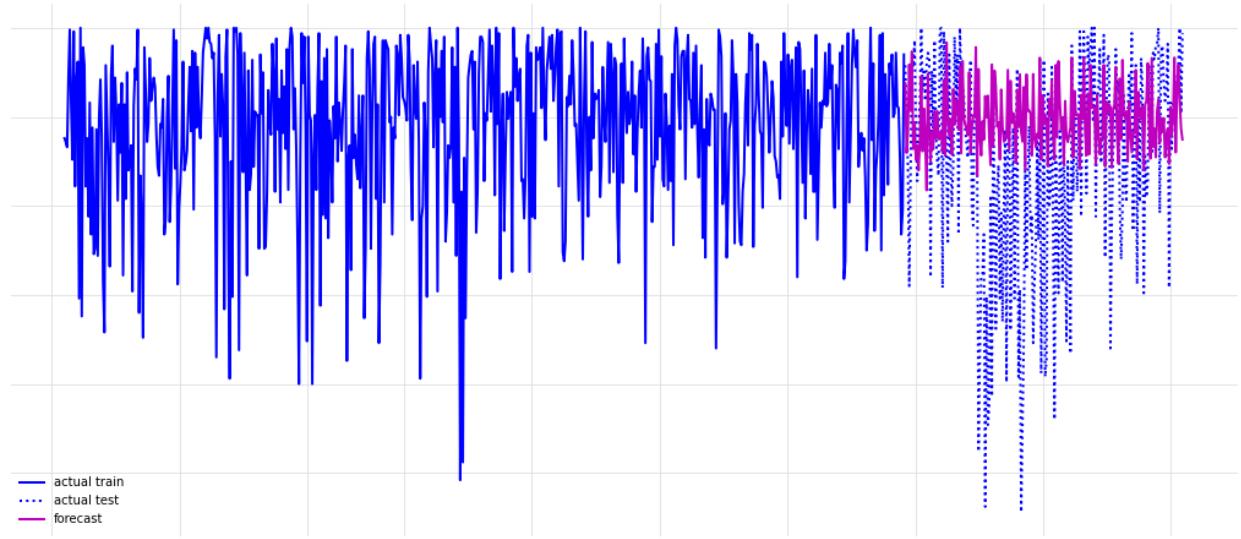
Forecast 12



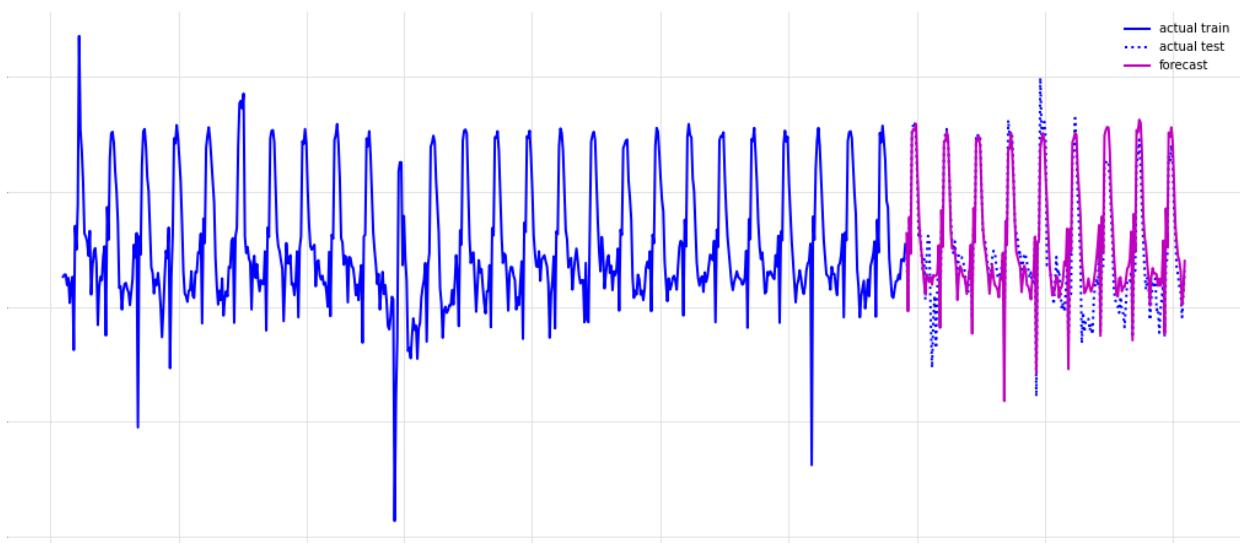
Forecast 13



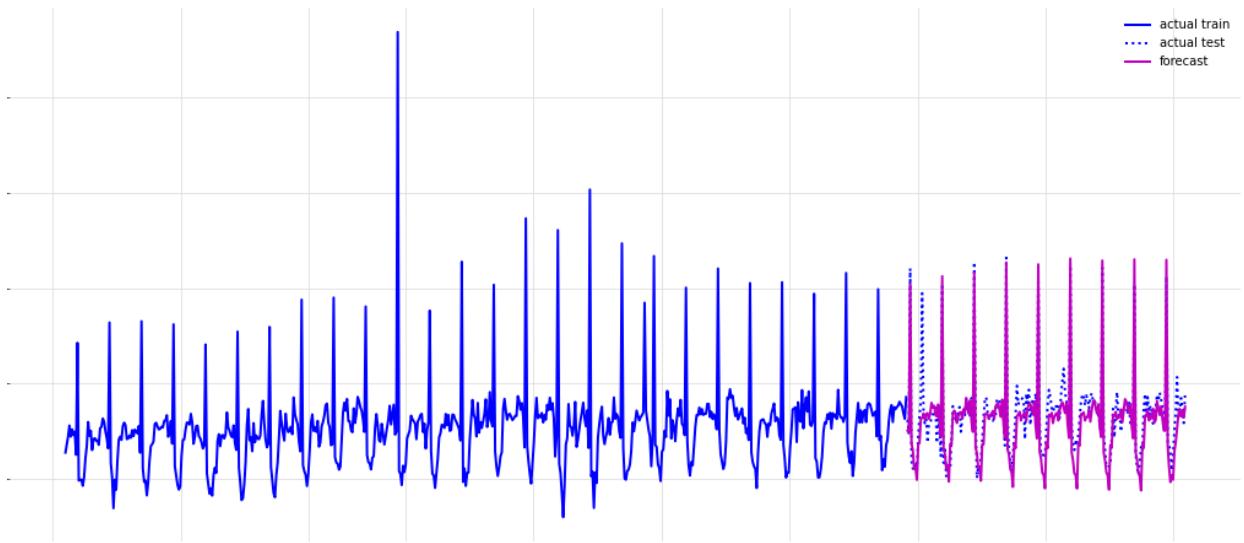
Forecast 14



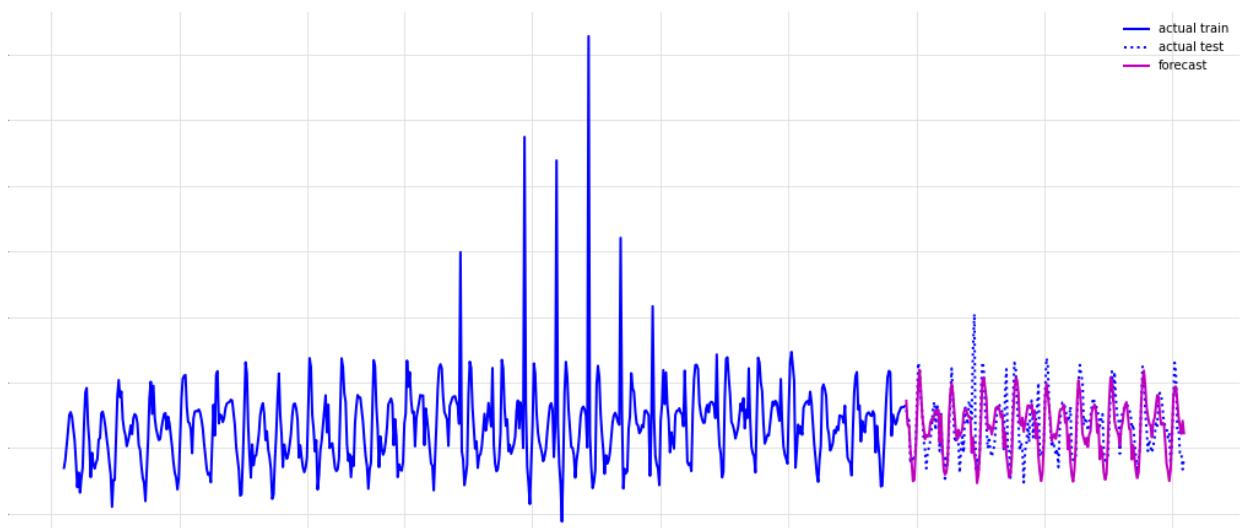
Forecast 15



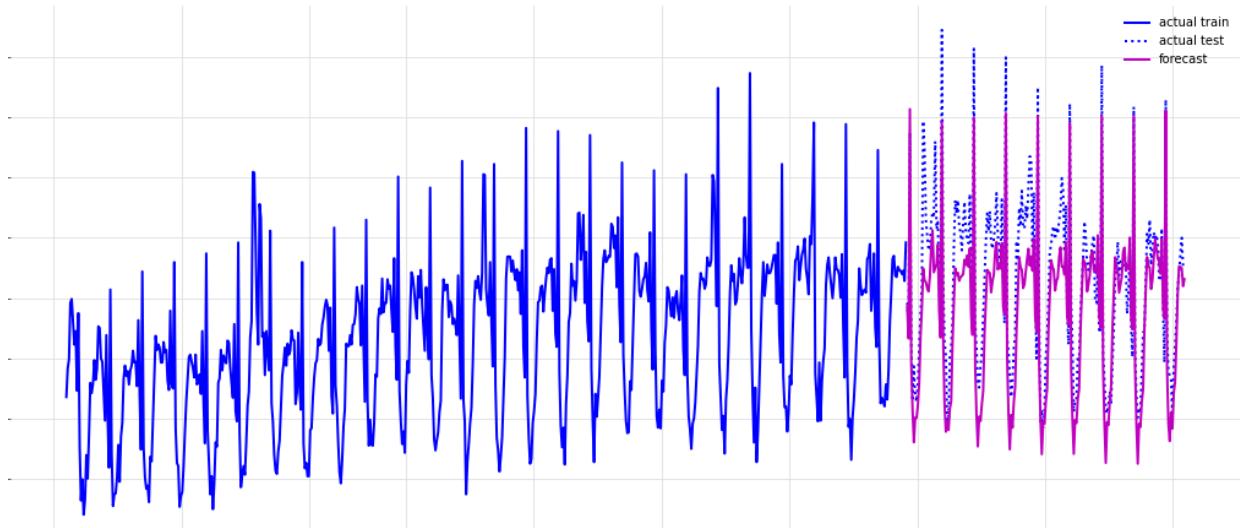
Forecast 16



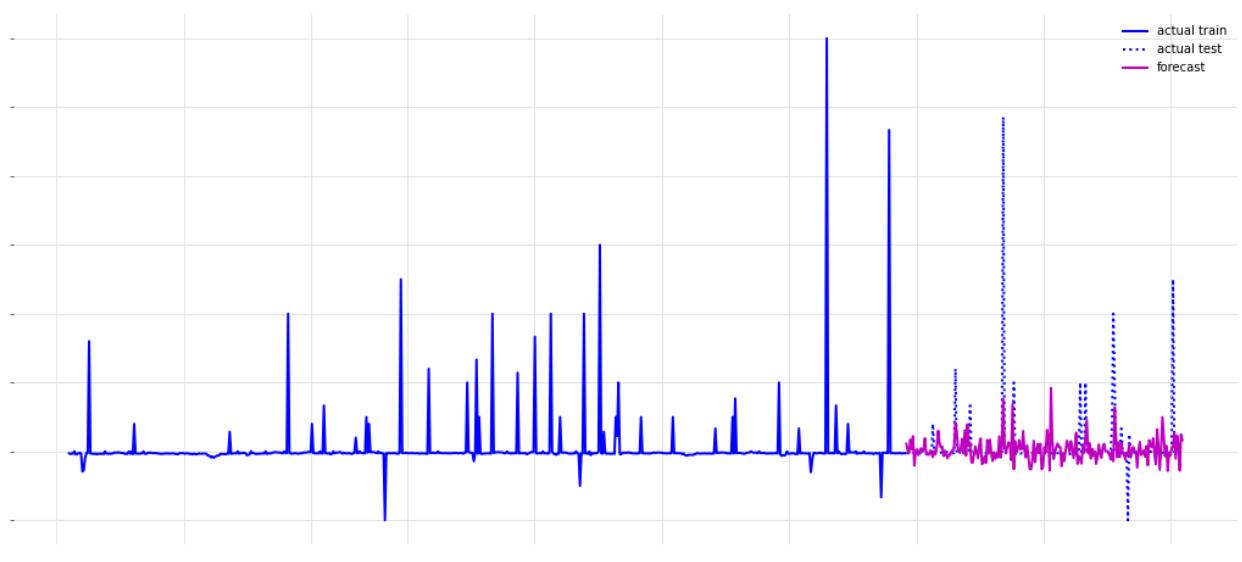
Forecast 17



Forecast 18



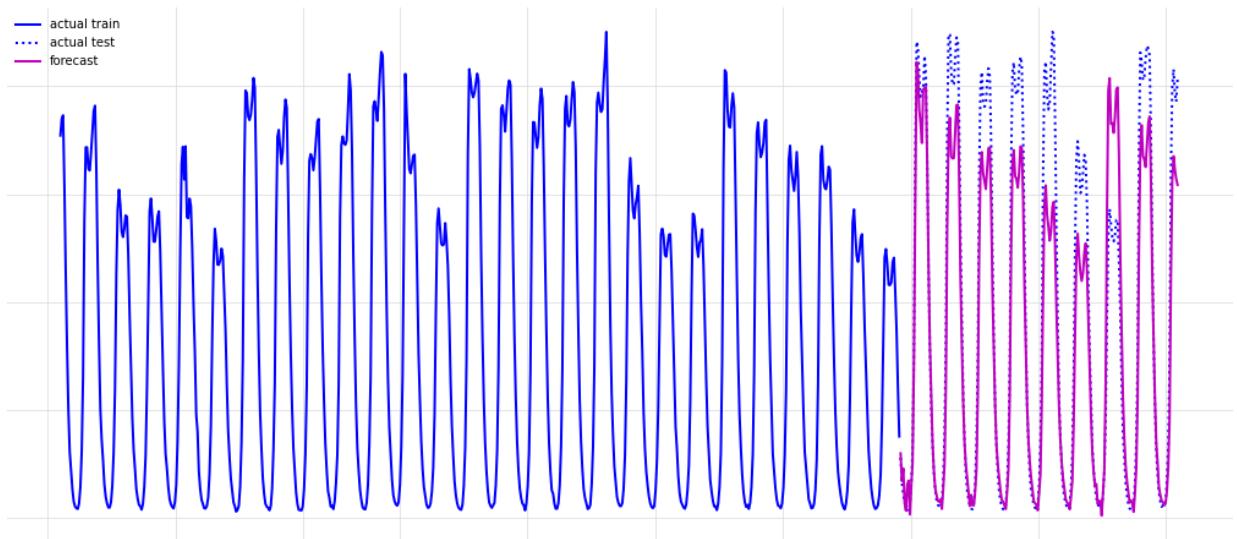
Forecast 19



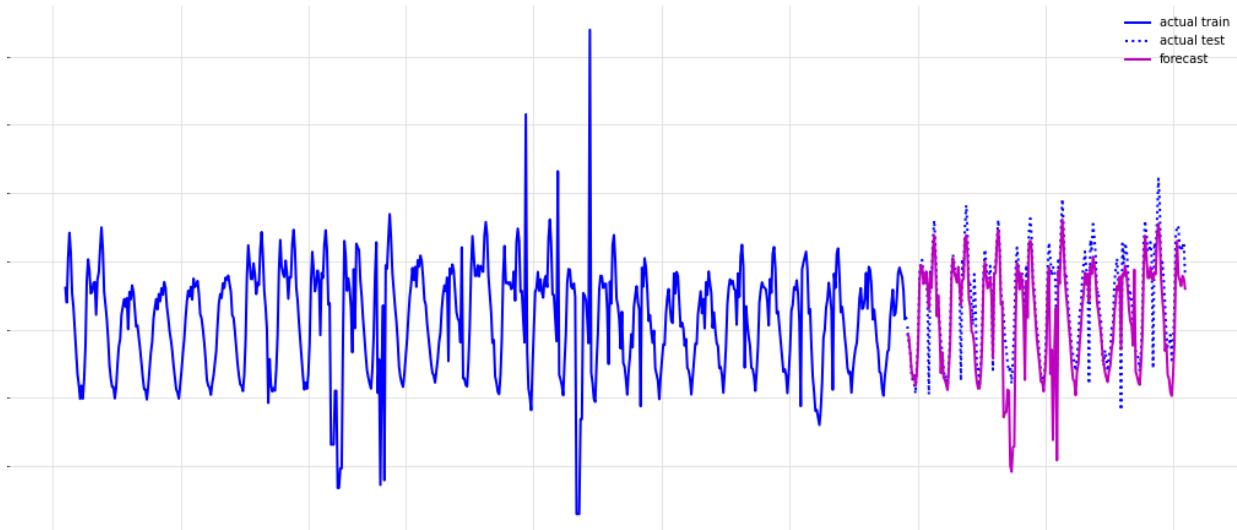
Forecast 20



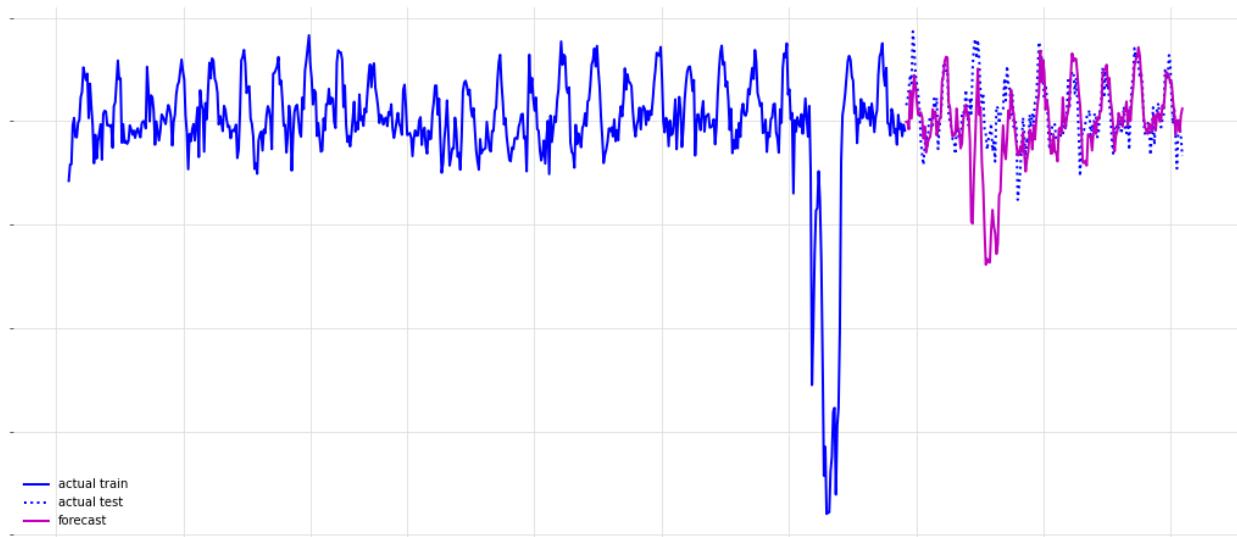
Forecast 21



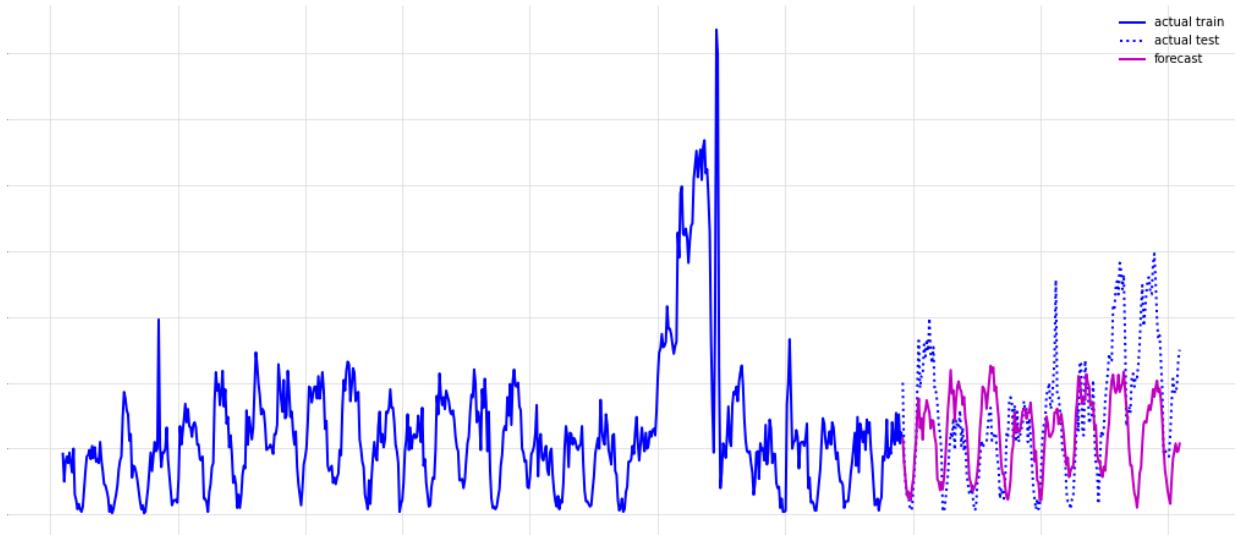
Forecast 22



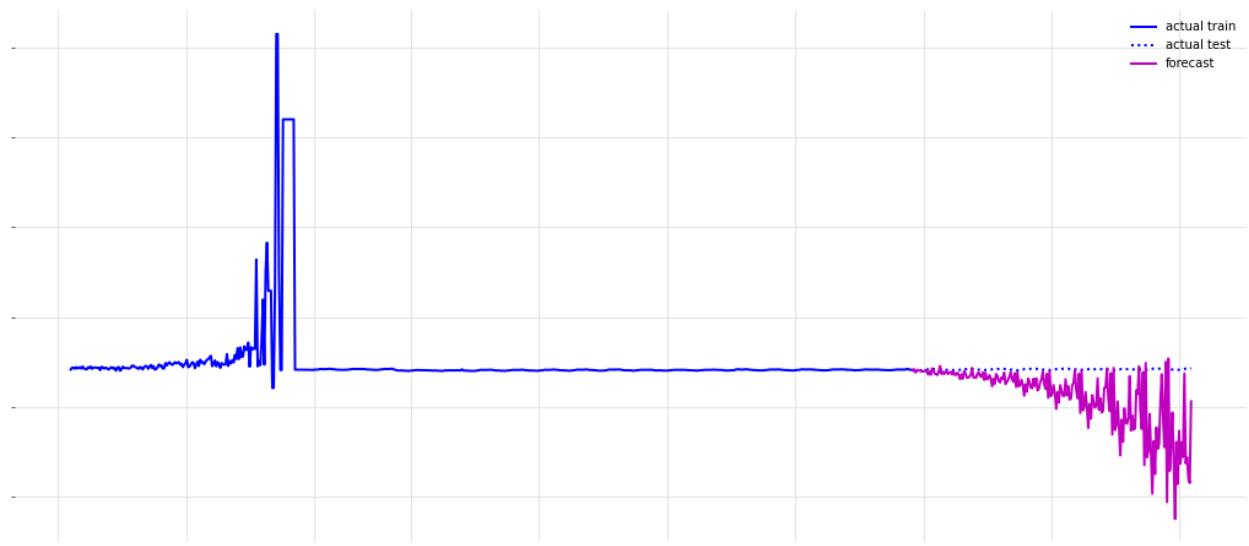
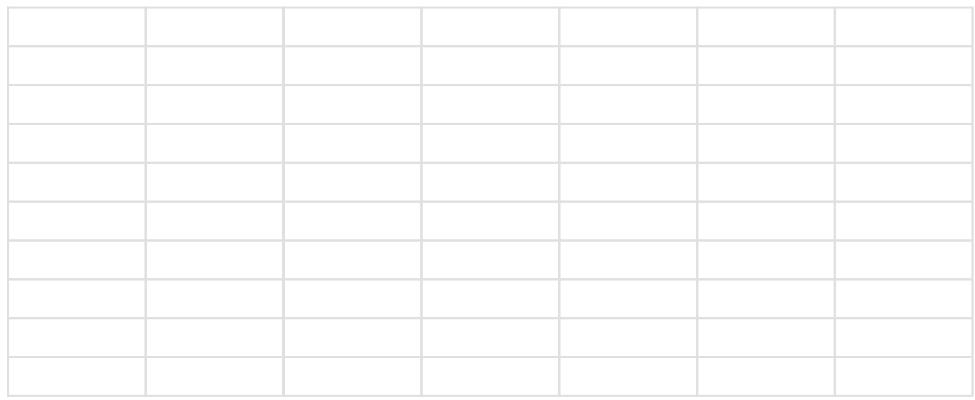
Forecast 23



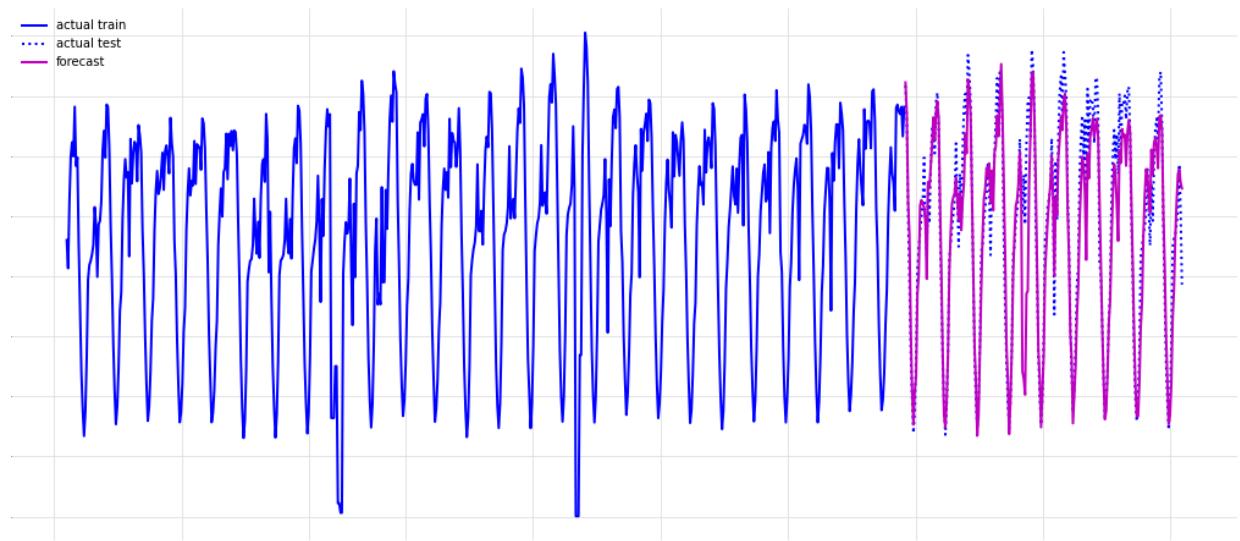
Forecast 24



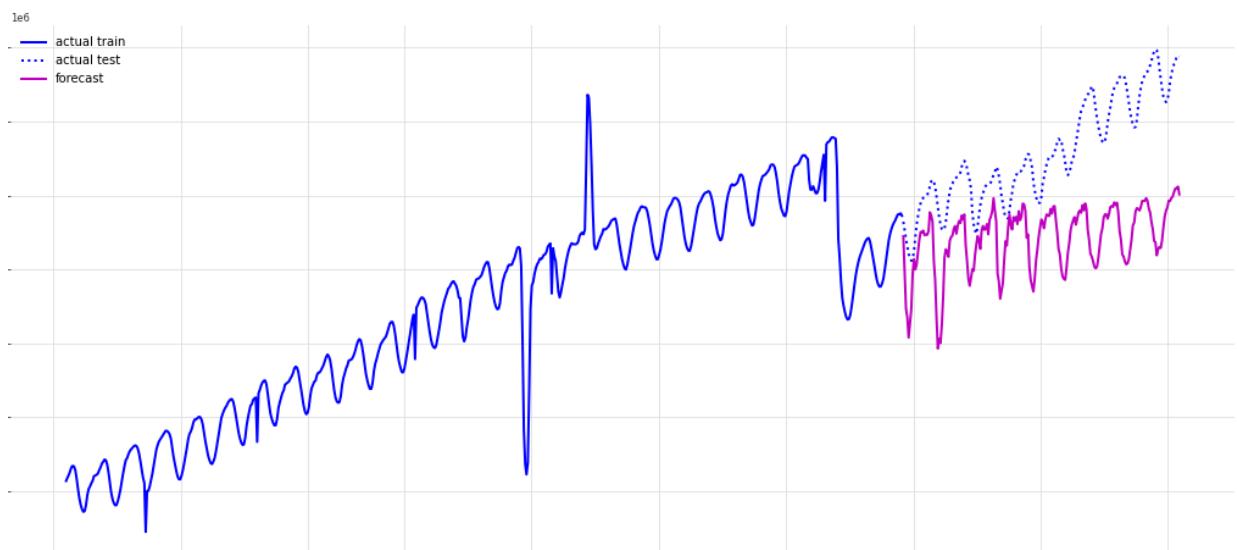
Forecast 25



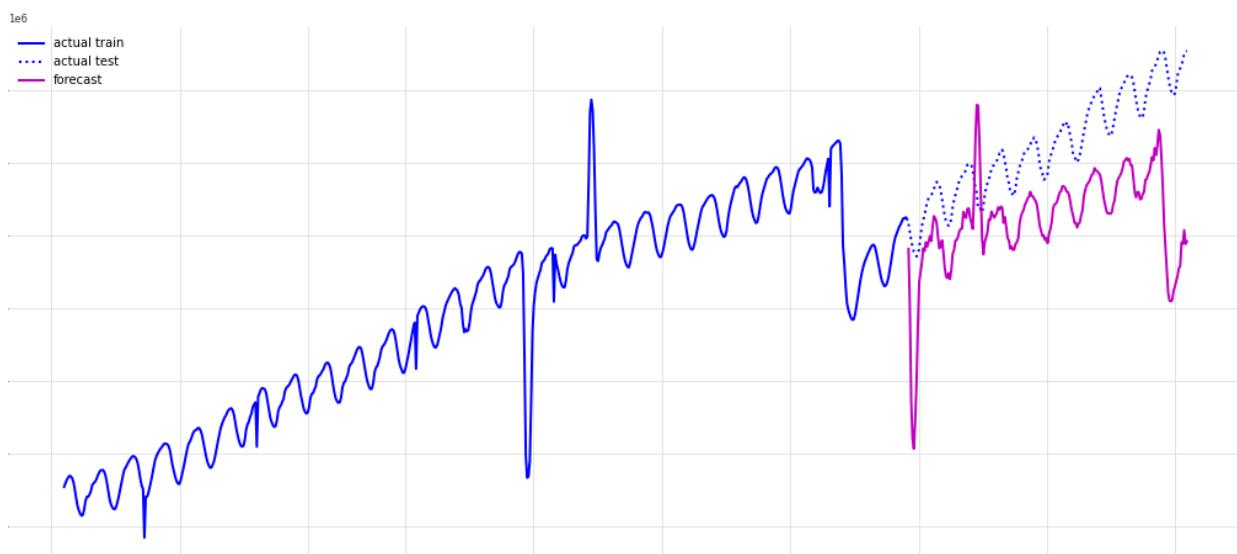
Forecast 26



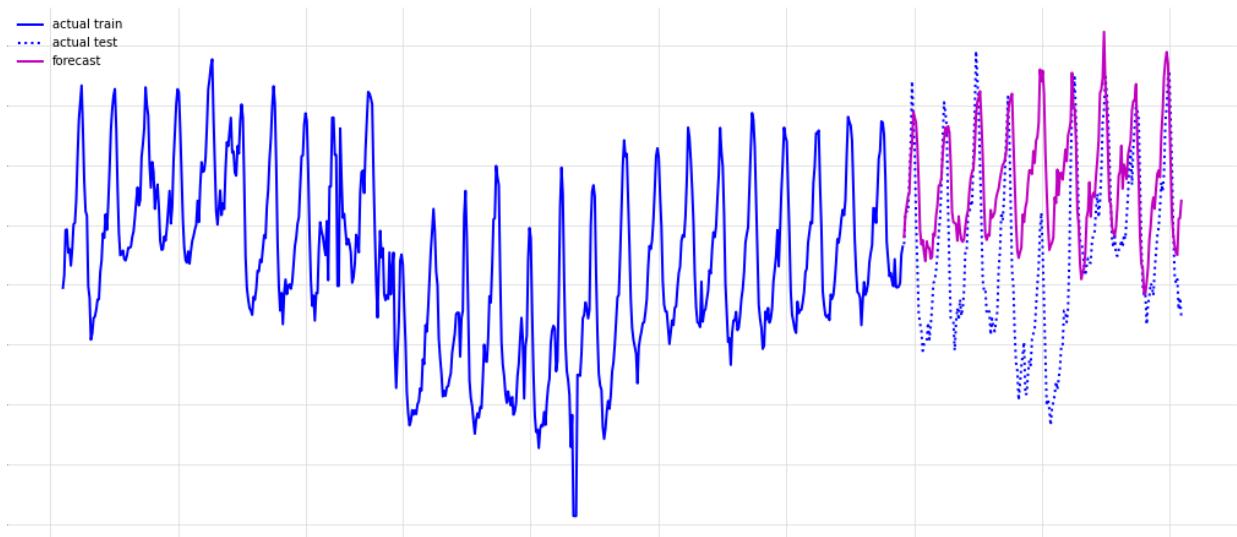
Forecast 27



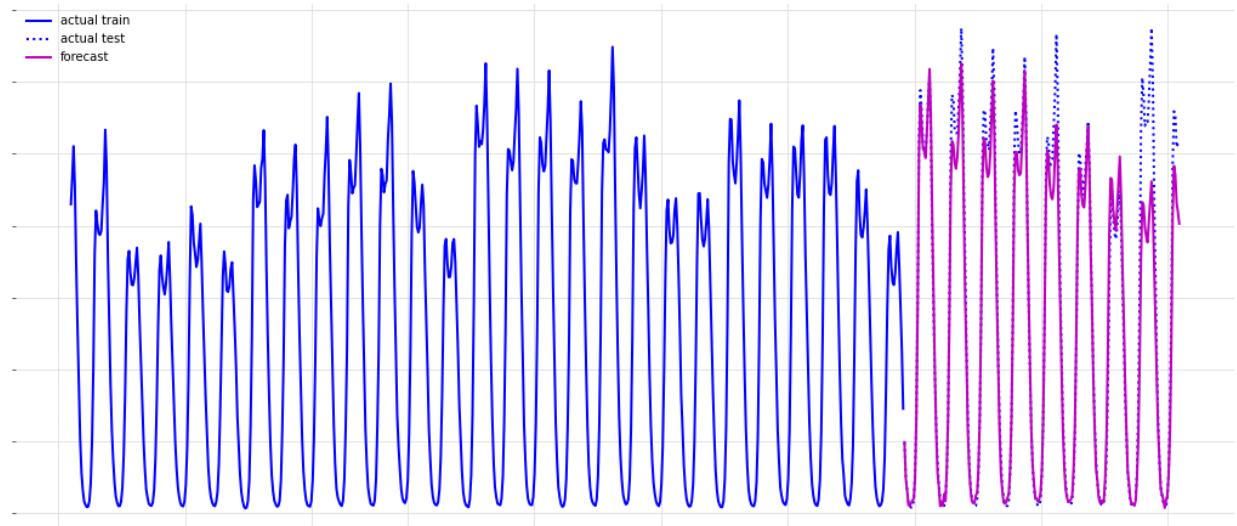
Forecast 28



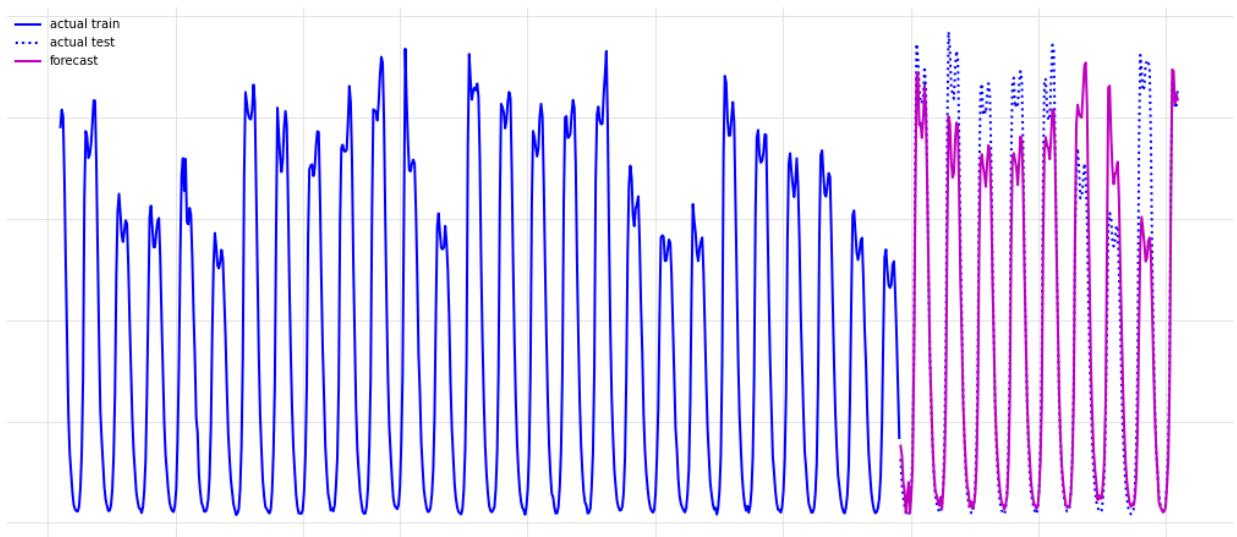
Forecast 29



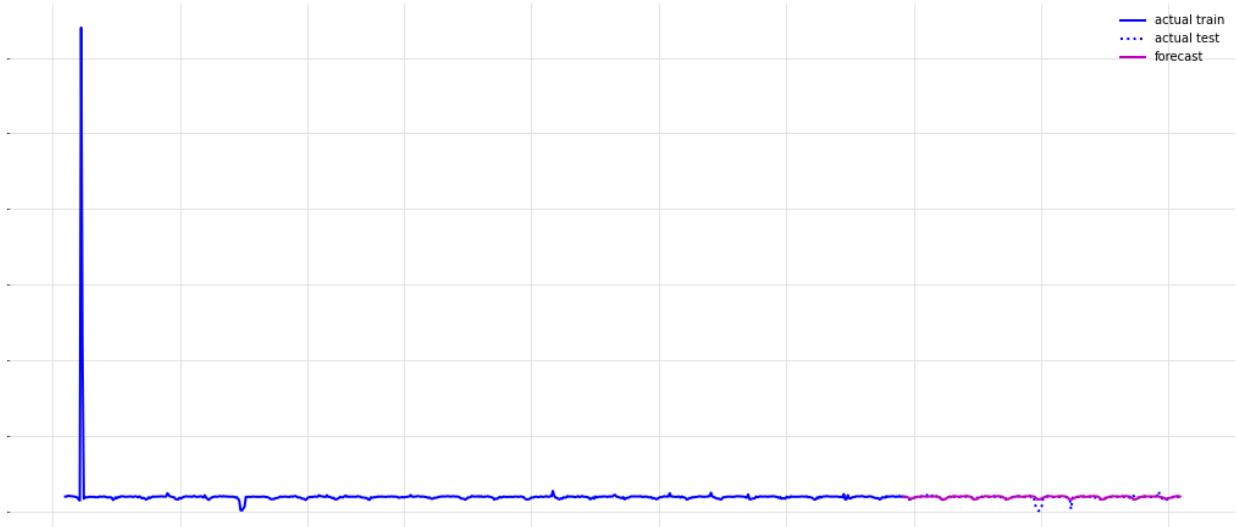
Forecast 30



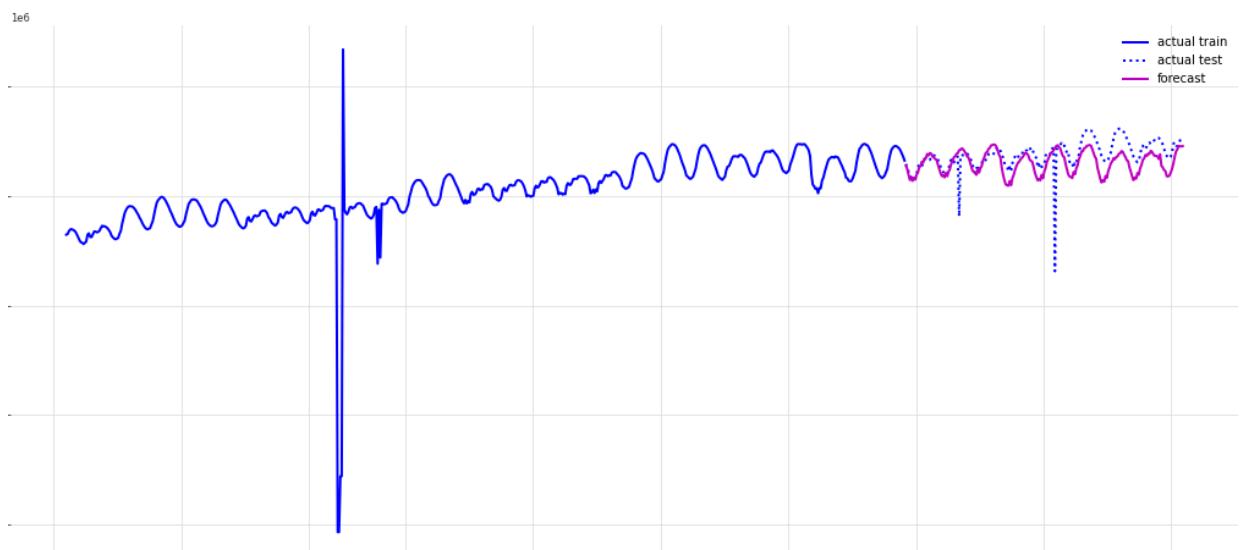
Forecast 31



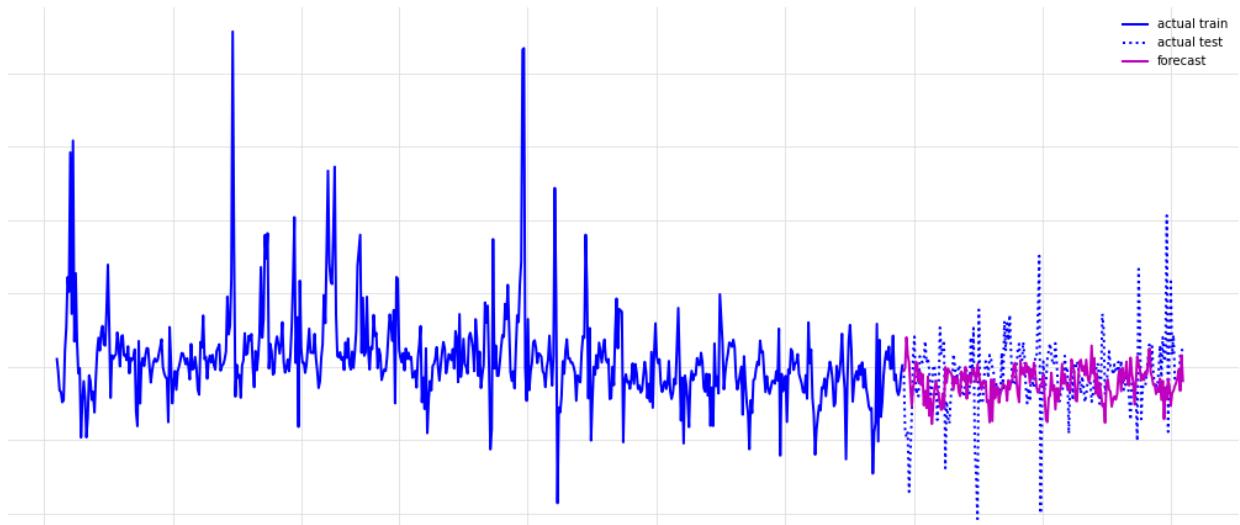
Forecast 32



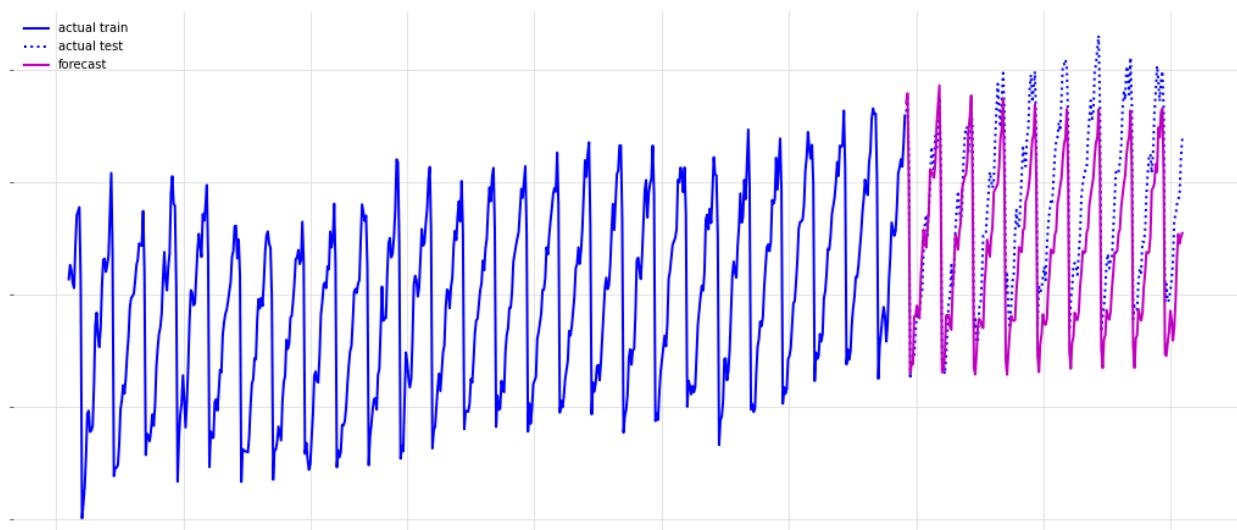
Forecast 33



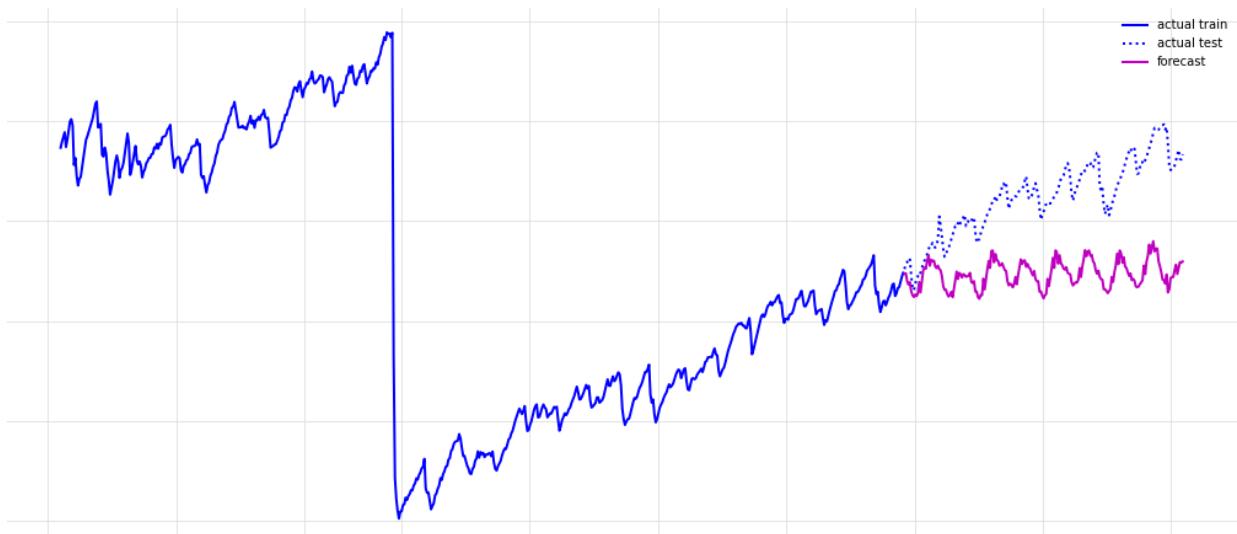
Forecast 34



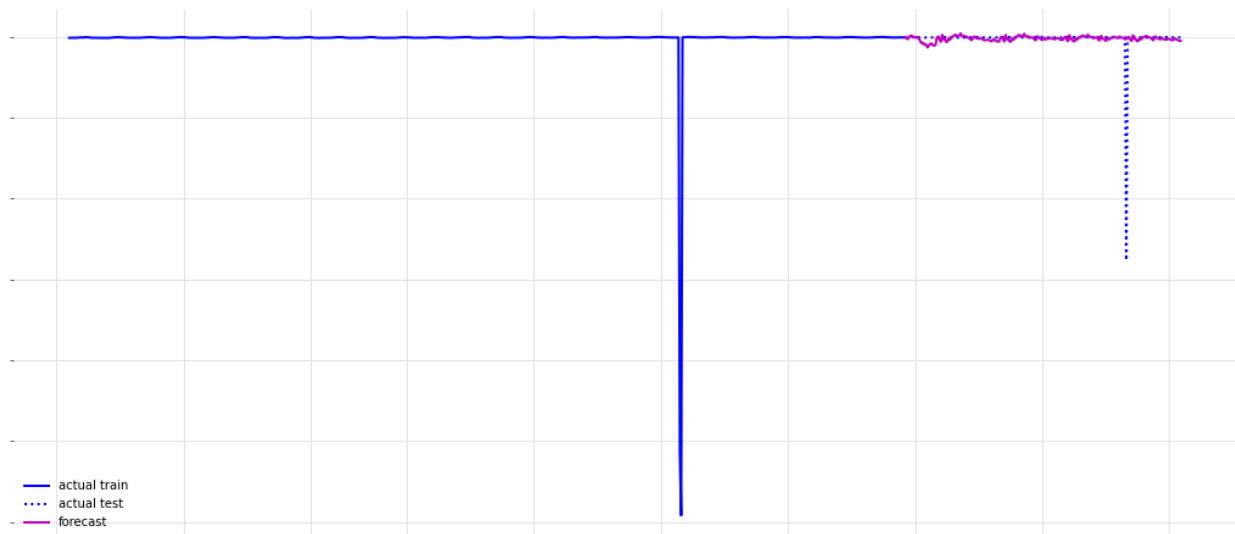
Forecast 35



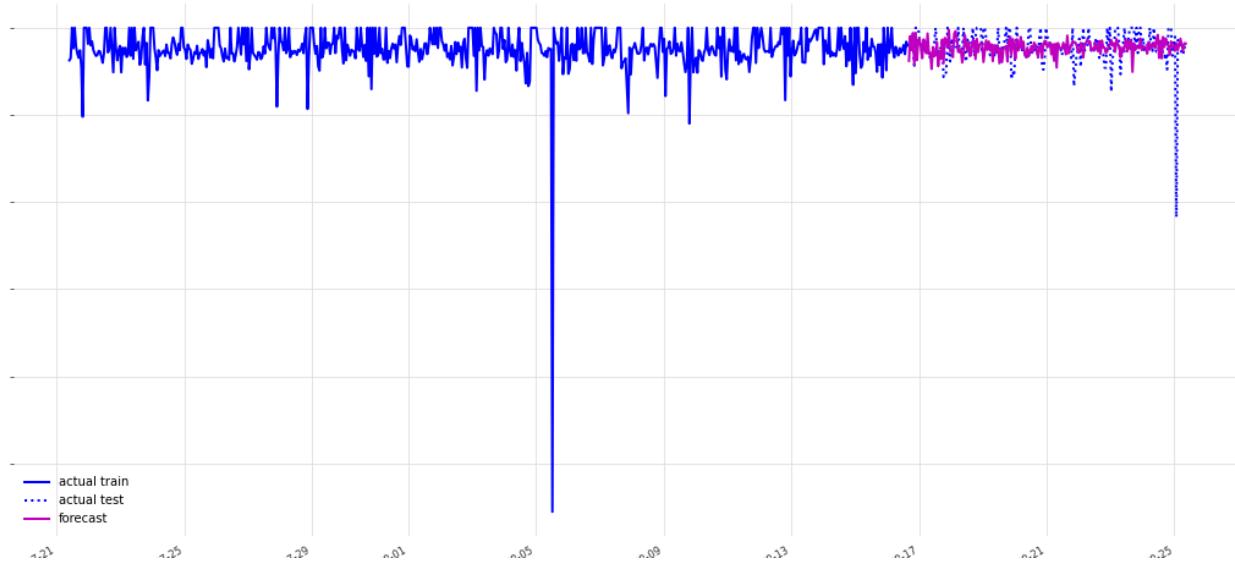
Forecast 36



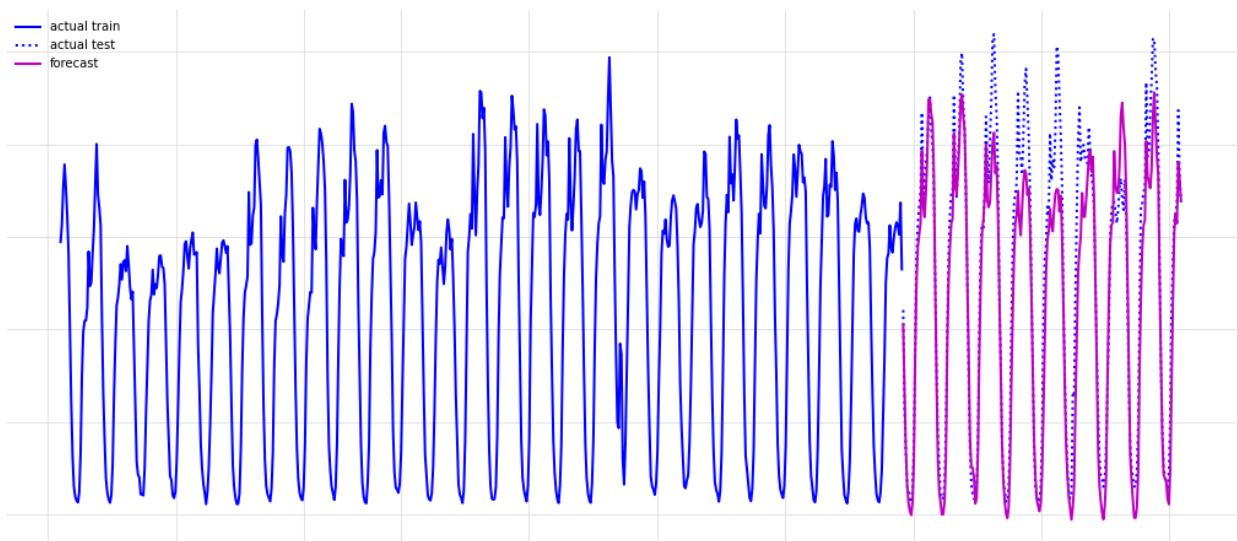
Forecast 37



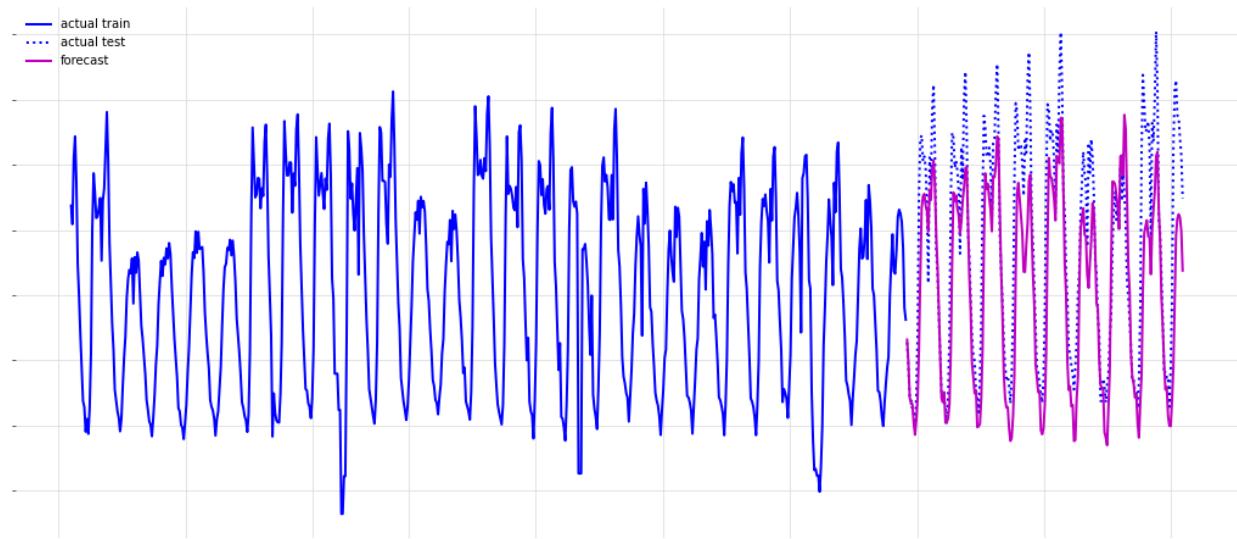
Forecast 38



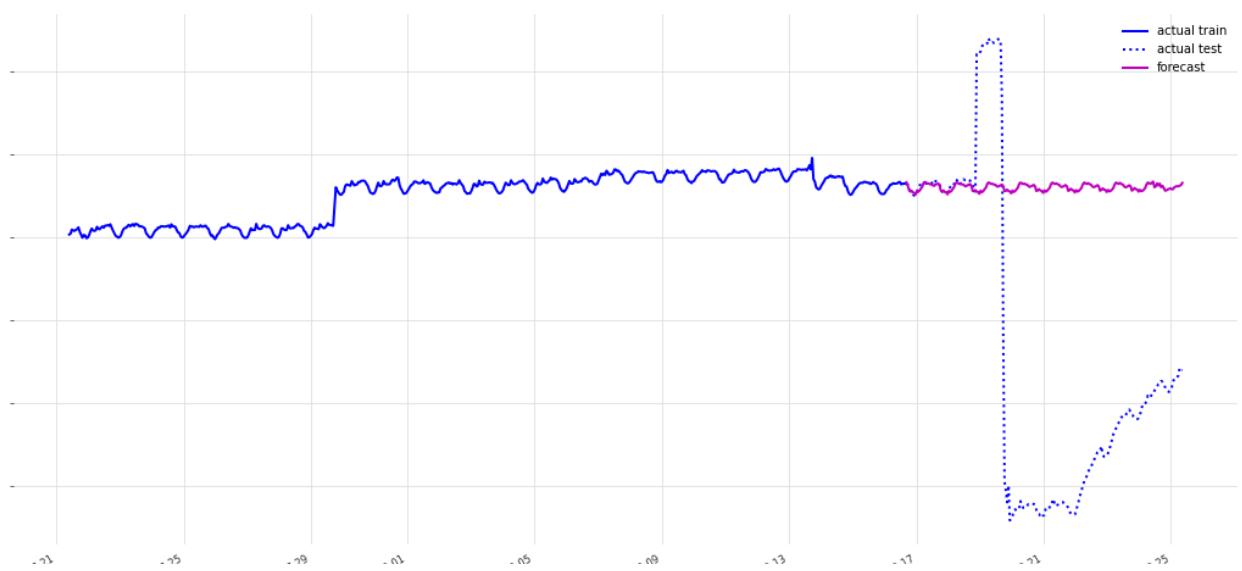
Forecast 39



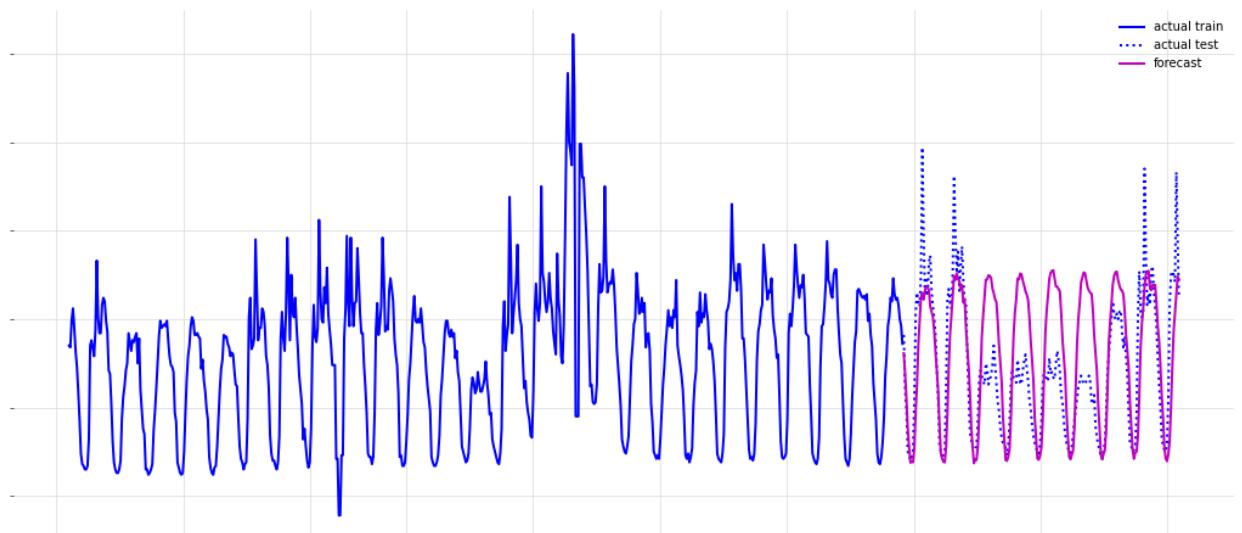
Forecast 40



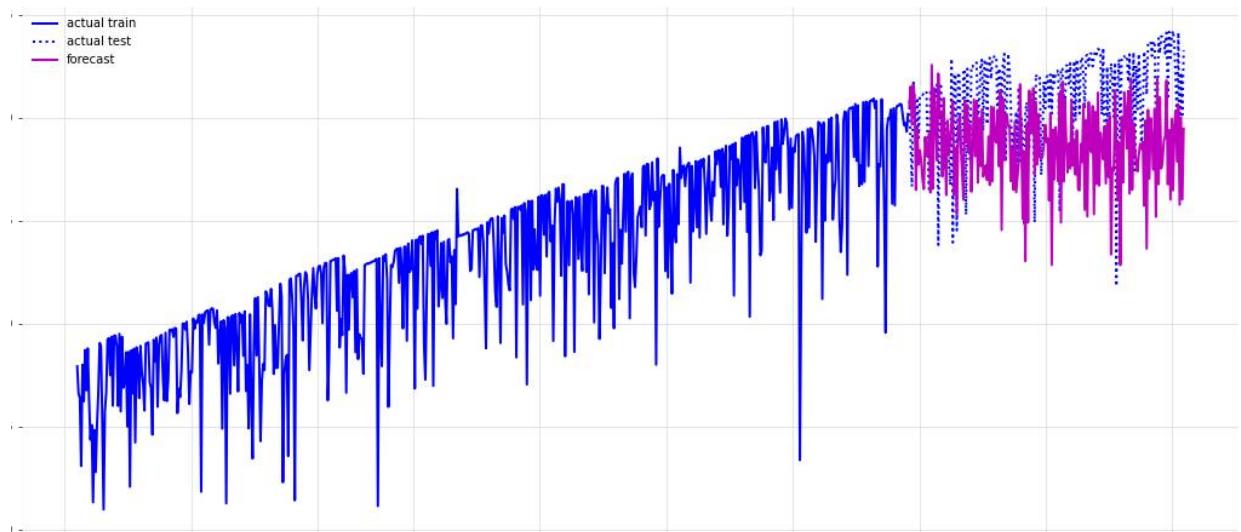
Forecast 41



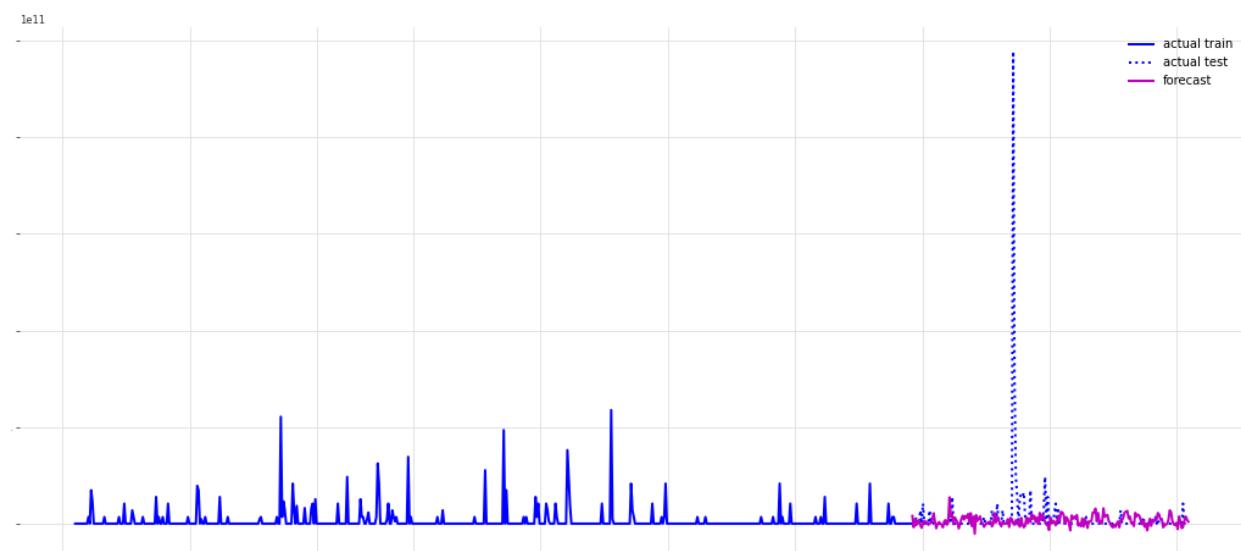
Forecast 42



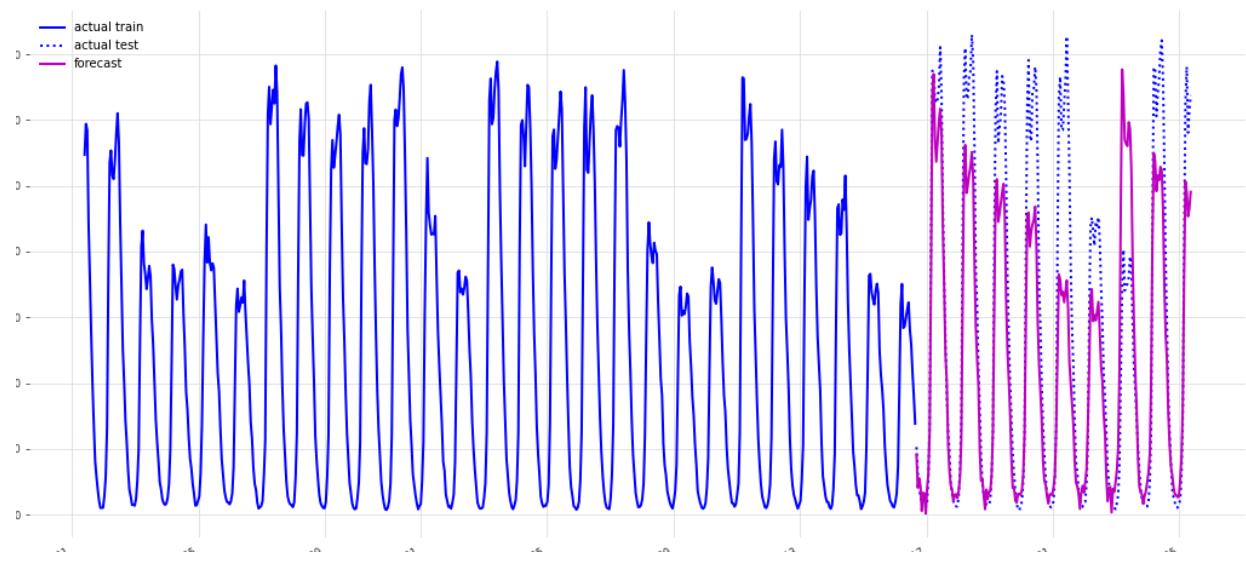
Forecast 43



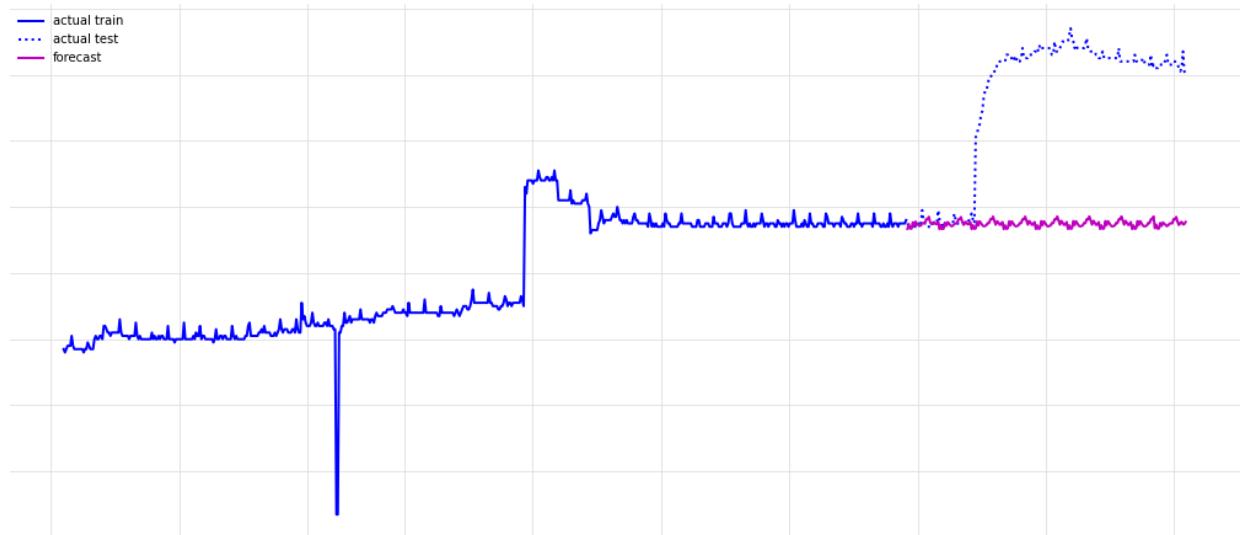
Forecast 44



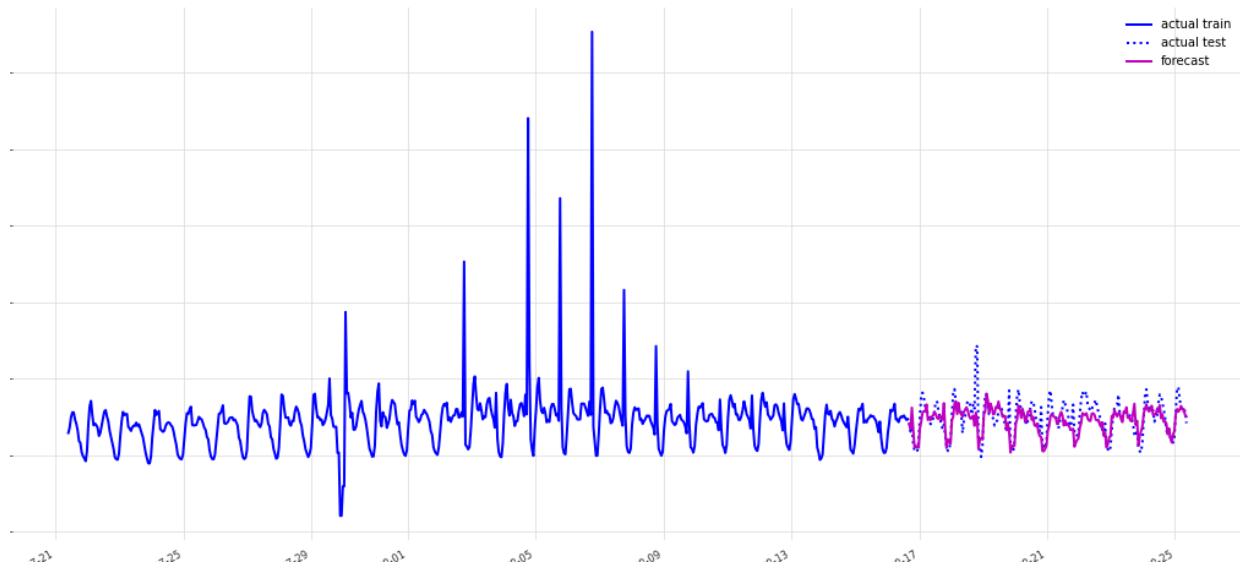
Forecast 45



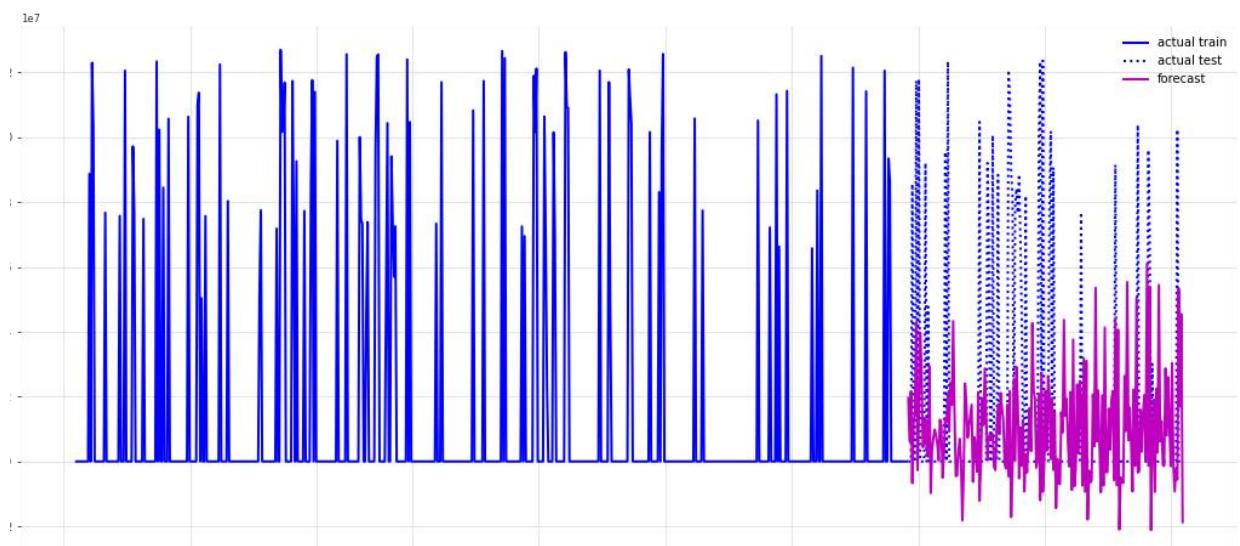
Forecast 46



Forecast 47



Forecast 48



Forecast 49

Benchmark Numbers

DATA	MAE	MSE	RMSE	MAPE
0	0.12612509900170069	0.024198295009439465	0.1555580117172994	0.26290317501470517
1	0.0655551819072414	0.010388990936702822	0.10192639960629837	0.22990608456039982
2	1.1617019888826996	2.037466163857422	1.4273983900290144	2.655545056665787
3	70.48688230897851	6553.408803521374	80.9531272003829	2.032005256858877
4	1.4084744606521534e-13	3.1115688126098876e-26	1.7639639487840696e-13	
5	3.039139617782327	20.88048603587931	4.569517046240151	72.60391462214513
6	8488952128.244324	1.363662944499226e+21	36927807198.63049	344756319.4946807
7	140.6379736135317	35770.9602962864	189.13212391417383	10.179926714459999
8	1877.5885559776436	7756569.713358287	2785.061886809391	30.018728996772957
9	3542.308605455195	24286868.004587237	4928.170857893143	47.777416674041355
10	0.07024170650305607	0.014467931671487454	0.12028271559741015	1.0320939669181088
11	0.047742708190771414	0.00332314210104095	0.057646700695191136	0.5294343627484632
12	5852.87252414172	58722886.830946505	7663.085986138124	22.038329867060696
13	2886068.541248724	18774410752400.434	4332944.812988095	38312716.134632975
14	0.12600381018542214	0.05761956544201853	0.24004075787669588	0.12662619979557901
15	0.48914169354784387	0.4376590475506016	0.6615580454885283	0.49508263042771716
16	0.35548380074414565	0.20983156205781772	0.4580737517669155	0.39161314440144396
17	0.07098968135003413	0.009550072694588385	0.09772447336562314	1.4631623618955127
18	6853.632921015647	85478101.2273193	9245.436778612426	8.159490551497283
19	0.05952956009538465	0.005224873333539411	0.07228328529846587	0.35106847251103346
20	9.145265839370174	461.2808902593384	21.477450739306523	7.73734713066623
21	33888.15927370629	1350720499.1972575	36752.149586075335	0.31759162330177126
22	4064.2374801167884	32610093.981285725	5710.524842891915	30.340762242132595
23	3514.6451818059686	25725087.22986155	5071.990460348043	10.958529558697904
24	0.5092879996255936	0.5891496655281042	0.7675608546090038	0.5186679522491247
25	35.65531101621262	1981.1184289923615	44.50975655957199	81.83476949799166
26	1118.0680364597617	2765489.174728455	1662.9759994445064	9.372476060502413
27	3.039139617782327	20.88048603587931	4.569517046240151	72.60391462214513
28	115146.50216348954	17279073382.760998	131449.88924590618	3.7293221804651027
29	106523.21814985787	14929595570.926537	122186.72420081707	3.2847897210555836
30	1.2878560378821422	2.561985239564101	1.6006202671352443	14.891567160549412
31	2928.852251371396	24809561.741004337	4980.91976857732	17.277042413999983
32	1877.5885559776436	7756569.713358287	2785.061886809391	30.018728996772957
33	0.6087130993970967	1.7965158762459685	1.340341701300817	22.73870113793243
34	20404.106006947026	853029046.2680435	29206.660991425288	2.3581181853944764
35	42.20025870844651	3408.786101619876	58.38481053852856	3.8903347774709482
36	0.08552587518235034	0.009584039547589447	0.09789810798779232	0.5779657900812831
37	0.7638648917715476	0.6980849872722453	0.8355148037421272	2.2877074484301305
38	0.3739558429113322	3.6857340596412254	1.9198265701987838	0.4242459536316819
39	0.2812693863413277	0.18947158261781974	0.43528333602128594	0.283267534754295
40	3732.9162466036564	25376494.43959771	5037.508753302342	24.99884629815229
41	15.790168293586634	453.8594364422328	21.30397700999118	12.899610341308426
42	1.1391007073283126	1.838257519564165	1.3558235576815167	11.66368116697407
43	0.2517552394899248	0.10484285402094216	0.3237944626162439	14.53742038865659
44	0.32810279096666933	0.14863118128095604	0.38552714726845894	1.125913704123885
45	9540720819.294909	1.3523454091709852e+21	36774249267.26561	35785906.6392745
46	424.1707646458827	352128.21391247964	593.403921382796	56.435301858395306
47	0.37601644374966614	0.18629205802382057	0.4316156369083731	2.343597518548591
48	1253.6903027227097	3045104.6537922765	1745.0228232869267	8.097194394930485
49	2595924.0578850713	15922715349168.938	3990327.7245320263	5282933.921404671

Understanding RevIN

When we train general neural network models to forecast univariate time series data, we observe that the predictions of the baselines are inaccurately shifted and scaled. Statistical properties such as mean and variance often change over time in a time series, these discrepancies between the distributions of the training and test data of forecasting models result in the issue we define as the distribution shift problem

There's only so much a neural network model can learn and therefore it is feasible to look for optimization in other parts of a traditional deep learning pipeline. RevIN is one of the latest methods developed to address such issues, the method proposes a simple, yet effective normalization method called reversible instance normalization (RevIN), a generally applicable normalization-and-denormalization method with learnable affine transformation.

Reversible Instance Normalization first normalizes the input sequences and then denormalizes the model output sequences to solve the time-series forecasting problems against distribution shift. RevIN is symmetrically structured to return the original distribution information to the model output by scaling and shifting the output in the denormalization layer in an amount equivalent to the shifting and scaling of the input data in the normalization layer. When adopted to the baselines, this method has significantly improved their forecasting performance and better aligns the distribution of the prediction results with the ground truth values.

Choice of Hyperparameters

- input_chunk_length – (aka lookback_period, context_length) The length of the input sequence fed to the model, which is processed at once, backcast. Suggested range of values to is 2-7 times the forecast length, is set to 48 (2 days) which is 4 times forecast length
- output_chunk_length – (aka forecast_length, prediction_length, forecast horizon) The length of the forecast of the model, set to 12 (1/2 day)
- num_stacks – as we are implementing interpretable mode, the architecture is two stacks - one for trend and one for seasonality.
- num_blocks – The number of blocks making up every stack. Suggested number for interpretable model is 3
- layer_widths – Determines the number of neurons that make up each fully connected layer in each block of every stack. set to [256, 2048] as recommended for interpretable model
- trend_polynomial_degree – The degree of the polynomial used as waveform generator in trend stacks. Set to 3 for cubic polynomial trend
- dropout – The dropout probability to be used in fully connected layers. This value is set to the default of 0.1, i.e., model is going to be highly selective

- activation – The activation function of encoder/decoder intermediate layer. Set to default 'ReLU'.
- loss_fn – PyTorch loss function used for training. Model is evaluated on Mean Squared Error Loss
- batch_size – Number of time series (input and output sequences) used in each training pass. Set to the default of 32
- n_epochs – Number of epochs over which to train the model. Each model is trained for 100 epochs