EP2420 Project 2 - Forecasting Service Metrics

Shriya Bhatija, Bo Zhang

December 23, 2022

Project Overview

Forecasting is the process of making predictions of the future based on past and present data which can be done by using different machine learning techniques. This project aims to forecast service metrics of infrastructure measurements, particularly of a Key-Value (KV) store and of a Video-on-Demand (VoD) service based on traces collected from a KTH testbed [1] (as used previously in project 1) and the FedCSIS 2020 challenge dataset. The FedCSIS trace is a publically available dataset that is provided by the analytics company EMCA Software. It contains around 2000 samples collected over a period of roughly three months. Each sample is aggregated from measurements over one hour, hence there are 24 samples per day. The data is gathered from several hosts and for this project we will use the statistics collected from one of the hosts. In regards to the KTH testbed, the service metrics of interest are $ReadsAvg$ for the KV service and $DispFrames$ for the VoD service. We concatenated the datasets JNSM_KV_flashcrowd_1 with JNSM_KV_flashcrowd_2 and JNSM_VoD_flashcrowd_1 with JNSM_VoD_flashcrowd_2. Hence, we now have a larger trace for each service and will use them for the means of forecasting service metrics.

The project is divided into four tasks with each of those having different emphasis'. However, the overall objective is to apply several machine learning models for forecasting and evaluate their performances. This report aims to describe the results obtained.

Background

A time series is a series of data points indexed in time order. Most commonly, it is a sequence taken at successive equally spaced points in time. What sets time series data apart from other data is that the analysis can show how variables change over time. In other words, time is a crucial variable because it shows how the data adjusts over its course. It provides an additional source of information and a set order of dependencies between the data. Time series analysis is a specific way of analysing a sequence of data points with the intention of extracting meaningful statistics and other characteristics of the data. Time series forecasting is the process of making predictions of the future based on past and present data. This can be done with different machine learning techniques as well as time series models. Given a type of model, one tries to fit it to historical data and use the resulting model to predict the future. In this project, we will use various types of models for forecasting service metrics. At the beginning of each section, we will briefly explain the relevant concepts and how we aim to perform forecasting.

Data Preparation

In order to enhance model performances, pre-processing and outlier removal will be applied. From the pre-processing methods discussed in project 1 we will chose the column-wise standardization since it gave the most promising results in regards to prediction accuracy. As for outlier removal, we will choose the threshold that will keep 99% of the samples in the dataset. This will allow us to remove significant outliers without harming the time-ordered structure of the datasets too much. Further, it is desired to reduce the dimensionality of the feature space by selecting the top 16 features. This will be done by fitting a random forest regressor (number of trees = 10, maximal depth =5) and applying the SelectFromModel() function. Having the feature space reduced will make training more efficient later on. Finally, we will perform a train-test split with a ratio of $70\% - 30\%$. Here, it is important to consider that the split shall not involve any shuffling as otherwise the time-ordered structure of the data will be destroyed. This data preparation stream will be applied to all traces and the resulting datasets will be used for all further tasks. Model fitting will always be performed with the training data whereas prediction and evaluation will be conducted with the test data.

Task I - Using Linear Regression for forecasting

In the first task, the main incentive is to formulate forecasting as a regression problem and evaluate the respective model performance. Forecasting may be expressed as a regression problem by making alternations to the respective dataset. Note that the dataset constists of samples of the form $(x^{(t)}, y^{(t)})$, that are collected at a timestamp t and do not take any previous or future measurements into consideration. In regression problems as in project 1, one does not assume the samples to be ordered by the timestamp t. However, for the purpose of forecasting it is important to preserve this order since now future predictions shall depend on the past ones.

In project 1, the objective for the regression problems was to find a model $M(\theta) : x \mapsto \hat{y}$ with parameter θ such that $\hat{y}$ closely approximates y for $x \in \mathbb{R}^n$ and $y \in \mathbb{R}$. This is attained by minimizing the loss function $\mathcal{L}\{(M(\theta, x^{(t)}), y^{(t)})\}$ given a trace $\mathcal{T} = \{(x^{(t)}, y^{(t)})\}$. For forecasting the regression problem will be expressed as follows: Given a *lag* value $l = 0, 1, 2...$ and a *horizon* value $h = 0, 1, 2, ...$, the goal is to find a model $M(\theta, h, l) : [x_{-l}, \dots, x_0] \mapsto [\hat{y}_0, \dots, \hat{y}_h]$ with parameter θ such that $\hat{y} = [\hat{y}_0, \dots, \hat{y}_h]$ closely approximates $y = [y_0, \dots, y_h]$ for $x \in \mathbb{R}^{n,l+1}$ and $y \in \mathbb{R}^{h+1}$. This model will be trained on new traces

$$\mathcal{T}_{new} = \{([x^{(l-1)}, \dots, x^{(t)}], [y^{(t)}, \dots, y^{(t+h)}])\}$$

in order to make predictions given past observations.

Task I.2: Data Alternation

We need to form new training and test sets by constructing new samples of the form $([x^{(l-1)}, \dots, x^{(t)}],$ $[y^{(t)}, \dots, y^{(t+h)}])$ given lag l and horizon h. For the KV service, h is incremented and l is decremented by 1 second intervals between samples. The same is done for the VoD service but we go through l and h by a time step $s = 30$. For this, I defined functions that receive l, h (and s) along with the measurements/observations as input and return the transformed data.

Task I.3: Fitting linear regression

Using linear regression, we will train models for data constructed with different lag values $l = 0, 1, \dots 10$ and a fix horizon $h = 10$. In the model with $l = 0$ predictions are made using the current sample whereas a model with $l > 0$ corresponds to learning on $l + 1$ samples and predicting 10 steps into the future. Next, we are going to evaluate these models by computing the NMAE on the test set with targets $[y^{(t)}, \dots, y^{(t+10)}]$. Note that we can even obtain the error for different horizon values by comparing the first h columns on the test set to the corresponding ones on the predictions. The following tables display the results for the KV service as well as for the VoD service:

	0	1	2	3	4	5	6	7	8	9	10
0	2158	2195	2181	2184	2171	2157	2155	2151	2151	2138	2138
1	2144	2183	2186	2179	2173	2159	2153	2149	2147	2140	2135
2	2128	2162	2181	2181	2172	2163	2154	2148	2145	2139	2136
3	2123	2147	2167	2179	2175	2164	2158	2150	2145	2139	2136
4	2138	2142	2155	2168	2172	2166	2158	2152	2145	2139	2135
5	2165	2153	2149	2157	2163	2162	2158	2150	2145	2138	2133
6	2197	2176	2158	2152	2153	2155	2154	2149	2143	2137	2132
7	2226	2201	2177	2159	2147	2145	2146	2144	2140	2134	2129
8	2259	2230	2202	2177	2155	2142	2139	2139	2138	2133	2128
9	2288	2261	2229	2200	2172	2149	2136	2133	2133	2130	2126
10	2316	2289	2258	2226	2194	2166	2144	2131	2129	2127	2124

Table 1: NMAE * 10^5 for the linear regression models applied on the KV service given different lag values l and horizon values h. Rows correspond to the horizon values and columns to the lag values.

	0	1	2	3	4	5	6	7	8	9	10
0	1292	1303	1294	1299	1298	1310	1314	1321	1329	1339	1350
1	1291	1290	1293	1288	1292	1298	1306	1314	1321	1329	1339
2	1328	1290	1291	1292	1290	1296	1301	1311	1319	1326	1334
3	1369	1305	1289	1288	1290	1291	1296	1303	1313	1320	1328
4	1407	1335	1303	1286	1286	1290	1292	1299	1307	1315	1323
5	1443	1375	1333	1298	1284	1286	1290	1295	1303	1310	1319
6	1472	1413	1370	1324	1294	1285	1288	1294	1299	1306	1314
7	1494	1445	1408	1360	1320	1296	1288	1293	1299	1304	1312
8	1511	1472	1440	1397	1355	1320	1298	1293	1299	1304	1310
9	1518	1490	1464	1426	1389	1351	1319	1301	1299	1304	1310
10	1525	1503	1484	1451	1418	1384	1349	1321	1307	1304	1309

Table 2: NMAE * 10^4 for the linear regression models applied on the VoD service given different lag values l and horizon values h. Rows correspond to the horizon values and columns to the lag values.

Task I.4: Discussion

In this section we are going to discuss Table 3 and Table 4 with the evaluation results and draw conclusions. Given the results for the KV service it is surprising to discover that the error does not vary too much given different lag values and horizon values. Most of the time, the NMAE is distributed between the values 0.021 and 0.022. Intuitively, one would expect that the error roughly increases with fixed l and increasing h since the future is more uncertain the further one moves in time. This however, holds only true for $l = 0, 1, 2$ and other than that Table 3 does not show any clear relation between the NMAE and the different lag values. This further implies that the model accuracy is not significantly impacted/worsened when performing forecasting which is desired. Moreover, it would be reasonable to expect that the error decreases with fixed h and increasing l since we saw in project 1 that ReadsAvg is locally stable. This is indeed the case here.

The situation looks similar for the VoD service for which we indeed can see a relation between lag/horizon values and NMAE. For low l we can observe that the NMAE increases with increasing h. This is somewhat to be expected since we are then trying to predict far into the future while only considering the recent past. Hence, accurate forecasting shows to be difficult in this case. However, if one takes the more distant past into account, the error decreases again. Therefore, one should only make predictions for the distant futures if there is information about the distant past available. Further, one can observe that the NMAE generally decreases with increasing l and fixed h which is to be expected since more information about the past will lead to better predictions.

All in all it can be said that for the KV service as well as for the VoD service accurate forecasting requires us to take the more distant past into account. This is probably the case, because ReadsAvg and DispFrames do not differ too much in their values when considering small local regions. It would also be interesting to consider lag and horizon values of larger magnitude since $l, h = 0, \ldots, 10$ are merely a tiny fraction of the entire traces. Moreover, it is worth noticing that the error is more than 10% lower on the KV trace as opposed to the VoD trace. This might be due to the fact that DispFrames has higher fluctuations that appear to be random in nature. However, this is only speculation and further studies may reveal the exact reason behind this observation.

Task II - Using Recurrent Neural Networks for Forecasting

In this task, the main incentive is to again formulate forecasting as a regression problem by training a so called $RecurrentNeuralNetwork$ and making predictions with the obtained model. Recurrent Neural Networks form a class of network architectures that allow for sequence-to sequence learning. In contrast to classical feed-forward networks, they allow cycles between nodes such that past information can affect future predictions. In order to regulate the flow of information, a specific version of RNNs called $Long\ Short-Term\ Memory$ (LSTM) can be utilized. It enables the network to have a short term memory that can last up to thousands of time steps making LSTMs widely used in machine translation and speech recognition amongst other fields. In our case, we want to build a non-linear LSTM model that can forecast the target variables at specific points in time. Hence, the model should learn to map a sequence of input states $(x^{(l-1)}, \ldots, x^{(t)})$ to a sequence of output states $(y^{(t)}, \ldots, y^{(t+h)})$, where l is the lag and h the horizon value.

Task II.2 Data Alternation

We need to form new training and test sets by constructing new samples of the form $([x^{(l-1)}, \ldots, x^{(t)}],$ $[y^{(t)}, \ldots, y^{(t+h)}])$ given lag l and horizon h. For the KV service, h is incremented and l is decremented by 1 second intervals between samples. The same is done for the VoD service but we go through l and h by a time step $s = 30$. For this, I defined functions that receive l, h (and s) along with the measurements/observations as input and return the transformed data with the shape given by (#samples, #time steps, #features). This data shape is needed for training the LSTM model.

Task II.3: Fitting an LSTM model

We will train LSTM models for data constructed with different lag values $l = 0, 1, \ldots 10$ and a fix horizon $h = 10$. Using the Keras library for Python, we defined a sequential model and added an LSTM layer with 50 units with tanh as the output activation and sigmoid used as the gating function. Further, we add a dense layer with 11 units because we want the network to forecast 11 target values. The models were trained using the Adam optimization technique and the mean squared error as loss function. Training takes place over 50 epochs and with a batch size of 72.

In the model with $l = 0$ predictions are made using the current sample whereas a model with $l > 0$ corresponds to learning on $l + 1$ samples and predicting 10 steps into the future. Next, we are going to evaluate these models by computing the NMAE on the test set with targets $[y^{(t)}, \ldots, y^{(t+10)}]$. Note that we can even obtain the error for different horizon values by comparing the first h columns on the test set to the corresponding ones on the predictions. The following tables display the results for the KV service as well as for the VoD service:

	0	1	2	3	4	5	6	7	8	9	10
0	2163	2206	2209	2275	2274	2294	2288	2231	2312	2419	2232
1	2154	2182	2197	2251	2255	2277	2260	2223	2299	2395	2215
2	2147	2170	2184	2230	2237	2257	2241	2212	2289	2390	2199
3	2148	2165	2177	2215	2219	2243	2227	2215	2275	2383	2189
4	2161	2171	2176	2206	2203	2232	2219	2221	2263	2384	2184
5	2178	2185	2187	2204	2202	2225	2216	2219	2252	2390	2182
6	2198	2201	2201	2212	2205	2223	2213	2218	2241	2396	2185
7	2219	2219	2219	2225	2212	2224	2217	2219	2230	2401	2187
8	2240	2237	2236	2237	2222	2231	2220	2222	2222	2402	2191
9	2260	2257	2254	2251	2236	2244	2229	2228	2219	2407	2194
10	2279	2273	2271	2265	2249	2257	2240	2239	2222	2413	2196

Table 3: NMAE * 10^5 for the LSTM models applied on the KV service given different lag values l and horizon values h. Rows correspond to the horizon values and columns to the lag values.

	0	1	2	3	4	5	6	7	8	9	10
0	1131	1135	1142	1123	1104	1097	1099	1099	1106	1109	1104
1	1131	1133	1142	1137	1112	1106	1099	1097	1103	1099	1101
2	1129	1134	1144	1137	1117	1111	1104	1099	1100	1099	1102
3	1128	1134	1146	1136	1121	1117	1110	1106	1102	1101	1102
4	1127	1133	1143	1138	1124	1122	1115	1110	1106	1106	1103
5	1130	1133	1144	1137	1126	1124	1120	1117	1110	1111	1104
6	1134	1136	1145	1137	1128	1127	1123	1123	1116	1114	1107
7	1137	1138	1147	1138	1129	1129	1125	1126	1120	1118	1111
8	1141	1141	1150	1141	1130	1130	1127	1128	1124	1124	1114
9	1144	1144	1152	1143	1132	1131	1128	1131	1129	1130	1120
10	1146	1146	1154	1145	1135	1134	1130	1133	1132	1136	1125

Table 4: NMAE * 10^4 for the LSTM models applied on the VoD service given different lag values l and horizon values h. Rows correspond to the horizon values and columns to the lag values.

Task II.4: Discussion

In this section we are going to discuss Table 3 and Table 4 with the evaluation results and draw conclusions. Given the results for the KV service it is surprising to discover that the error does not vary too much given different lag and horizon values. Most of the time, the NMAE is distributed between the values 0.021 and 0.023. Intuitively, one would expect that the error roughly increases with fixed l and increasing h since the future is more uncertain the further one moves in time. This however, does not generally hold true here. It further implies that the model accuracy is not significantly impacted/worsened when performing forecasting which is desired. Moreover, it would be reasonable to expect that the error decreases with fixed h and increasing l since we saw in project 1 that the values of ReadsAvg are stable in certain regions. This does seem to be the case for higher horizon values. Comparing these results with the ones obtained from linear regression, it is surprising to discover that the LSTM model performs slightly worse on average since for linear regression the NMAE was mostly distributed between the values 0.021 and 0.022. One would actually expect the RNN to have higher model capacity and hence make more accurate predictions. Perhaps, we would need to perform hyperparameter tuning for better model performance since LSTMs should indeed be able to effectively forecast time-series.

The situation looks different for the VoD service for which we indeed can see a relation between lag/horizon values and NMAE. We can observe that the NMAE increases with increasing h and fixed l. This is somewhat to be expected since the future is more uncertain the further one moves in time. Hence, accurate forecasting into the distant future shows to be difficult in this case. Further, one can observe that the NMAE generally decreases with increasing l and fixed h which is to be expected since more information about the past will lead to better predictions. Observe that the LSTM model performs better on average than linear regression from Task I since for Linear Regression the NMAE was mostly distributed between the values 0.012 and 0.015. This is to be expected since linear regression tends to have a high bias due to the underlying linearity of the model

All in all it is surprising to discover that our LSTM model performs better only for the VoD service and slightly worse for the KV service data as compared to linear regression. One can analyse the data structures in more detail to unravel why this might be the case. It is also entirely possible that the chosen hyperparameters define a model that is not fully capable of capturing the complex behavior of the values for ReadsAvg. Further studies can be conducted on this matter.

Task III - Time Series Analysis

In this task we are going to apply traditional univariate time-series analysis methods to the measured service metrics. This will lay the foundation for fitting different time-series models in the next task. Now, we will briefly introduce the relevant notions that will be needed in the upcoming tasks as stated in [2]. When analysing a time series, it is at first important to select a suitable probability model for the data by interpreting y_t as the realized value of some random variable Y_t over time t. This motivates the following definitions:

Definition. A *time series model* for the observed data $\{y_t\}$ is a specification of the joint distributions (or possibly only the means and covariances) of a sequence of random variables $\{Y_t\}$ of which $\{y_t\}$ is postulated to be a realization.

An interesting property of time series is the so called (*weak*) *stationarity*, which occurs when the time series has certain statistical properties similar to those of the time-shifted series. A rigorous definition can be stated as follows:

Definition. A time series $\{Y_t\}$ is called (*weakly*) *stationariy* if

(i) $E(Y_t)$ is independent of t;

(ii) $\mathrm{Cov}(X_{t+h}, X_t)$ is independent of t for each integer h.

Testing whether a time series is stationary can be done via the so called $Augmented\ Dicky - Fuller\ (ADF)\ test$ which evaluates the null hypothesis that a unit root is present in the respective time series. The alternative hypothesis, i.e. the time series does not have a unit root, points towards stationarity. The results of the test will indicate to wich degree the null hypothesis may be accepted or rejected. We will later apply this test with the pre-written `adfuller` function in the `statsmodels.tsa.stattools` library. Further explanations on how the test is conducted will follow with the specific examples.

In the last part of this report we will plot the so called $autocorrelation\ function\ (ACF)$ of our time series. It enables us to evaluate the correlation of present measurements with future ones. Formally, it is defined as follows:

Definition. The $autocorrelation\ function\ (ACF)$ of a time series $\{Y_t\}$ at lag h is given by

$$\rho_Y(h) = \mathrm{Cor}(X_{t+h}, X_t).$$

Note that $\mathrm{Cor}(X_{t+h}, X_t)$ is independent of t for each h, hence the above definition is well-defined.

Task III.3: Augmented Dicky-Fuller Test for evaluating Stationarity

The Augmented Dicky-Fuller test is one of the wider used statistical tests to evaluate the stationarity of time series. Firstly, we will create new traces by choosing a random start point t_0 of our existing traces and taking sequences each of 50, 500 and 5000 values. Here, we choose $t_0 = 1000$. On these new traces we apply the ADF test to infer whether these are stationary. The outcome of the tests is displayed in the table below:

ADF Statistic	-5.673
p - value	0.000
Critical Value: 1%	-3.571
Critical Value: 5%	-2.923
Critical Value: 10%	-2.599

(a) Sequence with 50 values

ADF Statistic	-2.810
p - value	0.056
Critical Value: 1%	-3.444
Critical Value: 5%	-2.867
Critical Value: 10%	-2.570

(b) Sequence with 500 values

ADF Statistic	-7.580
p - value	0.000
Critical Value: 1%	-3.432
Critical Value: 5%	-2.862
Critical Value: 10%	-2.567

(c) Sequence with 5000 values

Table 5: Results of the Augmented Dicky-Fuller test on sequences of ReadsAvg values (KV service) starting at $t_0 = 1000$ and having length 50, 500 and 5000.

ADF Statistic	-5.105
p - value	0.000
Critical Value: 1%	-3.571
Critical Value: 5%	-2.923
Critical Value: 10%	-2.599

(a) Sequence with 50 values

ADF Statistic	-4.111
p - value	0.056
Critical Value: 1%	-3.444
Critical Value: 5%	-2.868
Critical Value: 10%	-2.570

(b) Sequence with 500 values

ADF Statistic	-5.779
p - value	0.000
Critical Value: 1%	-3.432
Critical Value: 5%	-2.862
Critical Value: 10%	-2.567

(c) Sequence with 5000 values

Table 6: Results of the Augmented Dicky-Fuller test on sequences of DispFrames values (VoD service) starting at $t_0 = 1000$ and having length 50, 500 and 5000.

In order to discuss on the results of the ADF tests displayed in the table above, let us denote the trace of length $i \in \{50, 500, 5000\}$ corresponding to the KV service (VoD service) by $\text{trace}_{\text{KV}}(i)$ ($\text{trace}_{\text{VoD}}(i)$). For the chosen significance level of 5% the results of of the tests have to be interpreted as follows: The ADF Statistic value tells us the likehood of rejecting the null hypothesis, i.e. the more negative this values becomes, the more likely we are to have a stationary time series. If the ADF Statistic is less than the value at the critical value of 5%, then it indicated that we can reject the null hypothesis with a significance level of less than 5%. According to this we make the following conclusions about stationarity:

Table 5 yields the following results for the KV service data:

1. $\text{trace}_{\text{KV}}(50)$ is stationary with a significance level of 5% (even with a significance level of 1%).
2. For $\text{trace}_{\text{KV}}(500)$ we accept the null hypothesis and conclude that this time series is slightly more likely to be non-stationary.
3. $\text{trace}_{\text{KV}}(5000)$ is stationary with a significance level of 5% (even with a significance level of 1%).

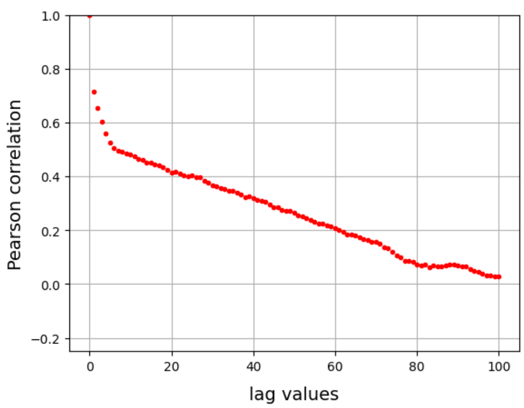
Table 6 yields the following results for the VoD service data:

1. $\text{trace}_{\text{VoD}}(50)$ is stationary with a significance level of 5% (even with a significance level of 1%).
2. $\text{trace}_{\text{VoD}}(500)$ is stationary with a significance level of 5% (even with a significance level of 1%).
3. $\text{trace}_{\text{VoD}}(5000)$ is stationary with a significance level of 5% (even with a significance level of 1%).

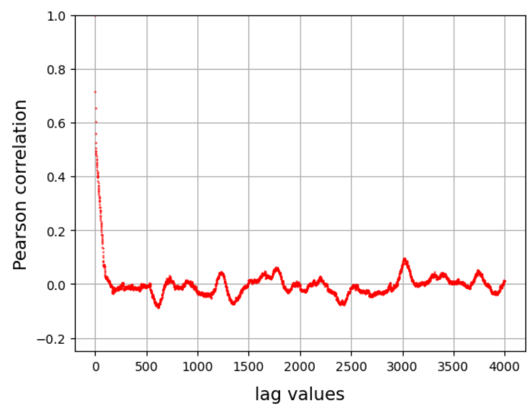
Note that in all cases in which the null hypothesis was rejected, we can even conclude that the corresponding traces are stationary even with a significance level of 1%. This means that they are with very high probability indeed stationary. However, these results are actually neither surprising nor entirely expected. From Project 1, we know that both target values, ReadsAvg and DispFrames, may have periods of stationarity and periods of high fluctuations. The results in the above tables will therefore depend on the length of the sequence as well as on the start point t_0. One may confirm these results by roughly examining the corresponding sequences on a visual level.

Task III.4-5: Autocorrelation functions

In this task we compute the autocorrelation function (ACF) for the KTH trace as well as for the FedCSIS trace.



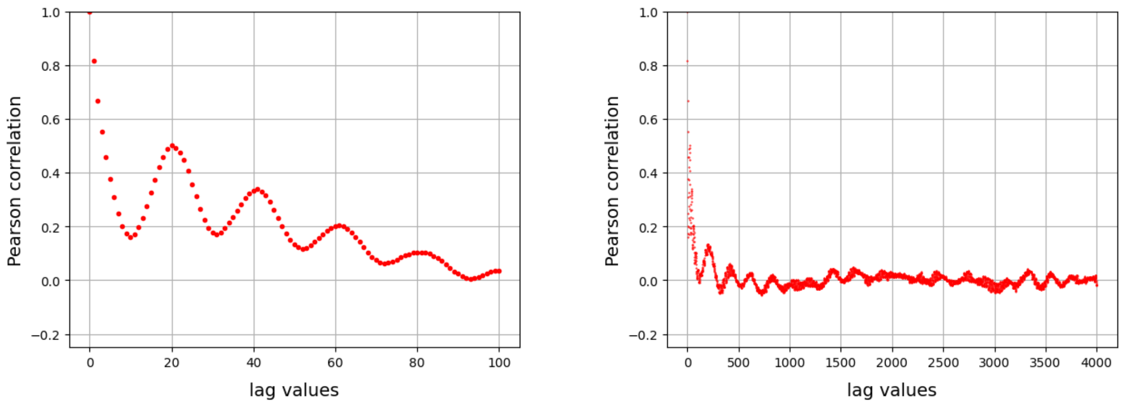
(a) Lag values in [0,100]



(b) Lag values in $[0, 4000]$

Figure 1: Correlogram for ReadsAvg values (KV service) for the lag values in the interval $[0, 100]$ and $[0, 4000]$. The x-axis shows the lag value and the y-axis shows the correlation coefficients (Pearson correlation).

For the KV trace, the autocorrelation functions behave very differently depending on the lag intervals. In the case of the lag values being contained in [0,100] we observe that the function decreases linearly from the beginning until it gets very close to the value 0 at $l = 100$. A decrease is to be expected since it means that observations that are further apart in time are less likely to be correlated. Moreover, the correlation is positive for all lag values indicating that sequences of length 100 are likely to have a trend. In the case of l taking values between 0 and 4000, note that the correlation drops exponentially in the beginning which is to be expected from the previous plot. After this drop it is to be noted that there are zero-centered fluctuations in function values without indication of periodicity. These fluctuations of correlation values are contained within the interval [-0.1,0.1]. This indicates that in the long term, the time series is not likely to have a trend. After evaluating these plots it is difficult to say which time series model and which parameters may perform best here.

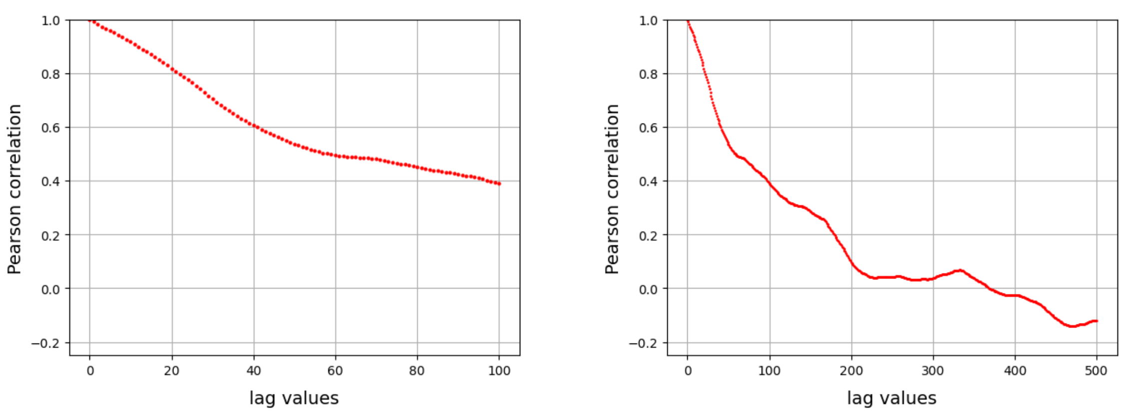


(a) Lag values in [0,100]

(b) Lag values in [0,4000]

Figure 2: Correlogram for DispFrames values (VoD service) for the lag values in the interval $[0, 100]$ and $[0, 4000]$. The x-axis shows the lag value and the y-axis shows the correlation coefficients (Pearson correlation).

In the case of the VoD service, the autocorrelation functions again behave a bit differently depending on the lag intervals. In the plot with the lag values between 0 and 100 we observe periodic behavior with period length 20 where the curves slowly flatten over time until they reach correlation 0. This suggests that the time series might have seasonality over short sequences. Moreover, all correlation values are positive indicating that sequences up to a length of 100 are likely to have a trend. When examining the autocorrelation function with lag values between 0 and 4000, we again observe a sudden drop in the beginning which is again to be expected from the previous case. After that drop we observe zero-centered fluctuations that seem to be of period behavior indicating that the time series might have seasonality over long periods of time. At this point it is again difficult to say which time series model and which parameters may perform best here.



(a) Lag values in [0,100]

(b) Lag values in [0,500]

Figure 3: Correlogram corresponding to the FedCSIS trace for the lag values in the interval $[0, 100]$ and $[0, 500]$. The x-axis shows the lag value and the y-axis shows the correlation coefficients (Pearson correlation).

Lastly, we examine the autocorrelation functions for the FedCSIS data trace. When the lag takes values in [0,100], we observe that the correlation values decrease linearly until they reach approximately the value 0.4 at $l = 100$. Since all function values are significantly positive here, it is reasonable to assume that sequences up to a length 100 are likely to have a trend. In the other case, with lag values between 0 and 500, we see a exponential decrease of correlation values where up until $l = 100$ all values are positive. In this case, the AR model with parameter $p = 1$ may be a good fit for our time series.

Task IV - Time Series Forecasting

In this task, we will fit several statistical models to our time series of service metrics in order to perform forecasting. In the beginning of each section, there will be a very brief description of the upcoming models. For a detailed explanation of these time series models, we refer to [2].

As previously, we are interested in using the last $l = 1, 2, ..., 10$ measurements in order to predict $h = 1, 2, ..., 10$ steps into the future. For the KV service, h is incremented and l is decremented by 1 second intervals between samples. The same is done for the VoD service but we go through l and h by a time step of 30 seconds. In the case of the FedCSIS data the step size is one hour.

Task IV.1: Autoregression

In an autoregression model, we forecast the service metric at the next time step by using a linear combination of previous measurements. The term autoregression indicates that it is a regression of the target values against themselves. Formally, an autoregressive model of order p predicts as follows:

$$Y_t = C + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + ... + \phi_p Y_{t-l} + \varepsilon_t,$$

where ε_t is white noise and $C, \phi_1, ..., \phi_l$ are the to be learned parameters. We denote an autoregressive model of order l as AR(l). This method is suitable for time series without trend or seasonal components. Therefore, it should be most successful for the KV service since it does not have a trend nor seasonality. We fitted autoregression models to the training sets of the different traces and performed forecasting on the test sets given different lag and horizon values. The results are displayed in the tables below:

	1	2	3	4	5	6	7	8	9	10
1	2389	2239	2188	2172	2165	2159	2150	2140	2139	2137
2	2554	2350	2287	2265	2253	2240	2227	2214	2210	2206
3	2712	2446	2373	2344	2327	2308	2289	2273	2267	2261
4	2861	2542	2454	2418	2394	2368	2345	2324	2315	2308
5	2994	2633	2529	2486	2456	2424	2395	2370	2358	2350
6	3108	2717	2597	2547	2511	2473	2439	2410	2395	2385
7	3207	2796	2662	2604	2563	2519	2480	2446	2428	2415
8	3292	2868	2723	2659	2612	2563	2520	2481	2459	2444
9	3364	2936	2781	2712	2660	2607	2558	2515	2489	2472
10	3427	3000	2836	2762	2707	2648	2596	2548	2519	2500

Table 7: NMAE * 10^5 for the Autoregression models applied on the KV service given different lag values l and horizon values h. Rows correspond to the horizon values and columns to the lag values.

	1	2	3	4	5	6	7	8	9	10
1	1506	1503	1508	1508	1508	1508	1508	1509	1511	1516
2	1478	1478	1483	1482	1483	1482	1483	1483	1486	1491
3	1538	1538	1542	1542	1542	1542	1542	1543	1546	1552
4	1563	1564	1567	1567	1568	1567	1567	1568	1573	1578
5	1573	1574	1577	1577	1577	1577	1577	1578	1582	1588
6	1575	1576	1578	1578	1578	1578	1577	1579	1583	1589
7	1575	1576	1578	1578	1578	1578	1578	1579	1584	1589
8	1576	1577	1579	1578	1578	1578	1578	1579	1584	1589
9	1578	1579	1581	1580	1580	1580	1580	1581	1586	1591
10	1580	1582	1584	1583	1583	1583	1582	1584	1588	1594

Table 8: NMAE * 10^4 for the Autoregression models applied on the VoD service given different lag values l and horizon values h. Rows correspond to the horizon values and columns to the lag values.

In case of the KV service (Table 7) as well as for the VoD service (Table 8), we notice that the error increases noticeably with increasing horizon values. This trend is similarly strong for both services (note that for the KV service we multiplied the errors by 10^5 and for the VoD service with 10^4). This means that the prediction accuracy is worse when we are aiming to predict further into the future which is of course to be expected. On the other hand, the error decreases for the KV service with increasing lag values, meaning that having more information about the past will lead to better predictions. For the VoD service we observe the opposite which is surprising. We conclude that for the KV trace, the most suitable model is AR(10) whereas AR(1) seems most promising for the VoD service. Moreover, it is also interesting to discover that the model performance is generally worse than of the Linear Regression and the LSTM model. Those models however take the measurements into account whereas autoregression does not. If there is a decent correlation between measurements and service metrics, it is to be expected that Linear Regression and LSTM will perform better. Further studies may reveal as to how large or small the correlation between measurements and service metrics may be.

For the FedCSIS trace (Table 9) we make similar observation as for the VoD trace. The error increases noticeably with increasing horizon values, indicating that prediction into the more distant future yields worse accuracy. On the other hand, the error increases slightly when taking the more distant past into account. This might indicate that only the very recent past is indicative for future values. Further, we observe that the NMAE values generally lie between 0.02 and 0.07 which is better than the model performance for the VoD service but worse than for the KV service.

	1	2	3	4	5	6	7	8	9	10
1	2297	2275	2280	2284	2283	2278	2283	2279	2285	2284
2	2996	3025	3011	3015	3013	3006	3005	3005	3017	3016
3	3596	3669	3642	3649	3646	3640	3634	3634	3646	3646
4	4108	4216	4178	4188	4184	4178	4174	4175	4187	4187
5	4551	4680	4636	4648	4642	4641	4641	4641	4652	4653
6	4987	5123	5078	5090	5083	5087	5091	5092	5105	5105
7	5397	5525	5482	5495	5487	5493	5499	5501	5512	5513
8	5789	5905	5864	5877	5870	5875	5883	5886	5895	5896
9	6164	6272	6234	6247	6239	6245	6251	6253	6259	6261
10	6524	6628	6590	6603	6596	6601	6607	6606	6610	6612

Table 9: NMAE * 10^5 for the Autoregression models applied on the FedCSIS trace given different lag values l and horizon values h. Rows correspond to the horizon values and columns to the lag values.

Task IV.2: Moving Average

A Moving Average model uses the weighted moving average of the past few forecast errors in a regression-like model as follows:

$$Y_t = C + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + ... + \theta_l \varepsilon_{t-l},$$

where ε_t is white noise, $C, \phi_1, ..., \phi_l$ are the to be learned parameters and $\varepsilon_{t-1}, ..., \varepsilon_{t-l}$ are the previous forecast errors. We denote a moving average model of order l by MA(l). This method is suitable for time series without trend or seasonal components. Therefore, it should be most successful for the KV service since it does not have a trend nor seasonality. We fitted moving average models to the training sets of the different traces and performed forecasting on the test sets given different lag and horizon values. The results are displayed in the tables below:

	1	2	3	4	5	6	7	8	9	10
1	3126	2773	2610	2492	2412	2379	2351	2319	2295	2276
2	3126	2773	2609	2492	2412	2378	2350	2319	2295	2275
3	3126	2773	2609	2491	2412	2378	2350	2319	2295	2275
4	3126	2773	2610	2492	2412	2378	2350	2319	2295	2276
5	3126	2773	2610	2492	2412	2378	2350	2319	2295	2276
6	3127	2774	2610	2492	2412	2378	2350	2319	2296	2276
7	3127	2774	2610	2492	2412	2379	2351	2319	2296	2276
8	3127	2774	2610	2492	2412	2379	2351	2319	2296	2276
9	3127	2774	2610	2492	2412	2379	2351	2319	2296	2276
10	3127	2774	2610	2492	2412	2379	2351	2319	2296	2276

Table 10: NMAE * 10^5 for the Moving Average models applied on the KV service given different lag values l and horizon values h. Rows correspond to the horizon values and columns to the lag values.

For the KV servie (Table 10) notice that the errors are almost of the same values given different horizon values and fixed lag values. This is not however a model mistake and the errors do differ but this is only to a very small degree. Further, we observe that the error decreased with increasing lag values which indicates that information about the more distant errors is valuable for future predictions. The errors are of similar magnitude as in the case of autoregression. All in all we conclude that MA(10) is the most promising model for the KV trace.

	1	2	3	4	5	6	7	8	9	10
1	1551	1561	1612	1664	1698	1722	1743	1764	1780	1797
2	1546	1589	1650	1701	1732	1753	1772	1792	1810	1826
3	1584	1636	1692	1736	1761	1777	1794	1813	1830	1845
4	1612	1667	1717	1754	1776	1791	1808	1827	1845	1860
5	1626	1677	1722	1756	1778	1795	1812	1833	1851	1866
6	1628	1677	1721	1755	1778	1797	1816	1837	1856	1871
7	1628	1676	1720	1756	1780	1799	1819	1842	1860	1875
8	1629	1678	1723	1761	1786	1806	1826	1849	1867	1882
9	1633	1684	1732	1770	1796	1815	1835	1857	1875	1890
10	1640	1694	1742	1780	1805	1824	1843	1864	1882	1897

Table 11: NMAE * 10^4 for the Moving Average models applied on the VoD service given different lag values l and horizon values h. Rows correspond to the horizon values and columns to the lag values.

For the VoD service (Table 11), notice that the error increases significantly for increasing lag values. This is to be expected since the autocorrelation for this trace (task III) drops significantly after $l = 1$ indicating that MA(1) might be the most suitable model for the VoD service data.

	1	2	3	4	5	6	7	8	9	10
1	11051	7470	5462	4641	3978	3648	3382	3257	3180	2986
2	11046	7468	5461	4641	3978	3648	3383	3256	3177	2985
3	11052	7473	5462	4642	3981	3650	3386	3256	3176	2983
4	11053	7473	5462	4642	3982	3650	3388	3256	3175	2981
5	11058	7476	5462	4641	3983	3649	3388	3254	3173	2980
6	11092	7508	5492	4672	4015	3680	3419	3286	3204	3012
7	11138	7566	5554	4734	4076	3741	3478	3345	3260	3067
8	11174	7621	5609	4783	4121	3787	3526	3392	3308	3116
9	11216	7670	5655	4828	4162	3831	3570	3434	3349	3157
10	11251	7717	5701	4862	4196	3867	3607	3469	3383	3190

Table 12: NMAE * 10^5 for the Moving Average models applied on the FedCSIS trace given different lag values l and horizon values h. Rows correspond to the horizon values and columns to the lag values.

In the case of the FedCSIS data (Table 12), the errors are significantly larger than in the autoregression setting. However, they decrease significantly with increasing lag values and we therefore expect that MA(10) will be the most promising model here.

Task IV.3: Autoregressive Integrated Moving Average

The Autoregressive Integrated Moving Average (ARIMA) combines the autoregression and moving average methods and applies these methods to data where it has previously removed seasonality by taking differences of past measurements. Formally, predictions are made as follows:

$$Y_t = C + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + ... + \theta_q \varepsilon_{t-q} + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + ... + \phi_p Y_{t-p},$$

where Y_t is the differenced series (it may have been differenced more than once). An ARIMA model depends on the parameters (p, d, q), where p is order of the autoregressive part, d is degree of first differencing and q is order of the moving average part. For our analysis, we will fix $d = 1$ and let $p = 1, ..., 10$, $q = 1, ..., 5$. In order to better understand the relation between p and q, we will fix a horizon $h = 5$ and display the error matrices with respect to p and q rather than h. The results obtained are displayed in the below tables.

	1	2	3	4	5	6	7	8	9	10
1	2170	5407	4666	4154	3600	3017	2508	3854	2894	2512
2	3490	2321	15358	9770	8990	3526	10827	11780	9031	5676
3	3565	3017	3233	18716	5115	3664	4328	4892	5066	3677
4	3546	3130	4498	2358	3029	3498	3338	5489	3905	4103
5	3561	5845	3217	5577	2420	4138	5647	5131	3979	4495

Table 13: NMAE * 10^5 for the ARIMA models applied on the KV service given different values for p and q. Rows correspond to the q values and columns to the p values.

	1	2	3	4	5	6	7	8	9	10
1	2269	2341	2609	2752	2744	2641	3016	2362	2438	2344
2	2364	2279	3731	3044	3267	3411	3322	3063	2888	3078
3	2410	2392	2290	2829	3508	4217	3947	3355	3195	3392
4	2402	2423	4620	3246	2355	4071	3745	3396	3251	3488
5	2353	2422	2448	3992	2287	2359	4373	3398	3531	3621

Table 14: NMAE * 10^5 for the ARIMA models applied on the FedCSIS trace given different values for p and q. Rows correspond to the q values and columns to the p values.

Since our ARIMA model depends on two parameters, p and q, it is interesting to study whether there is any relationship between them. For our traces there does not seem to be any clear relation between those parameters. For the KV service the errors fluctuate within a range of 0.005 given different values for p and q. In the case of the FedCSIS trace there is no such fluctuation but the error is lower than with the autoregression and moving average models. This goes to show that ARIMA has the capacity to achieve promising forecasting results. For both traces ARIMA(1,1,1) performs best amongst the considered models.

Task IV.4: Exponential Smoothing

The exponential smoothing method uses the weighted average of past observations to forecast new values given a smoothing factor $\alpha \in (0, 1)$. Formally, a prediction is made as follows:

$$Y_t = \alpha Y_{t-1} + \alpha(1-\alpha)Y_{t-2} + ... + \alpha(1-\alpha)^{l-1} Y_{t-l}.$$

We denote the exponential smoothing method with lag l as ES(l). There is the possibility of fitting the exponential smoothing method to the training data and allowing the model to learn the optimal α. However, here we will fix $\alpha = 0.5$ and perform forecasting given the above formula. The results are displayed in the tables below.

	1	2	3	4	5	6	7	8	9	10
1	5000	2500	1250	634	352	249	223	218	217	218
2	6250	3125	1563	788	424	279	236	227	226	227
3	7083	3750	1875	943	498	310	250	236	234	234
4	7656	4297	2188	1098	572	343	264	243	240	240
5	8063	4781	2485	1254	648	378	278	251	245	245
6	8359	5209	2767	1406	723	413	293	257	250	249
7	8583	5586	3036	1556	799	448	308	264	254	252
8	8755	5921	3292	1702	874	483	323	270	257	255
9	8891	6218	3534	1845	948	519	338	277	261	258
10	9001	6482	3765	1985	1022	555	354	283	265	261

Table 15: NMAE * 10^4 for the Exponential Smoothing method applied on the KV service given different lag values l and horizon values h. Rows correspond to the horizon values and columns to the lag values.

	1	2	3	4	5	6	7	8	9	10
1	5067	3071	2249	1864	1677	1585	1540	1517	1506	1500
2	6291	3546	2470	1985	1750	1635	1579	1551	1537	1530
3	7112	4056	2729	2145	1863	1725	1657	1624	1607	1599
4	7678	4537	2954	2267	1938	1777	1699	1660	1641	1631
5	8080	4978	3159	2371	1994	1809	1720	1676	1654	1643
6	8374	5374	3355	2465	2039	1832	1731	1681	1657	1645
7	8595	5728	3550	2558	2086	1855	1744	1689	1662	1648
8	8765	6045	3748	2652	2135	1882	1759	1699	1669	1654
9	8900	6328	3944	2749	2189	1915	1781	1716	1683	1667
10	9009	6581	4136	2845	2245	1950	1806	1735	1700	1682

Table 16: NMAE * 10^4 for the Exponential Smoothing method applied on the VoD service given different lag values l and horizon values h. Rows correspond to the horizon values and columns to the lag values.

	1	2	3	4	5	6	7	8	9	10
1	5001	2501	1263	668	416	333	309	303	301	301
2	6251	3127	1575	827	506	398	368	361	360	359
3	7084	3752	1885	982	589	455	419	412	410	410
4	7657	4299	2196	1136	667	506	465	456	454	454
5	8063	4784	2492	1289	743	552	506	496	494	494
6	8360	5211	2776	1441	820	598	546	536	535	535
7	8583	5588	3045	1588	896	643	585	575	574	574
8	8755	5923	3301	1733	971	686	621	611	610	611
9	8891	6220	3544	1876	1045	729	656	645	645	647
10	9001	6485	3775	2016	1117	771	689	678	679	681

Table 17: NMAE * 10^4 for the Exponential Smoothing method applied on the FedCSIS trace given different lag values l and horizon values h. Rows correspond to the horizon values and columns to the lag values.

Here, we make different observations than in the previous sections, since the performance of exponential smoothing will greatly depend on l. Particularly, we notice for all traces that the error is significantly higher for low lag values in which case the error can even reach 90% making exponential smoothing completely unsuitable in those instances. However, this observation is not surprising given the above formula. Fortunately, the error decreases with increasing lag values and for $l = 10$ exponential smoothing even yields similar performance as autoregression. It is worth noting that the error does not decrease linearly here but rather exponentially while increasing the lag. Hence, ES(10) is a satisfactory method for all traces.

Discussion

In this section, the key points of this report will be discussed. Recall that the overall objective was to apply machine learning models for the purpose of forecasting service metrics on traces collected from a KTH testbed and the FedCSIS 2020 challenge dataset. For task I and task II we modified the traces from the KTH testbeds to the form

$$\mathcal{T}_{new} = \{([x^{(l-1)}, \ldots, x^{(t)}], [y^{(t)}, \ldots, y^{(t+h)}])\}.$$

This allowed us to predict service metrics given past measurements. In task I we fitted linear regression models on these new traces and in task II Recurrent Neural Networks were trained. For both of these methods, forecasting was expressed as a regression problem given a design matrix and target values. Linear regression performed slightly better for the KV service whereas RNNs performed very well on the VoD trace as compared to the other models. Overall both models showed promising results.
In task III we analysed our time series of service metrics by studying their autocorrelation function and applying the Dicky-Fuller test to evaluate whether we have stationarity in our time series. This enabled us to foresee which models in task IV would be most promising. Finally, in task IV we fitted different time series models such as the autoregression model, moving average model, autoregressive integrated moving average model and the exponential smoothing method. The results varied greatly depending on the model parameters. However, it can be said that no model was able to outshine the performance of RNNs in the case of the VoD service. Moreover, out of all models the ARIMA model performed best on the FedCSIS dataset.

References

[1] Rolf Stadler, Rafael Pasquini, and Viktoria Fodor. Learning from network device statistics. *Journal of Network and Systems Management*, 25:1–27, 10 2017.

[2] Peter J. Brockwell and Richard A. Davis. *Introduction to Time Series and Forecasting*. Springer Texts in Statistics. Springer International Publishing, 2016.