Evaluation of Forecasting Methods
Predicting the Amount of People in a Room

Corbinian Stiglmair & Matthias Hermann

February 3, 2021

Contents

1 Training Data 1

2 Markov Chains 1

3 Linear Regression 2

4 SARIMAX 3

5 LSTM Neural Network 3

6 Conclusion 4

1 Training Data

Our training data includes two columns (*timestamp*, *count*) and 1241 observations. Figure (1) shows all values, while figure (2) shows only the first 100 results for visual clarity. We

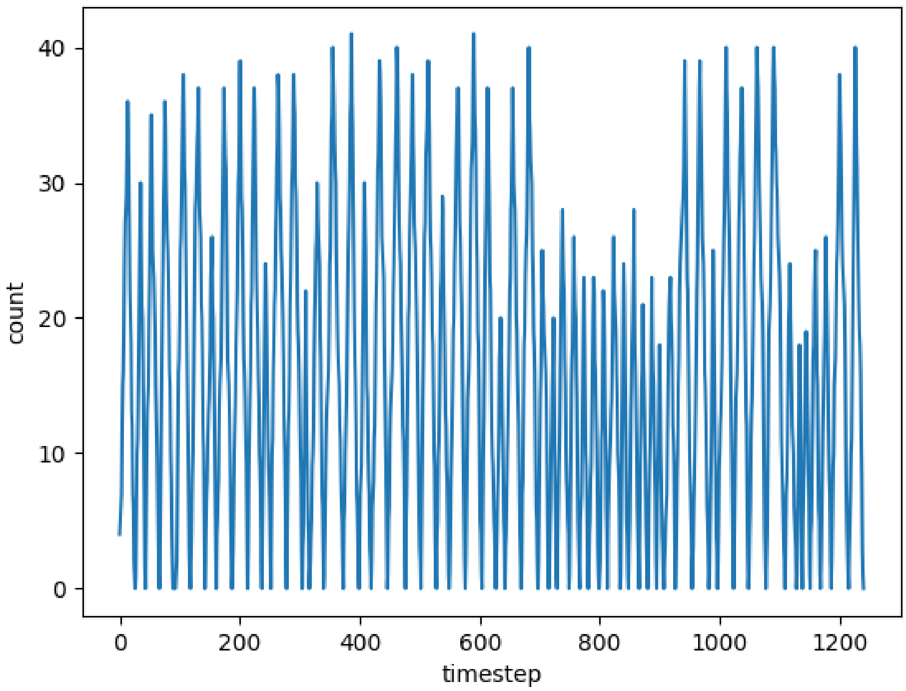

Figure 1: Trainingdata used for testing accuracy of our models

can see that the data is highly seasonal as expected with a frequency of approximately two hours. This can be easily explained by the class duration of two hours. For training or models we used the first 1000 results, while we used the next 100 results toi evaluate our models.

2 Markov Chains

Markov Chains model transitions from state to state via simple probabilities. These probabilities are derived fras follows:

$$p_{i,j} = \frac{t_{i,j}}{\sum_0^n t_{i,k}}$$

$p_{i,j}$ is then the transfer probability from state i to state j, denoted in the $n \times n$ transition matrix. $t_{i,j}$ is the total amount of observed state changes from i to j. One big drawback of markov chains is that they do not maintain knowledge of previously acessed states

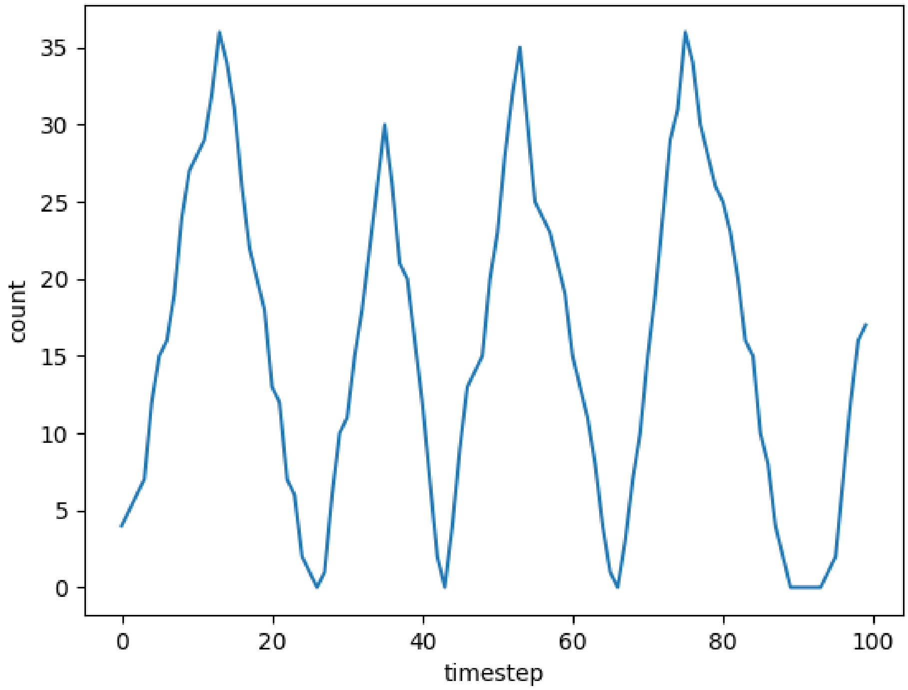

Figure 2: First 100 results of the trainingdata

(except for the current one). Additionaly, predictions by markov chains are inherently non-deterministic (even more so if they are not stationary or just contain absorbing states, which our model clearly does not).

Figure (3) shows the forecast in comparison to the validation set. This forecast is surprisingly accurate (although given the non deterministic nature there were also some less accurate ones). However, two things are noticable: Firstly, it never comes even close to zero as our validation set does and secondly, unlike the validation set which either rises or falls, the forecast fluctuates a lot. Nonetheless it delivers a good estimation.

3 Linear Regression

We used the method *Ordinary Least Squares (OLS)* in order to perform a linear regression on our dataset. Without going into too much detail OLS tries to minimize the residuals (meaning the difference between predictions and actual values) squares to the calculated interpolation. Since we only use one regressor, it is a straight line for us. It therefore struggles immensely with our highly seasonal data, which can be seen in figure (5) and table 6. Yet, the errors arent terrribly large, as the regression line is at the very average of the given set.

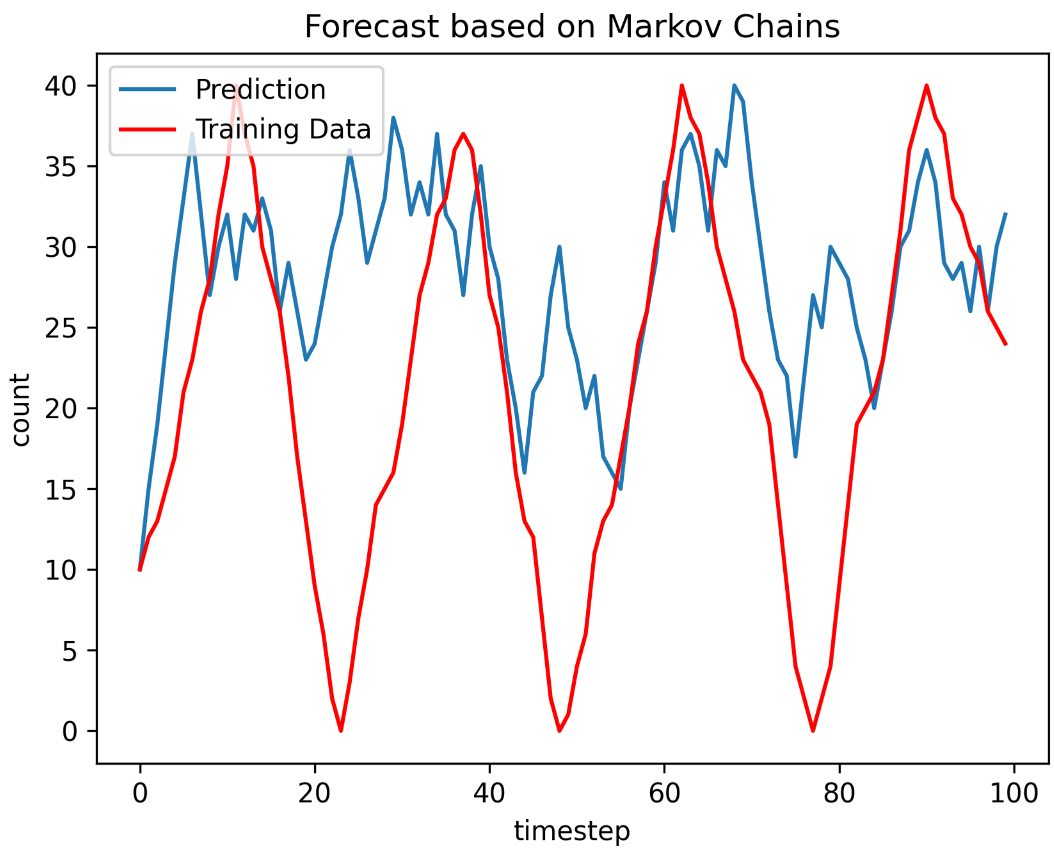

Figure 3: Forecast using Markov Chains

4 SARIMAX

The SARIMAX model tries to take seasonality and trends into account. Unfortunately for us, it didnt work out as well as expected. Our best attempt used the following parameter: order=(0,0,0) and seasonal_order = (1,0,1) and periodicity = 24. Figure (7) and table (8) show the corresponding information.

5 LSTM Neural Network

The *LSTM Neural Network* is a type of Recurrent Neural Network (RNN). These neural networks extend the standard model by adding long term memory cell, which keeps track of prior states. In our context it memorizes prior counts and will output a single value. the parameter *look_back* describes how many prior states are used to derive the next values. Figure (9) and table (13) show the corresponding data. This setup achieved by far the best results during our testing. Figure (10), (11) and (12) show the results for different

Mean Absolute Error	9.03
Root Mean Squared Error	12.285
Mean Absolute Percentage Error	0.323
Symmetric Mean Absolute Percentage Error	47.167
Mean Absolute Scaled Error	2.794

Figure 4: Accuracy Metrics for the Markov Chain Forecast

look_back values.

6 Conclusion

The best choice is clearly the LSTM Neural Network with look_back = 20.
On second place is the sarimax model which captures the seasonality well, but captures its eccentrity poorly.
The third best model is the markov chain, which in our trials approached the data quite well, but added variance, where none was. Additionally, given its volatile nature (due to being a purely probabilistic model) , some runs may perform significantly worse than others.
Lastly, there is also the linear regression using OLS. Since for the given data the algorithm is simply outputting a line parallel to the x axis, the forecast is not very useful given our highly seasonal data. Therefore, we cannot recommend using OLS here.

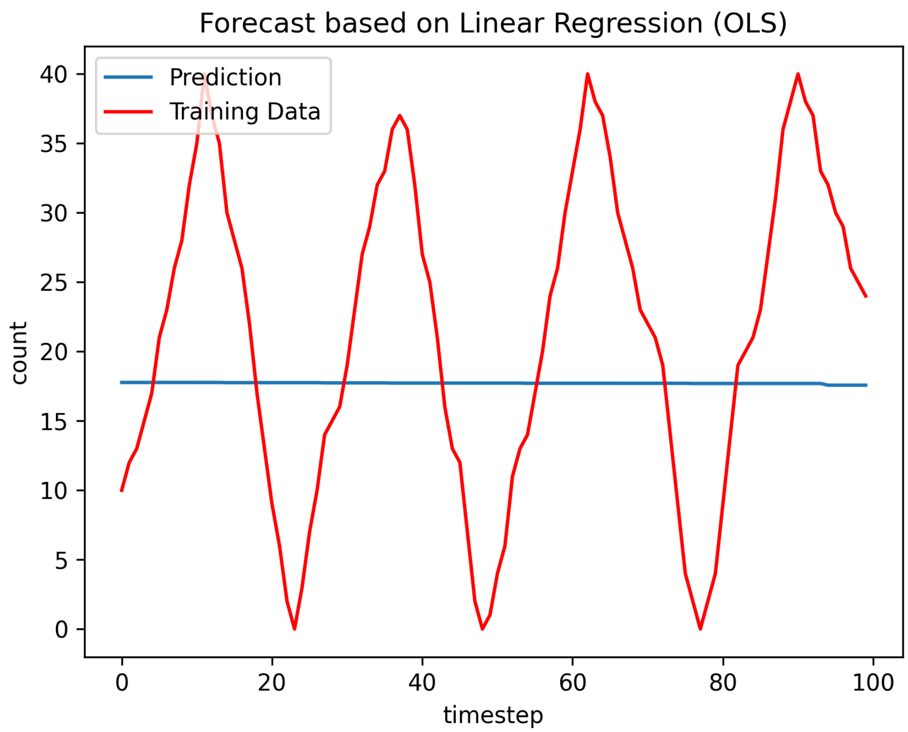

Figure 5: Forecast using Ordinary Least Squares

Mean Absolute Error	10.343
Root Mean Squared Error	12.004
Mean Absolute Percentage Error	1.048
Symmetric Mean Absolute Percentage Error	58.311
Mean Absolute Scaled Error	3.694

Figure 6: Accuracy Metrics for Ordinary Least Squares

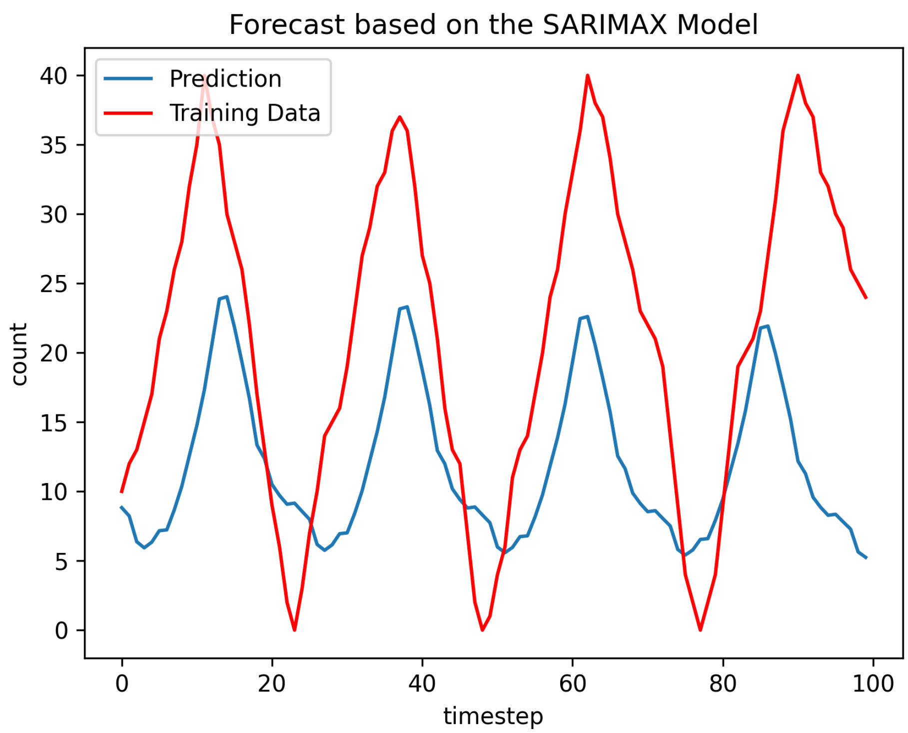

Figure 7: Forecast using SARIMAX order=(0,0,0) and seasonal_order = (1,0,1) and periodicity = 24

Mean Absolute Error	10.886
Root Mean Squared Error	12.941
Mean Absolute Percentage Error	0.637
Symmetric Mean Absolute Percentage Error	69.500
Mean Absolute Scaled Error	3.888

Figure 8: Accuracy Metrics for SARIMAX

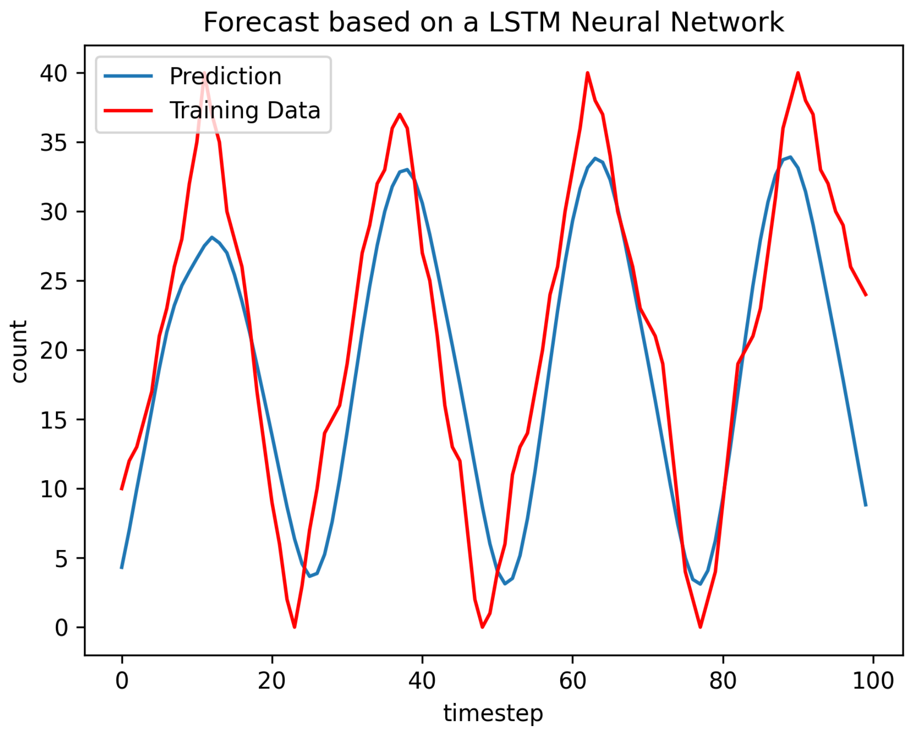

Figure 9: Forecast using a LSTM Neural Network with look_back = 20

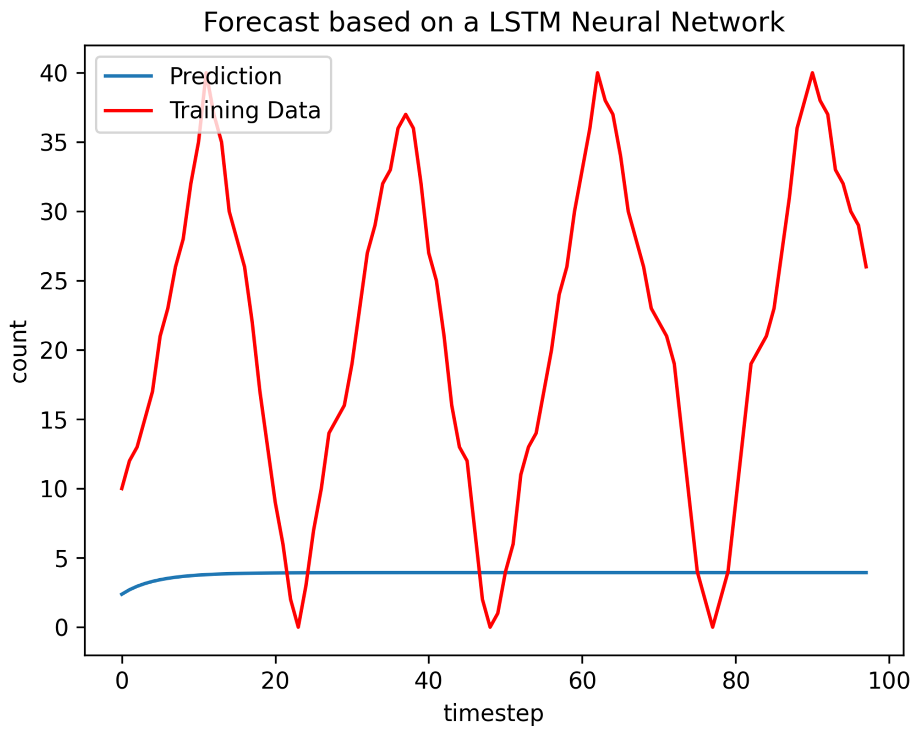

Figure 10: Forecast using a LSTM Neural Network with look_back = 1

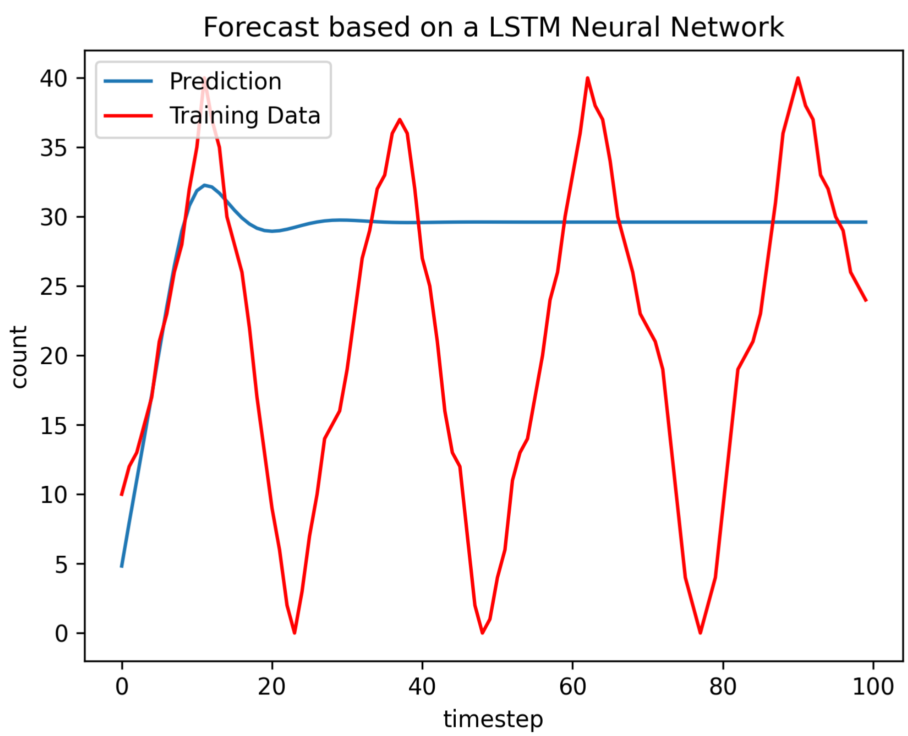

Figure 11: Forecast using a LSTM Neural Network with look_back = 3

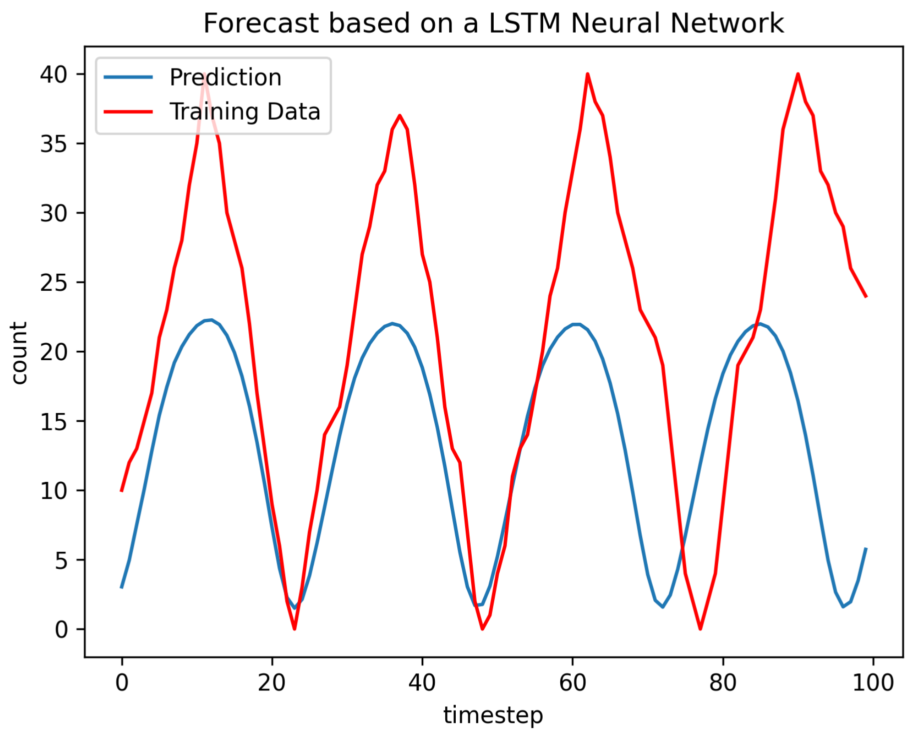

Figure 12: Forecast using a LSTM Neural Network with look_back = 10

Mean Absolute Error	4.653
Root Mean Squared Error	5.536
Mean Absolute Percentage Error	0.390
Symmetric Mean Absolute Percentage Error	36.524
Mean Absolute Scaled Error	1.661

Figure 13: Accuracy Metrics for LSTM Neural Network look_back = 20