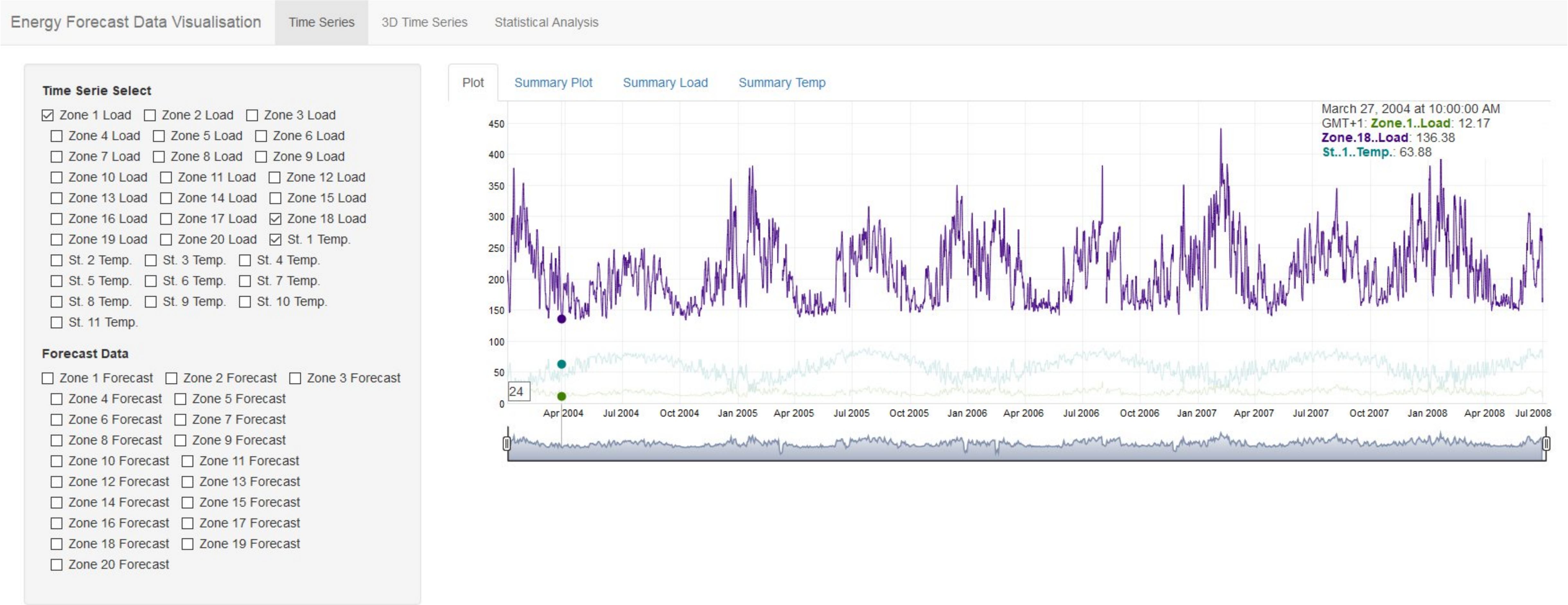
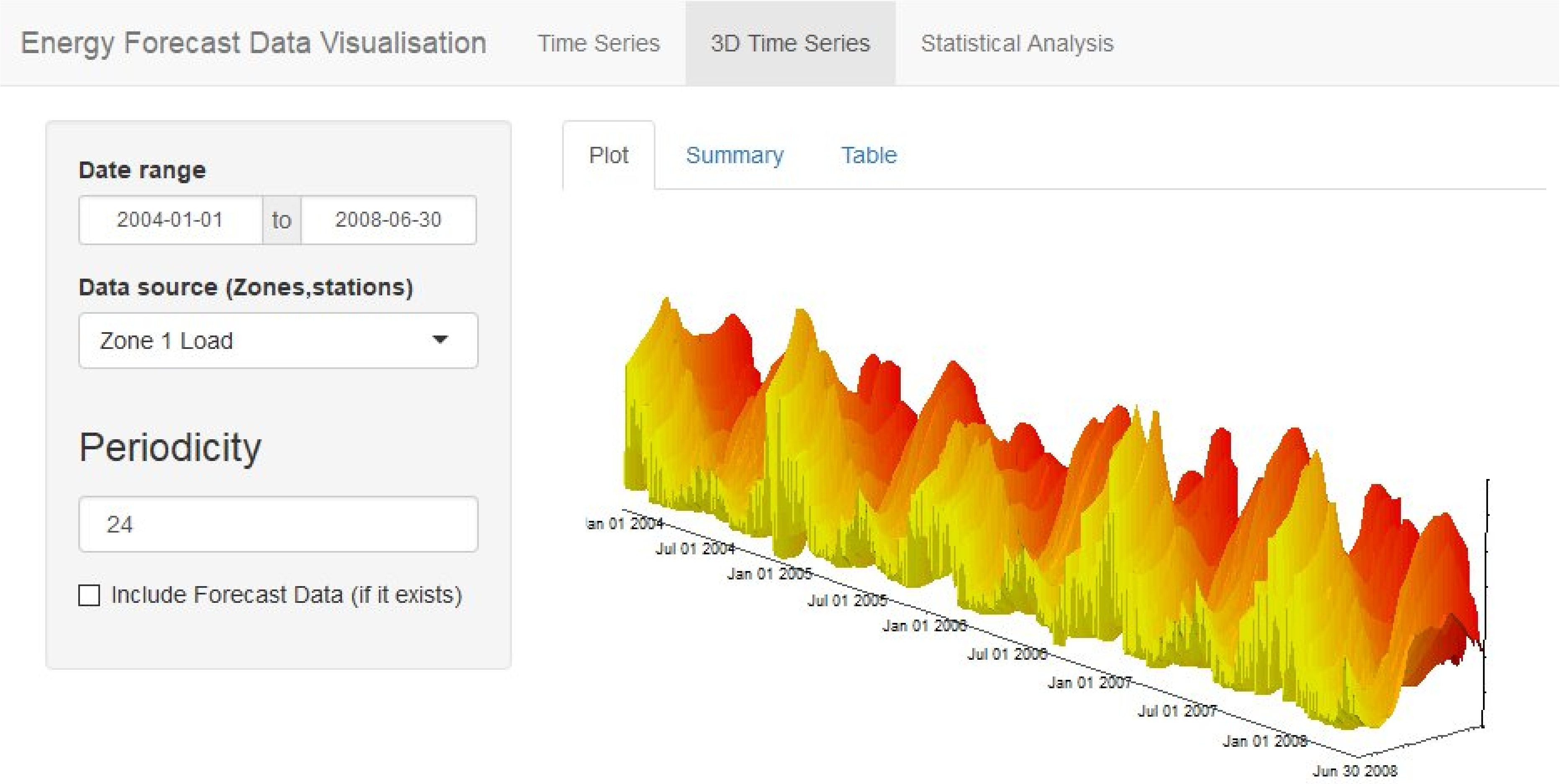
MAP 573 - Group 12

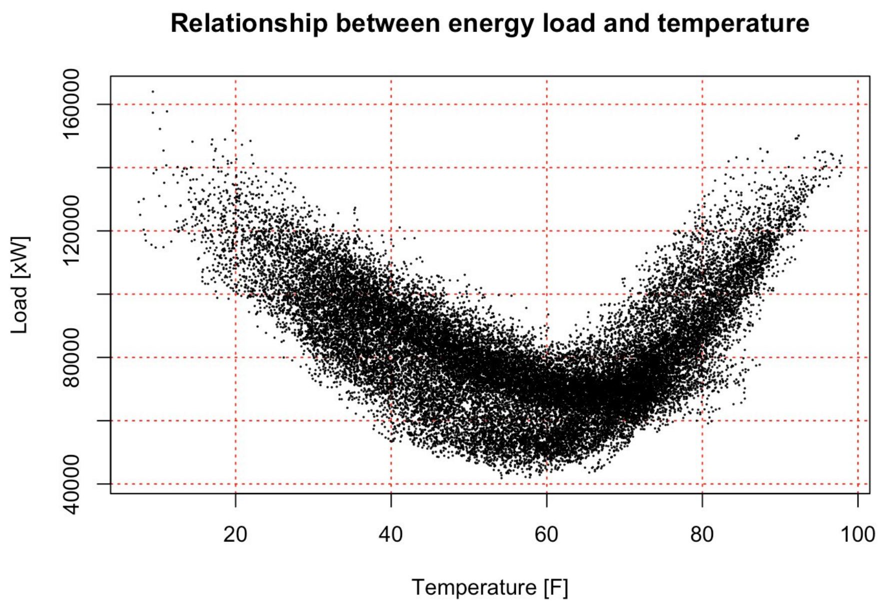
Time Series Forecast using (LSTM) RNN network

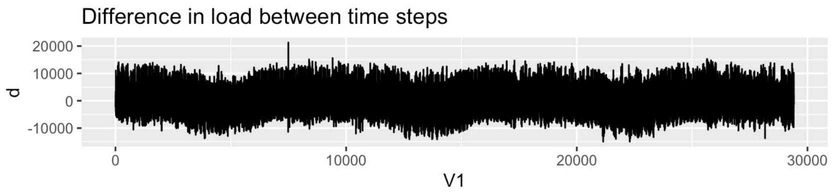
Data Visualization

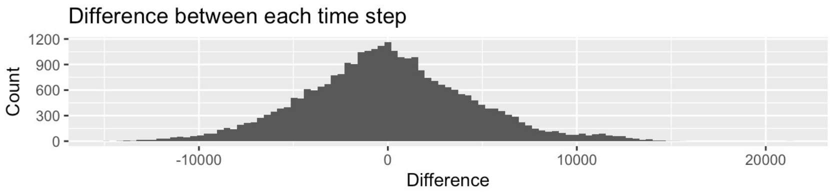


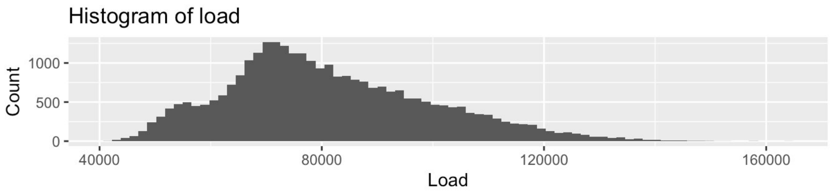
Data Visualization

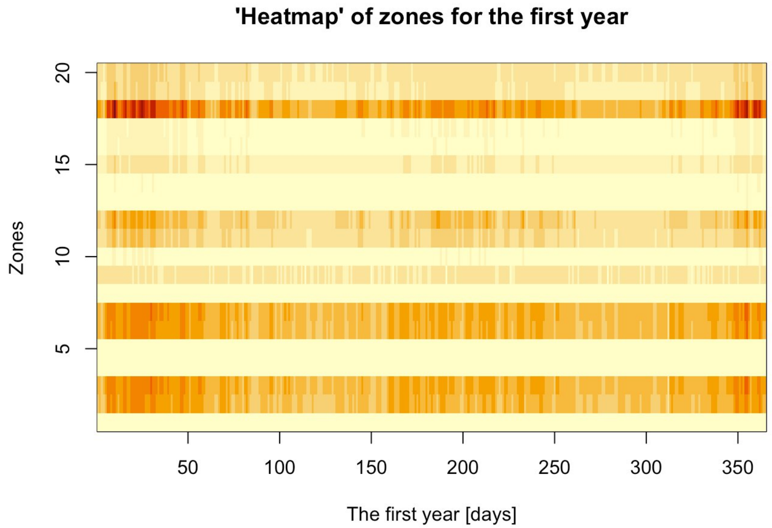


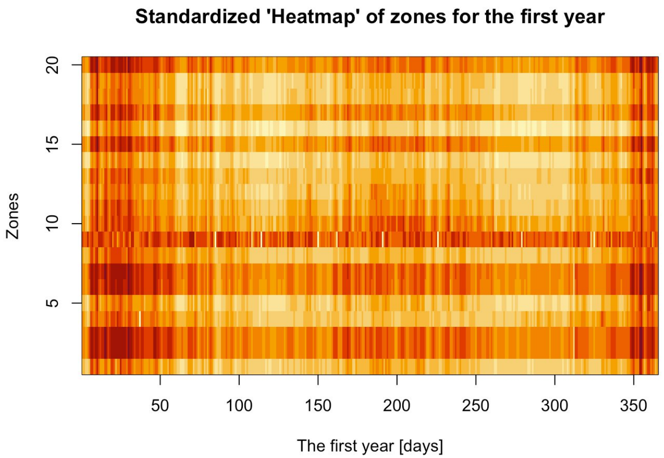

Relationship between energy load and temperature
Load [xW]
Temperature [F]

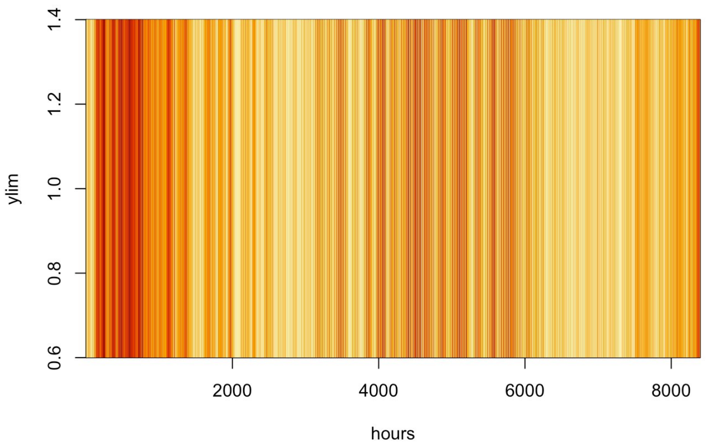

Difference in load between time steps
d
V1

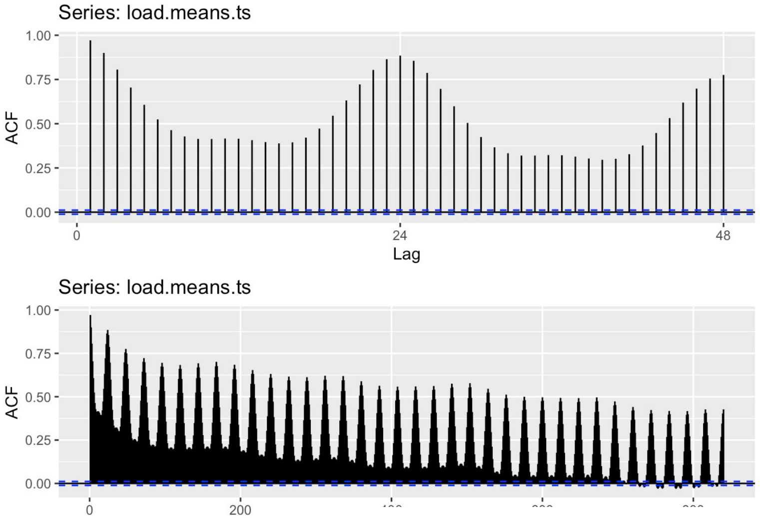

Difference between each time step
Count
Difference

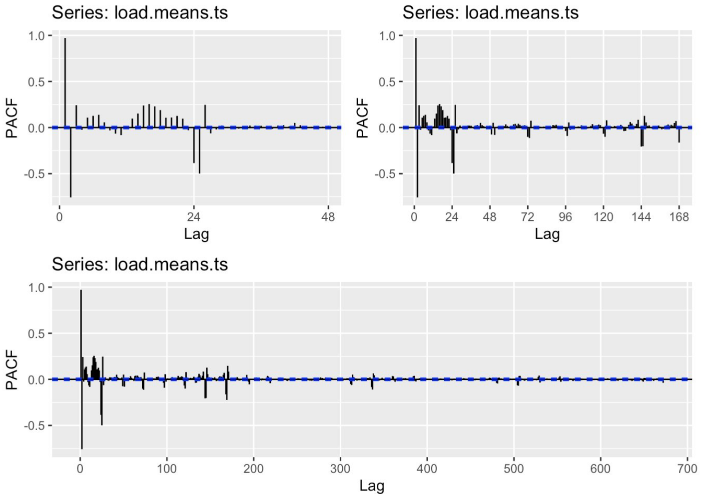

Histogram of load
Count
Load


'Heatmap' of zones for the first year
Zones
The first year [days]

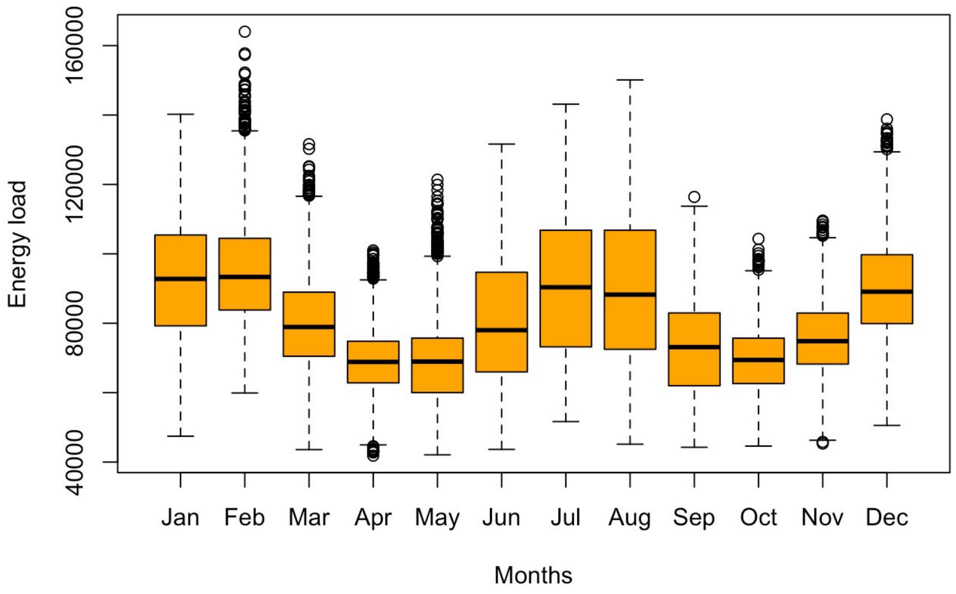

Standardized 'Heatmap' of zones for the first year
Zones
The first year [days]


ylim
hours





Average energy load over the zones vs months

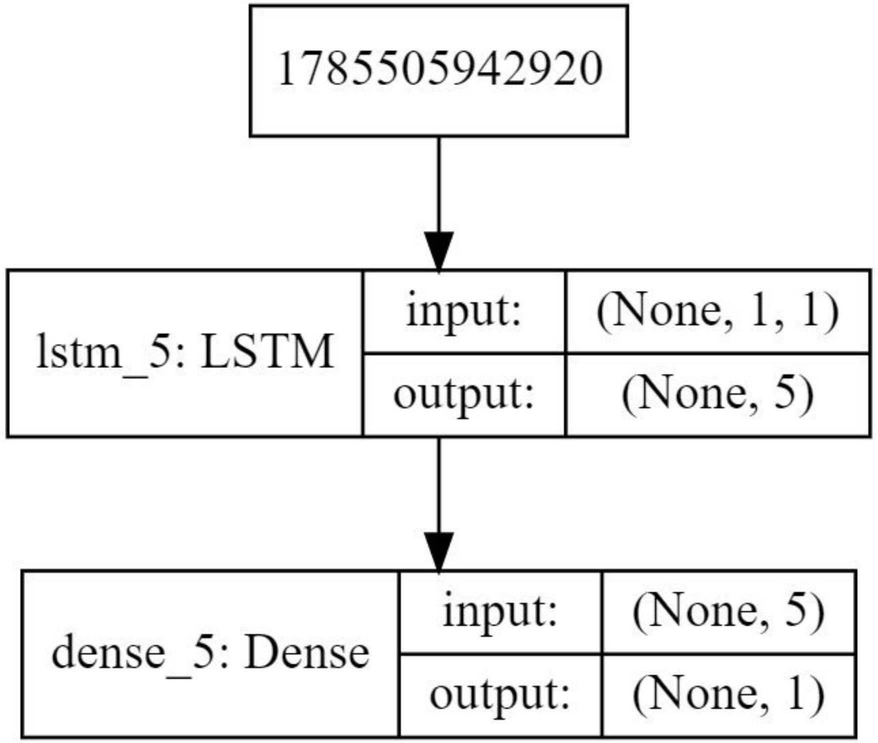


LSTM Network with Python

In [19]:

```python
def get_predict_and_score(model, X, Y):
    # transform the prediction to the original scale.
    pred = normalizer.inverse_transform(model.predict(X))
    # transform also the label to the original scale for interpretability.
    orig_data =normalizer.inverse_transform([Y])
    # calculate RMSE.
    score = math.sqrt(mean_squared_error(orig_data[0], pred[:, 0]))
    return(score, pred)

mse_train, train_predict = get_predict_and_score(vanilla_rnn, train_X, train_Y)
mse_test, test_predict = get_predict_and_score(vanilla_rnn, test_X, test_Y)

print("Training data error: %.2f MSE" % mse_train)
print("Test data error: %.2f MSE" % mse_test)
```

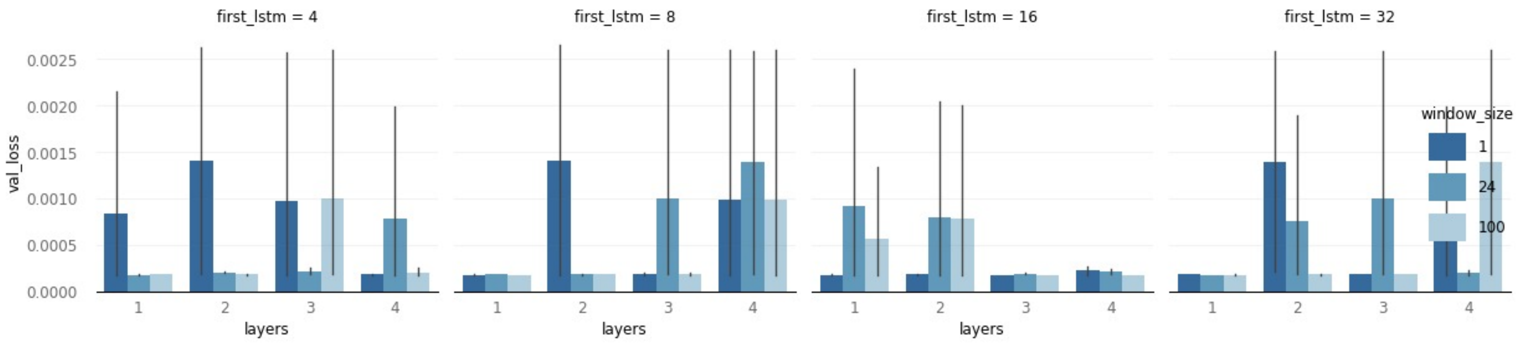
```
Training data error: 1437.70 MSE
Test data error: 1494.71 MSE
```

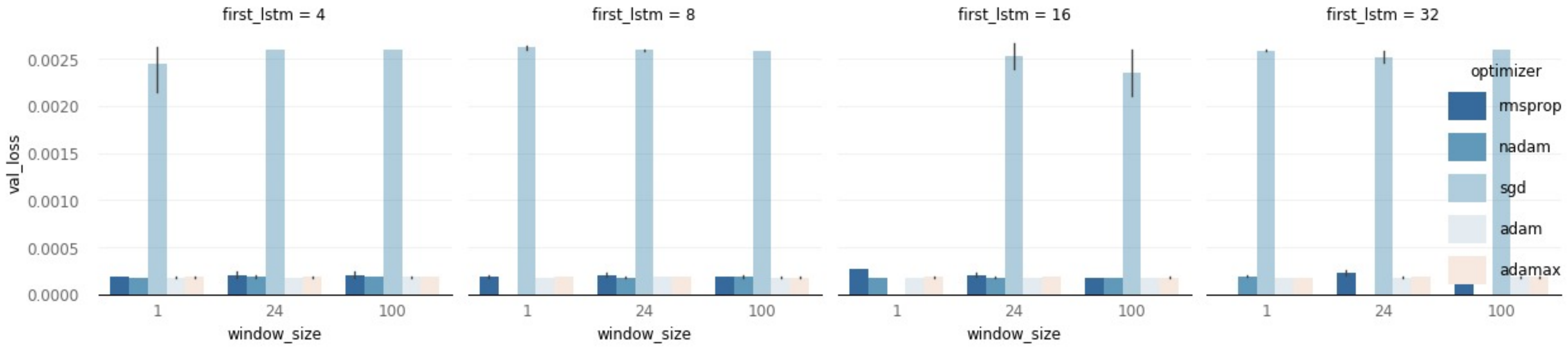


In [20]:

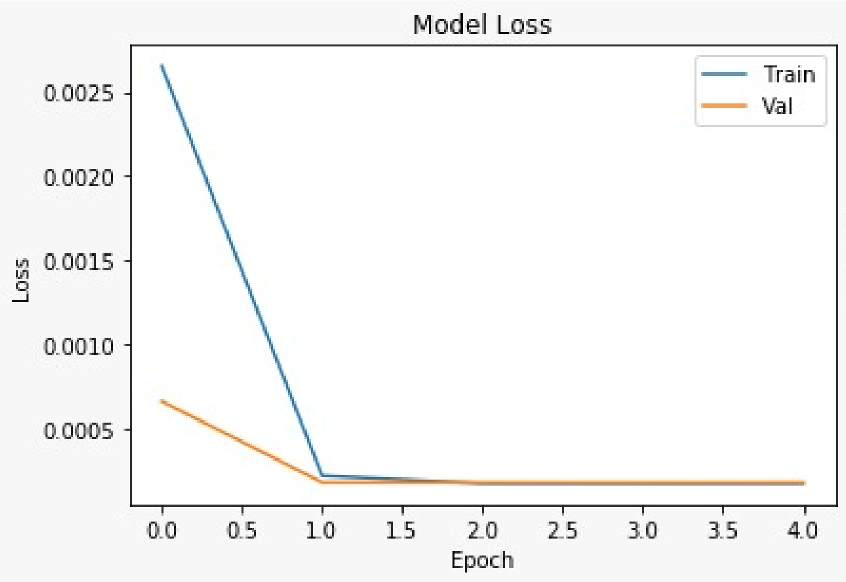
```python
# Training predictions.
train_predictions = np.empty_like(dataset)
train_predictions[:, :] = np.nan
train_predictions[window_size:len(train_predict) + window_size, :] = train_predict

# Test predictions.
test_predictions = np.empty_like(dataset)
test_predictions[:, :] = np.nan
test_predictions[len(train_predict) + (window_size * 2) + 1:len(dataset) - 1, :] = test_predict

# Create the plot.
plt.figure(figsize = (15, 5))
plt.plot(normalizer.inverse_transform(dataset), label = "True Value")
plt.plot(train_predictions, label = "Training Predictions")
plt.plot(test_predictions, label = "Test Predictions")
plt.xlabel("Hours")
plt.ylabel("Load (MWh)")
plt.title("Zone 1 Comparison True vs. Predicted in Training and Testing")
plt.legend()
plt.show()
```

Results

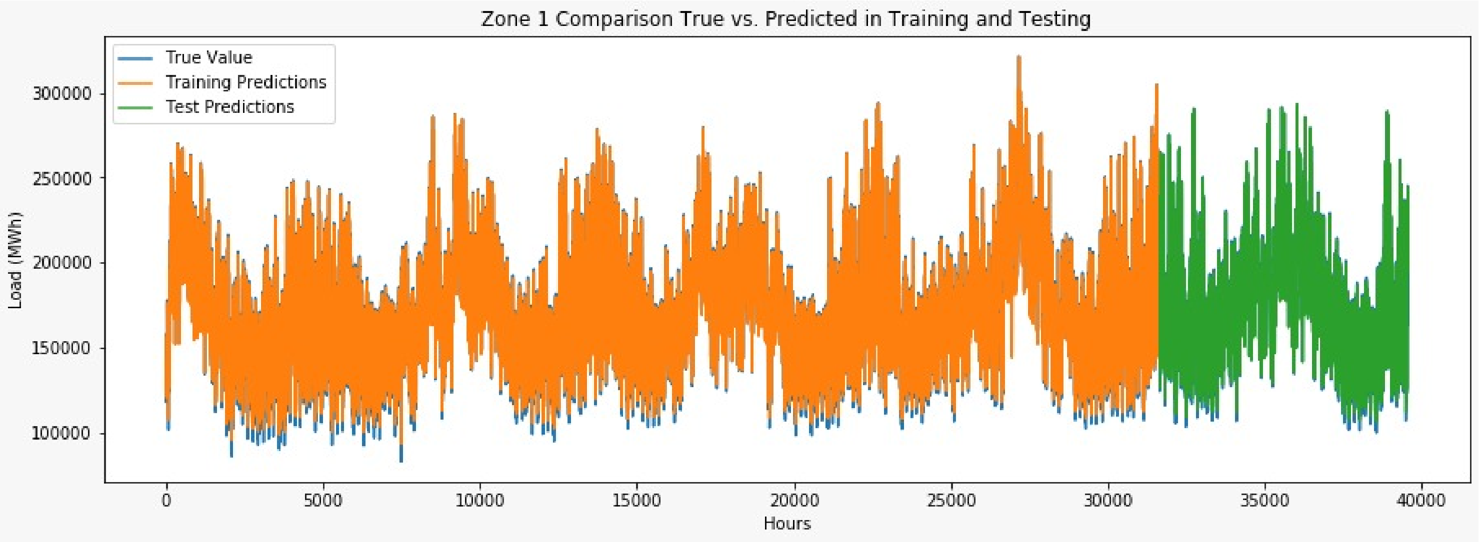

first_lstm = 4
first_lstm = 8
first_lstm = 16
first_lstm = 32
val_loss
0.0025
0.0020
0.0015
0.0010
0.0005
0.0000
1
2
3
4
layers
window_size
1
24
100


first_lstm = 4
first_lstm = 8
first_lstm = 16
first_lstm = 32
val_loss
0.0025
0.0020
0.0015
0.0010
0.0005
0.0000
1
24
100
window_size
optimizer
rmsprop
nadam
sgd
adam
adamax

Results



Model Loss
Train
Val
Loss
0.0025
0.0020
0.0015
0.0010
0.0005
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Epoch



Zone 1 Comparison True vs. Predicted in Training and Testing
True Value
Training Predictions
Test Predictions
Load (MWh)
300000
250000
200000
150000
100000
0
5000
10000
15000
20000
25000
30000
35000
40000
Hours