ECE 528 Homework Assignment 5

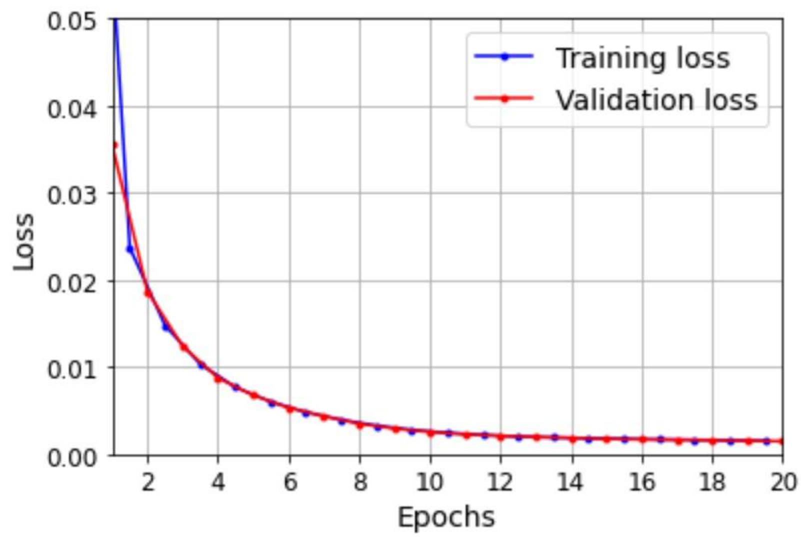
Time Series and Anomaly Prediction

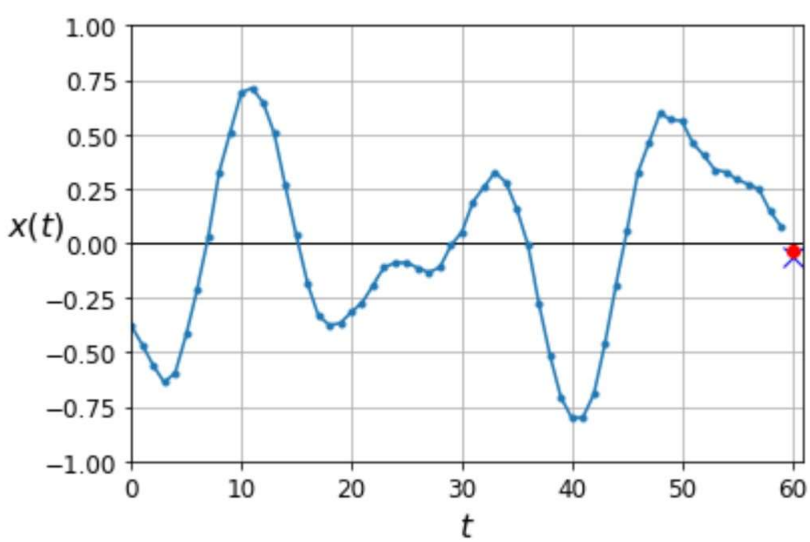
Q.1.(a) On the test set, the **linear regression model** achieves a mean square error (MSE) of **0.00142** whereas the **RNN model** that has been built in the same notebook code **univariate-time-step-series-1-step.ipynb** achieves a mean square error (MSE) of **0.0011576**.

i) Linear Regression Model:

```
model.evaluate(X_test, y_test)
```

```
32/32 [==============================] - 0s 2ms/step - loss: 0.0014

0.0014248630031943321
```

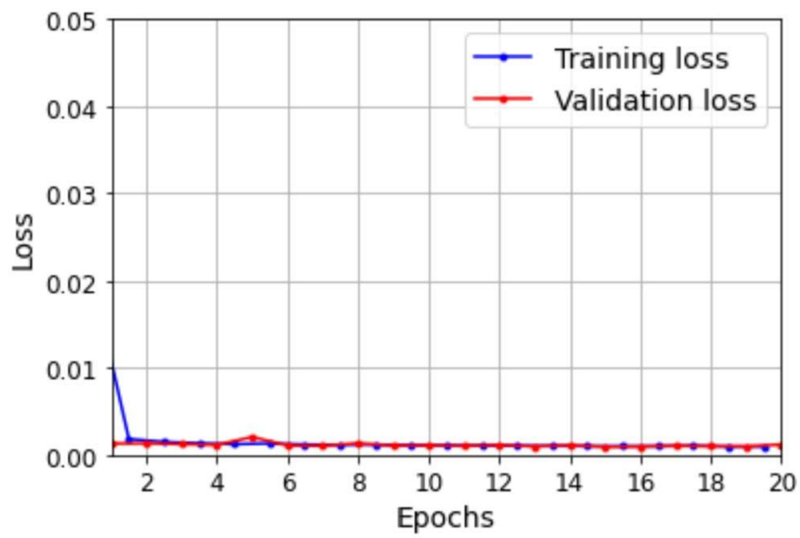


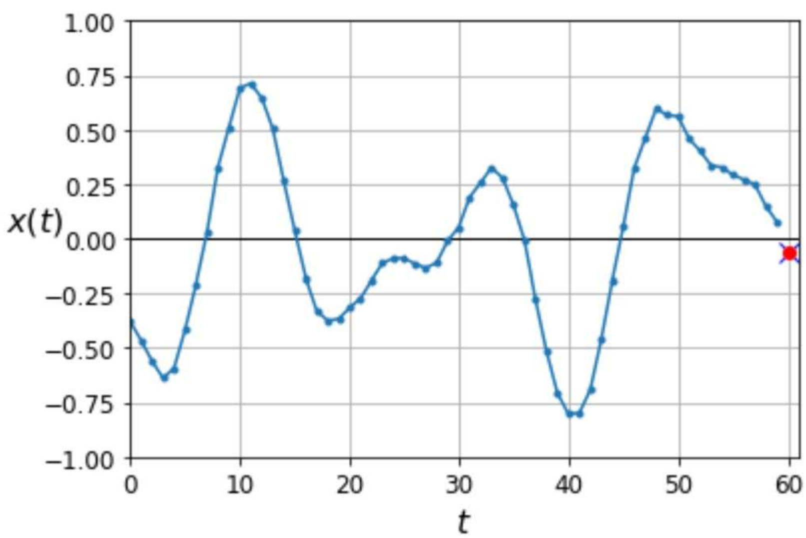


ii) Recurrent Neural Network Model:

```
model_RNN.evaluate(X_test, y_test)
```

```
32/32 [==============================] - 0s 10ms/step - loss: 0.0012

0.0011576558463275433
```



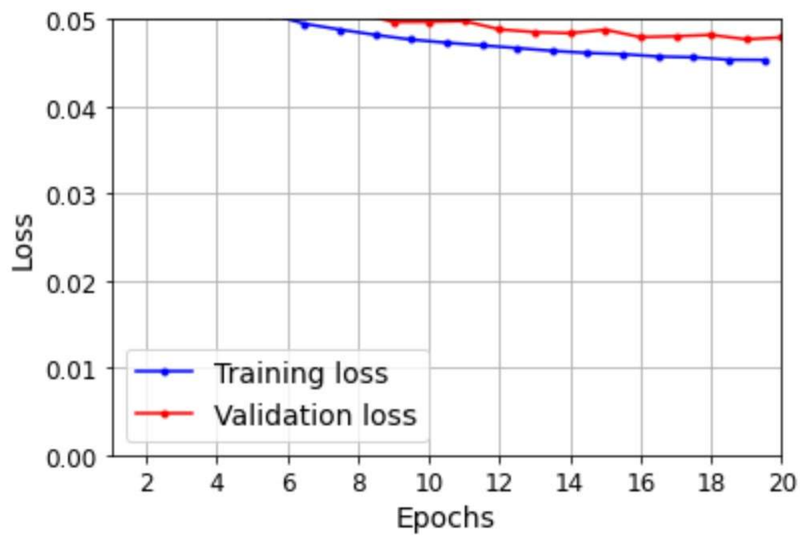


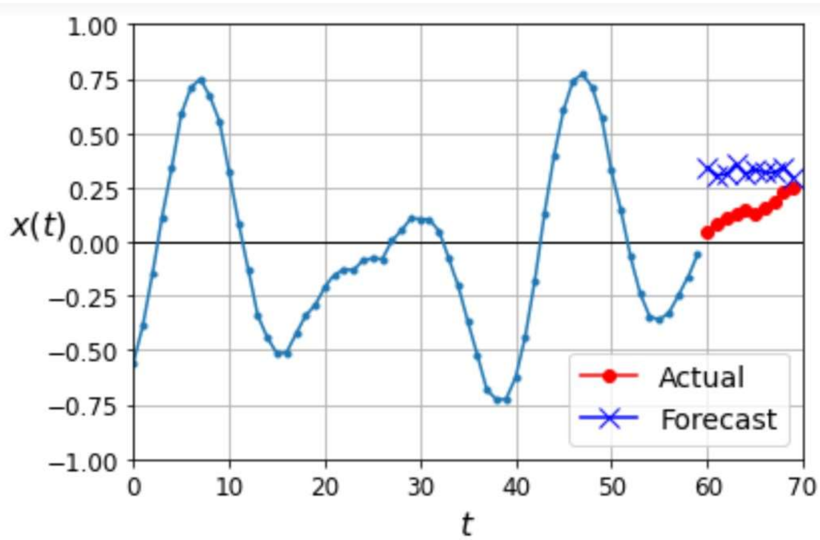
Q.1.(b) On the test set, the **linear regression model** achieves a mean square error (MSE) of **0.0454** whereas the **RNN model** that has been built in the same notebook code **univariate-time-series-multi-step.ipynb** achieves a mean square error (MSE) of **0.0176**.

i) Linear Regression Model:

```
model.evaluate(X_test, y_test)
```

```
32/32 [==============================] - 0s 1ms/step - loss: 0.0455
0.045478884130716324
```

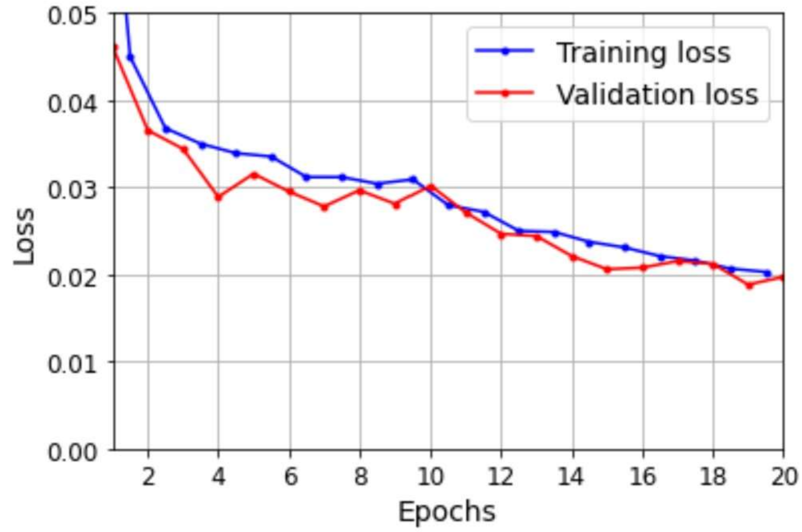


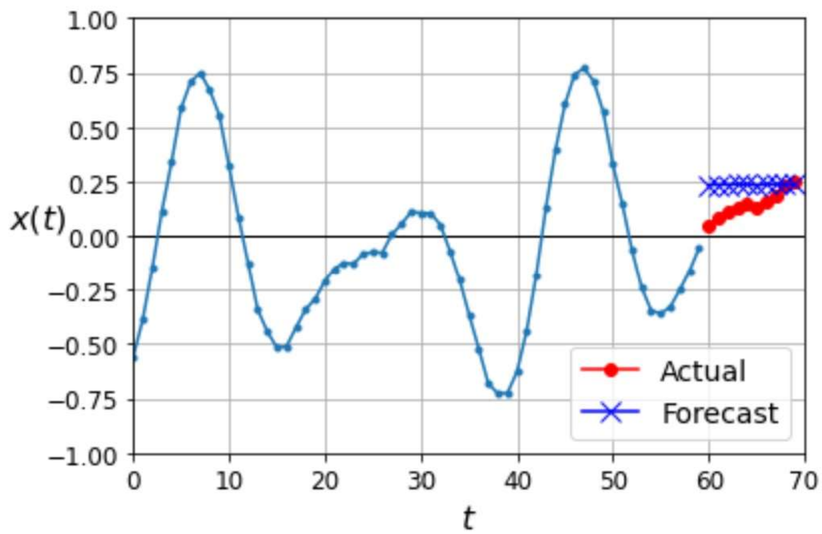


ii) Recurrent Neural Network Model:

```
model_RNN.evaluate(X_test, y_test)
```

```
32/32 [==============================] - 0s 10ms/step - loss: 0.0176
0.01755773462355137
```



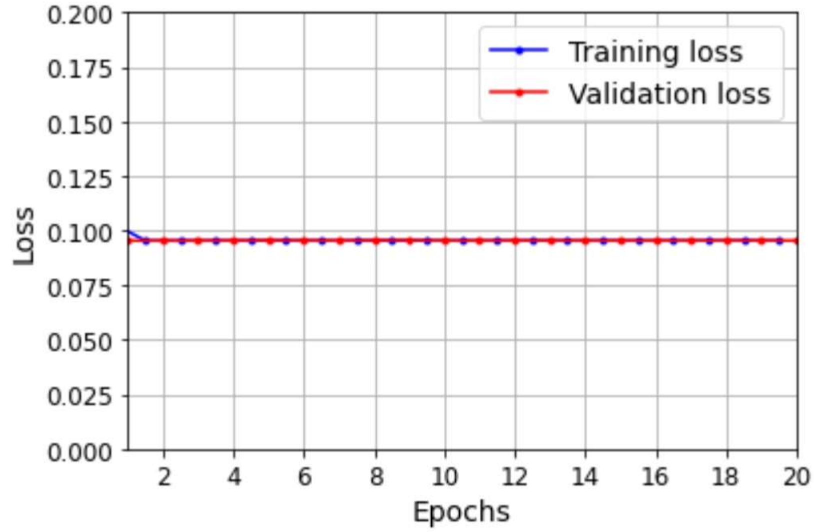


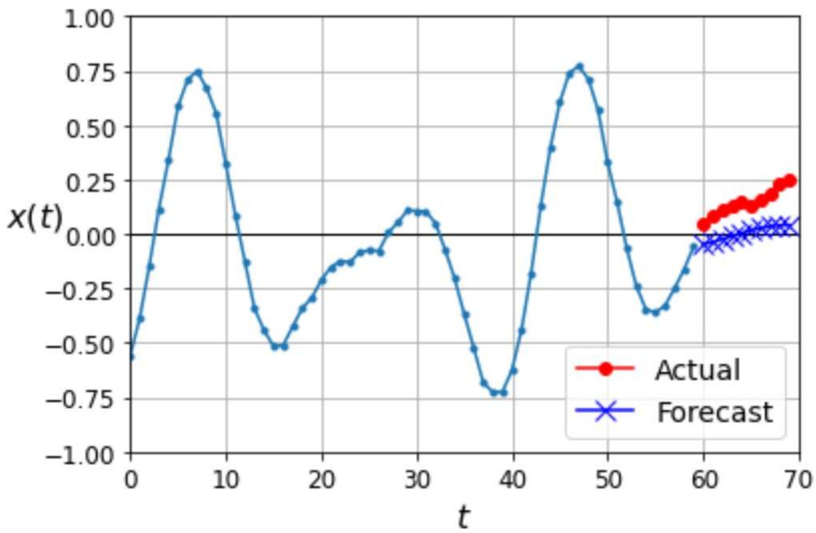
Q.1.(c) On the test set, the **linear regression model** achieves a mean square error (MSE) of **0.096** whereas the **RNN model** that has been built in the same notebook code **univariate-time-series-multi-step-enhanced.ipynb** achieves a mean square error (MSE) of **0.019**.

i) Linear Regression Model:

```
model.evaluate(X_test, Y_test)
#outputs [loss (mse), last_time_step_mse]
```

```
32/32 [==============================] - 0s 2ms/step - loss: 0.0960 - last_time_step_mse: 0.0996

[0.09603306651115417, 0.09957661479711533]
```

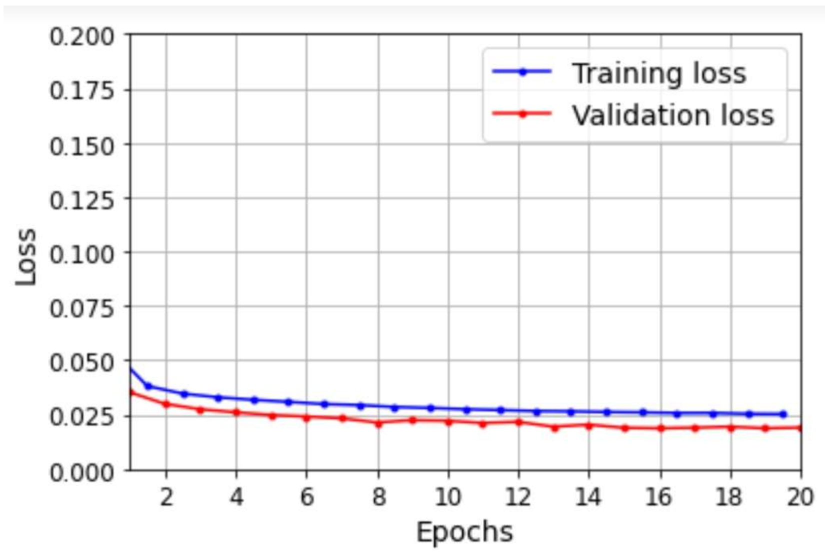


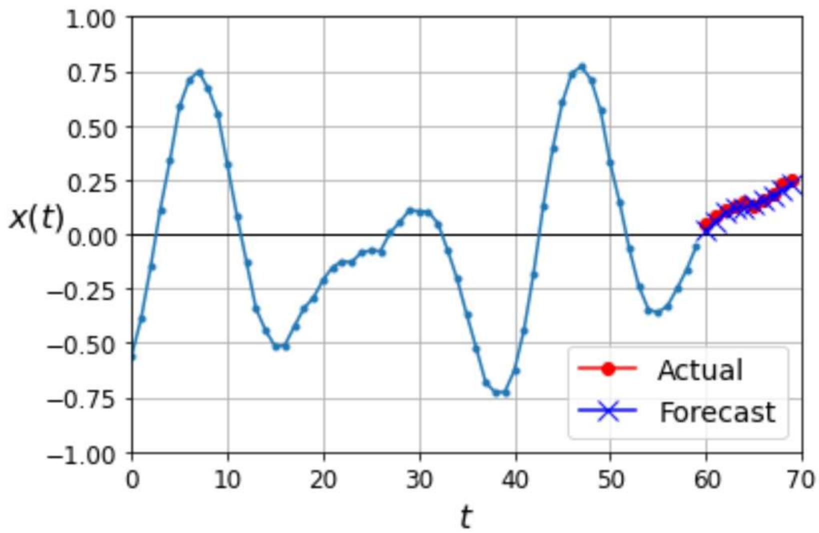


ii) Recurrent Neural Network:

```
model_RNN.evaluate(X_test, Y_test)
#outputs [loss (mse), last_time_step_mse]
```

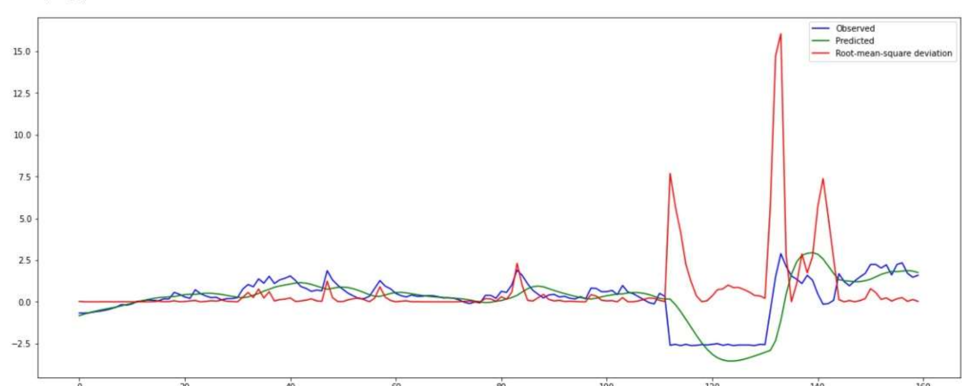
```
32/32 [==============================] - 0s 9ms/step - loss: 0.0192 - last_time_step_mse: 0.0087

[0.01915060728788376, 0.008739501237869263]
```





Q.2. The RMSE plot generated from the notebook code **time-series.ipynb** and the number of times at which the built RNN model detects the anomalies are presented as follows:

```
Reshaping predicted
```




```
Training duration:18.3094162940979
```

```
#anomaly threshold of 5 chosen based on empirical analyses
start_time = 0
threshold = 5
for i in range(0, len(predicted)-1):
    if ((y_test[i] - predicted[i]) ** 2) > threshold:
        print("Anomaly at time", start_time+i, "RMSE value:", (y_test[i] - predicted[i]) ** 2)
    i = i+1
```

```
Anomaly at time 112 RMSE value: 7.67496283629409
Anomaly at time 113 RMSE value: 5.685969144976077
Anomaly at time 131 RMSE value: 5.737776851010587
Anomaly at time 132 RMSE value: 14.748169266711814
Anomaly at time 133 RMSE value: 16.058523278862616
Anomaly at time 140 RMSE value: 5.701007081214678
Anomaly at time 141 RMSE value: 7.369210825594088
Anomaly at time 142 RMSE value: 5.048689965804947
```

Q.3. In the **isolation-forest.ipynb** notebook code, the built model makes use of the **Isolation Forest Classifier** on the **kddcup.data.corrected** dataset, where it achieves an **AUC-ROC value** of **98.84%** on the **validation set** and **98.43%** on the **test set**.

i) Validation Set:

```
from sklearn.metrics import roc_auc_score

anomalies = anomaly_scores > -0.19
matches = y_val == list(encoded.classes_).index("normal.")
auc = roc_auc_score(anomalies, matches)
print("AUC: {:.2%}".format (auc))
```

```
AUC: 98.84%
```

ii) Test Set:

```
anomalies_test = anomaly_scores_test > -0.19
matches = y_test == list(encoded.classes_).index("normal.")
auc = roc_auc_score(anomalies_test, matches)
print("AUC: {:.2%}".format (auc))
```

```
AUC: 98.43%
```