
# National and Kapodistrian University of Athens

Department of Informatics and Telecommunications

## software development for algorithmic problems

## Assignment 3 - A - Forecasting Stock Prices using LSTM NN

---

Project implemented by team № **59**:

CHARALAMPOS MARAZIARIS - 1115201800105
SPYRIDON CHALKIAS - 1115201800209

## Contents

# 1. Fine-tuning

## 1.1 A

### 1.1.1 Batch size

We experimented with batch sizes of 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096 samples. The results showcased next, indicate that a batch size of **64** or **128** is the most preferable, since both Training and Validation loss converge at a very low value (less than 0.01) and the predictions seem to neither underfit nor overfit the validation set.

Figure 1: Batch size = 4, Overfitting

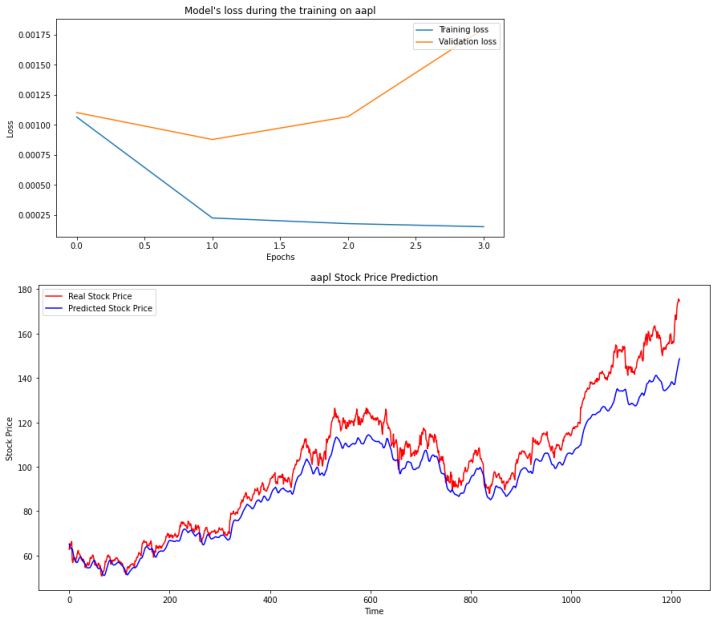



Figure 2: Batch size = 8, Greatly Overfitting

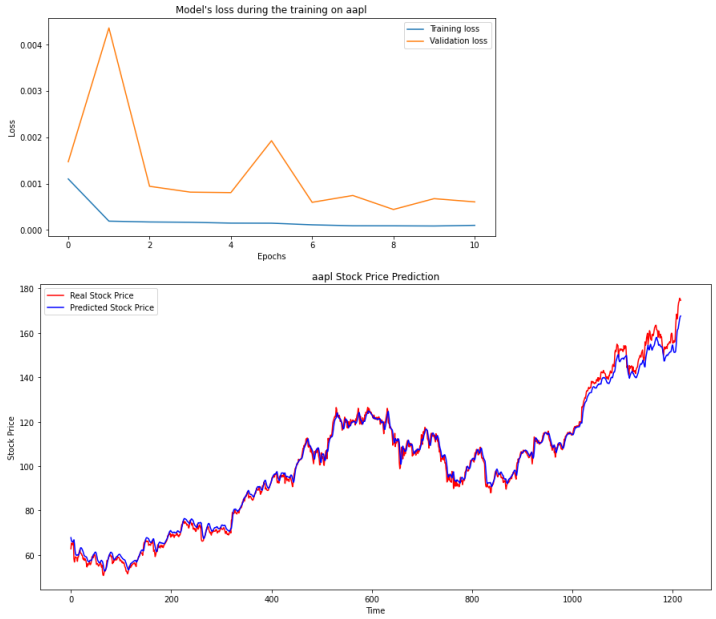


Figure 3: Batch size = 16, Overfitting

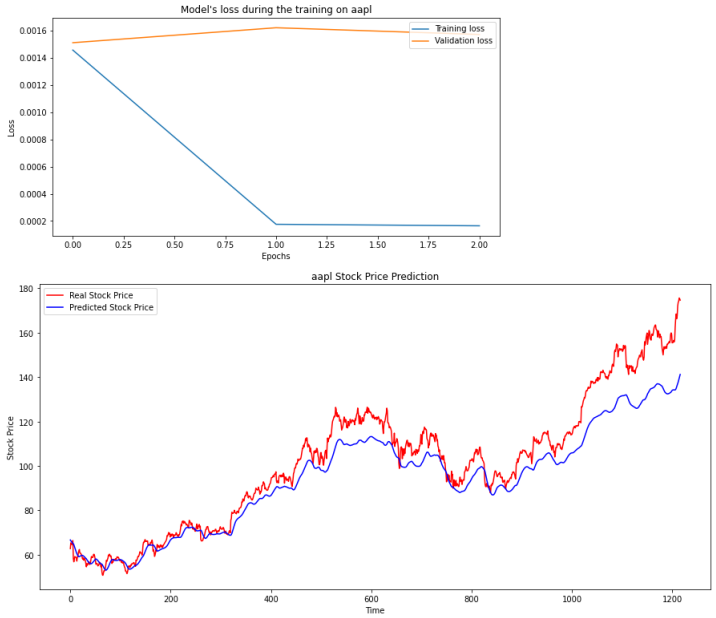
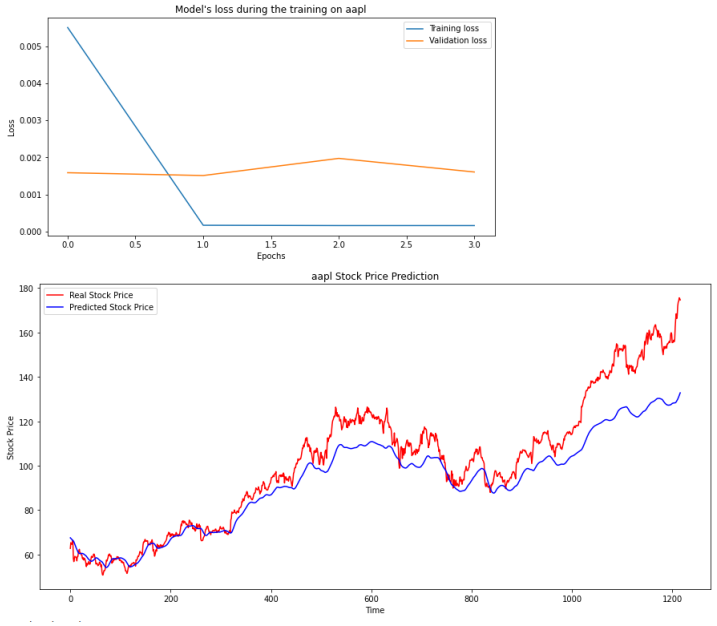



Figure 4: Batch size = 32, Overfitting




### 1.1.2 Dropout probability

We experimented with dropout probabilities of $0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8$ and $0.9$. Dropout layers are used to reduce the chances that our model will overfit the validation dataset. The results below indicate that a dropout probability between **20% and 30%** is more favorable, obtaining very low Train and Validation losses while preventing overfitting, when compared to a model with no dropout layers.

Figure 5: Batch size = 64, Train and Validation loss converging


Figure 6: Batch size = 128, Train and Validation loss converging


### 1.1.3 Layer dimensions

We experimented with **2 to 6** layers, each with **50, 100, 200, 300, 400, 500 or 600** nodes. We present some of the results below. The rest of them can be found in the auxiliary **experiments.ipynb** file provided. We conclude that **2 layers each with 100 nodes** and 5 layers each with 500 nodes give the best predictions while minimizing the Train and Validation losses. We also observe that when we employ a large number of layers (such as 5 and above) we also need to have a large number of nodes in each layer (around 300 and above) otherwise we may underfit the Validation data.

Figure 7: Batch size = 256, Overfitting

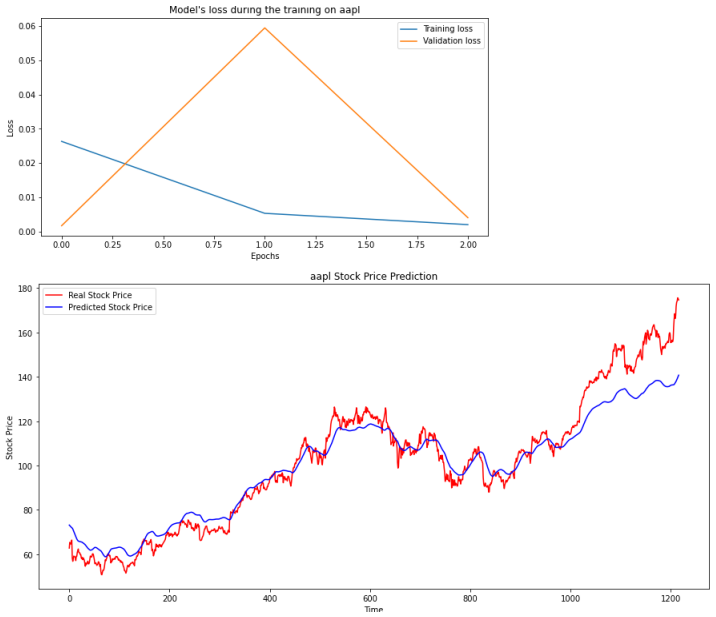



Figure 8: Batch size = 512, Underfitting


### 1.1.4 Time steps

We experimented with time steps: 1,2,3,4,5,10,15,20,...,95,100. Below we present some of the results obtained. We concluded that a time-step of around **40 to 60**, gives a good fit while maintaining very low Training and Validation losses.

### 1.1.5 Lag

We experimented with lag values in the range: 1,2,4,6,8,...,48,50. Selected outcomes are showcased below. We concluded that a lag value of **1** is optimal.

Figure 9: Batch size = 1024, Underfitting

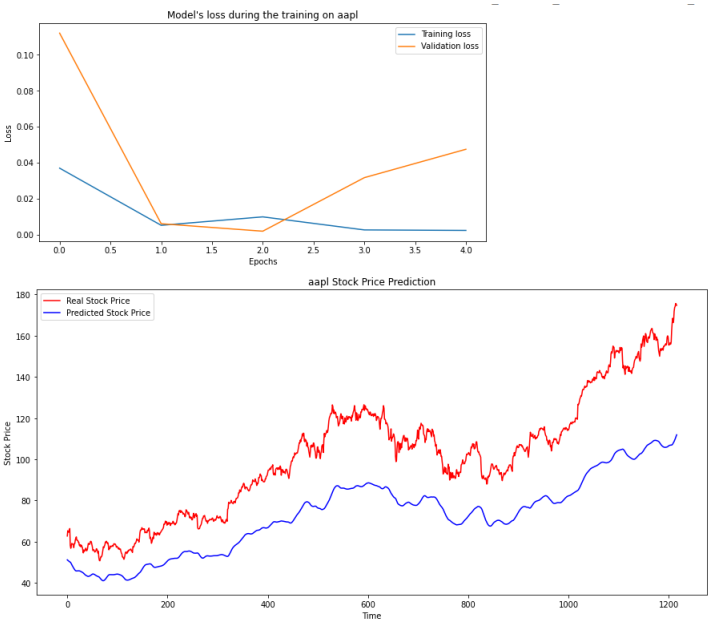



Figure 10: Batch size = 2048, Underfitting

Figure 11: Batch size = 4096, Train and Validation error converge, model slightly underfits


Figure 12: Dropout probability = 0%, No dropout - model overfits

Figure 13: Dropout probability = 10%, Slightly overfitting


Figure 14: Dropout probability = 20%, Decent Train and Validation losses, no overfit

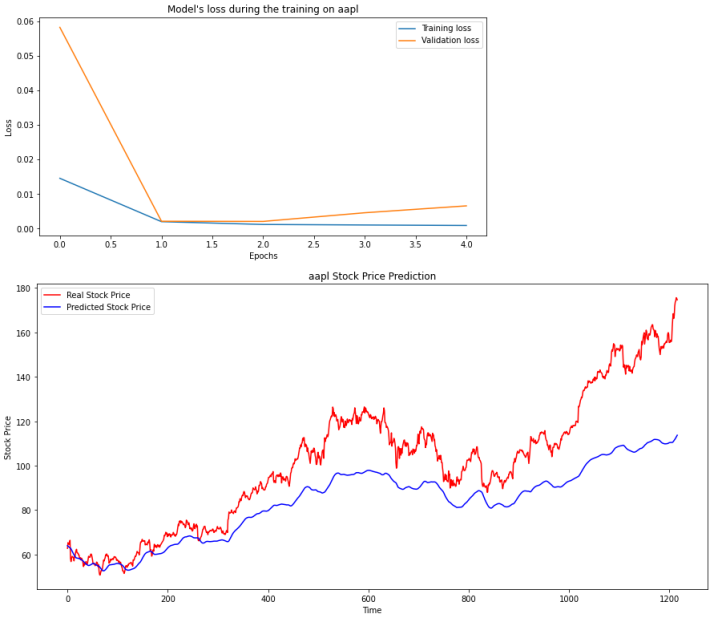


Figure 15: Dropout probability = 30%, Very low Train and Validation losses, no overfit


Figure 16: Dropout probability = 40%, Underfitting

Figure 17: Dropout probability = 50%, Underfitting


Figure 18: Dropout probability = 60%, Underfitting

Figure 19: Dropout probability = 70%, Underfitting

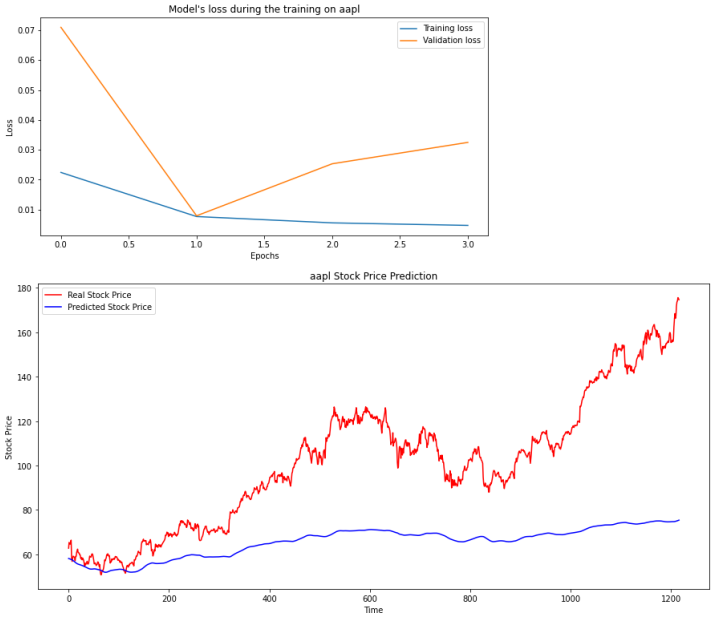



Figure 20: Dropout probability = 80%, Greatly Underfitting

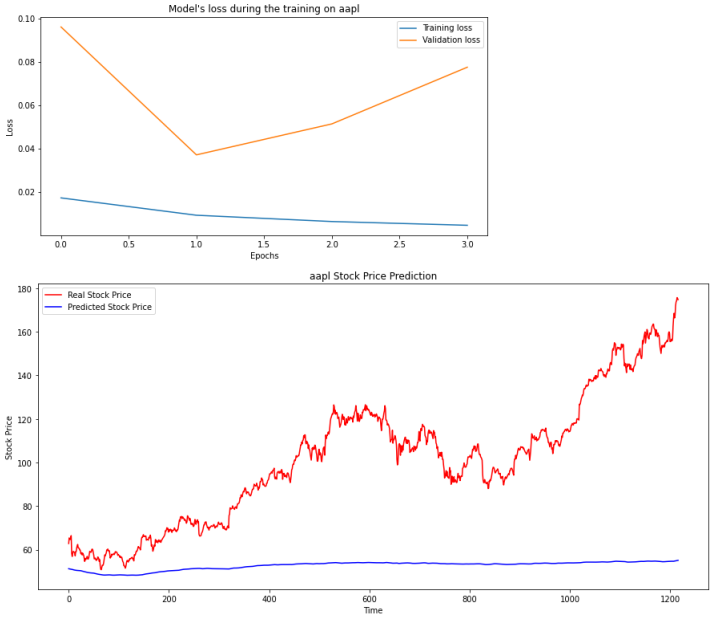


Figure 21: Dropout probability = 90%, Greatly Underfitting

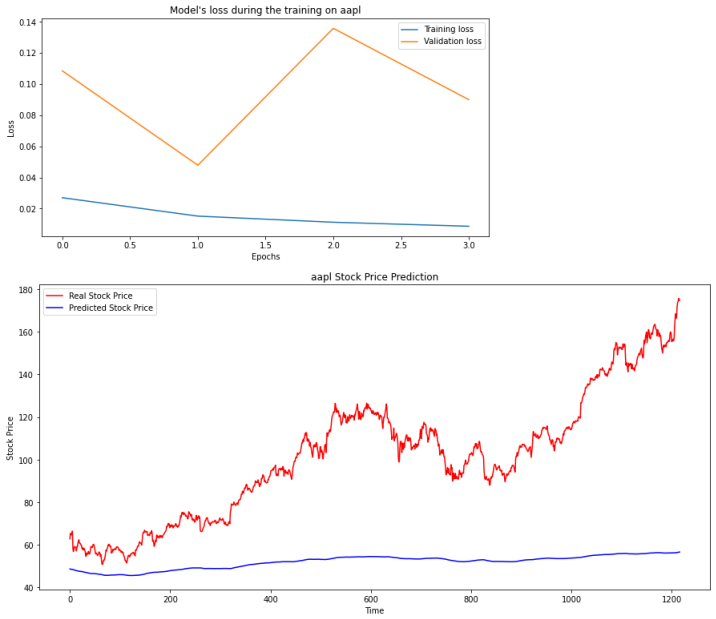



Figure 22: Layers = 2, Nodes in each layer = 100, Low Train and Validation losses

Figure 23: Layers = 3, Nodes in each layer = 100, Slight Underfit


Figure 24: Layers = 3, Nodes in each layer = 400, Train and Validation losses converge, good fit

Figure 25: Layers = 3, Nodes in each layer = 600, Slightly worse fit that with 400 nodes

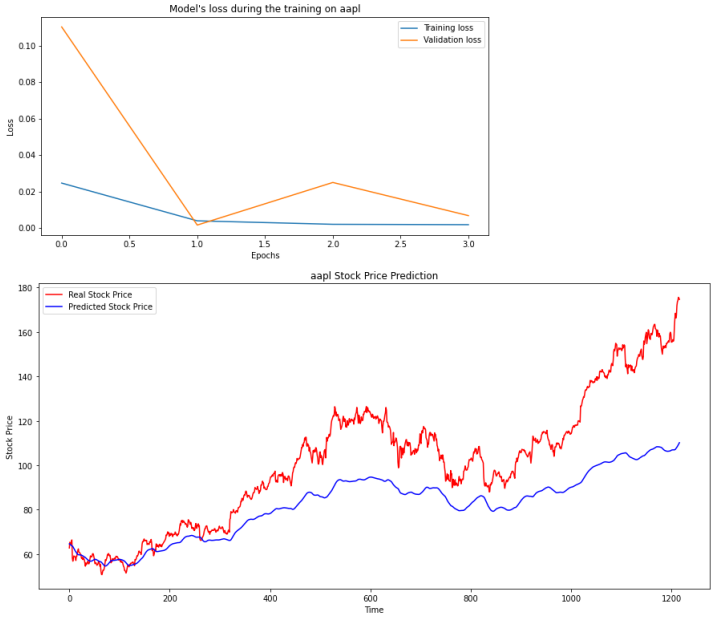



Figure 26: Layers = 4, Nodes in each layer = 100, Underfit

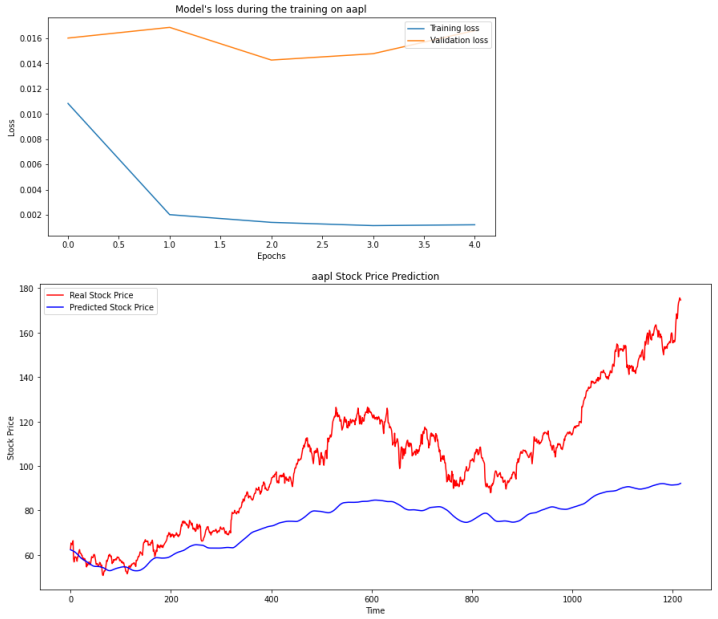


Figure 27: Layers = 4, Nodes in each layer = 400, Slight underfit


Figure 28: Layers = 5, Nodes in each layer = 500, Train and Validation losses converge, very good fit

Figure 29: Layers = 6, Nodes in each layer = 100, Underfit


Figure 30: Layers = 6, Nodes in each layer = 400, Losses converge, good fit

Figure 31: Time Step = 1, Heavy overfit


Figure 32: Time Step = 2, Weird fit, can predict precisely the changes but not absolute values-Overfit

Figure 33: Time Step = 10, Underfit


Figure 34: Time Step = 25, Decent fit, converging losses

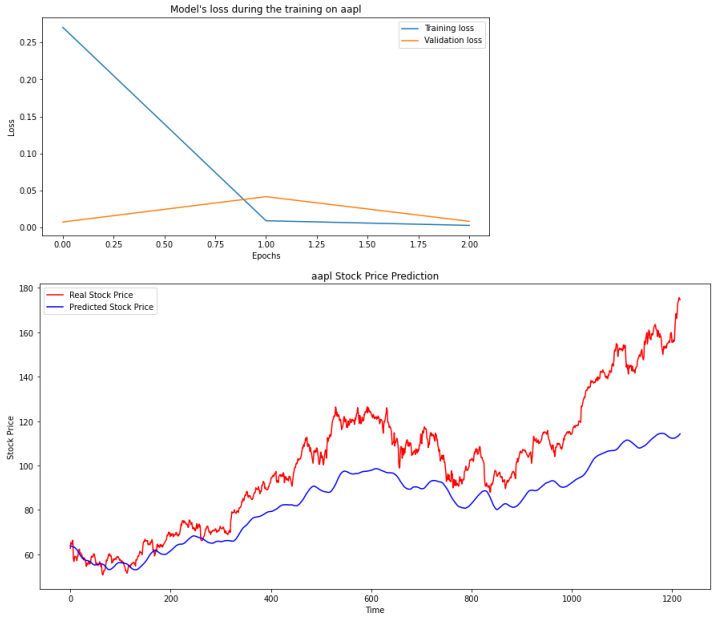


Figure 35: Time Step = 40, Very good fit, very low losses


Figure 36: Time Step = 60, Very good fit, very low losses

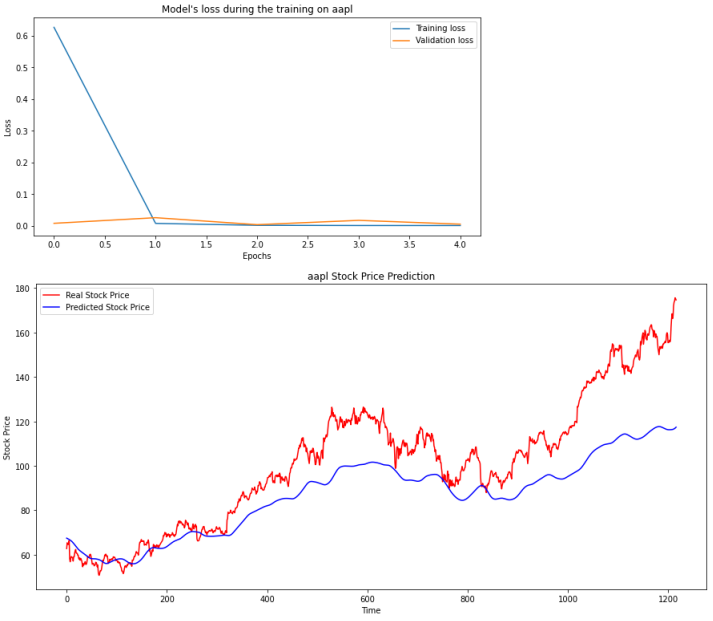


Figure 37: Time Step = 90, Underfit


Figure 38: Lag = 1, Good fit

Figure 39: Lag = 2, Slight underfit


Figure 40: Lag = 16, Underfit

Figure 41: Lag = 36, Underfit


Figure 42: Lag = 46, Underfit