National and Kapodistrian University of Athens

Department of Informatics and Telecommunications

software development for algorithmic problems

Assignment 3 - Γ- CNN Finetuning

Project implemented by team № **59**:

CHARALAMPOS MARAZIARIS - 1115201800105
SPYRIDON CHALKIAS - 1115201800209

Contents

1 Fine-tuning **3**

1.1 C 3

1.1.1 Batch Size 3

1.1.2 Filter layers and Filter sizes 4

1.1.3 Latent dimensions 5

1.1.4 Window size 6

1. Fine-tuning

1.1 C

1.1.1 Batch Size

We experimented with batch sizes of 8, 32, 64, 128, 512, 1024 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 require just a few epochs to train.

Figure 1: Batch size = 8

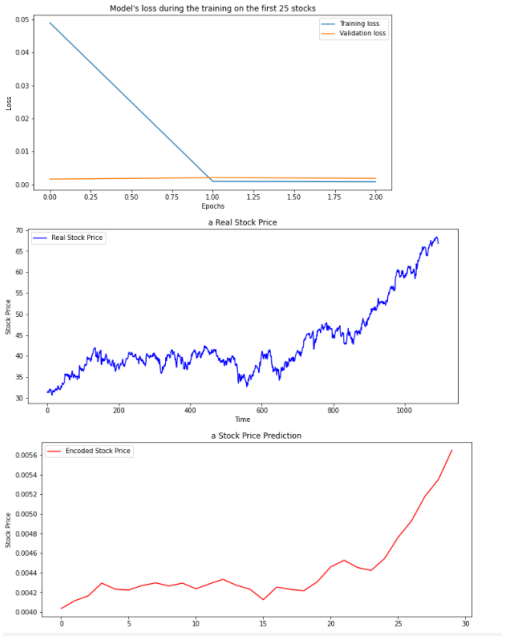


Figure 2: Batch size = 32

Figure 3: Batch size = 64

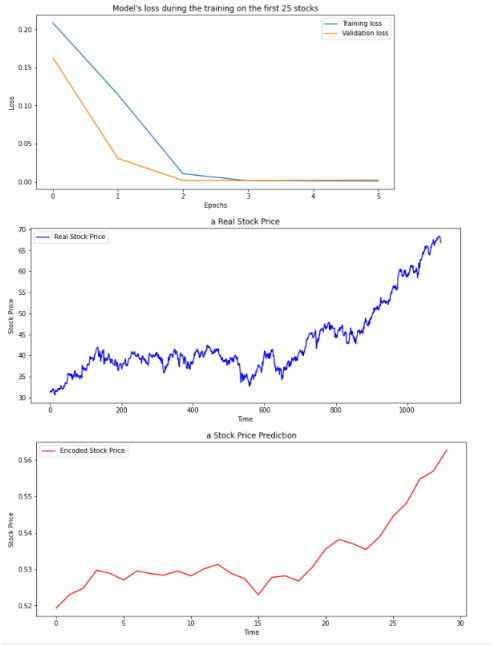
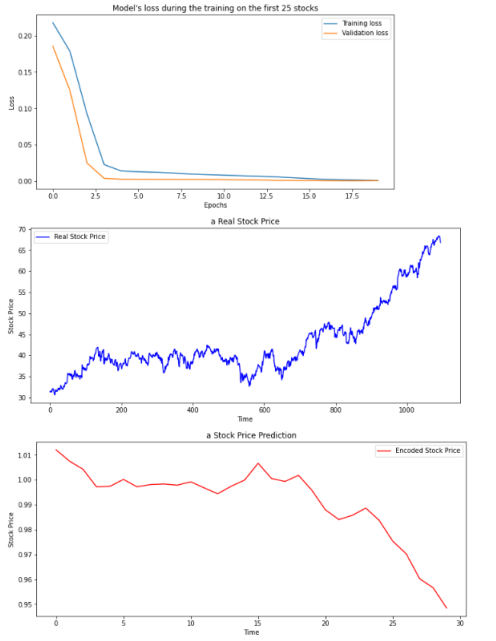


Figure 4: Batch size = 128



1.1.2 Filter layers and Filter sizes

We experimented with **2 to 6** layers, and more specifically:

- 2 layers: [10, 5]
- 3 layers: [10, 10, 5], [20, 10, 5], [30, 10, 8]
- 4 layers: [40, 20, 10, 10]

Figure 5: Batch size = 512, Many epochs to train

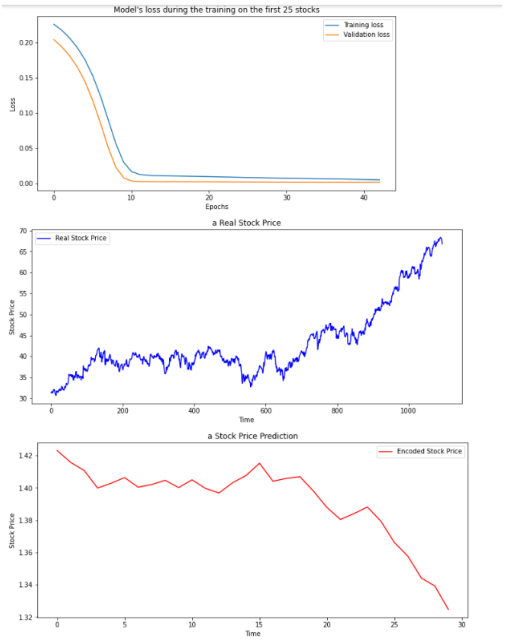
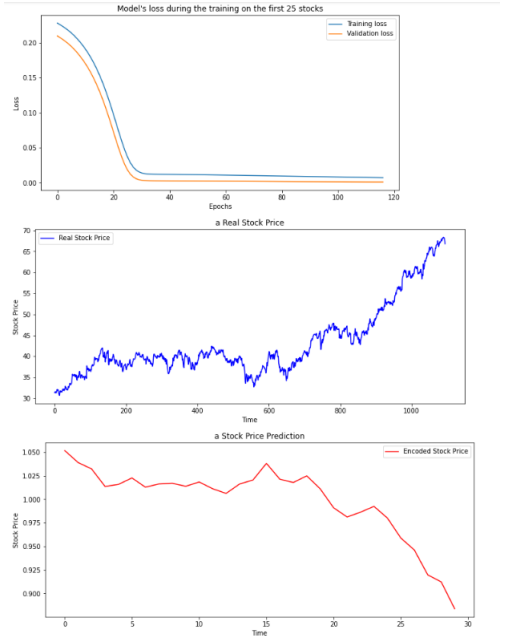


Figure 6: Batch size = 1024, Many epochs to train



- 5 layers: [50, 30, 10, 5, 5], [100, 75, 50, 25, 5]
- 6 layers: [50, 40, 20, 10, 5], [200, 150, 100, 50, 25, 5]

We present the results below. We conclude that **3 layers with 10, 10 and 5 nodes** give a representative compression while minimizing the Train and Validation losses.

1.1.3 Latent dimensions

We experimented with the following Latent Dimensions: **2, 3, 5, 7, 10, 13, 15**.

Figure 7: NN Layers = [10, 5]

1.1.4 Window size

We experimented with the following Window Sizes: **1, 15, 30, 60, 90**. We conclude the the optimal Window Size, providing enough compression of our data while preserving information about major sudden changes is about **30**.

Figure 8: NN Layers = [10, 10, 5]

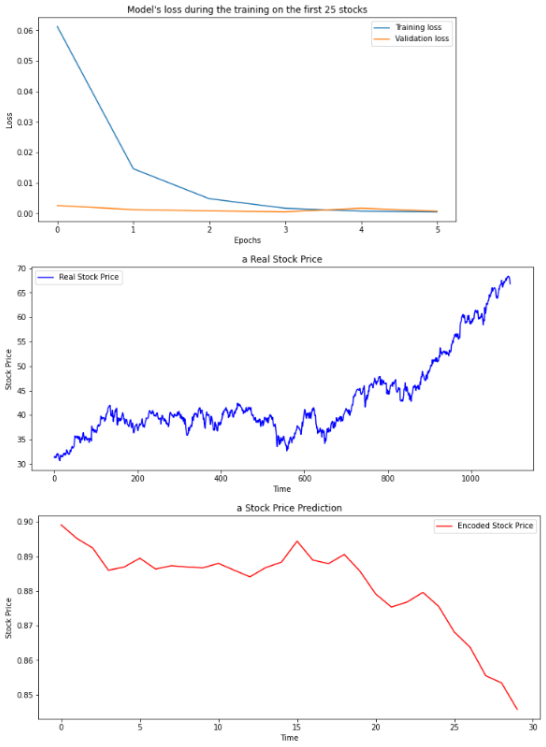


Figure 9: NN Layers = [20, 10, 5]

Figure 10: NN Layers = [30, 10, 8]

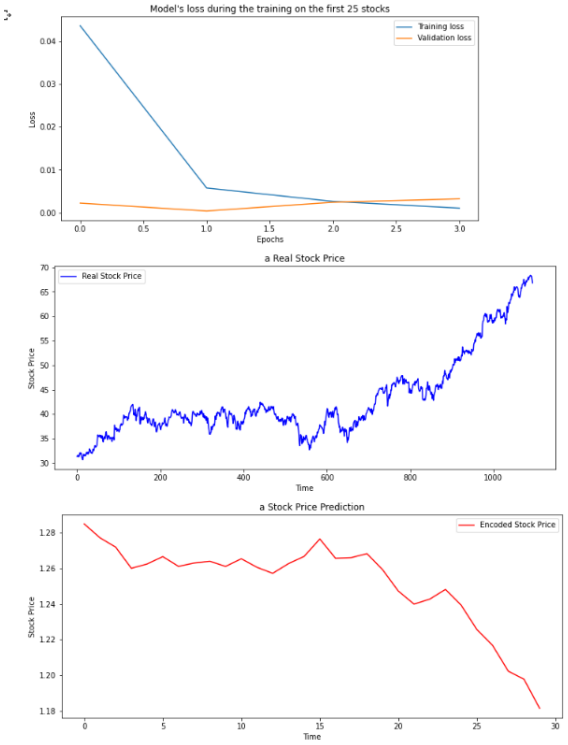


Figure 11: NN Layers = [40, 20, 10, 10]

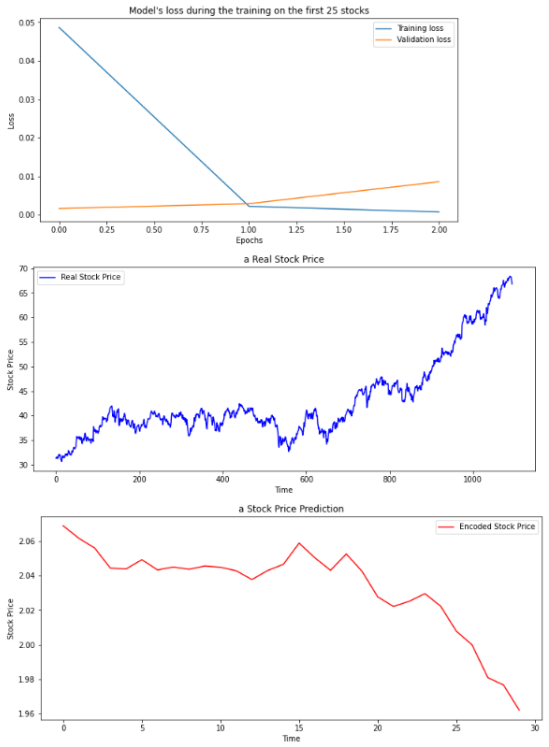


Figure 12: NN Layers = [50, 30, 10, 5, 5]

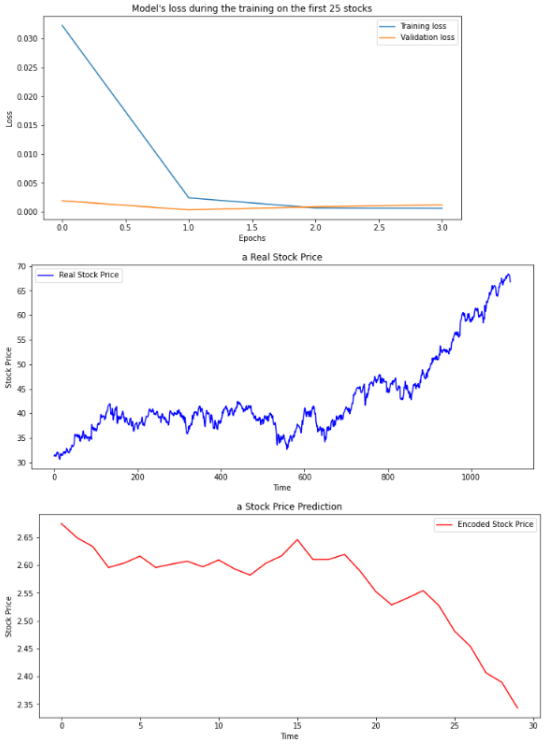


Figure 13: NN Layers = [50, 40, 30, 20, 10, 5]

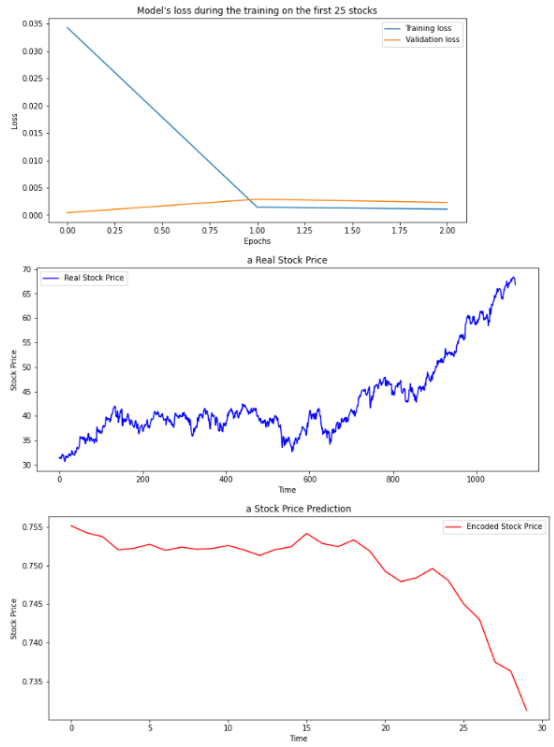


Figure 14: NN Layers = [100, 75, 50, 25, 5]

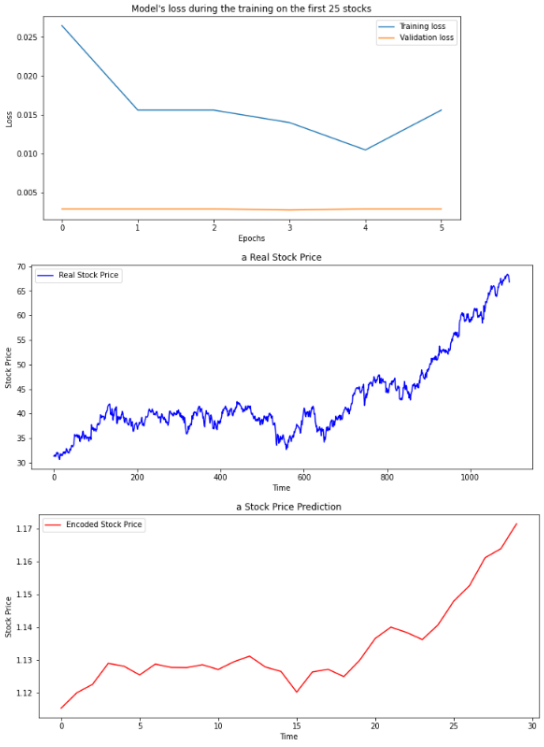


Figure 15: NN Layers = [200, 150, 100, 50, 25, 5]

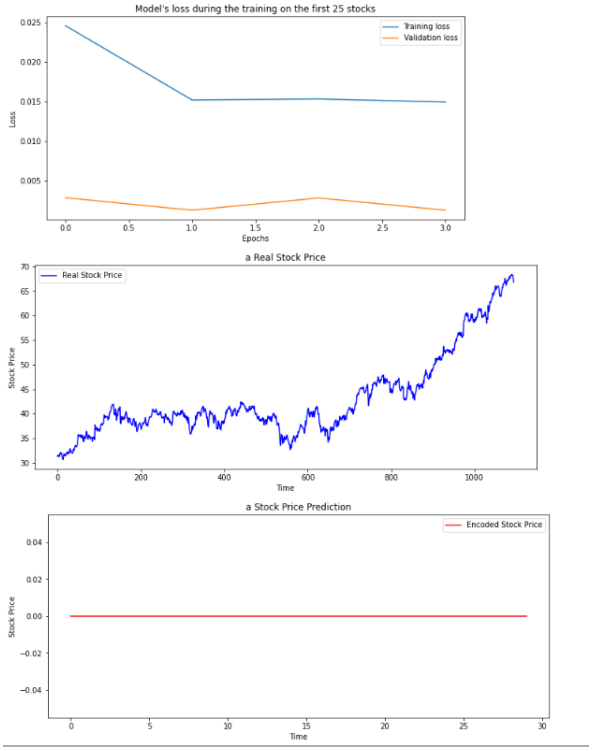


Figure 16: Latent Dim = 2

Figure 17: Latent Dim = 3

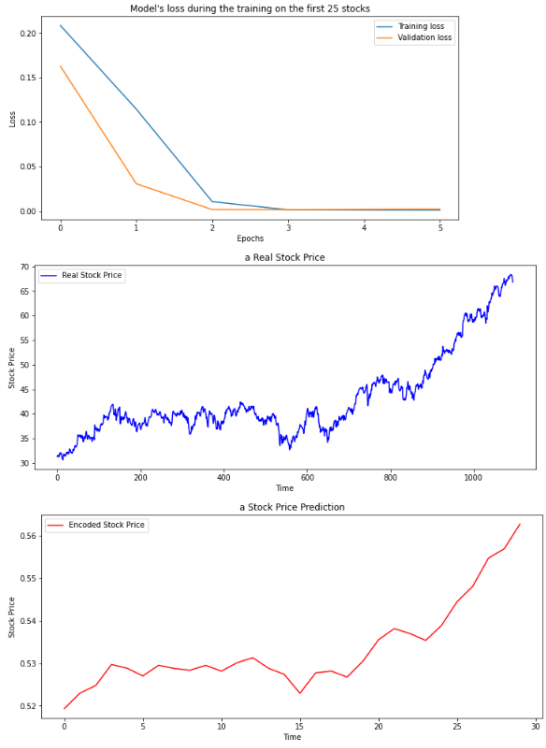


Figure 18: Latent Dim = 5

Figure 19: Latent Dim = 7

Figure 20: Latent Dim = 10

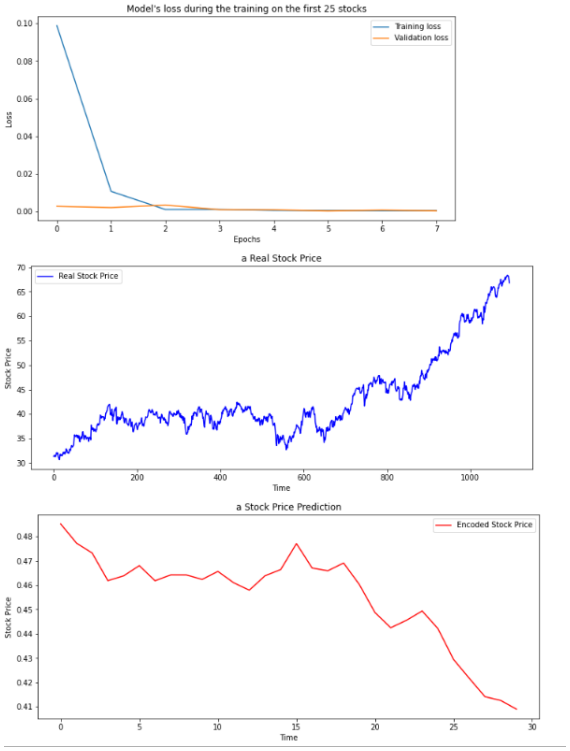


Figure 21: Latent Dim = 13

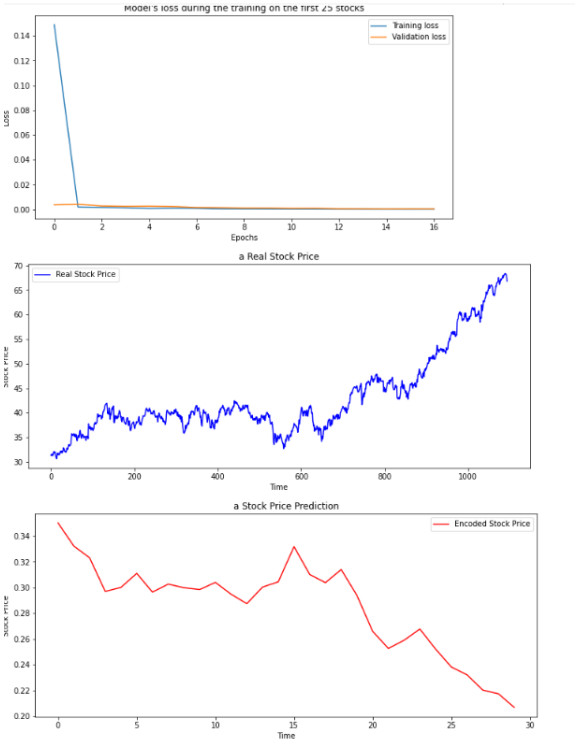


Figure 22: Latent Dim = 15

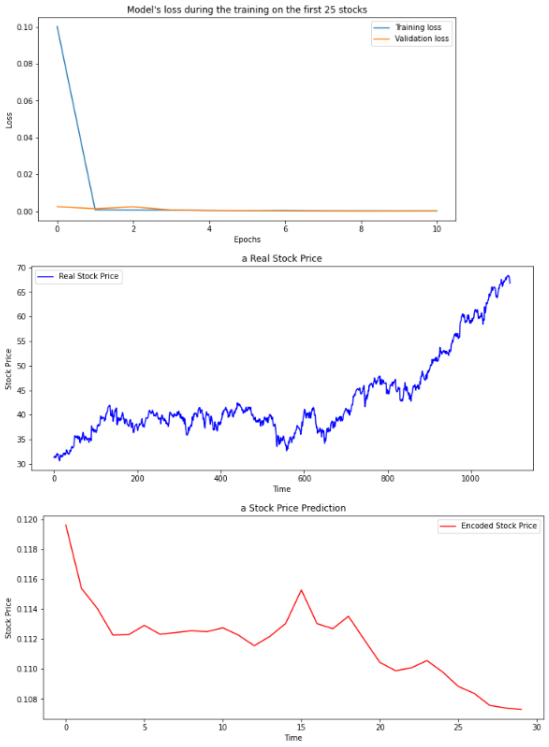


Figure 23: Window size = 1

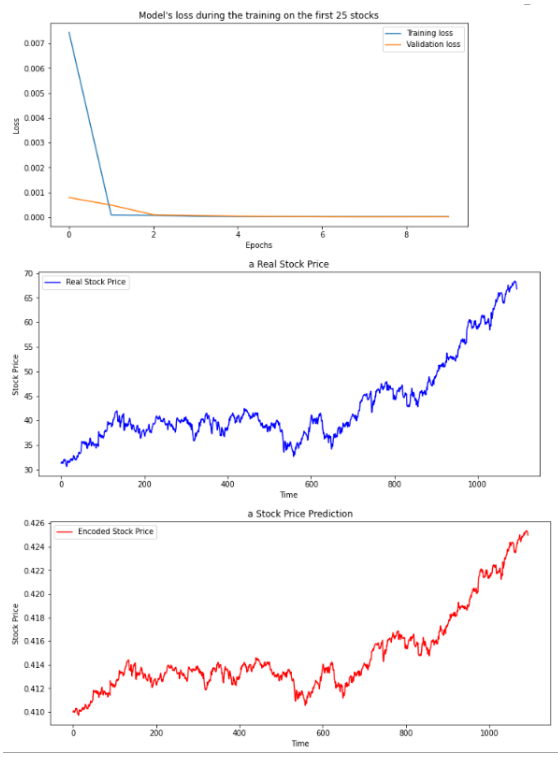


Figure 24: Window size = 15

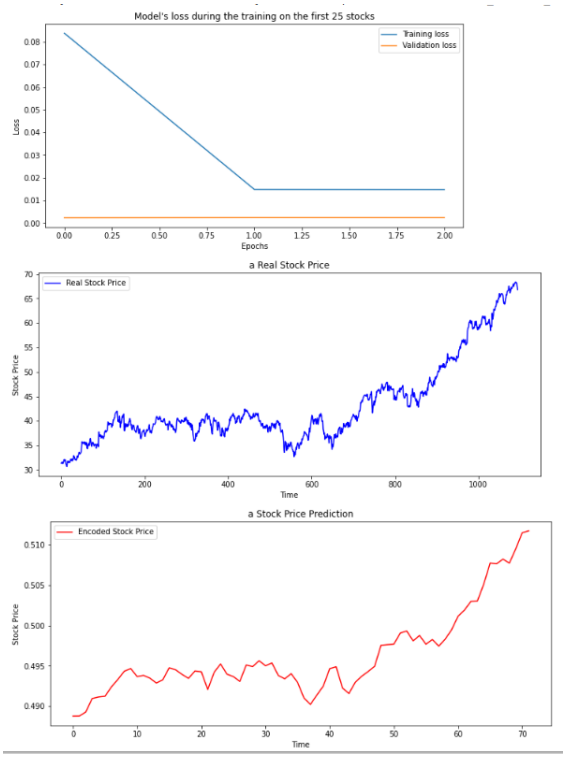


Figure 25: Window size = 30

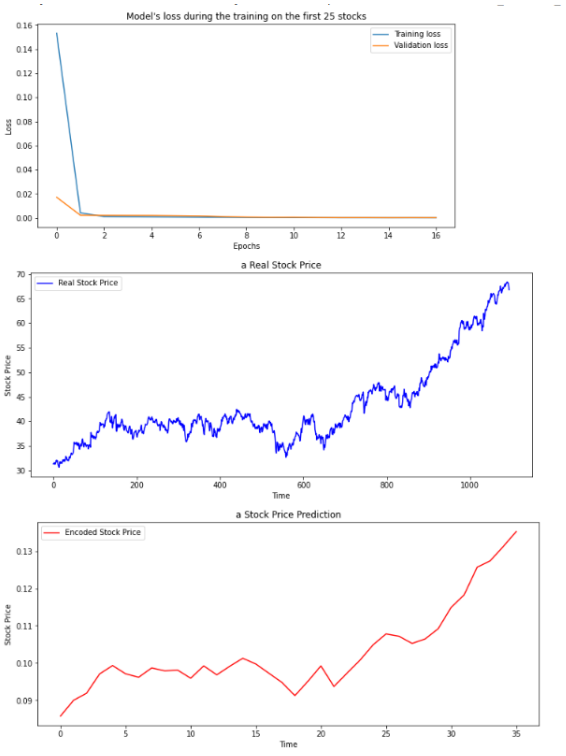


Figure 26: Window size = 60

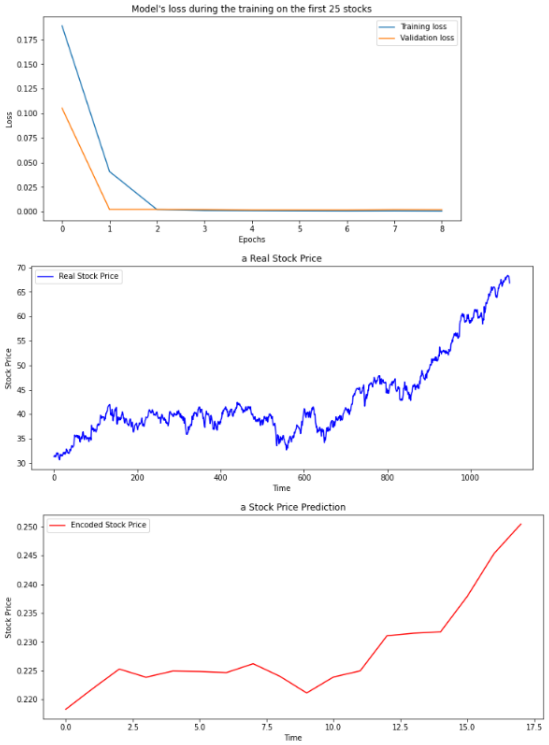


Figure 27: Window size = 90