

HYBRID WAVELET-NEURAL NETWORK MODELS FOR TIME SERIES DATA

A THESIS SUBMITTED TO  
THE GRADUATE SCHOOL OF APPLIED MATHEMATICS  
OF  
MIDDLE EAST TECHNICAL UNIVERSITY

BY

DENİZ KENAN KILIÇ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR  
THE DEGREE OF DOCTOR OF PHILOSOPHY  
IN  
FINANCIAL MATHEMATICS

FEBRUARY 2021





Approval of the thesis:

**HYBRID WAVELET-NEURAL NETWORK MODELS FOR TIME SERIES DATA**

submitted by **DENİZ KENAN KILIÇ** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Financial Mathematics Department, Middle East Technical University** by,

Prof. Dr. A. Sevtap Selçuk-Kestel  
Director, Graduate School of **Applied Mathematics**

\_\_\_\_\_

Prof. Dr. A. Sevtap Selçuk-Kestel  
Head of Department, **Financial Mathematics**

\_\_\_\_\_

Prof. Dr. Ömür Uğur  
Supervisor, **Institute of Applied Mathematics, METU**

\_\_\_\_\_

**Examining Committee Members:**

Prof. Dr. A. Sevtap Selçuk-Kestel  
Institute of Applied Mathematics, METU

\_\_\_\_\_

Prof. Dr. Ömür Uğur  
Institute of Applied Mathematics, METU

\_\_\_\_\_

Prof. Dr. Kasırga Yıldırak  
Actuarial Science, Hacettepe University

\_\_\_\_\_

Assoc. Prof. Dr. Ceylan Talu Yozgatlıgil  
Statistics, METU

\_\_\_\_\_

Assoc. Prof. Dr. Ümit Aksoy  
Mathematics, Atılım University

\_\_\_\_\_

**Date:**

\_\_\_\_\_



**I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.**

Name, Last Name: DENİZ KENAN KILIÇ

Signature :



# ABSTRACT

## HYBRID WAVELET-NEURAL NETWORK MODELS FOR TIME SERIES DATA

Kılıç, Deniz Kenan

Ph.D., Department of Financial Mathematics

Supervisor : Prof. Dr. Ömür Uğur

February 2021, 116 pages

The thesis aims to combine wavelet theory with nonlinear models, particularly neural networks, to find an appropriate time series model structure. Data like financial time series are nonstationary, noisy, and chaotic. Therefore using wavelet analysis helps better modeling in the sense of both frequency and time.

S&P500 ( $\wedge$ GSPC) and NASDAQ ( $\wedge$ IXIC) data are divided into several components by using multiresolution analysis (MRA). Subsequently, each part is modeled by using a suitable neural network structure. In this step, the design of the model is formed according to the pattern of the subseries. Then predictions of each subseries are combined. The combined prediction result is compared to the original time series's prediction result using only a nonlinear model. Moreover, wavelets are used as an activation function for LSTM networks to form a hybrid LSTM-Wavenet model. Furthermore, the hybrid LSTM-Wavenet model is fused with MRA as a proposed method.

In brief, it is studied whether using MRA and hybrid LSTM-Wavenet model decreases the loss or not for both S&P500 and NASDAQ data. Four different modeling methods are used: LSTM, LSTM+MRA, hybrid LSTM-Wavenet, hybrid LSTM-Wavenet+MRA (the proposed method). Results show that using MRA and wavelets as an activation function together decreases error values the most.

Keywords: nonlinear models, neural networks, LSTM, wavelets, time series analysis, finance, multiresolution analysis, wavelet neural network, wavenet, hybrid models



# ÖZ

## ZAMAN SERİSİ VERİLERİ İÇİN HİBRİT DALGACIK-SİNİR AĞI MODELLERİ

Kılıç, Deniz Kenan

Doktora, Finansal Matematik Bölümü

Tez Yöneticisi : Prof. Dr. Ömür Uğur

Şubat 2021, 116 sayfa

Tez, zaman serilerine uygun bir model yapısı bulmak için dalgacık teorisini doğrusal olmayan modellerle, özellikle sinir ağlarıyla birleştirmeyi amaçlamaktadır. Finansal zaman serileri gibi veriler durağan olmayan, gürültülü ve kaotik verilerdir. Bu nedenle dalgacık analizi kullanmak, hem frekans hem de zaman anlamında daha iyi modellemeye yapmaya yardımcı olmaktadır.

S&P500 (^GSPC) ve NASDAQ (^IXIC) verileri çoklu çözünürlük analizi (MRA) kullanılarak birkaç bileşene ayrılmaktadır. Daha sonra, her kısım uygun bir sinir ağı yapısı kullanılarak modellenmektedir. Bu adımda, modelin dizaynı alt serilerin yapısına göre oluşturulmaktadır. Sonra her bir alt dizinin tahminleri birleştirilmektedir. Birleşik tahmin sonucu, sadece doğrusal olmayan bir model kullanılarak tahminlenmiş orijinal zaman serisinin sonucu ile karşılaştırılmaktadır. Dahası, dalgacıklar bir hibrit LSTM-Wavenet modeli oluşturmak üzere LSTM ağları içinde aktivasyon fonksiyonu olarak kullanılmaktadır. Ayrıca, önerilen yöntem olarak hibrit LSTM-Wavenet modeli ve MRA birleştirilmektedir.

Kısacası, MRA ve hibrit LSTM-Wavenet modelinin kullanılmasının hem S&P500 hem de NASDAQ verileri için yitim fonksiyonunu azaltıp azaltmadığı incelenmektedir. Dört farklı modelleme yöntemi kullanılmaktadır: LSTM, LSTM + MRA, hibrit LSTM-Wavenet, hibrit LSTM-Wavenet + MRA (önerilen yöntem). Sonuçlar, dalgacıkların aktivasyon fonksiyonu olarak MRA ile birlikte kullanılmasının hata değerlerini en fazla azalttığını göstermektedir.

Anahtar Kelimeler: doğrusal olmayan modeller, sinir ağıları, LSTM, dalgacıklar, zaman serileri analizi, finans, çoklu çözünürlük analizi (MRA), dalgacık sinir ağı, dalga ağı, hibrit modeller







## ACKNOWLEDGMENTS

I would like to express my very great appreciation to my thesis supervisor Prof. Dr. Ömür Uğur for his patient guidance, enthusiastic encouragement, and valuable advice during the development and preparation of this thesis. His willingness to give his time and to share his experiences have brightened my path.

And, I thank all members of the Institute of Applied Mathematics of Middle East Technical University for their kindness and help.

Furthermore, I would like to especially thank my wife Dilek Aydoğan Kılıç for her friendship, support, and help.

Finally, I would like to express my pleasure in having such a great family. Thank to my mother Nermin Kılıç and to my father Erdoğan Kılıç for their endless love and existence.

I devote the thesis to my wife Dilek Aydoğan Kılıç, to my son Ilgar Ege Kılıç, to my mother Nermin Kılıç and to my father Erdoğan Kılıç.



# TABLE OF CONTENTS

ABSTRACT . . . . .	vii
ÖZ . . . . .	ix
ACKNOWLEDGMENTS . . . . .	xiii
TABLE OF CONTENTS . . . . .	xv
LIST OF TABLES . . . . .	xvii
LIST OF FIGURES . . . . .	xxi
LIST OF ABBREVIATIONS . . . . .	xxv
CHAPTERS	
1 INTRODUCTION . . . . .	1
1.1 Motivation . . . . .	1
1.2 Literature Review . . . . .	2
1.3 Plan of the Thesis . . . . .	6
2 NEURAL NETWORKS . . . . .	7
2.1 Long Short-Term Memory (LSTM) . . . . .	8
3 WAVELETS . . . . .	11
3.1 Maximal Overlap Discrete Wavelet Transform (MODWT) . . . . .	11

3.2	Multiresolution Analysis (MRA) . . . . .	13
4	WAVELET NEURAL NETWORK . . . . .	17
4.1	Wavenets . . . . .	17
4.2	Polynomial Powers of Sigmoid (PPS) . . . . .	18
5	EMPIRICAL RESULTS . . . . .	23
5.1	Descriptive Statistics . . . . .	27
5.2	LSTM Model without MRA . . . . .	29
5.3	LSTM Model with MRA . . . . .	50
5.4	Hybrid LSTM-Wavenet Model without MRA . . . . .	63
5.5	Hybrid LSTM-Wavenet Model with MRA . . . . .	73
5.6	Discussion . . . . .	95
6	CONCLUSION AND OUTLOOK . . . . .	101
	REFERENCES . . . . .	103
	APPENDICES	
A	MODELS OF API STRUCTURES . . . . .	109
	CURRICULUM VITAE . . . . .	115

## LIST OF TABLES

Table 5.1	Details of Data Preprocessing and Hyperparameter Optimization Parts Given in Figure 5.1 with and without MRA . . . . .	25
Table 5.2	Details of Modeling & Learning and Prediction & Visualization Parts Given in Figure 5.1 with and without MRA . . . . .	26
Table 5.3	Descriptive Statistics of S&P500 Data . . . . .	29
Table 5.4	Descriptive Statistics of NASDAQ Data . . . . .	32
Table 5.5	LSTM Model, Configuration 1 (S&P500): Talos Configuration . . .	33
Table 5.6	LSTM Model, Configuration 1 (S&P500): Mean Scores for the Train set and the Test set by Running 1000 Experiments . . . . .	33
Table 5.7	LSTM Model, Configuration 2 (S&P500): Talos Configuration . . .	34
Table 5.8	LSTM Model, Configuration 2 (S&P500): Mean Scores for the Train set and the Test set by Running 1000 Experiments . . . . .	34
Table 5.9	LSTM Model, Configuration 3 (S&P500): Talos Configuration . . .	35
Table 5.10	LSTM Model, Configuration 3 (S&P500): Mean Scores for the Train set and the Test set by Running 1000 Experiments . . . . .	35
Table 5.11	LSTM Model, Configuration 4 (S&P500): Talos Configuration . . .	36
Table 5.12	LSTM Model, Configuration 4 (S&P500): Mean Scores for the Train set and the Test set by Running 1000 Experiments . . . . .	36
Table 5.13	LSTM Model, Configuration 5 (S&P500): Talos Configuration . . .	37
Table 5.14	LSTM Model, Configuration 5 (S&P500): Mean Scores for the Train set and the Test set by Running 1000 Experiments . . . . .	37
Table 5.15	LSTM Model, Configuration 6 (S&P500): Talos Configuration . . .	38
Table 5.16	LSTM Model, Configuration 6 (S&P500): Mean Scores for the Train set and the Test set by Running 1000 Experiments . . . . .	38
Table 5.17	LSTM Model (S&P500): Scores of the all Configurations . . . . .	39

Table 5.18 LSTM Model, Configuration 1 (NASDAQ): Talos Configuration . . .	42
Table 5.19 LSTM Model, Configuration 1 (NASDAQ): Mean Scores for the Train set and the Test set by Running 1000 Experiments . . . . .	42
Table 5.20 LSTM Model, Configuration 2 (NASDAQ): Talos Configuration . . .	43
Table 5.21 LSTM Model, Configuration 2 (NASDAQ): Mean Scores for the Train set and the Test set by Running 1000 Experiments . . . . .	43
Table 5.22 LSTM Model, Configuration 3 (NASDAQ): Talos Configuration . . .	44
Table 5.23 LSTM Model, Configuration 3 (NASDAQ): Mean Scores for the Train set and the Test set by Running 1000 Experiments . . . . .	44
Table 5.24 LSTM Model, Configuration 4 (NASDAQ): Talos Configuration . . .	45
Table 5.25 LSTM Model, Configuration 4 (NASDAQ): Mean Scores for the Train set and the Test set by Running 1000 Experiments . . . . .	45
Table 5.26 LSTM Model, Configuration 5 (NASDAQ): Talos Configuration . . .	46
Table 5.27 LSTM Model, Configuration 5 (NASDAQ): Mean Scores for the Train set and the Test set by Running 1000 Experiments . . . . .	46
Table 5.28 LSTM Model, Configuration 6 (NASDAQ): Talos Configuration . . .	47
Table 5.29 LSTM Model, Configuration 6 (NASDAQ): Mean Scores for the Train set and the Test set by Running 1000 Experiments . . . . .	47
Table 5.30 LSTM Model (NASDAQ): Scores of the all Configurations . . . . .	50
Table 5.31 LSTM Model+MRA, Configuration 1 (S&P500): Talos Configuration	52
Table 5.32 LSTM Model+MRA, Configuration 1 (S&P500): Mean Scores of the Wavelet Levels for the Train set and the Test set by Running 1000 Experiments . . . . .	53
Table 5.33 LSTM Model+MRA, Configuration 1 (S&P500): Mean Scores for the Synthesized Train set and the Synthesized Test set by Running 1000 Experiments . . . . .	53
Table 5.34 LSTM Model+MRA, Configuration 1 (NASDAQ): Talos Configu- ration . . . . .	58
Table 5.35 LSTM Model+MRA, Configuration 1 (NASDAQ): Mean Scores of the Wavelet Levels for the Train set and the Test set by Running 1000 Experiments . . . . .	59



Table 5.36 LSTM Model+MRA, Configuration 1 (NASDAQ): Mean Scores for the Synthesized Train set and the Synthesized Test set by Running 1000 Experiments . . . . .	59
Table 5.37 Hybrid LSTM-Wavenet Model, Configuration 1 (S&P500): Talos Configuration . . . . .	65
Table 5.38 Hybrid LSTM-Wavenet Model, Configuration 1 (S&P500): Mean Scores for the Train set and the Test set by Running 1000 Experiments . . . . .	65
Table 5.39 Hybrid LSTM-Wavenet Model by API Structure, Configuration 2 (S&P500): Talos Configuration . . . . .	68
Table 5.40 Hybrid LSTM-Wavenet Model by API Structure, Configuration 2 (S&P500): Mean Scores for the Train set and the Test set by Running 1000 Experiments . . . . .	69
Table 5.41 Hybrid LSTM-Wavenet Model, Configuration 1 (NASDAQ): Talos Configuration . . . . .	70
Table 5.42 Hybrid LSTM-Wavenet Model, Configuration 1 (NASDAQ): Mean Scores for the Train set and the Test set by Running 1000 Experiments . . . . .	70
Table 5.43 Hybrid LSTM-Wavenet Model by API Structure, Configuration 2 (NASDAQ): Talos Configuration . . . . .	73
Table 5.44 Hybrid LSTM-Wavenet Model by API Structure, Configuration 2 (NASDAQ): Mean Scores for the Train set and the Test set by Running 1000 Experiments . . . . .	74
Table 5.45 Hybrid LSTM-Wavenet Model+MRA, Configuration 1 (S&P500): Talos Configuration . . . . .	76
Table 5.46 Hybrid LSTM-Wavenet Model+MRA, Configuration 1 (S&P500): Mean Scores of the Wavelet Levels for the Train set and the Test set by Running 1000 Experiments . . . . .	77
Table 5.47 Hybrid LSTM-Wavenet Model+MRA, Configuration 1 (S&P500): Mean Scores for the Synthesized Train set and the Synthesized Test set by Running 1000 Experiments . . . . .	77
Table 5.48 Hybrid LSTM-Wavenet Model+MRA by API Structure, Configuration 2 (S&P500): Talos Configuration . . . . .	82
Table 5.49 Hybrid LSTM-Wavenet Model+MRA by API Structure, Configuration 2 (S&P500): Mean Scores of the Wavelet Levels for the Train set and the Test set by Running 1000 Experiments . . . . .	83

Table 5.50 Hybrid LSTM-Wavenet Model+MRA by API Structure, Configuration 2 (S&P500): Mean Scores for the Synthesized Train set and the Synthesized Test set by Running 1000 Experiments . . . . .	83
Table 5.51 Hybrid LSTM-Wavenet Model+MRA, Configuration 1 (NASDAQ): Talos Configuration . . . . .	86
Table 5.52 Hybrid LSTM-Wavenet Model+MRA, Configuration 1 (NASDAQ): Mean Scores of the Wavelet Levels for the Train set and the Test set by Running 1000 Experiments . . . . .	87
Table 5.53 Hybrid LSTM-Wavenet Model+MRA, Configuration 1 (NASDAQ): Mean Scores for the Synthesized Train set and the Synthesized Test set by Running 1000 Experiments . . . . .	87
Table 5.54 Hybrid LSTM-Wavenet Model+MRA by API Structure, Configuration 2 (NASDAQ): Talos Configuration . . . . .	93
Table 5.55 Hybrid LSTM-Wavenet Model+MRA by API Structure, Configuration 2 (NASDAQ): Mean Scores of the Wavelet Levels for the Train set and the Test set by Running 1000 Experiments . . . . .	94
Table 5.56 Hybrid LSTM-Wavenet Model+MRA by API Structure, Configuration 2 (NASDAQ): Mean Scores for the Synthesized Train set and the Synthesized Test set by Running 1000 Experiments . . . . .	94
Table 5.57 Summary Table for Results (S&P500): Mean Scores for the Train set and the Test set by Running 1000 Experiments . . . . .	98
Table 5.58 Summary Table for Results (NASDAQ): Mean Scores for the Train set and the Test set by Running 1000 Experiments . . . . .	99

## LIST OF FIGURES

Figure 2.1	Structure of MNN. . . . .	8
Figure 2.2	Structure of a Recurrent Neural Network System. . . . .	9
Figure 2.3	Structure of a Long Short-Term Memory. . . . .	9
Figure 3.1	Flowchart of the pyramid algorithm for decomposing. . . . .	12
Figure 3.2	Flowchart of the pyramid algorithm for synthesis. . . . .	12
Figure 3.3	Nested Subspaces . . . . .	14
Figure 4.1	Polynomial Wavelet Functions Generated by the nth Derivative of the Sigmoid Function . . . . .	20
Figure 5.1	The Flowchart of Four Different Methods Applied in the Study . . .	24
Figure 5.2	Partitioning Graphs and Histograms . . . . .	28
Figure 5.3	Visuals and Histograms of the Training Set, the Validation Set and the Test Set of S&P500 . . . . .	30
Figure 5.4	Visuals and Histograms of the Training Set, the Validation Set and the Test Set of NASDAQ . . . . .	31
Figure 5.5	LSTM Model, Configuration 3 (S&P500): Model Structure . . . . .	40
Figure 5.6	LSTM Model, Configuration 3 (S&P500): Results . . . . .	41
Figure 5.7	LSTM Model, Configuration 3 (NASDAQ): Model Structure . . . . .	48
Figure 5.8	LSTM Model, Configuration 3 (NASDAQ): Results . . . . .	49
Figure 5.9	LSTM Model+MRA, Configuration 1 (S&P500): Model Structure of the First Detail . . . . .	51
Figure 5.10	LSTM Model+MRA, Configuration 1 (S&P500): Model Structure of the Second Detail . . . . .	53

Figure 5.11 LSTM Model+MRA, Configuration 1 (S&P500): Model Structure of the Approximation . . . . .	54
Figure 5.12 LSTM Model+MRA, Configuration 1 (S&P500): Loss Values of Each Level . . . . .	55
Figure 5.13 LSTM Model+MRA, Configuration 1 (S&P500): Train and Test Predictions of the First Detail, the Second Detail and the Approximation Parts . . . . .	56
Figure 5.14 LSTM Model+MRA, Configuration 1 (S&P500): Reconstructed Results . . . . .	57
Figure 5.15 LSTM Model+MRA, Configuration 1 (NASDAQ): Model Structure of the First Detail . . . . .	57
Figure 5.16 LSTM Model+MRA, Configuration 1 (NASDAQ): Model Structure of the Second Detail . . . . .	57
Figure 5.17 LSTM Model+MRA, Configuration 1 (NASDAQ): Model Structure of the Approximation . . . . .	60
Figure 5.18 LSTM Model+MRA, Configuration 1 (NASDAQ): Loss Values of Each Level . . . . .	61
Figure 5.19 LSTM Model+MRA, Configuration 1 (NASDAQ): Train and Test Predictions of the First Detail, the Second Detail and the Approximation Parts . . . . .	62
Figure 5.20 LSTM Model+MRA, Configuration 1 (NASDAQ): Reconstructed Results . . . . .	63
Figure 5.21 Hybrid LSTM-Wavenet Model, Configuration 1 (S&P500): Model Structure . . . . .	66
Figure 5.22 Hybrid LSTM-Wavenet Model, Configuration 1 (S&P500): Results	67
Figure 5.23 Hybrid LSTM-Wavenet Model by API Structure, Configuration 2 (S&P500): Results . . . . .	69
Figure 5.24 Hybrid LSTM-Wavenet Model, Configuration 1 (NASDAQ): Model Structure . . . . .	71
Figure 5.25 Hybrid LSTM-Wavenet Model, Configuration 1 (NASDAQ): Results	72
Figure 5.26 Hybrid LSTM-Wavenet Model by API Structure, Configuration 2 (NASDAQ): Results . . . . .	74

Figure 5.27 Hybrid LSTM-Wavenet Model+MRA, Configuration 1 (S&P500): Model Structure of the First Detail . . . . .	75
Figure 5.28 Hybrid LSTM-Wavenet Model+MRA, Configuration 1 (S&P500): Model Structure of the Second Detail . . . . .	77
Figure 5.29 Hybrid LSTM-Wavenet Model+MRA, Configuration 1 (S&P500): Model Structure of the Approximation . . . . .	78
Figure 5.30 Hybrid LSTM-Wavenet Model+MRA, Configuration 1 (S&P500): Loss Values of Each Level . . . . .	79
Figure 5.31 Hybrid LSTM-Wavenet Model+MRA, Configuration 1 (S&P500): Train and Test Predictions of the First Detail, the Second Detail and the Approximation Parts of S&P500 . . . . .	80
Figure 5.32 Hybrid LSTM-Wavenet Model+MRA, Configuration 1 (S&P500): Reconstructed Results . . . . .	81
Figure 5.33 Hybrid LSTM-Wavenet Model+MRA by API Structure, Configu- ration 2 (S&P500): Model Structure of the First Detail . . . . .	81
Figure 5.34 Hybrid LSTM-Wavenet Model+MRA by API Structure, Configu- ration 2 (S&P500): Model Structure of the Second Detail . . . . .	83
Figure 5.35 Hybrid LSTM-Wavenet Model+MRA, Configuration 2 (S&P500): Loss Values of Each Level . . . . .	84
Figure 5.36 Hybrid LSTM-Wavenet Model+MRA by API Structure, Configu- ration 2 (S&P500): Train and Test Predictions of the First Detail, the Second Detail and the Approximation Parts of S&P500 . . . . .	85
Figure 5.37 Hybrid LSTM-Wavenet Model+MRA, Configuration 2 (S&P500): Reconstructed Results . . . . .	87
Figure 5.38 Hybrid LSTM-Wavenet Model+MRA, Configuration 1 (NASDAQ): Model Structure of the First Detail . . . . .	88
Figure 5.39 Hybrid LSTM-Wavenet Model+MRA, Configuration 1 (NASDAQ): Model Structure of the Second Detail . . . . .	89
Figure 5.40 Hybrid LSTM-Wavenet Model+MRA, Configuration 1 (NASDAQ): Model Structure of the Approximation . . . . .	89
Figure 5.41 Hybrid LSTM-Wavenet Model+MRA, Configuration 1 (NASDAQ): Loss Values of Each Level . . . . .	90

Figure 5.42 Hybrid LSTM-Wavenet Model+MRA, Configuration 1 (NASDAQ): Train and Test Predictions of the First Detail, the Second Detail and the Approximation Parts of S&P500 . . . . .	91
Figure 5.43 Hybrid LSTM-Wavenet Model+MRA, Configuration 1 (NASDAQ): Reconstructed Results . . . . .	92
Figure 5.44 Hybrid LSTM-Wavenet Model+MRA by API Structure, Configu- ration 2 (NASDAQ): Model Structure of the First Detail . . . . .	92
Figure 5.45 Hybrid LSTM-Wavenet Model+MRA by API Structure, Configu- ration 2 (NASDAQ): Model Structure of the Second Detail . . . . .	94
Figure 5.46 Hybrid LSTM-Wavenet Model+MRA by API Structure, Configu- ration 2 (NASDAQ): Loss Values of Each Level . . . . .	95
Figure 5.47 Hybrid LSTM-Wavenet Model+MRA by API Structure, Configu- ration 2 (NASDAQ): Train and Test Predictions of the First Detail, the Second Detail and the Approximation Parts . . . . .	96
Figure 5.48 Hybrid LSTM-Wavenet Model+MRA by API Structure, Configu- ration 2 (NASDAQ): Reconstructed Results . . . . .	97
Figure A.1 Hybrid LSTM-Wavenet Model by API Structure, Configuration 2 (S&P500): Model Structure . . . . .	110
Figure A.2 Hybrid LSTM-Wavenet Model by API Structure, Configuration 2 (NASDAQ): Model Structure . . . . .	111
Figure A.3 Hybrid LSTM-Wavenet Model+MRA by API Structure, Configu- ration 2 (S&P500): Model Structure of the Approximation . . . . .	112
Figure A.4 Hybrid LSTM-Wavenet Model+MRA by API Structure, Configu- ration 2 (NASDAQ): Model Structure of the Approximation . . . . .	113

## LIST OF ABBREVIATIONS

ANN	Artificial Neural Network
ARFIMA	Autoregressive Fractionally Integrated Moving Average
ARIMA	Autoregressive Integrated Moving Average
AWNN	Adaptive Wavelet Neural Network
BP	Backpropagation
CNN	Convolutional Neural Network
CWT	Continuous Wavelet Transform
ddof	Delta Degrees of Freedom
DNN	Deep Neural Network
DW	Discrete Wavelet
DWT	Discrete Wavelet Transform
EMG	Electromyography
EVS	Explained Variance Score
FEBANN	Feed Forward Error Backpropagation Artificial Neural Network
FFNN	Feed Forward Neural Network
FNN	Fuzzy Neural Network
GRU	Gated Recurrent Unit
KDE	Kernel Density Estimate
LSTM	Long Short-Term Memory
LM	Levenberg-Marquardt
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MdAE	Median Absolute Error
ME	Maximum Residual Error
MLP	Multilayer Perceptron
MNN	Multilayer Neural Network
MODWT	Maximal Overlap Discrete Wavelet Transform
MRA	Multiresolution Analysis

MSE	Mean Square Error
MWFWNN	Multiple Wavelet Functions Wavelet Neural Network
NASDAQ	National Association of Securities Dealers Automated Quotations
NIR	Near-Infrared
NSE	National Stock Exchange
OS	Operating System
PSO	Particle Swarm Optimization
PPS	Polynomial Powers of Sigmoid
$R^2$	Coefficient of Determination
RBF	Radial Basis Function
REG	Multiple Linear Regression
ReLU	Rectified Linear Unit
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
RvNN	Recursive Neural Network
S&P500	Standard & Poor's 500
SARIMA	Seasonal Autoregressive Integrated Moving Average
SRMSE	Scaled Root mean Square Error
Std	Standard Deviation
STFT	Short-Time Fourier Transform
SVM	Support Vector Machine
SVR	Support Vector Regression
WA-ANN	Wavelet-Neural Network
WNN	Wavelet Neural Network
WREG	DWT-Based Multiple Linear Regression



# CHAPTER 1

## INTRODUCTION

Most of the time series in many fields like geology, astronomy, economy, politics, robotics, meteorology, medicine, control engineering, finance, etc., are nonstationary and chaotic. Hence nonlinear models are more appropriate than linear models for such complex time series.

Neural networks are one of the mostly used and widespread nonlinear models due to their wide varieties and flexibilities.

On the other hand, wavelets are used in a vast area, and where they are used are increasing day by day. In most cases, the frequency domain analysis is essential for complex time series. Furthermore, for both linear and nonlinear time series, the multiresolution analysis provides more reliable modeling and forecasting results (see [31]).

In the literature review below, studies which combine wavelet analyzes and nonlinear modeling are discussed.

Most studies use wavelets either to decompose data to use subseries as inputs of regular neural networks or just for activation functions of wavelet neural networks. In this paper, these two separate studies are merged. First of all, the effect of the two different wavelet approaches on predicting the financial time series is analyzed separately. Next, the two approaches are combined based on the prediction results of the financial time series.

Main scope of the thesis is to analyze whether utilizing wavelets increase accuracy performance in neural network modeling for financial time series.

### 1.1 Motivation

One step ahead prediction is significant for stock market investors since a small change in the prediction of the market index will significantly affect the profit or the loss. To predict the next step, using observed data is more suitable than using predictions in the model. On the other hand, if the aim is predicting an extended period (multi steps), it would be better to use consecutive predicted values to reach the recent prediction.

Our motivation is to improve prediction performances of stock market indexes to lead investors' behaviors on buying and selling. Moreover, it is noticed that many recent works of literature and competitions on online platforms cover subjects about classic and advanced machine learning. With combining MRA and wavenet with machine learning techniques, we aim to contribute to both literature and online studies in terms of both science and software.

## 1.2 Literature Review

Masset mentions in [38] that if the system relies on more than one frequency component, time-domain analysis is inadequate to reveal important information of the original data. Besides, it is stated in [31] that modeling after multiresolution analysis (MRA) gives better-fitted models, especially for forecasting the time series. In other words, dividing data into different scales and fitting various models to each scale provides better results than using a single model only.

In [57] wavelet multilayer perceptron (MLP) neural network is handled. It is stated that using wavelets helps discover hidden time frequency information. Moreover, Teo, Wang, and Lin [57] say that combining wavelet analysis with MLP networks gives better performance. On the other hand, it is mentioned that regular neural networks catch information only on the finest resolution of a signal. However, real human neural networks can process all scales of a signal. While mother wavelets' resolution increases by using  $2^j$  component, decomposition approaches to the original signal. Therefore, one can reconstruct the actual signal by using decomposed signals. Further, after removing less essential wavelets, the wavelet MLP neural network gives better results. On the other hand, it is mentioned that the wavelet MLP neural network without elimination has no critical improvement over the usual MLP network.

In [8] wavelet transforms, ordinary neural networks and statistical time series analytical techniques are applied to several exchange rates for the prediction. After decomposing time series, different neural networks are produced, trained, and used to predict each scale's exchange rates. Mean absolute error (MAE), mean absolute percentage error (MAPE), mean square error (MSE), and root mean square error (RMSE) are used to compare the results. The study is following statistical feature extraction, preprocessing, wavelet analysis, modeling and training of networks, and forecasting. However, results show that using wavelets fetches no improvement in their structure. It is stated that exchange rates are nonstationary, noisy, and chaotic time series. Moreover, Bozic and Babic [8] mention that linear models are not suitable for nonlinear time series. Wavelets are not affected by nonstationary cases [27]. Further, it is stated that an artificial neural network (ANN) is a multivariate, nonparametric statistical method that can represent any nonlinear function without a pre-assumption on the signal.

In [48] the monthly rainfall at Darjeeling is predicted by using the ANN model. Inputs of ANN are created by applying MRA to the data. Outcomes show that a wavelet-based neural network gives better results than ANN gives.

Nonlinear monthly precipitation time series is handled in [51]. The time series is de-

composed by wavelet analysis. Low-frequency subseries are forecasted using the seasonal autoregressive integrated moving average (SARIMA) model, and high-frequency subseries are forecasted using ANN. Afterward, predicted subseries are rebuilt to forecast the precipitation of future single months. It is stated that the wavelet-SARIMA-ANN model has better results than wavelet-ANN and wavelet-SARIMA models. The autoregressive integrated moving average (ARIMA) model restricts requiring stationary and linear time series. However, most of the financial time series are nonstationary and nonlinear.

In [44], wavelets and neural networks are combined to predict Turkey's daily precipitation. Partal and Cigizoglu [44] firstly decompose the original time series into several subseries by using discrete wavelet transform (DWT). Then, they select useful discrete wavelets to use as the inputs of ANNs for the estimation of daily precipitation. They state that selecting the dominant discrete wavelets (DWs) has a large positive effect on the ANN model's performance. Moreover, they use correlation coefficients between each DW of the meteorological data and the original precipitation data to select the ANN model inputs. The hybrid model yields better accuracy results than the classical ANN model.

In [5], steps of the wavelet-based forecasting are given as follows: firstly, data is divided into various elements; secondly, each component is modeled and forecast individually; thirdly, predictions are summed up. Basta [5] also mentions boundary conditions of wavelets, details of maximal overlap discrete wavelet transform (MODWT) and prediction with boundary conditions. Choosing the type of wavelet transform and the type of filter is vital for better prediction results.

In [41] hybrid wavelet recurrent neural network is used to predict over time the number of connection requests for service. It is stated that the model is very successful in estimating such number of connections.

Shah and Debnath combine DWT and Levenberg-Marquardt (LM) ANN to forecast yield spread for output growth in [52]. Firstly, they decompose different yield spreads by using several DWT filters. Secondly, they use subseries as inputs of ANN to forecast output growth. Outcomes indicate that the predictive powers of yield spread in the short term and policy-relevant areas of the yield curve are nice. However, the predictive power of yield spreads in the long term is not suitable for output growth.

In [30] wavelets and nonlinear models are used to predict NIFTY 50 (NSE) index financial data. It is stated that financial time series are nonlinear and nonstationary. Moreover, models like ARIMA have some problems with stock prices since such time series does not have a normal distribution, are not stationary and not linear. On the other hand, models like support vector regression (SVR) and ANN can deal with such time series, like stock prices. However, they may have an overfitting problem. One has to choose model parameters carefully and preprocess the data according to the type of the signal. Using wavelets is a better decomposing method than only using trend, seasonal and random components. Classical decomposition is good for linear time series [58]. There are many types of wavelet filters, such as Haar, Daubechies, Morlet, and Mexican Hat. One needs to choose the type of filter according to the problem. It is stated that ANN models need fewer pre-assumptions since they are broadly data-based and adaptive. Furthermore, ANN systems can model and predict

the nonlinear data. Practically, Jothimani, Shankar, and Yadav [30] decompose the original signal. Afterward, they use ANN and SVR models for predictions. For both hybrid models, the accuracy of prediction is higher than the accuracy of the classical models.

In [11] wavelet transform and backpropagation (BP) ANN are combined to create a hybrid method. This hybrid model is used to predict stock market data. The model is applied to five different data sets. Accuracy estimations for all data sets indicate that the hybrid model gives better results than classical models.

In [64] several wavelet functions are used with neural networks for target threat assessment. The Morlet mother wavelet function shows the best performance. Moreover, prediction results of the multiple wavelet functions wavelet neural network (MFWNN) are better than wavelet neural network (WNN), BP and particle swarm optimization and support vector machine (PSO\_SVM) methods.

Modified hybrid models are used to predict natural gas prices in [29]. Using wavelets with ANN and ARIMA gives better results than using only ANN or ARIMA models.

In [56] forecasting of exchange rate data is handled by combining wavelet transform, neural network, and statistical analysis techniques. In empirical results, it is seen that hybrid methods outperform the classical ones.

Wang et al. use a wavelet-based neural network structure for two deep learning models in time series classification and forecasting in [65]. Experimental results propose the hybrid methods.

Effects of wavelet decomposition and neural networks on realized volatility prediction are attempted in [34] for crude oil, gold, and S&P500 assets. Křehlik [34] use wavelet decomposition for realized volatility estimation. In the paper, the volatility forecasting follows realized volatility estimation. Wavelet-based realized volatility estimation and ANN modeling based volatility forecasting are compared to classical estimators and autoregressive fractionally integrated moving average (ARFIMA) model-based volatility forecasting. According to the results, wavelet decomposition provides more knowledge than familiar estimators. On the other hand, the ANN gives better forecasting results than the ARFIMA method.

In [1] groundwater level forecasting is the research subject by considering wavelet-neural network (WA-ANN), regular ANN, and ARIMA techniques. The method that gives the best results is WA-ANN concerning  $R^2$ , the Nash-Sutcliffe model efficiency coefficient, and RMSE metrics.

Okkan investigates whether using the WNN positively affects monthly reservoir inflow prediction in [42]. DWT is employed to decompose monthly meteorological data. Having decomposed the data, inefficient subseries are removed, and the remaining subseries become inputs for neural networks. Levenberg-Marquardt (LM) optimization algorithm-based Feed Forward Neural Networks (FFNN), multiple linear regression (REG), and DWT-based multiple linear regression (WREG) are other models that are handled for empirical results. Results indicate that WNN outperforms FFNN, WREG, and REG. Additionally, using DWT increases the accuracy of both multiple linear regression and neural networks.

In [2] ARIMA, deep neural network (DNN), gated recurrent unit (GRU), and long short-term memory (LSTM) methods are used with and without DWT to predict high-frequency financial data. It is seen that DWT increases the accuracy of each technique. Moreover, GRU with DWT and LSTM with DWT give better results than other DWT based methods.

In [63] DWT-convolutional neural network (CNN)-LSTM model is utilized for day-ahead solar irradiance forecasting. DWT is used to decompose the raw solar irradiance data of specific weather types into detail and approximation parts. Subsequently, CNN is used to collect local features of decomposed data, and these gathered features are then used as inputs for LSTM for forecasting. The proposed method is checked against CNN-LSTM, ANN, manually extracted features-ANN, persistence forecasting, CNN, and LSTM models. It is concluded that the proposed method outperforms other forecasting models.

In [35] nonstationary wind power time series is decomposed by DWT. After decomposition, each subseries is modeled by a different LSTM model. Finally, all predicted values are put together to get the exact result. DWT-LSTM is compared with DWT-recurrent neural network (RNN), DWT-BP, LSTM, RNN, and BP. The proposed method gives the best accuracy results when compared to other methods.

Sugiartawan et al. [55] take advantage of a hybrid of wavelet transform and LSTM neural networks to predict tourist arrivals. Data is decomposed into subseries, and these subseries are used as inputs for the LSTM network. The recommended model is compared with hybrid wavelet-Elman, hybrid wavelet-Jordan, LSTM RNN, Elman RNN, and Jordan RNN.

WNN approach is tackled to learn a class of functions in [53]. The presented method uses wavelets as activation functions by using the translation and the dilation parameters.

ANN and WNN methods are used to model a solar air heater in [16]. Morlet wavelet is used as an activation function in WNN structure. According to the consequences, using WNN is more beneficial than utilizing ANN.

Pindoriya et al. select the Mexican hat wavelet as an activation function in FFNN to forecast short-term price in the electricity markets in [46]. After getting results for day-ahead prediction of several electricity markets, they verify that adaptive wavelet neural network (AWNN) gives better results than wavelet-ARIMA, MLP, radial basis function (RBF) neural networks and fuzzy neural network (FNN).

In [54] feedforward error backpropagation artificial neural network (FEBANN) and WNN based classifiers are compared to classify electromyography (EMG) signals. It is reported that the WNN method outperforms FEBANN according to the accuracy results of classifications.

In [4] MLP and WNN approaches are compared for calibration model building based on gasoline near-infrared (NIR) spectra. WNN is seen as more viable and robust than MLP.

It is seen that MRA or WNN issues are discussed in many studies individually. Fusing

these two issues with machine learning algorithms motivates our work.

### **1.3 Plan of the Thesis**

In Chapter 2 the basics of the neural networks are given. Besides, we clarify the fundamentals of LSTM and mention numerous literature reviews concerning the implementation of LSTM.

In Chapter 3 we present the essential technical points of wavelets. In particular technicalities of MODWT and MRA are explained.

We mention the substantial details of polynomial powers of sigmoid (PPS) in Chapter 4. Wavelets, which are used in the empirical study, are derived from PPS. The purpose of this chapter is to understand how we obtain wavelets to be used as activation functions.

Theory and implementation of LSTM and wavelets are applied to the financial data in Chapter 5. First, the flowchart of the source code is described. The computer features, system information, and error metrics we use for analyses are given. Afterward, descriptive statistics of S&P500 and NASDAQ financial time series are handled. Explanations, configurations, and results of models are also discussed in this chapter for the following methods: LSTM model without MRA, LSTM model with MRA, hybrid LSTM-Wavenet model without MRA, and hybrid LSTM-Wavenet model with MRA. Summary tables of both S&P500 and NASDAQ are given for the four different methods discussed throughout the paper.

Finally, we conclude the thesis in Chapter 6. Further studies and possible improvements are considered.

## CHAPTER 2

### NEURAL NETWORKS

The neural networks are founded on artificial learning, which is stimulated by the cerebrum of creatures. Some nodes have comparable undertakings with neurons in a cerebrum. Most frameworks are shaped by input, hidden, and output nodes inter-linked with past and next ones. A network structure includes the number of layers and the number of neurons in each layer, each layer's activation function, and the weights between adjacent nodes. The basic structure of multilayer neural network (MNN) is given in Figure 2.1.

As stated in [25], outputs of the multilayer feed forward neural network is given as

$$y_n^L = \left[ b_n^L + \sum_m f_m^{L-1} \left[ \cdots \left[ f_k^2 \left( \sum_j w_{jk}^2 \left[ f_j^1 \left( \sum_i w_{ij}^1 x_i + b_j^1 \right) \right] + b_k^2 \right) \right] \cdots \right] w_{mn}^L \right]_n, \quad (2.1)$$

where  $L$  is the total number of hidden layers plus one output layer ( $L = 0$  for the input layer and  $L$ th layer is the output layer),  $n$  is the index number of neurons in the output layer,  $i$  is the index number of neurons in the input layer,  $j$  is the index number of neurons in the first hidden layer,  $k$  is index number of neurons in the second hidden layer,  $m$  is the index number of the last hidden layer,  $y$ 's are output values,  $f^{L-1}$  is the  $(L - 1)$ th activation function,  $w_{mn}^L$ 's are weights between the  $L$ th and the  $(L - 1)$ th layers. Moreover,  $b_j^1, \dots, b_m^{L-1}$  are biases for hidden layers and  $b_n^L$  shows biases for the output layer. If we have only one hidden layer then the equation can simply be written as

$$y_k = b_k + \sum_j f_j \left( b_j + \sum_i w_{ij} x_i \right) w_{jk}, \quad (2.2)$$

where  $y$ 's are outputs,  $f$ 's are activation functions,  $i$  shows the index number of neurons in the input layer,  $j$  shows the index number of neurons in the hidden layers,  $k$  shows the index number of neurons in the output layer,  $b_k$  represents biases for the output layer,  $b_j$  indicates biases for the hidden layer.

Here weights are need to be tuned in to minimize difference between predicted and observed outputs. Backpropagation can be used for training to reduce error terms. Then we get

$$E = \frac{1}{2} \sum_{i=1} (y_d(i) - y_p(i))^2, \quad (2.3)$$

where  $i$  is the index for input data,  $y_d$  is original data and  $y_p$  is prediction. One needs to update weights in network in order to reach a small amount of total error.

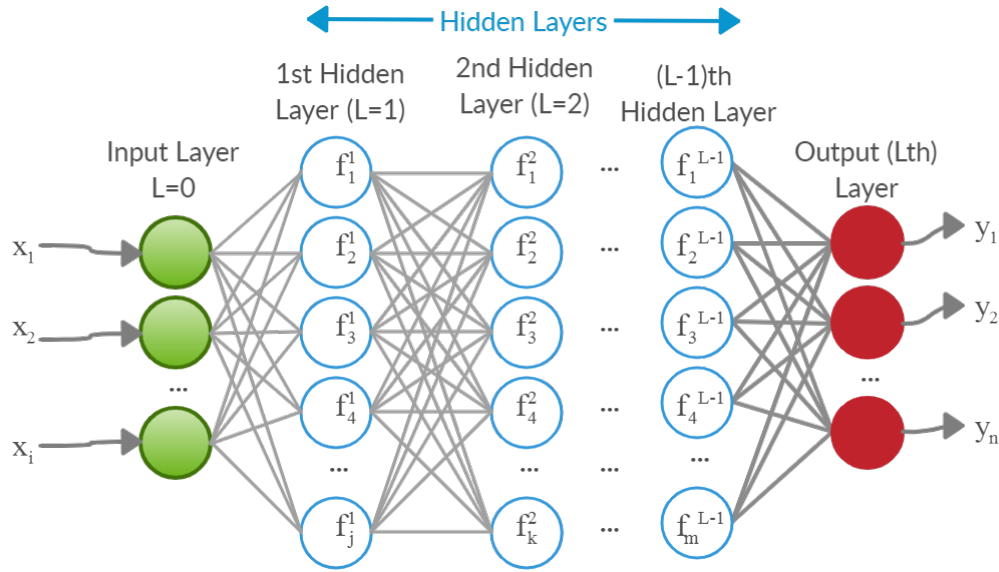


Figure 2.1: Structure of MNN.

There are several types of neural networks such as an artificial neural network (ANN), multilayer perceptron (MLP), convolutional neural network (CNN), recursive neural network (RvNN), recurrent neural network (RNN), gated recurrent unit (GRU) and long short-term memory (LSTM) which is a specific type of RNN, etc. [15]. Mainly, LSTM will be handled in the thesis for financial time series analysis since it is widely used for many time series types. Because financial data is complex along with several time intervals, the network structure needs to have memory of different time gaps. We refer to [26] for some advantages of LSTM.

## 2.1 Long Short-Term Memory (LSTM)

The difference between FFNN and RNN is adding extra weights for hidden layers as loops. Basic algorithm for RNN is written as

$$h_t = f_H (W_{IH}x_t + W_{HH}h_{t-1}), \quad (2.4)$$

$$y_t = f_O (W_{HO}h_t), \quad (2.5)$$

where  $x_t$  and  $y_t$  are input and output vectors,  $W_{IH}$ ,  $W_{HH}$  and  $W_{HO}$  are weight matrices,  $f_H$  and  $f_O$  are activation functions for hidden and output parts. There would be vanishing or exploding gradient problems in basic RNN. On the other hand, LSTM can solve these problems by adding extra parts like the input gate, the forget gate, and the output gate. Hence LSTM would be a better choice for time series modeling and prediction. Differences between ANN, RNN, and LSTM can be found in [61]. Details of LSTM structure and step by step explanation are given in [43]. In Figure 2.2 and in Figure 2.3 basic structure of a one-unit RNN and a one-unit LSTM are given respectively.



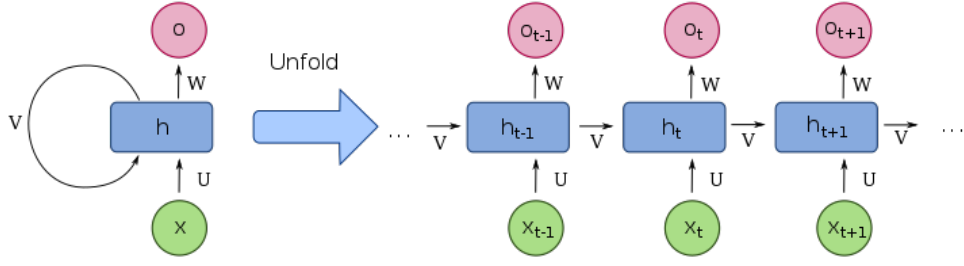


Figure 2.2: Structure of a Recurrent Neural Network System. ([https://commons.wikimedia.org/wiki/File:Recurrent\\_neural\\_network\\_unfold.svg](https://commons.wikimedia.org/wiki/File:Recurrent_neural_network_unfold.svg))

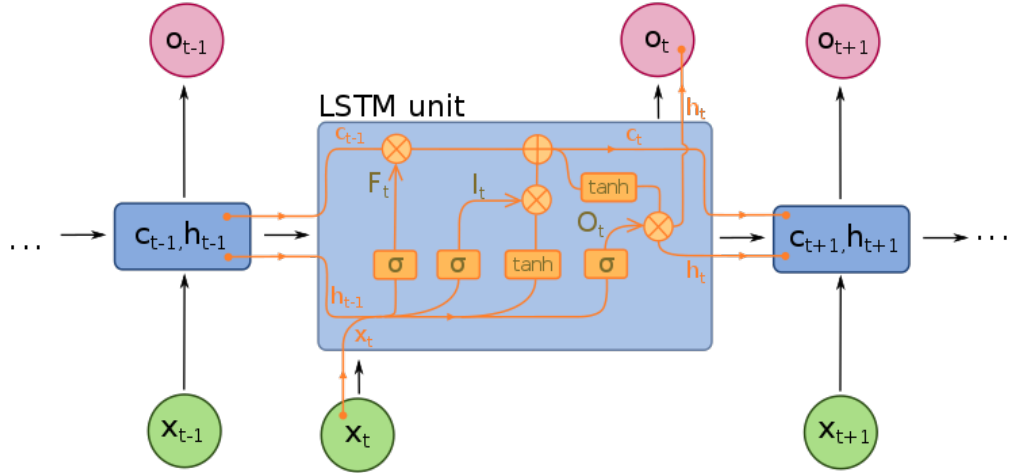


Figure 2.3: Structure of a Long Short-Term Memory. ([https://commons.wikimedia.org/wiki/File:Long\\_Short-Term\\_Memory.svg](https://commons.wikimedia.org/wiki/File:Long_Short-Term_Memory.svg))

In Figure 2.2,  $x$  is the input state,  $h$  is the hidden state and  $o$  is the output state. Moreover,  $U$ ,  $V$  and  $W$  are weights for the input, the recurrent, and the output parts.

Equations for calculating the value of the memory cell  $h$  at time  $t$  are given as

$$F_t = \sigma(W_F x_t + U_F h_{t-1} + b_F), \quad (2.6)$$

$$I_t = \sigma(W_I x_t + U_I h_{t-1} + b_I), \quad (2.7)$$

$$\tilde{C}_t = \tanh(W_C x_t + U_C h_{t-1} + b_C), \quad (2.8)$$

$$C_t = F_t * C_{t-1} + I_t * \tilde{C}_t, \quad (2.9)$$

$$O_t = \sigma(W_O x_t + U_O h_{t-1} + b_O), \quad (2.10)$$

$$h_t = O_t * \tanh(C_t), \quad (2.11)$$

where  $x_t$  is input vector to the LSTM at time  $t$ ,  $W$ 's and  $U$ 's are weight matrices,  $b$ 's are bias vectors,  $h_t$  is output vector of LSTM cell,  $C_t$  and  $\tilde{C}_t$  are state and candidate state vectors respectively,  $F_t$  is forget gate values,  $I_t$  is input gate values and  $O_t$  is output gate values. In (2.6)  $h_{t-1}$  and  $x_t$  are used to generate numbers between 0 and

1 for cell state  $C_{t-1}$ . If the number is 0 then information is completely forgotten. On the contrary, information is wholly preserved if the number is 1. Values between 0 and 1 point amount of information to be protected. In (2.8) new candidate state vector is created and later on combined by input layer gate values that are generated in (2.7). As a result we get updated cell state  $C_t$  in (2.9). Finally output of LSTM cell is calculated in (2.11) by using output gate values and updated state vector.

Many use the LSTM method for the time series analysis. In [10], Brownlee shows how to implement LSTM models for univariate, multivariate, and multi-step time series forecasting in Python. In [32], Kompella forecasts the sinusoidal time series using the LSTM structure with fixed window size. In [50], Schlosser mentions using the Keras library to implement LSTM to time series data sets in Python. Predictive techniques where LSTM is also included are handled in [24, 3]. Deep learning techniques are used to forecast sunspots data in [14] with the R programming. In [22] stock price movements are predicted by using LSTM in Python.

## CHAPTER 3

### WAVELETS

The Fourier transform has been used for a long time, and still, it is extensively used in many scientific and engineering areas. Unfortunately, it has several drawbacks, which are particularly found in the analysis of complex structured functions. Some of the disadvantages of Fourier transform are:

- The Fourier transform demands stationary data, i.e., there should not be any systematical changes in mean and variance of a time series. However, almost all financial time series and most time series in different disciplines are nonstationary.
- After taking the Fourier transform of a function, there will be no time information left.
- Modified version of the Fourier transform, which is called as short-time Fourier transform (STFT) or windowed Fourier transform, gives time information. But it still does not contain resolution information.

The wavelet transform was then intended to break into high-frequency and low-frequency parts of the signal in related time gaps. Wavelet transform is exceptionally appropriate for financial time series since financial time series have high-frequency segments for short time lengths and low-frequency parts for long time lengths.

#### 3.1 Maximal Overlap Discrete Wavelet Transform (MODWT)

Practically it is almost unattainable to use all wavelet coefficients to analyze the signal. One cannot compute almost an infinite number of coefficients by hand or by computer. In theory, it is more applicable to use continuous wavelet transform (CWT) for continuous functions but not for time series or discrete signals as Masset pointed out in [38]. As a result, it is better to work with sampled wavelets. To describe MRA, we use MODWT.

For DWT, the significant point is that one needs a time series with dyadic length to apply DWT. The reason for requiring a dyadic size is that coefficients of transform are computed by downsampling the original data as given in Mallat's Pyramid Algorithm [36] which can be seen in Figure 3.1 and Figure 3.2. One can analyze function

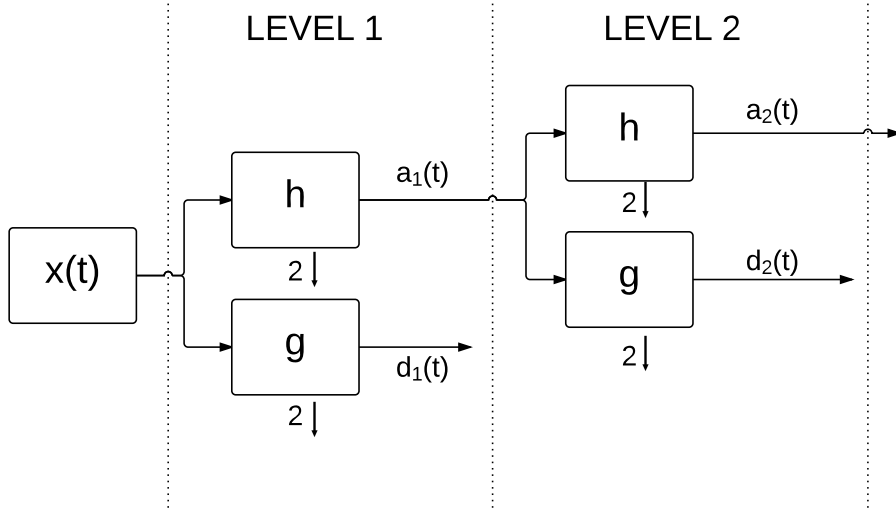


Figure 3.1: Flowchart of the pyramid algorithm for decomposing.

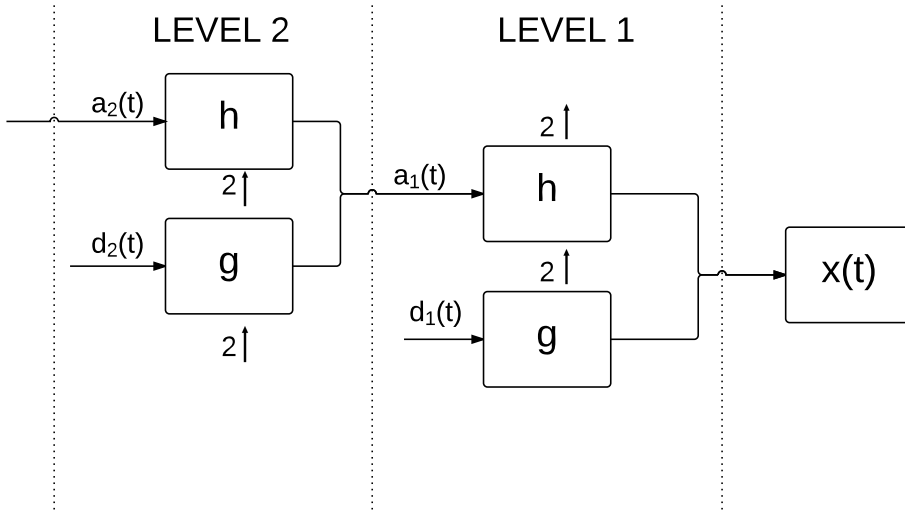


Figure 3.2: Flowchart of the pyramid algorithm for synthesis.

by using high-pass and low-pass filters to find smooth and detailed parts. The opposite process is called reconstruction or synthesis. Synthesis is used to reach the original signal by using smooth and detailed components by using upsampling.

The construction of MODWT is very similar to DWT. However, DWT has some drawbacks compared to MODWT. For instance, MODWT does not require dyadic length time series where DWT does. In other words, DWT limits the data for having a length of  $N = 2^J$  due to the upsampling and the downsampling. Since MODWT does not require dyadic length, the sizes of wavelet and scaling coefficients are equal to the original time series's length at every step of the transform. Moreover, MODWT is time-shift invariant while DWT is affected by time-shifting as stated in [23]. Further, the variance analysis of MODWT is more effective than the variance analysis of DWT as mentioned in [45].

Let  $\mathbf{w}$  contains wavelet and scaling coefficients of the MODWT,

$$\mathbf{w} = [w_1, w_2, \dots, w_J, v_J]^T, \quad (3.1)$$

where length of  $w_j$  is  $N/2^j$  and the length of  $v_j$  is  $N/2^j$  according to scale lengths as  $\lambda_j = 2^{j-1}$  and  $\lambda_j = 2^{J-1}$  respectively where  $j = 1, 2, \dots, J$ .

One can get the vector  $\mathbf{w}$  by using high-pass and low-pass filters as given in [23]. The vector that contains coefficients is given as

$$\tilde{\mathbf{w}} = \tilde{\mathbf{W}}x, \quad (3.2)$$

where  $\tilde{\mathbf{W}} = [\tilde{W}_1, \tilde{W}_2, \dots, \tilde{W}_J, \tilde{V}_J]^T$  is  $(J+1)N \times N$  matrix which implies that each  $\tilde{W}_j$  and  $\tilde{V}_J$  are  $N \times N$  matrices.

High-pass and low-pass filters are convolved with the time series to reach wavelet and scaling coefficients of the first level as following

$$w_1(t) = \sum_{l=0}^{L-1} h_l x(\dot{t}) \quad \text{and} \quad v_1(t) = \sum_{l=0}^{L-1} g_l x(\dot{t}), \quad (3.3)$$

where  $t = 0, 1, \dots, N-1$  and  $\dot{t} = t - l \pmod{N}$ . Wavelet and scaling coefficients of the second level is reached by convolving  $v_1(t)$  with the high-pass filter  $h_l$  and low-pass filter  $g_l$ . After  $J = \log_2 N$  iterations, wavelet and scaling coefficients are given as

$$w_J(t) = \sum_{l=0}^{L-1} h_l v_{J-1}(\dot{t}) \quad \text{and} \quad v_J(t) = \sum_{l=0}^{L-1} g_l v_{J-1}(\dot{t}), \quad (3.4)$$

where  $t = 0, 1, \dots, N-1$  and  $\dot{t} = t - 2^{J-1}l \pmod{N}$ . The vector which includes all coefficients is written as  $\mathbf{w} = [w_1 w_2 \dots w_J v_J]^T$ .

If the equations given in (3.3) and (3.4) are convolved with high-pass filter and low-pass filter respectively, the scaling coefficient of previous level is reached by summation of two convolving parts [23],

$$v_{J-1}(t) = \sum_{l=0}^{L-1} h_l w_J(\dot{t}) + \sum_{l=0}^{L-1} g_l v_J(\dot{t}), \quad (3.5)$$

where  $t = 0, 1, \dots, N-1$  and  $\dot{t} = t + l \pmod{N}$ . One can iterate this procedure up to the first level of wavelet and scaling coefficients to acquire the original time series as

$$x(t) = \sum_{l=0}^{L-1} h_l w_1(\dot{t}) + \sum_{l=0}^{L-1} g_l v_1(\dot{t}), \quad (3.6)$$

where  $t = 0, 1, \dots, N-1$  and  $\dot{t} = t + l \pmod{N}$ .

### 3.2 Multiresolution Analysis (MRA)

**Definition 3.1** (Multiresolution Analysis). After data is composed into vectors of coefficients which are related to the certain time scales. MRA is a sequence of closed nested subspaces  $\{V_j : j \in \mathbb{Z}\}$  in  $L_2(\mathbb{R})$  with the conditions given below, [21, 62, 60]

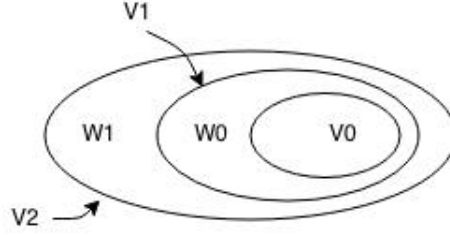


Figure 3.3: Nested Subspaces

1.  $\{\phi(x - k); k \in \mathbb{Z}\}$  is an orthonormal basis for  $V_0$  where  $\phi$  is scaling function of the MRA
2.  $\{0\} \subset \dots \subset V_j \subset V_{j+1} \subset \dots \subset L_2(\mathbb{R})$
3. Closure of  $(\cup_{j \in \mathbb{Z}} V_j) = L_2(\mathbb{R})$  such that any union is dense in  $L(\mathbb{R})$
4.  $\cap_{j \in \mathbb{Z}} V_j = \{0\}$  so zero element is the only common object
5.  $f(t) \in V_j \Leftrightarrow f(2t) \in V_{j+1}$  so that the spaces  $V$ 's are self-similar
6.  $V_{j+1} = V_j \oplus W_j$  where the space  $W_j$  is the  $j$ th resolution level of the MRA [62] and  $V_j \cap W_j = \{0\}$

Practically, construction of MRA is started from subspace  $V_0$  and larger subspaces are added to  $V_0$  in order to enlarge space. In Figure 3.3  $W_j$ 's are differences between  $V_{j+1}$  and  $V_j$  subspaces. Scaling and wavelet functions form bases for  $V_j$  and  $W_j$  subspaces by using scaling and translation parameters respectively as follows,

$$V_j = \text{span} \{\varphi_{j,k}(x)\}, \quad (3.7)$$

$$W_j = \text{span} \{\psi_{j,k}(x)\}. \quad (3.8)$$

Relation between subspaces is given by *Refinement Equations* using scaling and wavelet functions:

$$\varphi(x) = \sum_n h_\varphi(n) 2^{1/2} \varphi(2x - n), \quad (3.9)$$

$$\psi(x) = \sum_n h_\psi(n) 2^{1/2} \varphi(2x - n). \quad (3.10)$$

For instance, assume that  $\varphi(x)$  stands for  $V_1$ , then  $\varphi(2x - n)$  stands for  $V_2$ . Similarly if  $\psi(x)$  stands for  $W_1$  then  $\psi(2x - n)$  stands for  $W_2$ .

Dilation and translation parameters are employed to cover the  $L_2(\mathbb{R})$  space. The main idea of the MRA is to obtain  $L_2(\mathbb{R})$  space or any function by combining smooth and detailed parts.

$V_{j+1}$  can be written from last property in Definition 3.1 as

$$V_{j+1} = V_0 \oplus (W_0 \oplus W_1 \oplus \dots \oplus W_j). \quad (3.11)$$

After construction of MRA by using subspaces in Figure 3.3,  $L^2(\mathbb{R})$  space or any function in it can be written as a direct sum of  $V_0$  and  $W_j$  subspaces as

$$L^2(\mathbb{R}) = V_0 \oplus W_0 \oplus W_1 \oplus W_2 \oplus \dots \oplus W_j \text{ for } j = 0, 1, 2, \dots \quad (3.12)$$

In general form, any function can be written in the form of wavelet series expansion as

$$f(x) = \sum_k a_{j_0}(k) \varphi_{j_0,k}(x) + \sum_{j=j_0}^{\infty} \sum_k d_j(k) \psi_{j,k}(x) \text{ for } j > j_0. \quad (3.13)$$

In (3.13) the first sum with scaling function shows a smooth part and the second sum shows detailed parts of the function. In other words, the first sum covers  $V_j$  subspace and the second part covers  $W_j$  subspaces. Coefficients in (3.13) can be found by the following integrals,

$$a_{j_0}(k) = \int f(x) \varphi_{j_0,k} dx \quad \text{and} \quad d_j(k) = \int f(x) \psi_{j,k} dx. \quad (3.14)$$

In practice, scale level  $J$  is chosen finite and then time series is written in terms of approximation and detail parts as

$$x(t) = \sum_k a_{J,k} \varphi_{J,k}(t) + \sum_{j=j_0}^J \sum_k d_{j,k} \psi_{j,k}(t) \text{ for } j = 1, 2, \dots, J. \quad (3.15)$$

After using Mallat's pyramid algorithm, MODWT separates time series into smooth and detailed parts as

$$x(t) = A_{J,k} + \sum_{j=j_0}^J D_{j,k}, \quad (3.16)$$

where the component  $A_{J,k}$  holds the average information (or trend) of the original data at the largest scale and is associated with the scaling coefficients. Components  $D_{j,k}$ 's, from the first scale to the last scale, are concerned with wavelet coefficients. They are implemented for collecting higher frequency information, i.e., they cover detail coefficients [38].

Smooth and detailed parts can be found by multiplying coefficients by the scaling function (father wavelet) and wavelet function (mother wavelet) or directly from  $\mathcal{W}_j^T \mathbf{W}_j$  where  $\mathbf{W}$  contains approximation and detail coefficients and  $\mathcal{W}^T$  contains scaling and wavelet functions which are assembled of low-pass and high-pass filters.





## CHAPTER 4

### WAVELET NEURAL NETWORK

Concepts of wavelets and neural networks are associated with taking advantage of both at the same time. Wavelet neural network is appropriate for complex data like financial time series especially. Nonlinear modeling methods like neural networks are more suitable than linear methods for that kind of data. Furthermore, wavelet transforms allow us to model the problem more accurately in both time and frequency spaces.

Actually, the structure of wavelet neural networks is quite similar to the design of neural networks. Common activation functions like logistic, hyperbolic tangent, rectified linear unit (ReLU), softmax, etc., or some custom linear/nonlinear activation functions are used in neural networks. The difference between them is that wavelets are used in the hidden layer neurons as activation functions for wavelet neural networks. Then the hidden layer neurons are called as wavelons [28, 7, 67]. Wavelons consist of two parameters, which are called the translation and the dilation. The single wavelon can be written as

$$\psi_{u,v}(x) = \psi\left(\frac{x-u}{v}\right), \quad (4.1)$$

where  $u$  is the translation (or the location) and  $v$  is the dilation (or the scale) parameters. The function  $\psi$  is derived from wavelets as we mentioned in Chapter 3.

#### 4.1 Wavenets

In the learning process, if the translation and the dilation parameters given in (4.1) are unalterable, then the structure is called as wavenet [59]. On the other hand, if these parameters are adjustable during the learning process, the form is referred to as the wavelet network. Wavenets are used in the thesis for empirical studies.

Parameters mentioned in (4.1) may differ for each node in the hidden layer, but each is fixed for the corresponding node. Before setting these parameters, one needs to initialize them according to the data. Details of initializing these parameters are given in Chapter 5.

In [37], Marar and Bordin mention that there are many restrictions in the conventional backpropagation algorithm. Low dimensionality, the tensor product of wavelets, parameter initialization, and having one-dimensional output are samples of limitations.

A family of polynomial wavelets generated from powers of sigmoid functions is employed to discharge some of these constraints. In other words, the wavelet functions  $\psi$ 's are derived by using polynomial powers of sigmoid.

## 4.2 Polynomial Powers of Sigmoid (PPS)

In [17] it is stated that a family of polynomial wavelets produced from powers of sigmoid helps create robust neural network structures, particularly wavelet neural networks. One can form polynomial types of wavelet functions by using consecutive powers of sigmoid functions [18] where (4.2) and (4.3) need to satisfy. If (4.2) holds, then it means that every function  $f(x)$  is square-integrable, i.e. in  $L^2(\mathbb{R})$ . On the other hand, if (4.3) holds, it means that the mother wavelet  $\psi(x)$  satisfies the admissibility condition:

$$\int_{\mathbb{R}} |f(x)|^2 dx < \infty, \quad (4.2)$$

$$\int_0^{\infty} \frac{|\mathfrak{C}(\omega)|^2}{\omega} d\omega < \infty, \quad (4.3)$$

where  $\omega$  is frequency and  $\mathfrak{C}$  is the Fourier transform of  $\psi(x)$ .

Consider the following sigmoid function given in below

$$f(x) = \frac{1}{1 + \exp(-\alpha x)}, \quad (4.4)$$

where  $\alpha$  is the smoothness constant and  $f : \mathbb{R} \rightarrow [0, 1]$ . Even though the function given in (4.4) does not satisfy conditions in (4.2) and (4.3), functions derived from combination of it's powers satisfy the conditions. In order to create wavelet functions family from the sigmoid function, first of all we define  $n$ th power of the sigmoid function and set of all powers of the sigmoid function as given in (4.5) and (4.6) respectively.

$$f^n(x) = \left( \frac{1}{1 + \exp(-\alpha x)} \right)^n, \quad (4.5)$$

where  $f^n : \mathbb{R} \rightarrow [0, 1]$ , and

$$\Gamma = \{f^0(x), f^1(x), f^2(x), \dots, f^n(x)\}. \quad (4.6)$$

For  $\alpha = 1$  the first derivative of the sigmoid function is given as

$$\frac{df(x)}{dx} = \frac{d\left(\frac{1}{1+\exp(-x)}\right)}{dx} = \frac{f(-x)}{(1 + \exp(-x))^2} = f(x) \frac{\exp(-x)}{1 + \exp(-x)}. \quad (4.7)$$

After adding and subtracting 1 from the numerator  $\exp(-x)$  one can rewrite the first derivative of the sigmoid as

$$\begin{aligned} \frac{df(x)}{dx} &= f(x) \frac{\exp(-x) - 1 + 1}{1 + \exp(-x)} = f(x) \left( \frac{1 + \exp(-x)}{1 + \exp(-x)} - \frac{1}{1 + \exp(-x)} \right) \\ &= f(x) (1 - f(x)). \end{aligned} \quad (4.8)$$

Now let's consider the following wavelet equation, which consists of three sigmoid functions:

$$\psi(x) = f(x+h) - 2f(x) + f(x-h), \quad (4.9)$$

where  $h \in \mathbb{Z}$  and  $h \geq 2$ . After dividing both sides of the equation (4.9) by  $h^2$ , approximated second derivative of the sigmoid function is given in (4.10). In both  $p \geq 1$  and  $p < 1$  cases, while the limit goes to zero obtained functions satisfy wavelet conditions. Details can be found in [19, 20].

$$f''(x) \approx \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}. \quad (4.10)$$

One can find the second derivative of the function by taking the limit of the function  $\psi(x)$  which is divided by  $h^2$  while  $h$  goes to zero as:

$$\frac{d^2 f(x)}{dx^2} = \lim_{h \rightarrow 0} \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}. \quad (4.11)$$

Moreover, by using (4.8) the second derivative of the sigmoid function can be written as follows:

$$\frac{d^2 f(x)}{dx^2} = f(x)(1-f(x))^2 + (-f(x)^2(1-f(x))). \quad (4.12)$$

As a result by using (4.8) and (4.12) the PPS-wavelet functions produced by the first and the second derivative of the sigmoid function are given in (4.13) and (4.14) respectively,

$$\psi_1(x) = -f^2(x) + f(x) \quad (4.13)$$

and

$$\psi_2(x) = 2f^3(x) - 3f^2(x) + f(x). \quad (4.14)$$

Describing the set  $\Theta = \{\psi_1(x), \psi_2(x), \dots, \psi_n(x)\}$  keeps the same procedure as deriving the derivatives of the Gaussian function in structure of polynomials which are the Hermite polynomials [49]. In (4.15), (4.16) and (4.17) 3rd, 4th and 5th polynomial types of wavelet functions by using consecutive powers of sigmoid functions are given respectively as follows [17]:

$$\psi_3(x) = -6f^4(x) + 12f^3(x) - 7f^2(x) + f(x), \quad (4.15)$$

$$\psi_4(x) = 24f^5(x) - 60f^4(x) + 50f^3(x) - 15f^2(x) + f(x), \quad (4.16)$$

$$\psi_5(x) = -120f^6(x) - 360f^5(x) - 390f^4(x) + 180f^3(x) - 31f^2(x) + f(x). \quad (4.17)$$

In general, the polynomial wavelet function generated by the  $n$ th derivative of the sigmoid function is given as (see [40])

$$\psi_n(x) = \sum_k^n \sum_{j=0}^k (-1)^j (j+1)^n \binom{k}{j} f^{k+1}(x), \quad (4.18)$$

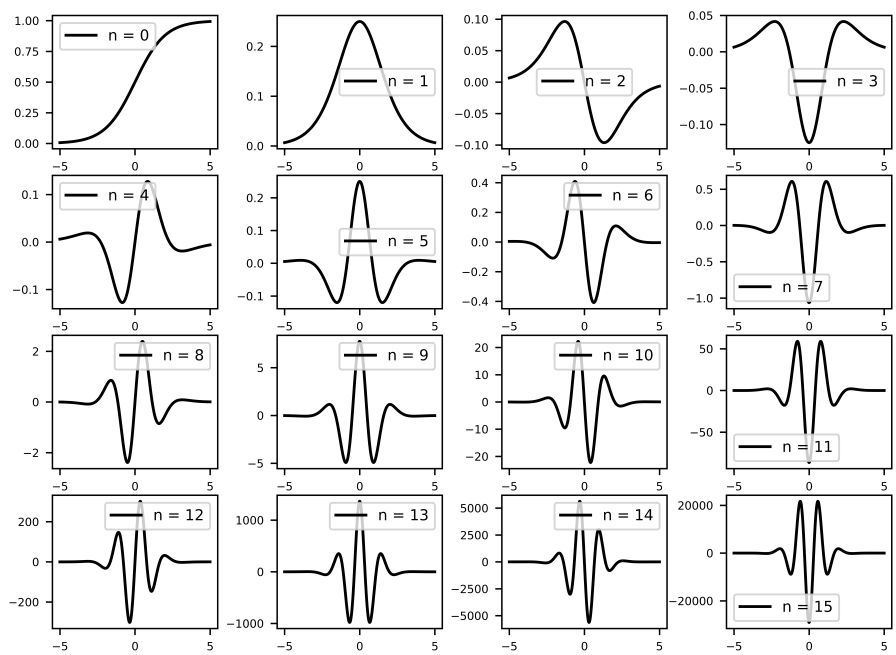


Figure 4.1: Polynomial Wavelet Functions Generated by the  $n$ th Derivative of the Sigmoid Function

where  $n$  is the derivative degree of the sigmoid function and  $f^{k+1}(x)$  is the  $(k + 1)$ th derivative of the sigmoid function. In Figure 4.1 each  $\psi_n(x)$  can be seen for  $n = 0, \dots, 15$  as in [39].

In [18] it is stated that observed family of polynomial wavelets satisfy that  $\psi_i \in L^2(\mathbb{R})$ . Consequently, the admissibility condition holds not only for this family but also for shifted and dilated versions of this family.



## CHAPTER 5

### EMPIRICAL RESULTS

In this chapter, we aim to show whether using MRA and hybrid LSTM-Wavenet model together or separately decreases the loss or not. First of all, in Section 5.2 we model financial time series data by using only LSTM neural network structure to predict the test set. Then in Section 5.3 the same data is modeled using the LSTM model with MRA. In Section 5.4, the wavenet structure is added to the LSTM (we call this structure a hybrid LSTM-Wavenet model), but MRA is not used. Finally, in Section 5.5 hybrid LSTM-Wavenet model with MRA is handled.

The flowchart of all the work done is given in Figure 5.1.

After taking the input (S&P500 and NASDAQ), there is a decision point that indicates whether MRA will be used or not. It is worth noting that in order to avoid confusion, that decision point is not an automated structure, but a manual action by the user. It is illustrated only to indicate whether the algorithm is running with or without an MRA. The general flow for both cases consists of data preprocessing, hyperparameter optimization, modeling & learning, prediction & visualization. Explanations of the flowchart are given in Table 5.1 and Table 5.2.

Initially, we download S&P500 (^GSPC) and NASDAQ (^IXIC) stock data between 1980-08-04 and 2019-08-02 from <https://finance.yahoo.com/>. Data has 9834 values for each “Open, High, Low, Close, Adj Close, Volume” column. Only closing prices are used. All analyses are performed using Python language in Anaconda. To perform ANN analysis, Keras library, which uses TensorFlow backend, is used [12].

For the sake of simplicity, we select the size of the data as 9830. This number corresponds to the dates between 1980-08-04 and 2019-07-29. Time series analysis is handled with/without using wavelet analysis. Talos library is used for hyperparameter optimizations [33]. After selecting the optimized hyperparameters, we also manually adjust the parameters by the hand-tuning<sup>1</sup> method. Batch normalization is used in all analyses between LSTM and dense layers to have smaller pieces of data with a mean of zero and a standard deviation of one. In [9, 13] it is stated that batch normalization has some advantages as listed below:

- (i) Reducing overfitting problems,
- (ii) Speeding up the training process,

---

<sup>1</sup> At this stage, kernel initializer, kernel regularizer, recurrent regularizer, and bias regularizer are optimized.

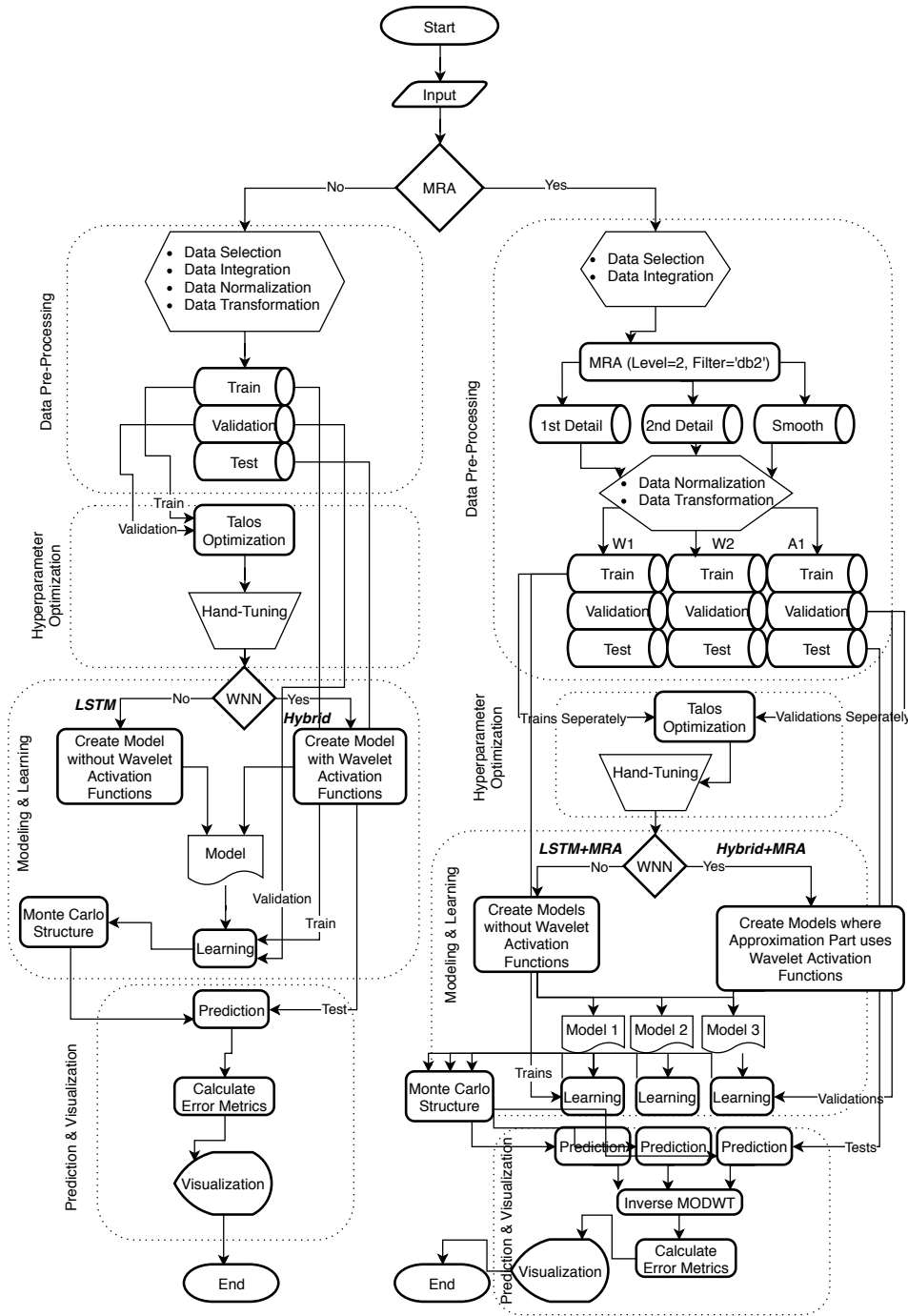


Figure 5.1: The Flowchart of Four Different Methods Applied in the Study



Table 5.1: Details of Data Preprocessing and Hyperparameter Optimization Parts Given in Figure 5.1 with and without MRA

	<b>MRA</b>	
	<i>without</i>	<i>with</i>
<b>Data Preprocessing</b>	In this step, the part of the data to be used is chosen. Then selected financial data is combined, and this combined data is normalized between 0 and 1. Subsequently, data is converted to the structure of a supervised learning type by using several time-steps <sup>a</sup> . Finally, the modified input is divided into the train set, the validation set, and the test set.	Difference from the data preprocessing part, that MRA is not used, is decomposing the data after selection and integration part. Data selection and data integration steps are made in the same way as given in without the MRA part. Each level of the decomposed data is normalized and transformed to produce pertinent data for a supervised learning problem. Later each level is divided into the train set, the validation set, and the test set.
<b>Hyperparameter Optimization</b>	The train set and the validation set are used in the Talos optimization process to get hyperparameters required to build a better model. Thereafter, automatically selected hyperparameters are again handled to be optimized manually.	Selection and optimization of hyperparameters are made for each created subseries by Talos optimization and hand-tuning methods, respectively.

<sup>a</sup> Different time step values are tested for the LSTM architecture applied to S&P500 data in [6]. The value 10 for the time step gives the best result. Therefore 10 is utilized as a time step in the thesis.

Table 5.2: Details of Modeling & Learning and Prediction & Visualization Parts Given in Figure 5.1 with and without MRA

	<b>MRA</b>	
	<i>without</i>	<i>with</i>
<b>Modeling &amp; Learning</b>	<p>In this step, there is another decision point for identifying whether WNN structure, specifically wavenets, will be used or not. If wavenets are not used, then the created model is called the LSTM model. On the other hand, if wavelet activation functions are used, the developed model is called a hybrid LSTM-Wavenet model. The formed model is applied to the train set and the validation set for the learning process.</p>	<p>In this step, the WNN decision point appears again. If wavelets are used as an activation function, then the model is referred to as a hybrid LSTM-Wavenet model with MRA. Otherwise, the model is denominated as an LSTM model with MRA. After the decision point, a model is created for each subseries. Each model is used for the learning process by using the train set and the validation set.</p>
<b>Prediction &amp; Visualization</b>	<p>A certain number of learning process calculations are done, and each learned model is used for the prediction process by using the test set. Moreover, the prediction process is also applied to the training set. Monte Carlo structure is used to get the mean error metric scores for the train set and the test set. Besides calculating different error metric results, visuals of analysis are obtained.</p>	<p>As in the other flow, the Monte Carlo structure is used here for the prediction process. After predicting train and test parts for each level, inverse MODWT is utilized to reconstruct the original time series. Finally, performance results for various error metrics and visuals of analysis are achieved.</p>

(iii) Increasing accuracy results/decreasing loss values.

Below is the list of the computing environment we work on.

- (i) OS Platform and Distribution: Windows 10
- (ii) Processor: Intel(R) Core(TM) i5-4210H CPU 2.90GHz
- (iii) Memory (RAM): 8,00 GB
- (iv) Conda version: 4.7.11
- (v) Conda-build version: 3.17.6
- (vi) Python version: 3.6.9
- (vii) Spyder version: 3.3.3
- (viii) TensorFlow version: 1.13.1
- (ix) Keras version: 2.2.4
- (x) Talos version: 0.5.0

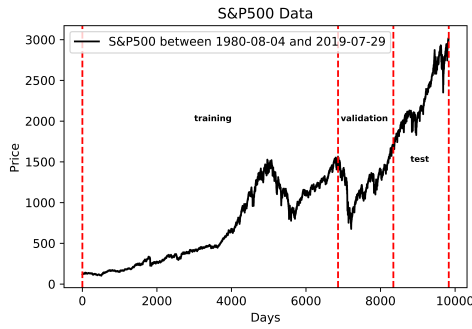
In order to measure the error terms, following metrics are used:

- (i) Root Mean Square Error (RMSE):  $\sqrt{\sum_{i=1}^n \frac{(y_i - \hat{y}_i)^2}{n}}$
- (ii) Scaled Root Mean Square Error (SRMSE):  $\text{RMSE} / (y_{\max} - y_{\min})$
- (iii)  $R^2$ :  $1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$
- (iv) Mean Absolute Error (MAE):  $\frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$
- (v) Explained Variance Score (EVS):  $1 - \frac{\text{Var}(y - \hat{y})}{\text{Var}(y)}$
- (vi) Maximum Error (ME):  $\max(|y_i - \hat{y}_i|)$
- (vii) Median Absolute Error (MdAE):  $\text{median}(|y_1 - \hat{y}_1|, \dots, |y_n - \hat{y}_n|)$

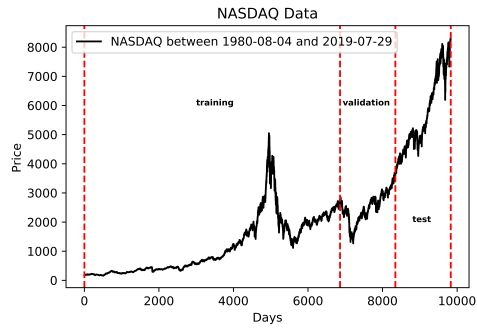
where  $y_i$  is the  $i$ th observed value,  $\hat{y}_i$  is the  $i$ th predicted value,  $y_{\max}$  is the maximum observation,  $y_{\min}$  is the minimum observation and  $\bar{y}$  is the mean of all observations.

## 5.1 Descriptive Statistics

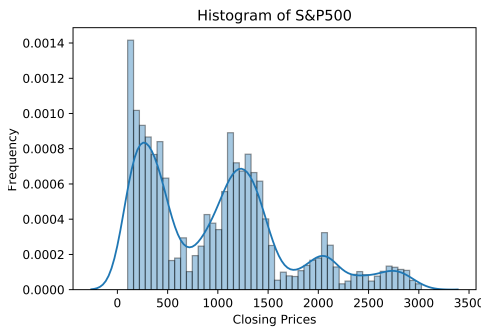
S&P500 is a USA based stock market index that includes 500 large companies listed on stock exchanges. S&P500 covers different sectors like finance, health care, industry, energy, information technology, and many others. On the other hand, NASDAQ



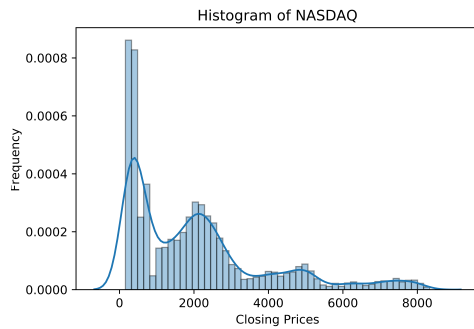
(a) Partitioning of S&P500



(b) Partitioning of NASDAQ



(c) Histogram of S&P500



(d) Histogram of NASDAQ

Figure 5.2: Partitioning Graphs and Histograms

Composite is also a USA based stock market index that covers the information technology sector. According to covering almost all sectors, S&P500 less volatile than NASDAQ does.

S&P500 includes also some stocks listed on the NASDAQ. Therefore the correlation between the two stock market indexes is expected to be high. As noted, Pearson's correlation coefficient is 0.9788, and Spearman's correlation coefficient is 0.9928.

For investors, NASDAQ is considered risky, while S&P500 is deemed to be risk-free. Many investors attach importance to portfolio diversification by investing in both riskless and risky markets.

This section observes descriptive statistics of the training set, the validation set, and the test set for both S&P500 and NASDAQ. Moreover, visuals of the observations and their histograms are given. Scores of the variance and the standard deviation are calculated for two different delta degrees of freedom (ddof) values. Firstly ddof is given as zero, and secondly, it is given as one. Histograms are plotted with a kernel density estimate (KDE) method, which provides visualizing the distribution of observations. Figure 5.2, part (a) and part (b), shows all S&P500 and NASDAQ data and their segmentation, respectively. Part (c) and part (d) present the histogram of S&P500 and NASDAQ with KDE one by one.

Descriptive statistics of S&P500 and its subsets are given in Table 5.3. According to skewness scores, the training data and the test data have more weight in the left tail of the distribution, while the validation data have more weight in the right tail

Table 5.3: Descriptive Statistics of S&amp;P500 Data

	All Data	Training Data	Validation Data	Test Data
Start Date	1980-08-04	1980-08-04	2007-10-10	2013-09-04
End Date	2019-07-29	2007-10-09	2013-09-03	2019-07-29
# of Observations	9830	6860	1485	1485
Minimum	102.42	102.42	676.53	1653.08
Maximum	3025.86	1565.15	1709.67	3025.86
Range	2923.44	1462.73	1033.14	1372.78
Mean	990.26	654.61	1247.78	2283.34
Median	993.52	459.59	1278.18	2143.16
Variance (ddof = 0)	511585.07	205914.77	46644.40	129766.45
Std (ddof = 0)	715.25	453.78	215.97	360.23
Variance (ddof = 1)	511637.11	205944.79	46675.83	129853.90
Std (ddof = 1)	715.29	453.81	216.05	360.35
Skewness	0.74	0.44	-0.24	0.37
Kurtosis	-0.10	-1.34	-0.38	-1.16

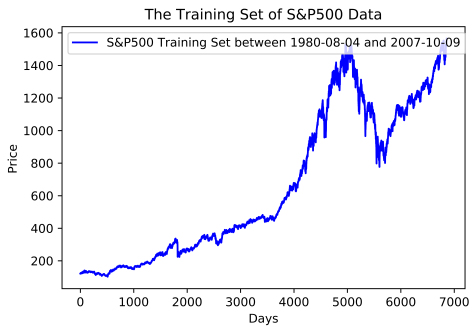
of the distribution. If we consider the kurtosis values, all the segmented sets have a distribution with lighter tails than the normal distribution has. In other words, all kurtosis types are platykurtic. In Figure 5.3 graphs and histograms with KDE of the training set, the validation set and the test set of S&P500 are given row by row, respectively.

In Table 5.3 descriptive statistics of NASDAQ data are given. Similar to S&P500, the training and the test sets are skewed left while the validation data is skewed right. The kurtosis type of the training set is leptokurtic, i.e., the curve has a higher peak than the normal curve. On the other hand, the kurtosis curves of the validation set and the test set have a flatter peak than the normal curve. In Figure 5.4 plots and histograms with KDE of the training set, the validation set, and the test set of NASDAQ are demonstrated.

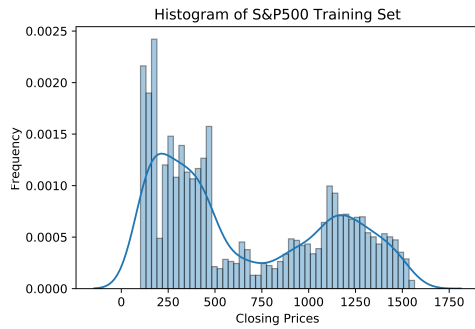
If we consider Figure 5.2a and Figure 5.2b, it is seen that there are large value changes between the training set, the validation set and the test set. In addition, there are fluctuations and several jumps in each separated sets. According to Table 5.3 and Table 5.4, minimum, maximum, mean, and standard deviation values are quite different for the training set, the validation set, and the test set. Each subset shows different characteristics for both S&P500 and NASDAQ in Figure 5.3 and in Figure 5.4 respectively. Due to these differences, it is expected that there are differences between the error metric results of the train set and the test set for both S&P500 and NASDAQ.

## 5.2 LSTM Model without MRA

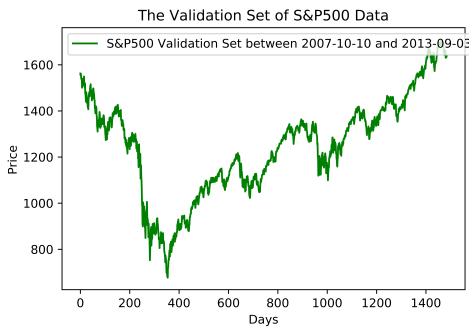
Closing prices of S&P500 and NASDAQ stocks are concatenated to get time series data to be used. After modeling the data, we predict closing prices of S&P500 and



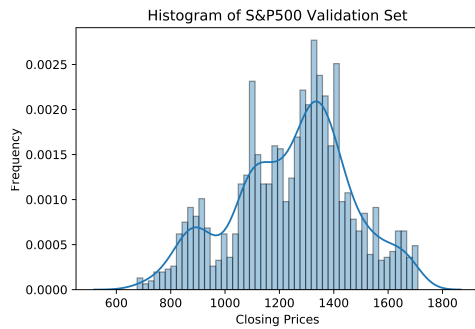
(a) The Training Set



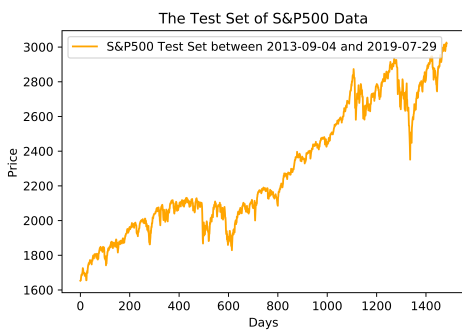
(b) Histogram of the Training Set



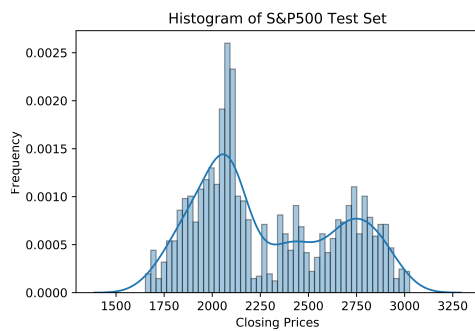
(c) The Validation Set



(d) Histogram of the Validation Set

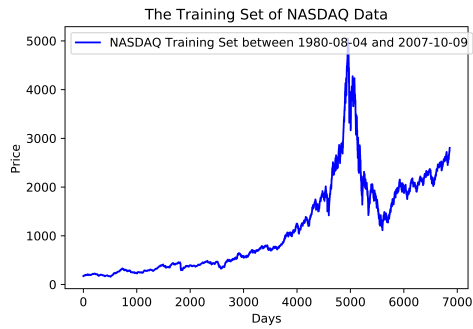


(e) The Test Set

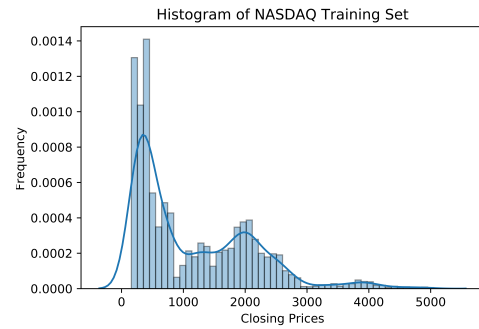


(f) Histogram of the Test Set

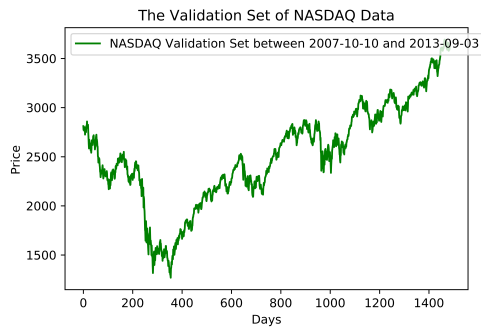
Figure 5.3: Visuals and Histograms of the Training Set, the Validation Set and the Test Set of S&P500



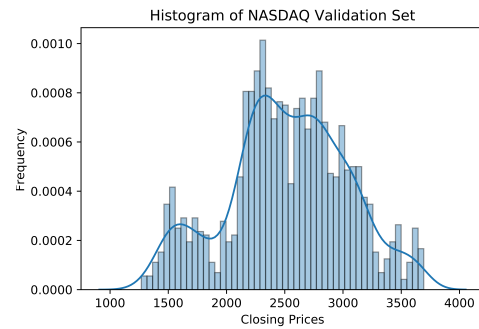
(a) The Training Set



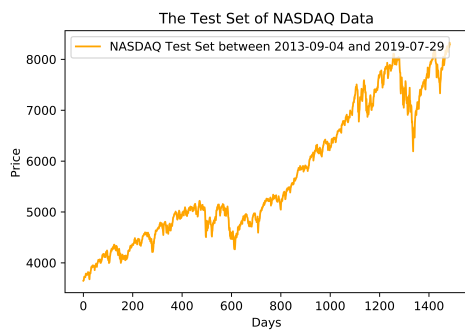
(b) Histogram of the Training Set



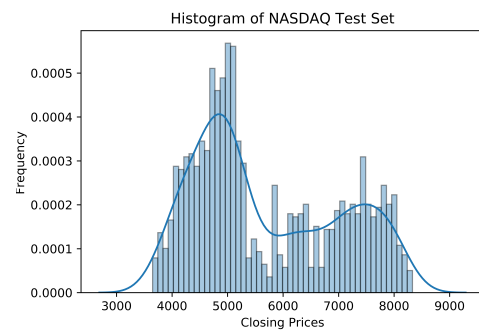
(c) The Validation Set



(d) Histogram of the Validation Set



(e) The Test Set



(f) Histogram of the Test Set

Figure 5.4: Visuals and Histograms of the Training Set, the Validation Set and the Test Set of NASDAQ

Table 5.4: Descriptive Statistics of NASDAQ Data

	All Data	Training Data	Validation Data	Test Data
Start Date	1980-08-04	1980-08-04	2007-10-10	2013-09-04
End Date	2019-07-29	2007-10-09	2013-09-03	2019-07-29
# of Observations	9830	6860	1485	1485
Minimum	159.14	159.14	1268.64	3649.04
Maximum	8330.21	5048.62	3692.95	8330.21
Range	8171.07	4889.48	2424.31	4681.17
Mean	2042.74	1147.35	2519.96	5701.76
Median	1697.72	743.45	2516.69	5190.10
Variance (ddof = 0)	3547004.08	909358.80	268338.30	1690667.54
Std (ddof = 0)	1883.35	953.60	518.01	1300.26
Variance (ddof = 1)	3547364.95	909491.37	268519.12	1691806.81
Std (ddof = 1)	1883.45	953.67	518.19	1300.69
Skewness	1.33	1.09	-0.14	0.44
Kurtosis	1.29	0.78	-0.34	-1.18

NASDAQ. Data (length of 9830) is split into the train, the validation, and the test sets. Almost %70 of all data is used in the train set (length of 6860), about %15 of all data is used in the validation set (length of 1485), and nearly %15 of all data is used in the test set (length of 1485). We prepare the dataset as two variables with ten time-steps. Each sample is related to index 0 (closing prices of S&P500) and index 1 (closing prices of NASDAQ) values, respectively. In other words, by using ten days window set with two variables, the next days' closing price is focused on being predicted. Two hidden layers are used, where the first one is formed of LSTM nodes and the second one consists of a regular densely-connected neural network. In the LSTM part, kernel, recurrent, and bias regularizers are put into practice. Batch normalization and ReLU activation function are added between dense and LSTM parts, respectively. In the dense part, kernel and bias regularizers are employed. After modeling the problem according to the selected configuration using Talos optimization and hand-tuning, we fit the model to data and predict train/test parts many times in a loop. Afterward, we calculate train/test predictions and the mean of error metrics.

We use six different configurations without MRA and wavenets in order to model the S&P500 and the NASDAQ datasets. In Table 5.5, Table 5.7, Table 5.9, Table 5.11, Table 5.13 and Table 5.15 six varying Talos configuration parameters which are used for the S&P500 time series analysis without wavelets can be seen. Only difference between these tables are the epoch sizes. Training and test results which are obtained using these structures are presented in Table 5.6, Table 5.8, Table 5.10, Table 5.12, Table 5.14 and Table 5.16 respectively. In Table 5.17 six distinct model results are given for S&P500. It is seen that results are affected by tuning  $L1$  regularization and  $L2$  regularization values in kernel, recurrent and bias regularizers for fixed epoch value.

Actually, from our point of view, the test scores are more important than the train scores.  $R^2$  and EVS error metrics show how well the model fits. RMSE, SRMSE,



Table 5.5: LSTM Model, Configuration 1 (S&P500): Talos Configuration

<b>Configuration 1</b>			
loss	mse		
optimizer	Adam		
time-steps	10		
batch size	1024		
epochs	100		
<i><u>Layer Parameters</u></i>	<u>LSTM Layer</u>	<u>Between Layers</u>	<u>Dense Layer</u>
# of nodes	16	-	128
kernel initializer	normal	-	-
batch normalization	-	yes	-
kernel regularizer	l1, l2=1e-4	-	l1, l2=1e-4
recurrent regularizer	l1, l2=1e-4	-	-
bias regularizer	l1, l2=1e-4	-	l1, l2=1e-4
activation	-	ReLU	ReLU

Table 5.6: LSTM Model, Configuration 1 (S&P500): Mean Scores for the Train set and the Test set by Running 1000 Experiments

<b>Configuration 1</b>		
experiment size	1000	
time taken by process	261m38s	
<i><u>Monte Carlo Scores</u></i>	<u>Reconstructed Train Scores</u>	<u>Reconstructed Test Scores</u>
RMSE	16.49	29.27
Scaled RMSE	0.01	0.01
$R^2$	1.00	0.99
MAE	13.35	23.53
EVS	1.00	1.00
ME	109.61	137.52
MdAE	11.78	20.20

Table 5.7: LSTM Model, Configuration 2 (S&P500): Talos Configuration

<b>Configuration 2</b>			
loss	mse		
optimizer	Adam		
time-steps	10		
batch size	1024		
epochs	60		
<i><u>Layer Parameters</u></i>	<u>LSTM Layer</u>	<u>Between Layers</u>	<u>Dense Layer</u>
# of nodes	16	-	128
kernel initializer	normal	-	-
batch normalization	-	yes	-
kernel regularizer	l1, l2=1e-4	-	l1, l2=1e-4
recurrent regularizer	l1, l2=1e-4	-	-
bias regularizer	l1, l2=1e-4	-	l1, l2=1e-4
activation	-	ReLU	ReLU

Table 5.8: LSTM Model, Configuration 2 (S&P500): Mean Scores for the Train set and the Test set by Running 1000 Experiments

<b>Configuration 2</b>		
experiment size	1000	
time taken by process	142m57s	
<i><u>Monte Carlo Scores</u></i>	<u>Reconstructed Train Scores</u>	<u>Reconstructed Test Scores</u>
RMSE	15.82	29.31
Scaled RMSE	0.01	0.01
$R^2$	1.00	0.99
MAE	12.75	23.61
EVS	1.00	1.00
ME	108.27	134.99
MdAE	11.10	20.27

Table 5.9: LSTM Model, Configuration 3 (S&P500): Talos Configuration

<b>Configuration 3</b>			
loss	mse		
optimizer	Adam		
time-steps	10		
batch size	1024		
epochs	100		
<i><u>Layer Parameters</u></i>	<u>LSTM Layer</u>	<u>Between Layers</u>	<u>Dense Layer</u>
# of nodes	16	-	128
kernel initializer	normal	-	-
batch normalization	-	yes	-
kernel regularizer	l1, l2=1e-6	-	l1, l2=1e-6
recurrent regularizer	l1, l2=1e-4	-	-
bias regularizer	l1, l2=1e-5	-	l1, l2=1e-5
activation	-	ReLU	ReLU

Table 5.10: LSTM Model, Configuration 3 (S&P500): Mean Scores for the Train set and the Test set by Running 1000 Experiments

<b>Configuration 3</b>		
experiment size	1000	
time taken by process	228m29s	
<i><u>Monte Carlo Scores</u></i>	<u>Reconstructed Train Scores</u>	<u>Reconstructed Test Scores</u>
RMSE	11.93	29.06
Scaled RMSE	0.00	0.01
$R^2$	1.00	0.99
MAE	8.06	23.30
EVS	1.00	1.00
ME	107.80	139.80
MdAE	4.86	19.93

Table 5.11: LSTM Model, Configuration 4 (S&P500): Talos Configuration

<b>Configuration 4</b>			
loss	mse		
optimizer	Adam		
time-steps	10		
batch size	1024		
epochs	60		
<i><u>Layer Parameters</u></i>	<u>LSTM Layer</u>	<u>Between Layers</u>	<u>Dense Layer</u>
# of nodes	16	-	128
kernel initializer	normal	-	-
batch normalization	-	yes	-
kernel regularizer	l1, l2=1e-6	-	l1, l2=1e-6
recurrent regularizer	l1, l2=1e-4	-	-
bias regularizer	l1, l2=1e-5	-	l1, l2=1e-5
activation	-	ReLU	ReLU

Table 5.12: LSTM Model, Configuration 4 (S&P500): Mean Scores for the Train set and the Test set by Running 1000 Experiments

<b>Configuration 4</b>		
experiment size	1000	
time taken by process	142m21s	
<i><u>Monte Carlo Scores</u></i>	<u>Reconstructed Train Scores</u>	<u>Reconstructed Test Scores</u>
RMSE	12.80	29.73
Scaled RMSE	0.00	0.01
$R^2$	1.00	0.99
MAE	8.76	24.00
EVS	1.00	1.00
ME	109.17	145.55
MdAE	5.20	20.77

Table 5.13: LSTM Model, Configuration 5 (S&P500): Talos Configuration

<b>Configuration 5</b>			
loss	mse		
optimizer	Adam		
time-steps	10		
batch size	1024		
epochs	100		
<i><u>Layer Parameters</u></i>	<u>LSTM Layer</u>	<u>Between Layers</u>	<u>Dense Layer</u>
# of nodes	16	-	128
kernel initializer	normal	-	-
batch normalization	-	yes	-
kernel regularizer	l1, l2=1e-5	-	l1, l2=1e-5
recurrent regularizer	l1, l2=1e-5	-	-
bias regularizer	l1, l2=1e-5	-	l1, l2=1e-5
activation	-	ReLU	ReLU

Table 5.14: LSTM Model, Configuration 5 (S&P500): Mean Scores for the Train set and the Test set by Running 1000 Experiments

<b>Configuration 5</b>		
experiment size	1000	
time taken by process	229m41s	
<i><u>Monte Carlo Scores</u></i>	<u>Reconstructed Train Scores</u>	<u>Reconstructed Test Scores</u>
RMSE	11.73	32.89
Scaled RMSE	0.00	0.01
$R^2$	1.00	0.99
MAE	8.66	26.82
EVS	1.00	1.00
ME	106.28	123.63
MdAE	6.79	22.79

Table 5.15: LSTM Model, Configuration 6 (S&P500): Talos Configuration

<b>Configuration 6</b>			
loss	mse		
optimizer	Adam		
time-steps	10		
batch size	1024		
epochs	60		
<i><u>Layer Parameters</u></i>	<u>LSTM Layer</u>	<u>Between Layers</u>	<u>Dense Layer</u>
# of nodes	16	-	128
kernel initializer	normal	-	-
batch normalization	-	yes	-
kernel regularizer	l1, l2=1e-5	-	l1, l2=1e-5
recurrent regularizer	l1, l2=1e-5	-	-
bias regularizer	l1, l2=1e-5	-	l1, l2=1e-5
activation	-	ReLU	ReLU

Table 5.16: LSTM Model, Configuration 6 (S&P500): Mean Scores for the Train set and the Test set by Running 1000 Experiments

<b>Configuration 6</b>		
experiment size	1000	
time taken by process	140m22s	
<i><u>Monte Carlo Scores</u></i>	<u>Reconstructed Train Scores</u>	<u>Reconstructed Test Scores</u>
RMSE	11.66	34.63
Scaled RMSE	0.00	0.01
$R^2$	1.00	0.99
MAE	8.52	28.39
EVS	1.00	1.00
ME	106.81	124.86
MdAE	6.52	24.13

Table 5.17: LSTM Model (S&P500): Scores of the all Configurations

	time	RMSE	SRMSE	$R^2$	MAE	EVS	ME	MdAE
<b>Configuration 1</b>	261m38s							
train scores		16.49	0.01	1.00	13.35	1.00	109.61	11.78
test scores		29.27	0.01	0.99	23.53	1.00	137.52	20.20
<b>Configuration 2</b>	142m57s							
train scores		15.82	0.01	1.00	12.75	1.00	108.27	11.10
test scores		29.31	0.01	0.99	23.61	1.00	134.99	20.27
<b>Configuration 3</b>	228m29s							
train scores		11.93	0.00	1.00	8.06	1.00	107.80	4.86
test scores		<b>29.06</b>	0.01	0.99	<b>23.30</b>	1.00	139.80	<b>19.93</b>
<b>Configuration 4</b>	142m21s							
train scores		12.80	0.00	1.00	8.76	1.00	109.17	5.20
test scores		29.73	0.01	0.99	24.00	1.00	145.55	20.77
<b>Configuration 5</b>	229m41s							
train scores		11.73	0.00	1.00	8.66	1.00	106.28	6.79
test scores		32.89	0.01	0.99	26.82	1.00	123.63	22.79
<b>Configuration 6</b>	140m22s							
train scores		11.66	0.00	1.00	8.52	1.00	106.81	6.52
test scores		34.63	0.01	0.99	28.39	1.00	124.86	24.13

and MAE metrics generate average errors by using residuals. Moreover, RMSE and SRMSE penalize large error values rather than other error metrics due to taking the square of residuals. RMSE and SRMSE are mostly used for model comparison. MdAE is quite robust against outliers. However, it is not an advantage in our case. Contrarily it causes a drawback since large errors occur at big jumps in financial time series, and the effect of these large errors, i.e., outliers, are significant. Lastly, ME metric does not matter because it gives a result of a single residual value.

Furthermore, almost all error metrics that we use generate robust error results. In other words, each model trial with thousand repetitions gives very similar results. However, it is seen that SRMSE,  $R^2$ , and EVS give identical good results for all the configurations. Therefore they are not useful for comparison in our situation. The best test scores occur in configuration 3 according to RMSE, MAE, and MdAE metrics as seen in Table 5.17.

We look at the output of configuration 3, which gives the best result from the six designs. In Figure 5.5, the model of the configuration 3 is given. Input and output sizes of each layer and modules between layers are seen. The input size of the LSTM and the output size of the last layer indicate that by looking at the previous ten days of the data, the next day is predicted.

In Figure 5.6a, configuration 3 loss values of all experiments are given for S&P500. It is an acceptable loss function graph since, after some epochs, validation loss falls under training loss, and both loss functions converge to the zero value. In Figure 5.6b and in Figure 5.6c, average values of predicted S&P500 train and test values with

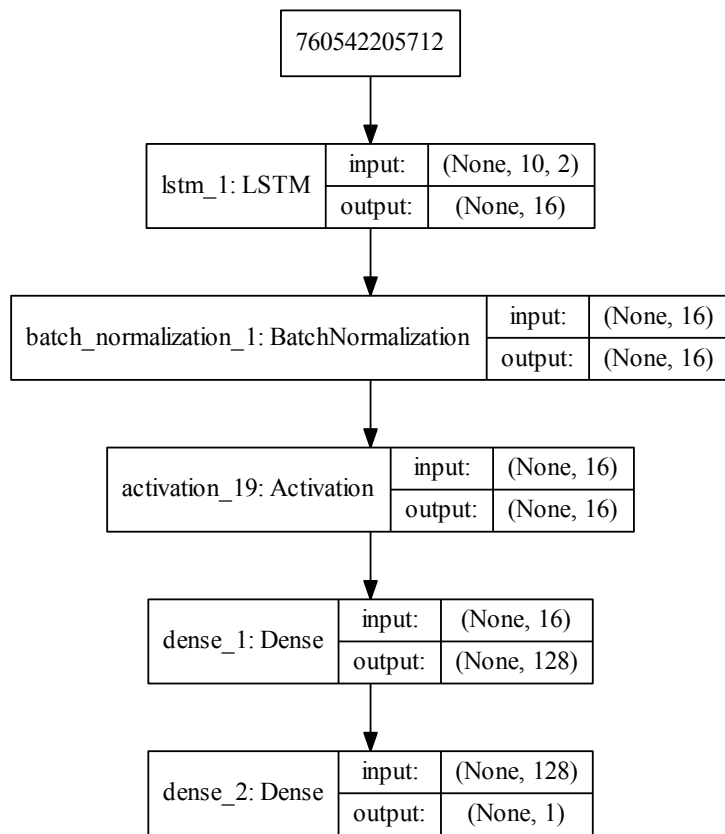
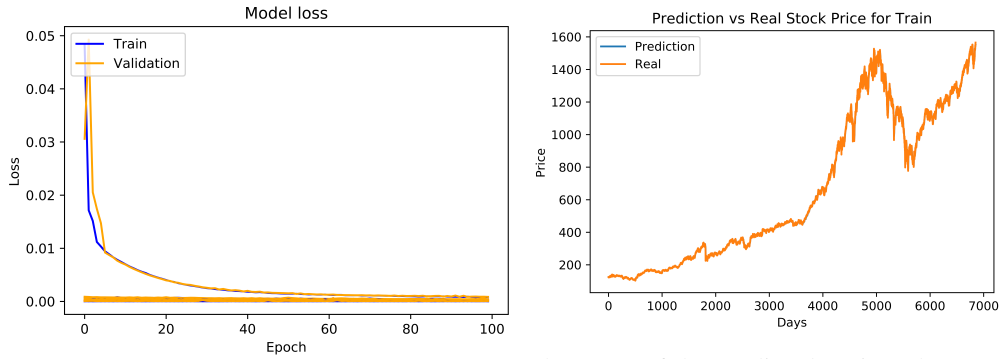


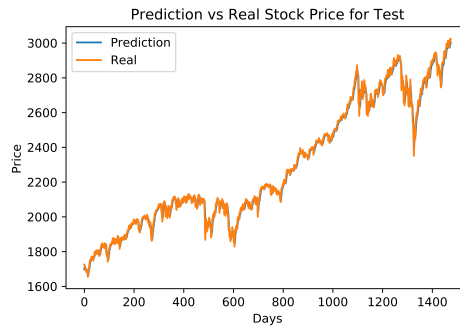
Figure 5.5: LSTM Model, Configuration 3 (S&P500): Model Structure





(a) Loss Values

(b) Mean of the Predicted Train Values vs. the Observed Train Values



(c) Mean of the Predicted Test Values vs. the Observed Test Values

Figure 5.6: LSTM Model, Configuration 3 (S&P500): Results

observed prices of the configuration 3 are given respectively. It is seen that both training and test predictions follow the observed values nicely.

Similarly in Table 5.18, Table 5.20, Table 5.22, Table 5.24, Table 5.26 and Table 5.28 six different Talos configurations for NASDAQ data are given. Again regularizer values are fixed and two different epoch values are used for consecutive binary configurations. Results for NASDAQ according to these configurations are given in Table 5.19, Table 5.21, Table 5.23, Table 5.25, Table 5.27 and Table 5.29 respectively. Summary table for NASDAQ data is given in Table 5.30. It is seen that the configuration 3 gives the best results for NASDAQ according to RMSE, MAE and MDAE metrics as we get for S&P500.

In Figure 5.7, the model structure of the configuration 3 is given for NASDAQ. As can be seen, the same structure utilized in the S&P500 data is used.

In Figure 5.8a, all of the loss values for each experiment are given for NASDAQ. Average values of predicted train and test values with observed prices for NASDAQ are given in Figure 5.8b and in Figure 5.8c respectively. Again both training and test predictions almost fit the observed values.

Table 5.18: LSTM Model, Configuration 1 (NASDAQ): Talos Configuration

<b>Configuration 1</b>			
loss	mse		
optimizer	Adam		
time-steps	10		
batch size	1024		
epochs	100		
<i><u>Layer Parameters</u></i>	<u>LSTM Layer</u>	<u>Between Layers</u>	<u>Dense Layer</u>
# of nodes	16	-	128
kernel initializer	normal	-	-
batch normalization	-	yes	-
kernel regularizer	l1, l2=1e-4	-	l1, l2=1e-4
recurrent regularizer	l1, l2=1e-4	-	-
bias regularizer	l1, l2=1e-4	-	l1, l2=1e-4
activation	-	ReLU	ReLU

Table 5.19: LSTM Model, Configuration 1 (NASDAQ): Mean Scores for the Train set and the Test set by Running 1000 Experiments

<b>Configuration 1</b>		
experiment size	1000	
time taken by process	286m52s	
<i><u>Monte Carlo Scores</u></i>	<u>Reconstructed Train Scores</u>	<u>Reconstructed Test Scores</u>
RMSE	45.32	95.58
Scaled RMSE	0.01	0.01
$R^2$	1.00	0.99
MAE	32.57	77.65
EVS	1.00	1.00
ME	498.44	374.15
MdAE	24.80	66.80

Table 5.20: LSTM Model, Configuration 2 (NASDAQ): Talos Configuration

<b>Configuration 2</b>			
loss	mse		
optimizer	Adam		
time-steps	10		
batch size	1024		
epochs	60		
<u>Layer Parameters</u>	<u>LSTM Layer</u>	<u>Between Layers</u>	<u>Dense Layer</u>
# of nodes	16	-	128
kernel initializer	normal	-	-
batch normalization	-	yes	-
kernel regularizer	l1, l2=1e-4	-	l1, l2=1e-4
recurrent regularizer	l1, l2=1e-4	-	-
bias regularizer	l1, l2=1e-4	-	l1, l2=1e-4
activation	-	ReLU	ReLU

Table 5.21: LSTM Model, Configuration 2 (NASDAQ): Mean Scores for the Train set and the Test set by Running 1000 Experiments

<b>Configuration 2</b>		
experiment size	1000	
time taken by process	143m16s	
<u>Monte Carlo Scores</u>	<u>Reconstructed Train Scores</u>	<u>Reconstructed Test Scores</u>
RMSE	43.13	92.46
Scaled RMSE	0.01	0.01
$R^2$	1.00	0.99
MAE	30.04	74.37
EVS	1.00	1.00
ME	497.08	367.99
MdAE	22.04	63.28

Table 5.22: LSTM Model, Configuration 3 (NASDAQ): Talos Configuration

<b>Configuration 3</b>			
loss	mse		
optimizer	Adam		
time-steps	10		
batch size	1024		
epochs	100		
<i><u>Layer Parameters</u></i>	<u>LSTM Layer</u>	<u>Between Layers</u>	<u>Dense Layer</u>
# of nodes	16	-	128
kernel initializer	normal	-	-
batch normalization	-	yes	-
kernel regularizer	l1, l2=1e-6	-	l1, l2=1e-6
recurrent regularizer	l1, l2=1e-4	-	-
bias regularizer	l1, l2=1e-5	-	l1, l2=1e-5
activation	-	ReLU	ReLU

Table 5.23: LSTM Model, Configuration 3 (NASDAQ): Mean Scores for the Train set and the Test set by Running 1000 Experiments

<b>Configuration 3</b>		
experiment size	1000	
time taken by process	234m49s	
<i><u>Monte Carlo Scores</u></i>	<u>Reconstructed Train Scores</u>	<u>Reconstructed Test Scores</u>
RMSE	36.65	85.31
Scaled RMSE	0.00	0.01
$R^2$	1.00	1.00
MAE	21.44	66.58
EVS	1.00	1.00
ME	492.62	324.16
MdAE	11.35	53.22

Table 5.24: LSTM Model, Configuration 4 (NASDAQ): Talos Configuration

<b>Configuration 4</b>			
loss	mse		
optimizer	Adam		
time-steps	10		
batch size	1024		
epochs	60		
<i><u>Layer Parameters</u></i>	<u>LSTM Layer</u>	<u>Between Layers</u>	<u>Dense Layer</u>
# of nodes	16	-	128
kernel initializer	normal	-	-
batch normalization	-	yes	-
kernel regularizer	l1, l2=1e-6	-	l1, l2=1e-6
recurrent regularizer	l1, l2=1e-4	-	-
bias regularizer	l1, l2=1e-5	-	l1, l2=1e-5
activation	-	ReLU	ReLU

Table 5.25: LSTM Model, Configuration 4 (NASDAQ): Mean Scores for the Train set and the Test set by Running 1000 Experiments

<b>Configuration 4</b>		
experiment size	1000	
time taken by process	142m22s	
<i><u>Monte Carlo Scores</u></i>	<u>Reconstructed Train Scores</u>	<u>Reconstructed Test Scores</u>
RMSE	37.27	86.72
Scaled RMSE	0.00	0.01
$R^2$	1.00	0.99
MAE	22.08	67.88
EVS	1.00	1.00
ME	495.08	326.28
MdAE	12.54	54.46

Table 5.26: LSTM Model, Configuration 5 (NASDAQ): Talos Configuration

<b>Configuration 5</b>			
loss	mse		
optimizer	Adam		
time-steps	10		
batch size	1024		
epochs	100		
<i><u>Layer Parameters</u></i>	<u>LSTM Layer</u>	<u>Between Layers</u>	<u>Dense Layer</u>
# of nodes	16	-	128
kernel initializer	normal	-	-
batch normalization	-	yes	-
kernel regularizer	l1, l2=1e-5	-	l1, l2=1e-5
recurrent regularizer	l1, l2=1e-5	-	-
bias regularizer	l1, l2=1e-5	-	l1, l2=1e-5
activation	-	ReLU	ReLU

Table 5.27: LSTM Model, Configuration 5 (NASDAQ): Mean Scores for the Train set and the Test set by Running 1000 Experiments

<b>Configuration 5</b>		
experiment size	1000	
time taken by process	231m15s	
<i><u>Monte Carlo Scores</u></i>	<u>Reconstructed Train Scores</u>	<u>Reconstructed Test Scores</u>
RMSE	42.98	87.34
Scaled RMSE	0.01	0.01
$R^2$	1.00	1.00
MAE	30.36	68.12
EVS	1.00	1.00
ME	499.32	328.67
MdAE	24.61	54.30

Table 5.28: LSTM Model, Configuration 6 (NASDAQ): Talos Configuration

<b>Configuration 6</b>			
loss	mse		
optimizer	Adam		
time-steps	10		
batch size	1024		
epochs	60		
<i><u>Layer Parameters</u></i>	<u>LSTM Layer</u>	<u>Between Layers</u>	<u>Dense Layer</u>
# of nodes	16	-	128
kernel initializer	normal	-	-
batch normalization	-	yes	-
kernel regularizer	l1, l2=1e-5	-	l1, l2=1e-5
recurrent regularizer	l1, l2=1e-5	-	-
bias regularizer	l1, l2=1e-5	-	l1, l2=1e-5
activation	-	ReLU	ReLU

Table 5.29: LSTM Model, Configuration 6 (NASDAQ): Mean Scores for the Train set and the Test set by Running 1000 Experiments

<b>Configuration 6</b>		
experiment size	1000	
time taken by process	143m38s	
<i><u>Monte Carlo Scores</u></i>	<u>Reconstructed Train Scores</u>	<u>Reconstructed Test Scores</u>
RMSE	39.57	87.49
Scaled RMSE	0.00	0.01
$R^2$	1.00	1.00
MAE	26.48	68.41
EVS	1.00	1.00
ME	498.17	327.93
MdAE	19.55	54.78

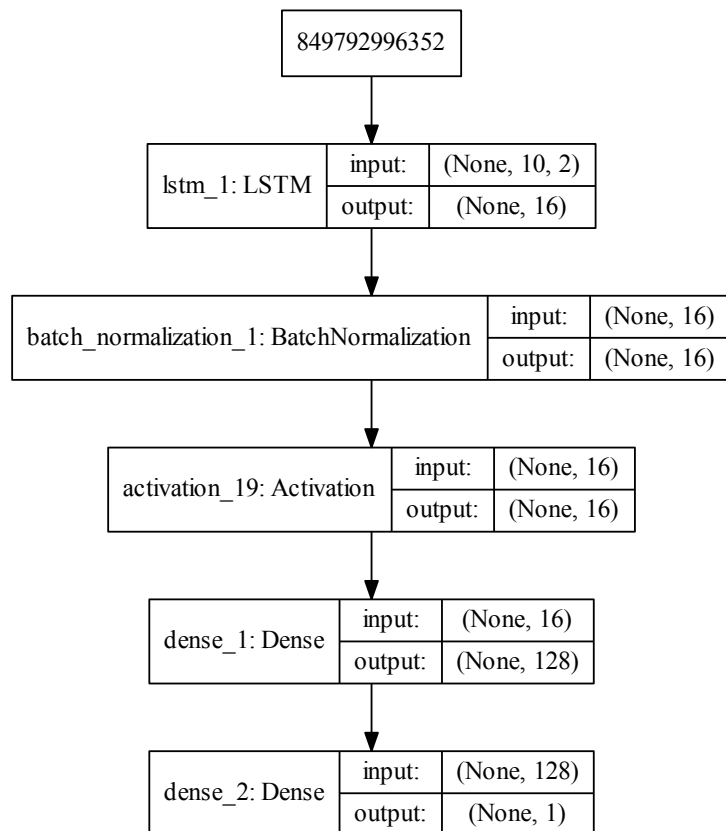
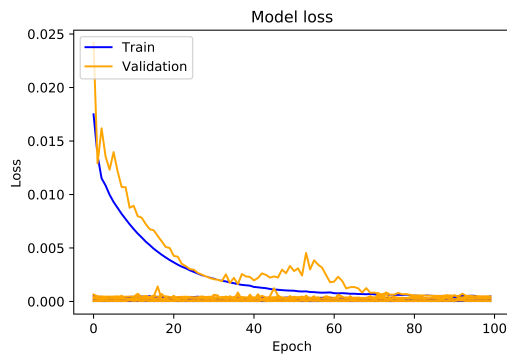
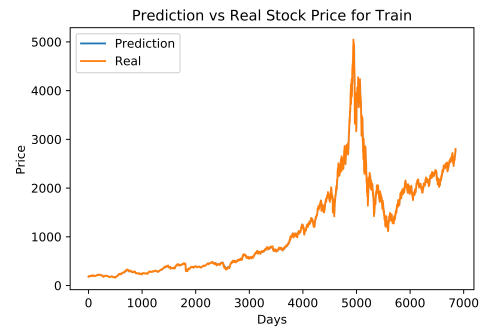


Figure 5.7: LSTM Model, Configuration 3 (NASDAQ): Model Structure

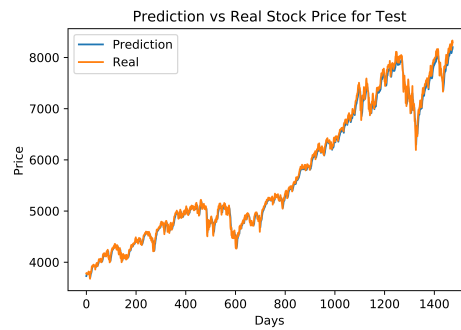




(a) Loss Values



(b) Mean of the Predicted Train Values vs. the Observed Train Values



(c) Mean of the Predicted Test Values vs. the Observed Test Values

Figure 5.8: LSTM Model, Configuration 3 (NASDAQ): Results

Table 5.30: LSTM Model (NASDAQ): Scores of the all Configurations

	time	RMSE	SRMSE	$R^2$	MAE	EVS	ME	MdAE
<b>Configuration 1</b>	286m52s							
train scores		45.32	0.01	1.00	32.57	1.00	498.44	24.80
test scores		95.58	0.01	0.99	77.65	1.00	374.15	66.80
<b>Configuration 2</b>	143m16s							
train scores		43.13	0.01	1.00	30.04	1.00	497.08	22.04
test scores		92.46	0.01	0.99	74.37	1.00	367.99	63.28
<b>Configuration 3</b>	234m49s							
train scores		36.65	0.00	1.00	21.44	1.00	492.62	11.35
test scores		<b>85.31</b>	0.01	1.00	<b>66.58</b>	1.00	324.16	<b>53.22</b>
<b>Configuration 4</b>	142m22s							
train scores		37.27	0.00	1.00	22.08	1.00	495.08	12.54
test scores		86.72	0.01	0.99	67.88	1.00	326.28	54.46
<b>Configuration 5</b>	231m15s							
train scores		42.98	0.01	1.00	30.36	1.00	499.32	24.61
test scores		87.34	0.01	1.00	68.12	1.00	328.67	54.30
<b>Configuration 6</b>	143m38s							
train scores		39.57	0.00	1.00	26.48	1.00	498.17	19.55
test scores		87.49	0.01	1.00	68.41	1.00	327.93	54.78

### 5.3 LSTM Model with MRA

In this section, we combine LSTM and MRA. First of all, data is decomposed into two detail parts and one approximation part by using MODWT. Daubechies wavelet, particularly the “db2=D4” filter and MODWT, are used to decompose the signal with a level of 2. After denormalizing predictions, each output part and prediction parts are combined by inverse MODWT separately for train and test parts. Each subseries (length of 9830) is split into train, validation, and test sets. Almost %70 of each level is used in the train set (length of 6860), about %15 of each level is used in the validation set (length of 1485), and nearly %15 of each level is used in the test set (length of 1485).

The same structure is used in the approximation level as we build in Section 5.2. On the other hand, one LSTM hidden layer is used for the first and the second detail levels separately. Each level is modeled by making use of Talos optimization and hand-tuning. Then similarly, we fit the model to data and predict train/test parts for 1000 experiments. Finally, we calculate the mean of error metrics and train/test predictions where we reached results by applying synthesis.

In Table 5.31 configurations of first detail, second detail and approximation are given for S&P500. Average values of Monte Carlo train RMSE and Monte Carlo test RMSE are given in Table 5.32. In Table 5.33 mean of 1000 reconstructed scores are presented for S&P500. It seems that using MRA significantly increases computation time. While using MRA appears to increase computation time significantly, it im-

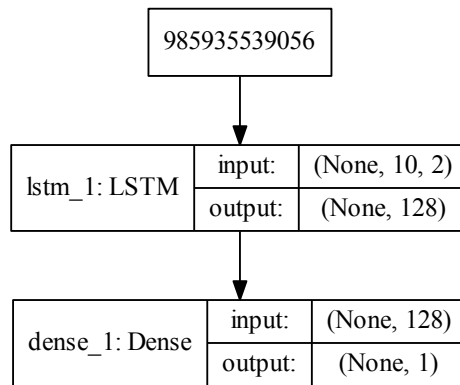


Figure 5.9: LSTM Model+MRA, Configuration 1 (S&P500): Model Structure of the First Detail

proves the prediction process.

Models of the first detail, the second detail and the approximation for S&P500 are given in Figure 5.9, Figure 5.10 and Figure 5.11 respectively.

Loss values of the first detail, the second detail and the approximation for S&P500 are given in Figure 5.12a, Figure 5.12b and Figure 5.12c respectively.

Predictions of the first detail, the second detail and the approximation parts for S&P500 are given in Figure 5.13. The first detail and the second detail levels are noticed more like noise modeling. For this reason, the models used in these levels are less complicated than the model used in the approximation level. It is seen that predictions of each part follow the observed values smoothly for all levels.

After synthesizing all wavelet levels for S&P500, average of 1000 predicted train and test graphs are given in Figure 5.14a and Figure 5.14b respectively.

Similarly, configurations of first detail, second detail and approximation are given for NASDAQ in Table 5.34. Mean values of train RMSE and test RMSE are given in Table 5.35. In Table 5.36 mean of 1000 reconstructed scores are given for NASDAQ. As with the S&P500 data, the use of MRA increases computation time but provides better prediction results for NASDAQ.

Models of the first detail, the second detail and the approximation for NASDAQ are given in Figure 5.15, Figure 5.16 and Figure 5.17 respectively.

Loss values of the first detail, the second detail and the approximation for NASDAQ are given in Figure 5.18a, Figure 5.18b and Figure 5.18c respectively.

Predictions of the first detail, the second detail and the approximation parts for NASDAQ are given in Figure 5.19. Again predictions fit observed values pretty well.

Average predictions of 1000 reconstructed train and test graphs for NASDAQ are

Table 5.31: LSTM Model+MRA, Configuration 1 (S&P500): Talos Configuration

<b>Configuration 1</b>							
wavelet filter = db2 (D4)							
wavelet level = 2							
<i>Detail 1</i>		<i>Detail 2</i>		<i>Approximation</i>			
loss	mse	loss	mse	loss	mse		
optimizer	Adam	optimizer	Adam	optimizer	Adam		
time-steps	10	time-steps	10	time-steps	10		
batch size	1024	batch size	1024	batch size	1024		
epochs	50	epochs	50	epochs	100		
<i>Layer Parameters</i>		<i>Layer Parameters</i>		<i>Layer Parameters</i>			
# of nodes	LSTM Layer 128	# of nodes	LSTM Layer 128	# of nodes	LSTM Layer 16	<i>Between Layers</i>	<i>Dense Layer</i> 128
kernel initializer	-	kernel initializer	-	kernel initializer	normal	-	-
batch normalization	-	batch normalization	-	batch normalization	-	yes	-
kernel regularizer	-	kernel regularizer	-	kernel regularizer	11, 12=1e-6	-	11, 12=1e-6
recurrent regularizer	-	recurrent regularizer	-	recurrent regularizer	11, 12=1e-4	-	-
bias regularizer	-	bias regularizer	-	bias regularizer	11, 12=1e-5	-	11, 12=1e-5
activation	-	activation	-	activation	-	ReLU	ReLU

Table 5.32: LSTM Model+MRA, Configuration 1 (S&P500): Mean Scores of the Wavelet Levels for the Train set and the Test set by Running 1000 Experiments

<b>Configuration 1</b>			
experiment size	1000	wavelet filter	db2 (D4)
time taken by process	1981m47s	wavelet level	2
	<b>Monte Carlo Train RMSE Scores</b>	<b>Monte Carlo Test RMSE Scores</b>	
<b>Detail 1</b>	3.63	7.52	
<b>Detail 2</b>	2.31	4.48	
<b>Approximation</b>	15.90	26.49	

Table 5.33: LSTM Model+MRA, Configuration 1 (S&P500): Mean Scores for the Synthesized Train set and the Synthesized Test set by Running 1000 Experiments

<b>Configuration 1</b>			
experiment size	1000	wavelet filter	db2 (D4)
time taken by process	1981m47s	wavelet level	2
<b>Monte Carlo Scores</b>	<b>Reconstructed Train Scores</b>	<b>Reconstructed Test Scores</b>	
RMSE	15.29	26.91	
Scaled RMSE	0.01	0.01	
$R^2$	1.00	0.99	
MAE	12.29	22.66	
EVS	1.00	1.00	
ME	239.30	104.68	
MdAE	11.11	20.36	

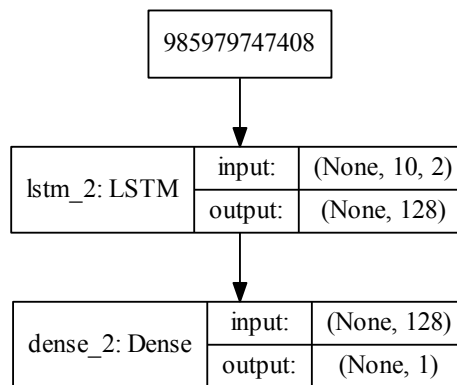


Figure 5.10: LSTM Model+MRA, Configuration 1 (S&P500): Model Structure of the Second Detail

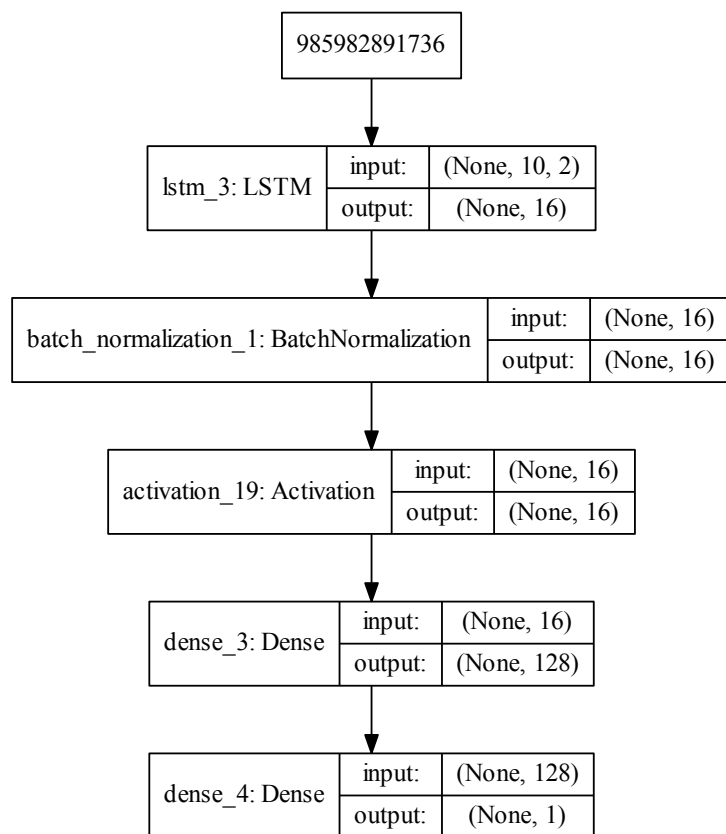
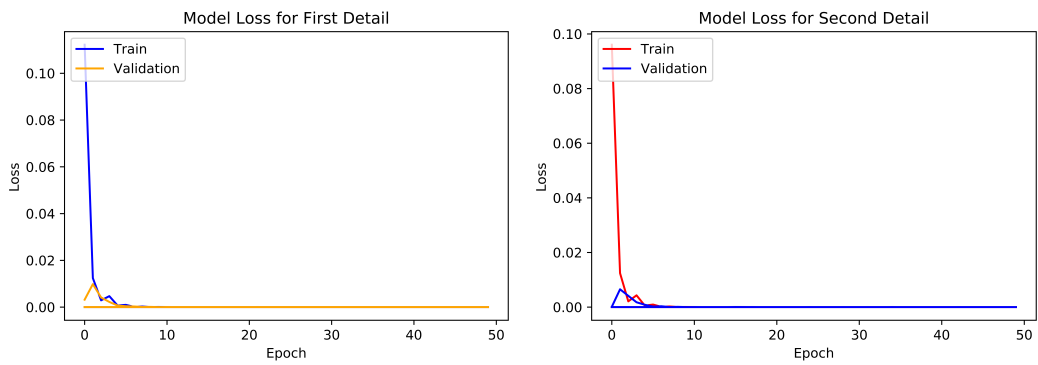
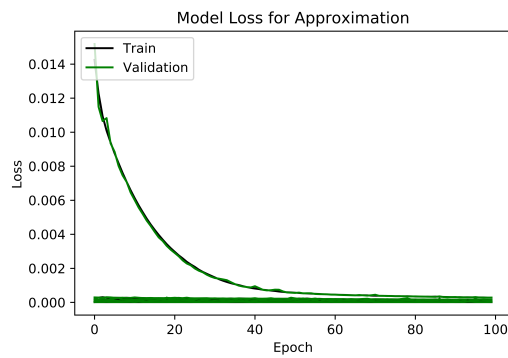


Figure 5.11: LSTM Model+MRA, Configuration 1 (S&P500): Model Structure of the Approximation



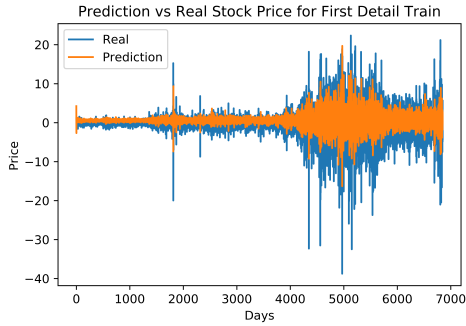
(a) Loss Values of the First Detail

(b) Loss Values of the Second Detail

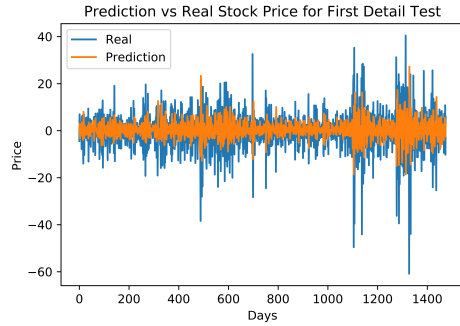


(c) Loss Values of the Approximation

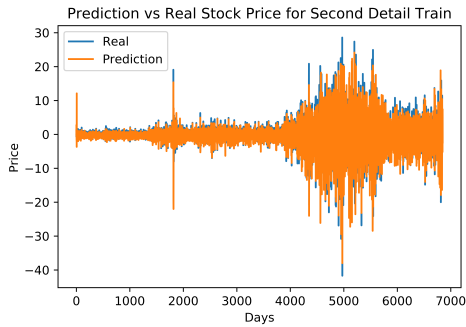
Figure 5.12: LSTM Model+MRA, Configuration 1 (S&P500): Loss Values of Each Level



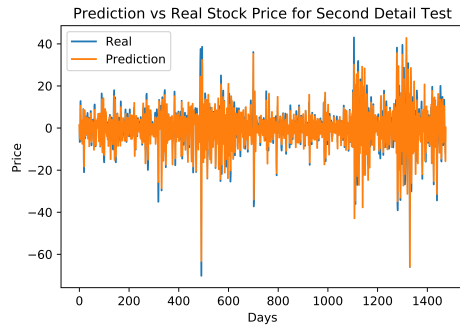
(a) 1st Detail-Train Prediction



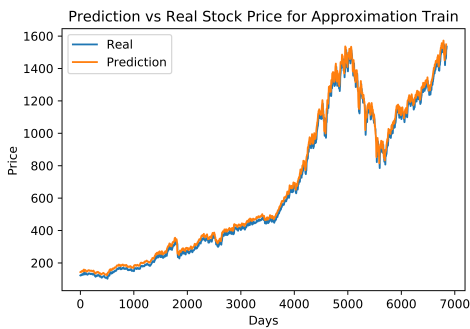
(b) 1st Detail-Test Prediction



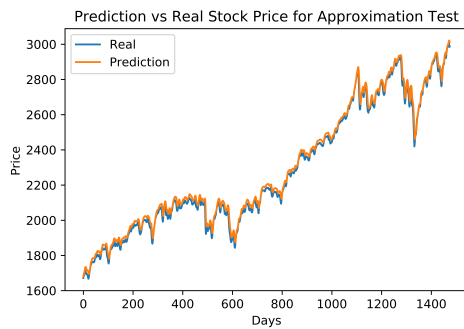
(c) 2nd Detail-Train Prediction



(d) 2nd Detail-Test Prediction



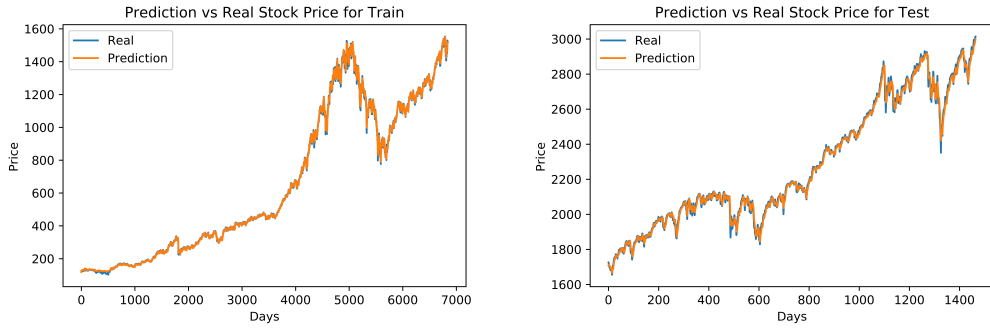
(e) Approximation-Train Prediction



(f) Approximation-Test Prediction

Figure 5.13: LSTM Model+MRA, Configuration 1 (S&P500): Train and Test Predictions of the First Detail, the Second Detail and the Approximation Parts





(a) Mean of the Predicted Train Values vs. Observed Train Values for Reconstructed Data (b) Mean of the Predicted Test Values vs. Observed Test Values for Reconstructed Data

Figure 5.14: LSTM Model+MRA, Configuration 1 (S&P500): Reconstructed Results

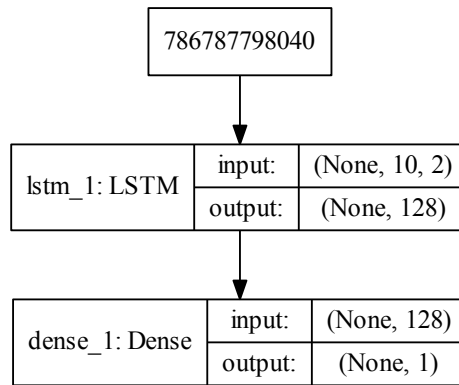


Figure 5.15: LSTM Model+MRA, Configuration 1 (NASDAQ): Model Structure of the First Detail

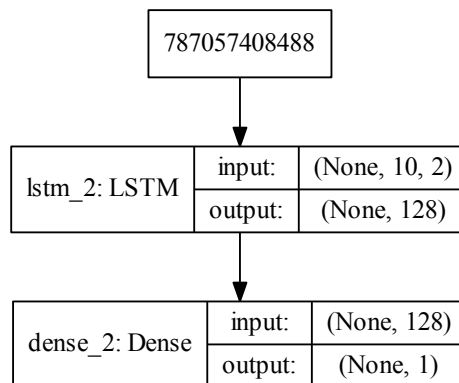


Figure 5.16: LSTM Model+MRA, Configuration 1 (NASDAQ): Model Structure of the Second Detail

Table 5.34: LSTM Model+MRA, Configuration 1 (NASDAQ): Talos Configuration

<b>Configuration 1</b>							
wavelet filter = db2 (D4)							
wavelet level = 2							
<i>Detail 1</i>		<i>Detail 2</i>		<i>Approximation</i>			
loss	mse	loss	mse	loss	mse		
optimizer	Adam	optimizer	Adam	optimizer	Adam		
time-steps	10	time-steps	10	time-steps	10		
batch size	1024	batch size	1024	batch size	1024		
epochs	50	epochs	50	epochs	100		
<u>Layer Parameters</u>	LSTM Layer	<u>Layer Parameters</u>	LSTM Layer	<u>Layer Parameters</u>	LSTM Layer		
# of nodes	128	# of nodes	128	# of nodes	16	<u>Between Layers</u>	<u>Dense Layer</u>
kernel initializer	-	kernel initializer	-	kernel initializer	normal	-	-
batch normalization	-	batch normalization	-	batch normalization	-	yes	-
kernel regularizer	-	kernel regularizer	-	kernel regularizer	11, 12=1e-6	-	11, 12=1e-6
recurrent regularizer	-	recurrent regularizer	-	recurrent regularizer	11, 12=1e-4	-	-
bias regularizer	-	bias regularizer	-	bias regularizer	11, 12=1e-5	-	11, 12=1e-5
activation	-	activation	-	activation	-	ReLU	ReLU

Table 5.35: LSTM Model+MRA, Configuration 1 (NASDAQ): Mean Scores of the Wavelet Levels for the Train set and the Test set by Running 1000 Experiments

<b>Configuration 1</b>			
experiment size	1000	wavelet filter	db2 (D4)
time taken by process	2008m34s	wavelet level	2
	<b><i>Monte Carlo Train RMSE Scores</i></b>	<b><i>Monte Carlo Test RMSE Scores</i></b>	
<b><i>Detail 1</i></b>	10.03	21.60	
<b><i>Detail 2</i></b>	4.83	12.66	
<b><i>Approximation</i></b>	35.26	64.93	

Table 5.36: LSTM Model+MRA, Configuration 1 (NASDAQ): Mean Scores for the Synthesized Train set and the Synthesized Test set by Running 1000 Experiments

<b>Configuration 1</b>			
experiment size	1000	wavelet filter	db2 (D4)
time taken by process	2008m34s	wavelet level	2
<b><i>Monte Carlo Scores</i></b>	<b>Reconstructed Train Scores</b>	<b>Reconstructed Test Scores</b>	
RMSE	33.12	65.41	
Scaled RMSE	0.00	0.01	
$R^2$	1.00	1.00	
MAE	20.74	52.67	
EVS	1.00	1.00	
ME	612.95	244.27	
MdAE	11.67	44.05	

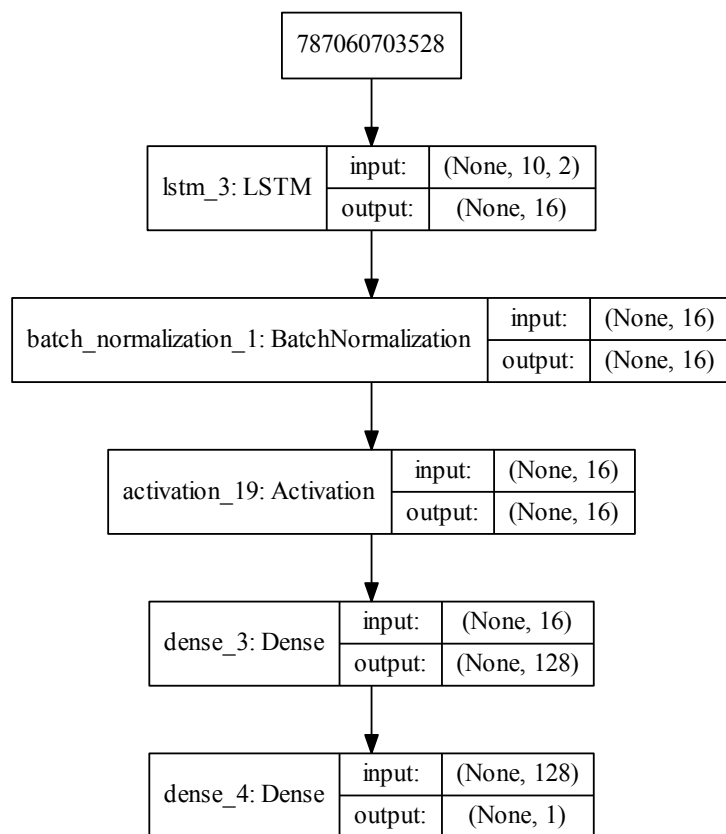
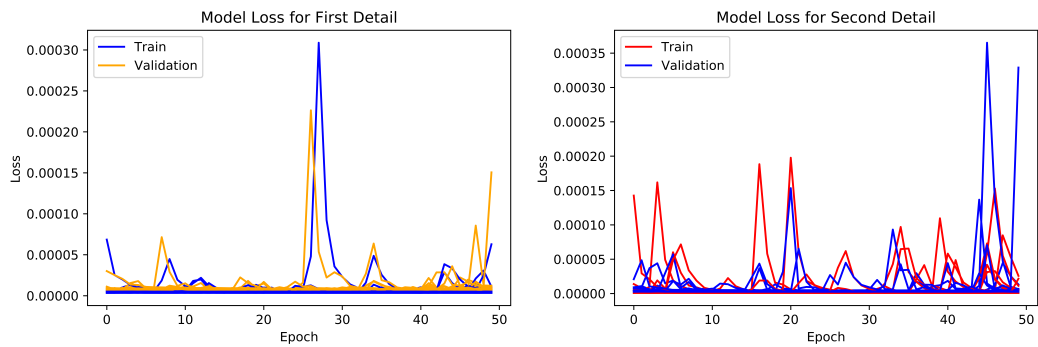
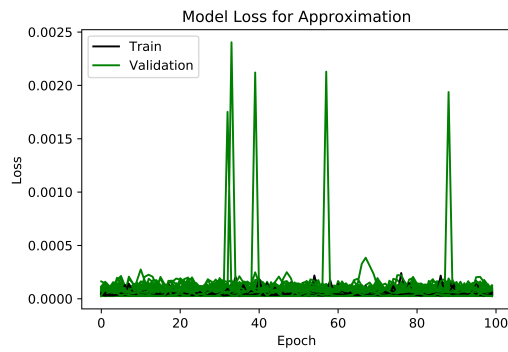


Figure 5.17: LSTM Model+MRA, Configuration 1 (NASDAQ): Model Structure of the Approximation



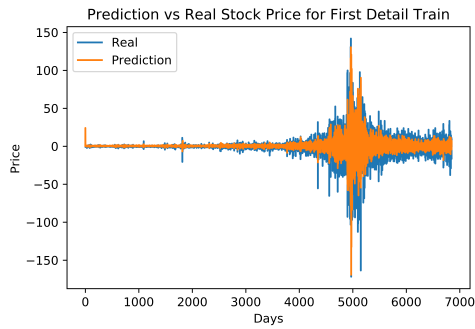
(a) Loss Values of the First Detail

(b) Loss Values of the Second Detail

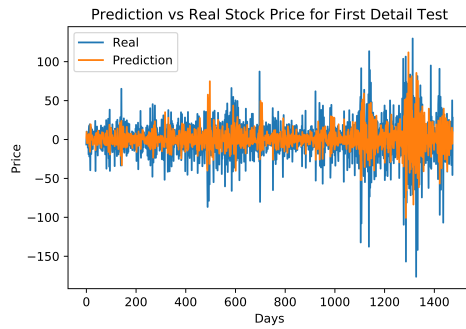


(c) Loss Values of the Approximation

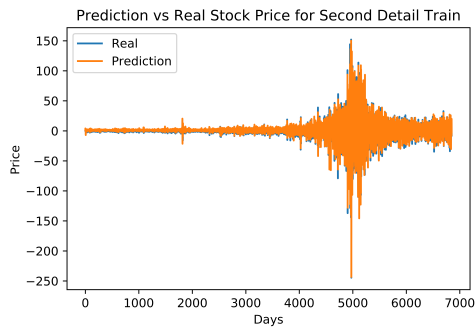
Figure 5.18: LSTM Model+MRA, Configuration 1 (NASDAQ): Loss Values of Each Level



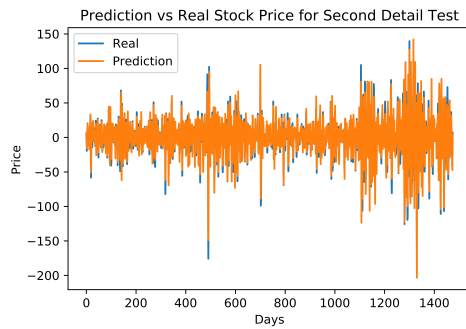
(a) 1st Detail-Train Prediction



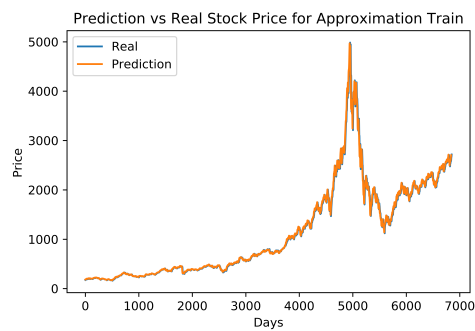
(b) 1st Detail-Test Prediction



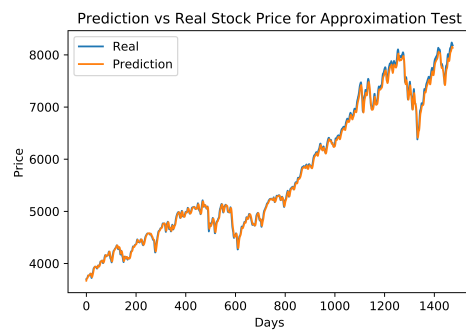
(c) 2nd Detail-Train Prediction



(d) 2nd Detail-Test Prediction

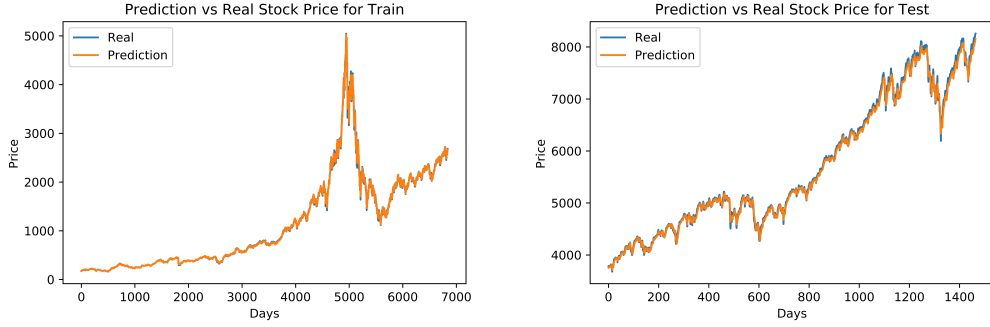


(e) Approximation-Train Prediction



(f) Approximation-Test Prediction

Figure 5.19: LSTM Model+MRA, Configuration 1 (NASDAQ): Train and Test Predictions of the First Detail, the Second Detail and the Approximation Parts



(a) Mean of the Predicted Train Values vs. Observed Train Values for Reconstructed Data (b) Mean of the Predicted Test Values vs. Observed Test Values for Reconstructed Data

Figure 5.20: LSTM Model+MRA, Configuration 1 (NASDAQ): Reconstructed Results

given in Figure 5.20a and Figure 5.20b respectively.

Although the predictions look very good in both S&P500 and NASDAQ figures, even small differences are significant in trading. In the approach using LSTM only, the predictions look very favorable also. However, no matter how good the models' predictions are, small differences between them are of great importance for investors. These differences can only be noticed from the results of error metrics.

#### 5.4 Hybrid LSTM-Wavenet Model without MRA

In this section, we use PPS as an internal activation function for the time series. Details of polynomial wavelets are given in Section 4.2. Suppose the polynomial wavelet function created by the  $n$ th derivative of the sigmoid function is utilized for the  $n$ th node of the LSTM layer. In that case, outcomes are much worse than employing a single polynomial wavelet function in each cell.

The polynomial wavelet function created by the 6th derivative of the sigmoid function is utilized to form a wavelet activation function in each cell. Coefficients of the polynomial function can be found by using (4.18) which is given in Section 4.2. Data preparation, time-steps, and network structure are the same as shown in Section 5.2.

First of all we create 16 different activation functions since we have 16 nodes in LSTM layer. We use the 6th derivative of the sigmoid function to create polynomial wavelet activation functions as we mentioned above. Each activation function with index  $j$  is generated as:

$$\psi_6^j(x) = \psi \left( \frac{x - u_j}{v_j} \right), \quad (5.1)$$

where  $u$  is the translation (or location) and  $v$  is the dilation (or scale) parameters for  $j = 1, 2, \dots, 16$ . The subscript 6 denotes the degree of derivative.

In [66] it is stated that since wavelets are quickly vanishing functions, it is important not to choose too small dilation parameters. Moreover, Radhwane and Bereksi in [47]

point out that random initialization of the translation and dilation parameters may results in too local wavelets. We initialize these parameters as following:

$$u_1 \approx \frac{1}{2} \left( \frac{\beta - \alpha}{n} \right) \quad (5.2)$$

and

$$v_1 \approx \frac{1}{2} \left( \frac{\beta + \alpha}{n} \right), \quad (5.3)$$

where  $\beta$  is the maximum value in the training set,  $\alpha$  is the minimum value in the training set and  $n$  is the number of LSTM nodes. Therefore initial translation and dilation parameters for S&P500 data are  $u_1 = 40$  and  $v_1 = 50$ . On the other hand, the same values of  $u_1 = 40$  and  $v_1 = 50$  are obtained under the same configurations for the initial parameters of NASDAQ.

Each activation function for S&P500 and NASDAQ time series is produced as follows:

$$\psi_6^j(x) = \psi \left( \frac{x - 40 * j}{50 * j} \right), \quad (5.4)$$

where  $j$  is the index of the activation functions.

We use two different approaches for hybrid LSTM-Wavenet model without MRA. In the first method the same wavelet activation function is used for all 16 LSTM nodes. Wavelet activation function that is used for S&P500 and NASDAQ data is

$$\psi_6^6(x) = \psi \left( \frac{x - 240}{300} \right), \quad (5.5)$$

where the superscript 6 denotes that we use the 6th activation function and the subscript 6 shows the degree of the derivative.

In the second strategy, we use (5.4) for S&P500 and NASDAQ. In other words, the translation and the dilation parameters are changing in each LSTM nodes. As a result, 16 different activation functions are produced.

In Table 5.37 configuration parameters of the first approach are given for S&P500. In other words, these are parameters for hybrid LSTM-Wavenet structure by using the same wavelet activation function for all LSTM nodes. After using these parameters we get results as given in Table 5.38. It is clear that results are better than LSTM and LSTM+MRA methods.

In Figure 5.21, the model of the configuration 1 is given for S&P500.

In Figure 5.22a, loss values of all experiments are given for S&P500 according to the first approach. It is a good loss function graph since after some epochs validation loss falls under training loss and both loss functions converge to zero value. In Figure 5.22b and in Figure 5.22c, average values of predicted train and test values versus observed prices are given respectively. It is seen that both training and test predictions follow the related observed values nicely. In previous approaches given in Section 5.2 and Section 5.3, it is also seen that the predictions follow the observed data very well. For this reason, it is more useful to examine the error scores.



Table 5.37: Hybrid LSTM-Wavenet Model, Configuration 1 (S&P500):  
Talos Configuration

<b>Configuration 1</b>			
loss	mse		
optimizer	Adam		
time-steps	10		
batch size	1024		
epochs	100		
<i><u>Layer Parameters</u></i>	<u>LSTM Layer</u>	<u>Between Layers</u>	<u>Dense Layer</u>
# of nodes	16	-	128
kernel initializer	normal	-	-
batch normalization	-	yes	-
kernel regularizer	l1, l2=1e-6	-	l1, l2=1e-6
recurrent regularizer	l1, l2=1e-4	-	-
bias regularizer	l1, l2=1e-5	-	l1, l2=1e-5
activation	$\psi_6^6(x)^a$	ReLU	ReLU

$$^a \psi_6^6(x) = \psi\left(\frac{x-240}{300}\right)$$

Table 5.38: Hybrid LSTM-Wavenet Model, Configuration 1 (S&P500): Mean Scores  
for the Train set and the Test set by Running 1000 Experiments

<b>Configuration 1</b>		
experiment size	1000	
time taken by process	459m45s	
<i><u>Monte Carlo Scores</u></i>	<u>Reconstructed Train Scores</u>	<u>Reconstructed Test Scores</u>
RMSE	9.71	22.82
Scaled RMSE	0.00	0.01
$R^2$	1.00	1.00
MAE	6.37	17.33
EVS	1.00	1.00
ME	86.70	126.88
MdAE	3.76	13.74

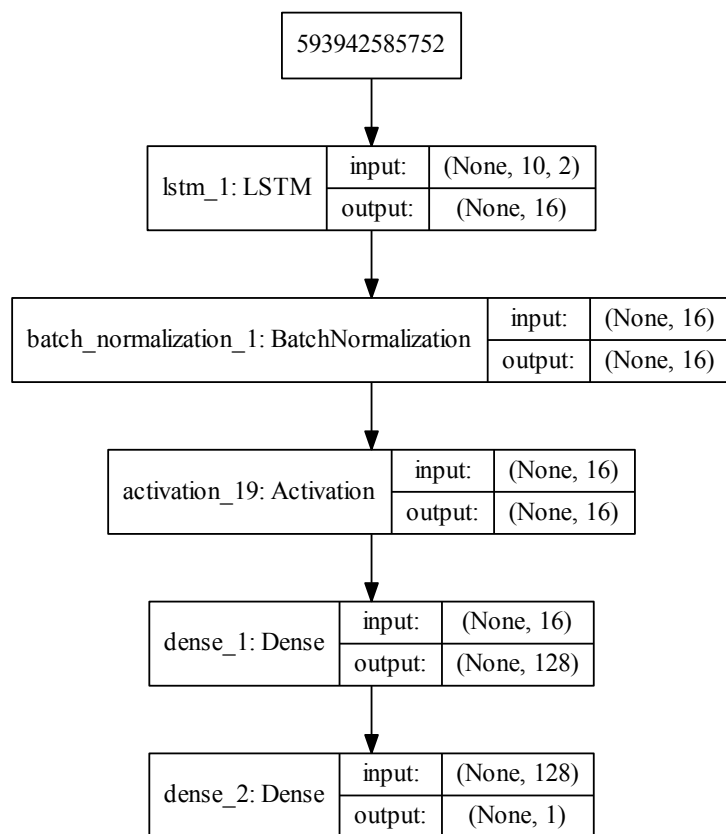
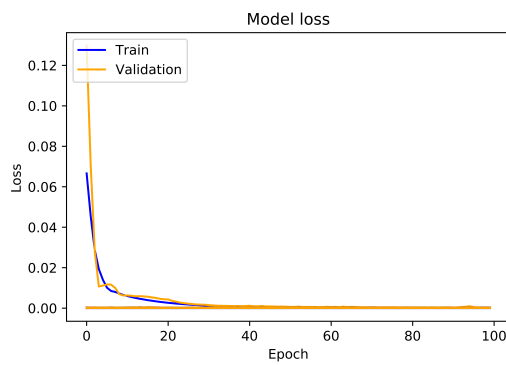
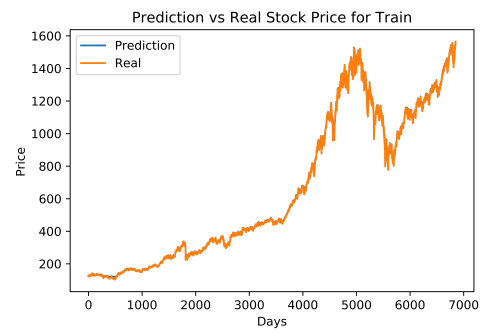


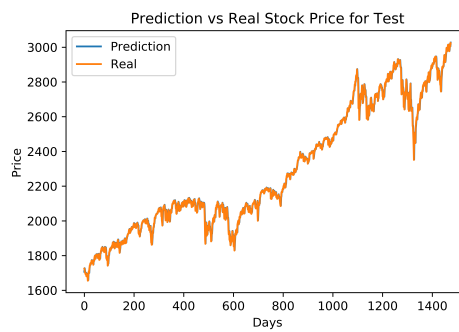
Figure 5.21: Hybrid LSTM-Wavenet Model, Configuration 1 (S&P500): Model Structure



(a) Loss Values



(b) Mean of the Predicted Train Values vs. the Observed Train Values



(c) Mean of the Predicted Test Values vs. the Observed Test Values

Figure 5.22: Hybrid LSTM-Wavenet Model, Configuration 1 (S&P500): Results

Table 5.39: Hybrid LSTM-Wavenet Model by API Structure, Configuration 2 (S&P500): Talos Configuration

<b>Configuration 2</b>			
loss	mse		
optimizer	Adam		
time-steps	10		
batch size	1024		
epochs	100		
<i><b>Layer Parameters</b></i>	<u>LSTM Layer</u>	<u>Between Layers</u>	<u>Dense Layer</u>
# of nodes	16x1	-	128
kernel initializer	normal	-	-
batch normalization	-	yes	-
kernel regularizer	l1, l2=1e-6	-	l1, l2=1e-6
recurrent regularizer	l1, l2=1e-4	-	-
bias regularizer	l1, l2=1e-5	-	l1, l2=1e-5
activation	$\psi_6^j(x)^a$	ReLU	ReLU

$$^a \psi_6^j(x) = \psi\left(\frac{x-40*j}{50*j}\right) \text{ for } j = 1, \dots, 16$$

Configuration parameters of the second approach are given in Table 5.39 for S&P500. In the second approach, the translation and the dilation parameters are altering in each node's activation function. Results are given in Table 5.40. Again outcomes are preferable to LSTM and LSTM+MRA methods. Nevertheless, operating unchanging dilation and translation parameters gives superior results in terms of both error scores and computation time.

The model of the the configuration 2 for S&P500 is presented in Figure A.1 in Appendix A. Because different activation functions are used in the LSTM layer, each LSTM node is shown in separate branches.

In Figure 5.23a, loss values of all experiments are given for S&P500 with respect to the second approach. In Figure 5.23b and in Figure 5.23c, mean values of predicted train and test values versus observed prices are given respectively.

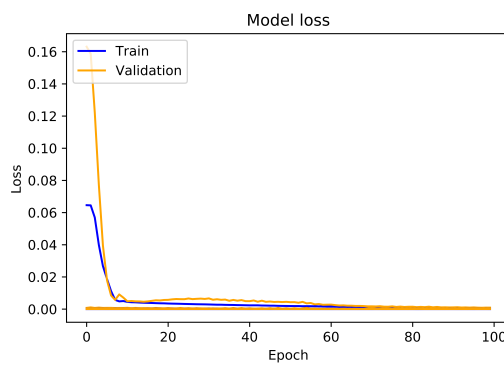
Similarly, the same two approaches are applied, and results are obtained for NASDAQ. Parameters of the The Talos configuration 1 are given in Table 5.41 for NASDAQ. Outcomes of error metrics are given in Table 5.42. Results are preferable to LSTM and LSTM+MRA approaches.

In Figure 5.24, the model of the configuration 1 is given for NASDAQ where hybrid LSTM-Wavenet model is utilized.

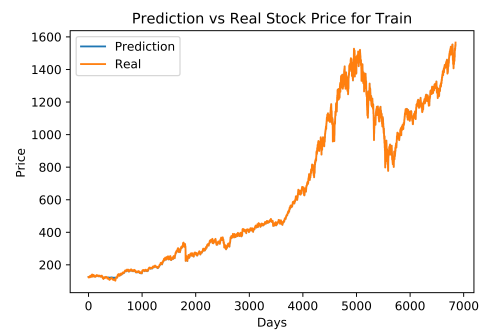
In Figure 5.25a, loss values of all experiments are given for NASDAQ with respect to the configuration 1. In Figure 5.25b and in Figure 5.25c, average values of predicted train and test values versus observed prices are given respectively. Since the prediction graphs are similar, improvements in LSTM and LSTM+MRA can be noticed by looking at the error scores.

Table 5.40: Hybrid LSTM-Wavenet Model by API Structure, Configuration 2 (S&P500): Mean Scores for the Train set and the Test set by Running 1000 Experiments

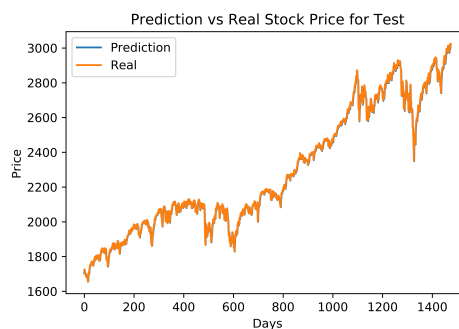
<b>Configuration 2</b>		
experiment size	1000	
time taken by process	2464m15s	
<i>Monte Carlo Scores</i>	Reconstructed Train Scores	Reconstructed Test Scores
RMSE	9.58	24.04
Scaled RMSE	0.00	0.01
$R^2$	1.00	0.99
MAE	6.29	18.40
EVS	1.00	1.00
ME	84.81	128.34
MdAE	3.77	14.48



(a) Loss Values



(b) Mean of the Predicted Train Values vs. the Observed Train Values



(c) Mean of the Predicted Test Values vs. the Observed Test Values

Figure 5.23: Hybrid LSTM-Wavenet Model by API Structure, Configuration 2 (S&P500): Results

Table 5.41: Hybrid LSTM-Wavenet Model, Configuration 1 (NASDAQ): Talos Configuration

<b>Configuration 1</b>			
loss	mse		
optimizer	Adam		
time-steps	10		
batch size	1024		
epochs	100		
<i><b>Layer Parameters</b></i>	<u>LSTM Layer</u>	<u>Between Layers</u>	<u>Dense Layer</u>
# of nodes	16	-	128
kernel initializer	normal	-	-
batch normalization	-	yes	-
kernel regularizer	l1, l2=1e-6	-	l1, l2=1e-6
recurrent regularizer	l1, l2=1e-4	-	-
bias regularizer	l1, l2=1e-5	-	l1, l2=1e-5
activation	$\psi_6^6(x)^a$	ReLU	ReLU

<sup>a</sup>  $\psi_6^6(x) = \psi\left(\frac{x-240}{300}\right)$

Table 5.42: Hybrid LSTM-Wavenet Model, Configuration 1 (NASDAQ): Mean Scores for the Train set and the Test set by Running 1000 Experiments

<b>Configuration 1</b>		
experiment size	1000	
time taken by process	391m24s	
<i><b>Monte Carlo Scores</b></i>	<u>Reconstructed Train Scores</u>	<u>Reconstructed Test Scores</u>
RMSE	30.15	64.57
Scaled RMSE	0.00	0.01
$R^2$	1.00	1.00
MAE	16.18	46.77
EVS	1.00	1.00
ME	355.23	373.57
MdAE	7.54	34.45

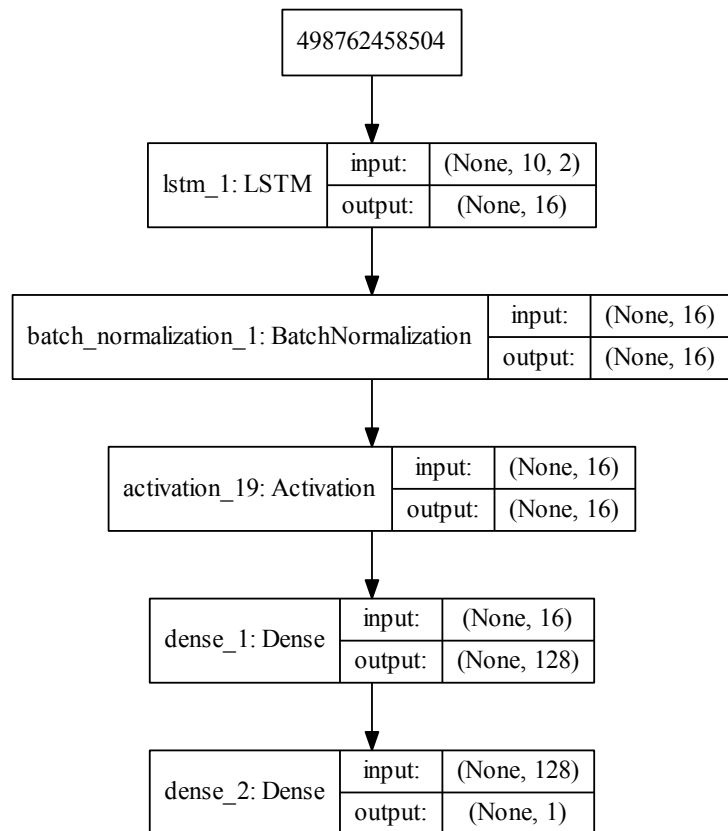
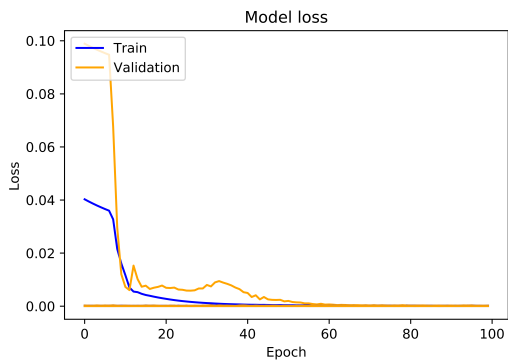
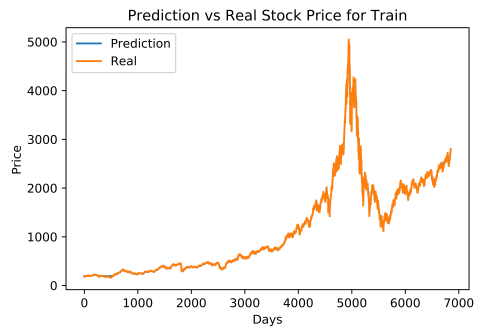


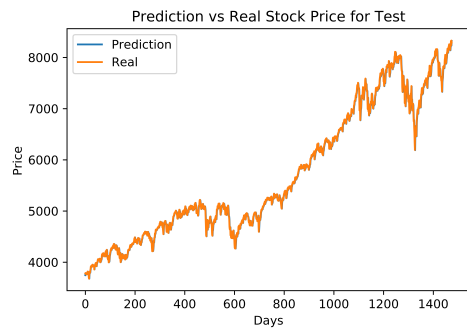
Figure 5.24: Hybrid LSTM-Wavenet Model, Configuration 1 (NASDAQ): Model Structure



(a) Loss Values



(b) Mean of the Predicted Train Values vs. the Observed Train Values



(c) Mean of the Predicted Test Values vs. the Observed Test Values

Figure 5.25: Hybrid LSTM-Wavenet Model, Configuration 1 (NASDAQ): Results



Table 5.43: Hybrid LSTM-Wavenet Model by API Structure, Configuration 2 (NASDAQ): Talos Configuration

<b>Configuration 2</b>			
loss	mse		
optimizer	Adam		
time-steps	10		
batch size	1024		
epochs	100		
<i>Layer Parameters</i>	<u>LSTM Layer</u>	<u>Between Layers</u>	<u>Dense Layer</u>
# of nodes	16x1	-	128
kernel initializer	normal	-	-
batch normalization	-	yes	-
kernel regularizer	l1, l2=1e-6	-	l1, l2=1e-6
recurrent regularizer	l1, l2=1e-4	-	-
bias regularizer	l1, l2=1e-5	-	l1, l2=1e-5
activation	$\psi_6^j(x)^a$	ReLU	ReLU

$$^a \psi_6^j(x) = \psi\left(\frac{x-40*j}{50*j}\right) \text{ for } j = 1, \dots, 16$$

Parameters of the Talos configuration 2 are given in Table 5.43 for NASDAQ. Results of error metrics are given in Table 5.44. It gives better results than the LSTM and LSTM+MRA methods. However, working with fixed dilation and translation parameters produces more favorable results in terms of both error scores and computation time.

The model of the configuration 2 for NASDAQ, where hybrid LSTM-Wavenet model is used, is given in Figure A.2 in Appendix A.

In Figure 5.26a, loss values of all experiments are given for NASDAQ with respect to the configuration 2. In Figure 5.26b and in Figure 5.26c, average values of predicted train and test values versus observed prices are represented respectively.

## 5.5 Hybrid LSTM-Wavenet Model with MRA

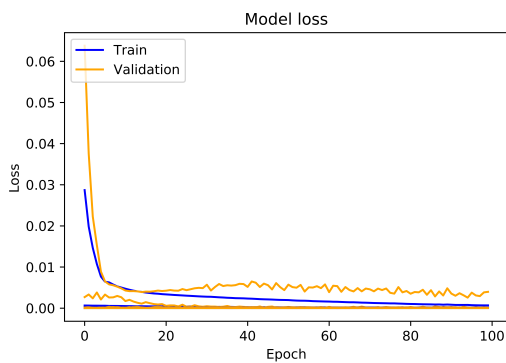
In this section, we combine MRA and the hybrid LSTM-Wavenet model. MRA, data preparation, and selection of the time-step are made precisely in the same way with the method given in Section 5.3. On the other hand, we utilize two different approaches mentioned in Section 5.4 to create activation functions.

We model each level by taking advantage of both Talos optimization and hand-tuning again. After that, a thousand experiments are operated for model fitting and train/test parts' predictions. After running experiments, the mean of error metrics and train/test predictions are estimated for each wavelet level.

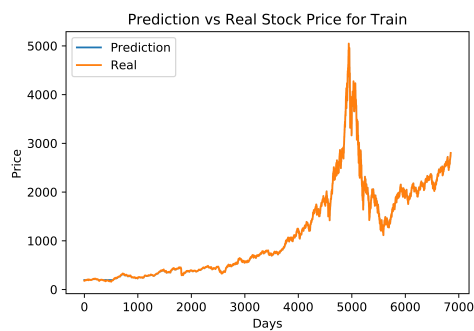
In Table 5.45 selected Talos configuration parameters of the first detail, the second

Table 5.44: Hybrid LSTM-Wavenet Model by API Structure, Configuration 2 (NASDAQ): Mean Scores for the Train set and the Test set by Running 1000 Experiments

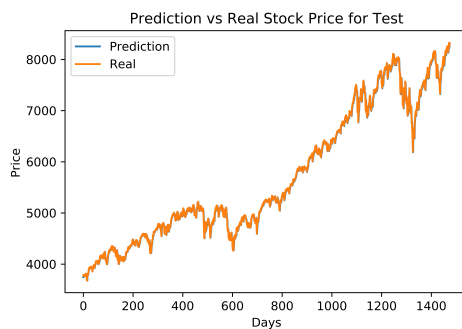
<b>Configuration 2</b>		
experiment size	1000	
time taken by process	2379m44s	
<b>Monte Carlo Scores</b>	<u>Reconstructed Train Scores</u>	<u>Reconstructed Test Scores</u>
RMSE	30.70	67.30
Scaled RMSE	0.00	0.01
$R^2$	1.00	1.00
MAE	16.94	49.41
EVS	1.00	1.00
ME	355.76	378.39
MdAE	8.53	36.80



(a) Loss Values



(b) Mean of the Predicted Train Values vs. the Observed Train Values



(c) Mean of the Predicted Test Values vs. the Observed Test Values

Figure 5.26: Hybrid LSTM-Wavenet Model by API Structure, Configuration 2 (NASDAQ): Results

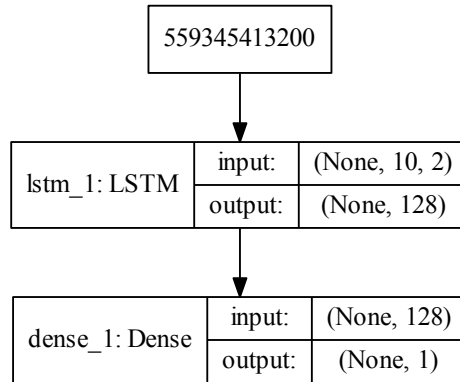


Figure 5.27: Hybrid LSTM-Wavenet Model+MRA, Configuration 1 (S&P500): Model Structure of the First Detail

detail and the approximation parts are given for S&P500. It is seen that the first approach of the wavenet structure is applied to the approximation part. Wavenet structure is not used in the first detail and the second detail since the mentioned levels may be counted as a noise. Catching the noise structure is handled without the need for wavenets. Daubechies wavelet, particularly the “db2=D4” filter and MODWT, are used to decompose the signal with a level of 2 like we select in Section 5.3.

Mean RMSE scores of each wavelet levels for S&P500 are given in Table 5.46 according to the configuration 1.

In Table 5.47 average error scores of reconstructed train and test data are shown for S&P500 with respect to configuration 1. The proposed method (hybrid LSTM-Wavenet+MRA) outperforms the previous methods according to the error results.

Model structures of the first detail, the second detail and the approximation for S&P500 by the configuration 1 are given in Figure 5.27, Figure 5.28 and Figure 5.29 respectively.

Loss values of the first detail, the second detail and the approximation for S&P500 by the configuration 1 are given in Figure 5.30a, Figure 5.30b and Figure 5.30c respectively.

Averaged train and test predictions of the first detail, the second detail and the approximation for S&P500 by the configuration 1 are depicted in Figure 5.31.

Reconstructed average train and test predictions for S&P500 by the configuration 1 are shown in Figure 5.32a and Figure 5.32b respectively.

In Table 5.48 selected Talos configuration parameters of the first detail, the second detail and the approximation parts are given for S&P500. At that point, we remark that the second approach, we mention in Section 5.4, is used to create activation functions. Since each activation function is created by (5.4), we observe 16 different

Table 5.45: Hybrid LSTM-Wavenet Model+MRA, Configuration 1 (S&P500): Talos Configuration

<b>Configuration 1</b>							
wavlet filter = db2 (D4)							
wavlet level = 2							
<b>Detail 1</b>		<b>Detail 2</b>		<b>Approximation</b>			
loss	mse	loss	mse	loss	mse		
optimizer	Adam	optimizer	Adam	optimizer	Adam		
time-steps	10	time-steps	10	time-steps	10		
batch size	1024	batch size	1024	batch size	1024		
epochs	50	epochs	50	epochs	100		
<b>Layer Parameters</b>	LSTM Layer	<b>Layer Parameters</b>	LSTM Layer	<b>Layer Parameters</b>	LSTM Layer		
# of nodes	128	# of nodes	128	# of nodes	16		
kernel initializer	-	kernel initializer	-	kernel initializer	normal		
batch normalization	-	batch normalization	-	batch normalization	-		yes
kernel regularizer	-	kernel regularizer	-	kernel regularizer	11, 12=1e-6		-
recurrent regularizer	-	recurrent regularizer	-	recurrent regularizer	11, 12=1e-4		-
bias regularizer	-	bias regularizer	-	bias regularizer	11, 12=1e-5		-
activation	-	activation	-	activation	$\psi_6^6(x)^a$		ReLU
							ReLU

<sup>a</sup>  $\psi_6^6(x) = \psi\left(\frac{x-240}{300}\right)$

Table 5.46: Hybrid LSTM-Wavenet Model+MRA, Configuration 1 (S&P500): Mean Scores of the Wavelet Levels for the Train set and the Test set by Running 1000 Experiments

<b>Configuration 1</b>			
experiment size	1000	wavelet filter	db2 (D4)
time taken by process	2103m30s	wavelet level	2
	<b>Monte Carlo Test RMSE Scores</b>	<b>Monte Carlo Train RMSE Scores</b>	
<b>Detail 1</b>	3.61	7.46	
<b>Detail 2</b>	2.30	4.48	
<b>Approximation</b>	7.73	17.25	

Table 5.47: Hybrid LSTM-Wavenet Model+MRA, Configuration 1 (S&P500): Mean Scores for the Synthesized Train set and the Synthesized Test set by Running 1000 Experiments

<b>Configuration 1</b>			
experiment size	1000	wavelet filter	db2 (D4)
time taken by process	2103m30s	wavelet level	2
<b>Monte Carlo Scores</b>	<b>Reconstructed Train Scores</b>	<b>Reconstructed Test Scores</b>	
RMSE	8.04	17.93	
Scaled RMSE	0.00	0.01	
$R^2$	1.00	1.00	
MAE	6.56	15.17	
EVS	1.00	1.00	
ME	50.05	77.81	
MdAE	5.46	13.47	

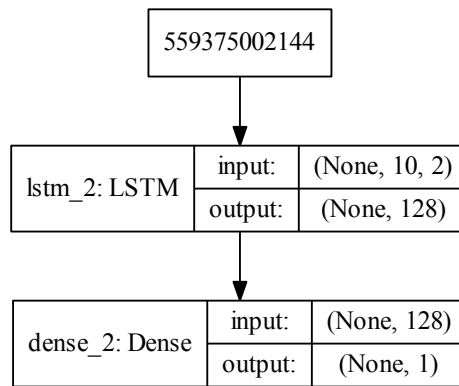


Figure 5.28: Hybrid LSTM-Wavenet Model+MRA, Configuration 1 (S&P500): Model Structure of the Second Detail

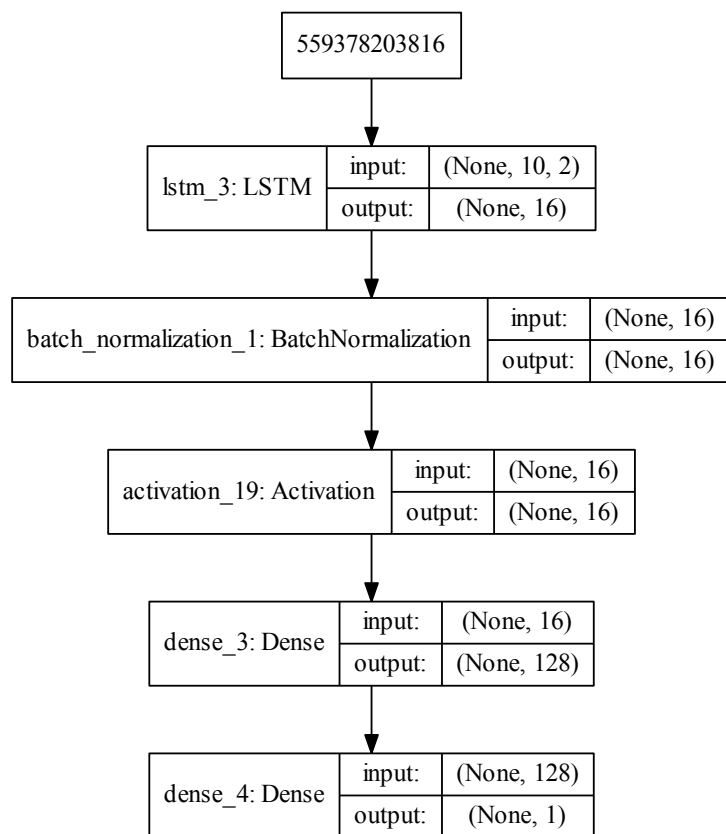
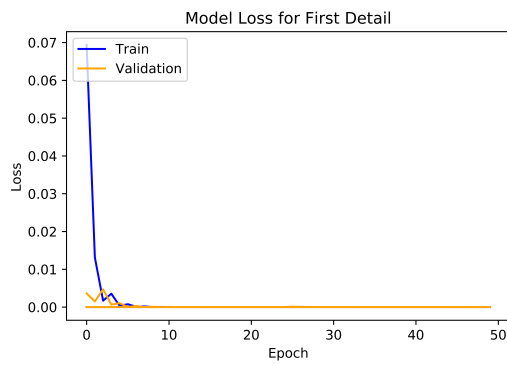
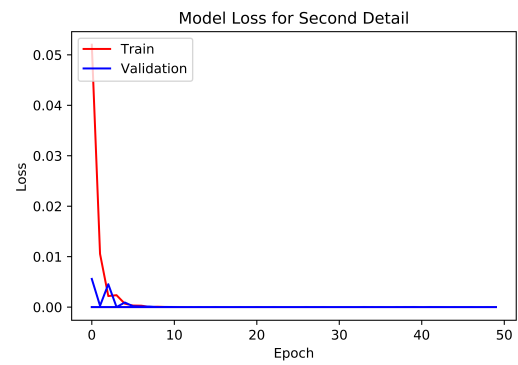


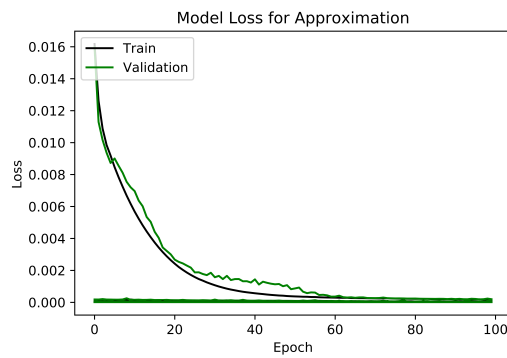
Figure 5.29: Hybrid LSTM-Wavenet Model+MRA, Configuration 1 (S&P500): Model Structure of the Approximation



(a) Loss Values of the First Detail

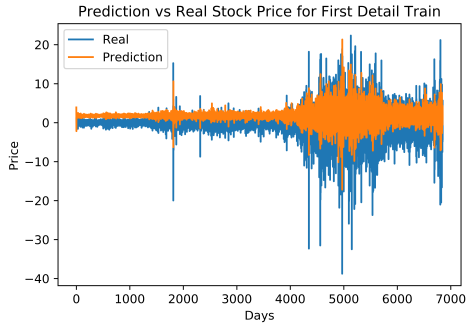


(b) Loss Values of the Second Detail

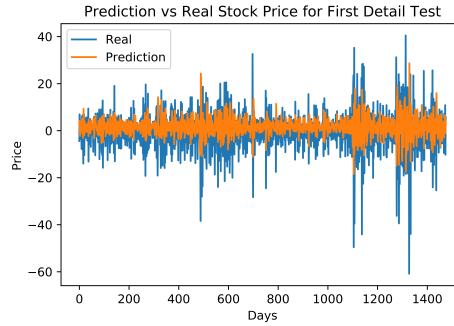


(c) Loss Values of the Approximation

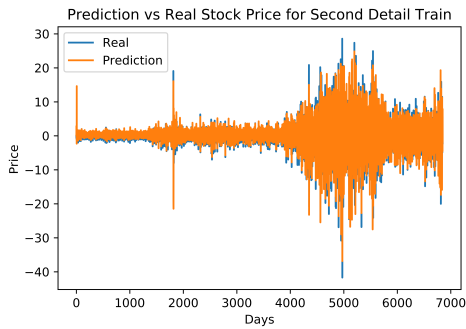
Figure 5.30: Hybrid LSTM-Wavenet Model+MRA, Configuration 1 (S&P500): Loss Values of Each Level



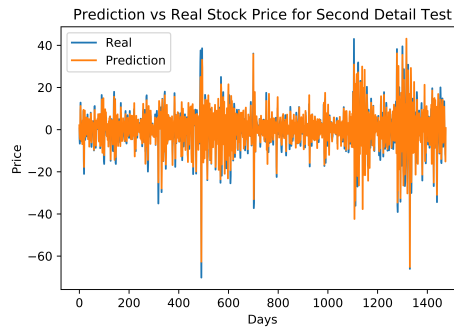
(a) 1st Detail-Train Prediction



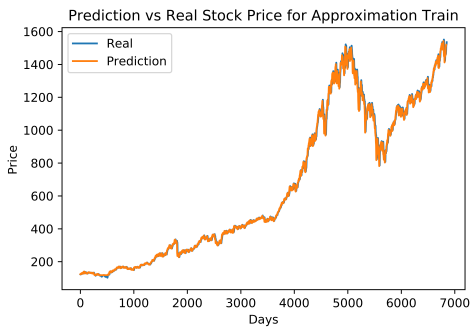
(b) 1st Detail-Test Prediction



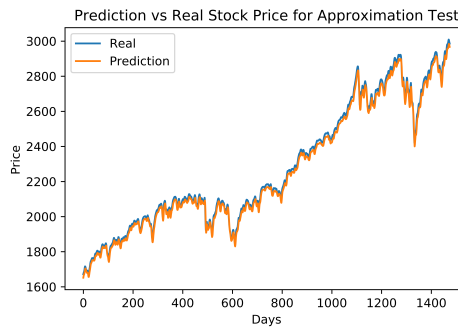
(c) 2nd Detail-Train Prediction



(d) 2nd Detail-Test Prediction



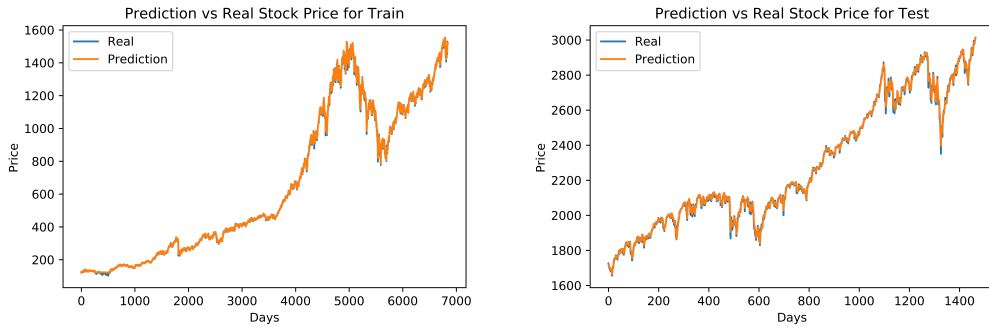
(e) Approximation-Train Prediction



(f) Approximation-Test Prediction

Figure 5.31: Hybrid LSTM-Wavenet Model+MRA, Configuration 1 (S&P500): Train and Test Predictions of the First Detail, the Second Detail and the Approximation Parts of S&P500





(a) Mean of the Predicted Train Values vs. Observed Train Values for Reconstructed Data (b) Mean of the Predicted Test Values vs. Observed Test Values for Reconstructed Data

Figure 5.32: Hybrid LSTM-Wavenet Model+MRA, Configuration 1 (S&P500): Reconstructed Results

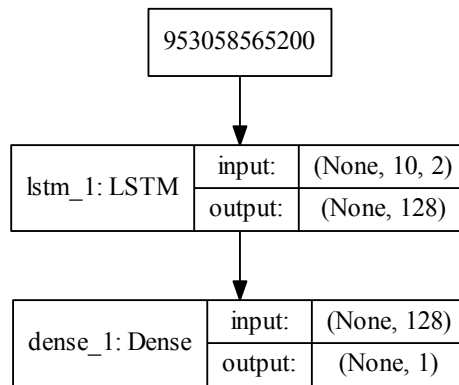


Figure 5.33: Hybrid LSTM-Wavenet Model+MRA by API Structure, Configuration 2 (S&P500): Model Structure of the First Detail

activation functions specific to the corresponding nodes.

Mean RMSE scores of each wavelet levels for S&P500 are given in Table 5.49 according to the configuration 2.

In Table 5.50 average error scores of reconstructed train and test data are given for S&P500 with respect to the configuration 2. This configuration also outperforms LTSM, LSTM+MRA and hybrid LSTM-Wavenet methods. However, the configuration 1 is preferable to the configuration 2, relating to the error results and the computation time.

Model structures of the first detail and the second detail for S&P500 by the configuration 2 are given in Figure 5.33, and Figure 5.34, respectively. The model for the approximation part is given by Figure A.3 in Appendix A.

Loss values of the first detail, the second detail and the approximation for S&P500 by

Table 5.48: Hybrid LSTM-Wavenet Model+MRA by API Structure, Configuration 2 (S&P500): Talos Configuration

<b>Configuration 2</b>							
wavelet filter = db2 (D4)							
wavelet level = 2							
<i>Detail 1</i>		<i>Detail 2</i>		<i>Approximation</i>			
loss	mse	loss	mse	loss	mse		
optimizer	Adam	optimizer	Adam	optimizer	Adam		
time-steps	10	time-steps	10	time-steps	10		
batch size	1024	batch size	1024	batch size	1024		
epochs	50	epochs	50	epochs	100		
<i>Layer Parameters</i>	LSTM Layer	<i>Layer Parameters</i>	LSTM Layer	<i>Layer Parameters</i>	LSTM Layer	Between Layers	Dense Layer
# of nodes	128	# of nodes	128	# of nodes	16	-	128
kernel initializer	-	kernel initializer	-	kernel initializer	normal	-	-
batch normalization	-	batch normalization	-	batch normalization	-	yes	-
kernel regularizer	-	kernel regularizer	-	kernel regularizer	11, 12=1e-6	-	11, 12=1e-6
recurrent regularizer	-	recurrent regularizer	-	recurrent regularizer	11, 12=1e-4	-	-
bias regularizer	-	bias regularizer	-	bias regularizer	11, 12=1e-5	-	11, 12=1e-5
activation	-	activation	-	activation	$\psi_6^j(x)^a$	ReLU	ReLU

$$^a \psi_6^j(x) = \psi\left(\frac{x-40*j}{50*j}\right) \text{ for } j = 1, \dots, 16$$

Table 5.49: Hybrid LSTM-Wavenet Model+MRA by API Structure, Configuration 2 (S&P500): Mean Scores of the Wavelet Levels for the Train set and the Test set by Running 1000 Experiments

<b>Configuration 2</b>			
experiment size	1000	wavelet filter	db2 (D4)
time taken by process	4318m36s	wavelet level	2
	<b>Monte Carlo Test RMSE Scores</b>	<b>Monte Carlo Train RMSE Scores</b>	
<b>Detail 1</b>	3.60	7.47	
<b>Detail 2</b>	2.29	4.48	
<b>Approximation</b>	7.12	17.27	

Table 5.50: Hybrid LSTM-Wavenet Model+MRA by API Structure, Configuration 2 (S&P500): Mean Scores for the Synthesized Train set and the Synthesized Test set by Running 1000 Experiments

<b>Configuration 2</b>			
experiment size	1000	wavelet filter	db2 (D4)
time taken by process	4318m36s	wavelet level	2
<b>Monte Carlo Scores</b>	<b>Reconstructed Train Scores</b>	<b>Reconstructed Test Scores</b>	
RMSE	7.45	17.96	
Scaled RMSE	0.00	0.01	
$R^2$	1.00	0.99	
MAE	5.81	15.18	
EVS	1.00	1.00	
ME	49.43	77.41	
MdAE	4.56	13.50	

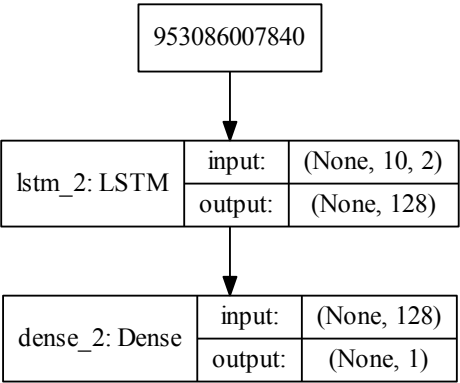


Figure 5.34: Hybrid LSTM-Wavenet Model+MRA by API Structure, Configuration 2 (S&P500): Model Structure of the Second Detail

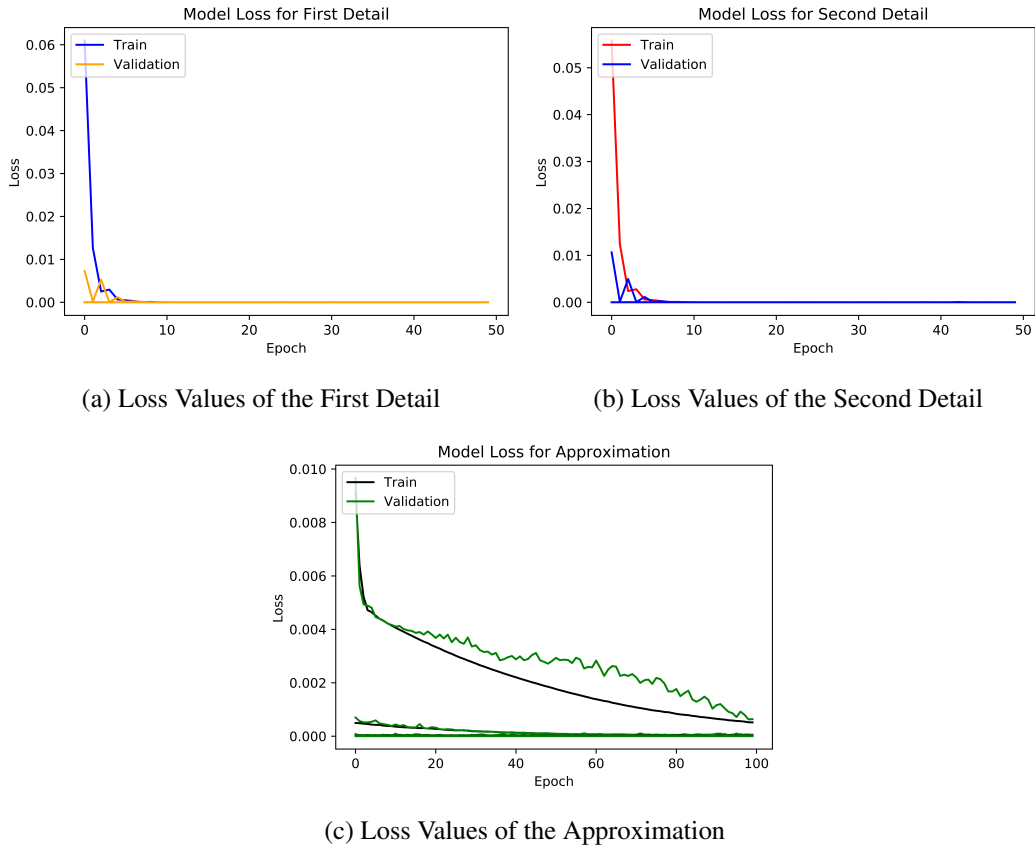


Figure 5.35: Hybrid LSTM-Wavenet Model+MRA, Configuration 2 (S&P500): Loss Values of Each Level

the configuration 2 are given in Figure 5.35a, Figure 5.35b and Figure 5.35c respectively.

Averaged train and test predictions of the first detail, the second detail, and the approximation for S&P500 by the configuration two are given in Figure 5.36.

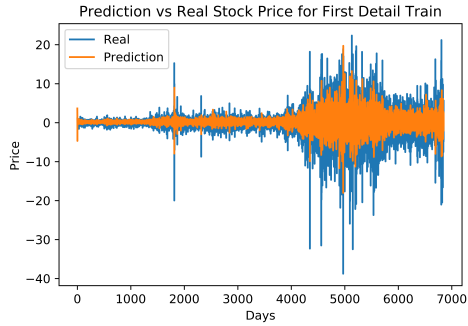
Reconstructed average train and test predictions for S&P500 by the configuration 2 are given in Figure 5.37a and Figure 5.37b respectively.

After showing the results for the S&P500, similarly, the outcomes for the NASDAQ are shown. Again, as stated in Section 5.4, two distinct approaches are employed. While activation function parameters are fixed in one, these parameters change for each node in the other.

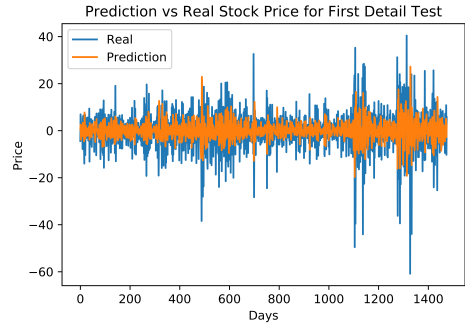
In Table 5.51 selected parameters of the first detail, the second detail and the approximation parts by Talos configuration 1 are given for NASDAQ.

Mean RMSE scores of each wavelet levels for NASDAQ are given in Table 5.52 according to the configuration 1.

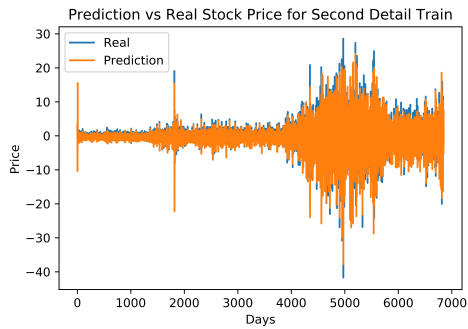
In Table 5.53 average error scores of reconstructed train and test data are given for NASDAQ with respect to the configuration 1. The proposed method outperforms the previous models.



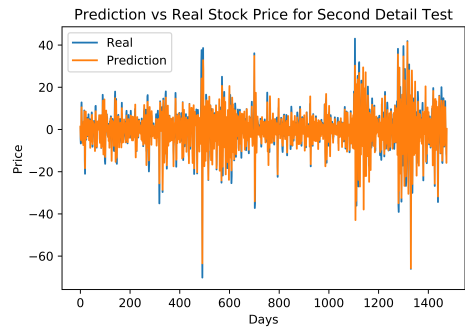
(a) 1st Detail-Train Prediction



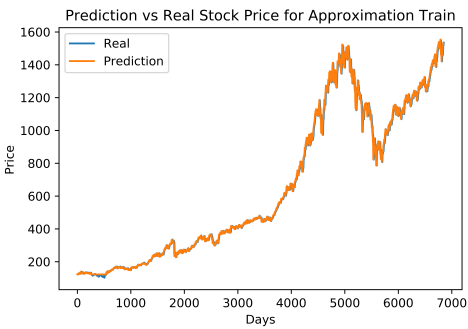
(b) 1st Detail-Test Prediction



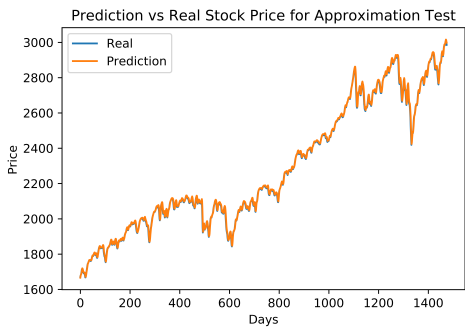
(c) 2nd Detail-Train Prediction



(d) 2nd Detail-Test Prediction



(e) Approximation-Train Prediction



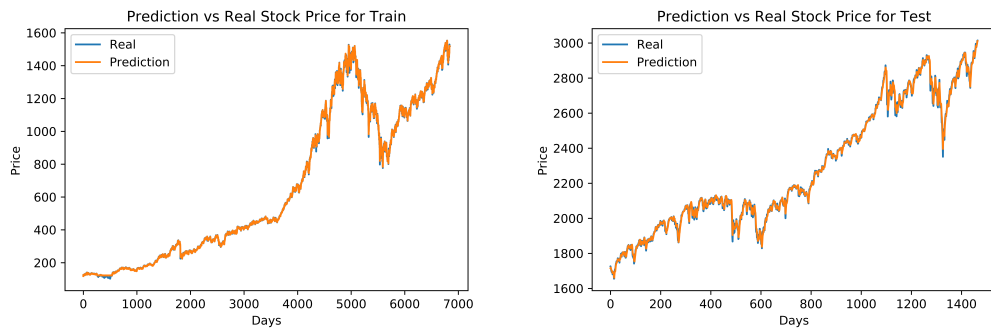
(f) Approximation-Test Prediction

Figure 5.36: Hybrid LSTM-Wavenet Model+MRA by API Structure, Configuration 2 (S&P500): Train and Test Predictions of the First Detail, the Second Detail and the Approximation Parts of S&P500

Table 5.51: Hybrid LSTM-Wavenet Model+MRA, Configuration 1 (NASDAQ): Talos Configuration

<b>Configuration 1</b>							
wavlet filter = db2 (D4)							
wavlet level = 2							
<b>Detail 1</b>		<b>Detail 2</b>		<b>Approximation</b>			
loss	mse	loss	mse	loss	mse		
optimizer	Adam	optimizer	Adam	optimizer	Adam		
time-steps	10	time-steps	10	time-steps	10		
batch size	1024	batch size	1024	batch size	1024		
epochs	50	epochs	50	epochs	100		
<b>Layer Parameters</b>	LSTM Layer	<b>Layer Parameters</b>	LSTM Layer	<b>Layer Parameters</b>	LSTM Layer		
# of nodes	128	# of nodes	128	# of nodes	16		Between Layers
kernel initializer	-	kernel initializer	-	kernel initializer	normal		Dense <sup>8</sup> Layer
batch normalization	-	batch normalization	-	batch normalization	-		-
kernel regularizer	-	kernel regularizer	-	kernel regularizer	11, 12=1e-6		11, 12=1e-6
recurrent regularizer	-	recurrent regularizer	-	recurrent regularizer	11, 12=1e-4		-
bias regularizer	-	bias regularizer	-	bias regularizer	11, 12=1e-5		11, 12=1e-5
activation	-	activation	-	activation	$\psi_6^6(x)^a$		ReLU

<sup>a</sup>  $\psi_6^6(x) = \psi\left(\frac{x-240}{300}\right)$



(a) Mean of the Predicted Train Values vs. Observed Train Values for Reconstructed Data (b) Mean of the Predicted Test Values vs. Observed Test Values for Reconstructed Data

Figure 5.37: Hybrid LSTM-Wavenet Model+MRA, Configuration 2 (S&P500): Reconstructed Results

Table 5.52: Hybrid LSTM-Wavenet Model+MRA, Configuration 1 (NASDAQ): Mean Scores of the Wavelet Levels for the Train set and the Test set by Running 1000 Experiments

<b>Configuration 1</b>			
experiment size	1000	wavelet filter	db2 (D4)
time taken by process	2257m23s	wavelet level	2
	<b>Monte Carlo Train RMSE Scores</b>	<b>Monte Carlo Test RMSE Scores</b>	
<b>Detail 1</b>	10.89	22.02	
<b>Detail 2</b>	5.88	12.78	
<b>Approximation</b>	17.63	38.79	

Table 5.53: Hybrid LSTM-Wavenet Model+MRA, Configuration 1 (NASDAQ): Mean Scores for the Synthesized Train set and the Synthesized Test set by Running 1000 Experiments

<b>Configuration 1</b>			
experiment size	1000	wavelet filter	db2 (D4)
time taken by process	2257m23s	wavelet level	2
<b>Monte Carlo Scores</b>	<b>Reconstructed Train Scores</b>	<b>Reconstructed Test Scores</b>	
RMSE	18.41	40.62	
Scaled RMSE	0.00	0.00	
$R^2$	1.00	1.00	
MAE	12.02	32.21	
EVS	1.00	1.00	
ME	214.12	184.95	
MdAE	7.86	26.91	

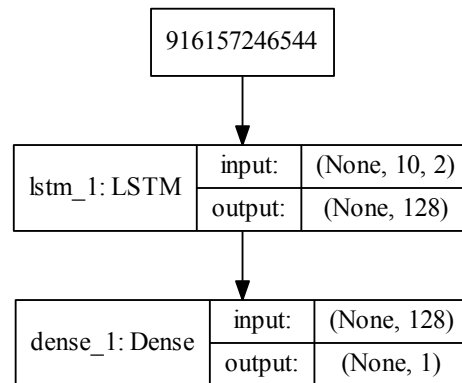


Figure 5.38: Hybrid LSTM-Wavenet Model+MRA, Configuration 1 (NASDAQ): Model Structure of the First Detail

In Figure 5.38, Figure 5.39 and Figure 5.40 model structures of the first detail, the second detail and the approximation for NASDAQ by the configuration 1 are given respectively.

In Figure 5.41a, Figure 5.41b and Figure 5.41c loss values of the first detail, the second detail and the approximation for NASDAQ by the configuration 1 are given respectively.

Averaged train and test predictions of the first detail, the second detail and the approximation for NASDAQ by the configuration 1 are given in Figure 5.42.

Reconstructed average train and test predictions for NASDAQ by the configuration 1 are given in Figure 5.43a and Figure 5.43b respectively.

Concerning the configuration 2 for NASDAQ, the parameters of activation functions are changing, as explained in Section 5.4.

In Table 5.54 selected parameters of the first detail, the second detail and the approximation parts by Talos configuration 2 are given for NASDAQ.

Mean RMSE scores of each wavelet levels for NASDAQ are given in Table 5.55 according to the configuration 2.

In Table 5.56 average error scores of reconstructed train and test data are given for NASDAQ with respect to the configuration 2. The results show that the configuration 2 has better error results than LSTM, LSTM+MRA, and hybrid LSTM-Wavenet methods have. But still, configuration 1 is superior to the configuration 2 according to error results and the computation time.

In Figure 5.44, and Figure 5.45 model structures of the first detail and the second detail for NASDAQ by the configuration 2 are given, respectively. In Chapter A, the approximation part's model structure is given in Figure A.4.



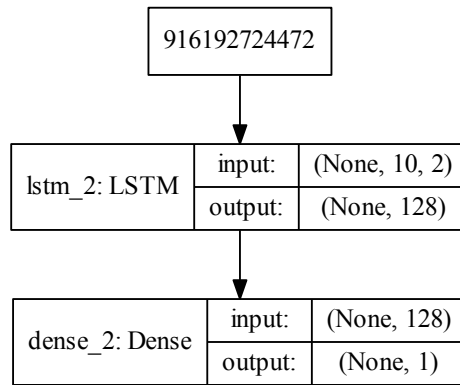


Figure 5.39: Hybrid LSTM-Wavenet Model+MRA, Configuration 1 (NASDAQ): Model Structure of the Second Detail

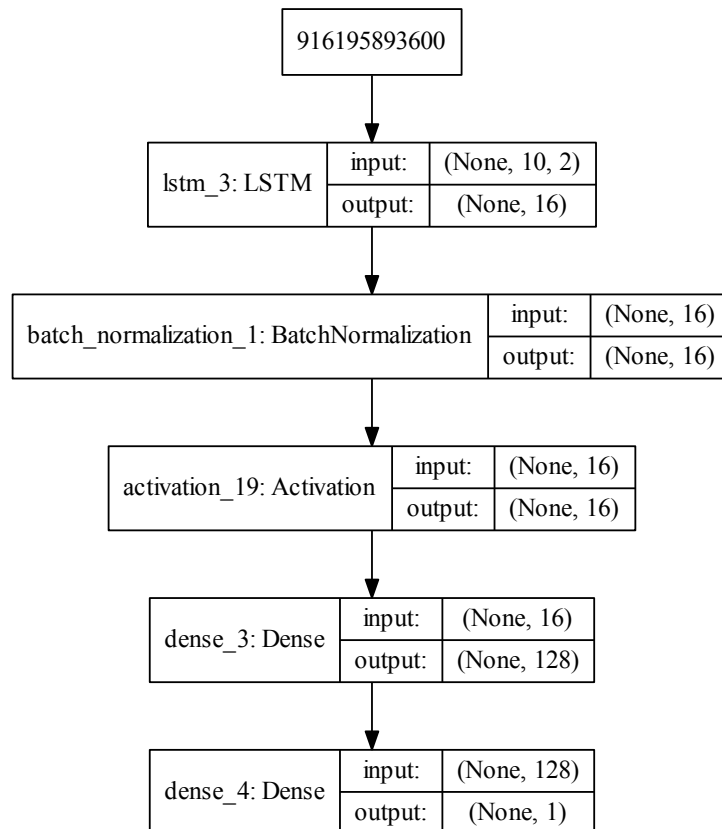
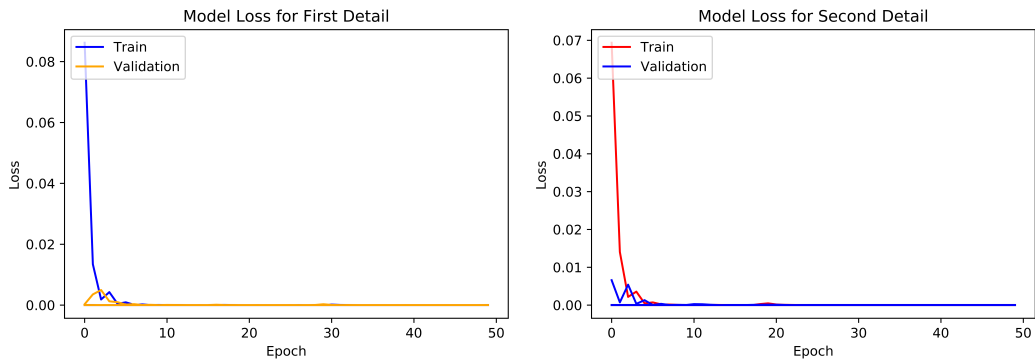
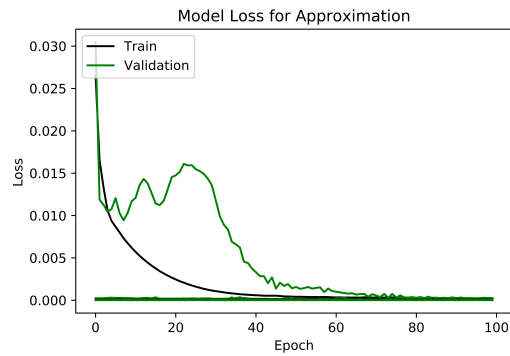


Figure 5.40: Hybrid LSTM-Wavenet Model+MRA, Configuration 1 (NASDAQ): Model Structure of the Approximation

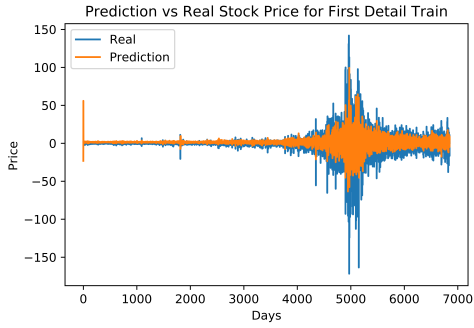


(a) Hybrid LSTM-Wavenet Model+MRA, Configuration 1 (NASDAQ): Loss Values of the First Detail  
 (b) Hybrid LSTM-Wavenet Model+MRA, Configuration 1 (NASDAQ): Loss Values of the Second Detail

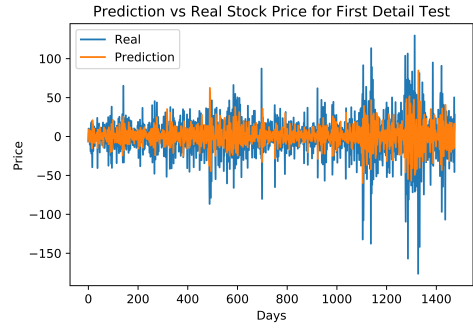


(c) Hybrid LSTM-Wavenet Model+MRA, Configuration 1 (NASDAQ): Loss Values of the Approximation

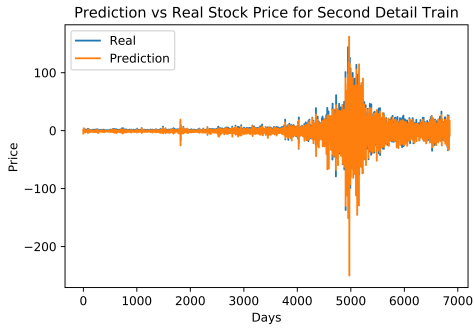
Figure 5.41: Hybrid LSTM-Wavenet Model+MRA, Configuration 1 (NASDAQ): Loss Values of Each Level



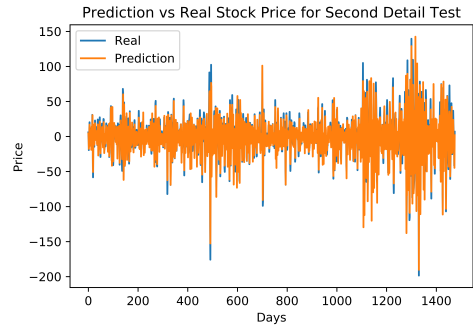
(a) 1st Detail-Train Prediction



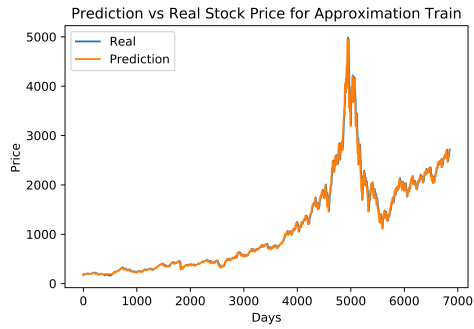
(b) 1st Detail-Test Prediction



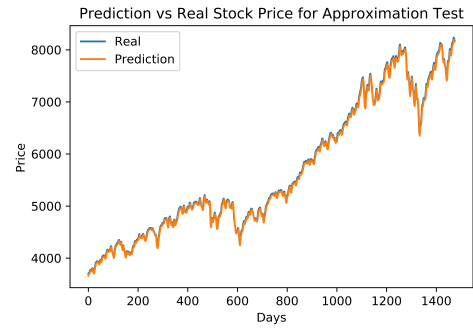
(c) 2nd Detail-Train Prediction



(d) 2nd Detail-Test Prediction

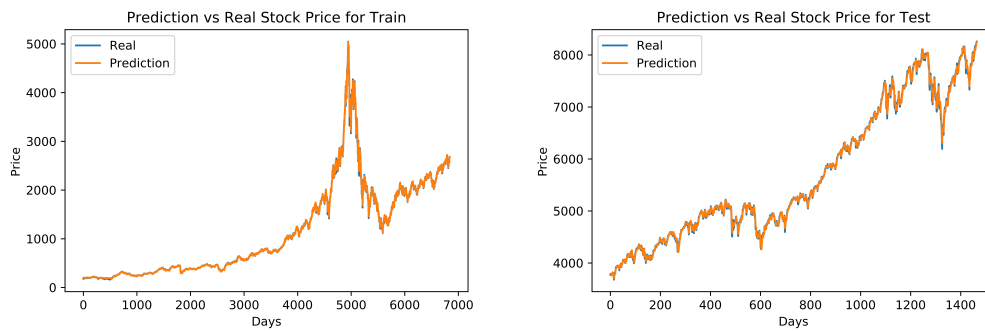


(e) Approximation-Train Prediction



(f) Approximation-Test Prediction

Figure 5.42: Hybrid LSTM-Wavenet Model+MRA, Configuration 1 (NASDAQ): Train and Test Predictions of the First Detail, the Second Detail and the Approximation Parts of S&P500



(a) Mean of the Predicted Train Values vs. Observed Train Values for Reconstructed Data      (b) Mean of the Predicted Test Values vs. Observed Test Values for Reconstructed Data

Figure 5.43: Hybrid LSTM-Wavenet Model+MRA, Configuration 1 (NASDAQ): Reconstructed Results

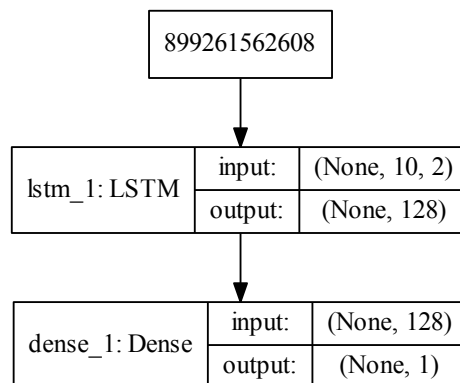


Figure 5.44: Hybrid LSTM-Wavenet Model+MRA by API Structure, Configuration 2 (NASDAQ): Model Structure of the First Detail

Table 5.54: Hybrid LSTM-Wavenet Model+MRA by API Structure, Configuration 2 (NASDAQ): Talos Configuration

<b>Configuration 2</b>							
wavelet filter = db2 (D4)							
wavelet level = 2							
<b>Detail 1</b>		<b>Detail 2</b>		<b>Approximation</b>			
loss	mse	loss	mse	loss	mse		
optimizer	Adam	optimizer	Adam	optimizer	Adam		
time-steps	10	time-steps	10	time-steps	10		
batch size	1024	batch size	1024	batch size	1024		
epochs	50	epochs	50	epochs	100		
<b>Layer Parameters</b>	LSTM Layer	<b>Layer Parameters</b>	LSTM Layer	<b>Layer Parameters</b>	LSTM Layer	<b>Between Layers</b>	<b>Dense Layer</b>
# of nodes	128	# of nodes	128	# of nodes	16	-	128
kernel initializer	-	kernel initializer	-	kernel initializer	normal	-	-
batch normalization	-	batch normalization	-	batch normalization	-	yes	-
kernel regularizer	-	kernel regularizer	-	kernel regularizer	11, 12=1e-6	-	11, 12=1e-6
recurrent regularizer	-	recurrent regularizer	-	recurrent regularizer	11, 12=1e-4	-	-
bias regularizer	-	bias regularizer	-	bias regularizer	11, 12=1e-5	-	11, 12=1e-5
activation	-	activation	-	activation	$\psi_6^j(x)^a$	ReLU	ReLU

$$^a \psi_6^j(x) = \psi\left(\frac{x-40*j}{50*j}\right) \text{ for } j = 1, \dots, 16$$

Table 5.55: Hybrid LSTM-Wavenet Model+MRA by API Structure, Configuration 2 (NASDAQ): Mean Scores of the Wavelet Levels for the Train set and the Test set by Running 1000 Experiments

<b>Configuration 2</b>			
experiment size	1000	wavelet filter	db2 (D4)
time taken by process	4368m23s	wavelet level	2
	<b>Monte Carlo Train RMSE Scores</b>	<b>Monte Carlo Test RMSE Scores</b>	
<b>Detail 1</b>	10.81	21.87	
<b>Detail 2</b>	5.84	12.75	
<b>Approximation</b>	17.85	40.27	

Table 5.56: Hybrid LSTM-Wavenet Model+MRA by API Structure, Configuration 2 (NASDAQ): Mean Scores for the Synthesized Train set and the Synthesized Test set by Running 1000 Experiments

<b>Configuration 2</b>			
experiment size	1000	wavelet filter	db2 (D4)
time taken by process	4368m23s	wavelet level	2
<b>Monte Carlo Scores</b>	<b>Reconstructed Train Scores</b>	<b>Reconstructed Test Scores</b>	
RMSE	18.59	41.97	
Scaled RMSE	0.00	0.01	
$R^2$	1.00	1.00	
MAE	11.84	33.24	
EVS	1.00	1.00	
ME	214.91	188.37	
MdAE	7.23	27.59	

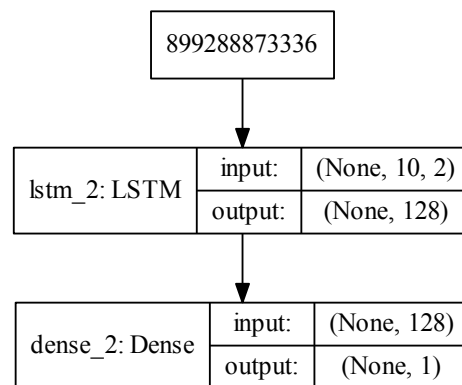
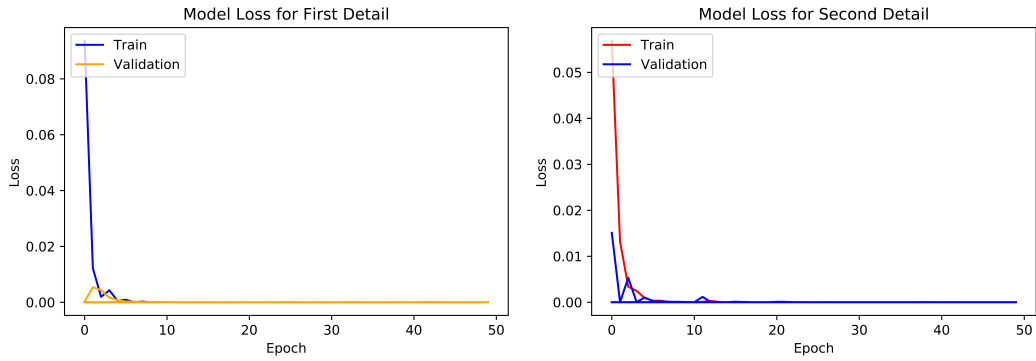
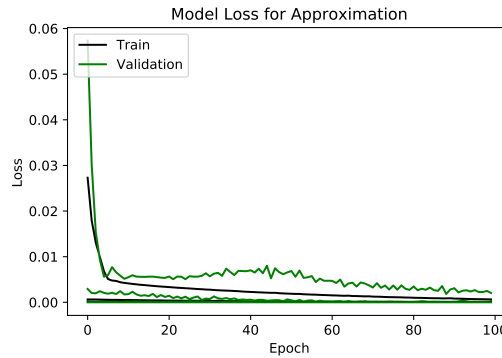


Figure 5.45: Hybrid LSTM-Wavenet Model+MRA by API Structure, Configuration 2 (NASDAQ): Model Structure of the Second Detail



(a) Loss Values of the First Detail

(b) Loss Values of the Second Detail



(c) Loss Values of the Approximation

Figure 5.46: Hybrid LSTM-Wavenet Model+MRA by API Structure, Configuration 2 (NASDAQ): Loss Values of Each Level

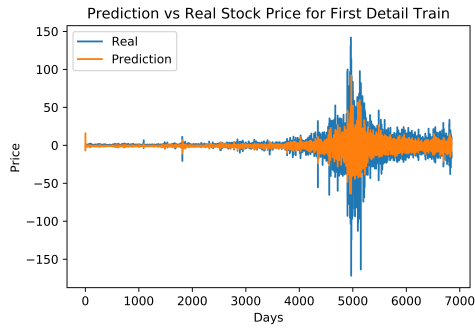
In Figure 5.46a, Figure 5.46b and Figure 5.46c loss values of the first detail, the second detail and the approximation for NASDAQ by the configuration 2 are given respectively.

Averaged train and test predictions of the first detail, the second detail and the approximation for NASDAQ by the configuration 2 are given in Figure 5.47.

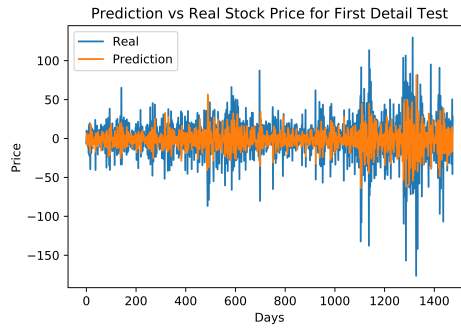
Reconstructed average train and test predictions for NASDAQ by the configuration 2 are given in Figure 5.48a and Figure 5.48b respectively.

## 5.6 Discussion

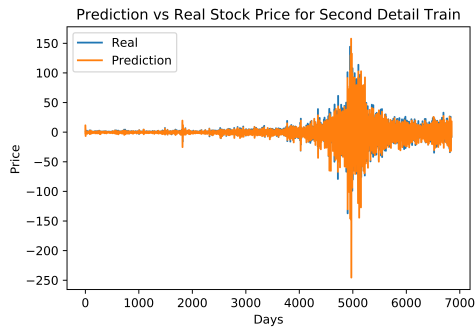
In Table 5.57 and in Table 5.58 error and computation time results of LSTM, LSTM+MRA, hybrid LSTM-Wavenet and hybrid LSTM-Wavenet+MRA methods are given for S&P500 and NASDAQ, respectively. It is seen that using MRA improves the capability of both LSTM and hybrid LSTM-Wavenet models. Changing dilation and translation parameters of the wavelet activation function does not change the error results too much compared to using constant dilation and translation parameters for all nodes in LSTM. However, it causes an excessive computation time. The proposed method (hybrid LSTM-Wavenet+MRA) outperforms LSTM, LSTM+MRA, and hybrid LSTM-



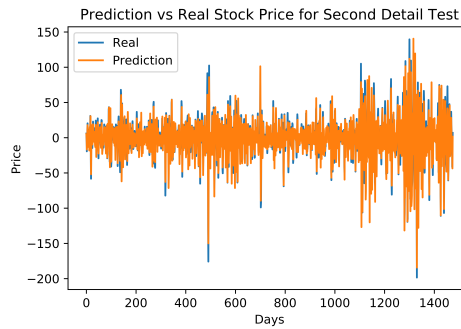
(a) 1st Detail-Train Prediction



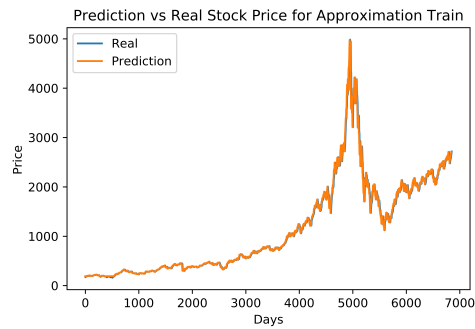
(b) 1st Detail-Test Prediction



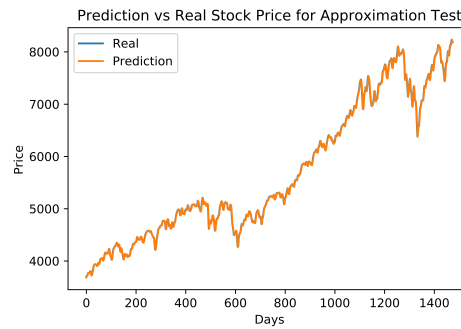
(c) 2nd Detail-Train Prediction



(d) 2nd Detail-Test Prediction



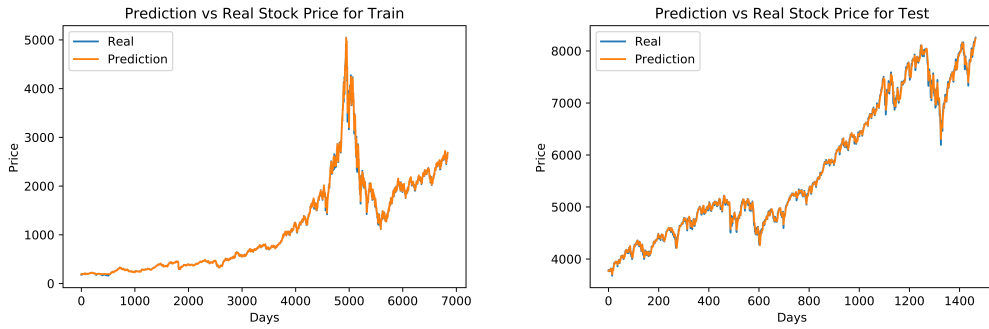
(e) Approximation-Train Prediction



(f) Approximation-Test Prediction

Figure 5.47: Hybrid LSTM-Wavenet Model+MRA by API Structure, Configuration 2 (NASDAQ): Train and Test Predictions of the First Detail, the Second Detail and the Approximation Parts





(a) Mean of the Predicted Train Values vs. Observed Train Values for Reconstructed Data (b) Mean of the Predicted Test Values vs. Observed Test Values for Reconstructed Data

Figure 5.48: Hybrid LSTM-Wavenet Model+MRA by API Structure, Configuration 2 (NASDAQ): Reconstructed Results

Wavenet methods for financial time series in terms of error metrics. Therefore, it is clear that utilizing wavelets in both MRA and activation functions improves the performance of the train/test predictions.

In Section 5.1 it is stated that all training, validation and, test sets show different characteristics. Notably, the changes between the training sets and the test sets are relatively significant. Distinctions between the descriptive statistics of the training set and descriptive statistics of the test set are noticeable for both S&P500 and NASDAQ. Since such a difference occurs in the time series's progression, it is reasonable to see an absolute distinction between the RMSE values of the training set and the test set for both S&P500 and NASDAQ. (see Table 5.57 and Table 5.58)

RMSE errors are relatively low compared to time series values. For example, the mean of training data is 654.61, and the mean of the test data is 2283.34 for S&P500. On the other hand, in the results obtained from the four methods, the maximum RMSE value for the training set is 15.29, and the minimum RMSE value is 7.45 for S&P500. Furthermore, in all results given in Table 5.57 and Table 5.58, the maximum RMSE is 29.06, and the minimum RMSE is 17.93 for the test set for S&P500.

On the other hand, when we look at NASDAQ data, the mean of training data is 1147.35, and the mean of the test data is 5701.76. Again, in the results belong to the four methods, the maximum RMSE value for the training set is 36.65, and the minimum RMSE value is 18.41 for NASDAQ. Additionally, the maximum RMSE is 85.31, and the minimum RMSE is 40.62, according to the test set for NASDAQ.

When the SRMSE values are examined, it is seen that the error difference between training and test sets is tiny for both S&P500 and NASDAQ. The reason is that the time series observations are quite high compared to the observed RMSE values. Besides, in the analysis of statistical values given in Section 5.1, it is seen that there are many differences and changes between training and test data. This situation causes some differences between the RMSE values of training and test data. (see Table 5.57 and Table 5.58)

As mentioned in Section 5.2, SRMSE,  $R^2$ , and EVS values are too close for both the

Table 5.57: Summary Table for Results (S&P500): Mean Scores for the Train set and the Test set by Running 1000 Experiments

	time	RMSE	SRMSE	MAE	MdAE
<b>LSTM (conf. 3)</b>	228m29s				
train scores		11.93	0.00	8.06	4.86
test scores		29.06	0.01	23.30	19.93
<b>LSTM+MRA (conf. 1)</b>	1981m47s				
train scores		15.29	0.01	12.29	11.11
test scores		26.91	0.01	22.66	20.36
<b>Hybrid (conf. 2, by API)</b>	2464m15s				
train scores		9.58	0.00	6.29	3.77
test scores		24.04	0.01	18.40	14.48
<b>Hybrid (conf.1)</b>	459m45s				
train scores		9.71	0.00	6.37	3.76
test scores		22.82	0.01	17.33	13.74
<b>Hybrid+MRA (conf. 2, by API)</b>	4318m36s				
train scores		7.45	0.00	5.81	4.56
test scores		17.96	0.01	15.18	13.50
<b>Hybrid+MRA (conf. 1)</b>	2103m30s				
train scores		8.04	0.00	6.56	5.46
test scores		17.93	0.01	15.17	13.47

Table 5.58: Summary Table for Results (NASDAQ): Mean Scores for the Train set and the Test set by Running 1000 Experiments

	time	RMSE	SRMSE	MAE	MdAE
<b>LSTM (conf. 3)</b>	286m52s				
train scores		36.65	0.00	21.44	11.35
test scores		85.31	0.01	66.58	53.22
<b>LSTM+MRA (conf. 1)</b>	2008m34s				
train scores		33.12	0.00	20.74	11.67
test scores		68.41	0.01	52.67	44.05
<b>Hybrid (conf. 2, by API)</b>	2379m44s				
train scores		30.70	0.00	16.94	8.53
test scores		67.30	0.01	49.41	36.80
<b>Hybrid (conf.1)</b>	391m24s				
train scores		30.15	0.00	16.18	7.54
test scores		64.57	0.01	46.77	34.45
<b>Hybrid+MRA (conf. 2, by API)</b>	4368m23s				
train scores		18.59	0.00	11.84	7.23
test scores		41.97	0.01	33.24	27.59
<b>Hybrid+MRA (conf. 1)</b>	2257m23s				
train scores		18.41	0.00	12.02	7.86
test scores		40.62	0.00	32.21	26.91

training set and the test set, so these metrics are insufficient to make comparisons between LSTM, LSTM+MRA, hybrid LSTM-Wavenet and hybrid LSTM-Wavenet+MRA models. On the other hand, ME does not provide an explanatory comparison since it looks at a single error value. Lastly, MdAE is not entirely useful for financial time series since it is robust against outliers. However, outliers like big jumps contain essential information for financial time series.

As a result, RMSE and MAE values of the test results are compared. For both S&P500 and NASDAQ data, RMSE and MAE test error values of both LSTM and hybrid LSTM-Wavenet methods decrease when MRA is used. Similarly, if wavelets are used as an activation function, then RMSE and MAE test error values of both LSTM and LSTM+MRA methods decrease for both stock data. Altering dilation and translation parameters gives insufficient results in terms of both computation time and error scores. Consequently, the best test scores are obtained by the hybrid LSTM-Wavenet+MRA method with constant dilation and translation parameters for both S&P500 and NASDAQ data. It is seen that the single use of wavelets in different subjects increases the performance in time series prediction, and even the use of wavelets for two various approaches (MRA and wavenet) at the same time increases the performance the most.

## CHAPTER 6

### CONCLUSION AND OUTLOOK

The hybrid LSTM-Wavenet+MRA method is presented as a proposed approach due to the motivation of the lack of combining two wavelet methods in the literature. The proposed method is compared with LSTM, LSTM+MRA, and hybrid LSTM-Wavenet methods to predict the next moment. S&P500 and NASDAQ global stock market indexes are selected for the application, where the next step of an index is essential.

Results emphasize the importance of using wavelets in modeling financial time series. When it is desired to make a forecast using observed data, it is noticed that the performance increases significantly. It is seen that the use of the two different wavelet methods gives the best results and also increases the performances. The technique that combines the two different wavelet approaches could enable investors to make better buying and selling decisions. Attention needs to be paid to the theory of wavelets in terms of both MRA and WNN to improve models in applied fields.

For more improvement, the LSTM method could be combined with other nonlinear methods (CNN, etc.) with wavelets to predict different types of time series. It should also be possible to develop a system that makes automatic parameter selection regardless of the kind of time series. Besides, determining the type of wavelets used as activation functions more smartly might also be analyzed in the parameter choice point. Specific to financial time series like stock market data, related news, and other features could also be obtained from different sources and used in the model. Ensuring that the model is correct, the parallel GPU programming, distributed hardware structure, time reduction optimization, and such issues that are required for a real-time operation could improve the applicability of the approach.



## REFERENCES

- [1] J. Adamowski and H. F. Chan, A wavelet neural network conjunction model for groundwater level forecasting, *Journal of Hydrology*, 407, pp. 28–40, 2011.
- [2] A. Arévalo, J. Nino, D. León, G. Hernandez, and J. Sandoval, Deep learning and wavelets for high-frequency price forecasting, in *Computational Science – ICCS 2018*, pp. 385–399, Springer International Publishing, 2018, ISBN 978-3-319-93701-4.
- [3] A. Azzouni and G. Pujolle, A long short-term memory recurrent neural network framework for network traffic matrix prediction, *arXiv: Networking and Internet Architecture*, 2017.
- [4] R. M. Balabin, R. Z. Safieva, and E. I. Lomakina, Wavelet neural network WNN approach for calibration model building based on gasoline near infrared NIR spectra, *Chemometrics and Intelligent Laboratory Systems*, 93, p. 58–62, 2008.
- [5] M. Basta, Additive decomposition and boundary conditions in wavelet-based forecasting approaches, *University of Economics, Prague*, 22(2), pp. 48–70, 2014.
- [6] C. Bergström and O. Hjelm, Impact of time steps on stock market prediction with LSTM, Technical Report TRITA-EECS-EX-2019:292, KTH Royal Institute of Technology, 2019.
- [7] N. Bhatnagar, *Introduction to Wavelet Transforms*, CRC Press, 2020.
- [8] J. Bozic and D. Babic, EUR/RSD exchange rate forecasting using hybrid wavelet-neural model: A case study, *ComSIS*, 12(2), pp. 487–508, 2015.
- [9] J. Brownlee, How to accelerate learning of deep neural networks with batch normalization, <https://machinelearningmastery.com/how-to-accelerate-learning-of-deep-neural-networks-with-batch-normalization/>, accessed: 2019-11-19.
- [10] J. Brownlee, How to develop LSTM models for time series forecasting, <https://machinelearningmastery.com/how-to-develop-lstm-models-for-time-series-forecasting/>, accessed: 2018-11-16.

- [11] S. K. Chandar, M. Sumathi, and S. N. Sivanandam, Prediction of stock market price using hybrid of wavelet transform and artificial neural network, *Indian Journal of Science and Technology*, 9(8), pp. 1–5, 2016.
- [12] F. Chollet et al., Keras, <https://keras.io>, 2015.
- [13] J. Collis, Glossary of deep learning: Batch normalisation, <https://medium.com/deeper-learning/glossary-of-deep-learning-batch-normalisation-8266dcd2fa82>, accessed: 2019-11-14.
- [14] M. Dancho, Time series analysis: Keras LSTM deep learning – part 1, <https://www.business-science.io/timeseries-analysis/2018/04/18/keras-lstm-sunspots-time-series-prediction.html>, accessed: 2018-11-16.
- [15] O. Davydova, 7 types of artificial neural networks for natural language processing, <https://medium.com/datamonsters/artificial-neural-networks-for-natural-language-processing-part-1-64ca9ebfa3b2>.
- [16] H. Esen, F. Ozgen, M. Esen, and A. Sengur, Artificial neural network and wavelet neural network approaches for modelling of a solar air heater, *Expert Systems with Applications*, 36, p. 11240–11248, 2009.
- [17] J. Fernando Marar, E. C. B. Carvalho Filho, W. Li, and L. Deane Sa, Activation function study for wavelet network, in *SPIE-AeroSense '97*, volume 3077, pp. 690–697, 1997.
- [18] J. Fernando Marar, E. C. B. Carvalho Filho, and G. C. Vasconcelos, Function approximation by polynomial wavelets generated from powers of sigmoids, in *SPIE-AeroSense '96*, volume 2762, pp. 365–374, 1996.
- [19] J. Fernando Marar and C. Helder, Human face verification based on multidimensional polynomial powers of sigmoid PPS, in *SciTePress, First International Conference on Health Informatics*, volume 2, pp. 99–106, 2008.
- [20] J. Fernando Marar and C. Helder, Multidimensional polynomial powers of sigmoid PPS wavelet neural networks, in *SciTePress, First International Conference on Bio-inspired Systems and Signal Processing*, volume 2, pp. 261–268, 2008.
- [21] P. Z. Fryzlewicz, *Wavelet Techniques for Time Series and Poisson Data*, Master's thesis, The University of Bristol, September 2003.
- [22] T. Ganegedara, Stock market predictions with LSTM in python, <https://www.datacamp.com/community/tutorials/lstm-python-stock-market>, accessed: 2018-11-16.



- [23] R. Gençay, F. Selçuk, and B. Whitcher, *An Introduction to Wavelets and Other Filtering Methods in Finance and Economics*, Academic Press, 2001.
- [24] L. B. Godfrey and M. S. Gashler, Neural decomposition of time-series data for effective generalization, *IEEE Transactions on Neural Networks and Learning Systems*, 29(7), pp. 2973–2985, 2017.
- [25] E. Hallström, Backpropagation from the beginning, <https://medium.com/erikhallstrm/backpropagation-from-the-beginning-77356edf427d>.
- [26] S. Hochreiter and J. Schmidhuber, Long short-term memory, *Neural Computation*, 9, pp. 1735–1780, 1997.
- [27] M. Holschneider, *Wavelets: An Analysis Tool (Oxford Mathematical Monographs)*, Oxford University Press, 1999.
- [28] L. Iliadis, I. Maglogiannis, G. Tsoumakas, I. Vlahavas, and M. Bramer, *Artificial Intelligence Applications and Innovations: Proceedings of the 5th IFIP Conference on Artificial Intelligence Applications and Innovations (AIAI'2009), April 23-25, 2009, Thessaloniki, Greece*, IFIP Advances in Information and Communication Technology, Springer US, 2009.
- [29] J. Jin and J. Kim, Forecasting natural gas prices using wavelets, time series, and artificial neural networks, *PLoS ONE*, 10(11), 2015.
- [30] D. Jothimani, R. Shankar, and S. S. Yadav, Discrete wavelet transform-based prediction of stock index: A study on national stock exchange fifty index, *Journal of Financial Management and Analysis*, 28(2), pp. 35–49, 2015.
- [31] D. K. Kılıç and O. Uğur, Multiresolution analysis of S&P500 time series, *Annals of Operations Research*, 260(1-2), pp. 197–216, 2018.
- [32] R. Kompella, Using LSTMs to forecast time-series, <https://towardsdatascience.com/using-lstms-to-forecast-time-series-4ab688386b1f>, accessed: 2018-11-16.
- [33] M. Kotila et al., Autonomio talos [computer software], talos: Hyperparameter experiments with tensorflow, pytorch and keras, <http://github.com/autonomio/talos>, 2019.
- [34] T. Křehlik, *Does wavelet decomposition and neural networks help to improve predictability of realized volatility?*, Master's thesis, Charles University in Prague, 2013.
- [35] Y. Liu, L. Guan, C. Hou, H. Han, Z. Liu, Y. Sun, and M. Zheng, Wind power short-term prediction based on LSTM and discrete wavelet transform, *Appl. Sci.*, 9(6), 2019.

- [36] S. G. Mallat, A theory for multiresolution signal decomposition: The wavelet representation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11, pp. 674–693, 1989.
- [37] J. F. Marar and A. Bordin, Multidimensional wavelet neural networks based on polynomial powers of sigmoid: A framework to image verification, *Transdisciplinarity: Projects, Materials, and Processes*, 1, pp. 106–123, 2016.
- [38] P. Masset, *Analysis of financial time-series using Fourier and wavelet methods*, University of Fribourg, 2008.
- [39] J. Mckenna, derivatives of the sigmoid function, <https://joepatmckenna.github.io/calculus/derivative/sigmoid%20function/linear%20albegra/2018/01/20/sigmoid-derivs/>, accessed: 2019-12-27.
- [40] A. A. Minai and R. D. Williams, On the derivatives of the sigmoid, *Neural Networks*, 6, pp. 845–853, 1993.
- [41] C. Napoli, G. Pappalardo, and E. Tramontana, A hybrid neuro–wavelet predictor for qos control and stability, in *AI\*IA 2013: Advances in Artificial Intelligence*, pp. 527–538, Springer International Publishing, 2013.
- [42] U. Okkan, Wavelet neural network model for reservoir inflow prediction, *Scientia Iranica*, 19(6), p. 1445–1455, Dec. 2012.
- [43] C. Olah, Understanding lstm networks, <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>, accessed: 2018-11-17.
- [44] T. Partal and H. K. Cigizoglu, Prediction of daily precipitation using wavelet—neural networks, *Hydrological Sciences Journal*, 54(2), pp. 234–246, 2009.
- [45] D. B. Percival and A. T. Walden, *Wavelet Methods for Time Series Analysis*, Cambridge University Press, 2000.
- [46] N. M. Pindoriya, S. N. Singh, and S. K. Singh, An adaptive wavelet neural network-based energy price forecasting in electricity markets, *IEEE Transactions on Power Systems*, 23(3), p. 1423–1432, Aug. 2008.
- [47] B. Radhwane and F. Bereksi Reguig, Cardiac arrhythmia recognition using wavelet neural network, in *ICMOSS'2010*, 2010.
- [48] R. V. Ramana, B. Krishna, S. R. Kumar, and N. G. Pandey, Monthly rainfall prediction using wavelet neural network analysis, *Water Resources Management*, 27(10), p. 3697–3711, 2013.
- [49] B. M. t. H. Romeny, *Front-End Vision and Multi-Scale Image Analysis*, Springer, Dordrecht, 2003, ISBN 978-1-4020-1503-8.

- [50] R. Schlosser, LSTM time series explorations with keras, <https://www.kaggle.com/ternaryrealm/lstm-time-series-explorations-with-keras>, accessed: 2018-11-16.
- [51] M. Shafaei, J. Adamowski, A. Fakheri-Fard, Y. Dinpashoh, and K. Adamowski, A wavelet-sarima-ann hybrid model for precipitation forecasting, *Journal of Water and Land Development*, pp. 27–36, 2016.
- [52] F. A. Shah and L. Debnath, Wavelet neural network model for yield spread forecasting, *Mathematics*, 5(4), 2017.
- [53] H. L. Shashidhara, S. Lohani, and V. M. Gadre, Function learning using wavelet neural networks, in *Proceedings of IEEE International Conference on Industrial Technology 2000 (IEEE Cat. No.00TH8482)*, volume 1, pp. 335–340, IEEE, 2000.
- [54] A. Subasi, M. Yilmaz, and H. R. Ozcalik, Classification of emg signals using wavelet neural network, *Journal of Neuroscience Methods*, 156, p. 360–367, Oct. 2006.
- [55] P. Sugiartawan, R. Pulungan, and A. K. Sari, Prediction by a hybrid of wavelet transform and long-short-term-memory neural network, (IJACSA) *International Journal of Advanced Computer Science and Applications*, 8(2), pp. 326–332, 2017.
- [56] C. Tan, *Financial Time Series Forecasting Using Improved Wavelet Neural Network*, Master’s thesis, Aarhus University, 2009.
- [57] K. K. Teo, L. Wang, and Z. Lin, Wavelet packet multi-layer perceptron for chaotic time series prediction: Effects of weight initialization, *Computational Science - ICCS 2001*, pp. 310–317, 2001.
- [58] M. Theodosiou, Forecasting monthly and quarterly time series using STL decomposition, *International Journal of Forecasting*, 27(4), pp. 1178–1195, 2011.
- [59] D. Veitch, *Wavelet Neural Networks and Their Application in the Study of Dynamical Systems*, Master’s thesis, Universtiy of York, 2005.
- [60] B. Videkovic, Basics of wavelets, <http://www2.isye.gatech.edu/~brani/isyebayes/bank/handout20.pdf>, accessed: 2019-10-20.
- [61] V. Viswanath, Deep learning - ANN, RNN, LSTM networks, [https://vishnuviswanath.com/ann\\_rnn\\_lstm.html](https://vishnuviswanath.com/ann_rnn_lstm.html), accessed: 2018-11-17.
- [62] T. A. Vuorenmaa, *A Multiresolution Analysis of Stock Market Volatility Using Wavelet Methodology*, Master’s thesis, Universtiy of Helsinki, 2004.

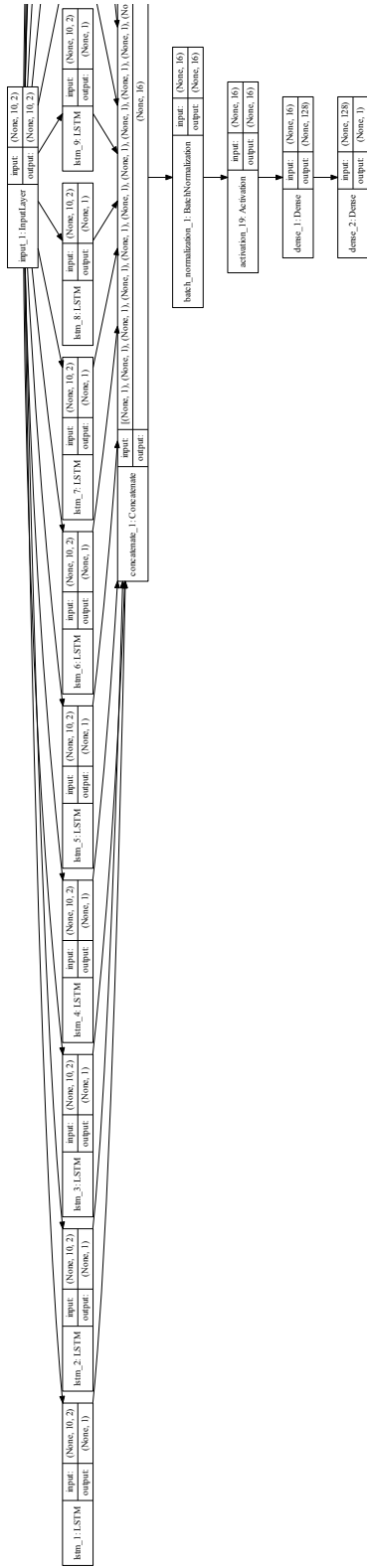
- [63] F. Wang, Y. Yu, Z. Zhang, J. Li, Z. Zhen, and K. Li, Wavelet decomposition and convolutional LSTM networks based improved deep learning model for solar irradiance forecasting, *Appl. Sci.*, 8(8), 2018.
- [64] G. Wang, L. Guo, and H. Duan, Wavelet neural network using multiple wavelet functions in target threat assessment, *The Scientific World Journal*, 2013.
- [65] J. Wang, Z. Wang, J. Li, and J. Wu, Multilevel wavelet decomposition network for interpretable time series analysis, in *KDD*, pp. 2437–2446, 2018.
- [66] X. Wen, H. Zhang, and F. Wang, A wavelet neural network for sar image segmentation, *Sensors*, 9(9), pp. 7509–7515, 2009.
- [67] T. Zheng, K. Fataliyev, and L. Wang, Wavelet neural networks for stock trading, in *Proceedings of SPIE - The International Society for Optical Engineering*, volume 8750, p. 87500A, 05 2013.

## **APPENDIX A**

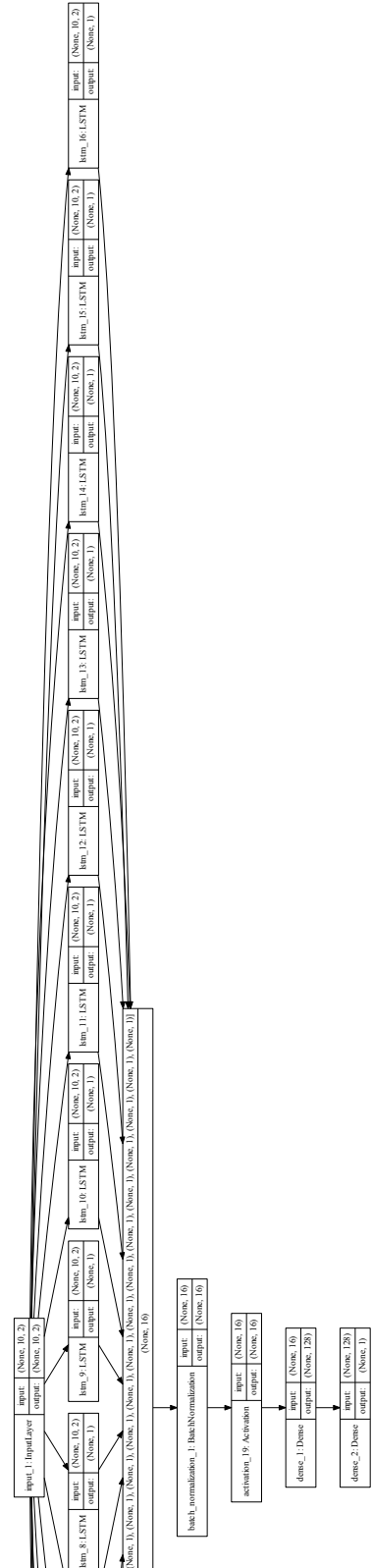
### **MODELS OF API STRUCTURES**

In the appendix part, API structure models of hybrid LSTM-Wavenet and hybrid LSTM-Wavenet+MRA approaches, which we mention in Section 5.4 and Section 5.5, are illustrated for S&P500 and NASDAQ.

Hybrid LSTM-Wavenet models by API structures for S&P500 and NASDAQ are shown in Figure A.1 and Figure A.2, respectively. Subsequently, in Figure A.3 and Figure A.4 hybrid LSTM-Wavenet+MRA models by API structures are presented for each stock market. Each figure shows the right and left parts of the respective model.

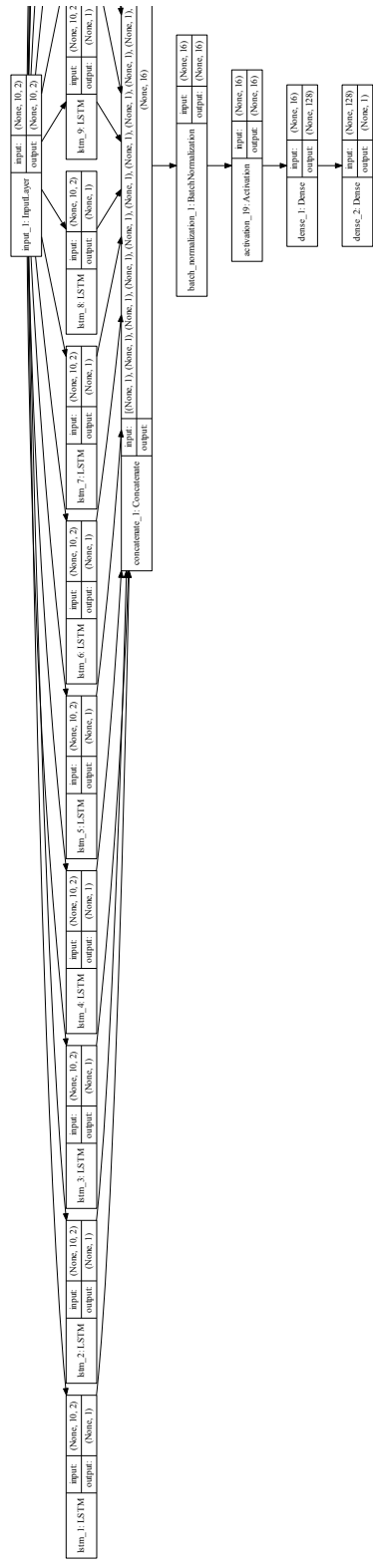


(a) Left Part

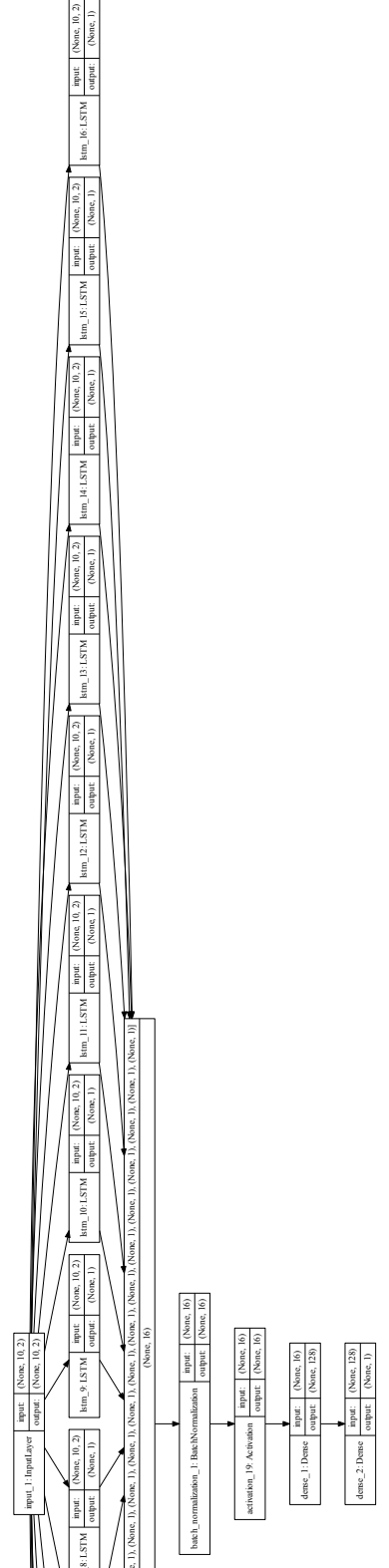


(b) Right Part

Figure A.1: Hybrid LSTM-Wavenet Model by API Structure, Configuration 2 (S&P500): Model Structure

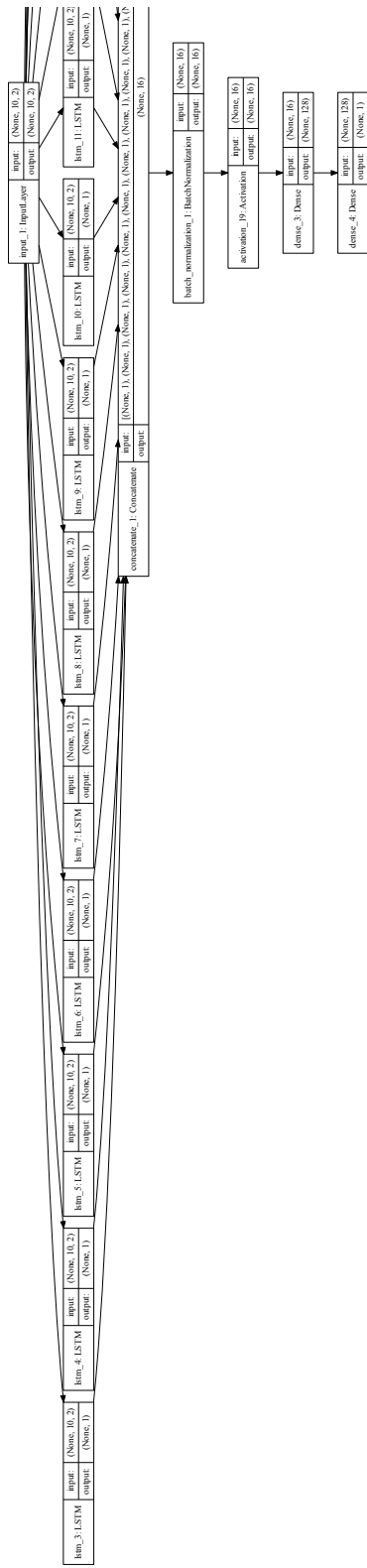


(a) Left Part

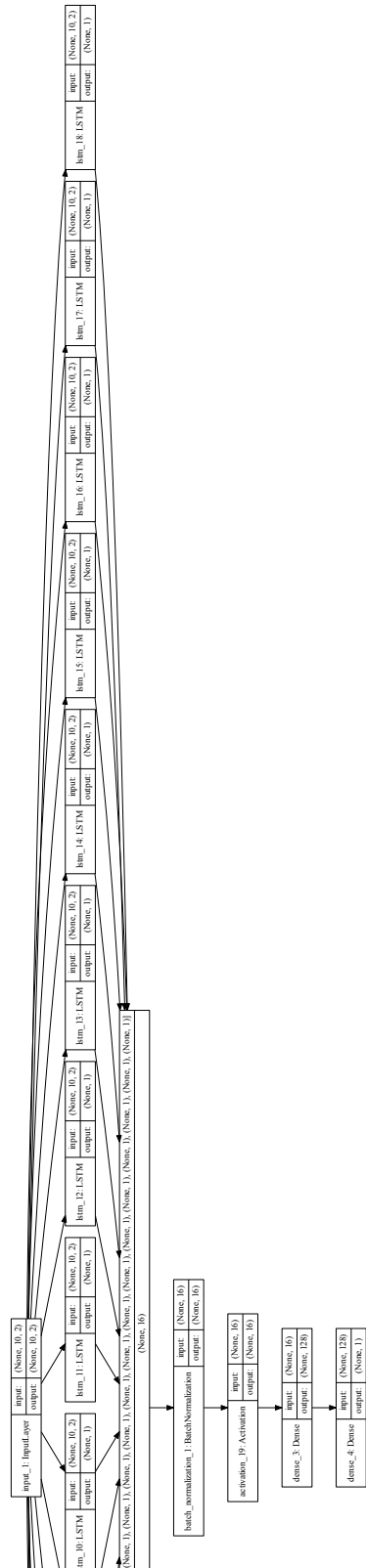


(b) Right Part

Figure A.2: Hybrid LSTM-Wavenet Model by API Structure, Configuration 2 (NAS-DAQ): Model Structure



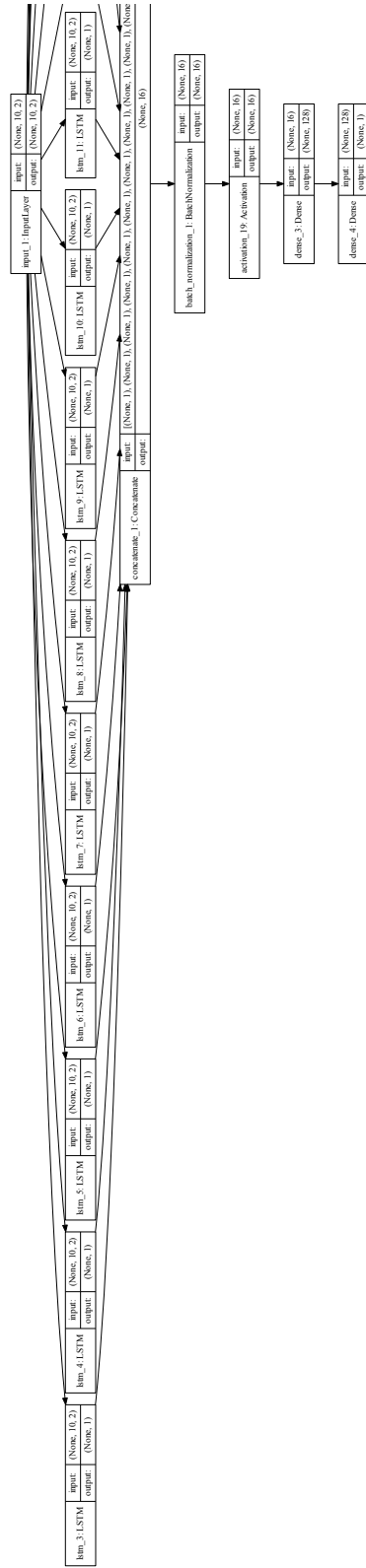
(a) Left Part



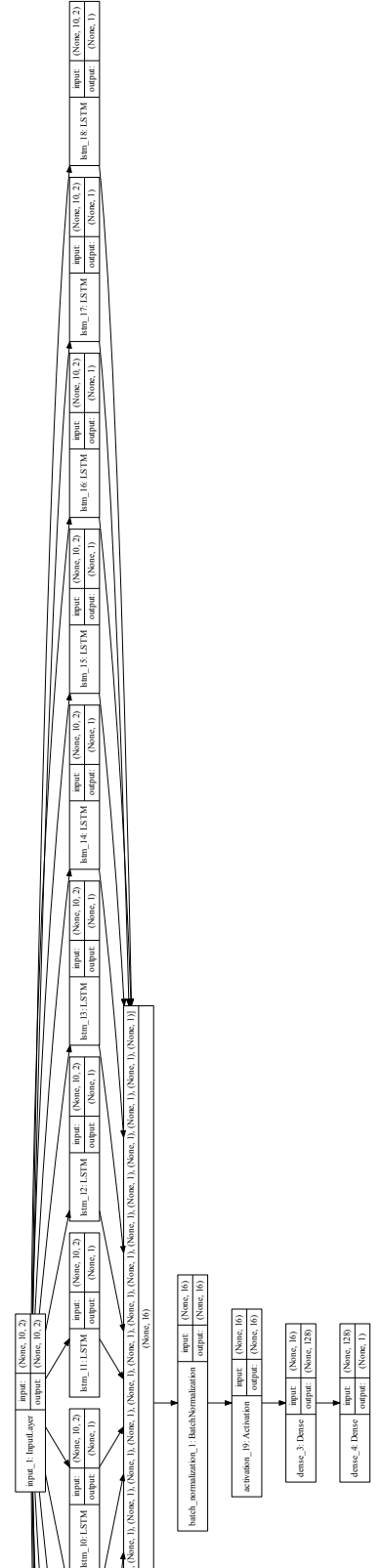
(b) Right Part

Figure A.3: Hybrid LSTM-Wavenet Model+MRA by API Structure, Configuration 2 (S&P500): Model Structure of the Approximation





(a) Left Part



(b) Right Part

Figure A.4: Hybrid LSTM-Wavenet Model+MRA by API Structure, Configuration 2 (NASDAQ): Model Structure of the Approximation



# CURRICULUM VITAE

## PERSONAL INFORMATION

**Surname, Name:** Kılıç, Deniz Kenan  
**Nationality:** Turkish (TC)  
**Date and Place of Birth:** 22.04.1989, Ankara  
**Phone:** +90 536 6847753

## EDUCATION

Degree	Institution	Year of Graduation
M.S. in Financial Mathematics	Middle East Technical University	2015
B.S. in Mathematics	Middle East Technical University	2012

## PROFESSIONAL EXPERIENCE

Year	Place	Enrollment
March 2021-Ongoing	VLMedia	R&D Software Engineer
June 2016-March 2021	ESEN	Research and Development Engineer
March 2015-June 2016	Key Yazılım	Business Development Specialist
February 2013-February 2015	Gate Elektronik	Research and Development Engineer

## PUBLICATIONS

### Thesis

Kılıç, D.K. Multiresolution Analysis of S&P500 Time Series. Master's thesis, Middle East Technical University, 2015.

### International Publications

Kılıç, D.K. & Uğur, Ö. Multiresolution Analysis of S&P500 Time Series, Springer, Annals of Operations Research (2018) 260(1-2): 197-216. DOI:/10.1007/s10479-016-2215-3

## CONFERENCE TALKS

- Kılıç, D.K. & Uğur, Ö. Multiresolution Analysis of S&P500 Time Series, in: International Congress on Industrial and Applied Mathematics (ICIAM 2019), Valencia, Spain, 2019.
- Kılıç, D.K. & Uğur, Ö. Hybrid Wavelet-Neural Network Models for Multivariate Time Series, in: BEYOND 2019: Computational Science and Engineering Conference, Ankara, Turkey, 2019.
- Kılıç, D.K. Multiresolution Analysis of S&P500 Time Series, in: 3rd Ankara-Istanbul Workshop on Stochastic Processes, Ankara, Turkey, 2016.
- Kılıç, D.K. & Uğur, Ö. Wavelet Analysis of Financial Time Series: a case study on S&P500, in: European Conference on Numerical Mathematics and Advanced Applications (ENUMATH), Ankara, Turkey, 2015.
- Kılıç, D.K. & Uğur, Ö. Wavelet Analysis of Financial Time Series: a case study on S&P500, in: 55th Meeting of the EURO Working Group for Commodities and Financial Modelling (EWGCFM), Ankara, Turkey, 2015.