

# Machine Learning-Driven Demand Prediction to Minimise Supply Chain Costs

**Author:**

Ahmed, Supriyo

**Publication Date:**

2023

**DOI:**

<https://doi.org/10.26190/unsworks/25303>

**License:**

<https://creativecommons.org/licenses/by/4.0/>

Link to license to see what you are allowed to do with this resource.

Downloaded from <http://hdl.handle.net/1959.4/101596> in <https://unswworks.unsw.edu.au> on 2025-05-21

# **Machine Learning-Driven Demand Prediction to Minimise Supply Chain Costs**

**Supriyo Shafkat Ahmed**

A thesis in fulfilment of the requirements for the degree of  
**Doctor of Philosophy**



**UNSW**  
**S Y D N E Y**

School of Engineering and Information Technology

Faculty of Engineering

The University of New South Wales

October 31, 2023

## ▲ Candidate's Responses

Submit your corrected thesis and responses here

The candidate's responses to examiners' comments and corrected thesis are displayed in the tabs below.

Thesis Title and Abstract	Declarations	Inclusion of Publications Statement	Corrected Thesis and Responses
---------------------------	--------------	-------------------------------------	--------------------------------

### ORIGINALITY STATEMENT

I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where acknowledgement is made in the thesis. Any contribution made to the research by others, with whom I have worked at UNSW or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.

### COPYRIGHT STATEMENT

I hereby grant the University of New South Wales or its agents a non-exclusive licence to archive and to make available (including to members of the public) my thesis or dissertation in whole or part in the University libraries in all forms of media, now or here after known. I acknowledge that I retain all intellectual property rights which subsist in my thesis or dissertation, such as copyright and patent rights, subject to applicable law. I also retain the right to use all or part of my thesis or dissertation in future works (such as articles or books).

For any substantial portions of copyright material used in this thesis, written permission for use has been obtained, or the copyright material is removed from the final public version of the thesis.

### AUTHENTICITY STATEMENT

I certify that the Library deposit digital copy is a direct equivalent of the final officially approved version of my thesis.

## ▲ Candidate's Responses

Submit your corrected thesis and responses here

The candidate's responses to examiners' comments and corrected thesis are displayed in the tabs below.

Thesis Title and Abstract	Declarations	Inclusion of Publications Statement	Corrected Thesis and Responses
---------------------------	--------------	-------------------------------------	--------------------------------

UNSW is supportive of candidates publishing their research results during their candidature as detailed in the UNSW Thesis Examination Procedure.

Publications can be used in your thesis in lieu of a Chapter provided:

- You contributed **greater than 50%** of the content in the publication and are the "primary author", i.e. you were responsible primarily for the planning, execution and preparation of the work for publication.
- You have approval to include the publication in their thesis in lieu of a Chapter from your Supervisor and Postgraduate Coordinator.
- The publication is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in the thesis.

**Some of the work described in my thesis has been published and it has been documented in the relevant Chapters with acknowledgement.**

**A short statement on where this work appears in the thesis and how this work is acknowledged within chapter/s:**

The findings from the paper titled "Poly-linear regression with augmented long short term memory neural network: Predicting time series data," published in the Journal of Information Sciences, are presented in Chapter 2 of this thesis. In Chapter 3, parts of the research are based on the article titled "A switching based forecasting approach (SBFA) for forecasting sales data in supply chains," which has been submitted to the Journal Neurocomputing and is currently undergoing the second round of review. Additionally, Chapter 3 also incorporates a similar approach proposed in a book chapter titled "Predictive big data analytics for supply chain demand forecasting" from the book "Computational Intelligence Techniques for Sustainable Supply Chain Management," published by ELSEVIER, which is currently under review. Lastly, Chapter 4 contains the results of another journal paper titled "A multi-source multi-layer-based transfer learning approach for forecasting customer demands of newly launched products." submitted in Expert Systems with Applications and is currently under review. It is important to note that proper acknowledgement of all the work by other authors of these papers has been provided at the beginning of each chapter, as well as in the Acknowledgements section of this thesis.

### CANDIDATE'S DECLARATION

I declare that I have complied with the Thesis Examination Procedure.



**THE UNIVERSITY OF NEW SOUTH WALES**  
**Thesis/Dissertation Sheet**

Surname or Family name: **Ahmed**

First name: **Supriyo Shafkat** Other name/s:

Abbreviation for degree as given in the University calendar: **PhD**

School: **School of Engineering and Information Technology**

Faculty: **Faculty of Engineering**

Title: Machine Learning-Driven Demand Prediction to Minimise Supply Chain Costs

**Abstract**

The application of artificial intelligence (AI), machine learning (ML), and data science technologies has gained significant traction in recent years, particularly within the supply chain sector. These technologies aim to enhance decision-making processes while modernizing the existing methods while acknowledging their historical significance and contributions. One notable application involves predictive analytics, such as time series forecasting, which plays a crucial role in identifying anomalies and predicting real-time movements in financial markets. In this thesis, a novel approach called poly-linear regression and Long Short-Term Memory (LSTM) with data augmentation (PLR-ALSTM-NN) is introduced. This approach represents the initial contribution of this research and is designed to provide more accurate and unbiased predictions for financial markets and supply chain time-series data. While PLR-ALSTM-NN excels in long-term prediction, a switching-based forecasting approach (SBFA) is also proposed that offers improved accuracy by considering various forecasting horizons. The SBFA system leverages different ML/DL techniques for different prediction timeframes, selecting the most suitable approach based on validation data performance. The AI-based SBFA approach has been successfully applied to forecast demand data in supply chains, contributing to a reduction in cost estimation errors when compared to traditional forecasting methods. This represents the second significant contribution of this thesis to the field of knowledge.

In the field of supply chain management, accurately predicting the demand for newly launched products can be difficult due to a lack of data. Traditional methods, such as market surveys and examining the behavior of similar products, are often inaccurate and can lead to incorrect cost estimates for a business. However the use of advanced AI techniques, such as Transfer Learning, can improve the accuracy of demand forecasting for new products by using historical data from similar products. This thesis also proposes a Multi-source Multi-layer based Transfer Learning approach with Recursive Feature Elimination (MSML-TL-RFE) to extract knowledge from various sources to predict demand for new products or stores in the market. This approach is tested using a supply chain model with different attributes, and reorder points are calculated based on the predictions of various Transfer Learning approaches. The overall cost of the supply chain is used to evaluate the effectiveness and applicability of the proposed MSML-TL-RFE when compared to traditional forecasting methods. The performance of MSML-TL-RFE, along with the development of a hierarchical supply chain model for cost analysis, makes the final thesis' contributing material.

It can be concluded from this thesis that using deep learning techniques, such as PLR-ALSTM-NN and SBFA, with a large amount of data are effective approaches for predicting demand in a supply chain. When data is limited, Transfer Learning can be used to make more accurate predictions of the demand for newly launched products. The use of these techniques can lead to cost minimisation and increased customer satisfaction, among other benefits. It is important to carefully evaluate the available data and choose the most appropriate approach for demand forecasting, in order to achieve related desired outcomes.

**Declaration relating to disposition of project thesis/dissertation**

I hereby grant the University of New South Wales or its agents a non-exclusive licence to archive and to make available (including to members of the public) my thesis or dissertation in whole or part in the University libraries in all forms of media, now or here after known. I acknowledge that I retain all intellectual property rights which subsist in my thesis or dissertation, such as copyright and patent rights, subject to applicable law. I also retain the right to use all or part of my thesis or dissertation in future works (such as articles or books).

For any substantial portions of copyright material used in this thesis, written permission for use has been obtained, or the copyright material is removed from the final public version of the thesis.

Signature **Supriyo Shafkat Ahmed**

Witness

Date **October 31, 2023**

**FOR OFFICE USE ONLY**

Date of completion of requirements for Award



# Dedication

To

*My Beloved Family*

*My Son Sharaaf Shafkat Ahmed*

*My Wife Sanjida Hossain Sabah*

*My Mother Shahana Khanam*

*My Father Aziz Ahmed*

# Abstract

The application of artificial intelligence (AI), machine learning (ML), and data science technologies has gained significant traction in recent years, particularly within the supply chain sector. These technologies aim to enhance decision-making processes while modernizing the existing methods while acknowledging their historical significance and contributions. One notable application involves predictive analytics, such as time series forecasting, which plays a crucial role in identifying anomalies and predicting real-time movements in financial markets. In this thesis, a novel approach called poly-linear regression and Long Short-Term Memory (LSTM) with data augmentation (PLR-ALSTM-NN) is introduced. This approach represents the initial contribution of this research and is designed to provide more accurate and unbiased predictions for financial markets and supply chain time-series data. While PLR-ALSTM-NN excels in long-term prediction, a switching-based forecasting approach (SBFA) is also proposed that offers improved accuracy by considering various forecasting horizons. The SBFA system leverages different ML/DL techniques for different prediction timeframes, selecting the most suitable approach based on validation data performance. The AI-based SBFA approach has been successfully applied to forecast demand data in supply chains, contributing to a reduction in cost estimation errors when compared to traditional forecasting methods. This represents the second significant contribution of this thesis to the field of knowledge.

In the field of supply chain management, accurately predicting the demand for newly launched products can be difficult due to a lack of data. Traditional methods, such as market surveys and examining the behavior of similar products, are often inaccurate and can lead to incorrect cost estimates for a business. However the use of advanced AI techniques, such as Transfer Learning, can improve the accuracy of demand forecasting for new products by using historical data from similar products. This thesis also proposes a Multi-source Multi-layer based Transfer Learning approach with Recursive Feature Elimination (MSML-TL-RFE) to extract knowledge from various sources to predict demand for new products or stores in the market. This approach is tested using a supply chain model with different attributes, and reorder points are calculated based on the predictions of various Transfer Learning approaches. The overall cost of the supply chain is used to evaluate the effectiveness and applicability of the proposed MSML-TL-RFE when compared to traditional forecasting methods. The performance of MSML-TL-RFE, along with the development of a hierarchical supply chain model for cost analysis, makes the final thesis' contributing material.

It can be concluded from this thesis that using deep learning techniques, such as PLR-ALSTM-NN and SBFA, with a large amount of data are effective approaches for predicting demand in a supply chain. When data is limited, Transfer Learning can be used to make more accurate predictions of the demand for newly launched products. The use of these techniques can lead to cost minimisation and increased customer satisfaction, among other benefits. It is important to carefully evaluate the available data and choose the most appropriate approach for demand forecasting, in order to achieve related desired outcomes.

# Acknowledgement

The fact that I am still alive and healthy enough to finish this thesis while the entire world is experiencing a pandemic due to the COVID-19 outbreak is a huge mercy from Almighty Allah. This thesis' completion has been the most difficult task I've ever undertaken in my life. In addition to the time and effort I put into strengthening my problem-solving abilities with the accomplishment the research milestones, I will always be appreciative of the tremendous mental support and inspiration from a small group of passionate people in my life.

First of all, I want to express my deep gratitude to Dr. Ripon Kumar Chakrabortty, one of my direct supervisors, with a great personality and a thorough gentleman. I was really very amazed by his composure as he patiently guided me while I was having difficulty in finding research gaps. He deserves extra praise for reading all of my disorganised, haphazard manuscripts and rewriting them into something more meaningful. Additionally, I want to thank him for his continuous mental support that helped me overcome the stress, anxieties and hardships that personally affected me throughout the lock-downs.

I am equally thankful to my principle supervisor Dr. Daryl L. Essam for being my mentor throughout my PhD journey. The only reason this thesis was finally completed was because of his encouragement in igniting my inner spirit and continuous motivation that helped keeping me on the right path. The early conversations we had in his office convinced me to consider more carefully about a brand-new discipline of machine learning and how it is applied to supply chain, both of which I knew very little about before. I was greatly motivated and encouraged by his consistent direction, support, patience, encouragement, and vast knowledge.

After that, I must thank UNSW at ADFA for providing me the tuition fee scholarship (TFS) that enabled me to cover my living expenses during my stay in Australia. I was very fortunate to spend quality time with a bunch of fellow students throughout the entire campus. My sincere gratitude goes out to all of my friends and brothers, especially Kamrul Hassan Rahi, Syed Mamun R Rashid, Nastaran Tui, Nishat Falguni, Kutub Rana, Jyoti, Lamia, Tarun Biswas, Shahed Mahmud, Joni, Sanowar, Shuvam, Ahmed Shama, Himel, Raju, Kaiser Ahmed, Kamrul Panna, Rasel, Dr. Mohiuddin Mamun, Dr. Kamrul Majumder, Dr. Asaduzzaman Khan, Dr. Ahsanul Haque, Dr. Ahsanul Habib, Dr. Tanvir Ishtiaque, Toufiq Hossain, Atiqur Rahman Chopol, Rajin, Sohrab, Sakib Omi, Koly,

Anisur Rahman Jony, Savah Rahman, Shawon Bosw, Priyanka Rani Bose, Saiket Bose and many more. I have spent countless hours with them talking about topics that cross disciplines, having fun, playing cricket, riding bicycles, and carrying out other activities.

Despite the fact that borders were closed for an unknown amount of time and I was miserable as I said goodbye to my family in Bangladesh and didn't know when I would see them again, the people of Canberra treated me incredibly kindly. Dr. Mamun and Ms. Rizwana Era showed me exceptional hospitality by surprising me with sudden treats. Additionally, I appreciate Sanowar and his family for inviting me to dinner on numerous occasions. The same appreciation goes for Ahsanul and his family, Shahed and his family, Tarun and his family, Joni and his family, Kaiser and his family, for treating me as their family members. In addition, I personally thank Shahed and his family for planning impromptu excursions and connecting me with them so I could refresh my mind. Additionally, the dinner and movie night at Ahsanul's new home will live long in my memory. Many thanks to Tarun for having coffee with me every evening, which was among the regular mind refreshing activities during hard lockdowns.

I would like to take this opportunity to thank my previous supervisor, Dr. Tarem Ahmed, who actually strengthened my basic skill of research. He has been supervising me since my undergraduate course and also my masters. I had numerous publications with him, a great researcher, a true mentor, advisor and a wonderful human being. I still sought Dr. Tarem's advice while I was going through difficult times in my PhD. I am forever grateful to Dr. Tarem for keeping me inspired despite my ups and downs.

I would also like to express my heartfelt gratitude to the esteemed colleagues and students of BRAC University who have played a significant role in shaping my academic journey. I am indebted to Prof. Md. Mosaddequr Rahman, Prof. A.K.M Abdul Malek Azad, Dr. Belal Bhuiyan, Dr. Amina Hasan Abedin, Dr. Sadia Hamid Kazi, Ms. Farzana Shabnam, Md. Shamsul Kaonain, Rumana Rahman, Ms. Afroza Bagum, Aldrin Nippon Bobby, Arif Shakil, and many more dedicated colleagues whose guidance, support, and expertise have been invaluable throughout my time at the university. I would also like to acknowledge the exceptional contributions of Ashique Mohammad Sadique, Towfiqur Rahman, Antara Fairooz Khan, Ashique Sikder, Rakin Bin Rabbani, Akibur Rahman Choton, Shahrin Manzoor, Farana Naz Tultul, Sadia Nodi, Imam Ush Shaheed, Farah Nazifa and numerous other talented students whose collaborations and interactions have enriched my learning experience. I am grateful for the enduring connections and friendships forged during my tenure at BRAC University.

Last but not least, acknowledging my family members, without whom I would not truly exist, brings me the utmost satisfaction. No matter what happened in my life, I always found them by my side. My late father, Aziz Ahmed, always encouraged me to pursue higher education, and he hoped that one day his son would earn a PhD degree. My mother, Shahana Khanam, a true fighter who sacrificed a lot just to provide me a better life. I spent two and half miserable years without seeing my son, Sharaaf Shafkat Ahmed, my own flesh and blood, my ultimate source of happiness. During that time, my beloved wife, Sanjida Hossain Sabah, took care of my son so that I can concentrate more on my

studies. During the most challenging period of the pandemic, when my mental health was affected by the strict lockdown measures, my in-laws provided me with invaluable support and encouragement throughout my PhD tenure. I am incredibly grateful to my father-in-law, Kabir Hossain, my mother-in-law, Syeda Shahin Hossain, my sister-in-law, Sabrina Hossain Sarah, my uncle-in-law, Nasim Hoq, and my aunt-in-law, Tamanna Nasim, for their unwavering support during that time. My other family members: uncles, aunts and grandparents, greatly appreciate the effort I make to do better in every area of my life. Finally, without any doubt, I would like to dedicate this thesis to the most important people in my life: my mother, my late father, my wife and my son. It is for their unconditional love, support and prayers for me that I have managed to achieve this, this far in my life.

# List of Publications

## Journal Papers

Supriyo Ahmed, Ripon K Chakrabortty, Daryl L Essam, and Weiping Ding. Poly-linear regression with augmented long short term memory neural network: Predicting time series data. *Information Sciences*, 2022.

Supriyo Ahmed, Ripon K Chakrabortty, Daryl L Essam, and Weiping Ding. A switching based forecasting approach (SBFA) for forecasting sales data in supply chains. *Neurocomputing*, Under review 2nd Round, submitted Oct. 2022.

Supriyo Ahmed, Ripon K Chakrabortty, Daryl L Essam, and Weiping Ding. A multi-source multi-layer-based transfer learning approach for forecasting customer demands of newly launched products. *Expert Systems with Applications*, Under review, submitted July. 2023.

## Book Chapter

Supriyo Ahmed, Ripon K Chakrabortty, and Daryl L Essam. Predictive big data analytics for supply chain demand forecasting. In *Computational Intelligence Techniques for Sustainable Supply Chain Management*. ELSEVIER, Under review, submitted May. 2023.



# Contents

<b>Abstract</b>	<b>iv</b>
<b>Acknowledgement</b>	<b>vi</b>
<b>List of Publications</b>	<b>ix</b>
<b>Journal Papers</b>	<b>ix</b>
<b>Book Chapter</b>	<b>ix</b>
<b>Contents</b>	<b>x</b>
<b>List of Figures</b>	<b>xvi</b>
<b>List of Tables</b>	<b>xix</b>
<b>List of Datasets Used in the Thesis</b>	<b>xxiii</b>
<b>List of Important Acronyms/Parameters/Variables</b>	<b>xxiv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background and Problem Description . . . . .	1
1.2 Definition of Some Common Time series Predictive Techniques and their Pros and Cons . . . . .	5

1.3	Motivation and Scope of Research . . . . .	8
1.4	Brief Literature Review . . . . .	10
1.4.1	Traditional, ML and DL approaches to predict sales data . . . . .	11
1.4.2	Hybrid Prediction Models . . . . .	13
1.4.3	Application of TL in Predicting Sales Data for Newly Launched Products . . . . .	14
1.5	Research Gaps And Possible Solutions . . . . .	16
1.5.1	Absence of Models for Long Term Forecasting with an abundance of data . . . . .	16
1.5.2	Scarcity of Hybrid Models that are Independent of Data Characteristics and make more Accurate Predictions . . . . .	17
1.5.3	Absence of Forecasting Models with Limited Data/Information . . . . .	18
1.6	Research Objectives . . . . .	19
1.6.1	An Overview of Expectations from Practices . . . . .	23
1.7	Contributions of Thesis . . . . .	23
1.7.1	Efficient and Effective Prediction Model for Long-term Time-series Data . . . . .	24
1.7.2	A Robust Hybrid Predictive Model for Accurate Sales Forecasting Across Various Time Horizons . . . . .	25
1.7.3	A Highly Effective and Robust TL Model That Leverages Multiple Similar Sources of Data for Optimal Sales Prediction of New Market Products . . . . .	26
1.8	Organisation of Thesis . . . . .	27
<b>2</b>	<b>Poly-linear Regression with Augmented Long Short Term Memory Neural Network: Predicting Time Series Data</b>	<b>30</b>
2.1	Introduction . . . . .	31
2.1.1	Arguments for Utilising Stock Market Data for Analysis and Prediction in the Context of Supply Chain Sales Forecasting . . . . .	34
2.2	Related Work . . . . .	37

2.2.1	Contribution of ML/DL/Hybrid Forecasting Approaches to Financial Market . . . . .	37
2.2.2	Drawbacks and Limitations of Existing Research . . . . .	40
2.2.3	Summary of Research Gaps and Possible Solutions . . . . .	41
2.3	Proposed Methodology . . . . .	42
2.3.1	Data Collection . . . . .	44
2.3.2	Regression . . . . .	45
2.3.3	Data Split . . . . .	46
2.3.4	Data Augmentation . . . . .	48
2.3.5	LSTM Architecture . . . . .	49
2.3.6	Data Structuring and Train the LSTM Network . . . . .	52
2.3.7	Testing the Trained Network . . . . .	53
2.4	Result Analysis and Model Validation . . . . .	56
2.4.1	Parameter Tuning . . . . .	56
2.4.2	Model validation on Dataset 1.(a) . . . . .	59
2.4.3	Model validation on recent S&P500 (Dataset 1.(b) and Dataset 1.(c))	63
2.4.4	Model Validation on Dataset 2 . . . . .	67
2.4.5	Model Validation on Dataset 5 . . . . .	71
2.5	Application of the Model to Predict Sales Data and Closing Price of DJI Data . . . . .	72
2.5.1	Predicting Time Series Sales Data . . . . .	72
2.5.2	Predicting the Closing Price of Dow Jones Industrial Average (DJI)	74
2.6	Statistical Significance Test . . . . .	78
2.7	Discussion . . . . .	80
2.8	Managerial Implications: Practitioners' Benefit from Long-Term Forecasting	81
2.9	Summary . . . . .	82

<b>3 A Switching Based Forecasting Approach (SBFA) for Forecasting Sales Data in Supply Chains</b>	<b>84</b>
3.1 Introduction . . . . .	85
3.2 Literature Review . . . . .	89
3.2.1 Research Gap . . . . .	93
3.3 Brief Introduction to Forecasting Methods . . . . .	94
3.3.1 Traditional Forecasting Technique . . . . .	94
3.3.2 ML/DL approaches . . . . .	96
3.4 The Proposed Switching Based Forecasting Approach (SBFA) . . . . .	108
3.5 The Considered Supply Chain Model . . . . .	110
3.5.1 Forecasting Sales Data by Traditional Forecasting Approaches . . .	113
3.6 Experimental Design For the Proposed SBFA . . . . .	114
3.6.1 Data Collection . . . . .	115
3.6.2 Cross Validation . . . . .	119
3.7 Result Analyses for the Proposed SBFA . . . . .	120
3.7.1 A comparative Analysis of SBFA with existing ML/DL Approaches	120
3.7.2 Statistical Analysis . . . . .	122
3.7.3 Application of SBFA to the Designed SC Model . . . . .	124
3.7.4 Behavioural/Sensitivity Analysis . . . . .	131
3.8 Theoretical Derivation . . . . .	133
3.8.1 Selection of Different Approaches on the Basis of Prediction Horizon	133
3.8.2 Selection of Different Approaches on the Basis of Accuracy and Time	136
3.8.3 Selection of Different Approaches based on Supply Chain Cost and Time . . . . .	137
3.9 Managerial Implications: Practitioners' Benefit from SBFA . . . . .	138
3.10 Summary . . . . .	139

<b>4 A Multi-Source Multi-Layer-based Transfer Learning Approach for Forecasting Customer Demands of Newly Launched Products</b>	<b>141</b>
4.1 Introduction . . . . .	142
4.2 Literature Review . . . . .	148
4.2.1 Research Gaps . . . . .	152
4.3 Problem Definition . . . . .	153
4.4 Proposed Methodology . . . . .	158
4.4.1 Simple CNN Network: No-TL Approach . . . . .	159
4.4.2 Single Source TL: SS-TL . . . . .	160
4.4.3 Multi Source TL Approaches . . . . .	161
4.5 Model Implementation . . . . .	168
4.5.1 Data Collection . . . . .	169
4.5.2 Feature Extraction . . . . .	172
4.5.3 Data Structuring . . . . .	173
4.5.4 Experimental Design for TL Approaches . . . . .	175
4.5.5 Effectiveness of the TL Predictions in the context of a Hierarchical SC Model . . . . .	181
4.6 Result Analyses . . . . .	181
4.6.1 Result Analyses using Online Datasets . . . . .	182
4.6.2 Application of TL on the Designed Hierarchical SC Model . . . . .	186
4.6.3 Behavioural/Sensitivity Analysis . . . . .	189
4.6.4 Theoretical Derivation . . . . .	192
4.6.5 Managerial Implications . . . . .	194
4.7 Summary . . . . .	194

<b>5 Conclusion and Future Directions</b>	<b>196</b>
5.1 Summary of Accomplished Research . . . . .	196
5.1.1 Efficient and Effective Predictions of Long-Term Time Series Data .	197
5.1.2 A Robust Hybrid Approach to Predict Sales Data over Varying Forecasting Horizons to Optimise Supply Chain Cost . . . . .	198
5.1.3 A TL Approach to Maximise Knowledge Transfer Process with Limited Available Data . . . . .	199
5.2 Limitations of This Thesis . . . . .	200
5.3 Future Research Directions . . . . .	202
5.3.1 Future Work Related to Long-Term Forecasting Approach . . . . .	202
5.3.2 Future Work Related to SBFA . . . . .	203
5.3.3 Future Work Related to TL Approach . . . . .	203
5.3.4 Reflection on the Use of Predictive Models in Pandemic Prediction and Management . . . . .	204
<b>References</b>	<b>206</b>



# List of Figures

1.1	A typical forward flow supply chain network . . . . .	1
1.2	A visual representation of SC Model . . . . .	2
1.3	The number of research articles on ML or DL approaches for making Supply chain demand predictions since 2008 (Source: Scopus) . . . . .	9
1.4	The Organisation of this Thesis . . . . .	28
2.1	Proposed Model . . . . .	43
2.2	LSTM's architecture [Olah, 2015a] . . . . .	49
2.3	Illustration of short-term and long-term prediction . . . . .	52
2.4	Factor level trend for each parameter. . . . .	58
2.5	Average Closing Price for Short Term Prediction for Each Network on Extended S&P500 dataset (Dataset 1.(b)) . . . . .	65
2.6	Average Closing Price for Long Term Prediction for Each Network on Extended S&P500 dataset (Dataset 1.(b)) . . . . .	65
2.7	Short Term Prediction by using the proposed approach using Dataset 2 . .	68
2.8	3 Days Ahead Prediction by using the proposed approach using Dataset 2 .	69
2.9	5 Days Ahead Prediction by using the proposed approach using Dataset 2 .	70
2.10	7 Days Ahead Prediction by using the proposed approach using Dataset 2 .	71
2.11	The best and average Predicted Sales (Dataset 3) for 1 Week Ahead . . . .	74
2.12	Short Predictions of the closing prices of DJI Stock index (Dataset 4.(a)) using different available networks with respect to the true values . . . . .	75

2.13 Long Term Predictions of the closing prices of DJI Stock index (Dataset 4.(a)) using different available networks with respect to the true values . . . . .	76
3.1 Establishing statistical histograms . . . . .	98
3.2 Leaf-wise growth strategy . . . . .	99
3.3 Folded and unfolded architecture of a 2 -layer RNN . . . . .	100
3.4 RNN cell structure [Olah, 2015b] . . . . .	101
3.5 LSTM's architecture [Olah, 2015b] . . . . .	102
3.6 Gated Recurrent Unit (GRU) [Olah, 2015b] . . . . .	103
3.7 One dimensional CNN [Jin et al., 2020] . . . . .	104
3.8 Classical Transformer Encoder Architecture [Jin et al., 2021] . . . . .	106
3.9 Switching Based Forecasting Approach: SBFA . . . . .	110
3.10 Extraction of Features From Data . . . . .	116
3.11 Impact of Forecasting Approaches on Supply Chain Cost with Static Lead Time and Different Levels of Safety Stock Based on Expt 1 Rossmann dataset	127
3.12 Impact of Forecasting Approaches on Supply Chain Cost with Dynamic Lead Time and Different Levels of Safety Stock Based on Expt 1 Rossmann dataset . . . . .	130
3.13 Average Cost for Different Entities of Supply Chain on the Kaggle Dataset	131
3.14 Average Cost for Different Entities of Supply Chain on the Rossmann Dataset	132
3.15 Percentage Contribution of Different Approaches for Proposed Switching Based Technique on Combined Dataset . . . . .	135
3.16 Performance Improvement vs Time: Kaggle dataset . . . . .	136
3.17 Performance Improvement vs Time: Rossmann dataset . . . . .	137
3.18 SC Cost vs Time: Kaggle Daataset . . . . .	138
4.1 Supply Chain Information Sharing Concept . . . . .	154
4.2 Overall flowchart of the MSML-TL-RFE . . . . .	159
4.3 One dimensional CNN (No-TL Approach) [Jin et al., 2020] . . . . .	160

4.4	Single Source TL Network (SS-TL) . . . . .	161
4.5	Multi-Source Weighted Average TL Network (MSWA-TL) . . . . .	163
4.6	Multi-source Switching Based TL Network (MSSB-TL) . . . . .	164
4.7	Multi-source Multi-Layer TL Network (MSML-TL) . . . . .	166
4.8	Multi-source Multi-Layer TL Network with RFE (MSML-TL-RFE) . . . . .	167
4.9	Extraction of Features From Data using Date-Time Feature Extraction Method . . . . .	173
4.10	Illustration of short-term and mid-term prediction . . . . .	174
4.11	Prediction of Newly Launched Without Info Sharing Using Different TL Approaches . . . . .	179
4.12	Average RMSE for Different Approaches using 3 Datasets, with and without Information Sharing . . . . .	183
4.13	Average RMSE of Different Approaches, with and without Information Sharing . . . . .	184
4.14	Supply Chain Cost (\$) over 3 Channels of SC having Fixed LT . . . . .	187
4.15	Supply Chain Cost (\$) vs Training Time vs Prediction Accuracy of Channel 1 of SC having Fixed LT . . . . .	193
4.16	Supply Chain Cost (\$) vs Training Time vs Prediction Accuracy of Channel 1 of SC having Dynamic LT . . . . .	193



# List of Tables

2.1	The different time series observations used in this chapter . . . . .	47
2.2	A statistical description of the dataset . . . . .	47
2.3	Combination of parameter values. . . . .	57
2.4	Orthogonal table with Average RMSE and Standard Deviations. . . . .	57
2.5	Response table for means. . . . .	58
2.6	Average RMSE for each setting using Dataset 1.(a) . . . . .	59
2.7	Average RMSE for Different Independent Approaches implemented on Dataset 1.(a) . . . . .	60
2.8	Evaluation of PLR-ALSTM-NN with respect to ARIMA, GRU, RNN and CNN for short term predictions using Dataset 1.(a) . . . . .	61
2.9	Evaluation of PLR-ALSTM-NN with respect to ARIMA, GRU, RNN and CNN for long term predictions using Dataset 1.(a) . . . . .	61
2.10	Prediction Accuracy of PLR-ALSTM-NN in comparison to the . . . . .	62
2.11	Prediction Accuracy of the Model in comparison to existing literature for Long Term Prediction (Dataset 1.(a)). . . . .	63
2.12	Average RMSE, MAPE and MAE for short term Close Price Prediction of S&P500 (Dataset 1.(b)) . . . . .	64
2.13	Average RMSE, MAPE and MAE for long-term Close Price Prediction of S&P500 (Dataset 1.(b)) . . . . .	64
2.14	Average RMSE, MAPE and MAE for short-term Close Price Prediction of S&P500 (Dataset 1.(c)) . . . . .	66

2.15	Average RMSE, MAPE and MAE for long-term Close Price Prediction of S&P500 (Dataset 1.(c)) . . . . .	66
2.16	Validation of the performance of the proposed model using a secondary dataset (Dataset 2) by comparing MAPE . . . . .	67
2.17	Average RMSE, MAPE, and MAE for different days ahead of forecasting using PLR-ALSTM-NN, ARIMA, GRU, RNN and CNN on Dataset 2 . . . . .	70
2.18	Comparison of performance of PLR-ASLTM-NN for long-term forecasting against ENANFIS and SVR-SVR (Dataset 5) . . . . .	72
2.19	The best, average and the standard deviation of RMSEs for different weeks ahead of sales predictions (Dataset 3) . . . . .	73
2.20	Average RMSE, MAPE and MAE for short-term Close Price forecasting of DJI Data (Dataset 4.(a)) . . . . .	75
2.21	Average RMSE, MAPE and MAE for long-term Close Price forecasting of DJI Data (Dataset 4.(a)) . . . . .	76
2.22	Average RMSE, MAPE and MAE for short-term Close Price forecasting of DJI Data (Dataset 4.(b)) . . . . .	77
2.23	Average RMSE, MAPE and MAE for long-term Close Price forecasting of DJI Data (Dataset 4.(b)) . . . . .	77
2.24	Friedman Test: Mean Rank . . . . .	78
2.25	Wilcoxon Signed Rank Test using Long Term Prediction Errors . . . . .	79
3.1	Extraction of mean, standard deviation, min and max value from the Kaggle Dataset (Subset A and B) . . . . .	95
3.2	Input data for the Supply Chain Model . . . . .	113
3.3	Impact of the Accuracy of Traditional Forecasting Approaches on Supply Chain Cost . . . . .	114
3.4	Data Structuring . . . . .	117
3.5	Hyperparameter Values . . . . .	118
3.6	Subsets of Dataset . . . . .	119
3.7	Experiments Performed using Cross-Validation of dataset . . . . .	119

3.8	Average test RMSE of 10 Runs for 30 days ahead of prediction using Kaggle Dataset . . . . .	120
3.9	Average test RMSE of 10 Runs for 30 days ahead of prediction using Rossmann Dataset . . . . .	121
3.10	Training Time of Different Approaches in seconds . . . . .	122
3.11	Friedman Test: Mean Rank . . . . .	123
3.12	Results of the Wilcoxon test using RMSE-testing . . . . .	124
3.13	SC Cost (Fixed Lead time with variable safety stock) associated with different advanced forecasting approaches using Kaggle Dataset . . . . .	125
3.14	SC Cost (Fixed Lead time with variable safety stock) associated with different advanced forecasting approaches using Rossmann Dataset . . . . .	125
3.15	Differences of SC Cost in terms of Percentage . . . . .	126
3.16	SC Cost (Dynamic Lead time with variable safety stock) associated with different advanced forecasting models using Kaggle Dataset . . . . .	128
3.17	SC Cost (Dynamic Lead time with variable safety stock) associated with different advanced forecasting models using Rossmann Dataset . . . . .	129
3.18	Differences of Dynamic Lead Time SC Cost in terms of Percentage . . . . .	129
3.19	Average Costs for Different Entities of Supply Chain Model . . . . .	131
3.20	Contribution of individual approaches within the proposed SBFA for different days ahead prediction on Combined Dataset . . . . .	134
4.1	Input data for the Supply Chain Model . . . . .	158
4.2	Dataset 1: Duration of Source/Target Domain Data for Training/Validation and Testing the TL Networks . . . . .	171
4.3	Time-Steps of Source/Target Domain Data for Training/Validation and Testing the TL Networks . . . . .	172
4.4	Data Structuring . . . . .	174
4.5	Euclidean Distance and Weights of Closest Three Items/Stores in Relation to Channel 1 Target Item/Stores for 3 datasets . . . . .	177
4.6	Euclidean Distance and Weights of Closest Three Items/Stores in Relation to Target Item/Stores for 3 datasets With Information Sharing . . . . .	180

4.7	Average RMSE for Different Approaches using 3 Datasets, with and without Information Sharing . . . . .	182
4.8	Average RMSE of Different Approaches, with and without Information Sharing . . . . .	183
4.9	Friedman Test: Mean Rank . . . . .	185
4.10	Results of the Wilcoxon test using RMSE-testing . . . . .	186
4.11	SC Cost (\$) for Fixed LT over 3 Channels using Dataset 1 with Information Sharing Concept . . . . .	187
4.12	SC Cost (\$) for Dynamic LT over 3 Channels using Dataset 1 with Information Sharing Concept . . . . .	188
4.13	Time and Accuracy of Different Approaches for Dataset 1 . . . . .	190
4.14	Performance of MSML-TL-RFE Approach with Different Number of Similar Source Items from Same Group . . . . .	190



# List of Datasets Used in the Thesis

- Chapter 2 Dataset 1 S&P500 (till 2021) Section 2.3.1
- Chapter 2 Dataset 2 S&P500 (till 2006) Section 2.3.1
- Chapter 2 Dataset 3 Sales Data Section 2.3.1
- Chapter 2 Dataset 4 DJI Section 2.3.1
- Chapter 2 Dataset 5 SZSE Section 2.3.1
- Chapter 3 Dataset 1 Kaggle Dataset Section 3.6.1
- Chapter 3 Dataset 2 Rossmann Dataset Section 3.6.1
- Chapter 4 Dataset 1 Kaggle Dataset Section 4.5.1
- Chapter 4 Dataset 2 Pasta Sales Dataset Section 4.5.1
- Chapter 4 Dataset 3 Rossmann Dataset Section 4.5.1

## Characterization of Dataset

- Group 1: Time Series Data All datasets in this group consist of time series data, which are sequences of data points collected or recorded over time.
- Group 2: Financial Market Data Chapter 2 Dataset 1, 2, 4, and 5 belong to this group. These datasets are focused on financial market data, which typically includes stock prices, trading volumes, and related financial indicators.
- Group 3: Supply Chain Data Chapter 2 Dataset 3, Chapter 3 Dataset 1 and 2, and Chapter 4 Dataset 1, 2, and 3 are categorized as supply chain data. Supply chain data involves sales of products or goods within a supply chain network over a time period.
- Group 4: Sales Information for Knowledge Adaptation Chapter 4 Dataset 1, 2, and 3 fall into this group. These datasets contain sales information for multiple products and stores, and they are used for knowledge adaptation strategies, specifically for improving sales predictions for new market products based on similar existing products.



# List of Important Acronyms/Parameters/Variables

$\bar{d}$	Average Predicted Demand;
$\bar{L}$	Average Lead Time
$\hat{y}$	Predicted Value
$\odot$	point-wise product
$\sigma_d$	Standard Deviation of Predicted Demand
$\sigma_L$	Standard Deviation of Lead time
$c_t$	memory cell
$f_t$	forget gate
$G, V$	weights of the connections between hidden layers and between the hidden layer and the output layer respectively
$h_t$	hidden state
$h_t$	new state
$h_{t-1}$	previous state
$i_t$	input gate
$n$	Number of Observations
$o_t$	output gate
$r$	reset gate
$tanh:$	activation function
$W_{hh}$	own weights of the cell
$W_{xh}$	weights of the input layer to the hidden layer

$x_t$	current input
$x_{norm}$	normalised data
$y_i$	True Value
$y_t$	current output
$z$	update gate
Adam	Adaptive Moment Estimation
AI	Artificial Intelligence
ANN	Artificial Neural Network
AR	Auto-Regressive
ARIMA	Autoregressive Integrated Moving Average
ARIMAX	Autoregressive Integrated Moving Average with Explanatory Variable
BiLSTM	Bidirectional Long Short Term Memory
CNN	Convolutional Neural Network
$D_S$	Source Domain
$D_T$	Target Domain
DJI	Dow Jones Industrial Average
DL	Deep Learning
DLSTM-AE	Deep LSTM autoencoder
DNN	Deep Neural Network
DOE	Design of Experiments
DTB	Decision Tree Based Approaches
DTr-CNN	Deep Transfer Learning method using CNN
DTW	Dynamic Time Wrapping
ENANFIS	Ensemble Adaptive Neuro-Fuzzy Inference System
EOQ	Economic Order Quantity
GA	Genetic Algorithm
GBM	Gradient Boosting Machine
GCN	Graph Convolutional Layers

GDP	Gross Domestic Product
GNN	Graph Neural Network
GRU	Gated Recurrent Unit
HC	Holding Cost
KNN	K-Nearest Neighbour
LGBM	Light Gradient Boosting Machine
LS	Lost Sales
LSTM	Long Short Term Memory
LT	Lead Time
LTPD	Lead Time Predicted Demand
MA	Moving Average
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
ML	Machine Learning
MLP	Multi-Layer Perceptron
MSML-TL	Multi-Source Multi-Layer TL Approach
MSML-TL-RFE	Multi-source Multi-layer based TL approach with Recursive Feature Elimination
MSSB-TL	Multi-Source Switching Based TL Approach
MSWA-TL	Multi-Source Weighted Average TL Approach
N.Dist	Normal Distribution
NN	Neural Network
NND	Neural Network Disaggregation
OC	Ordering Cost
PC	Purchasing Cost
PCA	Principle Component Analysis
PLR-ALSTM-NN	Poly-linear Regression with Augmented Long Short Term Memory Neural Network

POS	Point-of-Sales
PSO	Particle Swarm Optimisation
RA	Reorder Amount
RF	Random Forest
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
ROP	Reorder Point
SAE	Stacked Autoencoder
SBFA	Switching Based Forecasting Approach
SC	Supply Chain
SES	Simple Exponential Smoothing
SGD	Stochastic Gradient Descent
ss	Safety Stock
SS-TL	Single Source Transfer Learning Approach
SVM	Support Vector Machine
SZSE	Shenzhen Stock Exchange
$T_S$	Source Domain Learning Task
$T_T$	Target Domain Learning Task
TC	Total Cost
TL	Transfer Learning
XGBoost	Extreme Gradient Boost
Z	Number of Standard Deviations Corresponding to Service Level Probability



# Chapter 1

## Introduction

### 1.1 Background and Problem Description

The concept of a supply chain is defined as a group of three or more entities, such as suppliers, manufacturers, distributors, warehouses, third-party service providers, retailers, and customers, through which there are upstream and downstream flows of materials, information, and funds, with the ultimate objective of satisfying customer demand [Myerson, 2023]. Each entity in the chain is involved in tasks including obtaining raw materials, transforming them into partially-finished or finished goods, distributing those goods, providing customer service, and other logistical processes. Figure 1.1 presents the associated stakeholders involved in the links that make up this entire process.

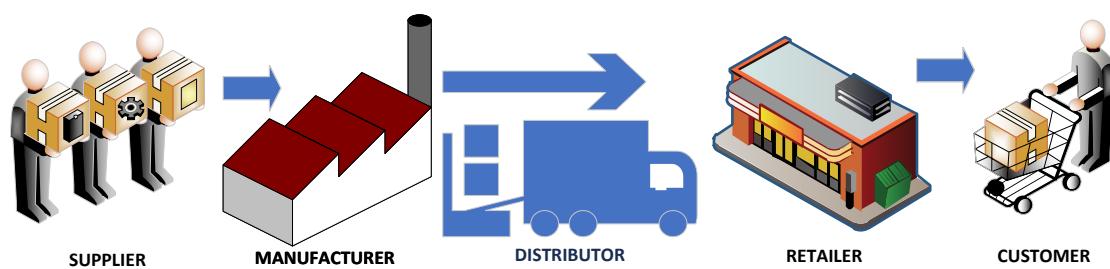


Figure 1.1: A typical forward flow supply chain network

## 1. INTRODUCTION

---

The success of a supply chain system depends on multiple factors, including on-time delivery, inventory management, quality control, sustainability, supply chain visibility, flexibility, and adaptability. In addition to these, forecast accuracy is another key factor, particularly for demand projections [Li and Liu, 2023, Bousqaoui et al., 2017], which can support other factors. For instance, a better forecasting system can ensure on-time delivery, efficient inventory management, and a more flexible and adaptable network. In short, it helps ensure that the right products are available in the right quantities at the right time. One way to demonstrate the importance of forecast accuracy is through the use of a simple supplier-retailer model, as presented in Figure 1.2.

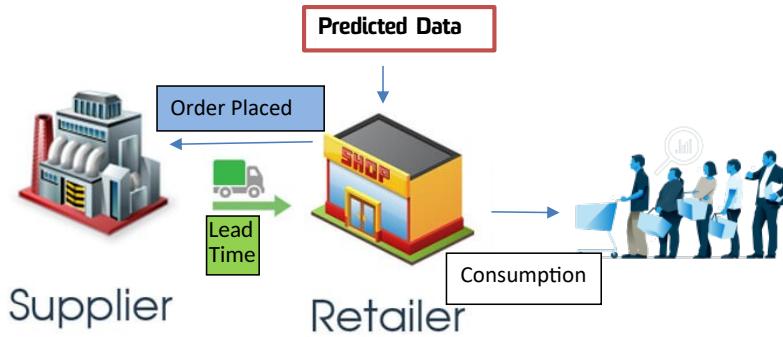


Figure 1.2: A visual representation of SC Model

In this model, a supplier produces a product and a retailer sells it to the end customer. The retailer uses historical demand data to place orders with the supplier, which arrive after a certain amount of lead time. The more accurate the demand predictions, the more efficiently the retailer can manage its inventory, reducing lead time and lowering its operation cost. However this demand prediction is quite challenging due to endogenous (such as seasonal patterns and trends, product life cycle, price changes, marketing, and advertising) and exogenous aspects (such as economic conditions, competition, weather and natural disasters, political and regulatory changes) factors [Cheng et al., 2019]. According to Gilliland [2010], numerous elements, including promotions, the environment, market trends, and the time of year, could affect consumer behaviour and increase demand volatility, which makes prediction challenging. In the retail sector, promotions in particular are a fairly widespread practice that might cause demand instability [López-

---

## *1. INTRODUCTION*

Salazar et al., 2023]. Subsequently, with the advancement of digital technologies where innovative sensors and Point-of-Sales (POS) stations are installed, a large amount of data is continuously being generated. Traditional forecasting methods like Moving Average (MA) and Simple Exponential Smoothing (SES) have limitations in accurately capturing the dynamic patterns necessary for reliable demand prediction. In contrast, advanced machine learning (ML) and deep learning (DL) algorithms excel in identifying hidden patterns within extensive datasets, offering enhanced performance in challenging scenarios [Sharma et al., 2020]. According to a report published in Forbes [Kazemi, 2019], supply chain practitioners should see Artificial Intelligence (AI), advanced ML and DL approaches as strategic investments in light of industry's growing digital transformation and the need to gain a competitive advantage over competitors.

ML and DL models are often tailored for short-term prediction tasks, given their effectiveness in handling immediate trends and patterns. However, the accuracy of long-term predictions becomes increasingly challenging due to the complex interplay of numerous influencing factors that shape future outcomes. These factors encompass a wide range of variables, including economic shifts, market dynamics, and unforeseen events, making long-term forecasting inherently more intricate and uncertain. Which means there is still a need for long-term prediction models in many areas, including supply chain management [Spieske et al., 2022]. Long-term prediction models can help organizations plan for the future and make informed decisions about production, inventory, and other aspects of their operations. So predicting long-term outcomes involves a deliberative and extended decision-making process. For instance, consider a manager in the transportation industry who must determine the logistics and strategies for distributing stock to various destinations. While this decision may seem focused on immediate operations, it can have long-term implications and consequences. This is particularly true when considering factors such as establishing new distribution networks, negotiating long-term contracts with carriers, or planning for seasonal demand fluctuations, all of which can impact the company's supply chain strategy over an extended timeframe.

Alongside this, developing a general model that is capable of making short, mid-range,

## *1. INTRODUCTION*

---

and long-term predictions, irrespective of the type of dataset, can be a challenging task [Somu et al., 2020]. One approach is to use a combination of different ML algorithms and techniques, such as time series analysis and ensemble learning, to build a more robust and universally acceptable model. It may also be useful to consider incorporating domain knowledge and expert input into a model [Ahmed and Kim, 2023], as this can help improve prediction accuracy. Alternatively, in the case of limited data, the use of transfer learning (TL) to build a model can also be considered, as this can help to leverage knowledge learned from related tasks and improve the accuracy of a model [Wang et al., 2023c].

Accurate demand prediction is an important aspect of supply chain management, as it can help to minimise the costs associated with the overproduction or underproduction of goods [Feizabadi, 2022]. Overproduction can lead to excess inventory, which can tie up capital and increase storage and handling costs. On the other hand, underproduction can result in lost sales and missed opportunities. Validating the accuracy of demand prediction models is an important step in ensuring that they can be effectively used to optimise costs in a supply chain. Through the development of supply chain models by incorporating demand prediction techniques based on ML and TL approaches, production and inventory can be optimised. By leveraging the data and insights generated by these models, companies can simulate and compare different production and inventory scenarios and evaluate their associated costs. This enables decision-makers to conduct cost comparisons across various scenarios and simultaneously evaluate the accuracy of demand prediction models. By analysing the cost variations between scenarios and their alignment with actual outcomes, decision-makers can gain insights into how effectively these models optimise supply chain costs and thereby gauge their predictive accuracy.

In addition to validating the accuracy of a model, it is also important to consider other factors that may affect the performance of a model, such as the quality, quantity and relevance of the data being used for training and testing, and the specific settings and parameters being used for ML algorithms. By carefully evaluating these factors and making adjustments as needed, the performance of models can be improved and ultimately help ensure that there is sufficient stock available to meet customer demand, while also

avoiding the costs associated with excess inventory. Overall, accurate prediction helps optimise cost and increase customer satisfaction in supply chains, which can be analysed through supply chain modelling before real-world implementation.

Considering all these, this thesis will initially present a thorough literature survey, especially related to ML and DL, and then looks for possible research gaps in the literature, especially in the application of ML and DL approaches to supply chains. From there on, the main problem is described, that can overcome the aforementioned challenges and from which the research objectives are identified. Achieving these research objectives will eventually help solve the main problem by bridging the fields of ML/DL and supply chain management together, so that ML and DL can help reduce supply chain costs.

## **1.2 Definition of Some Common Time series Predictive Techniques and their Pros and Cons**

As per the literature, the common models to do predictions are Regression based, ML based and DL based. Here's a definition of these techniques techniques, along with their pros and cons, as well as a list of requirements and evaluation criteria for these algorithms:

### **Predictive Techniques**

#### **Definition:**

**Regression Analysis:** Regression analysis is a statistical technique used to establish the relationship between a dependent variable (the variable you want to predict) and one or more independent variables (predictor variables). In time series forecasting, regression analysis can be used to model the relationship between past observations of a time series (dependent variable) and various features or lagged values of the same or other time series (independent variables). This can help in making predictions about future values of the time series.

## *1. INTRODUCTION*

---

**ML:** ML is a broader field of AI that encompasses various algorithms and techniques for learning patterns and making predictions from data. In time series forecasting, machine learning techniques involve the use of algorithms that can automatically learn from historical time series data and make predictions without explicitly programming rules.

**DL:** Deep learning is a subfield of machine learning that specifically focuses on neural networks with multiple layers (deep neural networks). In the context of time series forecasting, DL techniques involve the use of deep neural networks, such as recurrent neural networks (RNNs) or long short-term memory networks (LSTMs), to model and predict time series data.

### **Pros and Cons of Predictive Techniques**

- Regression Analysis:

**Pros:** Useful for understanding the impact of various factors on sales. Can handle both numerical and categorical data.

**Cons:** Assumes a linear relationship between variables, which may not always hold.

May not capture complex interactions between variables.

- ML Algorithms:

**Pros:** Can handle large and complex datasets. Can capture non-linear relationships.

Suitable for feature selection and engineering.

**Cons:** May require significant computational resources. Require careful parameter tuning. Interpretability can be a challenge.

- DL Algorithms:

**Pros:** Effective for capturing complex patterns in data. Suitable for unstructured data like images and text.

**Cons:** Require large amounts of data for training. Complex architectures may be difficult to interpret. Training can be computationally expensive.

**Requirements and Evaluation Criteria for Predictive Algorithms:**

- Data Quality: Ensure that the input data is clean, accurate, and relevant to the forecasting task.
- Feature Engineering: Select and preprocess features carefully, considering their relevance and impact on the forecast.
- Flexibility: It should be adaptable to different types of data, such as time series, numerical, and categorical.
- Interpretability: For business users, the model's results should be interpretable and explainable to gain trust and insights.
- Accuracy: The primary criterion is how well the algorithm predicts actual sales or demand. Accuracy metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), or Mean Absolute Percentage Error (MAPE) are commonly used to measure prediction accuracy. Also, statistical tests to check the significance of the predictive models.
- Scalability: The algorithm should scale efficiently as data volume and complexity increase.
- Forecast Horizon: Assess the algorithm's performance across different forecasting horizons, including short-term, medium-term, and long-term predictions. Evaluate its ability to provide accurate forecasts for various timeframes.
- Robustness: Evaluate how well the model performs under different scenarios, including periods of volatility, seasonal variations, or unforeseen events like supply chain disruptions. A robust model should handle such challenges effectively.
- Computational Efficiency: Assess the algorithm's speed and resource requirements, ensuring it can meet operational needs. Consider the time it takes to generate forecasts, especially for real-time or high-frequency demand predictions.

## *1. INTRODUCTION*

---

- Generalisability: Evaluate whether the model can generalise well to different products, markets, or business units within the supply chain. It should perform consistently across various scenarios.
- Integration: Assess how easily the algorithm can be integrated into existing supply chain management systems and workflows. Compatibility with existing tools and databases is crucial for seamless adoption.
- Forecast Horizon Trade-offs: Analyze the trade-offs between forecasting horizon, accuracy, and time and money saved. A good algorithm should strike a balance that aligns with the supply chain's specific needs.

By defining clear requirements and evaluation criteria, businesses can choose the most suitable predictive algorithm for their specific supply chain forecasting needs and ensure its successful implementation and performance.

### **1.3 Motivation and Scope of Research**

In recent years, demand prediction studies in supply chain has increased significantly, which can be judged from Figure 1.3.

This graph was plotted based on a search with the keywords “Machine Learning or Deep Learning AND Supply chain demand prediction/forecasting” from Scopus. Going through the keyword section, a majority of the published works (around 40 articles) are mainly related to ML forecasting, while only 4 articles are about analysing supply chain cost. Although it is clear that a certain amount of research has already been done on developing ML or DL tools for making predictions, however their applications and possible impacts on supply chain profitability are still to be explored for several reasons:

**Limited Implementation:** Although there is a considerable amount of research done on developing forecasting models, the implementation of these models in real-world supply chain scenarios is still limited. This is problematic as companies are often reluctant to

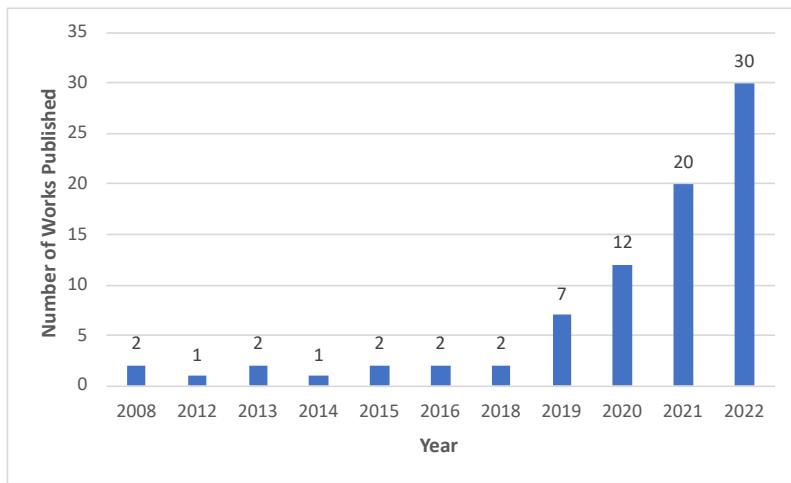


Figure 1.3: The number of research articles on ML or DL approaches for making Supply chain demand predictions since 2008 (Source: Scopus)

adopt new forecasting approaches due to the potential risks involved in changing their existing processes.

**Lack of Standardisation:** The lack of standardisation in forecasting approaches makes it difficult to compare the performance of different models. This makes it challenging to identify the most effective forecasting approach for a particular supply chain network.

**Complexities in Supply Chain:** Supply chain networks are complex, dynamic systems that make it challenging to develop accurate forecasting models that can account for all the variables involved in the supply chain.

These challenges can be addressed through real-world testing, developing standardised evaluation metrics and collaborative work between academia and industry. Real-world testing scenarios will help identify potential implementation challenges and provide insight into how these models can be effectively integrated into existing supply chain processes, while standardisation will enable companies to identify the most effective forecasting approach for their particular supply chain network. Finally, collaboration between academia and industry can help address the complexities involved in supply chains. Hence it can be said, that in spite of the applicability and necessity of having research on ML or DL models for making supply chain demand predictions, there are still a lot of gaps in the

## *1. INTRODUCTION*

---

current literature, which is one of the motivations behind this thesis.

There are also other motivations for carrying out this research work. One motivation is to explore the potential opportunities to improve the efficiency of demand forecasting algorithms and reduce supply chain process costs. Similarly, the use of ML to optimise transportation routes or warehouse operations can lead to reduced transportation costs and improved resource utilisation.

Another motivation for research in this area is the increasing complexity of modern supply chain systems [Rejeb et al., 2019, Tan and Sidhu, 2022]. With the rise of e-commerce and globalisation, supply chains have become more complex, with a larger number of stakeholders and more variables to consider [Rebs et al., 2019]. ML and DL approaches can help to manage this complexity by automatically identifying patterns and relationships in data, enabling better decision-making and problem-solving while answering the two most critical questions in a supply chain: when to order and how much to order [Li et al., 2022a].

Finally, research in this area is motivated by the potential power of ML and DL techniques to enable more personalised and responsive supply chain systems. For example, ML and DL algorithms can be used to personalise recommendations for customers based on their past purchasing history, leading to improved customer loyalty and satisfaction. In addition, these approaches can be used to continuously monitor and adjust supply chain operations, enabling more responsive and flexible systems.

## **1.4 Brief Literature Review**

Time series prediction involves forecasting future values of a variable based on its past behaviour over a time interval. In other words, it is the process of using historical data to predict future trends or patterns in a time series. In recent years, the fields of ML and DL have seen significant advancements due to the fact that ML and DL are capable of capturing complex patterns and relationships in data that may not be apparent to humans.

They can analyse large volumes of historical data and identify patterns and trends that can be used to make accurate predictions about future demands or sales. Several studies have explored the application of ML and DL techniques for this purpose and have reported promising results. In addition, there has been a growing interest in TL, where limited data is used to make predictions for a specific target item [Zhuang et al., 2022]. This chapter aims to provide a brief overview of the current research in this area, identify research gaps, and discuss potential solutions to these gaps.

#### **1.4.1 Traditional, ML and DL approaches to predict sales data**

Sales forecasting has long been a critical area of research and practice for businesses across industries. Traditional approaches for predicting sales data include time series models, such as exponential smoothing [Chern et al., 2023], moving averages [Wang and Luo, 2023], and autoregressive integrated moving average (ARIMA) models [Choudhary et al., 2023], as well as causal models, such as regression [Musora et al., 2023] and econometric models [Jiang et al., 2023]. These models rely on statistical techniques to forecast sales based on historical data and explanatory variables, such as economic indicators, demographics, and marketing activities [Baumohl, 2012]. While these traditional approaches have been widely used for decades, they have some limitations. For example, they may struggle to capture complex patterns and relationships in data, especially when there are non-linearities and interactions between variables [Omar et al., 2023]. Additionally, they may not be able to handle large and diverse datasets [Ame Rayan et al., 2023], which are increasingly common in today's digital age. As a result, there has been growing interest in applying ML and DL approaches to demand/sales forecasting.

Organisational modelling has recently received a lot of attention as new technologies like DL and artificial neural networks are used to simulate human brain function in computer applications [Chang et al., 2019]. When it comes to AI's predictive abilities, there was initially a lot of hype and hyperbole, but when looked more closely at the circumstances under which AI could deliver accurate predictive power, the enthusiasm has gradually

## *1. INTRODUCTION*

---

subsided [Makridakis et al., 2020]. One of the key focuses of modelling is to foster the decision-making process based on data by analyzing patterns and forecasting product demand, which has been influenced by many different AI models [Bohanec et al., 2017, Ahmad et al., 2022, Aichner and Santa, 2023, Dellino et al., 2018]. Only a small number of characteristics, such as trends, seasonality, and periodicity, that influence demand to develop a prediction have been taken into account by traditional demand forecasting methodologies. However the vast majority of causal factors with non-linear relationships may simultaneously be analysed and taken into account using ML-based forecasting systems, which integrate learning algorithms with large data [Chase Jr et al., 2016]. ML approaches can provide necessary tools for both modelling and managing operations in the uncertain supply chain field, mainly because the associated algorithms are capable of dealing with complex inter-dependencies among data. ML techniques can learn from data, make judgements depending on environmental factors, and then continue learning from those previous decisions [Mitra et al., 2023].

ML techniques have been used in a variety of fields, such as macroeconomics [Petropoulos et al., 2023], finance, and marketing [Ashtiani and Raahmei, 2023, Han et al., 2023, Langen and Huber, 2023, Benramdane et al., 2023]. In particular, ML techniques have been used for a variety of supply chain tasks, such as demand forecasting [Garinian et al., 2023, Umoren et al., 2023, Aichner and Santa, 2023], resource allocation, and transportation management [Vadlamani and Modashiya, 2023]. The impact of inaccurate demand forecasts on the bullwhip effect has also been studied using sophisticated and hybrid ML techniques [Garinian et al., 2023, Tavakol et al., 2023, Pham et al., 2023]. More discussions on ML models in the supply chain domain are discussed in Chapter 2.

The development of techniques to combine causal regression with time-series models under Artificial Neural Network (ANN) has made it possible to include a greater variety of explanatory variables, such as discounts, promotions, macroeconomic factors, and pricing. For instance, 17 endogenous and exogenous independent factors were utilised by Rabelo et al. [2004] to forecast demand over short- and long-term horizons. Another example is the use of autoregressive time series models by Bennett et al. [2014], Zhang et al. [2023],

in conjunction with other causative factors impacting demand (such as ARIMAX) and neural networks (NN) to forecast energy consumption in residential areas. To forecast intermittent demand patterns, Lolli et al. [2017], Aichner and Santa [2023] contrasted feed-forward neural networks with a back-propagation learning scheme with an easier and more effective learning method termed extreme learning machines. Another useful feature of ML algorithms is that these algorithms can effectively handle a wide range of explanatory variables and forecasting tasks without relying on assumptions about the probability distribution of the data. Therefore, an essential open research inquiry arises as to what is the most efficient and productive method to optimise a supply chain's costs by forecasting long-term sales data, which is crucial for making informed decisions and resource allocation over extended planning horizons, especially when faced with an enormous amount of information.

#### **1.4.2 Hybrid Prediction Models**

Hybrid models that combine different ML/DL techniques have been gaining attention for their ability to improve the accuracy and efficiency of sales and demand prediction. The use of specific ML/DL techniques for demand forecasting has yielded conflicting outcomes, which has forced scholars to deploy hybrid models. Hybridisation is important for sales prediction because it combines the strengths of different ML techniques and overcomes some limitations of individual models [Charef, 2023]. Hybrid models can handle complex and diverse data sets, provide more accurate and reliable predictions, and improve the efficiency of the prediction process. Hybrid models can also adapt to changes in the data and environment, making them more robust and flexible for sales prediction [Fianu, 2022].

For instance, several researchers discovered that conventional techniques, like exponential smoothing and Winters exponential smoothing, produced forecast accuracy equivalent to or occasionally greater than ANNs [Dugan et al., 1994, Foster et al., 1992]. That is why researchers such as Egrioglu and Bas [2022] and Dullah et al. [2022] combined these exponential smoothing approaches with ANN to create hybrid approaches so that the

## *1. INTRODUCTION*

---

models can extract the benefit of both and make better predictions. Numerous studies in the past support the notion that integrating different forecasting techniques will increase forecast accuracy [Armstrong, 2001, Hajirahimi and Khashei, 2019, Kaytez, 2020, Feng and Niu, 2021, Lv et al., 2022]. The need for more research led to the development of hybrid methods using ML and statistical techniques, such as Autoregressive Integrated Moving Average (ARIMA) mixed with ANNs [Aburto and Weber, 2007, Büyüksahin and Ertekin, 2019, Bhardwaj et al., 2022, Wahedi et al., 2022]. In Chapter 3, there is further information on hybrid forecasting models for supply chains. The accuracy of models can be further examined by observing the efficiency of a supply chain network through cost analysis. Such a type of analysis is not widely available in the current literature as combining predictive models with supply chain models for prescriptive cost analysis is challenging due to several reasons, such as complex supply chain networks [Iftikhar et al., 2022], dynamic market conditions [Wang et al., 2023a] and most notably due to the lack of standardisation [Mangla et al., 2022]. To address these complex challenges, businesses require expertise from data science and supply chain management, which is rare and still needs improvement especially due to the evolving nature of technology and the ever-increasing complexity of supply chain operations. Therefore, a pivotal research question emerges: What constitutes a robust hybrid approach, independent of specific data characteristics, capable of consistently generating accurate sales data predictions across various forecasting horizons to minimise supply chain costs? This is particularly pertinent given the dynamic nature of data and the evolving demands of modern supply chain management.

### **1.4.3 Application of TL in Predicting Sales Data for Newly Launched Products**

The ML/DL approaches mentioned above produce good results when ample data is available. But in cases of limited data, such as for a newly introduced market product, standalone statistical and ML models frequently produce predictions with low predictive accuracy, making expert knowledge an important tool for enhancing forecasting quality [Kahn,

2014]. This specialist knowledge can be applied when choosing comparable reference products to predict new product sales or revise forecasts for particular occasions. According to Kahn [2014], these decisions are based on an expert's judgement, assumptions, and implicit and explicit knowledge. Therefore, it is important to consider the possibility of human bias and the fact that judgements are made with incomplete data. Hence the ongoing accessibility of these professionals is crucial to business success. This creates an opportunity for scholars to develop another extension of a ML technique, namely Transfer Learning (TL).

TL, an extension of the ML technique, is where a model trained for one task is used as the starting point for a model for a second related task, which can be considered a replacement for human intervention, as it mimicks human behaviour in the system. It can be a useful approach when data is limited for the desired task because it allows the leverage of knowledge learned from a related task and applies it to the new task. Recently, a lot of attention has been paid by researchers in the field of TL across prediction domains such as energy demand [Ribeiro et al., 2018], energy production [Luo et al., 2022, Manandhar et al., 2023] and financial prices [Pal and Kar, 2022]. Neural networks, regarded as "black boxes", are being used as information carriers in business networks to enable inter-organizational learning because of the possibility of transferring ML models [Hirt and KÄ1/4hl, 2018]. Network-based TL with DNNs is appropriate for time series forecasting, as demonstrated by Otović et al. [2022]. The authors also demonstrated that TL models can outperform routinely trained DNNs in various domains. These findings are consistent with prior forecasting case studies that used a network-based TL technique with DNNs [Zhou et al., 2022, Yang and Schell, 2021, Voß et al., 2018, Zhou et al., 2020]. Further studies on TL are highlighted in Chapter 4. These encouraging findings open further avenues to use a TL method to solve the supply chain problem of anticipating new product sales, and in turn brings forth another research question: how to create a TL approach that can effectively maximise the knowledge transfer process in the context of limited data for forecasting purposes.

## **1.5 Research Gaps And Possible Solutions**

The literature review briefly described above has revealed several gaps in the current supply chain literature. By addressing those gaps, the understanding of the topic can be improved and contribute to the development of more effective supply chain networks. The following section of the thesis will delve deeper into these gaps in knowledge and discuss possible avenues to address them.

### **1.5.1 Absence of Models for Long Term Forecasting with an abundance of data**

Long-term forecasting is a crucial aspect of managing a successful business, especially in the supply chain domain. Despite the abundance of literature on long-term forecasting in other fields, there remains a gap in the availability of algorithms for predicting long-term customer demand in the supply chain domain, where there is a wealth of historical sales data available. Long-term forecasting provides a valuable tool for practitioners and managers to make informed decisions [Attaran and Celik, 2023], such as managing their transportation fleet and determining the optimal size and location of warehouse space for materials and finished products. This information can also benefit long-term investors as they can gain insights into the projected performance of a particular company, helping them make more informed investment decisions. Thus a lack of research in this area suggests that there may be challenges or limitations associated with using these approaches for long-term forecasting.

One possible solution to this research gap is to develop and improve DL models for more accurate long-term predictions. DL models possess the capability to handle extensive datasets and uncover intricate patterns, a feature that extends beyond stock market prediction and can be leveraged for enhancing demand forecasting and optimising supply chain costs. By improving these models and applying them specifically to long-term forecasting, researchers may be able to achieve more accurate predictions. Additionally, researchers

could investigate different architectures, such as Long Short Term Memory (LSTM), which are specifically designed for sequential data, such as time series data, to make predictions for longer periods of time. These architectures could be combined with other techniques, such as feature engineering or TL, to improve the performance of such models. Considering these research gaps, the first research objective of this thesis was designed, as explained in Section 1.6.

### **1.5.2 Scarcity of Hybrid Models that are Independent of Data Characteristics and make more Accurate Predictions**

The existing body of knowledge in forecasting also lacks a comprehensive hybrid prediction approach that can accurately predict outcomes over different forecasting horizons. This hybrid approach should be capable of handling large amounts of data and should not be influenced by the characteristics of the data or the specific forecasting horizon. Such a model would greatly benefit practitioners and managers in their decision-making process and provide a more robust forecasting system. According to researchers [Koohfar et al., 2023, Chen and Fu, 2023], traditional ML and DL models struggle with different prediction horizons as they are not designed to adapt to different scenarios. Additionally, these models also struggle with changes in the characteristics and features of their dataset [Grinsztajn et al., 2022], which can negatively impact prediction accuracy. This indicates a need for a more comprehensive solution that can adapt to different scenarios and can handle changes in the dataset.

A possible solution to this gap is to use a switching-based approach, which combines multiple ML and DL methods, for better accuracy and lower supply chain cost. This approach would likely be more robust to changes in the dataset and would improve prediction accuracy for different prediction horizons. Such an approach would also help to reduce the cost and complexity associated with supply chain management by providing accurate predictions that can be used to optimise the supply chain. Given the identified gaps in research, the second objective of this thesis has been developed and is elaborated in Section 1.6.

## *1. INTRODUCTION*

---

### **1.5.3 Absence of Forecasting Models with Limited Data/Information**

The need for TL arises due to the absence of forecasting approaches that can handle limited data or information. Usually, conventional DL models struggle with limited data and are not very effective in making predictions for newly launched products, which is a challenge for DL approaches. TL can help overcome this challenge by transferring knowledge from multiple similar source products with critical features to develop a pre-trained network, which can then be used for predicting sales patterns of new products with insufficient data. The use of TL in forecasting can benefit managers in selecting appropriate channels, through which newly launched products can be sold based on operating cost, as well as the corresponding time and accuracy of different TL approaches. Eventually, this will benefit businesses by making higher profits and better productivity. There is a lack of substantial research on this topic in the most recent literature. This is quite significant as accurate demand prediction is crucial for optimising supply chains and reducing costs associated with carrying excess inventory.

One potential way to address this research gap involves conducting experiments and devising a TL approach to enhance accuracy and reduce supply chain costs associated with newly introduced products. TL involves identifying similar products and using the knowledge gained from training a model on these similar products to make predictions for the newly launched item. The TL approach can also be used in combination with other techniques, such as feature engineering, which can help to extract relevant features from the limited data available for the newly launched item, or the ensemble method, which combines multiple models to improve the overall performance. Taking into account these areas of research that require attention, the third and final research aim of this thesis has been formulated and is elaborated on in Section 1.6.

## 1.6 Research Objectives

This thesis embarks on a comprehensive journey aimed at minimising supply chain costs through the implementation of cutting-edge machine learning and deep learning prediction models. The research is structured around three key objectives, each contributing distinct elements to the overarching goal. These objectives form a cohesive pathway toward the overarching aim of minimising supply chain costs. They collectively empower decision-makers with advanced predictive tools, enabling efficient stock management, robust forecasting, and data-driven decisions. This holistic approach contributes significantly to enhancing supply chain efficiency and cost optimization, ultimately benefiting a wide range of supply chain operations. This is be further discussed in the later parts of this thesis.

### ***Objective 1: Effectively and Efficiently Predict Long-term Sales data to Optimise Supply Chain Cost***

The first research objective for this thesis is to develop a DL algorithm that is capable of effectively and efficiently predicting long-term sales data to optimise costs. The effectiveness aspect pertains to the algorithm's ability to provide highly accurate and reliable forecasts of future sales, while the efficiency aspect focuses on generating these long-term forecasts in a timely manner, thereby reducing the computational burden and associated costs. This involves exploring different types of algorithms and techniques that are suitable for such a task, such as time series analysis with other ensemble learning methods. The algorithm can be specifically tailored to the needs of supply chain decision-makers, with the goal of helping them to optimise stock levels and make informed decisions about production and inventory management. Developing a novel DL algorithm that is capable of making long-term predictions in the context of supply chain management can contribute to the body of knowledge in this area and provide decision-makers with a valuable tool for optimising costs and increasing efficiency in their operations. To execute the research objective, the following steps will be taken (more discussed in Chapter 2):

## *1. INTRODUCTION*

---

- Data Collection
- Regression Analysis and Data Pre-processing
- Data Split and Data Augmentation
- Developing LSTM Architecture
- Data Structuring and Train the LSTM Network
- Testing the Trained Network
- Result Analysis and Model Validation with Parameter Tuning

### ***Objective 2: Design a Robust Hybrid Approach to Predict Sales Data over Varying Forecasting Horizon***

Another research objective for this thesis is to develop a hybrid DL algorithm that is capable of making short, medium, and long-term predictions, and is not limited to the characteristics of a specific dataset, thereby being a more robust and generalisable model. Experimentation with different techniques for pre-processing and feature engineering to ensure that the model is able to effectively extract relevant information from the data and make accurate predictions could also be explored. Considering the complexities and limited implementation of forecasting approaches in supply chains, a significant contribution can be done to the body of knowledge in this area, as well as it can be used as a valuable tool for decision-makers in a variety of contexts. This could help to optimise costs and increase efficiency in a variety of operations, including those in the supply chain management domain, such as minimising overproduction, reducing lost sales and optimising inventory handling.

This can be demonstrated through the development of a supply chain model to calculate costs. The accuracy and effectiveness of ML/DL predictive models can be evaluated by comparing the estimated costs made by the supply chain model. To accomplish this objective, data on various components of the supply chain, such as purchasing cost, holding cost, ordering costs, and other relevant factors such as lot size that impact the overall cost

of the supply chain, can be gathered, and cost predictions can be generated for different scenarios and configurations of the network. In the end, researchers can use these costs to gain a better understanding of the strengths and limitations of these approaches and identify ways to improve their performance.

To accomplish the above research objective, the following steps will be taken (as discussed in Chapter 3):

- Exploring Traditional and Advanced Forecasting Models
- Introducing the Proposed SBFA Model
- Developing the Considered Supply Chain Model
- Experimental Design For the Proposed SBFA
- Data Collection, Feature Extraction, Data Structuring and Hyperparameter Selection
- Cross Validation
- Result Analyses for the Proposed SBFA in terms of Accuracy and Supply Chain Cost
- Theoretical Derivation

**Objective 3: *Develop a TL Approach to Maximise Information Transfer Process with Limited Available Data***

Developing efficient TL algorithms for making predictions with limited data is another valuable research objective of this thesis. TL is a DL technique that allows one to leverage the knowledge learned from one task and apply it to a related task and can be particularly useful when working with limited data. For successful implementation of the TL approach, similar sources with an abundance of data can be identified that have the same behaviour as that of data with limited information. Next, important features can be extracted from

## *1. INTRODUCTION*

---

those identified sources which can be used to train a TL network. The network can then later be used to retrain on the limited dataset for making predictions of the target domain. The effectiveness of the TL approach can be compared against other TL approaches and can be further validated using the supply chain model.

In order to achieve that, the obtained datasets are transformed in such a way that they represent the sales of freshly launched products or stores, while containing the information of other products that have long sales history. The transformed dataset can be represented in a hierarchical supply chain structure with one supplier and different retailers where retailers are selling different finished items to the customers alongside the new products. The developed TL approach acquires knowledge from other similar items and makes more accurate predictions of recently introduced products that can be used to simulate different scenarios of the supply chain model with different configurations for cost calculations. Through such modelling, practitioners and managers will gain better visibility of the strengths and limitations of TL approaches, which will add value to their decision-making process.

To achieve the above research objective, the following steps will be taken (as discussed in Chapter 4):

- Problem Definition
- Defining the Proposed Model: MSML-TL-RFE
- Model Implementation
- Data Collection, Feature Extraction and Data Structuring and Hyperparameter Selection
- Experimental Design: Similar Source Products Identification, Prediction On Target Domain
- Simulating Different Supply Chain Scenarios for Cost Calculation with Predicted Data

- Result Analyses and Theoretical Derivation

### **1.6.1 An Overview of Expectations from Practices**

In the realm of supply chain management, the successful application of predictive algorithms hinges not only on their accuracy and efficiency, but also on their adaptability to the diverse risk-taking profiles of practitioners. The expectations from these predictive models encompass a spectrum of risk appetites. Some practitioners, guided by a risk-averse approach, place a premium on the reliability and precision of sales predictions. For them, minimising forecasting errors is paramount, as it enables conservative decision-making, reducing the likelihood of unexpected disruptions and associated financial losses. In contrast, there are practitioners who are inclined toward a risk-taking perspective, seeking to capitalise on opportunities and maximise operational agility. Efficiency in prediction algorithms becomes their focal point, as it enables a rapid response to market dynamics and shifts in demand. Therefore, the research objectives aim to address this duality of expectations, delivering predictive algorithms that can be fine-tuned to practitioners' risk profiles, ultimately offering a versatile toolkit for supply chain decision-makers. This adaptability ensures that predictions align with the specific needs and risk preferences of practitioners, allowing them to optimise stock levels, production schedules, and inventory management while either minimising losses or maximising responsiveness, depending on their individual risk propensities.

## **1.7 Contributions of Thesis**

This thesis makes the following contributions to the body of knowledge from conducting research to address the above-mentioned research objectives:

### **1.7.1 Efficient and Effective Prediction Model for Long-term Time-series Data**

Using a mix of statistical regression analysis and SLSTM, an advanced DL model is introduced, which is termed as PLR-ALSTM-NN. Statistical regression produces a strong association between input and output data, and SLSTM's short-term and long-term memory components allow it to retain short and long-term patterns of prior data selectively. Synthetic training data is manufactured alongside actual data to increase the proposed DL model's generalisation capabilities on test data. The accuracy of time series data forecasting is improved as a result of this enhancement. To acquire the greatest potential anticipated outcome, the entire model has been fine-tuned using the optimal mix of parameters. For four stock market datasets, extensive validation of the proposed model was performed in contrast to contemporary ML and DL techniques in the literature, to demonstrate its acceptable performance in financial market engineering. Incorporating these datasets is essential due to their relevance in addressing forecasting challenges, not only within financial markets, but also in broader contexts. Financial markets, characterised by their dynamic and multifaceted nature, influenced by factors like economic events and investor sentiment, serve as a rigorous testing ground for the proposed model. This evaluation creates an opportunity to determine its adaptability and effectiveness in predicting intricate financial dynamics, bolstering its credibility beyond finance. Furthermore, recognising the model's applicability in diverse sectors, this thesis extends its usage to time series supply chain data, demonstrating its versatility. Specifically, it is employed in predicting sales data, a pivotal challenge in supply chain management. This holistic analysis spanning financial and non-financial domains underscores the model's potential to tackle forecasting issues across various sectors, as detailed in Chapter 2.

In practice, PLR-ALSTM-NN is valuable for long-term investors as it allows them to focus on the growth of businesses and analyse the viability of new business models without reacting to short-term fluctuations. Long-term investing also eliminates the hassle of learning different trading systems and platforms and minimises investment risk by eliminating lost opportunities. Long-term DL predictions help investors stay invested over a longer time,

reducing the chance of missing out on the best days. Investing for the long term creates greater profit over time, which ultimately contributes to overall productivity and positively impacts GDP (Gross Domestic Product). The work related to objective 1 is published in "Information Sciences" journal [Ahmed et al., 2022].

Furthermore, the research related to objective 1, which addresses the application of PLR-ALSTM-NN in finance, has a significant impact on supply chain forecasting. By developing robust long-term prediction models in finance, we can establish a foundation for more accurate and reliable forecasting methods in supply chain management. The techniques and insights gained from financial data analysis can be extended to optimise supply chain operations, reduce costs, and enhance efficiency. Thus, addressing financial forecasting challenges can have a cascading effect, benefiting supply chain forecasting and contributing to the overall improvement of business operations.

### **1.7.2 A Robust Hybrid Predictive Model for Accurate Sales Forecasting Across Various Time Horizons**

As a second contribution, a switching-based forecasting approach (SBFA) is developed, which involves the selection of the most suitable forecasting method by evaluating their performance on different datasets. This selection is based on the lowest validation error achieved by each method. A two-echelon supply chain model with different attributes is used to validate the proposed SBFA against a few traditional forecasting approaches. The reorder points of the supply chain model are calculated based on the predictions from conventional/ML/DL forecasting approaches. Predictions from SBFA and other approaches are analysed by calculating the overall supply chain cost. Finally, a theoretical derivation is developed to establish a selection procedure among the forecasting approaches while considering the trade-offs between forecasting horizon, accuracy, time and money saved. Details about SBFA can be found in Chapter 3.

The practical significance of SBFA has significant benefits for managers and organisations. Instead of reacting to only short-term fluctuations, they can focus on their business's short

## *1. INTRODUCTION*

---

to long-term growth and analyse the viability of new business models. AI and ML advancements provide the advantage of calculating future expenses and adjusting pricing policies and inventory management. Supply chain forecasting enhances operations by planning for unforeseen events, delays, and changes in demand. Leveraging data and insights for smarter decision-making can reduce costs, satisfy customer demand, and maintain the efficiency of their supply chain, leading to increased productivity and greater profit. The work related to objective 2 is submitted in "Neurocomputing" journal [Ahmed et al., Under review 2nd Round, submitted Oct. 2022] and is currently under a second round of review.

### **1.7.3 A Highly Effective and Robust TL Model That Leverages Multiple Similar Sources of Data for Optimal Sales Prediction of New Market Products**

The last contribution of the thesis aims to enhance the prediction accuracy of TL through a knowledge adaption strategy. Firstly, similar sources are identified for an efficient knowledge transfer process. Then, the effect of harvesting knowledge from single to multiple sources on TL approaches is investigated. Using single to multiple sources, experiments are conducted to analyse the performance of different TL approaches. After conducting extensive experiments with different TL approaches and numerous datasets, the best one, which is Multi-source Multi-layer based TL approach with Recursive Feature Elimination (MSML-TL-RFE), is selected for further investigation. In the context of supply chains, two SC scenarios are simulated to show information sharing leads to better accuracy of TL approaches, rather than no sharing. The SC cost calculation is done to show that such sharing of information among retailers leads to better accuracy in the target domain, which ultimately helps in reducing SC cost along different channels. More details about MSML-TL-RFE can be found in Chapter 4.

MSML-TL-RFE is particularly beneficial for practitioners who deal with limited data. With AI and TL advancements, practitioners can calculate future expenses, adjust pricing policies and inventory management, and make future investments with lower risk.

They will be able to select the appropriate channel for delivering goods and services to their customers while incurring a minimum cost. This ability to make smarter decisions with limited data ultimately leads to overall improved productivity and profits for the entire network. The work related to objective 3 is submitted in "Expert Systems with Applications" journal [Ahmed et al., Under review, submitted July. 2023] and is currently under review.

The above contributions are established through numerical experiments for benchmarking and analysis on a range of analytical and practical problems. Comparisons are provided with a number of systematically constructed variants as well as state-of-the-art algorithms from the literature, to quantify the benefits obtained in various performance metrics of interest. Moreover, these offer a comprehensive suite of practical tools and approaches that hold the potential to revolutionise supply chain management practices for businesses. They encompass efficient long-term sales prediction through advanced deep learning techniques, robust sales forecasting that is adaptable across various time horizons, and optimised transfer learning strategies for new market products. Collectively, these contributions aim to minimise forecasting errors, resulting in cost-effective demand planning, reduced operational expenses, and enhanced inventory management. By providing decision-makers with valuable insights, they improve procurement, production, and distribution decisions, thereby enhancing overall supply chain efficiency and resilience. These advancements not only confer a competitive advantage by reducing lead times and costs but also pave the way for increased profitability, growth, and market expansion. In essence, the thesis bridges theoretical advancements with practical applications, offering valuable assets for researchers and practitioners alike in the realm of supply chain management.

## **1.8 Organisation of Thesis**

This thesis consists of five chapters in total, where three of them are technical chapters, as shown in Figure 1.4. These chapters are organised as follows:

## 1. INTRODUCTION

---

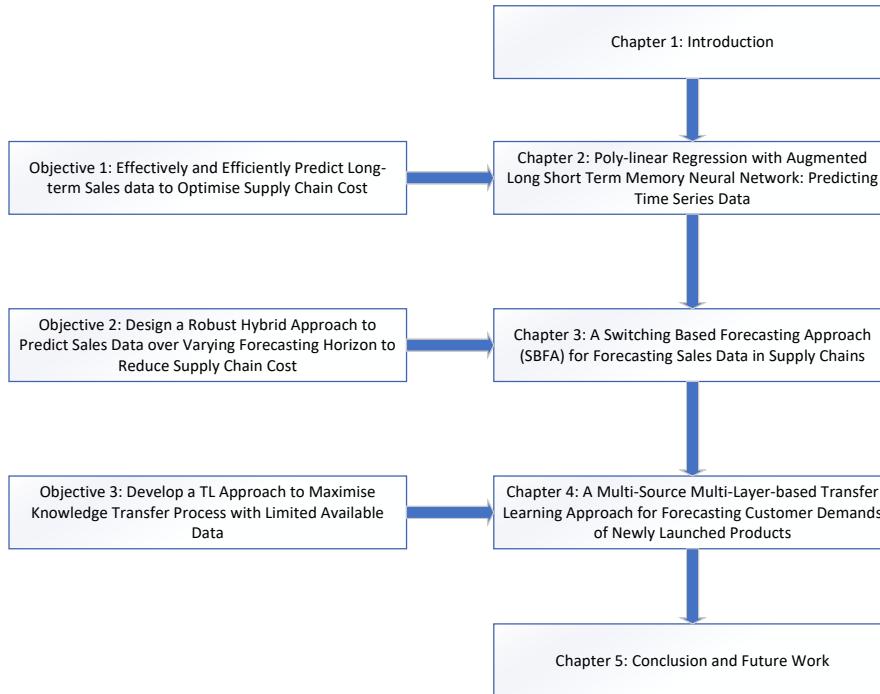


Figure 1.4: The Organisation of this Thesis

- In Chapter 1, an introduction of the thesis is presented. It first provides a background of the research field on Artificial Intelligence related to Supply Chains, followed by a brief literature review section highlighting research gaps. The motivation behind this thesis and the possible scope of its research is then discussed in this chapter. It also presents the considered problem statement and the importance of the thesis. Following that the objectives and a list of scientific contributions stemming from this research are presented. The last section of the chapter presents the organisation of the thesis.
- Next, in Chapter 2, a novel DL approach is presented for time series prediction using a combination of poly-linear regression with Long Short-Term Memory (LSTM) and data augmentation. It is consequently named Poly-linear Regression with Augmented Long Short Term Memory Neural Network (PLR-ALSTM-NN). The proposed DL model can be exploited to predict future financial markets more accurately than existing state-of-the-art neural networks and ML tools.

- After that, in Chapter 3, a switching-based forecasting approach (SBFA) is proposed to exploit the merit of different advanced ML/DL approaches for different days ahead of prediction. Based on the performance of validation data, the proposed system automatically switches between different approaches to determine a more appropriate forecasting approach. A two-echelon supply chain model with different attributes is developed to validate the proposed SBFA against a few traditional forecasting approaches. Predictions from SBFA and other approaches are analysed by calculating overall supply chain cost.
- In the last technical chapter, Chapter 4, several AI-based TL approaches are investigated to predict customer demand for new products/stores in the market. Among the different approaches, a Multi-source Multi-layer based TL approach with Recursive Feature Elimination (MSML-TL-RFE) strategy is proposed to exploit the knowledge extraction power of the model from different sources for different days ahead of prediction. Also in this chapter, an abstract concept of supply chain regarding information sharing among retailers is investigated to show that such concepts improve the knowledge transfer ability of the system. A hierarchical two-echelon supply chain model with different attributes is developed to validate the proposed MSML-TL-RFE approach against a few other TL forecasting approaches. The reorder points of this supply chain model are calculated based on the predictions from TL forecasting approaches considered in this paper. Predictions from MSML-TL-RFE and other TL approaches are analysed by considering overall supply chain cost.
- Lastly, Chapter 5 provides the concluding remarks and potential avenues for future research.

## Chapter 2

# Poly-linear Regression with Augmented Long Short Term Memory Neural Network: Predicting Time Series Data

*The work presented in this chapter has been published in the following article:*

- Ahmed, S. S., Chakrabortty, R. K., Essam, D. L., and Ding, W.. “Poly-linear Regression with Augmented Long Short Term Memory Neural Network: Predicting Time Series Data,” *Information Sciences*, pp. 573-600, 2022. (IF: 8.233)

## Overview

*Until recently, the supply chain sector, which had been getting by with scattered spreadsheets, phone conversations, and even paper-based records, was exposed for its antiquated methods during the epidemic. As a result, businesses have undergone a decade of digital change in only a few months, with the epidemic driving them to replace antiquated*

*procedures with AI, machine learning, and data science technology. The supply chain sector has reached a point in its AI adoption where the technology is solid and powerful enough to improve decision-making significantly. For example, predictive analytics (e.g., time series forecasting) has already proven its benefits. Such technology is smart enough to recognise irregularities and learn how a stock market will move in real-time. With the advancement of digital innovation, researchers have focused on deep learning (DL) models to get a more accurate and unbiased estimation. Consequently, this chapter presents a novel DL approach for time series prediction using a combination of poly-linear regression with Long Short-Term Memory (LSTM) and data augmentation. It is consequently named Poly-linear Regression with Augmented Long Short Term Memory Neural Network (PLR-ALSTM-NN). The proposed DL model can be exploited to predict future financial markets more accurately than existing state-of-the-art neural networks and machine learning tools. In order to make the model a more generic one, it is first validated on four financial market time-series datasets and then also implemented on a supply chain time-series dataset to predict sales data. LSTM, with its feedback connections, can process an entire series of data as well as single data points and statistical regression establishes the strength and character of the relationship between some dependent and independent variables. After doing experimental validations and based on the long-term and short-term predicted data, the suitability of the proposed PLR-ALSTM-NN is well-grounded against a few recent and advanced state-of-the-art machine learning, and DL approaches.*

## **2.1 Introduction**

Analysis of time series data is an important scientific issue that has a wide range of applications in the real world. Over the last few decades, a lot of work has developed and improved time series forecasting models. The work presented here focuses on developing a unique neural network (NN) architecture that improves accuracy and may be used to make broad-time-series predictions. It is often assumed that with the advent of Artificial Intelligence (AI), robots would be able to make future predictions. Although this is true

## *2. POLY-LINEAR REGRESSION WITH AUGMENTED LONG SHORT TERM MEMORY NEURAL NETWORK: PREDICTING TIME SERIES DATA*

---

in certain cases, it is not true in all cases. Prices in the financial market are unpredictable and uncertain owing to the market's short-term micro-structures. Humans are unable to predict such uncertainty using standard statistical analysis. As a result, predicting the movement of stock values becomes a difficult problem for AI researchers. Over the last several decades, a lot of research has been done on predicting stock market movement using a variety of machine learning (ML) techniques. Models like Auto-Regressive (AR), Auto-Regressive Moving Average (ARMA), and Auto-Regressive Integrated Moving Average (ARIMA) have been frequently utilised for projecting the future from prior time-series data. These are essentially statistical analytic approaches that define and model the data pattern by building a form of link between the variables. In contrast, ML models are meant to produce the most accurate predictions.

Time-series analysis is applicable in many sectors like electrical signal analysis [Pérez-Chacón et al., 2020], traffic condition analysis [Priambodo and Jumaryadi, 2018], weather forecasting [Wang et al., 2016], and rate of unemployment analysis [Katriss, 2019]. Among all these time series applications, financial markets also belong to the same category, however, it is a special type because of the various activities, noisiness and complexity of its nature. Deep learning (DL), a subset of ML, can be deployed to predict and classify such high-complexity data. Among DL, a more specific branch is Deep Neural Network (DNN), which shows superior performance across many areas of applications [Bi et al., 2021], [Alhnaity et al., 2021], [Shi et al., 2021]. The main reason for its success is its feature extraction ability which is done automatically in different layers of a network. It seems that DNN techniques can be a recommended solution in predicting financial time series while considering its complex nature. Within DNN, Convolutional Neural Networks (CNNs) and deep Recurrent Neural Networks (RNNs) are a few approaches that researchers currently use to solve high-complexity data. CNN is mostly used for solving classification problems, however, it can also be used for making time series predictions [Chen et al., 2021]. Deep RNN, on the other hand, is a more prominent algorithm that has been utilised in many prediction approaches. There are different versions of deep RNN, such as Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU). LSTM, in particular, can preserve previous network states as well as capture long-term dependencies, thus improving the

---

## *2. POLY-LINEAR REGRESSION WITH AUGMENTED LONG SHORT TERM MEMORY NEURAL NETWORK: PREDICTING TIME SERIES DATA*

---

ability of RNN. Since experimental analysis is done on the important financial market, the focus is more on the quality of prediction rather than computational time, which is considered the main reason for choosing LSTM over other NNs.

Bidirectional LSTM (BiLSTM) [Pradhan et al., 2021] is another sort of RNN that achieves the highest learning processing performance by using the full input data as previous and following input sequences. An LSTM network's depth can be increased by stacking more hidden layers together. The stacked LSTM (SLSTM) network is more significant than the number of memory cells in a particular layer to make the system more efficient. By adding several layers of abstraction to input observations over time, this technique may allow the hidden state at each level to function on distinct timelines. Researchers increased SLSTM prediction accuracy by combining it with other machine learning methods [Liu et al., 2020]. Stacked LSTMs are now a stable technique and can be a more suitable method for solving challenging sequence prediction problems, such as financial markets. When predicting the next state of a time series, most researchers simply evaluate one variable at a time. When it comes to stock prices, however, the production might be influenced by a variety of variables. The dataset must be sufficiently large to allow the algorithm to make more generic decisions. Although SLSTM is capable of dealing with a large number of input variables, the performance of the entire system can be improved by combining regression analysis with SLSTM and data augmentation. The regression analysis has a stronger nonlinear approximation capacity and data augmentation for increasing the size of the dataset, even though data augmentation is less widely used in the field of time-series data. Thus, by experimenting with different combinations of ML, data augmentation, and SLSTM, a unique strategy for better time series analysis may be studied to boost the efficacy further.

As a result, this study provides an enhanced DL approach capable of predicting the relatively more accurate stock market and sales data, also known as time-series data. A revolutionary financial time series prediction methodology, named Poly-linear Regression with Augmented Long Short Term Memory Neural Network (PLR-ALSTM-NN), is proposed in this work. Poly-linear regression (PLR) is employed in the new structure to

## *2. POLY-LINEAR REGRESSION WITH AUGMENTED LONG SHORT TERM MEMORY NEURAL NETWORK: PREDICTING TIME SERIES DATA*

---

develop a strong association between the output and multiple input parameters. Following the establishment of the association, the dataset is filled with noise to enhance its size, which aids the network's generalisation capacity. To create the prediction, the enhanced data is passed into some stacked layers of LSTM-NN.

This chapter compared and contrasted numerous DL techniques, notably SLSTM, SLSTM with Linear Regression (SLSTM-LR), SLSTM with Poly-linear Regression (SLSTM-PLR), and BiSLTM, with varied days ahead of prediction, mostly on financial time series prediction, to verify the suggested DL methodology. PLR-ALSTM-NN is primarily created and validated using three sets of data from financial time series. It is also deployed on a supply chain dataset to demonstrate the approach's general character. GRU, RNN, CNN, basic LSTM, RF, and the ideal ARIMA model are also compared to the model. Compared to other competitive algorithms, the suggested technique performs better for long-term forecasts and generates equivalent results for short-term predictions, according to the detailed comparative analysis. As a result, long-term financial market investors who want to maximise profits, in the long run, may benefit from this technique. On a bigger scale, the recommended strategy would encourage long-term investment and aid a country's economic progress. Investors will have more faith in a firm in the long run if they earn a long-term profit by buying and selling its shares. Consumers will be excited about the company as well. Consumers, investors, and businesses will be encouraged to recruit more people, resulting in lower unemployment. A greater workforce will boost a country's GDP in the long run.

### **2.1.1 Arguments for Utilising Stock Market Data for Analysis and Prediction in the Context of Supply Chain Sales Forecasting**

- Similarity in Time Series Data: Stock market data and sales data both share fundamental characteristics as time series data. In both cases, patterns and trends emerge over time, influenced by various external factors. The techniques and algorithms used to analyze and forecast time series data are applicable to both domains. Therefore, methodologies proven effective in stock market prediction can be lever-

## *2. POLY-LINEAR REGRESSION WITH AUGMENTED LONG SHORT TERM MEMORY NEURAL NETWORK: PREDICTING TIME SERIES DATA*

---

aged to enhance the accuracy of sales forecasting.

- Dynamic and Multifaceted Nature: Stock markets are dynamic and multifaceted, responding to a multitude of variables, such as economic indicators, investor sentiment, geopolitical events, and corporate performance. Similarly, sales data is influenced by various factors, including consumer behaviour, economic conditions, marketing strategies, and seasonal trends. The ability to capture and predict complex dynamics in stock markets can provide valuable insights into understanding and forecasting sales trends, especially in volatile or uncertain market conditions.
- Pattern Recognition and Forecasting: Advanced time series forecasting algorithms used in stock market analysis excel at pattern recognition and prediction. These algorithms can identify subtle patterns and trends in historical data, which is crucial for anticipating future movements. The same principle applies to sales forecasting, where recognising patterns in past sales data can lead to more accurate predictions of future demand.
- Risk Mitigation: Stock markets are known for their inherent risks and uncertainties. By applying techniques from stock market prediction to sales forecasting, businesses can better anticipate and mitigate risks associated with demand fluctuations, market volatility, and unexpected events. This proactive approach to risk management can lead to cost savings and improved supply chain resilience.
- Holistic Data Analysis: Combining insights from both stock market and sales data can lead to a more holistic understanding of economic conditions and consumer behaviour. This broader perspective can inform strategic decision-making, such as inventory management, production planning, and marketing strategies, resulting in more efficient supply chain operations.

In summary, while stock market data and sales data may appear dissimilar on the surface, their shared characteristics as time series data, dynamic nature, and potential for pattern recognition makes them interrelated in the context of forecasting. Leveraging analytics

## *2. POLY-LINEAR REGRESSION WITH AUGMENTED LONG SHORT TERM MEMORY NEURAL NETWORK: PREDICTING TIME SERIES DATA*

---

and forecasting techniques from one domain to the other can lead to improved predictive accuracy, better risk management, and enhanced overall supply chain efficiency.

Hence, contributions stemming from this work are multi-fold, such as:

1. Using a mix of statistical regression analysis and SLSTM, an advanced DL model is introduced in this work. Statistical regression produces a strong association between the input and output data, and SLSTM's short-term and long-term memory components allow it to retain short and long-term patterns of prior data selectively. Synthetic training data is manufactured alongside actual data to increase the proposed DL model's generalisation capabilities on test data. The accuracy of time series data forecasting is improved as a result of this enhancement.
2. To acquire the highest potential expected results, the entire model has been fine-tuned using the optimal mix of parameters. For four stock market datasets, extensive validation of the proposed model was performed in contrast to contemporary ML and DL techniques in the literature to demonstrate its acceptable performance in financial market engineering.
3. Moreover, the proposed model is implemented using time series supply chain data (non-financial market data) in order to demonstrate its general behaviour in different sectors (i.e., predicting sales data).

The rest of the chapter is organised as follows. Section 2.2 provides a comprehensive literature survey on the current work related to DL and its application to time-series prediction. Section 2.3 presents a detailed outline of the methodology. Using a design of experiments (DOE) based Taguchi process, a suitable combination of hyper parameters is found, and the model with that parameter setting is validated in Section 2.4 against recently published literature. Section 2.5 shows the utilization of time series supply chain data to predict sales data using the proposed methodology. Having conducted several experiments on different datasets, the accuracy measures obtained from them are used to carry out some statistical tests, and the results are presented in Section 2.6. Finally, some

advantages and disadvantages of the proposed method are discussed in Section 2.7, with a Section 2.8 discusses the practical usefulness of the model, followed by the summary in Section 2.9.

## 2.2 Related Work

DL has been a key player in the classification and prediction domains for a long time. Its primary uses include speech recognition [LeCun et al., 2015], text classification [Zhang et al., 2015], translation [Singh et al., 2017], correction [Alkhatib et al., 2020], and object recognition [Wang et al., 2021b]. It's also utilised in time-series forecasting. Wen et al. [2020], Wang et al. [2022], Jalali et al. [2021], Kamara et al. [2022] are only a few of the researchers that have been working on this. The impact of AI on financial market predictions and its downsides and limits will be discussed in this section. A review of the research gaps and potential remedies is also provided below.

### 2.2.1 Contribution of ML/DL/Hybrid Forecasting Approaches to Financial Market

Many academics are experimenting with various methodologies in order to improve the accuracy of ML/DL predictions. Di Persio and Honchar [2016] conducted several experiments on financial time series prediction using LSTM, and the results were compared to convolutional neural networks (CNN). Bao et al. [2017] took a different strategy, combining Stacked Autoencoders (SAEs) with LSTM to increase performance accuracy. The authors used a stacked auto-encoder (SAE) to de-noise the financial market data before training the LSTM network to make short-term predictions. Support vector machine (SVM), NN, and various combinations of algorithms are among the shallow ML models commonly utilised in current stock predictions. According to Hsu et al. [2016], machine learning can anticipate financial markets more accurately than other approaches such as Naive Bayes and Decision Trees. The market's maturity, input factors, timing, and forecasting method

## *2. POLY-LINEAR REGRESSION WITH AUGMENTED LONG SHORT TERM MEMORY NEURAL NETWORK: PREDICTING TIME SERIES DATA*

---

all have an impact on the market's projection. Similarly, Niaki and Hoseinzade [2013] employed a NN model to predict the S&P index using 27 various economic indicators such as oil, gold, exchange rates of different currencies, and returns of the S&P 500 index on different days. The exchange rates between the US dollar and three major currencies, including the British pound, Canadian dollar, and Japanese yen, have significantly impacted the S&P 500's daily direction. Their findings revealed that ANN with the most influential characteristics could predict the daily direction of the S&P 500 much better than conventional models.

An alternative decision tree-based forecasting approach was proposed by Ballings et al. [2015], where the authors standardised ensemble methods like Random Forest, AdaBoost and Kernel Factory to find the direction of stock prices but not the exact stock prices for one year ahead of prediction. Results were compared against single classifier models (e.g., NNs, Logistic Regression, Support Vector Machines and K-Nearest Neighbor). In terms of performance, Random Forest (RF) was at the top, followed by SVM, Kernel Factory, AdaBoost, NNs, K-Nearest Neighbors and Logistic Regression.

When compared to NN, simple ML methods like PCA can provide similar results. Zhong and Enke [2017] used 60 distinct economic factors and three different dimensionality reduction approaches, including PCA, Kernel PCA, and rapid robust PCA coupled with NN, to predict the S&P index. PCA surpassed the competition with a 57 percent accuracy rate. In the field of NNs, researchers have recently undertaken additional comparison studies between NNs and the ARIMA model used for stock price time series forecasting. Selvin et al. [2017] used RNN, LSTM, and CNN to estimate the stock values of three distinct businesses (Infosys, TCS, and Cipla) and compared the results to the linear forecasting technique ARIMA. In terms of error %, the NNs generated similar results, although ARIMA had the greatest mistake percentage. The authors made no mention of the ARIMA parameter settings that were used for predicting. Siami-Namini et al. [2018] undertook a similar sort of comparative investigation, comparing LSTM with ARIMA for stock market forecasts, however, this time, the authors revealed the characteristics of each network they used to create predictions. Even if LSTM beats ARIMA, the authors indicate that their ARIMA

---

## *2. POLY-LINEAR REGRESSION WITH AUGMENTED LONG SHORT TERM MEMORY NEURAL NETWORK: PREDICTING TIME SERIES DATA*

---

model is not the best. Khashei and Hajirahimi [2019] used ARIMA with NN such as MLP to create better prediction models for financial time series forecasting.

On another approach, Li et al. [2017] improved the prediction performance of LSTM by another hybrid approach that incorporated naïve Bayes to it. An innovative idea of blending the auto-encoding technique with LSTM and LSTM with an embedded layer for predicting stock market data was proposed by Pang et al. [2018]. The suggested methodology works better in predicting the Shanghai A-share index but the model's deficiency was that it failed to utilise the full-text information, e.g. news from historical input data. A similar type of model is also suggested by Wang et al. [2019], who have predicted the volatility of stock markets using a combination of CNN and LSTM sourcing data from news headlines. Likewise, the text mining technique was also adopted by some researchers to create more comprehensive datasets by extracting text from news and social media related to the stock market [Colasanto et al., 2022, Ronaghi et al., 2022], hoping to find the influence of investor's emotions or the effect of a pandemic such as Covid 19 on the financial market. Then using different forecasting methods such as DNN or hybridisation of LSTM with other approaches, the scholars tried to predict stock price movement. Using LSTM networks, Fischer and Krauss [2018] predicted out-of-sample directional movements of S&P500 stocks from 1992 till 2015. The authors proposed that LSTM networks performed better than traditional approaches like DNN, RF, etc. On another approach, Althelaya et al. [2018] used NN on stock market engineering and showed that a stacked architecture could enforce a deeper analysis of training data to model more sophisticated data patterns while making long and short-term predictions of financial market data. Thus it leaves space for researchers to work on developing a more robust hybrid generic approach by combining LSTM with other approaches and fine-tuning it to enhance the overall forecasting accuracy

### **2.2.2 Drawbacks and Limitations of Existing Research**

In a comparison of SVM and ANN for stock market prediction, it was discovered that SVM outperformed ANN in detecting the percentage change between the opening and closing prices of each day. There was no comparative analysis because the authors only looked at one dataset (BIST: Borsa Istanbul BIST 100 Index). Their approach excelled at detecting positive changes rather than negative ones. Nabipour et al. [2020] did a similar comparison investigation on the Tehran stock market with different tree-based forecasting models, such as RF, Adaboost, XGBoost, and NNs, and discovered that LSTM (from the NN group) was the highest-performing strategy when compared to other systems. The authors also employed only one dataset and a fixed set of parameters that they got by trial and error, whereas a more systematic approach to identifying the optimal parameter setting on a large number of nonlinear time-series data might yield more promising findings. Furthermore, the authors have not investigated the impact of incorporating additional analytical tools into their methodologies. As a result, there are still more paths for improving model performance accuracy that is yet to be explored.

Researchers are spending a lot of work on developing hybrid techniques [Castán-Lascorz et al., 2022], but their focus is mainly on short-term forecasting, with little attention paid to the models' effectiveness over longer forecasting horizons. Siami-Namini et al. [2018] and Khashei and Hajirahimi [2019] performed comparative studies with LSTM, ARIMA, and hybrid LSTM-ARIMA, where the authors evaluated the models' short-term performance while making long-term forecasts. Furthermore, the performance of ARIMA and hybrid models may be improved, opening up new study opportunities for academics.

Following the hybridisation trend, Lu et al. [2021] produced another hybrid forecasting system, in which the authors employed a mix of CNN, BiLSTM, and attention modules to predict stock market data. The hybridization of diverse methodologies is still going on as scholars attempt to forecast the closing values of cryptocurrencies like Bitcoin, Litecoin, Tether, Ripple, and others by merging ANN with Rao algorithm-based optimisation. In the studies, the authors used parameters gained through the trial and error technique to

train their systems unorganised and inefficiently. Extending the forecasting horizon from short to long term might entice long-term investors to participate. Although improving the short-term forecasting accuracy of existing ML/DL or hybrid approaches by incorporating text-mining or data-mining techniques has been discussed previously [Colasanto et al., 2022, Ronaghi et al., 2022], there is little evidence in the literature that evaluates how much text-based information can influence stock price in the long run. Furthermore, most current analyses are undertaken over a limited time horizon, with a single dataset, or with a set of preset parameters. A systematic approach to finding the best parameter across numerous datasets and forecasting horizons might lead to more intriguing research directions.

### **2.2.3 Summary of Research Gaps and Possible Solutions**

This section will provide an overview of the flaws identified in the literature and some potential remedies. In light of the literature already presented in Subsections 2.2.1 and 2.2.2, it can be seen that the majority of researchers used more mature stock markets, such as the S&P 500 and Nasdaq indices, whereas the minority used immature financial markets, such as the Shanghai stock market, where prediction is thus more difficult. Furthermore, a significant amount of financial market engineering has been done using only one indication out of a collection of all indicators. Others use the closing price as their research target [Khashei and Hajirahimi, 2019, Nabipour et al., 2020, Kamara et al., 2022], while some anticipate the opening price using past opening prices [Bukhari et al., 2020]. However, there may be a relationship between them. For example, the closing price of a stock may be influenced by its opening price, highest price, and/or lowest price. Linear or poly-linear regression can be used to describe the relationship between the dependent variable and one or more independent variables to solve these concerns. According to Arias et al. [2014], only linear regression models failed to forecast time series, whilst non-linear models performed better. Fitting a linear regression line on a graph with a low error value is extremely challenging. As a result, polynomial regression may be a superior alternative for fitting a polynomial line with the least amount of error.

## *2. POLY-LINEAR REGRESSION WITH AUGMENTED LONG SHORT TERM MEMORY NEURAL NETWORK: PREDICTING TIME SERIES DATA*

---

Although much effort has been dedicated to time series forecasting, the noted algorithms suffer from the following limitations and challenges.

1. The presence of outliers can seriously affect the results of the nonlinear analysis since the system alone is too sensitive to the outliers.
2. Unfortunately, there are fewer model validation techniques for detecting outliers in nonlinear regression than for linear regression, making time series analysis difficult.
3. Researchers are mostly focused on short-term forecasting of stock market data, with only a few studies focusing on long-term analysis. As a result, there aren't many well-known multivariate analytic algorithms capable of providing lengthy days-ahead predictions, which might be useful to long-term investors in practice.

### **2.3 Proposed Methodology**

A combination of regression and Stacked layers of LSTM, along with the incorporation of data augmentation, can be an effective way of making time-series predictions. Apart from using different activation functions and optimizers, an early stopping technique is also used to make more reliable predictions by avoiding over-fitting and reducing valuable computational resources. The proposed model's different steps are presented using a flow diagram given in Figure 2.1 followed by a brief explanation of each step.

The experiment is initially performed using stock market data while using linear and/or poly-linear regression on the data. The estimated closing price of the stock market from regression alongside the true closing price from one dataset. The data is split into two parts: training and testing. After that, the training data is augmented with noise to increase the size of the overall training set. Both the noisy augmented and test sets are then organised into a sequential pattern using a window size of 10. This data restructuring will be further discussed later in Section 2.3.6. That sequential noisy augmented train set is used to train some stacked layers of LSTM for a certain number of epochs. Using a

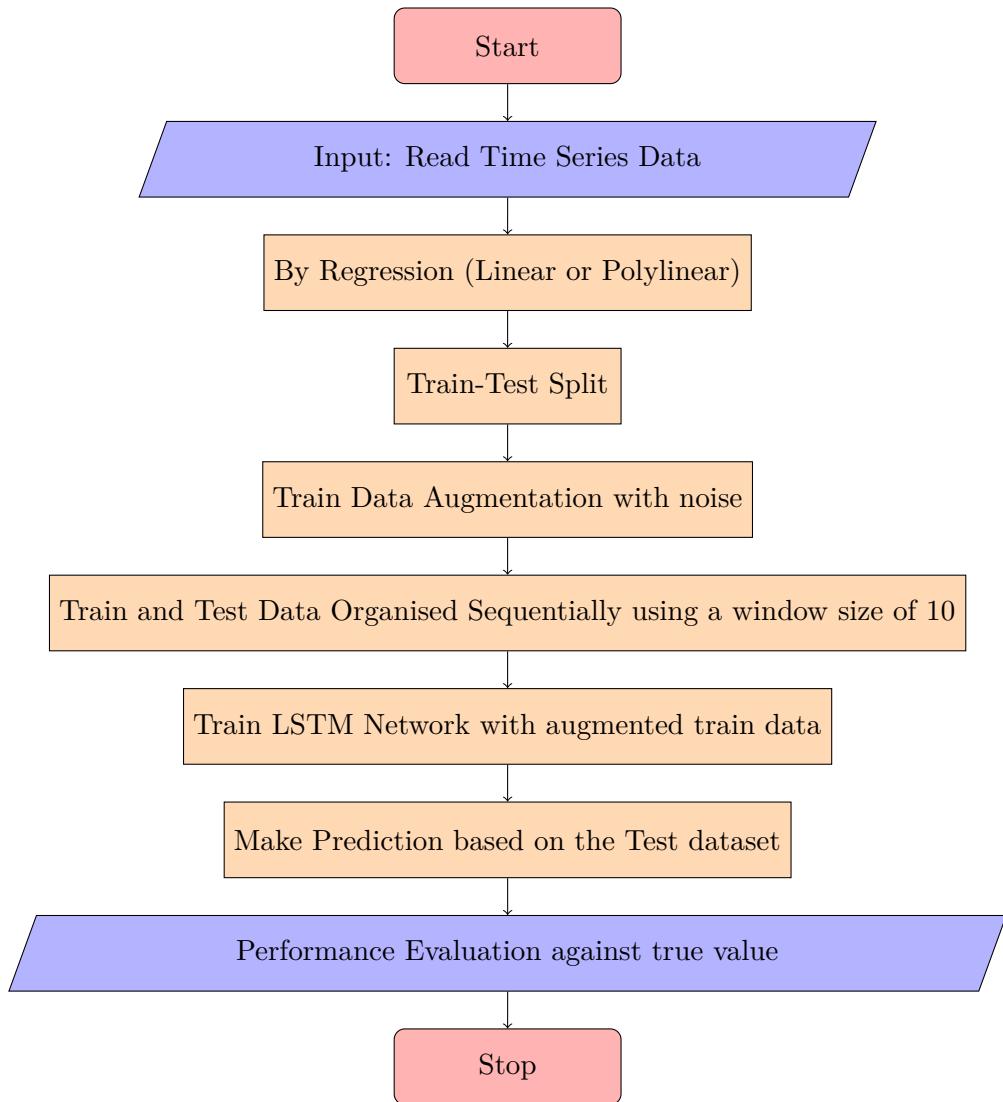


Figure 2.1: Proposed Model

technique called early stopping, the training of the stacked LSTM Network is stopped when the error gets to a minimum point. Once the training is done, the test set is used to make predictions of stock prices and this against the true values is used to evaluate the model.

### **2.3.1 Data Collection**

The proposed model is developed, tuned and validated on four datasets, while a fifth dataset is employed only to show its implementation. While developing the model, historical data of S&P500 (Standard and Poor 500 Index) was initially downloaded from the Yahoo finance website from 01 Jan 2010 to 30 Nov 2017. It was then extended to 27 Jan 2021 for further use. The model is validated on a secondary S&P500 dataset collected from the same website with the duration from 3 Jan 1994 until 23 Oct 2006. S&P500 is an important dataset that can be used as a benchmark as it is a leading indicator for 500 top-traded companies in the US market. For further validation against literature, the world's eighth-largest stock exchange index, Shenzhen Stock Exchange (SZSE) dataset (<https://finance.yahoo.com/quote/600422.SS/history?p=600422.SS>) is used from 1 Jan 2012 till 31 Jan 2017. From all the datasets, 4 price columns of each day are extracted for the purpose of experimentation, namely: Open, High, Low and Close. The performance of the proposed model is compared against standards from the literature, which is why some of the historical datasets are a little old, but the model has also been exposed to a recent dataset.

Finally, to demonstrate its usefulness, the model is implemented on two more datasets (i) the supply chain sales-related dataset and (ii) Dow Jones Industrial Average (DJI), which are thus the fourth and fifth datasets downloaded from the UCI Machine Learning repository ([https://archive.ics.uci.edu/ml/datasets/Sales\\_Transactions\\_Dataset\\_Weekly](https://archive.ics.uci.edu/ml/datasets/Sales_Transactions_Dataset_Weekly)) and Yahoo Finance (<https://au.finance.yahoo.com/quote/^DJI/history?p=~DJI>) websites respectively. The characteristics of the sales data are multivariate, and time-series, and these are similar to those used to train the proposed network. It consists of no missing values and has 811 instances with 53 attributes. Dow Jones Industrial Average (DJI) data are also characterised as a multivariate time series and exhibit similar characteristics to other previously used financial times series data in having open, low, high and close prices of the stock market. This particular dataset ranges from 19 Jan 2010 to 15 Jan 2021. The different time-series observations used in this chapter are mentioned in Table 2.1, and their statistical description will be discussed later in Section

### 2.3.3.

The intention of validating this proposed model with these datasets is to prove its usefulness to even predict supply chain data (i.e., sales data) and recent DJI data. According to the Trading Economics website, the DJI stock market index is one of the highly followed financial market indexes by investors, financial professionals and the media. It measures the stock performance of 30 large companies listed on stock exchanges in the United States. The proposed prediction methodology was implemented in Python using Keras, an open-source package for deep learning with a TensorFlow backend. A detailed description of all the datasets used in this chapter is given in Tables 2.1 and 2.2. All training and experiments are executed on a standard workstation (16 GB RAM, 3.20GHz Intel® Core™ i7-8700 CPU, graphical unit AMD RADEON WX 3100 with 4GB memory).

### 2.3.2 Regression

Linear Regression is a prominent statistical modelling technique that can be used to analyse market data. It allows the analyzer to consider multiple variables (Open Price, Daily High and Low Prices) that affect the predicted quantity (that is the closing price in the first two datasets). Initially, a linear regression model is applied to the entire dataset and the coefficients of the Open, High and Low Prices are calculated as follows:

$$\text{ClosePrice} = \alpha + \beta_1 \text{OpenPrice} + \beta_2 \text{HighPrice} + \beta_3 \text{LowPrice}$$

where  $\alpha = 0.5028$ ,  $\beta_1 = -0.6145$ ,  $\beta_2 = 0.8320$  and  $\beta_3 = 0.7826$

Using the coefficients, the closing price of the stock market is calculated for each day and is named as Price\_Prime. In linear regression, the closing price is assumed to be determined from the open, high and low stock prices. The latter are independent variables, while the former, closing price, is considered the dependent variable. The constant and other coefficients, such as  $\alpha$ ,  $\beta_1$ ,  $\beta_2$  and  $\beta_3$ , are chosen in such a way so that the difference between the true closing price and Price\_Prime, calculated from the regression equation, produces the minimum squared deviation. It can be stated that this calculated price

contains information on the other dependent stock prices. Then the actual dataset is now reduced to only two columns, thus reducing the overall complexity of the data. They are Price\_Prime, which is the closing price calculated using a linear regression model, and the actual closing price of each day. These help us increase our LSTM model's computational efficiency by limiting the dataset to two variables, Price\_Prime and Actual Closing Price.

Linear regression creates a linear relationship between the dependent and independent variables. However, polynomial regression may be another suitable option as it is capable of making more complex relationships between variables. It includes a higher power of one or more predictor variable(s) as well as the interaction effects of two or more variables. In the proposed model, the original stock market features are converted into higher-order terms (degree = 4) using PolynomialFeatures from scikit-learn. The usual linear regression is used as before to reduce the overall size of the dataset. Both these approaches provide the opportunity to add noise to the data, which enables more efficient training of the model by reducing the features while keeping the effect of the independent variables on the dependent ones.

### **2.3.2.1 Data pre-processing: data scaling using min-max scalar**

A normalization function, i.e. min-max scalar, has been applied for scaling the two remaining features of the data. This is done so that it can be easily handled if there are any sigmoid activation functions in the output layer of the model. Therefore, the normalization function uses the following formula given in equation 2.1:

$$x_{norm} = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (2.1)$$

### **2.3.3 Data Split**

The primary S&P500 dataset was collected from Yahoo Finance from 4 Jan 2010 till 29 Nov 2017 and was split into two parts, train and test, in the ratio of 80:20 respectively. It

## 2. POLY-LINEAR REGRESSION WITH AUGMENTED LONG SHORT TERM MEMORY NEURAL NETWORK: PREDICTING TIME SERIES DATA

---

was then extended till 27 Jan 2121 to observe the impact of the current global pandemic on prediction accuracy. The secondary dataset, which is also from S&P500, has been split into the same two segments but in the ratio of 77.806: 22.194. These datasets are split so that match the ones stated in the literature. This second dataset is split slightly differently in relation to other datasets, which allows us to make a comparative analysis with other approaches from literature, where researchers used 77.806% S&P500 data for training purposes and 22.194% for testing. As a result, the model's prediction performance can be compared against some benchmarks from the literature and its accuracy can be evaluated. The third dataset, containing only 52 weeks of data, is split in the ratio of 80:20 for simplicity. The proposed approach is further tested on another latest SZSE and DJI dataset, split in the ratio of 80:20, with the intention of showing its competitiveness with other approaches. Table 2.1 shows the different datasets used in this chapter alongside corresponding timesteps used for training and testing the accuracy of the models. The reason behind choosing some different versions of the same data is to observe the impact of the current global pandemic on the prediction performance of the model. Descriptive statistics of the target variable of each of the datasets is also shown in Table 2.2.

Table 2.1: The different time series observations used in this chapter

	Dataset	Train	Test	Train Time Steps	Test Time Steps
1.(a)	S&P500	4 Jan 2010 - 2 May 2016	3 May 2016- 29 Nov 2017	1593	399
1.(b)	S&P500 (pre-covid)	4 Jan 2010 - 28 Dec 2017	29 Dec 2017- 31 Dec 2019	2012	504
1.(c)	S&P500 (Covid)	4 Jan 2010 - 31 Dec 2019	2 Jan 2020- 27 Jan 2021	2516	270
2.	S&P500	3 Jan 1994 - 17 Dec 2003	18 Dec 2003- 20 Oct 2006	2510	716
3.	Sales Data	-	-	41	11
4.(a)	DJI (pre-covid)	19 Jan 2010 - 2 Jan 2018	3 Jan 2018- 31 Dec 2019	2004	502
4.(b)	DJI (Covid)	19 Jan 2010 - 31 Dec 2019	2 Jan 2020- 15 Jan 2021	2506	263
5.	SZSE (Code 600422)	1 Jan 2012 - 22 Jan 2016	23 Jan 2016- 31 Jan 2017	985	247

Table 2.2: A statistical description of the dataset

Dataset	Count	Mean	Std	Min	Max	Train : Test Ratio
1.(a)	1992	1736.21	430.90	1022.58	2627.04	80:20
1.(b)	2516	1962.61	588.91	1022.58	3240.02	80:20
1.(c)	2786	2087.80	685.90	1022.58	3855.36	90:10
2.	3226	1008.15	298.20	438.92	1527.46	77.8:22.2
3.	52	37.15	7.77	23.00	55.00	80:20
4.(a)	2506	17634.60	5138.34	9686.48	28645.26	80:20
4.(b)	2769	18528.05	5668.92	9686.48	31097.97	90:10
5.	1232	12.77	3.08	6.03	25.03	80:20

## *2. POLY-LINEAR REGRESSION WITH AUGMENTED LONG SHORT TERM MEMORY NEURAL NETWORK: PREDICTING TIME SERIES DATA*

---

Statistical information of each of the datasets (count, mean, standard deviation, min value and max value) is shown in Table 2.2. Most of the datasets are split in the ratio of 80:20 except for a few, as stated in the table. From the table, it can be observed that Dataset 1.(c) and 4.(b) have a greater standard deviation (685.9 and 5668.92) when compared with their predecessors (Dataset 1.(b) and 4.(a)) where each has standard deviations of 588.91 and 5138.34 respectively. This increase in variance is due to the uncertainty imposed by the global Covid19 pandemic on the stock market.

### **2.3.4 Data Augmentation**

The amount of training data is increased by generating modified versions of the samples in the actual training data. This technique is called data augmentation. How the data is augmented depends on the type of the data and the nature of the task. In the image classification task, images can be rotated, cropped, scaled, translated or flipped to create multiple versions to increase the overall training set. In the case of time series prediction, random noise can be added to augment the training dataset, thus expanding its size. Since a large amount of training data is needed to train a neural network, data augmentation is used to increase the size of such datasets. In this research, a small amount of Gaussian white noise is used to augment the training data. After experimenting with different sizes, mean and standard deviations of the noise, it was found that using the noise of size 2x to triple the size of the overall training set with mean = 0 and standard deviation of 0.01 provided the best result. Every time the training samples were fed to the model, the noise was added to the input variables to make a larger training set than the original set for the model to train on. For each day of the training data, two folds of separate random noisy data were created and added to the training data to increase the entire training data to 3 times that of the original dataset. This is done to minimise the chance of over-training the model.

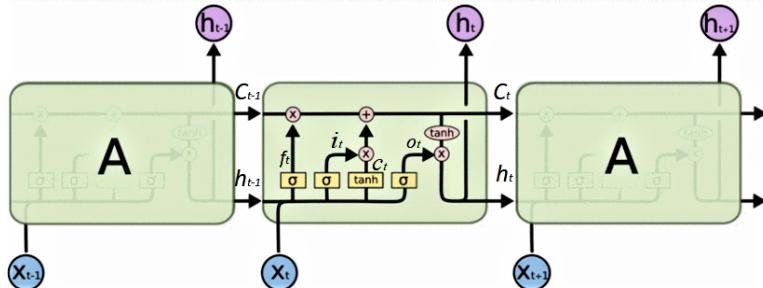


Figure 2.2: LSTM's architecture [Olah, 2015a]

### 2.3.5 LSTM Architecture

Figure 2.2 represents how each LSTM cell are connected together to form the LSTM architecture, with each line carrying some particular information, as explained by Equation 2.2, 2.3, 2.4, 2.5, 2.6 and 2.7. Both of these images are collected from Olah [2015a].

In RNN, the network learns by a technique of gradient descent and this gradient becomes too small to have any influence on the learning of the network when it trains over a large sequence of cells since the error propagates throughout the network. This problem is called Vanishing Gradient Descent and LSTM overcomes this issue as the architecture has a memory line at the top and a bottom line to represent the output i.e.  $h_t$  and  $x_{t-1}$  represents the previous block whereas,  $x_{t+1}$  represents the next future block where  $x_t$  becomes the input of that future block. Through back-propagation, the learning process approximates the weights and lets the data be stored or deleted in the cells. According to Figure 2.2 the transition equations of LSTM are stated below.

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + V_i c_{t-1}) \quad (2.2)$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + V_f c_{t-1}) \quad (2.3)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + V_o c_t) \quad (2.4)$$

$$\tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1}) \quad (2.5)$$

$$c_t = f_t^i \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (2.6)$$

## *2. POLY-LINEAR REGRESSION WITH AUGMENTED LONG SHORT TERM MEMORY NEURAL NETWORK: PREDICTING TIME SERIES DATA*

---

$$h_t = o_t \odot \tanh(c_t) \quad (2.7)$$

where  $i_t$  denotes the input gate and  $o_t$  denotes the output gate. The forget gate, memory cell, and hidden state are denoted by  $f_t$ ,  $c_t$ , and  $h_t$ , respectively [Graves and Schmidhuber, 2005].

It can also be noted that the result is impacted by the output coming from its previous block. Each block has three inputs and two outputs, and all of these are vectors, i.e. all these variables contain arrays of values rather than one value. The point-wise operations are like values, and the sigmoid function mainly governs their opening or closure. If open, information is allowed to pass through freely. The value is either added to the memory or not, depending on the decision of the operation and if so is passed to the next layer for operation. The point-wise operation contains values between zero and one, and the output valve is in the last point-wise operation where it decides how much of the memory reaches the output. There are various LSTM architectures available. However, the basic functionalities are the same.

This is usually defined empirically regarding the number of neurons in the hidden layers. The number of units in each of the layers are carefully chosen as too many units could potentially lead the network to memorise the training data (overfitting), instead of extracting the general characteristics that will allow generalization, nor a very small number, which can force the network to spend too much time trying to find an optimal representation. Since there are no fixed rules regarding the relationship between the number of neurons and the number of hidden layers with respect to the inputs and outputs; that is why in the proposed model, the final experimental analysis is done using two LSTM layers, each having a fixed number of units selected from the following units: 4, 8, 16, 32 or 64. While experimenting with 4 neurons, all the LSTM layers have 4 units and when using 8 neurons, all layers have 8 neurons and so on.

In the first LSTM layer, the proposed model uses a hard sigmoid as the activation function, and in the second layer, tanh or hyperbolic tangent is used as the activation function. These activation functions are prominent transfer functions used in time-series analysis

that map the input of one neuron to its output. These functions are used to determine the output of neural networks, like yes or no, and they map the resulting values between 0 and 1 or -1 and 1, depending on the type of function used. In the final layers of the model, a dense layer with one unit is used to give the output. Its purpose is to combine the outputs of all the LSTM layers into a single value, which is the closing price of the stock market. In the dense layer, a linear activation function is used as it is considered the best activation function by researchers for making time-series predictions. All these above-mentioned parameters were chosen after fine-tuning the model using different combinations of the different parameters and the method of finding the optimum parameter combination is discussed in the parameter tuning section.

In Keras, a hard sigmoid is implemented as,

$$f(x) = \max(0, \min(1, x * 0.2 + 0.5)) \quad (2.8)$$

and according to the Keras documentation [Abadi et al., 2015], it is faster than the regular sigmoid function. Similarly, tanh is given as

$$\tanh(x) = \frac{\sinh(x)}{\cosh(x)} = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.9)$$

and its value ranges from -1 to 1. The linear activation has a simpler representation and can be expressed as

$$f(x) = x \quad (2.10)$$

In a neural network, optimizers are algorithms that can change the weights, biases, and or learning rate to reduce loss while training the network and providing the most accurate results possible. During the compilation (training phase) of LSTM architecture, the Adam (Adaptive Moment Estimation) optimizer is used with mean squared error as the loss function. In the proposed model, experimental analysis shows that Adam provides better

## *2. POLY-LINEAR REGRESSION WITH AUGMENTED LONG SHORT TERM MEMORY NEURAL NETWORK: PREDICTING TIME SERIES DATA*

---

convergence than RMSprop and SGD (Stochastic Gradient Descent), which is why Adam optimizer is used during the training phase.

While training a NN, a problem that often arises is related to choosing the right number of training epochs. If too many epochs are used, it leads to over-training and the network fails to generalise and make decisions when exposed to a new dataset i.e the test dataset and if too little is used, it results in an under-fit model. Early stopping is a technique adopted from the Keras library that allows a user to specify an arbitrarily large number of training epochs and automatically stops the training once the performance of the model stops improving on the validation set, i.e. in this case, the test set. To avoid overfitting, early stopping [Prechelt, 1998] is adopted and incorporated into the model. Here, an arbitrary training epoch of 150 is chosen with the patience parameter set at 50. After getting to the point of best accuracy while having minimum loss, the model will wait for 50 more epochs hoping to find some improvement. If it does not, then it will retrieve the best weights that produced the best result to make future predictions. This also helps to reduce convergence time and saves important computational resources.

### **2.3.6 Data Structuring and Train the LSTM Network**

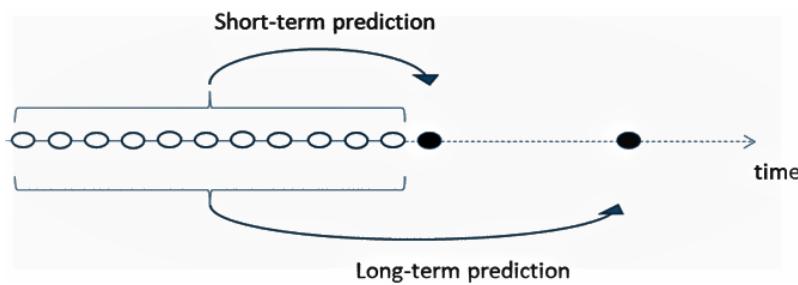


Figure 2.3: Illustration of short-term and long-term prediction

The time series analysis of the data is structured in a sequential pattern using a sliding window technique before feeding it into the network. In the experiments, a window of size 10 (working days of two consecutive weeks) is used to predict one-day ahead (short term) and 30-day ahead (long term) closing stock prices. Figure 2.3 illustrates the predictors (input values) and the target (output) while making short-term and long-term predictions.

An input vector of 10-time steps and two columns of noisy augmented training data is formed. The columns consist of the true closing price and the price obtained by passing it through the regression analysis. The output is termed price prime. Using these two columns of information, data is structured as the independent variables (X train) (10 days of True Closing Price and Price Prime) and the dependent variables (Y train), which is either the 11<sup>th</sup> days (short term) true closing price or 41<sup>st</sup> days (long term) closing price depending on the type of predictions the model is making. The window is then shifted one step down, and this time it contains information from the 2<sup>nd</sup> time step until the 11<sup>th</sup> time step of the independent training data and the 12<sup>th</sup> day's data as the output of that particular segment while making short term predictions. The model can also be used to predict anything between 1 and 30 days ahead into the future.

### **2.3.7 Testing the Trained Network**

Once the training is done, the test data are used to make predictions. The predicted values are de-normalised using the inverse min-max scalar function to get the output in the range of the true values. The proposed model was initially based on a Stacked LSTM network, and regression analysis was added to improve the model's performance. From there on, its performance was further enhanced by the addition of the data augmentation technique. As for performance evaluation parameters, Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE) and Mean Absolute Error (MAE) were calculated and compared against the existing literature using de-normalised predicted values. The gradual improvement in performance will be discussed in the result section in comparison to the existing literature.

#### **2.3.7.1 Setting Seed**

Due to the randomness of neural networks, it is difficult to get a consistent result for the same setting as the randomness inside the hidden layers changes the result at different iterations for the same setting. In order to get consistent output, seed values between

1000 and 1029 were chosen, and all experiments were conducted several times for different parameter combinations, and then the average was calculated. For parameter tuning and statistical testing, 10 Seed values were chosen between 1000 to 1009 in the first round, and then 5 seed values (1000 to 1004) were used in the second round and once the best setting is obtained, the final proposed model was evaluated by compiling it 30 different times using the 30 different seed values mentioned earlier.

### 2.3.7.2 Error Calculation

To evaluate trained models for usefulness, Root Mean Square Error (RMSE), Mean absolute percentage error (MAPE) or Mean Absolute Error (MAE) is calculated according to equations 2.11, 2.12 and 2.13 respectively. Although RMSE is better at showing bigger deviations than MAPE and MAE, in this chapter, all three are calculated for comparison.

Root Mean Square Error (RMSE)

$$RMSE = \sqrt{\left(\frac{1}{n}\right) \sum_{i=1}^n (\hat{y} - y_i)^2} \quad (2.11)$$

Mean absolute percentage error (MAPE)

$$MAPE = \left(\frac{100\%}{n}\right) \sum_{i=1}^n \left| \frac{|\hat{y} - y_i|}{|y_i|} \right| \quad (2.12)$$

Mean Absolute Error (MAE)

$$MAE = \left(\frac{1}{n}\right) \sum_{i=1}^n |\hat{y} - y_i| \quad (2.13)$$

where:

$\hat{y}$  is the predicted value,  $y_i$  is the true value and  $n$  is the number of observations in both the above mentioned equations.

The proposed approach, PLR-ALSTM-NN, is explained as a pseudo code in Algorithm 1.

---

**Algorithm 1** PLR-ASLTM-NN

---

**Input:** dataset, parameters(Neurons, Dropout, Optimizer, Regularizer, Layers, Epochs, Noise\_size, days\_ahead\_prediction, no\_of\_runs)

```

1: for nrons  $\in$  Neurons do
2:   for epoch  $\in$  Epochs do
3:     set_seed  $\leftarrow$  1000 (Initialise Seeds to get reproducible results);
4:     RMSE  $\leftarrow$  [ $\phi$ ]; MAPE  $\leftarrow$  [ $\phi$ ]; MAE  $\leftarrow$  [ $\phi$ ];
5:     for j = 1 : no_of_runs do
6:       dependent_target_variable  $\leftarrow$  polynomial_linear_regression (independent_target_variables);
7:       dataframe  $\leftarrow$  dependent_target_variable, true_target_variable;
8:       MinMaxScaler (dataframe);
9:       Train-Test Split (dataframe);
10:      Data Augmentation of Train Data(Noise_size);
11:      prediction_horizon (days_ahead_prediction): 1 for short term and 30 for long term;
12:      X_train, y_train, X_test and y_test  $\leftarrow$  Converting Test and Augmented Train Data into 3D
           data with multiple features and target output based on days_ahead_prediction;
13:      Train: LSTM_MODEL (X_train, y_train, nrons, Dropout, Optimizer, Regularizer, Layers, Epochs);
14:      predicted_values  $\leftarrow$  Test: LSTM_MODEL (X_test);
15:      Denorm_Predicted  $\leftarrow$  inverse_transform(predicted_values);
16:      Denorm_true  $\leftarrow$  inverse_transform(y_test);
17:      RMSE[j]  $\leftarrow$  root_mean_squared_error(Denorm_true, Denorm_Predicted);
18:      MAPE[j]  $\leftarrow$  mean_absolute_percentage_error(Denorm_true, Denorm_Predicted);
19:      MAE[j]  $\leftarrow$  mean_absolute_error(Denorm_true, Denorm_Predicted);
20:      seed_set=seed_set+1
21:    end for
22:    Average (RMSE, MAPE and MAE);
23:  end for
24: end for

```

---

## **2.4 Result Analysis and Model Validation**

This section first contains a detailed explanation of fine-tuning the model by Taguchi's Design of Experiment method to produce the best prediction accuracy. Then, the finely-tuned model is implemented on 5 different datasets (already described in Subsection 2.3.3). Its forecasting accuracy is compared and validated against existing models and, at times, against recent literature for both short and long-term prediction. While validating the performance against literature, PLR-ALSTM-NN was trained and tested for 30 runs using 30 different seed settings using Dataset 1.(a) and Dataset 2. For the other dataset, the model was tested for 10 runs using 10 different seed values. The results of the performance of the model for each of the dataset is thus presented in this section.

### **2.4.1 Parameter Tuning**

The proposed model has several vital parameters that need to be tuned to get the best available setting. For proper calibration of the algorithm, Taguchi's design of experiment (DOE) technique is used to reduce the required computational time and tests for identifying the best possible parameter combination for the proposed model. The combination of experimental parameters is shown in Table 2.3. The orthogonal array L27(3<sup>8</sup>) is employed since the number of parameters considered in this study is 8 and each has three levels. So, 27 treatments in the DOE are tested using the primary S&P500 dataset for 10 runs using 10 different seeds.

According to the orthogonal table, the trend of the selected parameters is stated and depicted in Table 2.5 and Figure 2.4. From Table 2.5, it can be noted that the Optimizer is the most significant parameter, followed by Dropout. Hence, a wise selection of these parameters is necessary to enhance the accuracy of prediction. In addition, regularizers and the number of LSTM layers also play a crucial role in the performance of the model. According to this analysis, a good choice of parameter is suggested as: Neuron Size = 16, Dropout = 0.0, Optimizer = Adam, Regularizer = No Regularizer, Number of LSTM

## 2. POLY-LINEAR REGRESSION WITH AUGMENTED LONG SHORT TERM MEMORY NEURAL NETWORK: PREDICTING TIME SERIES DATA

---

Table 2.3: Combination of parameter values.

Parameters	Parameter Level		
	1	2	3
Neuron size	16	32	64
Dropout	0	0.5	0.8
Optimizer	Adam	RMSprop	SGD
Regularizer	No Regularizer	L1	L2
Number of layers	2	4	6
Epochs	100	150	200
Regression level	Poly	Lin	No Reg
Noise level	1x	2x	3x

Table 2.4: Orthogonal table with Average RMSE and Standard Deviations.

Expt no.	Neuron Size	Parameters							Avg RMSE	Std Div
		Dropout	Optimizer	Regularizer	Number of Layers	Epochs	Regression Level	Noise Level		
1	1	1	1	1	1	1	1	1	41.930	22.546
2	1	1	1	1	2	2	2	2	117.671	15.527
3	1	1	1	1	3	3	3	3	172.475	20.347
4	1	2	2	2	1	1	1	2	334.567	85.949
5	1	2	2	2	2	2	2	3	214.968	78.097
6	1	2	2	2	3	3	3	1	628.583	190.085
7	1	3	3	3	1	1	1	3	736.648	3.193
8	1	3	3	3	2	2	2	1	734.556	2.517
9	1	3	3	3	3	3	3	2	718.509	4.130
10	2	1	2	3	1	2	3	1	118.928	11.192
11	2	1	2	3	2	3	1	2	146.737	5.535
12	2	1	2	3	3	1	2	3	212.942	35.809
13	2	2	3	1	1	2	3	2	194.138	17.413
14	2	2	3	1	2	3	1	3	697.890	16.727
15	2	2	3	1	3	1	2	1	677.521	10.806
16	2	3	1	2	1	2	3	3	307.735	40.935
17	2	3	1	2	2	3	1	1	725.804	2.336
18	2	3	1	2	3	1	2	2	727.058	6.244
19	3	1	3	2	1	3	2	1	713.399	26.248
20	3	1	3	2	2	1	3	2	729.568	33.969
21	3	1	3	2	3	2	1	3	719.609	7.559
22	3	2	1	3	1	3	2	2	132.294	14.435
23	3	2	1	3	2	1	3	3	217.670	56.624
24	3	2	1	3	3	2	1	1	191.903	9.582
25	3	3	2	1	1	3	2	3	224.233	24.422
26	3	3	2	1	2	1	3	1	260.496	35.836
27	3	3	2	1	3	2	1	2	698.471	11.244

Layers = 2, Training Epochs = 150, Regression Level = No Regression and Noise Level = 3x.

Taguchi's design of experiment (DOE) has provided a good starting point with suitable parameters for the proposed prediction model. But ANN's are hard to train due to their stochastic nature and behaviour, as they produce different results at different runs when trained on the same dataset. This randomness in results makes the neural network non-

## 2. POLY-LINEAR REGRESSION WITH AUGMENTED LONG SHORT TERM MEMORY NEURAL NETWORK: PREDICTING TIME SERIES DATA

---

Table 2.5: Response table for means.

Level	Neuron size	Dropout	Optimizer	Regularizer	No of layers	Epochs	Regression Level	Noise Level
1	411.1	330.4	292.7	342.8	311.5	437.6	477.1	454.8
2	423.2	365.5	315.5	566.8	427.3	366.4	417.2	422.1
3	432.0	570.4	658.0	356.7	527.5	462.2	372.0	389.4
Delta	20.9	240.0	365.3	224.1	215.9	95.8	105.1	65.4
Rank	8	2	1	3	4	6	5	7

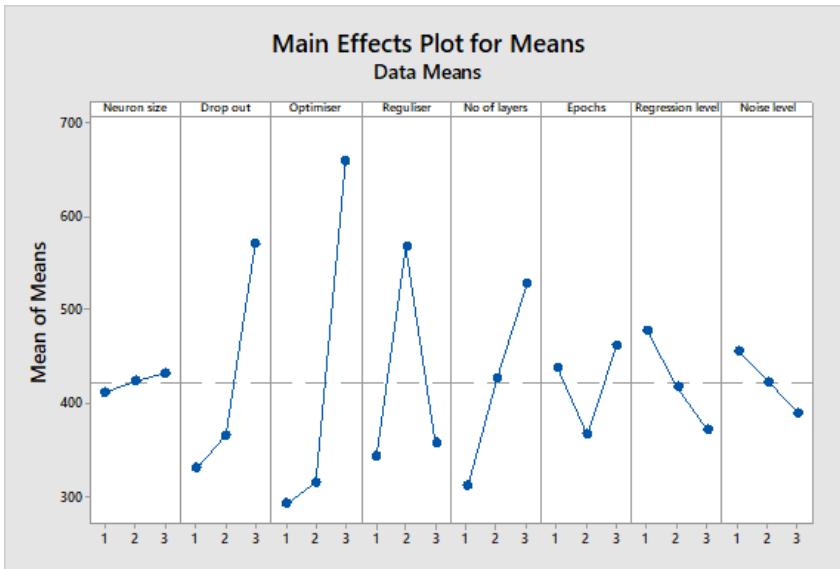


Figure 2.4: Factor level trend for each parameter.

deterministic, unstable and unreliable [Fuller et al., 2005]. In order to fix this randomness, the system is trained using some fixed seed values for 10 runs during Taguchi's DOE. But the obtained settings are not necessarily the optimum ones. So the second level of analysis is done by going through each individual parameter to find a global minimum RMSE. This time instead of training the network for 10 runs, the network is trained for 5 runs using 5 different previously mentioned seed values. After finding the minimum average RMSE for each parameter, the next parameter is chosen. In this way, all parameters and their variants are analysed. The average RMSE of these analyses is illustrated in Table 2.6.

The lowest average RMSE for each setting is stated in Table 2.6. Once the lowest RMSE was reached, that particular parameter was kept in the system, and the next parameter was explored. In this way, each of the parameters was more finely tuned to get the

## 2. POLY-LINEAR REGRESSION WITH AUGMENTED LONG SHORT TERM MEMORY NEURAL NETWORK: PREDICTING TIME SERIES DATA

---

Table 2.6: Average RMSE for each setting using Dataset 1.(a)

Noise level	1x	<i>2x</i>	3x
Avg RMSE	45.0743165	<i>34.52900436</i>	46.65147801
Regression level	<i>Polylinear</i>	Linear	No Regression
Avg RMSE	<i>32.9567096</i>	33.02053399	34.49751502
Epochs Size	100	<i>150</i>	200
Avg RMSE	39.83451781	<i>33.04219715</i>	35.76999331
No. of layers	2	4	6
Avg RMSE	<i>32.93406342</i>	120.4912076	173.6256448
Regularizer	<i>No Regularizer</i>	L1	L2
Avg RMSE	<i>32.99207905</i>	257.0783062	174.8534255
Optimizer	<i>Adam</i>	RMSprop	SGD
Avg RMSE	<i>32.83686546</i>	40.47984571	36.97774995
Dropout	0	0.5	0.8
Avg RMSE	<i>35.16839087</i>	241.1492139	414.3749165
Neuron size	16	32	64
Avg RMSE	32.7202389	21.83345991	<i>17.6611024</i>

most accurate result, even better than Taguchi's DOE. The setting that produced the lowest RMSE is presented as Italicized text in the table. There is not much difference from Taguchi's method apart from a few parameters, such as noise size, neuron size and regression level. Finally, the combination of parameters that produced the best result was: Neuron Size = 64, Dropout = 0.0, Optimizer = Adam, Regularizer = No Regularizer, Number of LSTM Layers = 2, Training Epochs = 150, Regression Level = Polylinear Regression and Noise Level = 2x.

### 2.4.2 Model validation on Dataset 1.(a)

Since the best value for each parameter was obtained, to evaluate the effect of various components, 4 different approaches are explored, with each one being updated by the addition of new features or techniques to the previous one. Starting off with SLSTM, the addition of two different Regression analyses creates two more different models, SLSTM-LR and SLSTM-PLR. And finally, data augmentations are applied to SLSTM-PLR to make the final proposed model PLR-ALSTM-NN.

Using the best possible setting obtained from the parameter tuning, these 4 models are trained 10 different times (using 10 different seed values: Seed 1000 to 1009), and their

## *2. POLY-LINEAR REGRESSION WITH AUGMENTED LONG SHORT TERM MEMORY NEURAL NETWORK: PREDICTING TIME SERIES DATA*

---

RMSEs are calculated. The average RMSE for 10 runs of each of the approaches are reported in Table 2.7.

Table 2.7: Average RMSE for Different Independent Approaches implemented on Dataset 1.(a)

	PLR	SLSTM	SLSTM-LR	SLSTM-PLR	PLR-ALSTM-NN
Average RMSE	23.29369605	21.51468182	24.22799027	24.25546807	<b>18.11095326</b>
Average MAPE	0.751344022	0.741538618	0.862134815	0.86187162	<b>0.614009775</b>
Average MAE	17.73854637	17.19552157	20.0568167	20.05362832	<b>14.02838407</b>

It can be seen from Table 2.7 that PLR-ALSTM-NN produces the lowest average RMSE of 18.11, which is the final proposed model in this chapter. A statistical significance test was carried out among all these different approaches, which will be discussed in the latter part of the Result Analysis section.

The performance of PLR-ALSTM-NN is also evaluated against some standard approaches such as ARIMA, GRU, RNN, CNN, Basic LSTM and RF while making both long and short-term predictions of the closing price of the stock market. A standard statistical ARIMA model has been implemented in Python having three parameters p (Number Auto-Regressive), d (Number of Differences) and q (Number of MA), with the following values: p=1, d=0 and q=0. It is worth mentioning that this ARIMA model is a univariate time series forecasting system that considers only the closing price column as the target variable for making predictions. More information regarding this ARIMA model can be found at the stated website: <https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python>.

The researchers, in many literatures, suggested autocorrelation function, partial autocorrelation function and augmented dicky-fuller test to be used to determine the parameters of ARIMA. In some of those papers, the results presented did not provide clear evidence about the parameter settings that led to those results. In this chapter, the parameters obtained from such a test did not return significant results. Hence, after some trial and error run and using brute force techniques, the best ARIMA model is eventually for making a comparison against PLR-ALSTM-NN.

## 2. POLY-LINEAR REGRESSION WITH AUGMENTED LONG SHORT TERM MEMORY NEURAL NETWORK: PREDICTING TIME SERIES DATA

---

Since GRU, RNN, CNN and Basic LSTM are all available in the Keras library framework, all three of these NNs are implemented in the same way as SLSTM. Experimental analysis was conducted independently using 2 layers of GRU and 2 layers of RNN instead of 2 layers of LSTM with other parameters of the NN remaining the same. A detailed implementation procedure of RNN and GRU can be found at the following url ([https://keras.io/guides/working\\_with\\_rnns/](https://keras.io/guides/working_with_rnns/)). While implementing CNN, in order to keep consistent with the proposed model 2 layers of 1 dimensional CNN with a Pooling layer in between and a flattened layer before the dense layer is used here in this chapter. For further comparison, Basic LSTM is applied with only one layer of LSTM with other parameters remaining the same as SLSTM and RF is also implemented on the same set of data for making predictions. RF, being an algorithm from a tree-based group that is not related to NN, has a different set of parameters and whose selection was based on the study conducted by Nabipour et al. [2020] on the stock market. According to the author, for short-term, 5-day and 7-day ahead analyses, tree sizes of 300, 500 and 300 are chosen, while for long-term analysis, it is considered as 50. The max depth of the tree is kept at 10 for all analyses. No data augmentation or regression was used during the analysis. For each network, the model is run 10 times using 10 different seed values, and the average is calculated. For all the different approaches, the predicted results for both long and short-term forecasts are compared against the true values in terms of average RMSE, MAPE and MAE calculated using equations 2.11, 2.12 and 2.13, respectively and represented in Table 2.8 and Table 2.9.

Table 2.8: Evaluation of PLR-ALSTM-NN with respect to ARIMA, GRU, RNN and CNN for short term predictions using Dataset 1.(a)

	PLR-ALSTM-NN	ARIMA (univariate)	GRU	SimpleRNN	CNN	Basic LSTM	RF
RMSE	18.111	<b>12.399</b>	22.523	22.752	21.198	17.404	252.104
MAPE	0.614	<b>0.375</b>	0.848	0.805	0.752	0.606	8.316
MAE	14.028	<b>8.499</b>	19.552	18.792	17.517	13.984	201.572

Table 2.9: Evaluation of PLR-ALSTM-NN with respect to ARIMA, GRU, RNN and CNN for long term predictions using Dataset 1.(a)

	PLR-ALSTM-NN	ARIMA (univariate)	GRU	SimpleRNN	CNN	Basic LSTM	RF
RMSE	<b>44.342</b>	59.347	47.607	71.595	45.127	45.226	298.802
MAPE	<b>1.579</b>	2.061	1.710	2.569	1.614	1.609	10.666
MAE	<b>35.856</b>	47.918	39.075	60.350	36.663	36.656	255.868

## *2. POLY-LINEAR REGRESSION WITH AUGMENTED LONG SHORT TERM MEMORY NEURAL NETWORK: PREDICTING TIME SERIES DATA*

---

For short-term predictions, ARIMA has a better forecasting accuracy as it has the lowest average RMSE (12.399), MAPE (0.375) and MAE (8.499), among all other different approaches stated in Table 2.8. It only considers the closing price of each day to predict the next day's closing price. Thus it is a univariate analysis statistical tool. The proposed approach PLR-ALSTM-NN is in 3rd position with respect to other approaches such as GRU, RNN, CNN and RF. The average RMSE, MAPE and MAE of PLR-ALSTM-NN is 18.11, 0.614 and 14.028, respectively, and these evaluation parameters are less than that of GRU, RNN, CNN and RF.

From Table 2.9 it can be observed that ARIMA does not work as well as it does for making short-term predictions. While making a 30days ahead prediction into the future, it produces a higher RMSE of 59.067, MAPE of 2.135 and MAE of 48.691 when compared to other approaches. On the other hand, the proposed approach PLR-ALSTM-NN has the best performance (RMSE: 44.342, MAPE: 1.579 and MAE: 35.856) for making long-term predictions producing slightly better results than CNN, GRU, Basic LSTM and significantly better results if compared against ARIMA, RNN and RF. Its performance will be further evaluated by comparing its performance accuracy against recent literature, which will be discussed next.

The result is also compared against a recently published literature [Althelaya et al., 2018] based on Dataset 1.(a), where the average value for several runs was calculated. The improvement of prediction performance for different numbers of neurons is shown in Table 2.10 by calculating RMSE on de-normalized tests and predicted data. Although the best result can be obtained for 64 neurons, as shown in the parameter tuning section, in this case, analysis is done for 4, 8, 16 and 32 neurons for a fair comparison.

Table 2.10: Prediction Accuracy of PLR-ALSTM-NN in comparison to the

existing literature for Short Term Prediction (Dataset 1.(a)).

Networks	From Literature [Althelaya et al., 2018]		PLR-ALSTM-NN		
	BiLSTM	SLSTM	Best	Avg	SD
4 N	70.92	84.014	19.550	59.811	25.027
8 N	43.372	67.779	15.859	40.638	20.477
16 N	36.451	53.113	13.142	30.781	9.915
32 N	34.893	41.495	13.326	21.131	5.821
Average	46.409	61.393		<b>38.090</b>	

From Table 2.10, it is seen that the proposed model's performance is significantly better than the current literature. The mean RMSE of 30 runs is 38.090, which is much lower than the lowest mean RMSE in the literature [Althelaya et al., 2018], which is 46.409, with each individual network's average RMSE also significantly lower than the literature.

Similarly, the model is also tested to make long-term predictions (30 days ahead) using the same testing set. While structuring the dataset using the sliding window technique, the output is 30 days ahead of the current day. The prediction accuracy of the proposed model (PLR-ALSTM-NN) for long-term analysis is shown in Table 2.11 with a comparison against the literature. For long-term prediction, the mean RMSE (87.132) is also lower than the lowest value (119.835) in the literature, demonstrating the model's robustness. Based on the information illustrated in Table 2.11, it can be noted that the model's performance under the best possible setting outperforms the results shown in the literature. Even though the model makes more accurate predictions in both these cases but still in order to show its robustness, it is also tested by exposing it to other datasets. Its performance is further evaluated and discussed in the next section.

Table 2.11: Prediction Accuracy of the Model in comparison to existing literature for Long Term Prediction (Dataset 1.(a)).

Networks	From Literature [Althelaya et al., 2018]		PLR-ALSTM-NN		
	BiLSTM	SLSTM	Best	Avg	SD
4 N	143.031	162.463	53.869	110.707	29.772
8 N	127.399	146.78	46.562	93.331	28.769
16 N	108.393	140.365	46.355	78.313	21.265
32 N	100.517	128.411	45.129	66.176	15.905
Average	119.835	144.505		<b>87.132</b>	

#### 2.4.3 Model validation on recent S&P500 (Dataset 1.(b) and Dataset 1.(c))

Using a more extended S&P500 dataset (Dataset 1.b), the performance of PLR-ALSTM-NN is evaluated and its prediction is compared against traditional approaches like ARIMA, GRU, RNN and CNN. As usual, the dataset is split in the ratio of 80:20, and the prediction accuracy for each of the approaches is determined using RMSE, MAPE and MAE. For

## 2. POLY-LINEAR REGRESSION WITH AUGMENTED LONG SHORT TERM MEMORY NEURAL NETWORK: PREDICTING TIME SERIES DATA

---

both short and long-term approaches, each of the models is trained 10 differently using the same exact setting as discussed in the parameter tuning section, and average RMSE, MAPE and MAE are calculated and reported in Tables 2.12 and 2.13. Similarly, a time series representation of the predicted values of different models alongside true values is depicted in Figures 2.5 and 2.6. Based on Tables 2.12 and 2.13, it can be seen that ARIMA again has the best prediction accuracy for short term forecasting (RMSE: 25.755, MAPE: 0.657, MAE: 18.235), but at the same time, it struggled while making long term analysis producing the highest errors among each of the performance indicators. The proposed PLR-ALSTM-NN, on the other hand, did not show promising performance for short term prediction. However, it produces acceptable results while predicting 30 days ahead of data. It acquired 2<sup>nd</sup> position, next to RNN (RMSE: 115.747, MAPE: 3.409 and MAE: 95.079), with an average RMSE of 116.407, MAPE of 3.438 and MAE of 95.904. The errors produced by each accuracy measuring function do not differ by much between the two approaches as they do with ARIMA, GRU, CNN, Basic LSTM and RF for long-term prediction. PLR-ASLTN-NN showed a competitive performance when compared to SimpleRNN but it is not the best when dealing with Dataset 1.(b). It may be because this particular dataset has fewer long-term dependencies in the testing set than it has in its training portion. Hence, further experimentation is conducted on other datasets, and statistical tests are carried out to check for the statistical significance of the proposed PLR-ALSTM-NN approach, which will be discussed in the latter part of this chapter.

Table 2.12: Average RMSE, MAPE and MAE for short term Close Price Prediction of S&P500 (Dataset 1.(b))

	PLR-ALSTM-NN	ARIMA	GRU	SimpleRNN	CNN	Basic LSTM	RF
RMSE	34.375	<b>25.755</b>	32.871	34.478	34.487	29.306	210.265
MAPE	0.905	<b>0.657</b>	0.895	0.982	0.990	0.779	5.590
MAE	25.361	<b>18.235</b>	24.955	27.436	27.952	21.616	164.280

Table 2.13: Average RMSE, MAPE and MAE for long-term Close Price Prediction of S&P500 (Dataset 1.(b))

	PLR-ALSTM-NN	ARIMA	GRU	SimpleRNN	CNN	Basic LSTM	RF
RMSE	116.407	121.717	119.069	<b>115.747</b>	120.208	117.880	215.544
MAPE	3.438	3.621	3.470	<b>3.409</b>	3.601	3.465	5.974
MAE	95.904	101.748	96.387	<b>95.079</b>	101.248	96.418	174.251

The performance of the model is observed using another version of the same S&P500 stock

## 2. POLY-LINEAR REGRESSION WITH AUGMENTED LONG SHORT TERM MEMORY NEURAL NETWORK: PREDICTING TIME SERIES DATA

---

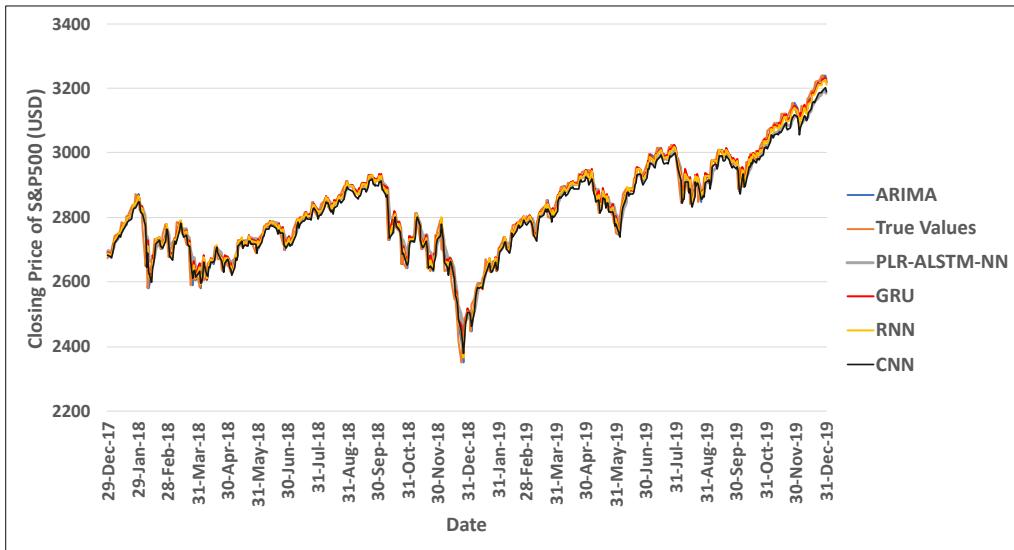


Figure 2.5: Average Closing Price for Short Term Prediction for Each Network on Extended S&P500 dataset (Dataset 1.(b))

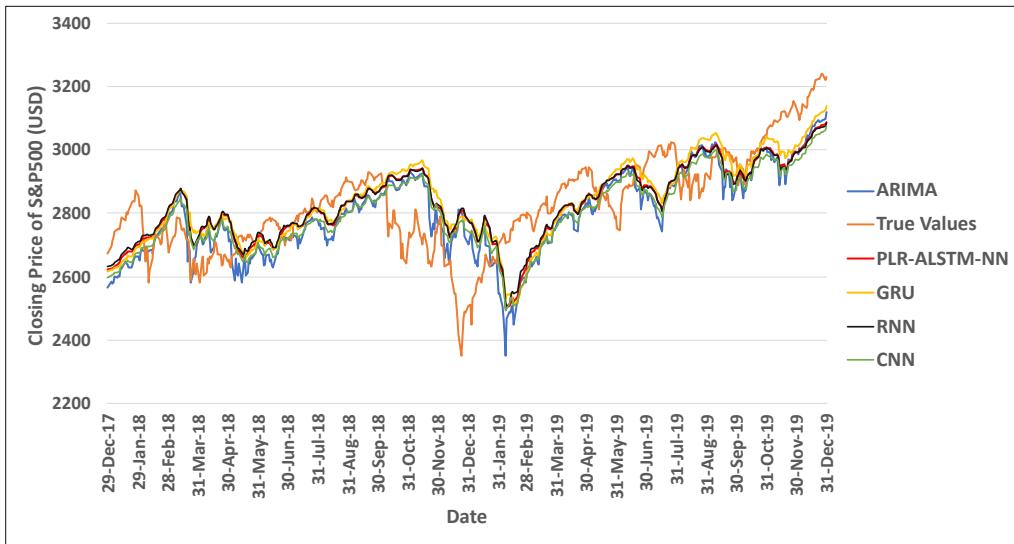


Figure 2.6: Average Closing Price for Long Term Prediction for Each Network on Extended S&P500 dataset (Dataset 1.(b))

data that has the effect of Covid 19. This is done by keeping data from 2 Jan 2020 till 27 Jan 2021 in the test phase while the past data is kept to train the network. It is termed Dataset 1.(c). With Dataset 1.(c), both long and short-term forecasting is done using the proposed methodology, and other traditional approaches and the results are presented in Tables 2.14 and 2.15. Based on the information in Table 2.14, Basic LSTM

## 2. POLY-LINEAR REGRESSION WITH AUGMENTED LONG SHORT TERM MEMORY NEURAL NETWORK: PREDICTING TIME SERIES DATA

---

produced the lowest RMSE, MAPE, MAE of 58.708, 1.309 and 39.886, respectively, when compared to all other approaches. Surprisingly, ARIMA, with this particular dataset, is not the best-performing approach for short-term forecasting, producing a slightly higher RMSE (59.788), MAPE (1.321) and MAE (39.854) in contrast to Basic LSTM. The proposed PLR-ALSTM-NN struggled a bit for short-time horizon prediction. However, its performance improves significantly for longer horizon forecasting (from Table 2.15). For long-term analysis, PLR-ALSTM-NN has an average RMSE of 298.839, MAPE of 7.253 and MAE of 216.352 which is marginally less than that of RNN (RMSE: 296.340, MAPE: 7.116 and MAE: 213.273).

Table 2.14: Average RMSE, MAPE and MAE for short-term Close Price Prediction of S&P500 (Dataset 1.(c))

	PLR-ALSTM-NN	ARIMA	GRU	SimpleRNN	CNN	Basic LSTM	RF
RMSE	67.586	59.788	60.806	58.908	64.968	<b>58.708</b>	242.126
MAPE	1.484	1.321	1.352	1.345	1.571	<b>1.309</b>	4.955
MAE	45.639	<b>39.854</b>	40.953	41.248	49.190	39.886	171.327

Table 2.15: Average RMSE, MAPE and MAE for long-term Close Price Prediction of S&P500 (Dataset 1.(c))

	PLR-ALSTM-NN	ARIMA	GRU	SimpleRNN	CNN	Basic LSTM	RF
RMSE	298.839	304.383	300.217	<b>296.340</b>	308.452	300.135	334.072
MAPE	7.253	7.708	7.477	<b>7.116</b>	8.072	7.396	8.386
MAE	216.352	231.647	223.701	<b>213.273</b>	245.963	216.657	264.532

Except for ARIMA, all other networks were trained 10 times using 10 different seed values, and the average predicted values were calculated for each time step. RMSE, MAPE and MAE were all calculated from the average predicted values against the test set of each dataset. Whilst the performance of PLR-ALSTM-NN needs improvement for short-term prediction in comparison to other methodologies, however, its accuracy is significantly better for long-term forecasting, where RNN and PLR-ALSTM-NN have similar lowest errors among other approaches. This may be due to the ability of the model to make multivariate analyses where the output is dependent on multiple variables of past data. To prove its effectiveness for long-term analysis, it is further tested with the other dataset in the next sections. The results reported in this section of the chapter are for researchers to compare in the future.

#### **2.4.4 Model Validation on Dataset 2**

In order to validate the model, it is further tested on two datasets, Datasets: 2 and 5. The dataset source was already mentioned in the data collection section and is similar to the primary dataset. Using Dataset 2, closing prices are predicted for 1, 3, 5 and 7 days ahead and are found respectively using the best setting obtained from the proposed model. The prediction performance of the model for different days ahead into the future is compared against the literature where the authors [Majhi et al., 2008] used Particle Swarm Optimization (PSO) to make predictions and compared their results to Multilayer Perceptron (MLP). After comparing against the literature, the results show improved performance for our model. Table 2.16 compares the closing price prediction for different days using MAPE as a performance indicator. Using dataset 5, the model is again compared against recent literature [Zhang et al., 2020] to evaluate its long-term prediction accuracy. The model is trained with 64 neurons for both cases since the best setting obtained from parameter tuning contains 64 Neurons. However, for Dataset 2, only the best parameter combination is chosen, and the best-evaluated result is compared against the literature (since the authors presented the best MAPE in their article), and for Dataset 5, average RMSE, MAPE and MAE are selected for comparison.

Table 2.16: Validation of the performance of the proposed model using a secondary dataset (Dataset 2) by comparing MAPE

Days ahead	MAPE					
	From Literature [Majhi et al., 2008]		PLR-ALSTM-NN			
	PSO	MLP	Best	Avg	SD	
1	0.6661	1.0049	<b>0.5204</b>	0.6810	0.2438	
3	1.0801	2.0922	<b>0.8697</b>	0.9878	0.1690	
5	1.2451	2.5097	<b>1.1202</b>	1.1654	0.0623	
7	1.4664	3.117	<b>1.3215</b>	1.3728	0.0949	

From Table 2.16, it can be seen that the best result (in bold) produced by the proposed model is significantly better than the best result from the literature. For 1 day ahead prediction, the best MAPE produced by the model is 0.5204, which is considerably lower than the one produced by PSO, which is 0.6661. In this way, a comparative analysis shows that the best value is consistently better than the literature for all days ahead of

## 2. POLY-LINEAR REGRESSION WITH AUGMENTED LONG SHORT TERM MEMORY NEURAL NETWORK: PREDICTING TIME SERIES DATA

---

forecasting. Moreover, it can be seen from the table that the average of 30 runs also produces a better result than the current literature. Additionally, the standard deviation is very low shows the consistency of each run. For better visualisation of the model's prediction accuracy, the predicted closing prices are plotted along with the true values in Figures 2.7, 2.8, 2.9 and 2.10 for different days ahead into the future.

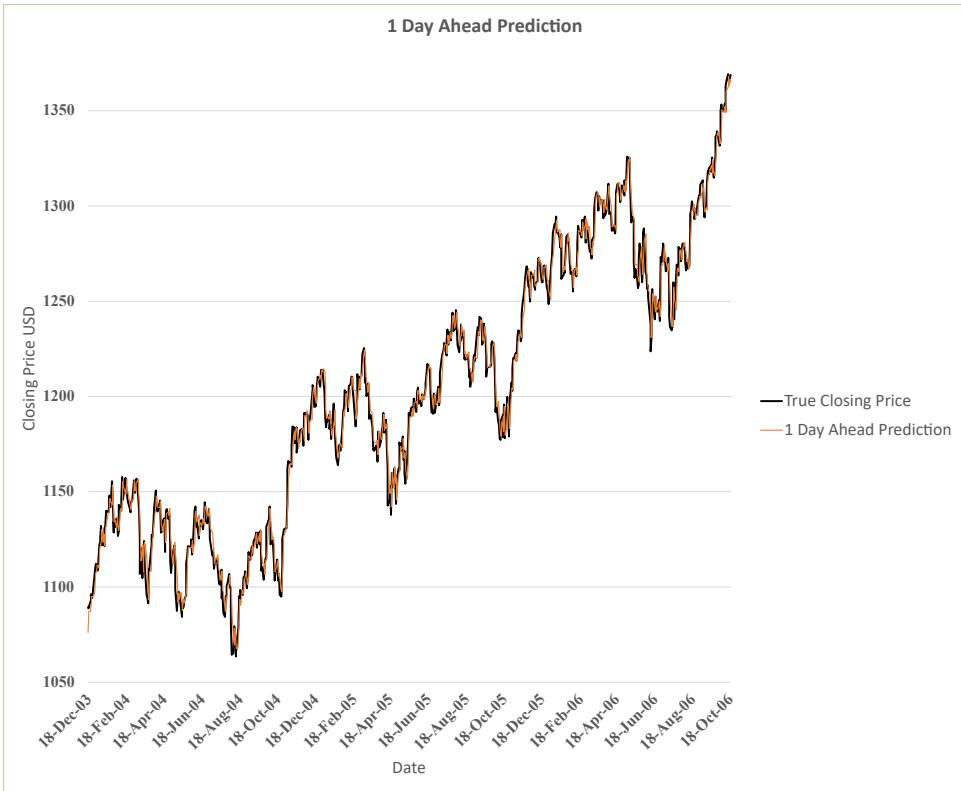


Figure 2.7: Short Term Prediction by using the proposed approach using Dataset 2

Figures 2.7, 2.8, 2.9 and 2.10 show the predicted closing prices of 1, 3, 5 and 7 days ahead with respect to the true price obtained from the best setting of the model. The obtained plots are selected from the best result among the 30 runs with 64 neurons. It can be seen from the above figures that the predicted results are very close to the actual values, with the model being able to precisely detect small subtle changes. The evaluation of the performance of the model shows that the predicted results produce substantially lower MAPE than the lowest MAPE produced by PSO and MLP. The average MAPE of different runs is also better than the literate. Hence, it can be concluded that the model developed

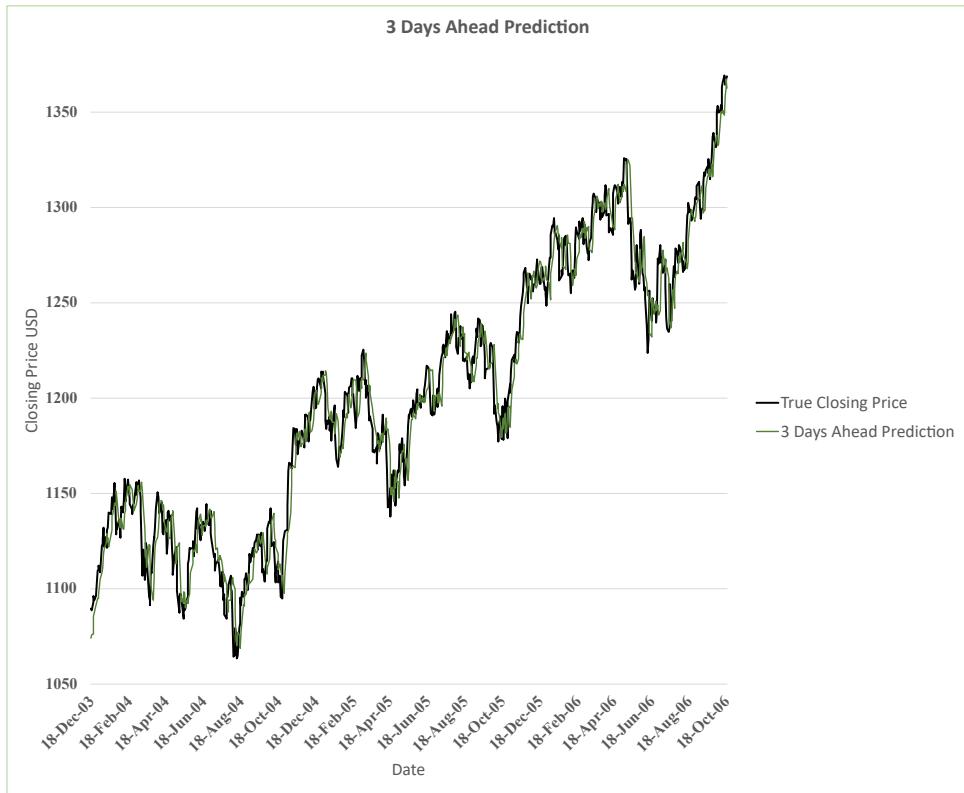


Figure 2.8: 3 Days Ahead Prediction by using the proposed approach using Dataset 2

and tuned with the previous optimum parameter settings also can produce better results for this 2nd dataset (Dataset 2).

In the same way, this particular closing price is predicted for different days ahead using other traditional approaches such as ARIMA, GRU, RNN, CNN, Basic LSTM and RF and the results are compared against PLR-ALSTM-NN. Each network is trained 10 times, and the average results are presented in Table 2.17. To keep the prediction results consistent with other datasets presented in this chapter, long-term predictions i.e. 30 days ahead of forecasting are also done to keep fairness in performance comparison.

It can be observed from Table 2.17, that ARIMA, like other previous attempts, is good at making short-term forecasts, whereas its performance deteriorates while making predictions more days into the future, especially when making 30 days ahead of predictions of financial stock market prices. For long-term predictions (30 days ahead), ARIMA pro-

## 2. POLY-LINEAR REGRESSION WITH AUGMENTED LONG SHORT TERM MEMORY NEURAL NETWORK: PREDICTING TIME SERIES DATA

---

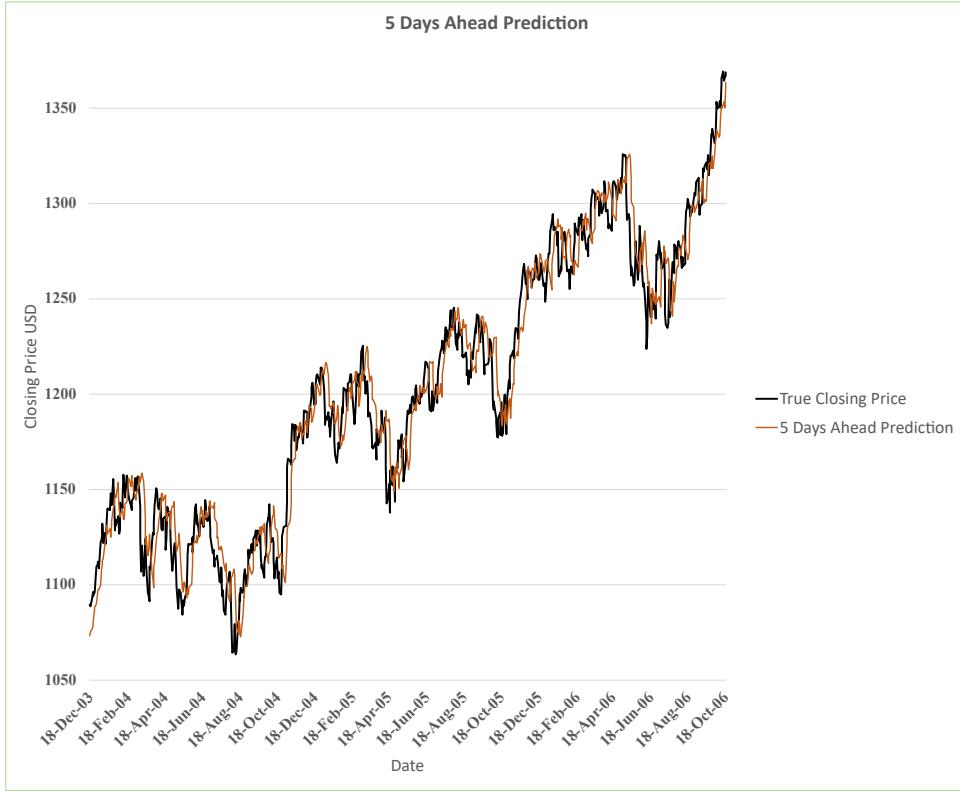


Figure 2.9: 5 Days Ahead Prediction by using the proposed approach using Dataset 2

Table 2.17: Average RMSE, MAPE, and MAE for different days ahead of forecasting using PLR-ALSTM-NN, ARIMA, GRU, RNN and CNN on Dataset 2

	1 day ahead					5 day ahead					30 day ahead				
	PLR-ALSTM-NN	ARIMA	GRU	RNN	CNN	Basic LSTM	RF	PLR-ALSTM-NN	ARIMA	GRU	RNN	CNN	Basic LSTM	RF	
RMSE	10.984	<b>7.985</b>	14.067	16.644	14.534	11.175	10.101	17.700	<b>16.707</b>	20.303	21.056	19.461	18.602	23.213	
MAPE	0.768	<b>0.521</b>	1.030	1.250	1.062	0.770	0.674	1.202	<b>1.126</b>	1.396	1.447	1.339	1.261	1.564	
MAE	9.186	<b>6.226</b>	12.337	14.981	12.704	9.206	8.093	14.337	<b>13.458</b>	16.685	17.271	15.988	15.053	18.900	
	7 day ahead					30 day ahead									
	PLR-ALSTM-NN	ARIMA	GRU	RNN	CNN	Basic LSTM	RF	PLR-ALSTM-NN	ARIMA	GRU	RNN	CNN	Basic LSTM	RF	
RMSE	19.924	<b>19.644</b>	22.439	23.132	23.241	20.198	25.092	35.172	37.668	37.135	36.765	35.120	<b>34.603</b>	59.618	
MAPE	1.379	<b>1.351</b>	1.541	1.593	1.563	1.375	1.690	2.405	2.560	2.547	2.503	2.407	<b>2.365</b>	3.935	
MAE	16.449	<b>16.126</b>	18.412	19.059	18.522	16.374	20.335	28.630	30.643	30.447	29.761	28.699	<b>28.183</b>	47.641	

duces an RMSE of 35.275, MAPE of 2.411 and MAE of 28.790, which is higher than that of the proposed approach indicating its drawbacks in making long-term predictions. The proposed approach PLR-ALSTM-NN has the 2<sup>nd</sup> lowest MAPE and MAE of 2.405 and 28.630, respectively with the lowest being Basic LSTM (MAPE: 2.365, and MAE: 28.183), and it has the 3<sup>rd</sup> lowest RMSE (35.172) which is marginally higher than CNN (35.120). In comparison to other approaches, PLR-ALSTM-NN has remarkable performance for long-horizon prediction. It can also be seen from Table 2.17 that for the short-term or any intermediate days ahead of predictions, PLR-ALSTM-NN performs better than other

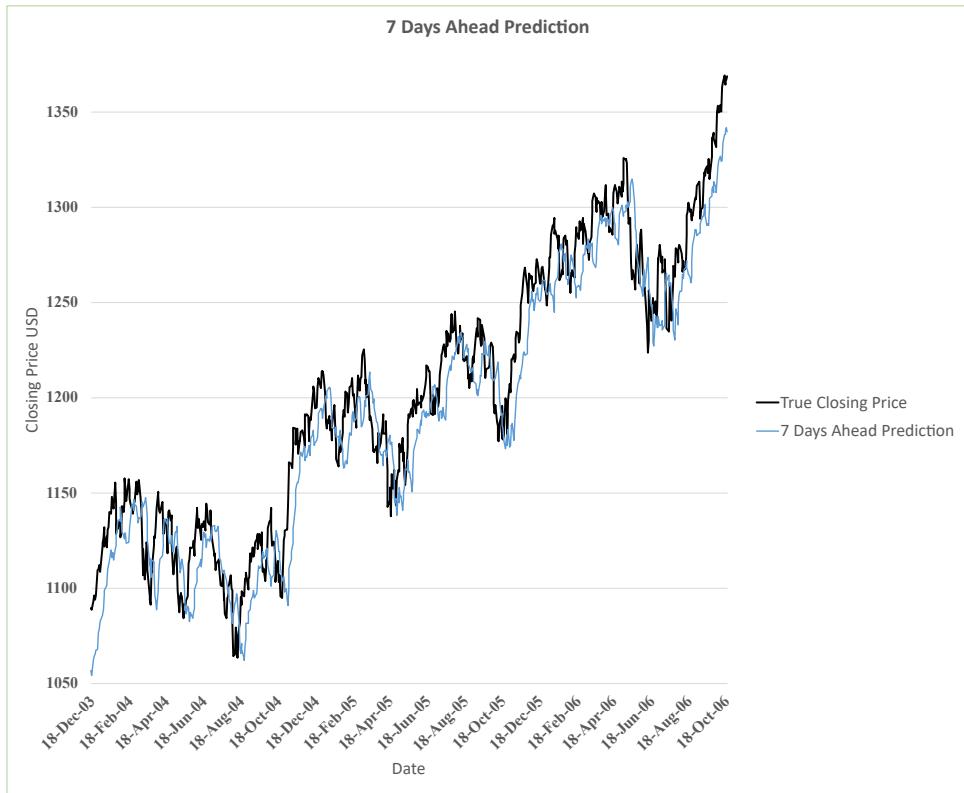


Figure 2.10: 7 Days Ahead Prediction by using the proposed approach using Dataset 2

NNs except ARIMA, which is a kind of shortcoming that can be addressed in the future.

#### **2.4.5 Model Validation on Dataset 5**

The approach is further validated by comparing its long-term prediction performance against another work [Zhang et al., 2020]. The authors have made long-term predictions of closing prices of SZSE data (Dataset 5) using state-of-the-art two-stage approaches. Among those approaches, ENANFIS (Ensemble Adaptive Neuro-Fuzzy Inference System) and SVR-SVR were chosen for comparison. Using the same principle of train-test split and the same setting, PLR-ALSTM-NN is used to make a long-term forecast (30 days ahead) of the closing price of the aforementioned stock market and a comparison is presented in Table 2.18.

From Table 2.18, PLR-ALSTM-NN has the best performance in terms of RMSE (lowest

## *2. POLY-LINEAR REGRESSION WITH AUGMENTED LONG SHORT TERM MEMORY NEURAL NETWORK: PREDICTING TIME SERIES DATA*

---

Table 2.18: Comparison of performance of PLR-ALSTM-NN for long-term forecasting against ENANFIS and SVR-SVR (Dataset 5)

		From Literature [Zhang et al., 2020]	
	PLR-ALSTM-NN	ENANFIS	SVR-SVR
RMSE	<b>1.313</b>	2.359	1.441
MAPE	7.248	6.686	<b>3.363</b>
MAE	<b>1.010</b>	1.686	0.824

RMSE), and it has a much higher MAE (1.010), which takes it to 2<sup>nd</sup> position after SVR-SVR (MAE: 0.824). Hence the accuracy of the proposed approach is very competitive when compared to modern approaches like ENANFIS and SVR-SVR.

## **2.5 Application of the Model to Predict Sales Data and Closing Price of DJI Data**

PLR-ALSTM-NN is first implemented in this section on a dataset from a field apart from financial markets. The model is tested with time series sales data, and its prediction success shows that it can also be used for other general purposes. Besides demonstrating its generic behaviour, it is again tested on recent DJI data to observe how well the model behaves on extremely uncertain Covid'19 affected stock market data. The results of the analysis on these two datasets are presented in this section.

### **2.5.1 Predicting Time Series Sales Data**

So far, all the analyses have been done using financial time series data. To prove the generality of this proposed model, a 3<sup>rd</sup> dataset (Dataset 3: a time series sales data) has been considered at this stage. For that, this 3<sup>rd</sup> dataset is related to the supply chain domain, which is fed into the model to make sales predictions. As discussed earlier, the entire set is split in the ratio of 80:20 where 20% contains the test data. Using this ratio, the model is trained and used to predict the sales of product P34. The reason for choosing this particular product is the fact that it had the highest number of sales over the

## *2. POLY-LINEAR REGRESSION WITH AUGMENTED LONG SHORT TERM MEMORY NEURAL NETWORK: PREDICTING TIME SERIES DATA*

---

entire year. The dataset is fed in the same way into the model but this time using linear regression instead of polylinear regression because creating higher degree (in the order of 4) coefficients of 811 variants makes the system run out of memory. So in future, this is one particular parameter that needs to be considered and adjusted accordingly within the model while using data from other sources. Additionally, the result cannot be compared to other literature as other researchers only used the dataset to solve classification problems [Tan and San Lau, 2014, Prakash et al., 2018] rather than time series prediction. This is probably the first time this particular sales data has been used to make time-series predictions.

Table 2.19: The best, average and the standard deviation of RMSEs for different weeks ahead of sales predictions (Dataset 3)

Weeks ahead	RMSE		
	Best	Avg	SD
1	7.1505	7.2713	0.1377
3	7.1241	7.3210	0.0576
5	7.3186	7.3346	0.0074
7	7.0906	7.3272	0.0452

RMSEs for different weeks ahead (1 week, 3 weeks, 5 weeks and 7 weeks) of sales prediction are calculated and shown in Table 2.19. Although the proposed model can consider multiple prediction horizons ranging from short to long-term prediction, it is believed that due to the small amount of training data, the prediction loses accuracy when looking more weeks into the future. Hence, using the previous settings, the best-predicted sales and their average for several different runs of Product 'P34' are plotted along with true sales in Figure 2.11. The graph in Figure 2.11 shows a comparison between the true trend vs the best and average predicted trend of the sales of the product 'P34' for the last 11 weeks of the year, i.e. from week 42 till 52 using the setting from the previous sections. From Figure 2.11 it can be seen that the best-predicted sales and the average (of multiple runs) predicted sales exhibit similar patterns over time and the model can also forecast the sales trend over 11 weeks. When dealing with many variables (811 in this case), poly-linear regression cannot be chosen over linear regression analysis, which can be considered a limitation of the proposed model. Finding out the optimum number of variables of the

## *2. POLY-LINEAR REGRESSION WITH AUGMENTED LONG SHORT TERM MEMORY NEURAL NETWORK: PREDICTING TIME SERIES DATA*

---

data with which poly-linear regression can be successfully used is a possible future direction. Moreover, in future, the model needs to be tested with other types of time series data like from the data supply chain domain. Seasonality factors can also be incorporated for making efficient future predictions. Moreover, feature extraction techniques like data cleaning, Fourier Transformations or Wavelet Transformations can be also be explored in the data augmentation section.

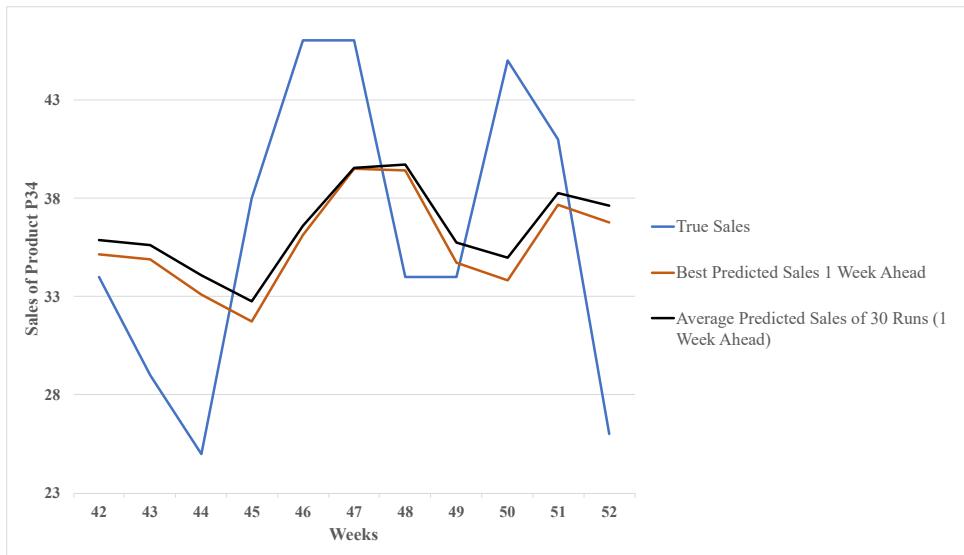


Figure 2.11: The best and average Predicted Sales (Dataset 3) for 1 Week Ahead

### **2.5.2 Predicting the Closing Price of Dow Jones Industrial Average (DJI)**

Since the sales data may not have enough features to optimize the network parameters, the validity of the proposed model PLR-ALSTM-NN is again justified on the latest Dow Jones Industrial Average public dataset (Dataset: 4.(a)) collected from the yahoo Finance website. The dataset has 2506 time steps of stock prices of 30 different companies consisting of the following features Open, High, Low and Close prices. As before, the dataset has divided the ratio of 80:20, where 80% is the training set, and the remaining is used for testing the network. That means for training purposes, 2004 days of data are used, whereas the remaining 502 days of data are reserved for testing the network. Both short

## 2. POLY-LINEAR REGRESSION WITH AUGMENTED LONG SHORT TERM MEMORY NEURAL NETWORK: PREDICTING TIME SERIES DATA

---

and long-term predictions are calculated and the results are compared against the standard approaches like ARIMA, GRU, RNN and CNN in terms of RMSE, MAPE and MAE. The parameters of the PLR-ALSTM-NN remain the same as before with poly-linear regression (PLR) in the initial phase, 2 hidden layers each having 64 neurons alongside other previously mentioned settings. The other approaches also had the same setting as mentioned before.

The average predicted closing price of DJI stock indices for both short and long term evaluation for different networks are depicted in Figures 2.12 and 2.13.



Figure 2.12: Short Predictions of the closing prices of DJI Stock index (Dataset 4.(a)) using different available networks with respect to the true values

A comparison of the results for short and long-term predictions of the closing price of the DJI stock market using PLR-ALSTM-NN, ARIMA, GRU, RNN, CNN, Basic LSTM and RF are illustrated in Table 2.20 and Table 2.21, respectively.

Table 2.20: Average RMSE, MAPE and MAE for short-term Close Price forecasting of DJI Data (Dataset 4.(a))

	PLR-ALSTM-NN	ARIMA	GRU	SimpleRNN	CNN	Basic LSTM	RF
RMSE	326.850	<b>243.045</b>	296.712	311.350	289.853	278.748	1482.484
MAPE	0.949	<b>0.682</b>	0.867	0.977	0.877	0.805	4.321
MAE	241.724	<b>172.356</b>	219.858	249.206	223.489	203.569	1147.462

It can be seen from Table 2.20 that PLR-ALSTM-NN produces higher average errors

## 2. POLY-LINEAR REGRESSION WITH AUGMENTED LONG SHORT TERM MEMORY NEURAL NETWORK: PREDICTING TIME SERIES DATA

---

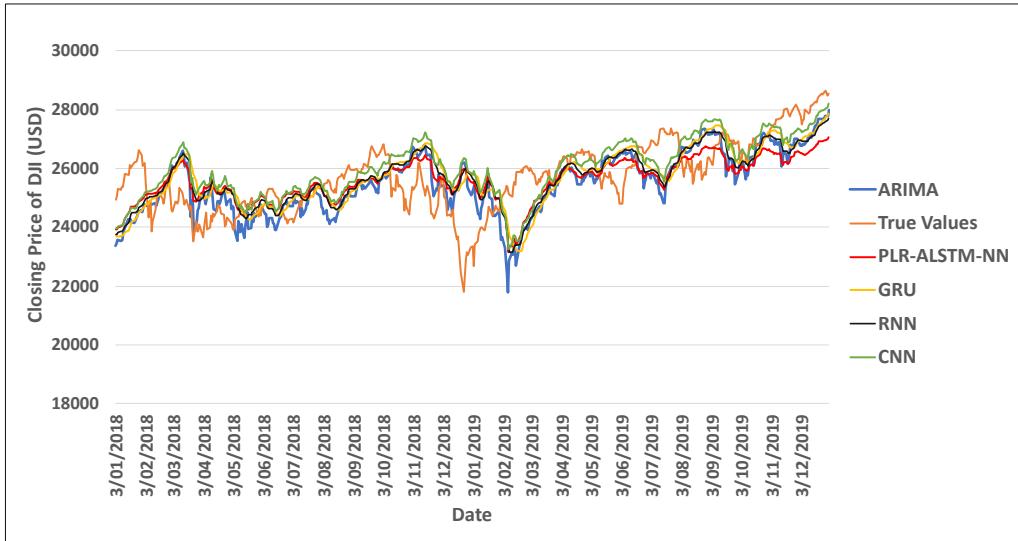


Figure 2.13: Long Term Predictions of the closing prices of DJI Stock index (Dataset 4.(a)) using different available networks with respect to the true values

Table 2.21: Average RMSE, MAPE and MAE for long-term Close Price forecasting of DJI Data (Dataset 4.(a))

	PLR-ALSTM-NN	ARIMA	GRU	SimpleRNN	CNN	Basic LSTM	RF
RMSE	<b>1044.110</b>	1102.149	1079.228	1053.053	1124.457	1082.501	1534.292
MAPE	<b>3.355</b>	3.531	3.431	3.387	3.551	3.444	4.683
MAE	<b>858.700</b>	903.750	872.749	862.631	897.417	874.154	1234.201

(RMSE: 326.850, MAPE: 0.949 and MAE: 241.724), with ARIMA being the best approach (RMSE: 243.045, MAPE: 0.682 and MAE: 172.356) for short term forecasting. But the limitation of ARIMA is that it is a univariate analysis that considers only the historical Closing Prices to make 1 day-ahead prediction.

Although ARIMA is good for making short-term predictions, its performance again degrades while making long-term predictions due to its limitations. While making 30 days ahead of Close price predictions of the more recent DJI dataset (Dataset 4.(a)), ARIMA's performance is not very promising in comparison to other NNs and RF. From Table 2.21 the average RMSE of ARIMA, GRU, RNN, CNN, Basic LSTM and RF is significantly higher at 1102.149, 1079.228, 1053.053, 1124.457, 1080.501 and 1534.292, respectively, which makes them less suitable for making long term predictions, compared to the proposed approach PLR-ALSTM-NN (RMSE: 1044.110). Also, in the other two performance

## 2. POLY-LINEAR REGRESSION WITH AUGMENTED LONG SHORT TERM MEMORY NEURAL NETWORK: PREDICTING TIME SERIES DATA

---

evaluation indicators, PLR-ALSTM-NN produces the lowest MAPE and MAE of 3.355 and 858.700, respectively, compared to other standard approaches. This further establishes the fact that PLR-ALSTM-NN, with its multivariate analysis ability performs better for long-term forecasting than it does for short-term analysis.

The long-term and short-term forecasting reliability of the model is tested by subjecting it to a subset of the same DJI dataset (Dataset 4.(b)) which is prepared by adding the duration of the Covid 19 pandemic in the test set while the rest of the data is used to train the network. This is done to observe how far the model is capable of predicting extreme changes in the data that may occur in cases such as a global pandemic or any other natural disaster. Using this DJI's dataset, all the models are trained, and the results of both long and short-term forecasting are reported in Tables 2.22 and 2.23.

Table 2.22: Average RMSE, MAPE and MAE for short-term Close Price forecasting of DJI Data (Dataset 4.(b))

	PLR-ALSTM-NN	ARIMA	GRU	SimpleRNN	CNN	Basic LSTM	RF
RMSE	582.560	<b>426.735</b>	540.968	520.068	559.602	541.550	897.619
MAPE	1.487	<b>1.158</b>	1.422	1.361	1.588	1.449	2.347
MAE	377.803	<b>292.869</b>	359.589	344.392	410.599	369.208	640.418

Table 2.23: Average RMSE, MAPE and MAE for long-term Close Price forecasting of DJI Data (Dataset 4.(b))

	PLR-ALSTM-NN	ARIMA	GRU	SimpleRNN	CNN	Basic LSTM	RF
RMSE	2612.171	2899.249	2636.393	<b>2496.893</b>	2669.134	2612.178	2547.750
MAPE	7.661	8.344	8.024	<b>7.049</b>	7.815	7.606	7.037
MAE	1918.829	2030.529	2035.612	<b>1768.084</b>	1951.985	1951.985	1749.784

From Table 2.22 it can be deduced that ARIMA, like all other previous analyses, is the best performing algorithm for short-term forecasting, whereas Table 2.23 shows that RNN has the least errors (RMSE: 2496.893, MAPE: 7.049 and MAE: 1768.084) while making long term forecasting. The proposed approach, PLR-ALSTM-NN, is in 3<sup>rd</sup> position in terms of RMSE (2612.171), while RNN and RF acquired the 1<sup>st</sup> and 2<sup>nd</sup>, position respectively. According to Table 2.23 The performance of PLR-ALSTM-NN is consistently producing slightly higher errors in relation to the best two approaches (RNN and RF). It is to be mentioned that, among all other previous experimental analyses, RF produced some comparable results for the 1st time having RMSE of 2547.750, MAPE of 7.037 and MAE of 1749.784. The other systems, such as ARIMA, CNN and GRU produced higher errors than

## *2. POLY-LINEAR REGRESSION WITH AUGMENTED LONG SHORT TERM MEMORY NEURAL NETWORK: PREDICTING TIME SERIES DATA*

---

that of the proposed one. Hence, experimental analysis of the pandemic data (Dataset 1.(c) and 4.(b)) shows that for long-term prediction PLR-ALSTM-NN slightly lacks the ability to handle extreme changes in the data (highly skewed data with a high standard deviation) that needs to be addressed in the future. However, for normal circumstances, i.e. while using Dataset 1.(a), 1.(b), 2 and 4.(a), PLR-ALSTM-NN performs comparatively better than the existing methodologies while making long-term predictions.

### **2.6 Statistical Significance Test**

To prove the overall effectiveness of the proposed model, statistical tests are conducted, and the results are reported in the following section. A Friedman test is carried out using all errors (RMSE, MAPE and MAE) previously generated by each of the 10 runs for all approaches using two groups of the dataset. The 1<sup>st</sup> group contains the pre-pandemic data having Datasets 1.(a), 1.(b), 2 and 4.(a) and the 2<sup>nd</sup> group has the two remaining subsets of existing datasets where there are the extreme effects of the Covid 19 pandemic. This 2<sup>nd</sup> group has datasets 1.(c) and 4.(b). Using different seed values, all the models were trained 10 different times and the recorded RMSE, MAPE and MAE are used to conduct this non-parametric Friedman test. It is to be noted that RF, with its given parameter setting, produces poor accuracy in most cases and hence it is not considered further for statistical tests. With other remaining models, the conducted test provides a ranking for short- and long-term predictions on two categories of data. The mean ranks of the Friedman Test are reported in Table 2.24.

Table 2.24: Friedman Test: Mean Rank

	Dataset 1.(a), 1.(b), 2 and 4.(a)		Dataset 1.(c) and 4.(b)	
	Short Term Prediction	Long Term Prediction	Short Term Prediction	Long Term Prediction
PLR-ALSTM-NN	4.06	<b>2.41</b>	5.23	3.03
ARIMA	<b>1.00</b>	5.50	<b>2.20</b>	4.88
GRU	4.16	3.58	3.43	3.78
SimpleRNN	4.48	3.38	2.70	<b>1.88</b>
CNN	4.11	3.42	4.48	4.33
Basic LSTM	3.19	2.73	2.95	3.08

## 2. POLY-LINEAR REGRESSION WITH AUGMENTED LONG SHORT TERM MEMORY NEURAL NETWORK: PREDICTING TIME SERIES DATA

---

It can be observed from Table 2.24 that for Datasets 1.(a), 1.(b), 2 and 4.(a), ARIMA (lowest mean score: 1.00) leads the way for short-term prediction, while the proposed approach PLR-ALSTM-NN, with a mean score of 2.41, has acquired 1<sup>st</sup> position for long term prediction in comparison to the other approaches. For Datasets 1.(c) and 4.(b), ARIMA is again 1<sup>st</sup> for making 1 day ahead prediction with its lowest mean score of 2.20, whilst for 30 days ahead forecasting, the proposed approach PLR-ALSTM-NN is in 2<sup>nd</sup> place (Mean Score: 3.03) with RNN being the 1<sup>st</sup> (Mean score: 1.88). It can be noted that for both groups of data, PLR-ALSTM-NN performed considerably well with the 1<sup>st</sup> group (Dataset 1.(a), 1.(b), 2 and 4.(a)) while it marginally struggled on the 2<sup>nd</sup> group (Dataset 1.(c) and 4.(b)) where there are extreme price movements due to pandemic. Moreover, PLR-ALSTM-NN has acceptable performance while making long-term predictions on both groups, securing 1<sup>st</sup> and 2<sup>nd</sup> positions. To prove this fact, Wilcoxon's signed rank tests are carried out using both groups of data while considering the errors produced by different models of long-term forecasting only.

The long-term prediction performance of PLR-ALSTM-NN is further evaluated by another non-parametric statistical test i.e. Wilcoxon signed rank tests. The test is carried out considering the top three models from the Friedman Test from both data groups for long-term forecasting. For the 1<sup>st</sup> group, PLR-ALSTM-NN, Basic LSTM and RNN are chosen, while RNN, PLR-ALSTM-NN and Basic LSTM are selected from the 2<sup>nd</sup> group of data as top three. The Wilcoxon Signed Rank test results with different paired groups are presented in Table 2.25.

Table 2.25: Wilcoxon Signed Rank Test using Long Term Prediction Errors

Dataset 1.(a), 1.(b), 2 and 4.(a)					Dataset 1.(c) and 4.(b)				
Model	Better	Worst	p-val	Decision	Model	Better	Worst	p-val	Decision
PLR-ALSTM-NN vs Basic LSTM	70	50	0.013	+	PLR-ALSTM-NN vs RNN	13	47	0.00	-
PLR-ALSTM-NN vs RNN	65	55	0.001	+	PLR-ALSTM-NN vs Basic LSTM	21	39	0.166	≈
Basic LSTM vs RNN	63	57	0.361	≈	Basic LSTM vs RNN	12	48	0.000	-

For Datasets 1.(a), 1.(b), 2 and 4.(a) there are a total of 120 instances. Using a 5%

significance level, one of three signs (+, – and  $\approx$ ) is assigned, where “+” means that the first algorithm is significantly better than the second one, “–” means that it is significantly worse and ‘ $\approx$ ’ represents that there is no significant difference between two algorithms. It can be seen from Table 2.25 that PLR-ALSTM-NN has significantly better statistical performance when compared to Basic LSTM and RNN with p-values of 0.013 and 0.001, respectively, while there is not much significant difference between the performance of Basic LSTM and RNN. With Datasets 1.(c) and 4.(b) RNN is the best-performing approach with a statistical significance (p-value 0.000) when compared against PLR-ALSTM-NN and also with Basic LSTM while PLR-ALSTM-NN produces a p-value of 0.166 against Basic LSTM proving that the errors are not statistically significant. Hence, PLR-ALSTM-NN, with its multivariate analysis capability, is a generic approach with excellent performance over other approaches for long-term prediction, however, it loses accuracy marginally when the data is extremely skewed with high standard deviation with lots of uncertainties, such as in the case of the global Covid 19 pandemic.

## **2.7 Discussion**

Some discussion regarding the benefits and drawbacks of the proposed methodology is presented here. First of all, the proposed method produces comparable results for short-range prediction but it certainly delivers admirable performance in long-range forecasting. ARIMA, a well-established forecasting approach, works well for short-term prediction however for long-term analysis, the effect of non-linear relationships between the features and target data is more pronounced, which is more easily identified by PLR-ALSTM-NN, thus making more accurate predictions of long term stock market data. Secondly, the proposed approach slightly struggles when there are extreme changes in data (that may occur in case of a global pandemic or natural disaster). Such exceptional fluctuations are hard to detect by any ML or DL tools due to the lack of evidence of it in the historical data. These sorts of extreme scenarios occur very rarely, but the proposed approach can be a useful tool for making predictions for regular operations. The above-mentioned limitations

may be addressed in the future by fine-tuning the network further and hybridizing it with ARIMA so that the system can adapt to short-term analysis more efficiently.

It is common for stock prices to experience significant variability, non-linearity and non-stationarity over time, and these are correlated to multiple indicators. For short-term analysis, although not the best, PLR-ALSTM-NN has comparative performance over other approaches as it could not easily identify such relationships for predicting closing stock price when compared to other approaches. However, with a larger time horizon and normal data (without extremely unavoidable situations), these relationships seem to be more pronounced and thus successfully identified by the model due to its multivariate analysis ability. In case of extreme uncertainties in data, e.g. a global pandemic, the model may not be the best performing one but performs comparatively better ( $2^{nd}$  Position) than many other approaches. The model loses accuracy when there are extreme changes in data (highly skewed with high standard deviation), but given the fact that such extreme uncertainty infrequently occurs, for regular daily operation, the model could be beneficial to stakeholders and practitioners for making long time horizon forecasting which will be further discussed next in Section 2.8.

## **2.8 Managerial Implications: Practitioners' Benefit from Long-Term Forecasting**

The study related to long-term stock market predictions using AI would be especially beneficial for long-term investors. Instead of reacting to short-term rapid fluctuations, they will be able to focus on the long-term growth of a business or analyse the viability of a new business model. They will have the advantage of compounding dividends to create greater profit over time. Even more, investing for a long time creates no hassle about learning different trading systems or platforms. The most critical benefit is that long-run investment minimises investment risk by eliminating lost opportunities. According to research conducted by J.P. Morgan Asset Management [Morgan, 2016], trying to buy and sell stocks just at the right time could ultimately cause missing big up days in the financial

market. Hence, using ML predictions allows investors to stay invested over a longer time, thus minimising the chance of missing out on doing business on the best days. As investors make a profit, the business becomes wealthy by earning consumers' confidence. The overall productivity increases and this ultimately creates a positive impact on GDP.

## **2.9 Summary**

In this study, a model, PLR-ALSTM-NN, was developed that can make short-term, long-term, and intermediate days ahead predictions using multivariate time series data, such as the financial market S&P500 stock index. Although the underlying NN is SLSTM, its performance was further improved using poly-linear regression analysis along with some data augmentation techniques. The parameters are finely tuned using Taguchi's design of experiment approach so that the model produces more accurate predictions. It is often said that NNs are prone to over-fitting. Although many avenues have been explored to avoid over-fitting, early stopping is an efficient technique that is incorporated into the model to improve the overall performance of the model. The model is tested using four financial datasets that provide low detection error with very little deviation from the true values, in contrast to the results obtained from other literature.

The comparative analysis showed that the proposed model produces significantly more accurate predictions than the existing literature, especially for long-term forecasting, demonstrating the robustness of the model. The model proposed in this chapter is a general approach for time series forecasting, which was tested using four sets of stock market data to establish its value. It was also used in making sales predictions of supply chain time series data that demonstrates its acceptability in other fields. In future work, the model can be trained with the k-fold redcross-validation method, and it can be implemented in other domains like supply chains for making future predictions, with the aim of reducing overall costs, which would be widely beneficial.

The approach is tested on five sets of data. Among those data, it is believed that the

---

## *2. POLY-LINEAR REGRESSION WITH AUGMENTED LONG SHORT TERM MEMORY NEURAL NETWORK: PREDICTING TIME SERIES DATA*

---

sales do not have enough samples to optimise the network parameters, Hence it is just to show the model's suitability in other application fields. With the rest of the four datasets, the approach showed competitive performance for long-term forecasting through statistical tests. In particular, it showed significantly better accuracy when working with one group of the dataset where there is no effect of a global pandemic in comparison to other traditional approaches such as ARIMA, GRU, RNN, CNN, Basic LSTM and RF. For short-term predictions, ARIMA, with its best setting, performed consistently better than PLR-ALSTM-NN, whereas ARMIA did not show promising results in making long-term predictions.

The proposed approach PLR-ALSTM-NN, on the other hand, has the best forecasting accuracy for long-term forecasting with pre-pandemic data. This considerable performance of the model has been repeatedly shown by implementing it on different datasets and comparing its forecasting accuracy against some state-of-the-art approaches from the literature. With the current methodology, it can be observed that PLR-ALSTM-NN can make a stronger relationship with various factors for long-term analysis rather than for shorter analysis. The proposed model has some limitations too as it may fall behind when dealing with extreme data changes triggered by rare uncertain events. So for regular usage, more accurate long-term forecasting operations can be carried out with the proposed approach.

Hence, from this approach, practitioners such as long-term stock market investors adopting a buy-and-hold strategy will be able to enjoy the benefits by evaluating markets' long-term prospects and investing accordingly, which, ultimately, creates an overall positive impact on the economic growth of a nation. Given some of the limitations of the approach, it is shown through all the experimental analyses that the overall accuracy of PLR-ASLTN-NN is superior in cases of long-term prediction, at least in financial market data. However, shortcomings may be addressed in future by further fine-tuning the network. In the future, instead of tuning the network with the Taguchi DOA method, the hyper-parameters can be adjusted using meta-heuristic approaches, such as Genetic Algorithm (GA) or PSO, to further increase its accuracy.

## Chapter 3

# A Switching Based Forecasting Approach (SBFA) for Forecasting Sales Data in Supply Chains

*The work presented in this chapter has been reported in the following articles:*

- **Ahmed, S. S.**, Chakrabortty, R. K., Essam, D. L., and Ding, W.. “A Switching Based Forecasting Approach (SBFA) for Forecasting Sales Data in Supply Chains,” *Neurocomputing*, Under review 2nd round.
- **Ahmed, S. S.**, Chakrabortty, R. K., and Essam, D. L.. “Predictive big data analytics for supply chain demand forecasting,” In *Computational Intelligence Techniques for Sustainable Supply Chain Management*, ELSEVIER, Under review.

## Overview

*Forecasting future demand has been a challenging task for supply chain practitioners, which is further exacerbated due to the effects of the recent pandemic . As shown in the relevant literature, probabilistic distribution-based traditional forecasting approaches are less accu-*

*rate and generally lead to erroneous results, which ultimately leads to a misestimation of the overall cost of a business. Meanwhile, with the advancement of artificial intelligence (AI) approaches, such as machine learning (ML) and deep learning (DL), this misestimation of cost can be reduced by forecasting demand more accurately from historical data. Consequently, this chapter applies several AI-based approaches to predict demand data. According to the findings of the previous chapter (Chapter 2), it is evident that the forecasting performance of various approaches varies depending on both the forecasting horizon, as well as the unique characteristics of each dataset. Since no fixed AI approach works best for all datasets, a switching-based forecasting approach (SBFA) is proposed to exploit the merit of different advanced ML/DL approaches for different days ahead of prediction. Based on the performance of validation data, the proposed system automatically switches between different approaches to determine a more appropriate forecasting approach. A two-echelon supply chain model with different attributes is developed to validate the proposed SBFA against a few traditional forecasting approaches. The reorder points of this supply chain model are calculated based on the predictions from conventional/ML/DL forecasting approaches. Predictions from SBFA and other approaches are analysed by calculating overall supply chain cost. Based on overall supply chain costs under static and dynamic lead time settings, the effectiveness and applicability of the proposed SBFA against traditional forecasting approaches are demonstrated.*

### **3.1 Introduction**

As noted in the previous chapter, each forecasting approach has varying levels of accuracy depending on the forecasting horizon. While PLR-ALSTM-NN, proposed in the previous chapter, was effective for long-term forecasting, it has limitations for short and mid-term predictions. To minimise costs and optimise supply chain planning, a more robust approach is required that can make more accurate predictions over different forecasting horizons, regardless of a particular datasets' characteristics. Moreover, more precise predictions can lead to lower operating costs by enabling efficient inventory management, and

### *3. A SWITCHING BASED FORECASTING APPROACH (SBFA) FOR FORECASTING SALES DATA IN SUPPLY CHAINS*

---

lower ordering costs with reduced lost sales.

In inventory management, one of the primary concerns for managers revolves around determining how much and when to place orders. A commonly employed method for order quantity estimation in such scenarios is the Economic Order Quantity (EOQ), which is widely used for quantity estimation. However, it's important to note that EOQ is just one of several methods available. The accuracy of EOQ calculations relies heavily on the precise estimation of customer demands, as the effectiveness of inventory models hinges on the correctness of demand predictions. An accurate inventory model, or precise demand data estimation, plays a pivotal role in enabling businesses to effectively manage and adapt to unexpected shortages. Nevertheless, forecasting future demands is a challenging endeavour and often relies on historical data. The inherent volatility in customer demand patterns, the potential for mass customization across manufacturers, and the vast volume of customer data, including big data, can render traditional forecasting approaches (e.g., moving averages, exponential smoothing, naive methods) inefficient [Zhu et al., 2021].

It's worth emphasizing that estimating demand accurately is a critical aspect, serving as an input for various inventory management models, including EOQ calculations for cycle inventory, safety stock inventory estimation, and reorder point estimations. In practice, EOQ is commonly used for order quantity estimation, but it's essential to recognize that its effectiveness may be limited when dealing with fluctuating and unpredictable demand patterns.

Meanwhile, the emergence of Artificial Intelligence (AI) has been significantly transforming industries in the last decade. Industrial evolution (e.g., industry 4.0, industry 5.0) also focuses on digitizing business communications. Deep learning (DL) and Machine Learning (ML) approaches have recently been introduced to predict customer demands [Pereira and Frazzon, 2021, Tirkolaee et al., 2021, Zhu et al., 2021]. With the digitization of the world, supply chain historical data is now more easily accessible, which makes ML/DL models easier to make future predictions. The forecasting ability of DL/ML from historical data can be used to predict future demand, which can help a business to set better their reorder point and lot size. Instead of traditional approaches, ML-based inventory management

### *3. A SWITCHING BASED FORECASTING APPROACH (SBFA) FOR FORECASTING SALES DATA IN SUPPLY CHAINS*

---

will help businesses operate at a lower cost than conventional approaches. An accurate prediction can also significantly improve the response time for customers. The overall effect on lost sales or holding cost or order cost is minimised, thereby minimising overall supply chain cost [Seyedan and Mafakheri, 2020].

Even though many researchers have developed time-series forecasting models based on statistical methods, those models cannot handle complicated market situations. Statistical forecasting models, being mostly uni-variate analysis methods [Weller and Crone, 2012], struggle to perform under complex situations where there is a nonlinear relationship between data that may arise due to strong seasonality of sales, trend, volatile demand or even with lack of historical data [Deng et al., 2021, Thomassey, 2010]. Recently, scholars have used Decision Tree Based (DTB) approaches and neural network (NN) models for effective management of issues related to supply chains, such as supply chain risk forecasting [Yan et al., 2019, Kamble et al., 2021], supplier selection [Guo et al., 2009], demand estimation and [Ni et al., 2020], inventory management through lead time forecasting [Gumus et al., 2010].

Researchers are also working on sales estimation [Spiliotis et al., 2021] of retail industries, where the authors used DTB approaches for better accuracy. In contrast, approaches from NNs are also combined and explored by Li et al. [2021] to obtain more accurate time series sales forecasting. Simulation of supply chain models is implemented by Kantasa-Ard et al. [2021] to minimise costs through adopting simple forecasting approaches from the NN group. Some scholars are working on improving prediction accuracy, while others are trying to reduce supply chain costs through modelling. Little evidence of research demonstrates how much supply chain costs are being reduced by more accurate AI prediction. Moreover, no universally accepted approach performs better in estimating short to long-time horizons with all-time series datasets. It is hard to differentiate between DTB and NN approaches, as both groups are widely used and exhibits higher accuracy and stability while forecasting nonlinear and non-stationary time series data [Gu et al., 2020, Masini et al., 2021] with many features.

Therefore, scholars have yet to explore a combination of efficient forecasting models be-

### *3. A SWITCHING BASED FORECASTING APPROACH (SBFA) FOR FORECASTING SALES DATA IN SUPPLY CHAINS*

---

longing to the DTB category and NNs. Even more, such a model's prediction performance over a long time horizon is little explored. Additionally, only a few research works have been conducted to demonstrate the effectiveness of a better forecasting approach in the context of supply chains. In this research, a hybrid forecasting model is proposed that can switch between two groups of models and select the best one based on a validation score as the models are trained to predict different days ahead into the future. During the training phase, DTB models are fine-tuned using an exhaustive grid search method in order to get the best outcome from each of them. The parameter settings related to NN models are kept constant to save time while still, at times producing considerably better predictions when compared to other DTB approaches. The practicality of better predictions is later demonstrated using a supply chain model to reduce overall cost.

To close the above-mentioned research gaps, this chapter adapts the following methodology. This segment uses ML to analyze historical sales data to predict future sales. The predicted sales are then fed through a two-echelon supply chain model with a specific dynamic lead time (LT). The model calculates the reorder point (ROP) against a fixed lot size  $Q$  from the lead time predicted sales values instead of the traditional average sales values. The model operates using a  $(r, Q)$  policy where  $Q$  is fixed, and  $r$  is the decision variables that have been calculated from the predicted sales data. The supply chain model carries out daily operations and replenishes its stock once the inventory and order on delivery are less than the reorder point. To check if the model is fully operational, some conventional forecasting tools, such as: moving average, exponential smooth and normal distribution, are deployed to make predictions from which reorder point, and the associated costs of the supply chain model may be calculated. The developed supply chain model is also further tested with predictions from a switching-based forecasting network (which is formed from DTB and NN approaches) to show that better accuracy leads to lower supply chain costs.

Hence, in short, the contributions stemming from this work are multi-fold and include:

- A switching based forecasting approach (SBFA) is proposed by combining advanced forecasting approaches, where the switching is done based on the lowest validation

error for different datasets.

- A two-echelon supply chain model with different attributes is developed to validate the proposed SBFA against a few traditional forecasting approaches. The reorder points of the supply chain model are calculated based on the predictions from conventional/ML/DL forecasting approaches.
- Predictions from SBFA and other approaches are analysed by calculating the overall supply chain cost.
- A theoretical derivation is developed to establish a selection procedure among the forecasting approaches considering the trade-off forecasting horizon, accuracy, time and money saved.

The rest of this chapter is organized as follows: Section 3.2 presents a brief overview of existing literature related to sales/demand forecasting approaches and their applications in supply chain models. Based on the existing literature, Section 3.3 provides a brief summary of traditional and advanced forecasting approaches. The methodology of SBFA is described in Section 3.4, followed by modelling and validation of our supply chain network in Section 3.5. Section 3.6 discusses the details of the experimental setup, while findings are described in Section 3.7. The findings of this research work are further enhanced through the development of a Theoretical Derivation presented in Section 3.8. Section 3.9 discusses the ultimate benefit of the stakeholders from the SBFA approach. Finally, a summary and future directions are presented in Section 3.10.

## 3.2 Literature Review

With poor demand forecasting, companies face two negative business cases: overstocking and/or understocking. Overstocking creates wastage and increases unnecessary inventory holding costs, whereas being out of stock leads to increased lost sales where customers cannot purchase their desired products and may instead purchase them from competitors

### *3. A SWITCHING BASED FORECASTING APPROACH (SBFA) FOR FORECASTING SALES DATA IN SUPPLY CHAINS*

---

[Kilimci et al., 2019]. In order to deal with such problems, demand forecasting is critical to supply chain managers. Time series forecasting techniques have been considered an essential research topic in this regard [Luce, 2019, Mihirsen et al., 2020]. Many researchers have put a lot of effort into developing and improving time series forecasting techniques' performances across many applications in the real world. For example, electrical signals analysis [Lu et al., 2019], traffic condition analysis [Priambodo and Jumaryadi, 2018], weather forecasting [Wang et al., 2016], rate of unemployment analysis [Katriss, 2019] and financial market analysis [Thakkar and Chaudhari, 2021]. Among the application areas, ML forecasting approaches have to deal with different types of datasets. Scholars, like Darvazeh et al. [2020] and Tirkolaee et al. [2021], found that the performance of ML and DL approaches depends on the volume and type of data, such as structured, unstructured and/or nonlinear data. Thus, a forecasting approach that combines multiple ML/DL approaches can enhance ultimate performance when trained on such diverse datasets.

The decision tree-based approach (another branch of neural networks that researchers also explore to make sales predictions) has also been used recently to forecast sales data. For example, Niu [2020] forecasted Walmart Sales data by using XGBoost while daily retail demand was forecasted by Huber and Stuckenschmidt [2020] using LGBM, in which authors considered external factors, including special calendar days. The authors found that their approach was more suitable for large-scale demand prediction situations than other ML approaches. K-Nearest Neighbour (KNN) regression was used by Kohli et al. [2021] to estimate sales, and Ramachandra et al. [2021] used a random forest (RF) algorithm to estimate black Friday sales. Among the recently published literature on sales/demand forecasting using neural networks, Muhammin et al. [2021] used recurrent neural network (RNN) to forecast demand from Walmart sales data, where the authors showed by using a comparative analysis with conventional methods, such as DNN and SES, to prove RNN's superiority with that data over other approaches. jiang Li et al. [2021] proposed a combined method using GRU, Prophet and Attention Mechanisms to predict clothing sales. From this, it is quite clear that practitioners and researchers can use combinations of machine learning and/or deep learning approaches to provide better solutions for time series forecasting problems. Deep learning methods provide better results than classi-

### *3. A SWITCHING BASED FORECASTING APPROACH (SBFA) FOR FORECASTING SALES DATA IN SUPPLY CHAINS*

---

cal techniques when properly configured and trained [Mahmoud and Mohammed, 2021]. However, it is hard to find one particular approach that works best for all datasets, as the relative performance of forecasting approaches depends on the size and nature of the data.

In addition to the approaches mentioned above to forecasting sales or demand data, different time series forecasting approaches are also common in the supply chain literature. Experts specializing in supply chains use ML/DL approaches to take managerial decisions as products move through a chain. To predict sales, Zheng and Li [2018] used an ARMA (autoregressive moving average) model to forecast the purchase demand of unmanned retail outlets, such as smart vending machines, situated at different locations. The authors then used the predicted demand to design an efficient transportation route for stock replenishment that incurred lower costs. While designing the ML model, the authors considered only 1 day ahead forecast, i.e. short-term prediction. However, midterm (15 days ahead) to long-term prediction (30 days ahead) systems could lead to some interesting research areas.

A similar type of sales forecasting was developed by Weng et al. [2019], where a combination of lightGBM and LSTM was proposed to improve overall prediction performance. The authors conducted experiments on 3 types of sales data and proved that feature engineering with the hybrid DL model could increase the accuracy of predicted sales. The authors also suggested that the predicted results are interpretable, which could be used to improve enterprise supply chains' efficiency in operation, production, daily management, price management and marketing. Hence, an extension of their forecasting approach to the supply chain could be implemented to analyze the actual benefits of accuracy in terms of cost. Taking satellite data of container vessels travelling between Europe and Asia, Viellechner and Spinler [2020] predicted delay in the arrival of vessels by using neural networks and support vector machine classification models. The predicted delays were interpreted by port managers, which ultimately helped reduce congestion and improve the efficiency of ports. The predictions influenced the decision-making processes of the managers involved. However, time series forecasting of vessels' delays using ML and its

### *3. A SWITCHING BASED FORECASTING APPROACH (SBFA) FOR FORECASTING SALES DATA IN SUPPLY CHAINS*

---

managerial implications on the overall supply chain can be further explored.

Information sharing through digitization, collaboration and partnerships between different supply chain entities can reduce the gap between demand and supply and result in lower cost and buffer stock while simultaneously enhancing customer satisfaction levels [Pereira and Frazzon, 2021]. That being the case, Pereira and Frazzon [2021] proposed a data-driven method by combining ML and designing an omnichannel retail supply chain through a simulation-based optimization technique. Their adopted technique synchronized demand and supply and led to a reduction in operational costs. This encourages business owners to incorporate demand forecasting approaches to improve resiliency, which takes them a step ahead in their competitive markets [Sharma et al., 2021]. Although omnichannel retail looks more profitable, it often leads to a weaker profit margin as extra expenses are associated with each channel, such as storage, packaging and delivery through multiple sites, which needs extra maintenance costs. Thus, ML/DL forecasting techniques, in combination with a mono-channel supply chain model, could be more profitable.

Case-based research can lead to novel insights and the development of new theories since it is unconstrained by rigid limitations and assumptions associated with surveys and models. Phillips and Nikolopoulos [2019] recently advanced case-based research by successfully conducting a three-year action research intervention at a pharmaceutical company. By applying forecasting theory/principles in practice, the researchers assisted practitioners in improving their use and understanding of a forecasting function to minimise inventory and waste and improve supply chain efficiency, thereby saving the company millions of dollars. Despite the difficulties associated with case-based action research (such as companies being reluctant to engage and implement changes fully), this is a commendable step towards rectifying the gap between theory and practice. Although it may be true that case studies can be used to explore and investigate events deeply and thoroughly. Still, according to the same author Phillips [2021], mathematical modelling of the supply chain can provide useful information to stakeholders through problem simplification techniques.

More accurate demand predictions reduce the bullwhip effect on supply chains [Wiedenmann and Größler, 2019]. Yousefi et al. [2018] used a hybrid approach to predict uncertain

demand and used that prediction in a 3-stage supply chain, proving a reduction of the bullwhip effect with accurate forecasting. The authors used a uni-variate dataset from Lynx Company and used that to train a hybrid model consisting of discrete wavelet theory and an artificial neural network for making predictions. However, using a multivariate dataset with other DL approaches, e.g. CNN, LSTM and RNN, and considering the impact of forecasting on aspects such as shortage or excess demand, delay etc., of supply chains is yet to be explored.

### **3.2.1 Research Gap**

Sales predictions made by advanced ML/DL approaches, capable of dealing with multivariate data, and their accuracy over different time horizons, can lead to interesting research avenues. According to a comprehensive study conducted by Tirkolaee et al. [2021], it is suggested that managers and practitioners should wisely choose suitable forecasting algorithms for the industry, as the performance of approaches depends on the type and volume of data. Hence, there is still scope for improvement through developing a hybrid forecasting approach that can be applied to all types of data. On top of that, finding out how these predictions improve the efficiency of supply chains in terms of cost can be even more fruitful. Although there is evidence of case studies and modellings of supply chains that use ML predictions to reduce cost, a simplified model can provide unforeseen insights into business owners' changes in supply chain costs, thus enabling them to make necessary preparations beforehand. Thus, this chapter presents a switching-based forecasting approach (SBFA) that is constituted from some state-of-the-art forecasting models that are based on DTB approaches (such as XGBoost, LightGBM, K-Nearest Neighbour, Random Forest ) and NNs (including RNN, LSTM, GRU and CNN), for product demand forecasting from sales data. This approach is applied to a supply chain model to demonstrate that this improved demand forecasting leads to lower overall supply chain costs.

### **3.3 Brief Introduction to Forecasting Methods**

In this section, a short summary regarding the working principles of some classical forecasting techniques, followed by some advanced ML/DL approaches, is given. All the techniques are adopted from recently published literature and are mostly related to sales and demand forecasting in supply chains.

#### **3.3.1 Traditional Forecasting Technique**

The implementation of three traditional forecasting techniques is discussed in this subsection. These techniques are Normal Distribution (N.Dist), Moving Average (MA) and Simple Exponential Smoothing (SES). These techniques are long-established and require fewer parameters than other advanced approaches. Later in this chapter, to establish their practicality, the prediction accuracy of each of these models will be compared and further analysed using a supply chain model.

##### **3.3.1.1 Normal Distribution (N.Dist)**

Although it may not be an ideal forecasting technique, it is usually assumed that real-world demand is normally distributed [Ernst and Kamrad, 2006]. According to Disney et al. [2021], demand is said to be normally distributed when such demand, originating from independent customers, is accumulated into daily buckets with the central limit theorem in mind. The authors used a normal distribution system to forecast demand and then used the forecasted demand data to analyse the behaviour of lost sales in a supply chain model.

In another research, mean and standard distribution of historical data were used to present a more precise model of future sales [Arinaminpathy et al., 2016]. Taking this as motivation, a similar approach is explored in this chapter. Using the Kaggle dataset upto 30 April 2016 (Refer to Table 3.6 for more information), mean, standard deviation, max and

min values are denoted and artificial demands were created for the remaining period using a truncated normal distribution system. The truncation was needed to avoid negative sales/demand and to keep maximum demand within a certain range.

Table 3.1: Extraction of mean, standard deviation, min and max value from the Kaggle Dataset (Subset A and B)

Period	Time-steps	Mean B	Std. Dev	Min	Max
1 Jan 2013 -30 April 2016	1216	18.5880	6.2847	4	43

Using information from Table 3.1, future sales are predicted for the next period of analysis which is from 1 May 2016 to 31 Dec 2017. The predictions will be later fed into a supply chain model for further evaluation.

### **3.3.1.2 Moving Average (MA)**

Moving Average is another classical and simple prediction technique that can be considered for predicting sales data. The forecast of a target variable is the average of its value over the previous n-time steps [Tsoumakas, 2019]. As n gets larger, the prediction becomes smoother over time, and it fails to capture sharp changes in data. At the same time, a lower value of n can be used to detect larger changes in data. For this chapter, using the Kaggle Dataset, a simple MA is implemented with n=1 that is capable of estimating sales over a time horizon of up to 30 days. Subsets A and B were used to train the model, and Subset C was used for testing (refer to Table 3.6 for more information).

### **3.3.1.3 Simple Exponential Smoothing (SES)**

One more conventional time series forecasting approach is used in this segment of the thesis, i.e. Simple Exponential Smoothing (SES). This method is applicable for univariate data without trend or seasonality. Forecasted values produced by the SES approach are the weighted means of previous observations, with weights that progressively decay exponentially as observations get older. Hence, a higher weight is associated with more recent

data.

SES needs a single parameter called the smoothing coefficient. This parameter controls the exponential decay rate of the influencing observations at prior time steps. The value ranges between 0 and 1 where a larger value (closer to 1) indicates that predictions are only influenced by the recent past values, and smaller values (closer to 0) represent a large influence of past observations on forecasted values. In this chapter, SES is also implemented in Python through the Statsmodels class, where the model automatically optimises the smoothing coefficient.

#### **3.3.1.4 Autoregressive (AR)**

A familial forecasting approach is an Autoregressive (AR) model, which determines the future behaviour of data concerning its past behaviour. It is a stochastic system and is mainly used for predictions when there is some correlation between the values in a time series pattern. So in AR, the predicted value depends on previous values and some random noise known as lags and  $p$ . The AR method presented in this chapter is mainly adapted from Kamalov et al. [2021], where the best-performing lag parameter ( $p=2$ ) is chosen for comparison against other traditional approaches. AR is implemented in Python through Statsmodels, which is similar to implementing SES.

### **3.3.2 ML/DL approaches**

The working principles of state-of-the-art ML/DL models are briefly explained in this section. These explanations are collected from recently published literature. This section also contains how each of these approaches is slightly tailored so that they are suitable for making sales predictions using a combined forecasting system, which is the main contribution of this chapter.

### 3.3.2.1 XGBoost (XGB)

According to Niu [2020], the XGBoost approach has been adopted by many researchers as it has relatively better accuracy, higher speed and strong anti-noise ability. XGBoost is a kind of regression tree that has the same decision rules as those of a classical decision tree. It consists of numerous decision trees with every tree created by the gradient descent method, and optimisation is done through the minimisation of the objective function. Each inner node of the regression tree constitutes values for attributes tests and a decision is represented with leaf node scores. The final output is the summation of individual scores forecasted by K trees, as shown in equation 3.1.

$$\hat{y} = \sum_{k=1}^K f_k(x_i), f_k \in F \quad (3.1)$$

Here,  $x_i$  is the  $i^{th}$  sales training sample,  $f_k$  is the  $k^{th}$  tree score and  $F$  represents the space function that contains all regression trees. XGBoost uses a similar gradient boosting technique from Gradient Boosting Machine (GBM) [Friedman, 2001] but makes a slight advancement on the regularized objective mentioned in equation 3.2, that penalizes models' complexity.

$$L = \sum_i l(\hat{y}, y_i) + \sum_k \Omega(f_k) \quad (3.2)$$

Where,  $L$  represents the total objective function, the distance between the predicted value and the true value is measured by a differentiable convex loss function known as  $l$ . The regularization term  $\Omega$  is expressed by equation 3.3.

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \|w\|^2 \quad (3.3)$$

Here, the constant coefficients are  $\gamma$  and  $\lambda$ ,  $T$  is the number of leaves in the tree, and the score of each leaf is represented by  $w$ . The difference between GBM and XGBoost is that the loss function of XGBoost is extended using a second-order approximation and the constant term is removed to reach the simplest goal [Shilong et al., 2021]. This helps the

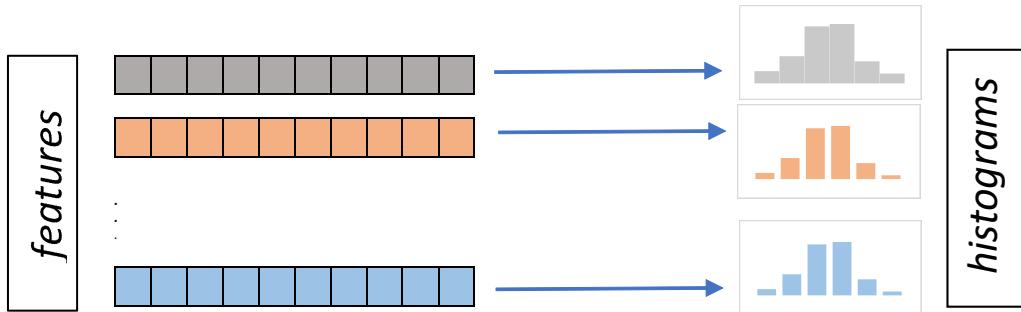


Figure 3.1: Establishing statistical histograms

system to optimise the objective function in its general setting quickly. Hence, XGBoost can be used to solve a wide range of problems, making it a suitable choice for our approach.

### 3.3.2.2 LightGBM (LGBM)

Another well-known model in the series of gradient boost regression trees is LightGBM and it is sometimes considered an alternative neural network [Huber and Stuckenschmidt, 2020]. LightGBM has added benefits of faster training, lower memory and higher accuracy when compared to its predecessors. The gradient calculation performed by LightGBM is based on Histogram. Each feature is interpreted as a statistical histogram, and the optimal split point is found according to its discrete value, as depicted in Figure 3.1. The node's histogram can be inherited from its parent node and from its siblings. This further speeds up the calculation process. Based on the histogram, a Leaf-wise growth strategy is adopted by LightGBM for creating regression trees, as shown in Figure 3.2. Whereas XGBoost supports level-wise tree growth, the Gradient-based One-Side Sampling technique is used by LightGBM to sample and weight assignment to each sample. Moreover, the dimensionality of a high-dimensional sparse dataset can be reduced.

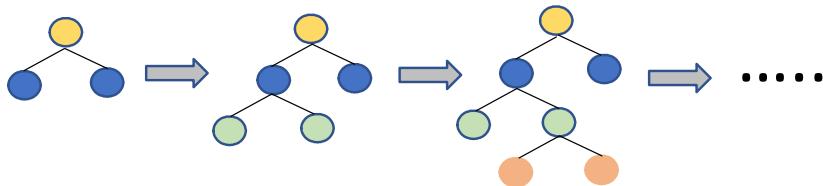


Figure 3.2: Leaf-wise growth strategy

### 3.3.2.3 K-Nearest Neighbour Regression (KNN)

The k-nearest neighbours algorithm predicts values on the basis of a similarity measure calculated from previously stored instances. A technique known as ‘feature similarity’ is used to forecast the values for the test dataset [Kohli et al., 2021]. The prediction is made depending on the resemblance to a training dataset. There are two main steps in KNN regression. Firstly, it calculates the mean of the target data of the K-nearest neighbours. Then, an inverse distance weighted average is computed of the K-nearest neighbours. Just like KNN classification, KNN regression uses the same distance measuring functions, such as Minkowski, Manhattan or Euclidean.

### 3.3.2.4 Random Forest Regressor (RF)

Random Forest (RF) is a well-known statistical learning approach that has the ability to perform both classification and regression tasks using numerous decision trees and a Bootstrapping technique, usually known as bagging [Correia et al., 2020, Gao et al., 2019]. The RF model can capture non-linear interaction between features and a target, and it exhibits excellent performance while dealing with data having several categorical or numerical features [Ramachandra et al., 2021]. The main idea behind Random Forest is the combination of multiple decision trees while deciding the final value rather than depending on any particular decision tree. This approach allows RF to obtain more reliable predictions. Thus, it can be defined as a kind of additive model that forecasts sales from the combinations of decisions from various base models. The equation of RF is stated in 3.4.

$$P(x) = f_0(x) + f_1(x) + f_3(x) + \dots \quad (3.4)$$

### 3.3.2.5 Recurrent Neural Network (RNN)

A simple representation of NN is a recurrent neural network (RNN). It is a powerful tool for time series data (sequential data) [Muhaimin et al., 2021] as it can predict the next output from previous inputs. RNN has repetitive loops, and these loops are the hidden units/neurons for one particular hidden layer of the network. These loops allow neurons to store past input information for some time so that the network, consisting of hidden layers, can forecast the output. When the number of iterations is finished, the output of a recursive neuron is transmitted to the next layer. Hence, the previous information is kept for longer and makes the output more comprehensive Apaydin et al. [2020]. Lastly, mean square errors are calculated and propagated backward to update their weights. Thus, in this study, a two-layer stacked RNN network is used and its architecture is stated in Figure 3.3. The cell structure of RNN is also shown in Figure 3.4.

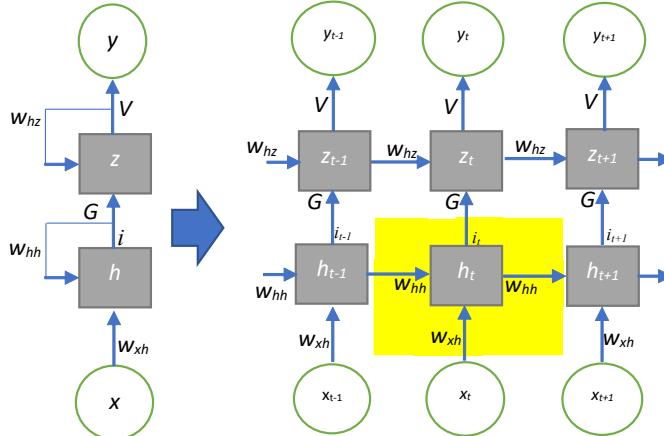


Figure 3.3: Folded and unfolded architecture of a 2 -layer RNN

The equation of a simple RNN cell (highlighted memory cell) can be represented by Equation 3.5 and 3.6

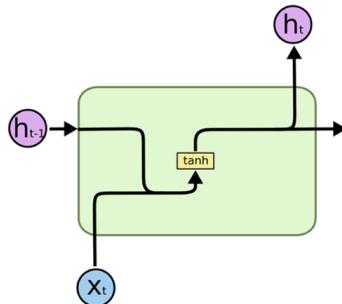


Figure 3.4: RNN cell structure [Olah, 2015b]

$$h_t = \tanh(W_{hh}h_{t-1} - 1 + W_{xh}x_t) \quad (3.5)$$

$$i_t = Gi_t \quad (3.6)$$

where,  $W_{xh}$ : weights of the input layer to the hidden layer;  $W_{hh}$ : own weights of the cell that changes according to the change in cell state;  $G, V$ : weights of the connections between hidden layers and between the hidden layer and the output layer respectively;  $h_t$ : new state;  $h_{t-1}$ : previous state, while  $x_t$  and  $y_t$  are the current input and output respectively.  $\tanh$ : activation function

### 3.3.2.6 Long Short Term Memory (LSTM)

RNN can store short-term data but it cannot maintain long-term memory and this drawback of RNN is overcome by the LSTM approach. So an advancement of RNN for handling long sequential data is LSTM. Like RNN, a similar type of two stacked layers of LSTM network is used in this study and a brief explanation of the working principle of a single memory cell of LSTM is stated in this section and its architecture with interconnection is represented in Figure 3.5. An LSTM cell consists of three gates: input, forget and output. The gates operate in such a way that each line carries some particular information, as explained by Equation 3.7, 3.8, 3.9, 3.10, 3.11 and 3.12.

### 3. A SWITCHING BASED FORECASTING APPROACH (SBFA) FOR FORECASTING SALES DATA IN SUPPLY CHAINS

---

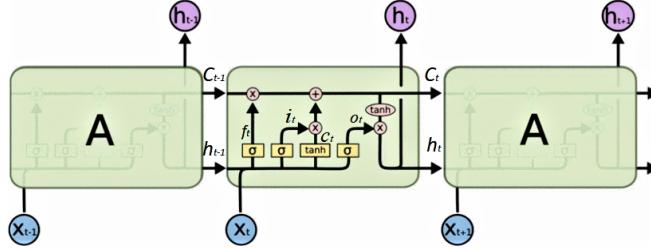


Figure 3.5: LSTM's architecture [Olah, 2015b]

In RNN, the network learns by a technique of gradient descent and this gradient becomes too small to have any influence on the learning of the network when it trains over a large sequence of cells since errors propagate throughout the network. This problem is called Vanishing Gradient Descent, and LSTM overcomes this issue as the architecture has a memory line at the top and bottom to represent the output i.e.  $h_t$  and  $x_{t-1}$  represents the previous block, whereas  $x_{t+1}$  represents the next future block where  $x_t$  becomes the input of that future block. Through back-propagation, the learning process approximates the weights and lets data be stored or deleted in the cells. According to Figure 3.5, the transition equations of LSTM are stated below.

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + V_i c_{t-1}) \quad (3.7)$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + V_f c_{t-1}) \quad (3.8)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + V_o c_t) \quad (3.9)$$

$$\tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1}) \quad (3.10)$$

$$c_t = f_t^i \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (3.11)$$

$$h_t = o_t \odot \tanh(c_t) \quad (3.12)$$

where,  $i_t$  denotes the input gate and  $o_t$  denotes the output gate.  $\odot$  characterises a point-wise product. The forget gate, memory cell, and hidden state are denoted by  $f_t$ ,  $c_t$ , and  $h_t$ , respectively [Graves and Schmidhuber, 2005].

It can also be noted that the result is impacted by the output coming from its previous block. Each block has three inputs and two outputs, and all of these are vectors i.e. all

these variables contain arrays of values rather than one value. The point-wise operations are like values, and their opening or closure is mainly governed by the sigmoid function. If open, information is allowed to pass through freely. The value is either added to the memory or not depending on the decision of the operation and if so is passed to the next layer for operation. The point-wise operation contains values between zero and one, and the output value is in the last point-wise operation where it decides how much of the memory reaches the output. There is various LSTM architecture available, however, the basic functionalities are the same.

### 3.3.2.7 Gated Recurrent Unit (GRU)

GRU is a similar, but more simple, type of LSTM network. The difference between LSTM and GRU is that the input and forget gates of LSTM are combined together to form an update gate. Hence, there are fewer parameters in GRU than in LSTM, which makes training easier [Apaydin et al., 2020]. The structure of a single cell of GRU is shown in Figure 3.6.

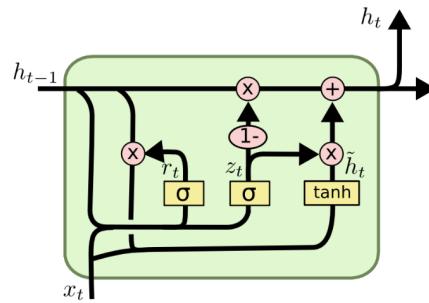


Figure 3.6: Gated Recurrent Unit (GRU) [Olah, 2015b]

The output of GRU is  $h_t$ . It can be computed in the following ways:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \quad (3.13)$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \quad (3.14)$$

### 3. A SWITCHING BASED FORECASTING APPROACH (SBFA) FOR FORECASTING SALES DATA IN SUPPLY CHAINS

---

$$\tilde{h}_t = \tanh(W_h x_t + (r_t * h_{t-1})U_h) \quad (3.15)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \quad (3.16)$$

where,  $r$  represents the reset gate and  $z$  is an update gate. An element-wise product among two vectors is implied by the operand  $*$ . The current input can be combined with the past memory in the reset gate. The amount of earlier memory that needs to be kept is controlled by the update gate. A fully preserved previous memory is represented by an update gate value of 1, and a fully forgotten previous memory is represented by a 0. In LSTM, the forget gate automatically calculates how much previous information is maintained. Whereas, in GRU, all past information is fully kept or completely removed. Experimental analysis shows that the performance of GRU is comparable to that of LSTM when provided with the same quantity of parameters but with the added benefit of a shorter training period [jiang Li et al., 2021].

#### 3.3.2.8 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) is a type of deep NN that is commonly used in image analysis. With time series data, Convolutional layers have the ability to extract vital information and learn the internal patterns of data [Livieris et al., 2020]. The Convolution operation defines the inner product (multiplication and summation) of window data with a filter matrix which is a fixed set of weights. The purpose of the maxpooling layer is to simultaneously reduce the eigenvectors' size to simplify the network's computational complexity and compression and extraction of the main features. Figure 3.7 shows a one-dimensional CNN model.

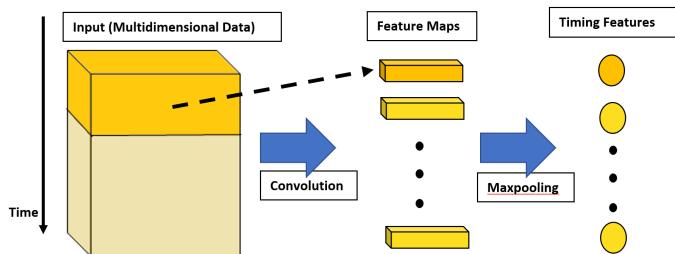


Figure 3.7: One dimensional CNN [Jin et al., 2020]

In this chapter, for experimentation, two layers of a convolutional network are implemented with a maxpooling layer in between. The output of the final convolution layer is passed through a flattened layer, whose purpose is to convert the output into a one-dimensional array so as to create a single long feature vector. Finally, like other NN approaches, a dense layer is used to combine that single vector to produce the predicted result.

### **3.3.2.9 Transformer Encoder Architecture (TEA)**

Although LSTM can remember information regarding long sequence data, Transformer models can deal with much longer sequences. These models have the power of parallel computation and can learn multiple representations of inputs. Transformer models are especially well-suited for time series data since they are built on a multi-headed attention mechanism. These models simultaneously represent each input sequence element by considering its context (future-past), and multiple attention heads consider various representation subspaces. In the case of time series data, this may correspond to various periodicities of the input signal. The transformer uses an encoder-decoder design, and its encoder and decoder are stacked using different independent feature extractors.

In this chapter, transformer encoder architecture is used, while the decoder part can be implemented in future. According to recent articles [Zerveas et al., 2021, Afteniy, 2021], the whole encoder-decoder system is more suitable for machine translation or natural language processing due to its ability to produce a sequence of no pre-specified length.

For a thorough explanation of the transformer model, readers can be directed to the original work [Vaswani et al., 2017]. In this section of the thesis, the idea was adapted from recent literature [Rathnayaka et al., 2022, Ramos-Pérez et al., 2021] and a vanilla Transformer encoder model [Jin et al., 2021] is implemented for different days ahead of prediction with multivariate data mentioned in this chapter, and a fair comparison is made with proposed SBFA. The architecture is the Transformer Model is depicted in Figure 3.8.

Sine and Cosine functions are used for positional encoding of different cyclic features the

### 3. A SWITCHING BASED FORECASTING APPROACH (SBFA) FOR FORECASTING SALES DATA IN SUPPLY CHAINS

---

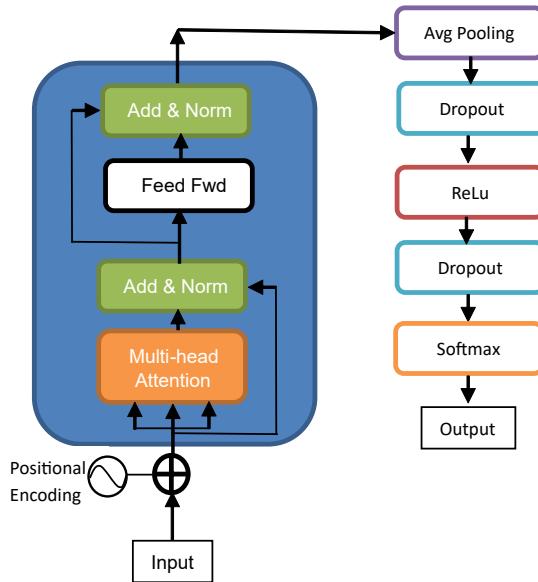


Figure 3.8: Classical Transformer Encoder Architecture [Jin et al., 2021]

data such as 'month', 'week', 'day' to obtain positional values and then appended with the input sales data to create the final data stream for training and testing the network.

In literature, all Transformer networks are implemented using a fixed set of layers and hyper-parameters, however, fine-tuning the parameters and layers could create a more accurate prediction. In this chapter, a Transformer network is initially implemented according to the literature [Jin et al., 2021, Vaswani et al., 2017], and then several experiments were conducted with different parameters and layers to enhance the accuracy of the model. Removing the last four layers and adding a dense layer after the Avg Pooling layer with a smaller learning rate than 1 (mentioned in the literature) produces a much better prediction on both datasets. The setting producing the minimum RMSE is also presented alongside the RMSE of the model produced by the implementation of the model from the literature. In both cases, the average RMSE is higher than the proposed SBFA, but it is believed that further tuning the network could produce a much lower error. Regarding training time, a higher learning rate (1.0) from literate takes a significant amount of time as compared to our SBRA method, while lowering the learning rate (1e-4) decreases the training time. All experiments regarding the Transformer model are performed for 10

runs for each day ahead of prediction with fixed layers and parameters. The accuracy of the model can be further enhanced by using a grid search approach, but the training time would be significantly higher in that case. In both cases, the SOTA Transformer architecture's accuracy (in terms of RMSE) is lower than that of the SBFA approach with a comparable training time.

### **3.3.2.10 Graph Neural Network (GNN)**

Graph Neural Network (GNN) is another deep learning technique that is intended to make inferences from data described by graphs. GNN is widely applied in a vast range of fields, starting from medical sciences to predicting the molecular properties of quantum. It is also vastly used in NLP. GNN can be used for classification and regression purposes, and in this portion of the thesis, the regression property of GNN is used for making sales predictions. For comparison with the SBFA model, a GNN is implemented according to the model 'Graph 1' mentioned in literature [Zanfei et al., 2022]. Numerous GNN models are deployed in the literature to make time-series water demand forecasts. The best-performing model architecture and parameters are adapted in this chapter for comparison. The GNN model is based on an open-source Python function called Spektral [Grattarola and Alippi, 2021] which is built on TensorFlow and Keras API framework.

From Spektral, a basic GraphConvolution graph layer is implemented that employs a series of convolution operations within learnable weights, external node features and correlation matrixes. The GNN model has 3 layers of Graph Convolutional Layers (GCN) for graph feature extraction. Then Temporal feature extraction of the sequential data is done through two LSTM layers. The parameters are also selected based on the best-performing model mentioned in the literature. The first 3 GCN layers have sizes 32, 32 and 18, while the remaining two LSTM layers have sizes 18 and 32 units, respectively.

The available Kaggle and Rossmann datasets are split into three parts: training, validation and testing, as described before, and each subset is basically restructured into three parts. The actual sales data has a sequential form, as usual, while the remaining features (year,

### *3. A SWITCHING BASED FORECASTING APPROACH (SBFA) FOR FORECASTING SALES DATA IN SUPPLY CHAINS*

---

month, week, day, trend, seasonal) are used to create a separate feature matrix for each input sequence. For each day ahead of predictions, these features, along with sales, are used to create a correlation matrix which is then converted to a Laplacian matrix for efficient layer-wise propagation of the learning phase. Hence, training is done using the true sales data, its feature matrix and the Laplacian matrix with different days ahead of sales data as the target variable. Once trained, the same set of matrices from the validation and test dataset is used to evaluate the network's performance. The performance analysis is done by calculating normalized test RMSE as other approaches mentioned in this chapter. A classical SOTA GNN is implemented for sales forecasting, and its accuracy is compared against SBFA and other prominent approaches for the same dataset carrying out the same series of experiments. The findings are compared and described in the later part of this chapter in the result analysis section.

## **3.4 The Proposed Switching Based Forecasting Approach (SBFA)**

Advanced ML/DL approaches are combined together to form a switching based approach, termed as SBFA. A layout diagram of the approach is illustrated in Figure 3.9.

From the diagram, it can be seen that three subsets of data are given as input to train 8 different advanced forecasting approaches, along with days ahead predictions. For different days ahead of forecasting, different algorithms perform differently. Hence, based on the minimum validation error, each time prediction is taken from the particular approach that produces the minimum validation error. For a single day ahead of predictions, each approach is trained, validated and tested 10 times to reduce uncertainties due to randomness in the weights and bias of the networks. Thus, for any particular days ahead of prediction, there are 10 different test RMSEs (Root Mean Squared Errors) from 10 different predictions and the average for each of them is calculated. The proposed switching based technique is further explained by pseudo code in Algorithm 2.

---

**Algorithm 2** Proposed: SBFA

**Input:** Training\_set, Validation\_set, Testing\_set, days\_ahead\_prediction, number\_of\_iterations **Output:** Test RMSE: XGB, LGBM, KNN, RF, LSTM, RNN, GRU, CNN

**Predicted data:** XGB, LGBM, KNN, RF, LSTM, RNN, GRU, CNN and Switching\_based\_predictions

```
1: for each_day = 1,2,...,days_ahead_prediction do
2:   for i=1,2,...,10 do
3:     Train Algorithms with Arguments in parenthesis and obtain
   train_RMSE, validation_RMSE and test_RMSE.
   XGB(Training_set,Validation_set,Testing_set,each_day)
   LGBM(Training_set,Validation_set,Testing_set,each_day)
   KNN(Training_set,Validation_set,Testing_set,each_day)
   RF(Training_set,Validation_set,Testing_set,each_day)
   LSTM(Training_set,Validation_set,Testing_set,each_day)
   RNN(Training_set,Validation_set,Testing_set,each_day)
   GRU(Training_set,Validation_set,Testing_set,each_day)
   CNN(Training_set,Validation_set,Testing_set,each_day)
4:   Perform algorithm selection based on lowest validation_RMSE
5:   Obtain predictions ( $y_{\text{hat}}$ ) and test_RMSE ( $rmse_{\text{test}}$ ) from the corresponding
   selected approach
6:   append  $y_{\text{hat}}$  to  $y_{\text{hat\_array}}$ 
7:   append  $rmse_{\text{test}}$  to  $rmse_{\text{test\_array}}$ 
8: end for
9: append  $y_{\text{hat\_pred}}$  to  $\text{mean}(y_{\text{hat\_array}})$ 
10: append  $rmse_{\text{test}}$  to  $\text{mean}(rmse_{\text{test\_array}})$ 
11: end for
```

---

### 3. A SWITCHING BASED FORECASTING APPROACH (SBFA) FOR FORECASTING SALES DATA IN SUPPLY CHAINS

---

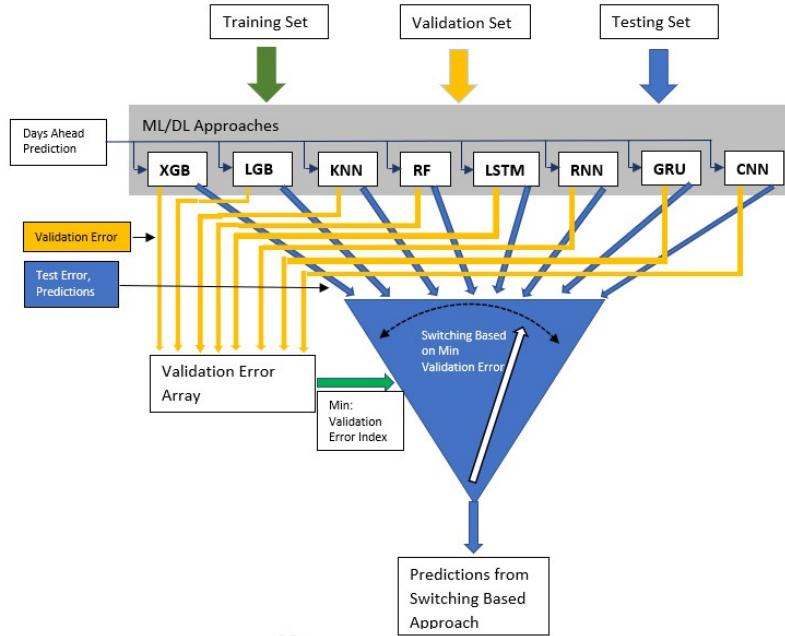


Figure 3.9: Switching Based Forecasting Approach: SBFA

Using the training part of the data, each approach is trained individually for the particular day ahead predictions. Once trained, the validation data is used to calculate the validation error and an array is created to store the validation errors. From the array, the index with the minimum validation error is found and thus that index is used to determine the predictions from the most appropriate approach. In particular, all the approaches are trained 10 different times and based on minimum validation error, 10 different test RMSEs and predictions are found. The average is calculated for each particular day ahead predictions. Since the lead time is considered as 30 days, the models are trained from 1, up to 30 days, ahead of predictions. The predictions are later used in the supply chain model for further analysis.

## 3.5 The Considered Supply Chain Model

A two-stage supply chain model is considered which consists of a retailer and a distributor (supplier). The simple model is developed based on the following assumptions: the retailer sells only one type of commodity, there is no backorder cost but rather a cost for lost sales

is incurred whenever the retailer fails to satisfy customers' needs, the supplier stores and supplies finished product to the retailer after an order is placed by the retailer. The decision of whether to place an order or not is determined by the reorder point (ROP). This ROP is calculated from the predicted demand during the lead time (LTPD) produced by the machine learning approaches. Once an order is placed, retailers' inventory is replenished after a certain amount of lead time. The question of when to place an order is dependent on the summation of the amount in delivery and the current inventory level. The model can be operated with fixed or dynamic lead time. In the context of the model, fixed lead time is defined as the finite amount of time between when an order for a certain number of units is received by the supplier and when the units are fully delivered to the retailers' inventory. This fixed lead time becomes dynamic when the delivery time varies within a given range but it is known during the placement of an order.

The developed supply chain model also calculates the predicted demand during the lead time, named as lead time predicted demand (LTPD), which is assumed to be provided by the ML forecasting algorithms. Using the variation of previous lead time and the predicted demand, the model can be operated with a certain amount of safety stock (ss). Safety stock is an extra quantity of goods stored in the inventory which prevents a business from a stock-out situation. The summation of LTPD and ss together creates the reorder point (ROP). The model also keeps track of the total amount of orders in delivery. So, whenever, ROP becomes greater than the summation of the current inventory level and orders in delivery, an order placement is issued to the supplier for replenishment. The ROP can also be varied by changing the level of safety stock. This is an added feature to make the model more realistic. Meanwhile, motivated by the work of Gonçalves et al. [2020], the safety stock and the ROP are determined by equations 3.17 and 3.19, respectively.

$$ss = Z \sqrt{(\sigma_d^2)\bar{L} + (\sigma_L^2)\bar{d}} \quad (3.17)$$

where,  $\sigma_d$ = standard deviation of predicted demand;

$\sigma_L$ = standard deviation of Lead time;

$\bar{L}$ = average lead time;

### *3. A SWITCHING BASED FORECASTING APPROACH (SBFA) FOR FORECASTING SALES DATA IN SUPPLY CHAINS*

---

$\bar{d}$ = average predicted demand;

$Z$ =number of standard deviations corresponding to service level probability.

Safety stocks have two associated dimensions: demand variability and lead time variability of suppliers. Variations in demand can be found from the ML/DL predictions but with lead time there are two different avenues: fixed lead time and dynamic lead time. When the lead time is fixed,  $\sigma_L=0$ , and hence equation 3.17 is reduced to equation 3.18. The symbols in equation 3.18 have their usual meanings.

$$ss = Z\sigma_d\sqrt{L} \quad (3.18)$$

$$ROP = LTPD + ss \quad (3.19)$$

The model has four costs associated with it: purchasing costs, ordering costs, holding costs and lost sales. The summation of these costs is considered as the total cost (TC) of the supply chain according to Equation 3.20.

The following are the assumptions regarding associated costs:

Purchasing cost (PC): The purchasing cost per unit product is known and fixed.

Ordering Cost (OC): It is assumed to be known, fixed and independent of order quantity.

Holding Cost (HC): It increases or decreases linearly with the amount of inventory in stock.

Lost Sales (LS): It is assumed to be fixed and it increases linearly with the number of items out of stock. It is relatively the highest per unit cost among all other costs as the cost of losing a single lost sale could actually be several times higher than gross profit.

$$TC = \sum(PC, OC, HC, LS) \quad (3.20)$$

The supply chain model's parameters are slightly changed to ensure proper triggering because of different sales levels in different datasets. The parameters used for the experimentation are presented in Table 3.2.

Table 3.2: Input data for the Supply Chain Model

	Kaggle Dataset	Rossmann Dataset
Ordering Cost (per order)	\$20	\$200000
Purchasing Cost (per unit)	\$2	\$2
Holding Cost (per unit per year)	\$5	\$5
Lost Sales Cost (per unit)	\$35	\$35
Retailer's Beginning Inventory (units)	\$200	\$20000000
Retailer Lot Size Q (units)	\$150	\$15000000

From Table 3.2, it can be seen that ordering cost, retailer's beginning inventory and lot size  $Q$  are changed, while purchasing cost, holding cost and lost sales cost are kept constant. For the first part of the experimentation, some traditional forecasting methodologies are used using the Kaggle dataset and hence, the parameters associated with Kaggle are used to establish the effectiveness of the supply chain model. Meanwhile, in the latter part of this chapter, a more thorough analysis is done using state-of-the-art ML/DL approaches. Thus, parameters associated with both Kaggle and Rossmann datasets are considered.

### 3.5.1 Forecasting Sales Data by Traditional Forecasting Approaches

For the initial part of the analysis, traditional forecasting techniques discussed in Section 3.3.1 are implemented to predict sales data for the Kaggle dataset, which were later applied to the designed supply chain model. As mentioned in Table 3.1, data from 1 Jan 2013 till 30 April was used to train MA, SES and N. Dist systems while the remaining Kaggle data from 1 May 2016 to 31 Dec 2017 was used to test the performances of all the traditional forecasting approaches. The accuracy of the forecasting approaches was measured by the Root Mean Square Error (RMSE) according to Equation 3.21. RMSE works better while showing larger deviations than other evaluation systems. For different days ahead of prediction, average test RMSEs for all those approaches are reported in Table 3.3.

$$RMSE = \sqrt{\left(\frac{1}{n}\right) \sum_{i=1}^n (\hat{y}_i - y_i)^2} \quad (3.21)$$

### 3. A SWITCHING BASED FORECASTING APPROACH (SBFA) FOR FORECASTING SALES DATA IN SUPPLY CHAINS

---

where,  $\hat{y}$  is the predicted value,  $y_i$  is the true value and  $n$  is the number of observations in both the above-mentioned equations.

Table 3.3: Impact of the Accuracy of Traditional Forecasting Approaches on Supply Chain Cost

		N. Dist	MA	SES	AR
	Average Test RMSE	0.2315	0.1542	0.1317	0.1430
SC Models	Safety Stock	TC (\$)			
Fixed Lead Time (30 Days)	No Safety Stock (SS)	120524.66	88653.14	<b>65178.89</b>	72851.05
	SS 95% Service Level	98641.03	83290.93	<b>64498.25</b>	69847.56
	SS 99% Service Level	90483.85	77371.53	<b>62968.46</b>	67461.99
	SS 99.9% Service Level	83256.28	77432.27	<b>60298.12</b>	67516.02
Dynamic Lead Time (27 + 3 Days)	No Safety Stock (SS)	123795.52	87145.49	<b>70096.51</b>	72225.09
	SS 95% Service Level	94552.62	77427.84	<b>56160.20</b>	60433.42
	SS 99% Service Level	84740.57	66030.52	<b>54177.99</b>	55523.84
	SS 99.9% Service Level	80006.60	64224.88	<b>52307.70</b>	53418.01

The purpose of this section is to establish that the overall cost of a supply chain model is reduced for its different variants as the accuracy of forecasting systems improves. From Table 3.3, it is evident that, among the 3 traditional approaches, SES produces the lowest RMSE of 0.1317 when compared against MA and N.Dist and that also corresponds to the lowest overall supply chain cost (highlighted in bold). As mentioned in the previous section, the designed supply chain model has a fixed lead time of 30 days and a dynamic lead time of between 27 and 30 days for the product to reach the retailer. Both can be operated with variable ss. Table 3.3 depicts that for static lead time, the supply chain cost gradually reduces as the ML accuracy gradually improves from N.Dist to SES. Moving downward, the addition of ss reduces the lost sales, which further reduces the overall supply chain cost. This is also true for dynamic lead time. Moving horizontally across the table shows the increasing forecasting accuracy of the models, resulting in a reduction in cost, while increasing ss further reduces cost by minimising lost sales.

## 3.6 Experimental Design For the Proposed SBFA

This section describes the methodology of applying the proposed SBFA for the supply chain model.

### **3.6.1 Data Collection**

Two sets of data were collected from an open-source platform for experimenting with the proposed SBFA. Daily sales data of different stores were downloaded from <https://www.kaggle.com/c/demand-forecasting-kernels-only/data>, which acts as the primary dataset. The secondary dataset was also downloaded from the same source (<https://www.kaggle.com/c/rossmann-store-sales/data>) but has fewer time steps. Both datasets were initially used in a competition to predict the sales level of different products at different stores. These are time series data and information available from 1 Jan 2013 to 31 Dec 2018 for the Kaggle dataset and 1 Jan 2013 to 31 July 2015 for the Rossmann dataset. Notably, the dataset and the day ranges are not from recent times, which is due to the absence of any recent sales data in the recent literature. For the purpose of our experimental analysis, publicly made available data was collected from the open source platform and filtered by eliminating other products, prices and store information, such that it represents sales of one particular item related to one specific store with their corresponding dates. For the initial phase of experimentation with some traditional machine learning techniques, the primary Kaggle sales dataset was used. After extracting features from the data, the dataset was split into two parts: train and test. As discussed earlier, the first 4 years of Kaggle data is used for training, while the data from 2018 is used for testing. In the latter part, where advanced machine learning techniques are used, features from both datasets, Kaggle and Rossmann, were extracted the same way as before, but the subdivision of the dataset was done differently for cross-validation purposes, as explained in Section 3.6.2. To keep consistency in experimentation, all ML/DL approaches, mentioned in this chapter, were fed with the same set of features with sales as the target variable.

#### **3.6.1.1 Feature Extraction**

Feature extraction and their proper organization are essential steps for data to be fed into ML/DL models. In order to train the ML/DL models, important features, such as year,

### 3. A SWITCHING BASED FORECASTING APPROACH (SBFA) FOR FORECASTING SALES DATA IN SUPPLY CHAINS

---

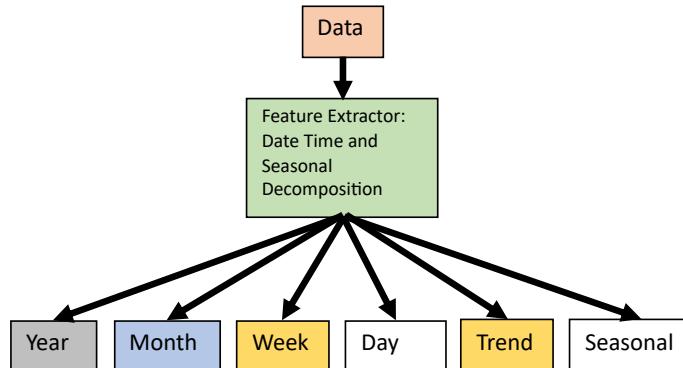


Figure 3.10: Extraction of Features From Data

month, week, day, trend and seasonality were extracted. Using Python's built-in function `to_datetime`, year, month, week, and day were acquired, while using `seasonal_decompose` from stats models, trend and seasonality of the sales data were extracted. Thus the data frame now contains seven features, as shown in Figure 3.10, with sales being the target variable.

#### 3.6.1.2 Data Structuring

A min-max scalar normalisation function [Géron, 2019] has been applied for scaling the features of the data between 0 and 1. This is done so that it can easily be handled if there are any sigmoid activation functions in the output layer of any neural network. Therefore, the normalization function was calculated based on the following formula given in equation 3.22:

$$x_{norm} = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (3.22)$$

In the next stage, the scaled data frame was structured in a sequential pattern. For multivariate predictions, sales (dependent variable) are based on other independent variables (year, month, week, day, trend and seasonality). Independent variables are called X and the dependent variables are y. From the data frame, the data is structured in a 2D array consisting of Independent and dependent variables, according to Table 3.4.

Table 3.4: Data Structuring

Independent Variables		Target Variable
Variable X1	Variable X2	y
$X1_{t-n-days\_ahead\_prediction}$	$X2_{t-n-days\_ahead\_prediction}$	$y_{t-n+days\_ahead\_prediction}$
...	...	...
...	...	...
$X1_{t-2-days\_ahead\_prediction}$	$X2_{t-2-days\_ahead\_prediction}$	$y_{t-2+days\_ahead\_prediction}$
$X1_{t-1-days\_ahead\_prediction}$	$X2_{t-1-days\_ahead\_prediction}$	$y_{t-1+days\_ahead\_prediction}$
$X1_{t-days\_ahead\_prediction}$	$X2_{t-days\_ahead\_prediction}$	$y_{t+days\_ahead\_prediction}$

where  $t$ : current time;  $n$ : number of past time steps;  $days\_ahead\_prediction$ : forecasting horizon

In this way, three different segments were selected from the data frame for training, validation and testing. For training,  $X\_train$  was formed from the independent variables while  $y\_train$  was the target column. Similarly,  $X\_val$ ,  $y\_val$  were taken for validation and finally  $X\_test$ ,  $y\_test$  for testing. If, at any time, there is no data at the beginning, then only the available time steps are considered. For the decision tree-based approaches (XGB, LGBM, KNN, RF),  $X\_train$ ,  $X\_val$  and  $X\_test$  are 2D arrays whereas, for neural networks (e.g., LSTM, RNN, GRU and CNN), these  $X\_train$ ,  $X\_val$  and  $X\_test$  are converted to suitable 3D arrays using a sliding window approach [Selvin et al., 2017]. In the experiments with NN, a window of size 10 (working days of two consecutive weeks) is used to predict one day into the future (short term) and up to 30 days ahead (long term) sales.

### 3.6.1.3 Hyperparameters

Hyperparameters are an integral part of any DL/ML-based forecasting approaches. In this chapter, eight different advanced forecasting approaches are applied to making sales predictions. Among those approaches, four are decision tree-based approaches and the rest are neural networks. The decision tree-based approaches are trained using an extensive GridsearchCV process [Buitinck et al., 2013] while the NNs are trained based on some fixed sets of parameters. The range of parameters for decision tree-based processes is selected based on the best possible default values acquired from Keras documentation.

Table 3.5 shows the list of the possible parameters used for forecasting sales data. While

### 3. A SWITCHING BASED FORECASTING APPROACH (SBFA) FOR FORECASTING SALES DATA IN SUPPLY CHAINS

---

Table 3.5: Hyperparameter Values

Algorithm	Hyperparameters	
XGB	Gridsearchcv Parameters	min_child_weight: [1, 5, 10], gamma: [0.5, 1, 1.5, 2, 5], subsample: [0.6, 0.8, 1.0], colsample_bytree: [0.6, 0.8, 1.0], max_depth: [3, 4, 5]
LGBM		num_leaves: [7, 14, 21, 28, 31, 50], learning_rate: [0.1, 0.03, 0.003], max_depth: [-1, 3, 5], n_estimators: [50, 100, 200, 500]
KNN		n_neighbours: [2,3,4], weights: ['uniform','distance']
RF		bootstrap: [True], max_depth: [80, 90, 100, 110], max_features: [2, 3], min_samples_leaf: [3, 4, 5], min_samples_split: [8, 10, 12], n_estimators: [100, 200, 300, 1000]
LSTM RNN GRU	Number of Layers: 2, Hidden Units Size: 64, Learning Rate: 0.001 (Default), Dropout: 0 (Default), Epochs: 150, Activation Function: hard_sigmoid (Layer 1), tanh (Layer 2), Optimiser: ADAM, Loss: MSE, Dense Layer: Hidden Units:1, Activation Function: Linear	
CNN	Filters: 64, kernel_size: 1, Layers: 2 Conv1D layers, MaxPooling1D layer in between Conv1D layers, Last Layers: Flatten Layer Dense Layer (Hidden Units: 1, Activation Function: Linear) Other parameters remain the same as before	

implementing the gridsearchCV technique, only the critical parameters for each model were chosen and the range is a slight deviation from the default values. It is presumed that the default parameter values are the most appropriate values. The default values are mostly collected from the documentation of the individual approaches [Documentation—XGBoost, 2022, Documentation—LGBM, 2022, Buitinck et al., 2013].

### 3.6.2 Cross Validation

Essentially, each of the two datasets is subdivided into three equal sections according to Table 3.6. Using different combinations of the two data-set, 12 experiments were conducted. A certain portion of data was used to train networks while another part was used for validation. The last portion of data was kept for testing. In this way, different combinations of segmented data were used for cross-validation while conducting experiments. Table 3.7 represents the combinations used to cross-validate the networks using the two datasets.

Table 3.6: Subsets of Dataset

	Subset A	Time-steps	Subset B	Time-steps	Subset C	Time-steps
Kaggle Dataset	1 Jan 2013 - 31 Aug 2014	608	1 Sept 2014 -30 April 2016	608	1 May 2016 - 31 Dec 2017	610
Rossmann Dataset	1 Jan 2013 - 31 Oct 2013	304	1 Nov 2013 - 31 Aug 2014	304	1 Sep 2014 - 31 July 2015	334

Table 3.7: Experiments Performed using Cross-Validation of dataset

Kaggle Dataset	Training Dataset	Validation Dataset	Testing Dataset
Expt 1	Kaggle A	Kaggle B	Kaggle C
Expt 2	Kaggle A	Kaggle C	Kaggle B
Expt 3	Kaggle B	Kaggle A	Kaggle C
Expt 4	Kaggle B	Kaggle C	Kaggle A
Expt 5	Kaggle C	Kaggle B	Kaggle A
Expt 6	Kaggle C	Kaggle A	Kaggle B
Rossmann Dataset			
Expt 1	Rossmann A	Rossmann B	Rossmann C
Expt 2	Rossmann A	Rossmann C	Rossmann B
Expt 3	Rossmann B	Rossmann A	Rossmann C
Expt 4	Rossmann B	Rossmann C	Rossmann A
Expt 5	Rossmann C	Rossmann B	Rossmann A
Expt 6	Rossmann C	Rossmann A	Rossmann B

Here, in Table 3.7, Kaggle A refers to Subset A of the Kaggle dataset containing 608-time steps. Whereas, Rossmann C refers to Subset C of the Rossmann dataset with 334-time steps. When referred to Expt 1, Kaggle A is used to train the network, Kaggle B then validates it and finally, for testing, the Kaggle C dataset was used. These subsets were shuffled to conduct different experiments for cross-validation with each of the ML/DL models.

### 3.7 Result Analyses for the Proposed SBFA

This section describes the performance of the SBFA approach in contrast to other ML/DL approaches for different datasets, evaluated in terms of RMSE, statistical analysis and training time. The predictions made by some prominent approaches, along with the proposed SBFA, were further evaluated by using the designed supply chain model, as explained in Section 3.5. The contribution of each approach while using predictions from SBFA over a long time horizon is also discussed here. Finally, it is shown that the proposed SBFA produces comparatively lower supply chain cost than the standalone approaches.

#### 3.7.1 A comparative Analysis of SBFA with existing ML/DL Approaches

Each ML/DL approach, including SBFA, was trained, validated and tested with different subsets of data as per Table 3.7 for different days ahead of predictions (up to 30 days). For any particular day ahead of the forecast, each of the models was trained 10 times with 10 different seeds. After that, for any specific day ahead predictions, average predicted values and their corresponding RMSEs were calculated. The results from the experiments on both datasets are reported in Tables 3.8 and 3.9. Fixed seed values were used to make the results reproducible.

Table 3.8: Average test RMSE of 10 Runs for 30 days ahead of prediction using Kaggle Dataset

Kaggle Dataset	Proposed (SBFA)	XGB	LGBM	KNN	RF	LSTM	RNN	GRU	CNN	TEA Lit [Jin et al., 2021]	TEA Tuned	GNN
Expt 1	0.1103	0.1475	0.1213	0.1319	0.1236	0.1202	0.1182	0.1190	0.1146	0.3358	0.1444	0.1358
Expt 2	0.1019	0.1226	0.1069	0.1202	0.1086	0.1083	0.1059	0.1082	0.1081	0.2760	0.1261	0.1208
Expt 3	0.0998	0.1268	0.1072	0.1264	0.1075	0.1065	0.1037	0.1070	0.1062	0.3345	0.1368	0.1353
Expt 4	0.0888	0.1050	0.0950	0.1117	0.0960	0.0943	0.0939	0.0962	0.0951	0.2410	0.1196	0.1143
Expt 5	0.1001	0.1259	0.1176	0.1707	0.1171	0.1049	0.1077	0.1079	0.0994	0.5077	0.1332	0.1175
Expt 6	0.1015	0.1205	0.1078	0.1279	0.1082	0.1079	0.1110	0.1094	0.1057	0.2718	0.1352	0.1206
Mean RMSE	<b>0.1004</b>	0.1247	0.1093	0.1315	0.1102	0.1070	0.1067	0.1079	<b>0.1048</b>	0.3278	0.1325	0.1241

From Table 3.8, it can be seen that for the Kaggle Dataset, the average Test RMSE for the proposed SBFA is 0.1004. Among all of the other standalone approaches, CNN was the best-performing algorithm, producing an RMSE of 0.1048, which is still higher than the proposed switching-based approach. TEA having parameter settings according to

### 3. A SWITCHING BASED FORECASTING APPROACH (SBFA) FOR FORECASTING SALES DATA IN SUPPLY CHAINS

---

Table 3.9: Average test RMSE of 10 Runs for 30 days ahead of prediction using Rossmann Dataset

Rossmann Dataset	Proposed (SBFA)	XGB	LGBM	KNN	RF	LSTM	RNN	GRU	CNN	TEA Lit [Jin et al., 2021]	TEA Tuned	GNN
Expt 1	0.1196	0.1406	0.1269	0.1849	0.1361	0.1352	0.1230	0.1274	0.1250	0.4587	0.2630	0.1692
Expt 2	0.1173	0.1399	0.1228	0.1794	0.1307	0.1329	0.1194	0.1246	0.1211	0.4493	0.2309	0.1616
Expt 3	0.1188	0.1302	0.1101	0.1784	0.1168	0.1274	0.1212	0.1227	0.1233	0.4588	0.2301	0.1918
Expt 4	0.1025	0.1102	0.1062	0.1694	0.1068	0.1099	0.1046	0.1035	0.1046	0.4244	0.2197	0.1491
Expt 5	0.1006	0.1134	0.1043	0.1759	0.1059	0.1074	0.1020	0.1035	0.1013	0.4245	0.2230	0.1552
Expt 6	0.1117	0.1326	0.1084	0.1764	0.1130	0.1217	0.1142	0.1178	0.1166	0.4493	0.2277	0.1611
Mean RMSE	<b>0.1117</b>	0.1278	<b>0.1131</b>	0.1774	0.1182	0.1224	0.1141	0.1166	0.1153	0.4442	0.2324	0.1647

the literature performed poorly (RMSE: 0.3278) while fine-tuning its hyper-parameters and layers (TEA Tuned) produced a much better result producing an RMSE of 0.1325. GNN also produced comparatively better RMSE (0.1241) than both the TEA approaches; however, the proposed SBFA stands out among all approaches for the Kaggle dataset.

Results from experiments on the Rossmann dataset are stated in Table 3.9. The proposed switching-based technique still produces the lowest test RMSE of 0.1117, with the best-performing standalone approach being LGBM producing an RMSE of 0.1131. Just like the previous dataset, the SOTA approaches, such as TEA and GNN, at their given settings from the literature, produced higher RMSE than other approaches. Among other standalone approaches, CNN is the best in the Kaggle dataset, while LGBM, from the DTB group, performs best in the Rossmann dataset. Hence, it can be concluded that there is no fixed approach that produces the best predictions for all datasets.

Next, the time for training each of the models, along with the switching-based one, is reported in Table 3.10. As mentioned previously, the DTB models are trained using an extensive grid-search technique to find the optimum parameter values, hence taking comparatively longer to produce the predictions. Although KNN is the fastest approach (since there are not many parameters involved) among the models, the accuracy of KNN is very low due to its higher test RMSE values (according to Tables 3.8 and 3.9).

NNs are trained with some fixed sets of parameters. CNN is the quickest among the NNs as it takes around 2650 seconds to train itself. TEA, with a larger learning rate of 1.0, takes a similar amount of time (209695 sec), in comparison to SBFA, to train itself while

### 3. A SWITCHING BASED FORECASTING APPROACH (SBFA) FOR FORECASTING SALES DATA IN SUPPLY CHAINS

---

Table 3.10: Training Time of Different Approaches in seconds

Kaggle Dataset	Proposed: SBFA(s)	XGB(s)	LGBM(s)	KNN(s)	RF(s)	LSTM(s)	RNN(s)	GRU(s)	CNN(s)	TEA Lit	TEA Tuned	GNN
Expt 1	228989	28205	8322	0	164860	13115	9469	11933	3398	267066	33388	7890
Expt 2	240299	25706	7894	0	164611	11728	6435	10659	3287	280236	31071	7609
Expt 3	231659	26239	8111	0	164562	9486	6210	8242	2822	286176	30741	7530
Expt 4	231666	26894	8331	0	166891	9617	5631	8433	3202	273267	35287	7433
Expt 5	240079	28000	8497	0	167757	13371	9188	12357	3127	271359	34630	7327
Expt 6	234456	25513	8143	0	165852	11773	6376	11396	2842	307263	26664	7333
Rossmann Dataset												
Expt 1	197882	23246	6325	0	146361	7977	5930	8662	2681	151412	16435	7005
Expt 2	193896	21258	5926	0	147453	7077	5077	6584	1945	132624	16966	6850
Expt 3	192101	21126	6113	0	146534	7171	4808	7227	1762	130207	16778	6987
Expt 4	193379	21130	6082	0	148209	7330	5757	6755	2163	143774	19444	6870
Expt 5	202534	22442	6544	0	149382	7889	6051	7853	2503	138740	20963	6999
Expt 6	196413	22040	6417	0	152070	7715	5181	9059	2066	134220	16917	6827
Mean	215279	24317	7225	0	157045	9521	6343	9097	2650	209695	24940	7222

tuning the parameter and lowering the learning rate to  $10^{-4}$  sped up the process. GNN takes a reasonable amount of time (7222 sec) as compared to other approaches to make predictions. The proposed SBFA approach takes more time to make predictions as it selects the best approach from training each model for each day ahead of prediction based on validation RMSE. It takes an average time of around 215279 sec to make predictions, which is basically the summation of the training time of the individual approaches, but eventually, it produces better accuracy.

This is obviously a limitation to this research work, but in the future, such substantial training time can be reduced by limiting the number of parameters used during the grid search process related to the DTB forecasting approaches. Also, eliminating slower and less accurate approaches from the SBFA system would make the system faster. Despite the slow nature of the proposed SBFA system, the suitability of the approach is more appropriate for practical situations where prediction quality is more important than time.

#### 3.7.2 Statistical Analysis

From 1 to 30 days ahead of predictions, each experiment produces 300 different RMSEs (i.e., when each model is being trained 10 times for making each of 30 specific days ahead predictions). Since there are 12 different experiments (6 for each dataset) conducted over 2 datasets, hence 3600 test RMSEs were obtained for each model for the switching based technique. The average test RMSEs related to the proposed SBFA are compared against

the RMSEs of other prominent models using a Friedman Test [Friedman, 1937]. Since TEA and GNN produced comparatively higher RMSE, these two approaches are not included in the statistical tests and only the best nine approaches are taken into consideration. The results of the mean rank of the best nine approaches are reported in Table 3.11.

Table 3.11: Friedman Test: Mean Rank

Algorithms	Mean Rank: Combined dataset	Position (Combined Dataset)
Proposed: SBFA	<b>3.17</b>	1
XGB	7.66	8
LGBM	<b>3.42</b>	2
KNN	8.41	9
RF	4.48	5
LSTM	4.81	7
RNN	4.29	4
GRU	4.58	6
CNN	<b>4.18</b>	3

From Table 3.11, it can be seen that SBFA leads the rank by having the lowest mean rank of 3.17, while LGBM and CNN acquire the 2nd and 3rd positions respectively with mean rank scores of 3.42 and 4.18. Since the performance measure is RMSE, the lowest mean rank represents the least error (statistically). In the next phase, the three best-ranked models (SBFA, LGBM, and CNN) were chosen to carry out Wilcoxon Signed rank test to verify if there is the statistical significance in the score. The result of the Wilcoxon Signed rank test using RMSE is reported in Table 3.12. Using a 5% significance level, one of three signs (+, – and  $\approx$ ) is assigned, where the “+” sign means the first algorithm is significantly better than the second one, “–” means that it is significantly worse and “ $\approx$ ” represents that there is no significant difference between these approaches. Wilcoxon Signed Rank test, conducted between SBFA and other standalone approaches namely LGBM and CNN, produces p-values of 0.00 which is lower than 0.05 in every situation. Thus, the statistical test reveals that the proposed SBFA is performing statistically consistently against its two nearest competitive ML/DL approaches when RMSE is the performance measure.

### *3. A SWITCHING BASED FORECASTING APPROACH (SBFA) FOR FORECASTING SALES DATA IN SUPPLY CHAINS*

---

Table 3.12: Results of the Wilcoxon test using RMSE-testing

Algorithms	Better	Similar	Worst	p-value	Decision
Proposed: SBFA vs LGBM	1449	962	1189	0.00	+
Proposed: SBFA vs CNN	2015	674	911	0.00	+
LGBM vs CNN	2007	0	1593	0.006	+

#### **3.7.3 Application of SBFA to the Designed SC Model**

The predictions made by the Proposed SBFA, alongside predictions from some traditional approaches (namely MA and SES) and the two other best-performing individual approaches (CNN and LGBM), in terms of RMSE, were fed into the supply chain system with a static and dynamic lead time for further evaluation. The SC parameters for the Kaggle dataset remain the same, however, a few parameters are changed for the Rossmann dataset. The Rossmann data ranges from a few thousand to some millions and hence some changes are required for triggering order placement at the right time. The parameter changes were already described in Table 3.2. It is to be noted that N.Dist is not used for any further analysis as demand/sales data are not always normally distributed [Kozłowski et al., 2018]. It was initially used to establish the fact that the safety stock level of the SC model can compensate for the variation of demand during replenishment lead time [Brunaud et al., 2019].

##### **3.7.3.1 Fixed Lead time with Variable Safety Stock**

For the initial analysis, the supply chain model assumes only a fixed lead time of 30 days with different levels of safety stock applied on both the Kaggle and Rossmann datasets. The results of using predictions from SBFA, LGBM, CNN and the three classical approaches of MA, SES and AR on the SC model, are presented in Tables 3.13 and 3.14.

*3. A SWITCHING BASED FORECASTING APPROACH (SBFA) FOR FORECASTING SALES DATA IN SUPPLY CHAINS*

---

Table 3.13: SC Cost (Fixed Lead time with variable safety stock) associated with different advanced forecasting approaches using Kaggle Dataset

		SC Cost using Kaggle Dataset (\$)					
		MA	SES	AR	LGBM	CNN	SBFA
Expt 1	No SS	85633.24	65170.28	72001.24	85960.45	54474.89	65289.80
	95% SS	78834.26	64505.63	69906.13	75575.02	52245.73	57103.14
	99% SS	77405.39	62961.53	67521.89	70381.91	51664.15	54167.81
	99.9% SS	71001.82	62306.34	62595.44	66885.86	50517.30	51769.04
Expt 2	No SS	64356.35	61305.41	57274.14	56240.53	53995.41	61206.13
	95% SS	58633.01	58993.41	51894.09	49007.24	47347.84	50116.78
	99% SS	58315.51	54083.31	50506.06	47174.71	45295.91	48461.83
	99.9% SS	57963.26	53461.63	49951.90	44164.99	42504.33	45237.46
Expt 3	No SS	78842.58	65178.89	72017.57	67291.22	62887.10	64246.00
	95% SS	75890.07	64498.25	67484.42	56762.07	56538.41	52280.36
	99% SS	70989.11	62968.46	67512.17	55222.15	53400.53	51738.11
	99.9% SS	71015.65	60298.12	63440.95	51902.58	52471.34	50384.47
Expt 4	No SS	41257.32	49294.54	43049.65	36395.93	44128.81	36359.12
	95% SS	37383.81	45302.28	37506.00	29626.56	38058.29	29622.17
	99% SS	37420.18	39466.18	37552.42	29676.43	35287.57	28301.89
	99.9% SS	36586.91	39475.99	36398.15	28708.73	32039.48	28369.62
Expt 5	No SS	33721.32	49288.37	37583.97	27808.02	35280.80	28000.61
	95% SS	29999.60	48002.42	32102.12	27944.72	28914.39	27846.90
	99% SS	29885.82	44372.09	31557.44	28003.67	28754.32	27915.88
	99.9% SS	29923.44	43547.42	31624.76	28052.58	28477.47	27972.32
Expt 6	No SS	48644.86	59024.06	45017.80	42428.00	63801.44	62006.66
	95% SS	45771.89	58956.37	43746.23	39379.93	53413.53	52252.02
	99% SS	43417.35	54037.80	43393.85	39409.54	50177.78	47326.83
	99.9% SS	43460.17	53446.83	42742.75	38432.16	46896.61	46892.41
	Avg	54431.37	54997.73	50599.21	46768.13	46190.56	<b>45619.47</b>
Rank (Lowest Cost)		5	6	4	3	2	1

Table 3.14: SC Cost (Fixed Lead time with variable safety stock) associated with different advanced forecasting approaches using Rossmann Dataset

		SC Cost using Rossmann Dataset (\$)					
		MA	SES	AR	LGBM	CNN	SBFA
Expt 1	No SS	12162635079	12379442343	12170614469	12663437496	12471780352	11322212818
	95% SS	12165907970	12163465917	11179593123	9799456363	9810481984	9657921727
	99% SS	12167737560	12164142833	11184577615	9667693658	9465067399	9358807278
	99.9% SS	12168852552	12164294104	11187622040	9399790586	9252705213	9263475894
Expt 2	No SS	10109437037	10369954801	10650843243	10831506785	9113115758	8920509299
	95% SS	10058190600	10369954801	9956874084	8480546321	8397238093	8403652566
	99% SS	10058173568	10369954801	9666499638	8398965066	8406926714	8415068162
	99.9% SS	10016096490	10369954801	9248365377	8410133951	8443439872	8451087899
Expt 3	No SS	10902644311	10898630584	10685905021	10899208643	13359043805	13127384644
	95% SS	10689012902	10899052608	10691759713	8844383708	9672939737	9672319802
	99% SS	10690637423	10899052608	10586559090	8758142940	9326022847	9330392648
	99.9% SS	10691520888	10901747219	10442862950	8797888449	9234284277	9019686454
Expt 4	No SS	8134888988	8037418076	8280491285	7823029212	7747851318	7844590304
	95% SS	8135900780	8037418076	8273450983	7721439294	7731643376	7722714804
	99% SS	8038384289	8037673178	7933676502	7758173988	7770418886	7762000519
	99.9% SS	8000777152	8037673178	7936227522	7825363784	7840414804	7828935213
Expt 5	No SS	7804458134	7670791216	7934952012	7811453529	8545638979	7842200507
	95% SS	7806410816	7584763784	7940232728	7725265825	7670576029	7722969906
	99% SS	7740482411	7585273988	7793918621	7763531131	7738786233	7761235213
	99.9% SS	7740482411	7585529090	7757192355	7828935213	7810312764	7828169906
Expt 6	No SS	8929888645	8929148513	9042787745	8428083936	9649754685	9402642717
	95% SS	8930875487	8929374995	8907652458	8355756320	8323190530	8330098425
	99% SS	8930875487	8929374995	8908886011	8424443162	8392370793	8399772109
	99.9% SS	8931615618	8929374995	8745629458	8490909609	8429870793	8438258951
Avg		9625245275	9676810896	9462798918	<b>8787814124</b>	8941828135	8826087823
Rank (Lowest Cost)		5	6	4	1	3	2

### 3. A SWITCHING BASED FORECASTING APPROACH (SBFA) FOR FORECASTING SALES DATA IN SUPPLY CHAINS

---

After conducting 6 experiments over 2 datasets, it can be seen that the proposed SBFA produces the lowest average supply chain cost among all approaches on Kaggle Sales Dataset. Meanwhile, for the Rossmann Dataset, the SBFA is the second to minimise the supply chain cost. Therefore, it is evident that better forecasting accuracy can directly help to reduce the overall supply chain costs. The percentage differences in cost among the proposed SBFA, CNN and LGBM (best two standalone approaches) are presented in Table 3.15 and calculated using Equation 3.23.

Table 3.15: Differences of SC Cost in terms of Percentage

Approaches	Kaggle Dataset	Rossmann Dataset	Average	SC Cost Status
Proposed: SBFA vs CNN	-1.24%	-1.29%	<b>-1.27%</b>	decrease
Proposed: SBFA vs LGBM	-2.48%	0.44%	<b>-1.02%</b>	decrease

$$\%_{diff} = \frac{SC_{SBFA} - SC_{Estbd}}{SC_{Estbd}} \times 100 \quad (3.23)$$

where,  $\%_{diff}$ : Percentage difference between SC costs produced by different approaches;

$SC_{SBFA}$ : Average SC Cost from Proposed: SBFA;

$SC_{Estbd}$ : Average SC Cost from Established Approach (CNN/LGBM)

From Table 3.15, it can be seen that the predictions from the proposed SBFA can reduce 1.27% SC cost when compared against CNN and when compared against LGBM the cost can be reduced by 1.02%. Thus a more accurate prediction system such as SBFA can definitely help reduce business expenses that operate on a static lead time policy. Furthermore, with increasing safety stock level, supply chain cost related to static lead time for different forecasting systems gets reduced. This fact is demonstrated in Figure 3.11 where cost associated with the Rossmann Dataset related to Expt 1 is considered. The figure depicts that the supply chain cost related to SBFA is significantly reduced when compared to MA, SES and AR (Traditional approaches) and marginally decreased when compared to CNN and LGBM.

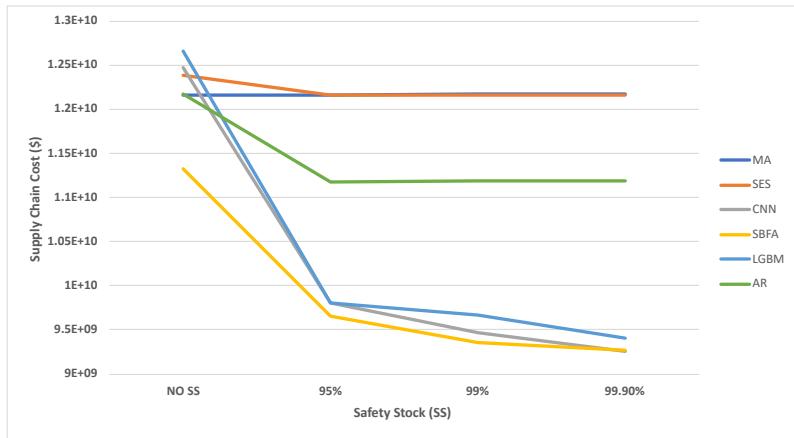


Figure 3.11: Impact of Forecasting Approaches on Supply Chain Cost with Static Lead Time and Different Levels of Safety Stock Based on Expt 1 Rossmann dataset

From Figure 3.11, it can be seen that the supply chain costs with varying levels of safety stock associated with CNN, LGBM and SBFA, are significantly lower than that of MA, SES and AR. When the lead time is fixed, the associated standard deviation ( $\sigma_L$ ) becomes zero. Predictions from MA, SES and AR do not have enough variation (hence less  $\sigma_d$ ) to produce a significant impact on ss even with a higher z-value. These two factors produce lower ss which eventually lowers ROP by creating higher lost sales for MA, SES and AR. On the other hand, SBFA, proposed in this chapter, has the ability to capture enough variation (comparatively higher  $\sigma_d$ ) in the test data, which corresponds to a higher ROP, resulting in a lower supply chain cost than other approaches.

### 3.7.3.2 Dynamic Lead time with Variable Safety Stock

The impact of SBFA is further explored on business models that operate on dynamic lead time policy. For this analysis, it is assumed that replenishment of the retailer's inventory takes between 27 and 30 days. Hence, the lead time is dynamic as it ranges from 27 days plus an additional 3 days of delay for replenishment. The same set of six experiments with cross-validation was carried out with the Kaggle dataset on a supply chain model that operates with the dynamic lead time policy and the costs associated with different forecasting approaches are presented in Table 3.16.

### 3. A SWITCHING BASED FORECASTING APPROACH (SBFA) FOR FORECASTING SALES DATA IN SUPPLY CHAINS

---

Table 3.16: SC Cost (Dynamic Lead time with variable safety stock) associated with different advanced forecasting models using Kaggle Dataset

		SC Cost using Kaggle Dataset (\$)					
		MA	SES	AR	LGBM	CNN	SBFA
Expt 1	No SS	85583.41	70087.37	70932.37	91699.52	59005.27	65457.57
	95% SS	75854.89	56167.94	60434.25	70395.51	48517.09	52370.62
	99% SS	66058.86	54174.30	54562.20	62227.63	48614.01	50690.20
	99.9% SS	63328.01	52294.39	53435.92	55705.22	46791.04	48459.09
Expt 2	No SS	63531.15	61983.41	60187.84	63067.31	54876.79	59883.96
	95% SS	53773.63	50215.23	47735.02	47200.53	42377.94	47190.90
	99% SS	49702.36	48853.03	44025.90	43829.05	39561.94	43179.58
	99.9% SS	48997.56	43966.01	44479.56	40511.76	37579.51	39547.80
Expt 3	No SS	85640.21	70096.51	70922.89	71707.97	64296.17	69163.28
	95% SS	66044.61	56160.20	60449.52	55711.98	50909.76	52400.10
	99% SS	64237.81	54177.99	54534.26	47773.13	49267.21	49130.88
	99.9% SS	60711.32	52308.93	53457.72	47004.20	49373.11	46207.93
Expt 4	No SS	40155.00	51186.33	42833.88	40022.55	49880.46	37994.77
	95% SS	32781.24	40428.64	31887.17	30307.15	33127.18	28256.23
	99% SS	30480.13	36960.38	31963.68	27545.77	32053.57	27548.28
	99.9% SS	28822.06	32070.21	28628.62	27642.34	28660.99	27286.82
Expt 5	No SS	30480.13	49818.76	36512.80	27582.31	36424.78	30257.08
	95% SS	28906.09	40416.10	29053.24	26723.34	28674.19	26620.28
	99% SS	29005.17	39386.19	29047.70	26826.19	28593.73	26623.01
	99.9% SS	28031.14	35894.46	28090.48	27267.84	26648.09	26723.34
Expt 6	No SS	48953.01	61952.59	48923.45	42281.74	68339.65	68324.90
	95% SS	41414.53	50206.60	39218.00	38370.37	51041.01	48719.18
	99% SS	40331.27	48823.08	38322.58	37946.07	45302.43	42666.62
	99.9% SS	38396.06	43964.78	36689.82	36478.88	40431.09	40589.63
	Avg	50050.82	50066.39	45680.37	45242.85	44181.13	<b>43970.50</b>
	Rank (Lowest Cost)	5	6	4	3	2	<b>1</b>

From Table 3.16, it can be seen that SBFA produces the lowest average cost of \$45619.47 on the Kaggle dataset when compared to CNN which is \$46190.56. Other approaches, such as LGBM, MA, SES and AR produced comparatively higher overall supply chain cost. The impact of SBFA on dynamic lead time supply chain model is further investigated considering the Rossmann dataset. The assumptions regarding the model are kept the same with parameters slightly changed for the Rossmann dataset according to Table 3.2. The results of the impact of SBFA and other forecasting approaches on supply chain model operating on dynamic lead time using the Rossmann dataset are presented in Table 3.17.

It can be observed from Table 3.17 that LGBM produces the lowest average cost of \$8787814124 while SC cost produced by SBFA is slightly higher (\$8826087823), occupying a second position in terms of lowest cost. The other approaches such as MA, SES,

*3. A SWITCHING BASED FORECASTING APPROACH (SBFA) FOR FORECASTING SALES DATA IN SUPPLY CHAINS*

---

Table 3.17: SC Cost (Dynamic Lead time with variable safety stock) associated with different advanced forecasting models using Rossmann Dataset

		SC Cost using Rossmann Dataset (\$)					
		MA	SES	AR	LGBM	CNN	SBFA
Expt 1	No SS	13121319460	13148989241	13082359315	14636503993	13616919003	12628019917
	95% SS	10664985621	10666285636	10958778579	10225160453	9306308303	9316862195
	99% SS	10180148276	10180185803	9977519225	9330923755	9345124537	9355864279
	99.9% SS	9775651953	9774827215	9665727434	8903129235	8913009474	8929015639
Expt 2	No SS	10932310884	10932824988	11137523421	11626930307	10141831405	10149134256
	95% SS	9168141995	9164851871	9173233953	8367949165	7973044617	7979459091
	99% SS	8680718338	8681892077	8751676593	8298851712	7965386338	7970320549
	99.9% SS	8372601851	8453357285	8299343530	7967606733	7949052699	7984927041
Expt 3	No SS	11644339312	11640843812	11904569250	12167900772	14605383926	14606508298
	95% SS	10182242015	10183505890	9643935336	8333845601	10212992514	10215866728
	99% SS	9733861147	9275864210	9282289177	8376959374	9017304224	8828906306
	99.9% SS	9326652941	9327555492	9172406271	8302953964	8899824022	8738971504
Expt 4	No SS	8511582082	8114137725	7931332812	8305951972	7809025060	8306377332
	95% SS	7335619491	7183810542	7418335554	7217382042	7224269797	7218147348
	99% SS	7149937144	7149426940	7283237144	7256667757	7270443267	7261259593
	99.9% SS	7168559593	7168814695	7331549389	7324112655	7337633063	7328704491
Expt 5	No SS	7887250840	7887486231	7936321436	7812986334	9054263563	7973143409
	95% SS	7145600410	7144324899	7323438960	7222739185	7195953471	7221208573
	99% SS	7161926940	7163202450	7293696328	7261514695	7230392246	7258708573
	99.9% SS	7216616736	7216106532	7339967757	7332531022	7311867757	7326918777
Expt 6	No SS	9656727543	9656480832	9685503319	9460096787	11124123090	10634098420
	95% SS	8371403112	8372636665.08	8265936293	7849280331	7960709091	7886090943
	99% SS	8250888554	8250395133.11	8097108837	7918213883	7889595462	7897736910
	99.9% SS	7981186065	7982913038.22	7988093959	7995042173	7929562567	7939430988
		Avg	8984178013	8946696634	8955995161	<b>8728968079</b>	8886834146
		Rank (Lowest Cost)	6	4	5	1	3
							2

AR and CNN produced even more, higher cost as per the table. Combining the average cost from the best three forecasting approaches (SBFA, LGBM and CNN from Tables 3.16 and 3.17) on the dynamic lead time, a percentage difference of supply chain cost is again calculated (using Equation 3.23) to demonstrate the overall effectiveness of SBFA and the results are presented in Table 3.18.

Table 3.18: Differences of Dynamic Lead Time SC Cost in terms of Percentage

Approaches	Kaggle Dataset	Rossmann Dataset	Average	SC Cost Status
Proposed: SBFA vs CNN	-0.48%	-1.09%	<b>-0.78%</b>	decrease
Proposed: SBFA vs LGBM	-2.81%	0.70%	<b>-1.06%</b>	decrease

From Table 3.18, it can be observed that the proposed SBFA produces consistently the lowest overall average supply chain cost than CNN and LGBM when dynamic lead time is used. While comparing SBFA with CNN, SBFA produces a 0.78% lower average cost while 1.06% lower cost is produced than the LGBM. Again, the implications of ss on the

### 3. A SWITCHING BASED FORECASTING APPROACH (SBFA) FOR FORECASTING SALES DATA IN SUPPLY CHAINS

---

supply chain model operating on dynamic lead time policy is demonstrated in Figure 3.12.

The figure is constructed based on Rossmann data taken from Expt 1 of Table 3.17.

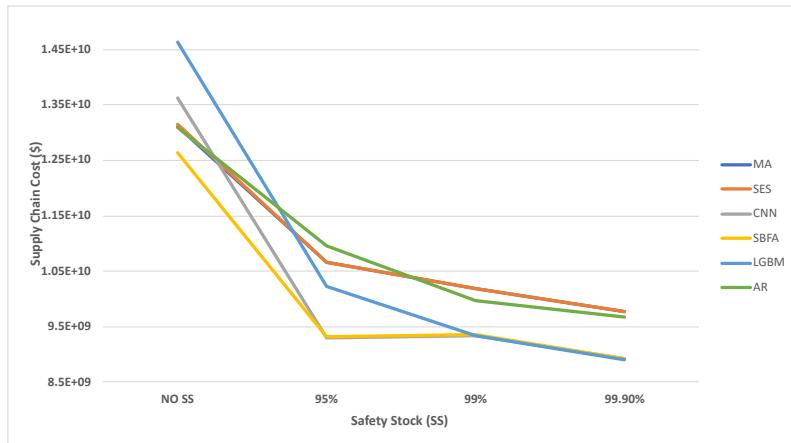


Figure 3.12: Impact of Forecasting Approaches on Supply Chain Cost with Dynamic Lead Time and Different Levels of Safety Stock Based on Expt 1 Rossmann dataset

While dealing with dynamic lead time, the standard deviation related to lead time ( $\sigma_L$ ) is more pronounced and this eventually increases the overall ss level with a higher ROP. Higher ss reduces out-of-stock situations which eventually reduces the overall supply chain cost. From Figure 3.12, it can be seen that classical forecasting approaches, such as MA, SES and AR, produces lower cost due to higher  $\sigma_L$ , which is related to dynamic lead time. The cost associated with SBFA is lower than other approaches for lower safety stock levels and it coincides with LGBM when the safety stock level reaches around 99.9%. It is worth noting that business experts often consider safety stock as a burden for their business and such a high level of safety stock is hardly maintained by business management as it incurs a significant cost to their expenditure. So within a reasonable range, such as around 95% - 99%, the cost produced by SBFA is consistently lower than other other forecasting approaches for the supply chain models operating with different levels of ss and using dynamic lead time for replenishment of their inventories.

### 3.7.4 Behavioural/Sensitivity Analysis

After comparing SBFA with other approaches, the sensitivity of the SC model is tested by changing its static lead time parameter. For this part of analysis, the model's lead time is considered to be fixed, but it is varied at 5 day intervals from 5 to 25 days. Varying the lead time of the finished product to reach the retailer from the supplier provides an opportunity to investigate the sensitivity of the SC model based on the forecasted data acquired from the SBFA technique. This is done to demonstrate the practicality of the SC model in the context of real life scenarios. The results are also compared against the traditional forecasting approaches of MA, SES and AR. For both the Kaggle and Rossmann datasets, subsets that are related to Experiment 1 are chosen and their corresponding predictions using MA, SES, AR and SBFA are considered for sensitivity analysis. The average SC costs (considering all levels of safety stock) for both datasets are calculated and presented in Table 3.19 .

Table 3.19: Average Costs for Different Entities of Supply Chain Model

	Kaggle Dataset				Rossmann Dataset			
	MA	SES	AR	SBFA	MA	SES	AR	SBFA
SC Cost (LT fixed 5 Day)	47582.02	34234.25	42338.70	<b>30278.18</b>	10082866743	10159061986	8181509984	<b>4928651142</b>
SC Cost (LT fixed 10 Day)	57925.43	39489.39	48901.15	<b>32009.96</b>	9282144102	9369104425	8799296979	<b>6054639577</b>
SC Cost (LT fixed 15 Day)	63072.36	44679.56	51777.10	<b>38516.15</b>	9670572777	9671256876	9250935836	<b>7110259359</b>
SC Cost (LT fixed 20 Day)	68742.51	51212.01	55609.57	<b>44487.62</b>	10575548708	10325109645	10160887350	<b>8027595414</b>
SC Cost (LT fixed 25 Day)	74164.13	56617.30	62539.94	<b>51422.76</b>	11729301991	12054739605	10490639987	<b>8984852955</b>

Each of these costs are also presented using graphical plots in Figure 3.13 and 3.14.

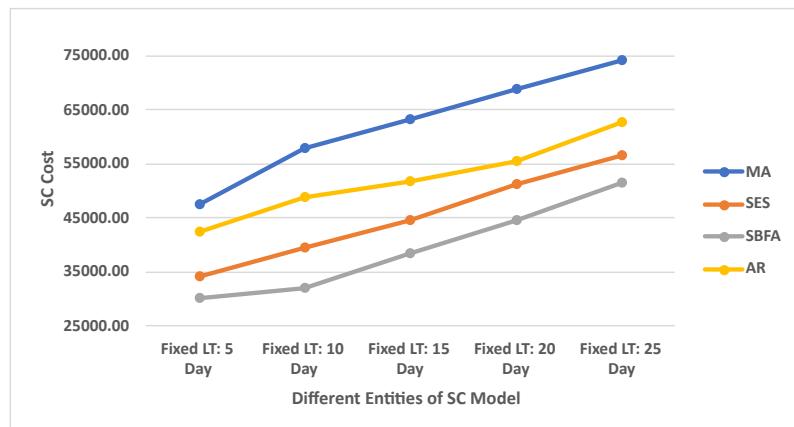


Figure 3.13: Average Cost for Different Entities of Supply Chain on the Kaggle Dataset

### 3. A SWITCHING BASED FORECASTING APPROACH (SBFA) FOR FORECASTING SALES DATA IN SUPPLY CHAINS

---

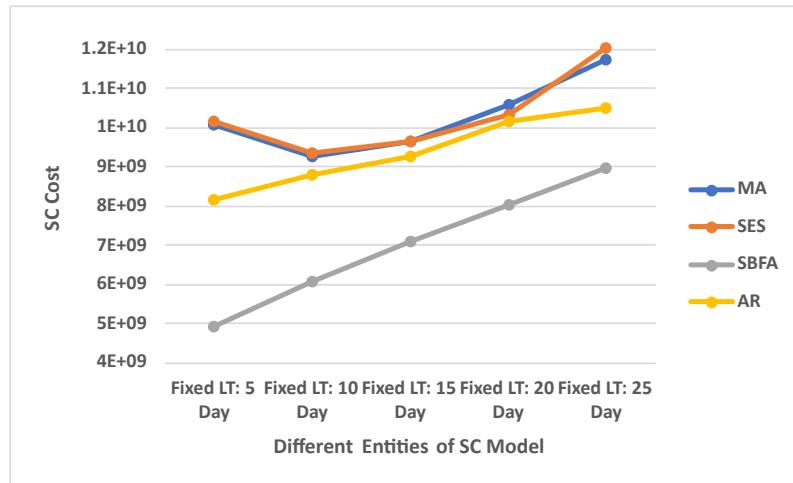


Figure 3.14: Average Cost for Different Entities of Supply Chain on the Rossmann Dataset

For the Kaggle dataset, the cost produced through MA, SES and AR gradually increases as the lead time (LT) increases from 5 to 25 days at 5 day intervals. However, for the Rossmann dataset, the cost from the same traditional forecasting approaches decreases slightly for a lead time of 10 days and then starts to increase again. This is due to the huge fluctuation in true data, while the predicted data does not have as much variations as that of real data. This leads to lower standard deviations  $\sigma_d$  which ultimately lowers ss and subsequently reduces ROP. For shorter lead time, ss is more pronounced as less predicted time steps are taken into consideration, which keeps ROP at an optimum level, but as the lead time becomes longer, the ss level gets lower due to lower  $\sigma_d$ . A lower ss means that the retailer does not have enough stock to satisfy customer demand, which increases the lost sales cost. Thus, a slight decrease in cost for shorter lead time is observed for the traditional approaches for the Rossmann dataset. While using predictions from SBFA, the fluctuations are more closely captured for both datasets. This results in a similar increasing trend of SC Cost. Overall, the cost produced from SBFA predictions is always lower than that of MA, SES and AR (presented in bold font).

### **3.8 Theoretical Derivation**

Three theories can be derived from the findings of this research work: (i) the selection process among different approaches can be made while considering the forecasting horizon; (ii) the accuracy of the combined SBFA and other approaches in regards to their training time can also be considered as a trade-off by the practitioners which might influence their decision-making process; and (iii) the supply chain cost minimisation for different forecasting system can be considered concerning their training time that can also aid managers and practitioners to take decisions and act accordingly.

#### **3.8.1 Selection of Different Approaches on the Basis of Prediction Horizon**

For different forecasting horizons, different approaches give the best results. Approaches from neural networks, such as LSTM, RNN, GRU and CNN, perform better for short-term predictions, whereas DTB approaches are stronger for long-term predictions. The contributions to the switching approach made by each sub-approach for different days ahead of predictions in terms of percentage are shown in Table 3.20. The proposed SBFA makes the selection from all the best possible approaches to make its overall predictions. For example, while making 5 day ahead prediction, out of 120 instances on both datasets, the switching-based system selects LGBM 25% of the time, CNN 25.83% and RNN 24.17% of the time. All these choices are based on the lowest validation error. Apart from the previous three major contributing approaches, other approaches such as LSTM, GRU and RF make minor contributions to SBFA's results.

For short-term analysis, RNN leads the prediction with 82.5% for 1-day ahead and 73.33% for making 2-day ahead predictions. While making 3-day ahead predictions, LSTM, RNN, GRU and CNN have significant contributions. From 4-day ahead predictions, approaches such as LGBM and RF start making contributions as NN loses accuracy. As the prediction horizon increases, contributions of LGBM and RF also increase. It can be noted that

### 3. A SWITCHING BASED FORECASTING APPROACH (SBFA) FOR FORECASTING SALES DATA IN SUPPLY CHAINS

---

Table 3.20: Contribution of individual approaches within the proposed SBFA for different days ahead prediction on Combined Dataset

Days ahead Prediction	XGB(%)	LGBM(%)	KNN(%)	RF(%)	LSTM(%)	RNN(%)	GRU(%)	CNN(%)
1	0.00	0.00	0.00	0.00	0.00	82.50	0.00	17.50
2	0.00	0.00	0.00	0.00	5.00	73.33	5.83	15.83
3	0.00	0.00	0.00	0.00	15.00	28.33	33.33	23.33
4	0.00	17.50	0.00	2.50	9.17	23.33	19.17	28.33
5	0.00	25.00	0.00	0.83	7.50	24.17	16.67	25.83
6	0.00	10.00	0.00	25.83	15.83	19.17	17.50	11.67
7	0.00	20.83	0.00	20.83	15.00	17.50	15.83	10.00
8	0.00	13.33	0.00	15.83	7.50	25.83	11.67	25.83
9	0.00	19.17	0.00	7.50	4.17	30.83	8.33	30.00
10	0.00	25.83	0.00	5.83	2.50	30.00	10.00	25.83
11	0.00	21.67	0.00	6.67	7.50	18.33	27.50	18.33
12	0.00	38.33	0.00	2.50	7.50	19.17	14.17	18.33
13	0.00	44.17	0.00	10.83	12.50	13.33	13.33	5.83
14	0.00	38.33	8.33	20.00	7.50	4.17	11.67	10.00
15	0.00	43.33	8.33	19.17	5.83	9.17	6.67	7.50
16	0.00	39.17	0.00	6.67	0.00	28.33	3.33	22.50
17	0.00	26.67	0.00	9.17	4.17	30.00	5.83	24.17
18	0.00	27.50	0.00	10.83	2.50	32.50	6.67	20.00
19	0.00	27.50	0.00	8.33	9.17	20.00	23.33	11.67
20	0.00	31.67	0.00	8.33	15.00	21.67	12.50	10.83
21	0.83	43.33	0.00	18.33	10.00	13.33	5.00	9.17
22	0.00	33.33	0.00	12.50	3.33	21.67	7.50	21.67
23	0.00	40.83	0.00	6.67	5.00	20.00	5.83	21.67
24	0.00	35.83	0.00	0.00	4.17	32.50	5.00	22.50
25	0.00	36.67	0.00	2.50	5.83	26.67	5.00	23.33
26	0.00	43.33	0.00	0.83	6.67	19.17	10.83	19.17
27	0.00	25.00	0.00	6.67	16.67	17.50	15.00	19.17
28	0.00	18.33	0.00	16.67	17.50	18.33	15.00	14.17
29	0.00	25.83	0.00	4.17	5.83	36.67	5.83	21.67
30	0.00	39.17	0.00	7.50	0.00	30.00	4.17	19.17

approaches such as XGB and KNN have the least contributions over other approaches. In the end, predicting 30 days ahead of time, LGBM has 39.17% contribution, which is the highest among all other approaches. The proposed SBFA technique selects the best after searching all the approaches based on the lowest validation error. The same result is also represented in a graphical form in Figure 3.15.

It can be seen from Figure 3.15 that given the current parameter settings of the approaches used in the proposed SBFA, RNN is good for short-term forecasting, while other approaches make similar contributions over longer horizons. The contribution made by CNN is slightly less pronounced but is consistent throughout the forecasting period. Hence, SBFA is an approach that can be used when more accurate forecasting is required, as it makes the best selection among different approaches based on validation data.

Further analysis of Figure 3.15 shows that when the forecasting horizon is short, such as 1 day or 2 day ahead, NNs seem to dominate the selection process. RNN has signifi-

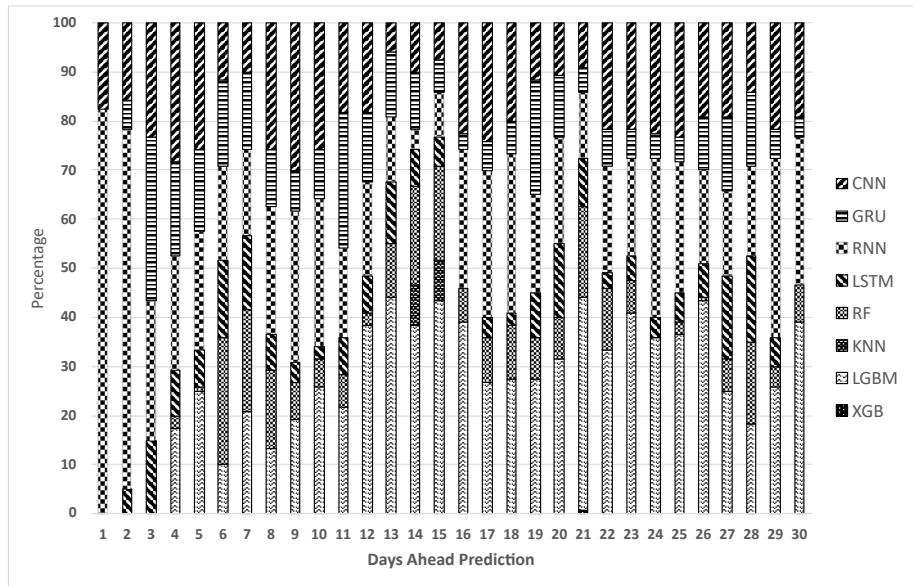


Figure 3.15: Percentage Contribution of Different Approaches for Proposed Switching Based Technique on Combined Dataset

cant influence, followed by its successors such as LSTM and GRU. This seems to be the usual case since, according to Devooght and Bersini [2017], RNNs perform well for short-term prediction, especially with data at its usual state (without any augmentation). For medium-term prediction, e.g. 12 to 15 days ahead prediction, the tree-based structures started to lead their way in the selection process [Yin et al., 2021]. Especially, LGBM plays a critical role in predicting intermediate days in the future because it can provide better judgements with a high dimensional feature matrix in data. However, at a certain stage, it does not matter whichever network the ML model chooses as it can be seen that from 16 days ahead to 20 days ahead predictions, all of them are almost in the equivalent stage. It is a good insight for the manager; if someone plans to do 20 days ahead prediction, it does not really matter if it is RNN, CNN, or tree-based architecture. And if it is too long or too far ahead predictions, then again, possibly, RNN or GRU is a good approach down the path.

### *3. A SWITCHING BASED FORECASTING APPROACH (SBFA) FOR FORECASTING SALES DATA IN SUPPLY CHAINS*

#### **3.8.2 Selection of Different Approaches on the Basis of Accuracy and Time**

The average test RMSE of SBFA and each of the eight individual methods included in the proposed system is analysed with respect to their training time for both datasets. For Kaggle datasets, average RMSE and training time from Tables 3.8 and 3.10 are combined and represented in Figure 3.16 while the for the other Rossmann dataset, the same is considered from Table 3.9 and 3.10 and depicted in Figure 3.17 for further analysis.



Figure 3.16: Performance Improvement vs Time: Kaggle dataset

It can be seen from Figures 3.16 and 3.17 that an ensemble approach such as SBFA consistently produces more accurate forecasting considering a larger amount of training time irrespective of the type of data. Thus, practitioners can select SBFA if they are looking for better forecasting accuracy considering the larger amount of time. However, they also have the freedom of choosing other approaches that consumes less time but at the cost of lower accuracy. Depending on the context of their situations, the practitioners can now decide which approach to select and whether it is worth waiting that long for accurate predictions.

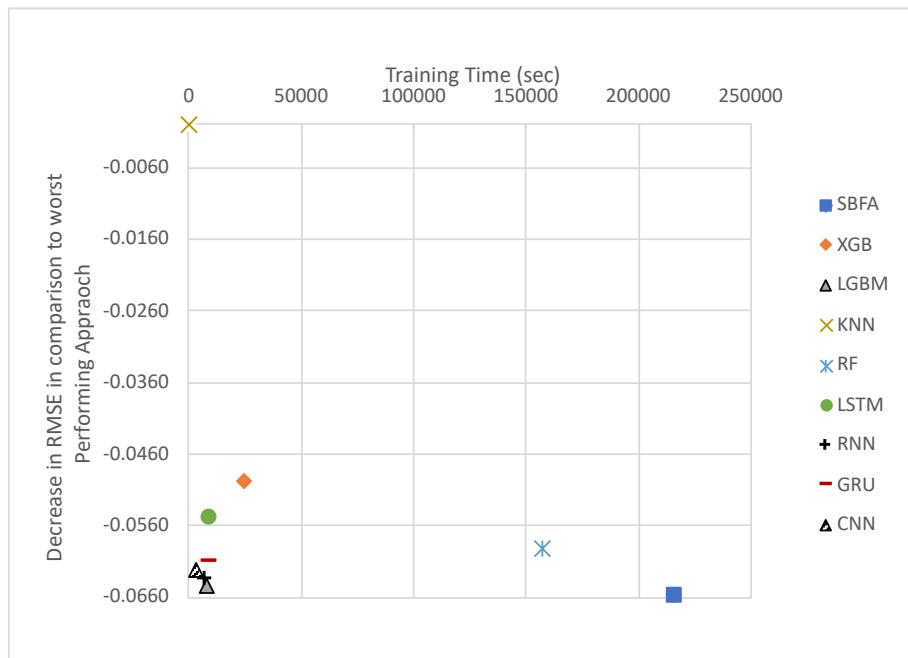


Figure 3.17: Performance Improvement vs Time: Rossmann dataset

### 3.8.3 Selection of Different Approaches based on Supply Chain Cost and Time

Selection of SBFA or any other approaches can also be done considering the cost involved in the supply chain system. The average cost of the supply chain system for the Kaggle dataset with various forecasting approaches (already reported in Table 3.13) is analysed with respect to the training time for each of the corresponding approach (reported in Table 3.10). The results are combined and represented in a graphical form in Figure 3.18.

The traditional approaches such as MA, SES and AR took less time to make predictions, while other higher level approaches such as CNN, LGBM and the proposed SBFA took a lot of time for training but produced much lower supply chain cost which is the eventually the main goal of any organisation. From Figure 3.18, it can be seen that the advanced approaches, such as LGBM and CNN, took comparatively larger training time than traditional approaches. Still, the supply chain cost associated with these approaches is significantly lower than the traditional ones. The proposed SBFA approach with even

### *3. A SWITCHING BASED FORECASTING APPROACH (SBFA) FOR FORECASTING SALES DATA IN SUPPLY CHAINS*

---

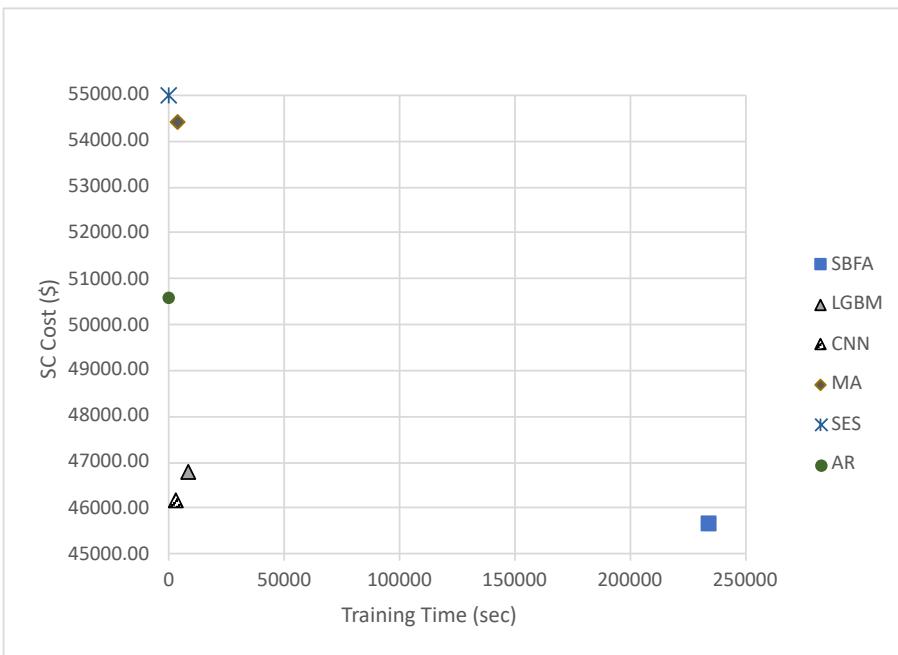


Figure 3.18: SC Cost vs Time: Kaggle Daataset

greater training time produces the lowest supply chain cost, making it an ideal choice for a forecasting approach. Although from the graph, the supply chain cost difference for SBFA is less eminent than LGBM ad CNN, for higher order supply chains or organisations, the approach could save millions of dollars. Overall, the concerned managers can make an informed choice regarding cost saving with time while selecting a particular forecasting approach.

## **3.9 Managerial Implications: Practitioners' Benefit from SBFA**

The study related to sales prediction predictions using AI would be especially beneficial for practitioners and organisations. Instead of reacting to rapid, short-term fluctuations, they can see the entire picture, focusing on a business's short to eventually long-term growth or analysing the viability of a new business model. With the advances in AI and machine learning, they will have the advantage of calculating future expenses and

adjusting their pricing policies and inventory management. They could even think of future investments with minimal risk. Supply chain forecasting enhances operations by planning for unforeseen events, delays, and changes in demand. One may reduce costs, satisfy customer demand, and maintain the efficiency of the supply chain by leveraging data and other insights to make smarter decisions. As organisations make more profit by earning consumers' confidence, the overall productivity increases, ultimately creating a positive impact on GDP.

### **3.10 Summary**

This chapter proposes a switching-based forecasting approach (SBFA) that can make short to long-time horizon forecasts. SBFA consists of state-of-the-art DL/ML approaches with validation error to be the discriminator for selecting between different approaches. Cross-validation analysis and statistical tests proved that SBFA produced lower RMSE than other approaches. Furthermore, the predictions' managerial implications were analysed using a well-developed SC model, where the reorder point is calculated based on the predicted values. Based on varied settings (e.g., static and dynamic lead times, varied safety stock levels) and their corresponding results, it has shown that better prediction leads to lower overall SC cost and that the proposed SBFA can achieve this. A forecasting algorithm selection procedure is also demonstrated from the findings of this research work. It has been shown empirically that the selection of any approach, including SBFA, can be influenced through consideration of forecasting horizon, accuracy, time and money saved. A combination of these factors can help practitioners select a particular approach.

In future, the research could be extended by incorporating more approaches within SBFA. The impact of grid search on NN can also be investigated in future. Implementing transfer learning could be another interesting research avenue enabling practitioners to predict the sales of a newly launched product. According to Ni et al. [2020], developing an ML tool and comparing its accuracy with other approaches is not enough. More research needs to be done to explore the interpretability of ML algorithms in dynamic settings that can

### *3. A SWITCHING BASED FORECASTING APPROACH (SBFA) FOR FORECASTING SALES DATA IN SUPPLY CHAINS*

---

be tested in future. This could be done by developing a computerised SC model, known as a digital supply chain twin, that works with real data and in parallel to actual SC, thereby increasing end-to-end visibility [Ivanov et al., 2019a]. An Artificial Intelligence-based digital supply chain twin can work with real-time data and can be used to quantify a supply chain network through different key performance indicators, such as sales, revenue, and cost [Ivanov and Dolgui, 2020].

## Chapter 4

# A Multi-Source Multi-Layer-based Transfer Learning Approach for Forecasting Customer Demands of Newly Launched Products

*The work presented in this chapter has been reported in the following article:*

- Ahmed, S. S., Chakrabortty, R. K., Essam, D. L., and Ding, W.. “A Multi-Source Multi-Layer-based Transfer Learning Approach for Forecasting Customer Demands of Newly Launched Products,” *Expert Systems with Applications*, Under review.

## Overview

*While the previous chapter dealt with predicting the demand for products that have been in the market for quite some time, however forecasting the future demand for newly launched products has been challenging for supply chain practitioners, often due to the lack of data. However market surveys and extracting knowledge by examining similar market products to*

## *4. A MULTI-SOURCE MULTI-LAYER-BASED TRANSFER LEARNING APPROACH FOR FORECASTING CUSTOMER DEMANDS OF NEWLY LAUNCHED PRODUCTS*

---

*find the behaviour of a new product can be inaccurate and lead to erroneous results, which ultimately lead to a misestimation of the overall cost of a business. Meanwhile, with the advancement of artificial intelligence (AI) approaches, such as Transfer Learning (TL), this misestimation of cost can be reduced by more accurately forecasting the demand for newly launched products by seeking knowledge from the historical data of other similar products. Consequently, this chapter investigates several AI-based TL approaches to predict customer demand for new products/stores. Among the different approaches, a **Multi-Source Multi-Layer-based TL approach with a Recursive Feature Elimination (MSML-TL-RFE)** strategy, is proposed to exploit the knowledge extraction power of the model from different sources, for different days-ahead-prediction. In this chapter, an abstract concept of a supply chain, with information sharing among retailers, is investigated to show that such concepts can improve the information transfer ability of a system. A hierarchical two-echelon supply chain model with different attributes is developed to validate the proposed MSML-TL-RFE approach against a few other TL-based forecasting approaches. The feature-rich datasets are then transformed in such a way that they depict a hierarchical supply chain structure, allowing for the effective application of TL for forecasting consumer demand for recently introduced products. Continuing with that idea of information sharing, finding comparable sources for a quick and effective information transfer procedure is investigated, while considering the peculiarities of certain data sets. MSML-TL-REF predictions and other TL-based approaches are analysed by calculating overall supply chain costs. Based on overall supply chain costs under static and dynamic lead time settings, the effectiveness and applicability of the proposed MSML-TL-RFE against traditional forecasting approaches are demonstrated.*

### **4.1 Introduction**

In the previous two chapters, two approaches, namely PLR-ASLTNN and SBFA, were proposed under the assumption that a significant amount of existing data is available to improve future estimations of demands. However, both approaches were not tested

with limited data, and the literature suggests that deep learning models do not perform well under such circumstances. For instance, in the context of supply chains, products that are recently launched in the market represent limited data. To address this issue, artificial intelligence (AI) advancements enable practitioners to leverage knowledge from similar market products, which will be discussed in detail in this chapter. Furthermore, this chapter will assist managers in selecting the most cost-effective channel to carry out business operations with their new products.

Of late, developments in the field of AI, especially with deep learning models and their applications, have been substantial. In applications, such as fully autonomous cars [Faisal et al., 2019] and pre-trained chat-bots [Floridi and Chiriatti, 2020] where human-like movements, reactions and generations of texts can be generated with the advent of AI-based approaches. These advancements in applications of AI were made partly possible due to the application of Transfer Learning (TL) [Niu et al., 2020]. In simple terms, TL is a system where a deep learning architecture is pre-trained on a dataset  $D_S$ , and then fine-tuned on a target domain dataset  $D_T$  based on the knowledge obtained from the training.

Current deep learning networks are successfully deployed in many areas, for example, medical care [Kaul et al., 2022], renewable energy [Abualigah et al., 2022], stock markets [Ahmed et al., 2022], production [Wang et al., 2018] and supply chain stock optimisation [Kilimci et al., 2019]. Humans make decisions based on the predictions derived from these deep models. For example, in the field of demand prediction of spare parts [Ma et al., 2021], the purchasing team of a company can take informed decisions based on deep learning predictions [Amin-Naseri and Tabar, 2008].

In supply chain management, AI is being implemented gradually over time [Garinian et al., 2023]. Through more accurate demand forecasting, businesses can gain significant competitive advantages in production, procurement and logistics [Ivanov et al., 2019b]. But the requirement of using neural networks is to have good quality data on the one hand while an appropriate quantity of data on the other [Oliveira et al., 2005]. Compared to large companies, it is often difficult for small and medium-sized enterprises to have ample

#### *4. A MULTI-SOURCE MULTI-LAYER-BASED TRANSFER LEARNING APPROACH FOR FORECASTING CUSTOMER DEMANDS OF NEWLY LAUNCHED PRODUCTS*

---

data to train such a network. This is relevant in the case of newly launched products, as there is often an insufficient amount of historical data for neural networks to make predictions [Li et al., 2022b].

Time series data refers to sequential data points that are collected over time. Such data can be quite complex, displaying a range of diverse behaviours. As Syntetos et al. [2005] have pointed out, some time series data can be erratic and unpredictable, while others are smoother and more predictable. In some cases, the data can be lumpy and intermittent, with unpredictable spikes and dips. These various types of behaviours make it difficult to use common measurement metrics, like Root Mean Squared Error (RMSE), to evaluate the data and also limit the use of a single machine learning or deep learning approach. Essentially the nature of time series data requires a more nuanced and adaptive approach in order to analyse and extract insights from it effectively. Despite these challenges, researchers are keen to use machine learning (ML) and deep learning (DL) approaches for predicting such complex time series data. In doing so, scholars have found that it is not evident whether AI methods consistently produce better accuracy than statistical systems. For example, Kourentzes [2013] and Kiefer et al. [2021] showed that AI systems do not always provide users with the best results, indicating that there is a need for alternative methods. Additionally, research gaps have been identified in the field of lumpy and intermittent demand forecasting by Nikolopoulos [2021].

Over the years, along with the advancement of mathematical and machine learning theories, many algorithms with good performance for time series forecasting have been proposed by many prominent scholars. These include: Exponential Smoothing (ES) [Corberán-Vallet et al., 2011], Auto-Regressive Integrated Moving Average (ARIMA) [Lee and Tong, 2011] model, Gradient Boosting Machines (GBM) [Taieb and Hyndman, 2014], Neural Networks (NNs) [Ranganayaki et al., 2020], Long Short Term Memory (LSTM) [Liu et al., 2022a], Gated Recurrent Unit (GRU) [Veeramsetty et al., 2022] and numerous others [Gu and Dai, 2021]. However, only linear and stationary time series data, such as ARIMA, can be used with linear models. In order to get reasonable prediction results, such as with the ARIMA model, practitioners try to transform nonlinear and nonstationary data

into smooth time series data. Traditional linear prediction approaches struggle to adapt to real-world time series data because of their nonlinear and non-stationary nature. However, Difference processing can partially produce stationary time series data from non-stationary data by removing underlying trends and seasonality. According to Gu and Dai [2021], this is usually insufficient to explain how time series data varies over time adequately. In addition, if inaccurate or incomplete data have been added to a time series, difference processing cannot be properly carried out.

According to empirical investigations, nonlinear models typically function more effectively and reliably than linear algorithms [Gu et al., 2021]. Since artificial neural networks can recognise non-linear cause-and-effect relationships, this chapter intentionally concentrates on deep learning techniques. Time series forecasting of demand using deep learning approaches has already shown reliable results according to [Criado-Ramón et al., 2022], [Huang et al., 2022] and [Ashtari et al., 2022]. Two crucial assumptions are used in conventional demand forecasting with time series data [Gu and Dai, 2021]: (i) the feature space from which the training and test sets of data are drawn should be the same, and those should follow the same probability distribution. (ii) enough training samples must be accessible in order to develop a strong prediction model. In many real-world application scenarios, the evolution of time series data over time results in a sizable disparity between recent and historical data. The amount of data that is readily accessible in real-world applications, however, is quite minimal, which leaves inadequate training data. The two key presumptions cannot be satisfied in these circumstances.

To address all these challenges, Transfer Learning (TL) has been suggested as a promising solution for time series prediction. By leveraging pre-trained models, TL can reduce the need for extensive training data and computational resources, which means it can produce good improvements, especially when the target data set is sparse [Pan and Yang, 2009]. Recent studies, such as [Gautam, 2022, Weber et al., 2021], have shown that TL can significantly improve the accuracy of time series prediction in comparison to traditional methods. By transferring knowledge from a similar domain, TL can be utilised to enhance a learner's knowledge of one particular domain. Suppose there is a learning task  $T_S$  in

#### *4. A MULTI-SOURCE MULTI-LAYER-BASED TRANSFER LEARNING APPROACH FOR FORECASTING CUSTOMER DEMANDS OF NEWLY LAUNCHED PRODUCTS*

---

source domain  $D_S$  and a learning task  $T_T$  in target domain  $D_T$ . TL seeks to enhance the learning of the target predictive function  $f(t)$  in  $D_T$  by acquiring knowledge from  $D_S$  and  $T_S$ . In this way, TL can be effectively applied in predicting time series data.

To date, TL methods have not been broadly investigated by researchers, especially in the field of supply chain domain [Maschler, 2023, Chen et al., 2023, Li and Zheng, 2020]. However, there are some pieces of evidence of applications of TL in time series anomaly detection [Xiong et al., 2018] and in stock market predictions [He et al., 2019]. This shows that the use of TL has gained momentum in other fields, while it needs further investigation in the supply chain domain to be able to predict outcomes for newly launched products with limited data. The accurate forecasting of demand for such products is crucial for effective inventory management and meeting customer expectations. However insufficient data on a new product can significantly challenge the training of accurate predictive models [Zhang et al., 2022b, Liu et al., 2022b].

To address these challenges, the research methodology used in this chapter involves the implementation of a base CNN network for predictions, followed by the establishment of different aspects of the TL methodology. A single item is identified from  $D_S$  that is close to  $D_T$ , and it is then used to train the CNN network. Pre-trained models are then used to make predictions in  $D_T$ . Next, the study identifies multiple sources of information for effective information transfer and develops three different TL approaches for multiple sources, namely the: Multi-Source Weighted Average TL Approach, Switching-based TL Approach, and Multi-source Multi-layer based TL Approach. The performances of each of these approaches are compared to establish the superiority of one over the other in terms of accuracy. Additionally, two supply chain scenarios are simulated to showcase the benefits of information sharing among retailers, ultimately leading to better accuracy in the target domain and reducing supply chain costs along different channels. Cost calculations are also performed to show the improvement in supply chain performance along different channels of the network due to information sharing. Finally, the chapter evaluates the cost minimisation potential of transfer learning in a supply chain scenario for predicting demand for a newly launched product. Hence, in short, the contributions stemming from

this work are multi-fold and include:

- Identification of similar sources for efficient information transfer process is explored, where all available features of a particular data are taken into consideration. In order to find comparable items with sales patterns similar to those of the new product, a KNN network is trained using all available sales features from restructured datasets aside from the identified newly introduced market product.
- The effect of harvesting knowledge from single to multiple sources on transfer learning (TL) approaches is explored. Starting from a base forecasting model, as many as five TL approaches are pre-trained on identified similar sales data that are later used to be retrained on new product data for future estimation of customers' demand. The performance of these different TL techniques is evaluated against each other to explore their effectiveness.
- A novel Multi-source Multi-Layer Transfer Learning with Recursive Feature Elimination Technique (MSML-TL-RFE) is proposed, which is capable of efficiently transferring knowledge from a source domain to a target domain to make predictions.

The remainder of this chapter is organised as follows: Section 4.2 presents a brief overview of existing literature, mostly related to the application of TL. Based on certain assumptions regarding limited data, the problem formulation is discussed in Section 4.3. Then from the existing literature and problem definition, Section 4.4 provides a brief summary of the different TL approaches for making sales/demand forecasts. The implementation of the model, along with data collected from different public platforms and their feature extraction techniques, is discussed in Section 4.5. The findings from the experimentation are analysed and discussed in Section 4.6. Finally, a summary and future directions are presented in Section 4.7.

## **4.2 Literature Review**

The literature review section of this chapter aims to provide an in-depth understanding of transfer learning and its current methods, as well as their application fields. This section explores the existing research on transfer learning and its different approaches and also discusses their advantages and limitations. Moreover, the different areas of application where transfer learning has been used with promising results are examined. Towards the latter part of this section, some research gaps in the existing literature are highlighted, and potential avenues to address these gaps are discussed. This comprehensive review aims to provide a solid foundation for the subsequent sections of the chapter that present the specific contributions of this research work.

Businesses often face a challenge when launching new products in the market, as the demand for a new product is often unknown. With very limited historical data, it becomes difficult for a business to predict the future demand of their product using regular machine learning or deep learning tools. Of late, a lot of effort has been undertaken by researchers regarding image-based transfer learning [Saleem et al., 2022, Ju et al., 2022, Jangam et al., 2022, Wang et al., 2021a, Muhammad et al., 2020]. However very little is being explored by the researchers in the area of time series analysis, especially in the context of supply chain sales prediction of newly introduced products.

In the field of supply chains, researchers are trying to use TL to reduce machine downtime through prompt and accurate detection of faults, so that manufacturing organisations can stay competitive throughout the year [Rai et al., 2021, Wang et al., 2023b, Zhang et al., 2022a]. As more and more clients demand manufacturers to expedite the delivery of high-quality products cheaply, machine learning algorithms in the manufacturing industry will see growing utilisation for allowing better fault diagnosis of manufacturing systems. Using deformable convolutional neural network (CNN), deep long short-term memory (DLSTM) and transfer learning strategies, Wang et al. [2021c] presented a CNN-DLSTM-based TL approach to finding the faults in the roller bearings of machines. By detecting potential faults in advance, manufacturers and suppliers can plan for maintenance and

repair, reducing the risk of unplanned downtime and disruption to the supply chain and improving the overall efficiency of a network. Their system was trained using sample data (since real data is difficult to find) during a pre-training phase and then later tested on their system's experimental data to assess their approach.

In another work, researchers like Wu et al. [2022] used a information transfer technique from an industrial chain to predict stock market movement. The authors presented a TL idea of fusing information related to the industry chains with deep learning models to forecast stock prices. The authors' research work was based on the TL of industrial chain data of upstream businesses. After being first trained using the stock data of upstream businesses, the DL models (i.e., MLP, RNN, LSTM and GRU) were then retrained using the stock data of the appropriate downstream businesses to forecast the trend of the target stock. The authors found that MLP's performance was better in transferring knowledge, having an accuracy of 68.4%, as compared to other approaches such as RNN, LSTM and GRU whose accuracy was in the range of 57%-58% while predicting the target's stock market index.

Continuing with TL systems, Ramezankhani et al. [2021] proposed a transfer learning (TL) architecture for the autoclave composite manufacturing process. The authors used historical data to train a fully connected neural network (FCNN) and transferred the knowledge to a new process. They avoided over-fitting with limited data by using a sequential freezing method. The model's generalization loss was reduced by 88%, but the authors used only one type of DL technique, while other ML/DL models remain unexplored. Moreover, they suggested changing the number of frozen layers to study the model's performance and conducting a similarity analysis between the source task and target task to maximise information transfer and minimise negative transfer. This could help identify suitable knowledge sources for future TL researchers.

Considering the source of knowledge could also contribute to selecting an appropriate technology as a source. According to Da Silva et al. [2019], industries do not commence their operation with all the characteristics of Industry 4.0 from the very beginning. They evolve over time and adapt new technologies from other stakeholders. The study conducted

#### *4. A MULTI-SOURCE MULTI-LAYER-BASED TRANSFER LEARNING APPROACH FOR FORECASTING CUSTOMER DEMANDS OF NEWLY LAUNCHED PRODUCTS*

---

by Da Silva et al. [2019] encouraged the continuous development of technologies that need to be transferred to meet market demand [De Carvalho and Viana Cunha, 2013]. TL could be an effective way to deal with this technological knowledge as a transfer system can aid in the performance management of industries through managing control systems and evaluating the performance indicators of a business.

In the case of future data predictions in supply chains, researchers proposed that TL can be an efficient tool. A study by Karb et al. [2020], investigated a TL approach for DNN to predict the demand of newly launched food sales for an Austrian food retail industry. The authors transferred knowledge from preexisting stock products to predict the demand for newly introduced products and showed that DNN with TL works better than DNN without TL. However, the researchers trained only one single model for their entire application, and using a parameter-based TL approach with multiple neural networks and hyper-parameter optimisation remains unexplored.

In order for TL approaches to work effectively, it is necessary to identify the proper sources of information. Over the years, researchers tried to harvest information from a single source and also from multiple sources [Nicholson et al., 2022, Huang et al., 2023, Zhao et al., 2023]. In an effort to deal with gathering knowledge from a single source and also trying to simulate real-life scenarios where there are limitations of labelled data, Ye and Dai [2021] proposed a DTr-CNN approach, that is a deep transfer learning method using the architecture of CNN to cope with insufficient data to make time series forecasting. Their transfer learning framework uses dynamic time wrapping (DTW) and Jensen-Shannon divergence to identify similar source tasks to the target task and then uses DTr-CNN to transfer knowledge from the source to the target. The authors conducted experiments on three types of datasets and found that DTr-CNN outperformed other approaches in terms of RMSE and sMAPE. However, this approach can only select one source dataset from multiple datasets and can only work with time series data with the same feature size. The authors suggested adapting multi-source transfer-learning techniques for better accuracy and using an appropriate selection approach in the future for varying feature sizes.

Research related to time series forecasting using TL is rare, especially in the context of

supply chains. An article by Kiefer et al. [2022] analysed the use of TL in combination with DL for time series forecasting in the context of supply chains. The authors used M5 competition data to conduct their experiments and found that the TL approach can reduce prediction error for all time series data, particularly for short and medium time series. However, for time series with a long history, benchmark models work better than TL. The chapter suggests future research directions, such as conducting a similarity analysis of data in the source and target domains, using more advanced neural networks, and training a model on multiple source domains.

A TL framework was presented by Sagheer et al. [2021] to address the hierarchical time series prediction problem, which was not directly related to the supply chain problem. In their research work, the authors used the DLSTM autoencoder (DLSTM-AE) system in a TL framework. They generated a base forecast using DLSTM-AE from the bottom of a hierarchy, froze the weights of the base models, and transferred the learned features to attain synchronous training to the time series for the upper levels in the hierarchy. The authors showed that their proposed DLSTM-AE produced a more accurate and coherent forecast than ARIMA and ES over different levels of the hierarchy. They plan to adapt their technique to other domains in the future.

One of the most challenging tasks for any business is to identify how a newly launched product would behave in the market. Although a lot of market surveys can be done before a launch, its behaviour could still be better identified by TL approaches [Wu et al., 2022]. During the launching of a new product in the market, it is important to determine its demand as early as possible. Research has found that a lack of consumer-related information makes this demand prediction unreliable and difficult. That is why scholars have been trying to harvest information from other market products to predict the demand for recently introduced products.

Newly introduced products have some commonality and/or differentiation with their predecessors, and such data is widely available and studied when required [Afrin et al., 2018]. Yang et al. [2019] proposed a mathematical architecture to forecast demand for a newly launched product by using a weighted product differentiation index from the market de-

#### *4. A MULTI-SOURCE MULTI-LAYER-BASED TRANSFER LEARNING APPROACH FOR FORECASTING CUSTOMER DEMANDS OF NEWLY LAUNCHED PRODUCTS*

---

mand of its predecessor. The authors predicted the demand for a new product by using an exponentially weighted moving average machine learning forecasting approach. The method was tested on automobile industry data, and the authors demonstrated its efficacy. However, the method needs validation on other datasets, and the weights for different components and options of products were manually assigned, which is cumbersome for large datasets. The relationship between different features of products that influence their corresponding demand is yet to be established, which can be done using a deep neural network model.

As mentioned previously, companies often face challenges predicting the demand for newly launched products due to the lack of historical data. Kharfan et al. [2021] proposed a three-step model to predict the demand for newly launched products in fashion retail, using clustering, classification, and prediction. They identified similar products from a training set using clustering, established a relationship between the styles of pre-assigned clusters from the train set and new styles from the validation and test sets through classification, and predicted sales of the new product using the identified clusters. The authors suggested that regression trees can be used for complex clustering problems with multiple life-cycle lengths, while k-NN and linear regression approaches can be used for identifying high sales volume and average unit retail products. The chapter had some limitations as well, such as not considering lost sales and the potential use of deep TL instead of averaging historical sales data.

##### **4.2.1 Research Gaps**

As discussed, although much effort has been dedicated to the transfer learning approach for forecasting customer demands, the noted algorithms suffer from the following limitations and challenges.

1. More research needs to be conducted to identify potential sources of information to enhance the TL processes.

2. As suggested by many researchers, more investigations need to be done to observe the effect of harvesting knowledge from single to multiple sources using one or more TL approaches.
3. They also want to examine how the performance of different TL approaches varies over the selection of the different number of sources.
4. Although it has been mentioned as an abstract concept in past research articles that information sharing leads to a more resilient SC network, but almost no one has simulated this concept to make an empirical comparison.
5. Moreover, researchers mentioned cost minimisation as their future work, but a more thorough investigation needs to be done to check whether and how much TL really helps SC to minimise cost, especially when it is related to a newly introduced market product.

Addressing the above-mentioned gaps will certainly benefit practitioners in making informed decisions and choices regarding the use of TL approaches to predict the demand for a newly launched product. It will also help them to run their business at a lower cost by reducing unnecessary holding and lost sales costs. Consequently, this chapter has identified multiple sources of information for effective information transfer, in predicting demand for a newly launched product. It investigates various transfer learning approaches with varying numbers of sources for the same target product. The chapter also demonstrates the concept of supply chain collaborative learning through information sharing and conducts an analysis to evaluate the cost minimisation potential of transfer learning in a supply chain scenario for a newly introduced market product.

### **4.3 Problem Definition**

In order to provide a practical idea of how much TL approaches actually benefit an SC network, all the TL approaches explored in this chapter, are applied to supply chain data to

#### 4. A MULTI-SOURCE MULTI-LAYER-BASED TRANSFER LEARNING APPROACH FOR FORECASTING CUSTOMER DEMANDS OF NEWLY LAUNCHED PRODUCTS

---

evaluate their impact. It is worth mentioning that all the experiments are conducted based on the assumption that a product has been recently launched for around one month in the market. Based on the sales pattern of that product during that time, similar items are identified from other groups of products to extract knowledge. For such identifications, a well-known KNN network is used and then a base CNN model is trained with data obtained from similar products and then the weights and bias of the pre-trained network are transferred to form a new NN which is then retrained using one month of data from the recently launched product to predict 6 months of data.

To further extrapolate the assumptions, the collected data is restructured in such a way that it represents a hierarchical supply chain structure where there is one supplier and three retailers selling different products. Single and multi-source TL approaches are implemented on three datasets under two scenarios (1) no information sharing and (2) with information sharing. A detailed representation of horizontal information sharing among the stores is shown in Figure 4.1.

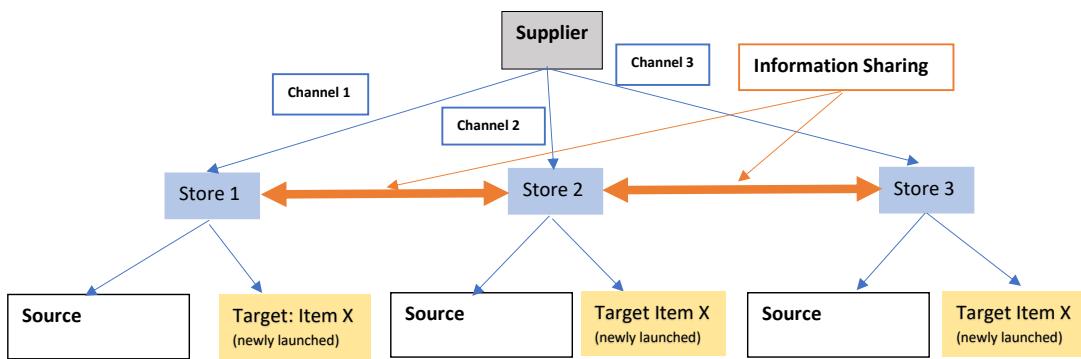


Figure 4.1: Supply Chain Information Sharing Concept

From Figure 4.1, it can be seen that each store sells numerous items for some previous time, which act as  $D_S$  data while there is also one product (Item X) which is supposedly the new launch product (according to the assumptions) in the market to act as the  $D_T$  data in this experimental analysis. With no information sharing, only the locally long-term sold items per store are considered, among which the most suitable product/s are chosen as per the KNN algorithm, while with information sharing, items sold at other stores are also taken into consideration. The two concepts are discussed in detail in the latter part

of this subsection. Once the predictions and accuracy of the different TL approaches are established, those predictions are then passed through a supply chain model for further cost calculations. The working principle of the supply chain model is also discussed later in this chapter.

For properly establishing the TL approaches on the no information sharing concept, with distance matrix being the determinant, similar items are ranked and identified using KNN from a fixed set of products that are sold by a single retailer, e.g. Store 1, thus keeping the knowledge sharing concept contained within the specified Store 1. Once the items are ranked, the 1st three products that are the shortest distance from the target product are chosen. The Euclidean distances from KNN are used to calculate the weights of each similar item through a method called inverse weighted distance [Pereira et al., 2022]. These weights are later assigned for finding the impact of the weighted average of a multi-source TL system where multiple layers get information from multiple pre-trained models in the entire TL system.

On the other hand, if the retailers agree to share their product information with each other, then similar items can be identified from a larger pool of items using the same KNN system. According to Figure 4.1, with the horizontal information sharing concept, each store now has information on the items and sales patterns of other stores. This means that information can now be extracted from finding out similar items from other retail outlets, as well as its own set of products. It has been mentioned by several researchers in the past [Ivanov et al., 2021, Dolgui et al., 2020, Ivanov et al., 2018], that sharing information among competitors adds value to the entire supply chain network. It enhances flexibility and makes the network more resilient. This chapter focuses on the cost-minimisation aspect of information exchange through TL in supply chains.

Next, a hierarchical supply chain model is considered in this chapter which consists of at least 3 retailers and 1 distributor (supplier). This means that there are three channels through which products can be delivered to customers. Each of the channels is basically a two-stage supply chain architecture that consists of a retailer and a supplier. The supplier-retailer simple model is developed based on the following assumptions: the retailer sells

---

*4. A MULTI-SOURCE MULTI-LAYER-BASED TRANSFER LEARNING APPROACH FOR FORECASTING CUSTOMER DEMANDS OF NEWLY LAUNCHED PRODUCTS*

---

the newly launched product alongside other commodities, there is no backorder cost, but rather a cost for lost sales is incurred whenever the retailer fails to satisfy customers' needs, the supplier has storage facility to store and deliver the finished product to the retailer after an order is placed by the retailer.

The decision of whether to place an order or not by the retailers is determined by the reorder amount (RA). This RA is calculated from the predicted demand during the lead time (LTPD) produced by the machine learning approaches. Once an order is placed, the retailer's inventory is replenished after a certain amount of lead time. The question of when to place an order is dependent on the summation of the amount in delivery and the current inventory level. The model can be operated with fixed or dynamic lead time. In the context of the model, fixed lead time is defined as a finite amount of time between when an order for a certain number of units is received by the supplier and when the units are fully delivered to the retailer's inventory. This fixed lead time becomes dynamic when the delivery time varies within a given range but it is known during the placement of an order.

The developed supply chain model also calculates the predicted demand during the lead time, named lead time predicted demand (LTPD), which is assumed to be provided by the TL forecasting algorithms. Using the variation of previous lead time and the predicted demand, the model can be operated with a certain amount of safety stock (ss). Safety stock is an extra quantity of goods stored in the inventory which prevents a business from a stock-out situation. The summation of LTPD and ss together creates a RA. The model also keeps track of the total amount of orders in delivery. So for each of the channels, whenever RA becomes greater than the summation of the current inventory level and orders in delivery, an order placement is issued to the supplier for replenishment. RA can also be varied by changing the level of safety stock. This is an added feature to make the model more realistic. Meanwhile, motivated by the work of Gonçalves et al. [2020], the safety stock and the RA are determined by equations 4.1 and 4.3, respectively.

$$ss = Z \sqrt{(\sigma_d^2)\bar{L} + (\sigma_L^2)\bar{d}} \quad (4.1)$$

where,  $\sigma_d$ = standard deviation of predicted demand;  
 $\sigma_L$ = standard deviation of Lead time;  
 $\bar{L}$ = average lead time;  
 $\bar{d}$ = average predicted demand;  
 $Z$ =number of standard deviations, corresponding to service level probability.

Safety stocks have two associated dimensions: demand variability and lead time variability of suppliers. Variations in demand can be found from the ML/DL predictions, but with lead time there are two different possibilities: fixed lead time and dynamic lead time. When the lead time is fixed,  $\sigma_L=0$ , and hence equation 4.1 is reduced to equation 4.2. The symbols in equation 4.2 have their usual meanings.

$$ss = Z\sigma_d\sqrt{\bar{L}} \quad (4.2)$$

$$RA = LTPD + ss \quad (4.3)$$

The model has four costs associated with it: purchasing costs, ordering costs, holding costs and lost sales. The summation of these costs is considered as the total cost (TC) of the supply chain according to Equation 4.4.

The following are the assumptions regarding the associated costs:

Purchasing cost (PC): The purchasing cost per unit product is known and fixed.

Ordering Cost (OC): It is assumed to be known, fixed and independent of order quantity.

Holding Cost (HC): It increases or decreases linearly with the amount of inventory in stock.

Lost Sales (LS): It is assumed to be fixed per unit and the overall lost sales increase linearly as more goods are out of stock.

$$TC = \sum(PC, OC, HC, LS) \quad (4.4)$$

---

#### *4. A MULTI-SOURCE MULTI-LAYER-BASED TRANSFER LEARNING APPROACH FOR FORECASTING CUSTOMER DEMANDS OF NEWLY LAUNCHED PRODUCTS*

---

Out of three datasets, only one dataset (Dataset 1) is considered for evaluation of the proposed TL approach in terms of supply chain cost in this chapter, for which the supply chain model's parameters are set as follows. But it could be slightly changed to ensure proper triggering of order placement for different datasets, because of the different sales levels in different datasets. The parameters used for the experimentation are presented in Table 4.1.

Table 4.1: Input data for the Supply Chain Model

SC Parameters	Dataset 1
Ordering Cost (per order)	\$20
Purchasing Cost (per unit)	\$2
Holding Cost (per unit per year)	\$7
Lost Sales Cost (per unit)	\$5
Retailer's Beginning Inventory (units)	250
Retailer Lot Size Q (units)	200

From Table 4.1, it can be seen that ordering cost is assumed to be \$20 per order while purchasing cost is \$2. The annual holding cost is \$7 per unit while the cost of a lost sales opportunity is assumed to be \$5 for each product unit. For each fresh set of experiments, the retailers beginning inventory is initialised to 250 units and once the remaining inventory orders in place drop below RA, 200 units are ordered to replenish the stock, which is the lot size  $Q$  in this case.

## **4.4 Proposed Methodology**

In this chapter, a simple base CNN network is 1st implemented for predictions and then it is extended to establish different aspects of our TL methodology. From  $D_S$ , a single item is identified which is close to  $D_T$ , and it is used to train the CNN network. The pre-trained models are later used for making predictions in  $D_T$ . Next, instead of limiting the knowledge pool to a single source, multiple sources are identified to enhance the performance of the TL system. For multiple sources, three different TL approaches are developed (i) Multi-Source Weighted Average TL Approach (ii) A switching-based TL

approach and (iii) Multi-source Multi-layer based TL approach. This Multi-source Multi-layer based TL (MSML-TL) approach is further extended by the addition of RFE for enhanced performance and it is demonstrated by an overall flowchart in Figure 4.2. The performances of each of these approaches are compared to establish the superiority of one over the other in terms of accuracy.

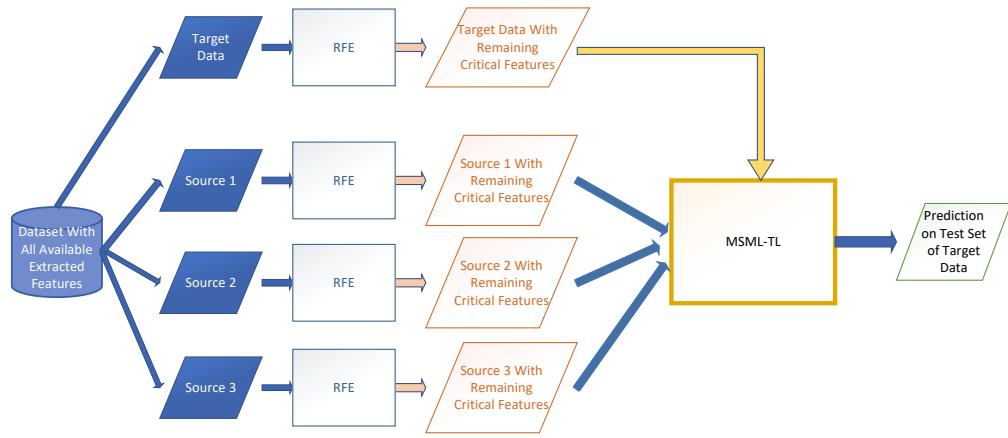


Figure 4.2: Overall flowchart of the MSML-TL-RFE

#### **4.4.1 Simple CNN Network: No-TL Approach**

Initially, a simple Convolutional Neural Network (CNN), adapted from the literature [Mehtab and Sen, 2022], is deployed in the target domain data for predictions. CNN is a kind of deep NN that is commonly used in experimentation with image data. With sequential data, Convolutional layers have the capability to withdraw critical information and learn hidden patterns of data [Livieris et al., 2020]. The Convolution function defines the inner product (product and sum) of window data with a filter matrix which is a fixed set of weights. The objective of the MaxPooling layer is to simultaneously minimise the eigenvectors' size, which is needed to simplify the network's computational complexity and compression and extraction of critical features. Figure 4.3 shows a one-dimensional CNN model.

In this chapter, a simple CNN-based neural network from the literature [Mehtab and Sen, 2022] is deployed with slight modification, by some additional layers. It is trained and

#### 4. A MULTI-SOURCE MULTI-LAYER-BASED TRANSFER LEARNING APPROACH FOR FORECASTING CUSTOMER DEMANDS OF NEWLY LAUNCHED PRODUCTS

---

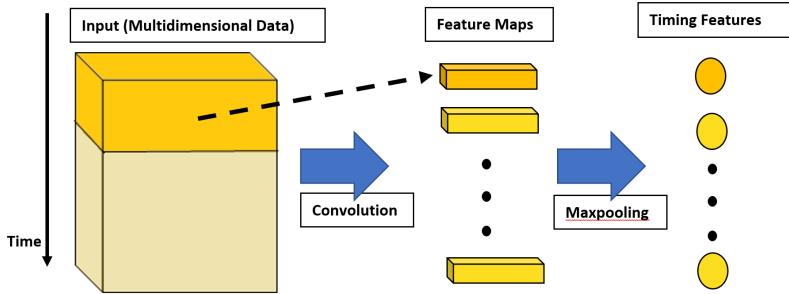


Figure 4.3: One dimensional CNN (No-TL Approach) [Jin et al., 2020]

validated with one month of data which is related to the recently launched product (target product). The CNN network contains three 1D convolutional layers with MaxPooling layers in between each. The last two layers are flattened and dense layers. The output of the final convolution layer is passed through the flattened layer, whose purpose is to convert the output into a one-dimensional array so as to create a single long feature vector. Finally, the dense layer is used to combine that single vector to produce the predicted result. Since this network is only deployed in the target domain, hence this system is making predictions without any previous knowledge from any other source and thus this system is termed a No-TL system.

##### 4.4.2 Single Source TL: SS-TL

The same CNN-based network, mentioned in Section 4.4.1, was used to obtain a pre-trained model using sales data of a similar item and then the weights and biases from the pre-trained network of the 1st 4 layers are frozen and the remaining lower level layers, along with the frozen layers, were combined together to create a new NN which was trained and validated with the data from the newly launched item. A layout diagram of the Single source TL (SS-TL) approach is illustrated in Figure 4.4.

In this approach, a similar source product is first identified using K-Nearest Neighbors (KNN), which is a very well-known classification machine learning tool. This KNN algorithm works by comparing the features of the target product to those of the sources

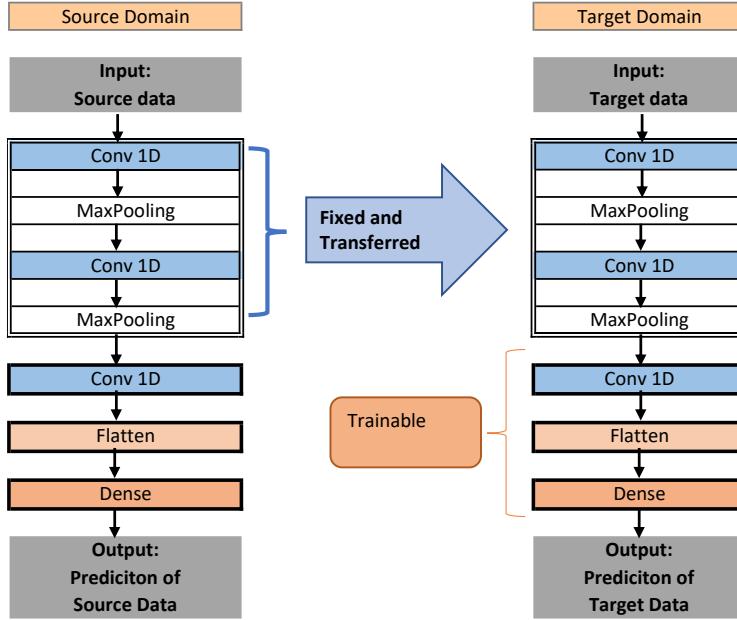


Figure 4.4: Single Source TL Network (SS-TL)

and finding the k-nearest sources based on a Euclidean distance. Once the source item is identified using KNN, it is used to train the CNN network belonging to the source domain according to the diagram 4.4. Then some layers are frozen and transferred to form a new CNN network which is then re-trained and validated using target domain data. It is worth mentioning that apart from the frozen layers, the other lower-level layers are kept trainable so that the system adapts itself to the new target domain data to make predictions. For each day ahead of prediction, the experiment is repeated and the mean predictions are calculated.

#### 4.4.3 Multi Source TL Approaches

This chapter explores three multi-source TL approaches. A multi-source weighted average is considered where different single-source models are trained using different source data

#### *4. A MULTI-SOURCE MULTI-LAYER-BASED TRANSFER LEARNING APPROACH FOR FORECASTING CUSTOMER DEMANDS OF NEWLY LAUNCHED PRODUCTS*

---

and ultimately tested using a single target set. The total prediction is a weighted sum of the individual predictions obtained from different models pre-trained using different sources. Another way to transfer knowledge from multiple sources is a switching-based TL approach, which is also presented in this chapter. Here the selection of a particular TL approach is switched between different approaches based on validation data in the target domain. The last TL approach investigated in this chapter is a Multi-Source Multi-Layer TL approach enhanced with recursive feature elimination (RFE). In this system, important features are extracted using RFE from all source and target data before training the pre-trained networks. After training, the weights and biases from multiple pre-trained networks are combined together using the weighted average method to form a new network in the target domain. All these approaches showed promising results and a detailed explanation of each one is given in the following sections.

##### **4.4.3.1 Multi-Source Weighted Average TL Approach: MSWA-TL**

To explore another dimension of the TL framework, the single source TL approach from Section 4.4.2 is used three times with three different sources with the same target domain data for a particular dataset. The source domain items/stores are selected based on the KNN algorithm. The structure of the Multi-source Weighted Average TL Approach used in this chapter is presented in Figure 4.5.

It can be seen from Figure 4.5 that there are basically three single-source TL approaches, where each of those is fed with similar source data for training. After that, the obtained pre-trained models are retrained using the target domain data. The same process is carried out for other similar products acquired from the distances of each product or store produced by the KNN model. The distances are then used to assign the corresponding inverse distance weights of each item/store. The output predictions of each of the systems acquiring knowledge from a particular source are then multiplied with the corresponding weights (which will be discussed later in this chapter) and added together to find the final predictions of the testing set of the target dataset. To remove randomness, multiple

#### 4. A MULTI-SOURCE MULTI-LAYER-BASED TRANSFER LEARNING APPROACH FOR FORECASTING CUSTOMER DEMANDS OF NEWLY LAUNCHED PRODUCTS

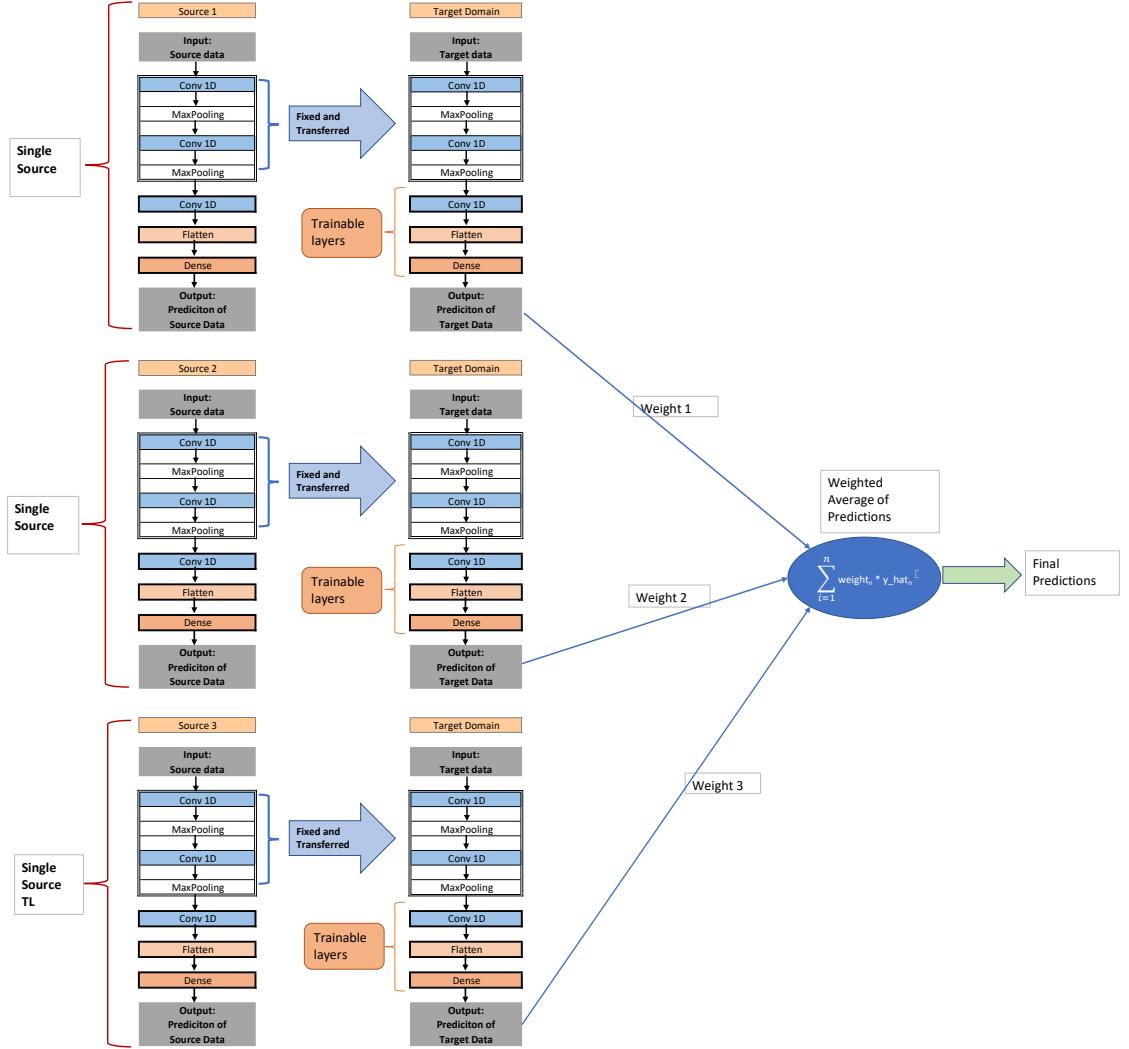


Figure 4.5: Multi-Source Weighted Average TL Network (MSWA-TL)

experiments are carried out and the average of all the weighted average predictions is calculated to find the final predictions. The finally predicted results are later compared against the testing set of the target dataset for further evaluation.

##### 4.4.3.2 Multi-Source Switching Based TL Approach: MSSB-TL

The same single source network mentioned in Subsection 4.4.2 was used multiple times to make a multi-source TL network. Multiple similar sources of information are identified

#### 4. A MULTI-SOURCE MULTI-LAYER-BASED TRANSFER LEARNING APPROACH FOR FORECASTING CUSTOMER DEMANDS OF NEWLY LAUNCHED PRODUCTS

---

using KNN and then each of the sources is used to train the network to obtain a pre-trained network for each source. The layers from each one are then transferred to create a new individual CNN network for each source which is then re-trained and validated using the target dataset. Figure 4.6 represents the architecture of the multi-source switching-based TL (MSSB-TL) approach which has been explored to establish that information transfer from multiple courses often leads to better performance in a target domain.

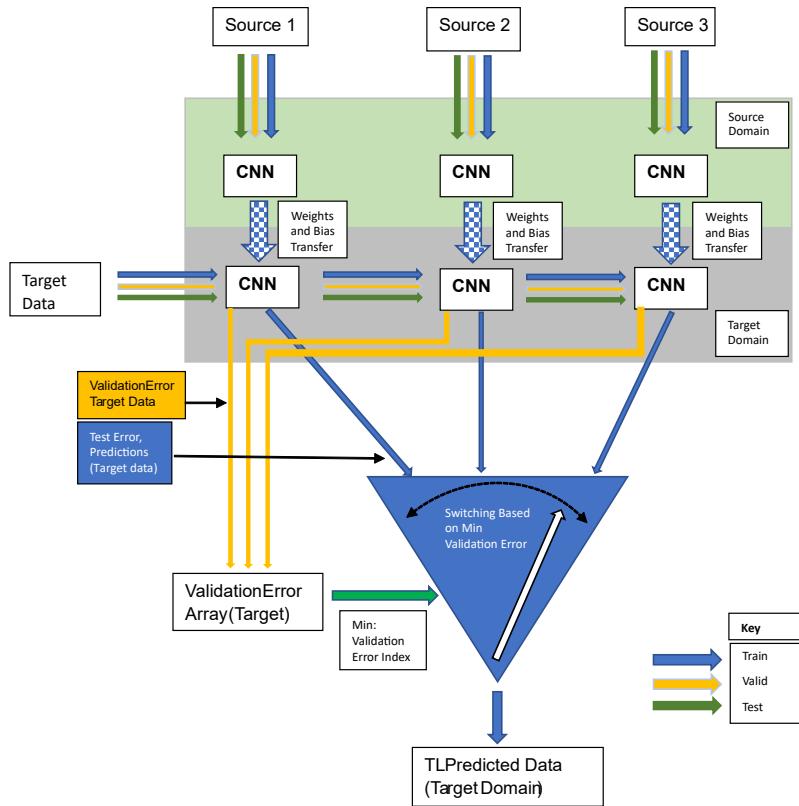


Figure 4.6: Multi-source Switching Based TL Network (MSSB-TL)

According to Figure 4.6, the switching between different models trained from different sources is based on the minimum validation of root mean squared error (RMSE). In the MSSB-TL system, predictions from a single set of TL networks are selected when there is minimum validation RMSE in the target domain. The same experiment is repeated and the average predictions are taken into consideration. The same is repeated for different forecasting horizons.

#### **4.4.3.3 Multi-Source Multi-Layer TL Approach: MSML-TL**

Adapting and extending the idea further from SS-TL, MSWA-TL, and MSSB-TL networks, another multi-source TL framework is presented in this chapter, where the same network is trained using multiple sources at different times. Then the weights and biases of each of the networks from the pre-trained models are then combined together using the weighted average method to form the weights and biases of some new frozen layers of another CNN model. The newly formed model with its new frozen layers with other trainable lower-level layers is re-trained and validated using data from the target domain. A layout of the proposed Multi-Source Multi-layer based TL (MSML-TL) approach that harnesses knowledge from multiple sources is illustrated in Figure 4.7.

From Figure 4.7 it can be seen that there are three pre-trained models from which knowledge is harnessed. As mentioned before, this chapter initially focused on three sources when multi-source TL is mentioned. The three sources are used to pre-train three CNN models and then a weighted average is calculated for weights and biases of each layer of the three CNN models. The weighted average of each of the corresponding layers is then passed to new CNN models where those particular layers are frozen and the entire network is re-trained with target domain data. As usual, for each day ahead of predictions, multiple experiments are carried out and the average value is calculated to overcome randomness in the system.

#### **4.4.3.4 Multi-Source Multi-Layer TL Approach with RFE: MSML-TL-RFE**

Before feeding data into the previously mentioned MSML-TL approach, critical features were extracted using a popular technique called Recursive Feature Elimination (RFE) [Lee et al., 2022]. The process is presented in Figure 4.8 where the important features are mainly selected from similar source data.

Using RFE, around 40-60% of features are extracted from each data set. This was done to train the networks using the most important features and thus enhance the information

#### 4. A MULTI-SOURCE MULTI-LAYER-BASED TRANSFER LEARNING APPROACH FOR FORECASTING CUSTOMER DEMANDS OF NEWLY LAUNCHED PRODUCTS

---

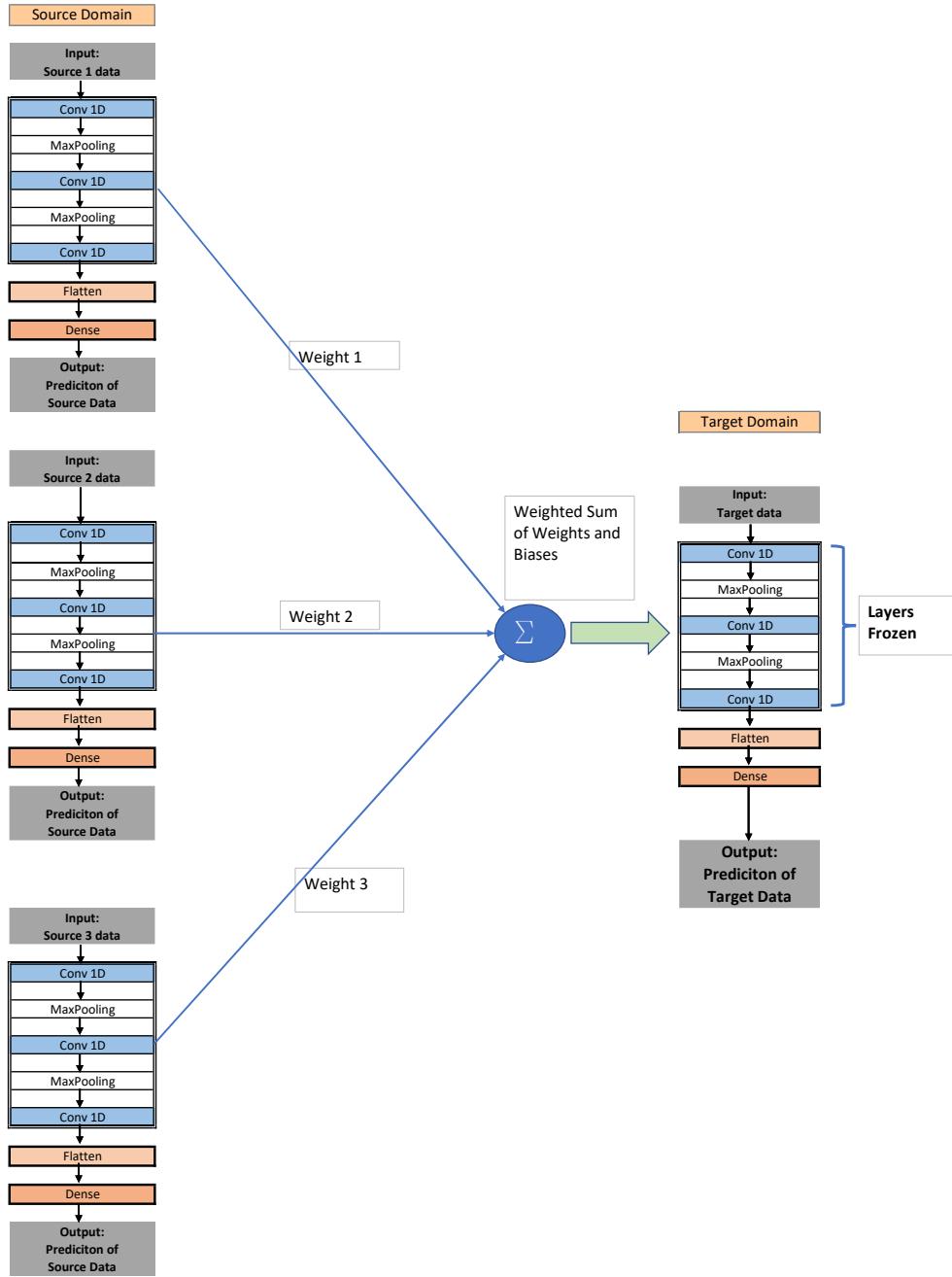


Figure 4.7: Multi-source Multi-Layer TL Network (MSML-TL)

transfer ability of the system. In addition, RFE helps reduce the number of features of the datasets and thus aids in minimising the complexity of the entire model. As usual, the weights from each of the layers of the trained models from different sources are added using

**4. A MULTI-SOURCE MULTI-LAYER-BASED TRANSFER LEARNING APPROACH FOR FORECASTING CUSTOMER DEMANDS OF NEWLY LAUNCHED PRODUCTS**

---

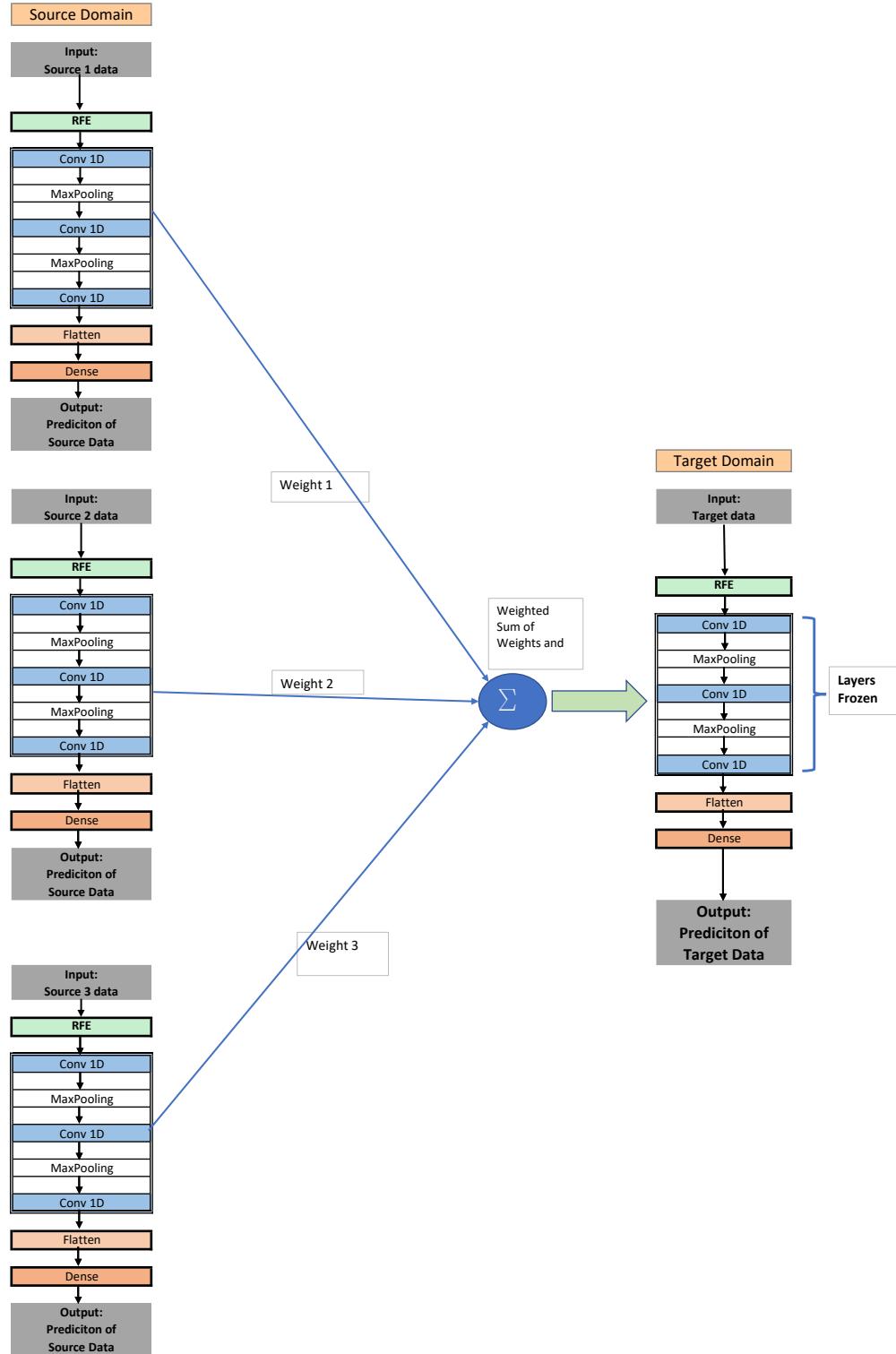


Figure 4.8: Multi-source Multi-Layer TL Network with RFE (MSML-TL-RFE)

the weighted average method to create the initial frozen layers of the new model. The frozen layers with other trainable layers are retrained together using target data and this approach enhances the information transfer ability of the system with increased prediction accuracy in the target domain. The combined method of extracting important features from multiple similar sources using RFE and feeding the information into a Multi-source Multi-Layer TL framework is presented as MSML-TL-RFE in this chapter.

#### **4.4.3.5 Pseudo-code of the proposed MSML-TL-RFE**

The proposed MSML-TL-RFE technique is further explained by a pseudo-code in Algorithm 3.

The pseudo-code in Algorithm 3 represents a TL workflow for predicting values of a target domain test set and evaluating with root mean squared error (RMSE). The workflow includes the following steps: first, the KNN algorithm is applied to identify three similar items to the target domain item based on Euclidean distances. Next, RFE is used to reduce the number of features for the target item and similar products. The data is then split into training, validation, and test sets. Independent convolutional neural network (CNN) models are trained for each similar item, and their weights and biases are combined to form a new CNN network using weighted averaging based on the Euclidean distances. Finally, the new CNN network is retrained using the target domain test data, and the RMSE is calculated using the predicted values and the true values of the target test set.

## **4.5 Model Implementation**

This section presents the implementation of the proposed MSML-TL-RFE approach for the analysis of supply chain data. The implementation is divided into several stages, starting with data collection, followed by feature extraction, data structuring, experimentation design, and finally, the evaluation of the hierarchical supply chain model. Each stage is described in detail in its respective subsection, providing a step-by-step guide for the

---

*4. A MULTI-SOURCE MULTI-LAYER-BASED TRANSFER LEARNING APPROACH FOR FORECASTING CUSTOMER DEMANDS OF NEWLY LAUNCHED PRODUCTS*

---

---

**Algorithm 3** Proposed MSML-TL-RFE approach

---

**Input:** Dataset with available and extracted features, Target Domain Item as per assumption

**Output:** Predicted Values of the target domain test set and the test RMSE

```
// KNN to identify three similar items with corresponding Euclidean Distances
1: S1, S2, S3 = KNN(Dataset - Target Domain Item, Target Domain Item)
2: distance1 = euclidean_distance(Target Domain Item, S1)
3: distance2 = euclidean_distance(Target Domain Item, S2)
4: distance3 = euclidean_distance(Target Domain Item, S3) // RFE to reduce the number of features for target
   item and similar products
5: Target_RFE, S1_RFE, S2_RFE, S3_RFE = RFE(Target Domain Item, S1, S2, S3) // Split Data
6: train_data, val_data, test_data = split_data(Target_RFE, S1_RFE, S2_RFE, S3_RFE) // Train indepen-
   dent CNN networks for each similar item
7: CNN1 = train_CNN(S1_RFE_train,S1_RFE_val)
8: CNN2 = train_CNN(S2_RFE_train,S2_RFE_val)
9: CNN3 = train_CNN(S3_RFE_train,S3_RFE_val) // Combine weights and biases from trained networks to
   form a new CNN network
10: weights1, biases1 = CNN1.get_weights()
11: weights2, biases2 = CNN2.get_weights()
12: weights3, biases3 = CNN3.get_weights()
13: sum_distance = distance1 + distance2 + distance3
14: weights_avg = [distance1/sum_distance, distance2/sum_distance, distance3/sum_distance]
15: new_weights = [weights1*weights_avg[0] + weights2*weights_avg[1] + weights3*weights_avg[2]]
16: new_biases = [biases1*weights_avg[0] + biases2*weights_avg[1] + biases3*weights_avg[2]]
17: new_CNN = CNN_model()
18: new_CNN.set_weights(new_weights, new_biases) // Train the new CNN network using the target domain
   test data
19: train(new_CNN, Target_train_data, Target_val_data)
20: test_predictions = predict(new_CNN, Target_test_data)
21: test_RMSE = RMSE(test_predictions, Target_true_values)
```

---

implementation of the MSML-TL-RFE approach. Overall, this section provides a complete guide to the implementation of the MSML-TL-RFE approach, from data collection to final evaluation.

#### 4.5.1 Data Collection

Before starting to apply input data to the various TL frameworks, it was essential to acquire data from acceptable sources to ensure the credibility of the proposed frameworks. Three very common and publicly available datasets were used for the experimentation

#### *4. A MULTI-SOURCE MULTI-LAYER-BASED TRANSFER LEARNING APPROACH FOR FORECASTING CUSTOMER DEMANDS OF NEWLY LAUNCHED PRODUCTS*

---

purposes of this chapter. These datasets are also used by many prominent researchers for their studies. The 1st dataset (Dataset 1) [Joseph et al., 2022, Kharfan et al., 2021] was taken from Kaggle ([link:`https://www.kaggle.com/c/demand-forecasting-kernels-only/data?select=train.csv`](https://www.kaggle.com/c/demand-forecasting-kernels-only/data?select=train.csv)) and it is known as the Store Item Demand Challenge. It contains five years (2013-2017) of sales data on 50 items sold over 10 different stores. The aim of the challenge was to make 3 months of predictions and it has the following attributes: date of sales, Store ID, Item ID and the number of items sold at a given date. In total, the dataset contains 9130000 features.

This chapter uses only the information of the first 3 stores and their corresponding 10 sold items in each of those three stores. For the initial part of the experiment, the 1st 9 items from each store were considered source products, while the 10th item (i.e., Item X) is the target domain data.

The second dataset (Dataset 2) was collected from a chapter [Mancuso et al., 2021] where the authors used an ML approach, termed as Neural Network Disaggregation (NND), which was developed by combining MLP with CNN for making hierarchical time series forecasting. This particular dataset contains 118 daily time series, which represents the demand for pasta from 01/01/2014 to 31/12/2018 ([link: `https://data.mendeley.com/datasets/njdkntcpc9/1`](https://data.mendeley.com/datasets/njdkntcpc9/1)). Apart from being univariate time series data, promotional activities are also linked with the sales of pasta in the dataset.

This chapter also uses a third dataset (Dataset 3), commonly known as Rossmann Store Sales data [Pavlyshenko, 2019, 2022], which is also available in the Kaggle competition platform ([link: `https://www.kaggle.com/code/mithilesh/time-series-analysis-and-forecasts-with-prophet/data`](https://www.kaggle.com/code/mithilesh/time-series-analysis-and-forecasts-with-prophet/data)). The daily sales of different Rossmann stores are described in this dataset. The data also has some main features such as store type, holiday, and Store Type.

Thus, in this chapter, as many as three datasets were used to carry out the experimental analysis. All the downloaded datasets were separated into two parts: source data and target data, according to the previously mentioned assumptions in Section 4.3. One par-

ticular product from each channel is assumed to be the newly launched product, which makes it the target domain data with limited data (1 month of data). On the other hand, the remaining other products or stores are all considered potential sources of information. The source data was then split into three parts, mainly for training, validation and testing the network. The target dataset was also split in the same way as the source data, but with less training and validation data, while the testing data had 6 months of sales information, which was later used to verify the accuracy of the networks.

Table 4.2: Dataset 1: Duration of Source/Target Domain Data for Training/Validation and Testing the TL Networks

Source Domain	Duration	Target Domain	Duration
TRAIN 1/1/2013- 31/12/2016	3 Years		
Validation 1/1/2017- 30/6/2017	6 Months	TRAIN 1/6/2017- 15/6/2017	15 Days
		Validation 16/6/2017- 30/6/2017	15 Days
Test 1/7/2017- 31/12/2017	6 Months	Test 1/7/2017- 31/12/2017	6 Months

Table 4.2 describes the duration of data from Dataset 1, that was used to obtain the pre-trained network in the source domain and also the duration of data used in the target domain to train, validate and test the TL approaches. This was explicitly done for Dataset 1 for explanation purposes and the other two datasets were also split in a similar way.

Instead of mentioning the duration, all three datasets are represented in a number of time-steps and their proportions in Table 4.3. In the source domain, a majority portion of data (around 80%) is used to train the network, while 10% is kept for validation and the remaining 10% is used for testing. Whereas in the target domain, a very small portion of the dataset is used to re-train and validate (around 6%-7% in each task) a pre-trained network, while a significant part (around 85% data) is used for testing the performance of

---

*4. A MULTI-SOURCE MULTI-LAYER-BASED TRANSFER LEARNING APPROACH FOR FORECASTING CUSTOMER DEMANDS OF NEWLY LAUNCHED PRODUCTS*

---

Table 4.3: Time-Steps of Source/Target Domain Data for Training/Validation and Testing the TL Networks

	Source Domain		Target Domain	
Dataset 1	No. of Time Steps	Percentage	No. of Time Steps	Percentage
TRAIN	1461	80.01%	15	6.98%
Valid	181	9.91%	15	6.98%
Test	184	10.08%	185	86.05%
Dataset 2				
Train	1443	80.26%	14	6.73%
Valid	176	9.79%	15	7.21%
Test	179	9.96%	179	86.06%
Dataset 3				
Train	577	61.25%	16	7.55%
Valid	184	19.53%	15	7.08%
Test	181	19.21%	181	85.38%

the TL approach.

#### 4.5.2 Feature Extraction

It is essential to extract features from data so that ML/DL has enough information for training. In order to train and re-train the ML/DL models, important features, such as year, month, week and day were extracted. Using python's built-in function `to_datetime`, year, month, week, and day were acquired, while if other information was available from the data, it was appended. Among the three datasets, not all of them have the same set of additional information, the 1st dataset has no information except the sales of stores, while the 2nd set has promotional information. The third dataset has other store-related information, such as promotional activities, holidays, whether a store is an open or closed, and customer-related information. All this additional information is appended as features alongside the previously mentioned extracted `_datetime` features. Thus the data frame contains at least 4 features for the 1st dataset, 5 features for the 2nd dataset and as many as 8 features for the 3rd dataset, as shown in Figure 4.9, with sales of a particular store item being the target variable.

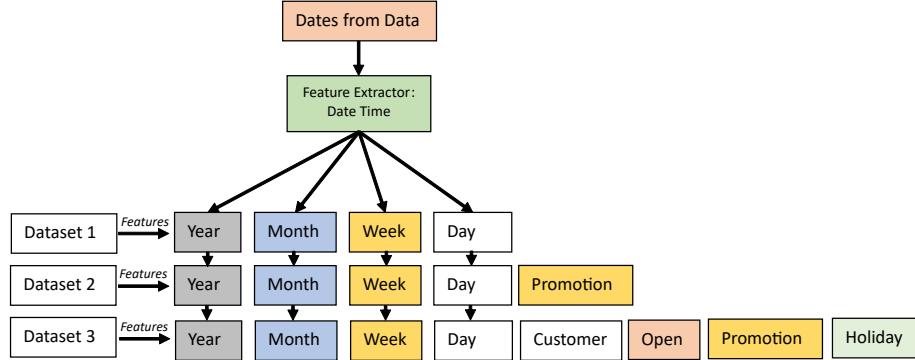


Figure 4.9: Extraction of Features From Data using Date-Time Feature Extraction Method

#### 4.5.3 Data Structuring

A min-max scalar normalisation function [Géron, 2019] was applied to scale the features of the data between 0 and 1. This was done so that it can easily be handled if there are any sigmoid activation functions in the output layer of any neural network. Therefore the normalization function was calculated based on the following formula in equation 4.5:

$$x_{norm} = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (4.5)$$

In the next stage, each scaled data frame was structured in a sequential pattern. For multivariate predictions, sales (dependent variable) are based on other independent variables (year, month, week, day, trend and seasonality). Independent variables are termed X and the dependent variables are y. For each data frame, the data is structured in a 2D array consisting of Independent and dependent variables, according to Table 4.4.

In this way, three different segments were selected from the data frame for training, validation and testing, using datasets from both the source and target domains. CNN, being a part of neural networks and also being the baseline algorithm used in this chapter, X\_train, X\_val and X\_test are converted to suitable 3D arrays using a sliding window approach [Selvin et al., 2017]. For each of the subsets of the data frame, the time series analysis of each subset is structured in a sequential pattern using a sliding window tech-

#### 4. A MULTI-SOURCE MULTI-LAYER-BASED TRANSFER LEARNING APPROACH FOR FORECASTING CUSTOMER DEMANDS OF NEWLY LAUNCHED PRODUCTS

---

Table 4.4: Data Structuring

Independent Variables		Target Variable
Variable X1	Variable X2	y
$X1_{t-n-days\_ahead\_prediction}$	$X2_{t-n-days\_ahead\_prediction}$	$y_{t-n+days\_ahead\_prediction}$
...	...	...
...	...	...
$X1_{t-2-days\_ahead\_prediction}$	$X2_{t-2-days\_ahead\_prediction}$	$y_{t-2+days\_ahead\_prediction}$
$X1_{t-1-days\_ahead\_prediction}$	$X2_{t-1-days\_ahead\_prediction}$	$y_{t-1+days\_ahead\_prediction}$
$X1_{t-days\_ahead\_prediction}$	$X2_{t-days\_ahead\_prediction}$	$y_{t+days\_ahead\_prediction}$

where  $t$ : current time;  $n$ : number of past time steps;  $days\_ahead\_prediction$ : forecasting horizon

nique before feeding it into the network. In the experiments, a window of size 10 (working days of two consecutive weeks) is used to predict one-day ahead (short term) and 5-day ahead (mid-term) sales information.

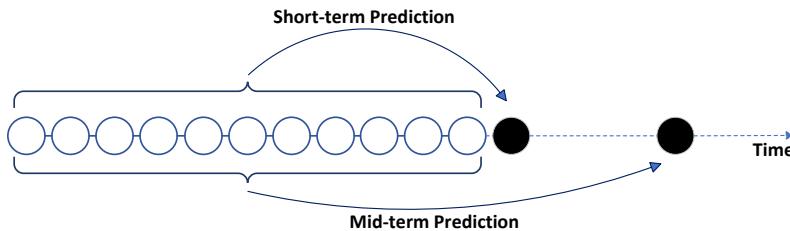


Figure 4.10: Illustration of short-term and mid-term prediction

Figure 4.10 illustrates the predictors (input values) and the target (output) while making short-term and mid-term predictions. An input vector of 10-time steps and numerous columns of data features is formed. The columns consist of the date time features, as well as other available features of the data, such as a holiday or promotional activities. Using these columns of information, data is structured as the independent variables (X train) (10 days of features and past sales information) and the dependent variables (Y train), which is either the 11<sup>th</sup> day's (short term) true sales or 15<sup>th</sup> day's (mid-term) sales, depending on the type of predictions the model is making. The window is then shifted one step along, and this time it contains information from the 2<sup>nd</sup> time step until the 11<sup>th</sup> time step of the training data and the 12<sup>th</sup> day's data as the output of that particular segment while making short term predictions. The model can also be used to predict anytime between 1 and 5 days ahead into the future. Since it is assumed that there is limited data available in the target domain, hence, as a limitation, in this chapter only 5 days ahead of prediction

is made using this system.

#### **4.5.4 Experimental Design for TL Approaches**

Sales of the target items were initially predicted using the simple base CNN model mentioned in 4.4.1. In this approach, No-TL was used for predicting the sales pattern of the assumed target item. From there, the system was customized to the SS-TL method. A single similar item was identified based on one month of sales pattern of the target data and then a matching item was found during that duration from other products for the same retailer. For more accurate prediction, the system was further upgraded to be a multi-source TL network where multiple items (at least 3 or more) were identified from the same retailer in order to obtain the pre-trained models and then was re-trained and the model was tested with target data. It was later extended to simulate the benefits of information sharing in a SC network. The items from other retailers were also considered in the TL approaches.

##### **4.5.4.1 Identification of Similar Products using KNN**

K-nearest neighbour (KNN) is a very popular approach often adopted by researchers to determine similar objects [Tao et al., 2022, Kużelewska, 2022]. To quantify the similarity between different items or stores, the distance between them was calculated. In this chapter, Euclidean distance was calculated for finding out the similarity between two items. Using this, items that have similar sales patterns are often closely grouped together at a smaller distance, whereas dissimilar items are situated further apart. All the available features of each dataset were used with the KNN algorithm for the determination of closely related items.

From each store, the 1st 10 items were chosen, while it was assumed that the 10th item was the targeted newly introduced market product. This made the other 9 items, with their extracted and available features, potential sources of information to compile a KNN

network for finding the most appropriate information source. It is worth mentioning that, considering the practicality of the situation, it was assumed that the target product had been in the market for only 30 days, which means that there was limited sales data available for analysis. Using that 30 days' worth of sales data in the target domain, the corresponding sales data of other items were extracted during that period. Thus KNN calculates the distance between items having 30 days of data and among which only the top 3 closest items are considered for further analysis using the TL approaches.

#### **4.5.4.2 Prediction of Newly Launched Item**

For the 1st two datasets, it was assumed that the 10th item from each retailer/brand was the newly launched product while for the 3rd dataset, it was presumed that in a certain region, the 10th store had recently started its operation with nine other stores in the same region. There were also other groups of stores carrying out operations in other regions of a particular country. To establish the effectiveness of the TL approaches, the 1st 10 items, related to one particular store from the 1st two datasets, were selected and from the 3rd dataset, the sales patterns of the 1st 10 stores were chosen for analysis. As mentioned previously, the 10th item or the sales of Store 10 or Item 10 were chosen as the new market being introduced, while it was assumed that other items or stores were in operation for quite some previous time in the market, which means the entire duration of the datasets were considered as potential sources.

For Datasets 1 and 2, initially without any TL approach (No-TL), where a base CNN approach was used to predict six months of sales from the target Item 10 (refer to Figure 4.1 Item X) of Store 1 or Brand 1 only. For Dataset 3, the sales of 6 months of sales of Store 10 were forecasted. Then the TL approaches are introduced gradually. For the TL systems to come into effect, it was essential to identify similar source data from the remaining 9 items or store sales. For the identification of closely related items or stores, KNN was used with all the extracted available features of data mentioned in Figure 4.9. Then according to the distance, the 1st three items were selected from each dataset, especially for Channel

***4. A MULTI-SOURCE MULTI-LAYER-BASED TRANSFER LEARNING APPROACH FOR FORECASTING CUSTOMER DEMANDS OF NEWLY LAUNCHED PRODUCTS***

---

1. The distance and weights of each of the datasets in relation to the target item or brand or store, are presented in Table 4.5.

Table 4.5: Euclidean Distance and Weights of Closest Three Items/Stores in Relation to Channel 1 Target Item/Stores for 3 datasets

	Source Domain Items/Store		
	Source 1	Source 2	Source 3
Dataset 1	Store 1 Item 7	Store 1 Item 8	Store 1 Item 2
	Distance	101.58	103.50
Inverse Distance Weights	0.3493	0.3428	0.3078
	Dataset 2	Brand 1 Item 4	Brand 1 Item 6
Distance	24.98	26.85	26.85
Inverse Distance Weights	0.3496	0.3252	0.3252
Dataset 3	Store 6 Sales	Store 2 Sales	Store 1 Sales
Distance	4901.97	5767.16	5895.77
Inverse Distance Weights	0.3729	0.3170	0.3101

From Table 4.5, it can be seen that for Dataset 1, Items 7, 8 and 2 are the items from Store 1 that are the three closest to Item 10 of the same store. The distance of Item 7 from Item 10 of store 1 is 101.58 while Item 8 is the next nearest to Item 10 with a separation of 103.50 units. Item 2 is the next closest item with a distance of 115.28 units. The inverse weights are as follows: Item 7 has a weight of 0.3793 while Item 8 has 0.3428 and the last Item 2 has 0.3078. The sum of all the weights is equal to 1. With this Inverse Distance method, the closest item with minimum distance gets the highest weight and the items with higher distance gradually have lower weights assigned.

The 2nd dataset (Dataset 2) is also similar to Dataset 1 but contains a bit of extra information, such as promotional activities. Here the KNN algorithm determines the three similar products belonging to Brand 1 that is closest to the target product (Item 10

of Brand 1). The closest Items are Items 4, 6 and 8 having distances of 24.98, 26.85 and 26.85 respectively while the corresponding weights are 0.3495, 0.3252 and 0.3252. The 3rd dataset is a little bit different with only the overall sales information of each store, rather than itemised sales pattern. Assuming the fact that the 1st 10 stores (Store 1 to store 10) are geographically located in the same region (Region 1), while the next 10 stores (Store 11 to store 20) are located in a different region (Region 2) and from Store 21 to Store 30 are situated in a 3rd region (Region 3). Considering the 1st 10 stores of Region 1, Store 10 was the assumed target store with Stores 6, 2 and 1 being the closest source stores as per the KNN algorithm. The associates' distances and weights are also stated accordingly in Table 4.5. The identified sources were then used in the TL approaches for the prediction of Target domain data.

#### **4.5.4.3 Simulated SC Scenario 1: Without information Sharing**

In the context of supply chains, there is a usual assumption among retailers that they want to keep their business information secret. Keeping that in mind, a retailer can use its own products to find similar source products and then use the TL approach to make predictions for items that are newly introduced by that retailer. An outline of the previously mentioned TL approaches explored in this chapter is presented in Figure 4.11.

According to Figure 4.11, the source and target data all belong to Store 1 of Dataset 1 due to the consideration of the fact that Stores do not wish to publicise their sales information to other retail outlets. In this case, while training the single source approach, only the best or the closed source data is considered. For the remaining three multi-source TL approaches, all three are considered and the ultimate effectiveness of each of these approaches is evaluated using the test set of the target domain data which is 6 months of Sales data of Store 1 Item 10.

With the remaining two datasets (Datasets 2 and 3), the same approach was used but with different source data for training the different networks. The target data is also changed accordingly, as for Dataset 2, the target data is Item 10 of Brand 1 and for Dataset 3,

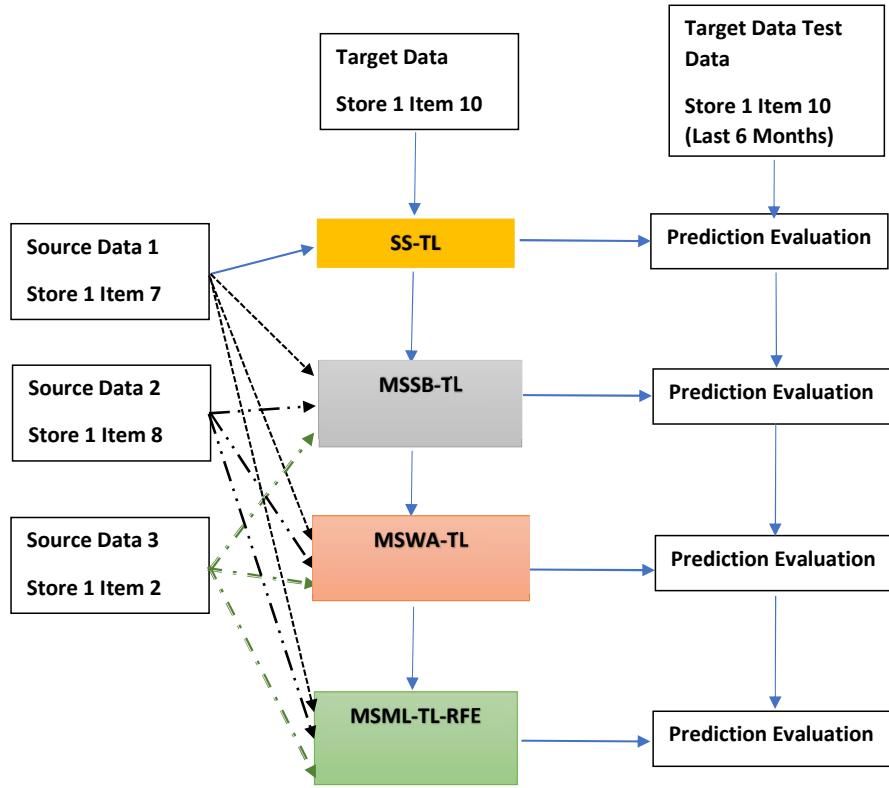


Figure 4.11: Prediction of Newly Launched Without Info Sharing Using Different TL Approaches

it is the Sales of Store 10 which is considered as the Target data. Likewise the source domain data is also changed while dealing with Dataset 3. Please refer to Table 4.5 for the source data of each of the datasets. All the experimental analyses were carried out to establish that obtaining knowledge from other sources enhances the TL capability over systems without the implementation of the information-sharing concept.

#### **4.5.4.4 Simulated SC Scenario 2: With information Sharing**

The same three datasets were used to simulate the information-sharing concept. For demonstration purposes, the target data of each of the datasets was the same while the source data were now selected based on a larger pool of source data. This was intentionally done to establish whether choosing more similar sources leads to a more accurate

---

*4. A MULTI-SOURCE MULTI-LAYER-BASED TRANSFER LEARNING APPROACH FOR FORECASTING CUSTOMER DEMANDS OF NEWLY LAUNCHED PRODUCTS*

---

TL prediction system and also enhances the overall effectiveness of TL approaches when information was shared among stakeholders. The source data with distance and weights of the different source data are presented in Table 4.6.

Table 4.6: Euclidean Distance and Weights of Closest Three Items/Stores in Relation to Target Item/Stores for 3 datasets With Information Sharing

		Source Domain Items/Store		
		Source 1	Source 2	Source 3
Dataset 1	Store 3	Store 2	Store 3	
	Item 6	Item 9	Item 7	
Distance	63.98	68.16	70.94	
Inverse Distance Weights	0.3520	0.3305	0.3175	
Dataset 2	Brand 1	Brand 2	Brand 3	
	Item 4	Item 3	Item 2	
Distance	24.98	25.98	26.00	
Inverse Distance Weights	0.3422	0.3290	0.3288	
Dataset 3	Store 23 Sales	Store 14 Sales	Store 6 Sales	
	3723.62	3978.98	4901.97	
Inverse Distance Weights	0.3710	0.3472	0.2818	

The concept of information sharing among stakeholders is simulated in this section. Assuming that competitors share data with each other creates a larger pool of data to identify closer sources. From Table 4.6 it can be seen that three closer sources are identified for the same target domain data for each of the datasets. For Dataset 1, the closest item to the target item (Store 1 Item 10) is Item 6 of Store 3. The next closest item is Item 9 from Store 2 and the third closest item is Item 7 from Store 3. The distance and weights are assigned according to the Euclidean Distance from KNN. For Dataset 2, the closest items are not confined to a particular brand but now similar items from other brands are also considered potential sources of information. In the case of Dataset 3, Stores from another region can also be used to extract important information to train models.

#### **4.5.5 Effectiveness of the TL Predictions in the context of a Hierarchical SC Model**

The predictions from different TL approaches were practically considered by developing a Hierarchical SC Model. The model calculates costs for a newly launched item along each Channel (Refer to Figure 4.1 for more information). The presented architecture has 3 channels for each of the datasets. For the 1st dataset (Dataset 1) The target domain data for Channel 1 is Store 1 Item 10, for Channel 2 is Store 2 Item 10 and for the 3rd channel (Channel 3) it is Item 10 of Store 3. Similarly, it can be deployed over the other two datasets as well for cost calculation, which can be implemented in the future to observe the cost of new business using different channels. For Dataset 2, over three channels the probable three Target domain data are as follows: Channel 1: Brand 1 Item 10, Channel 2: Brand 2 Item 10 and Channel 3: Brand 3 Item 10. In the last dataset (Dataset 3) the target domain probable stores for the future calculation can be Store 10, Store 20 and Store 30 as Channel 1, 2 and 3 respectively. Considering the target Items of each channel, the Source Items are identified accordingly using KNN. For the time being, only Dataset 1 is taken into consideration for such cost estimation over different channels.

### **4.6 Result Analyses**

This section provides a comprehensive analysis of the results obtained from applying the proposed MSML-TL-RFE approach to three datasets. The purpose of this analysis is to validate the suitability of the MSML-TL-RFE approach for the analysis of supply chain data. In the following subsections, the results obtained from the application of the MSML-TL-RFE approach are presented in detail. This is followed by a behavioural analysis and theoretical derivation of the findings, which will shed light on the underlying mechanisms and relationships in the data. Finally, managerial implications are discussed based on the results, providing practical guidance for decision-makers in the supply chain domain. Overall this section presents a thorough and comprehensive evaluation of the MSML-TL-RFE approach and its potential for use in the supply chain domain.

---

#### *4. A MULTI-SOURCE MULTI-LAYER-BASED TRANSFER LEARNING APPROACH FOR FORECASTING CUSTOMER DEMANDS OF NEWLY LAUNCHED PRODUCTS*

---

##### **4.6.1 Result Analyses using Online Datasets**

As many as 4 different TL approaches are explored in this chapter, over three datasets, demonstrating two SC scenarios. Limited target data was presumed for predictions, which led to adapting knowledge from other domains. For the simplicity of the analysis and to show the potency of the TL approaches, target domain data was initially limited to one particular channel for all three datasets. This was later extended to three channels other for more analysis on using an SC model. The average RMSE from each of the approaches over individual datasets, demonstrating the value of information sharing, is presented in Table 4.7.

Table 4.7: Average RMSE for Different Approaches using 3 Datasets, with and without Information Sharing

Approaches	No-TL	SS-TL		MSWA-TL		MSSB-TL		MSML-TL-RFE	
		Without Info-Sharing	With Info-Sharing	Without Info-Sharing	With Info-Sharing	Without Info-Sharing	With Info-Sharing	Without Info-Sharing	With Info-Sharing
Dataset 1	0.2067	0.1938	0.1842	0.2035	0.1932	0.1976	0.1901	0.1763	0.1736
Dataset 2	0.1049	0.1120	0.1120	0.1081	0.1104	0.1119	0.1120	0.1099	0.1075
Dataset 3	0.2833	0.3145	0.3070	0.3018	0.3027	0.2895	0.2891	0.2873	0.2856

Based on the data in the table 4.7, a comparison between the different approaches used in the experiments for each of the three datasets is also shown in Figure 4.12. It also highlights the impact of information sharing on the performance of the approaches.

To evaluate the performance of each of the models, 1 to 5 days ahead of predictions was obtained from each individual model. For each day ahead of prediction, multiple different experiments were performed and the mean predicted value was calculated. The predicted mean value was then compared against the true target domain test data and the corresponding RMSE was calculated for performance evaluation. From Table 4.7, it can be seen that in the majority of the datasets, information sharing improves the accuracy of the cases. Moreover, the average RMSE for SS-TL for both without and with information sharing (0.1938 and 0.1842 respectively) is lower than the No-TL approach, especially for Dataset 1 which is 0.2067. For the other two datasets, SS-TL has comparable results with the No-TL system, even though the RMSE of both No-TL and SS-TL approaches are slightly on the higher side. This may be due to the fact that Datasets 2 and 3 have

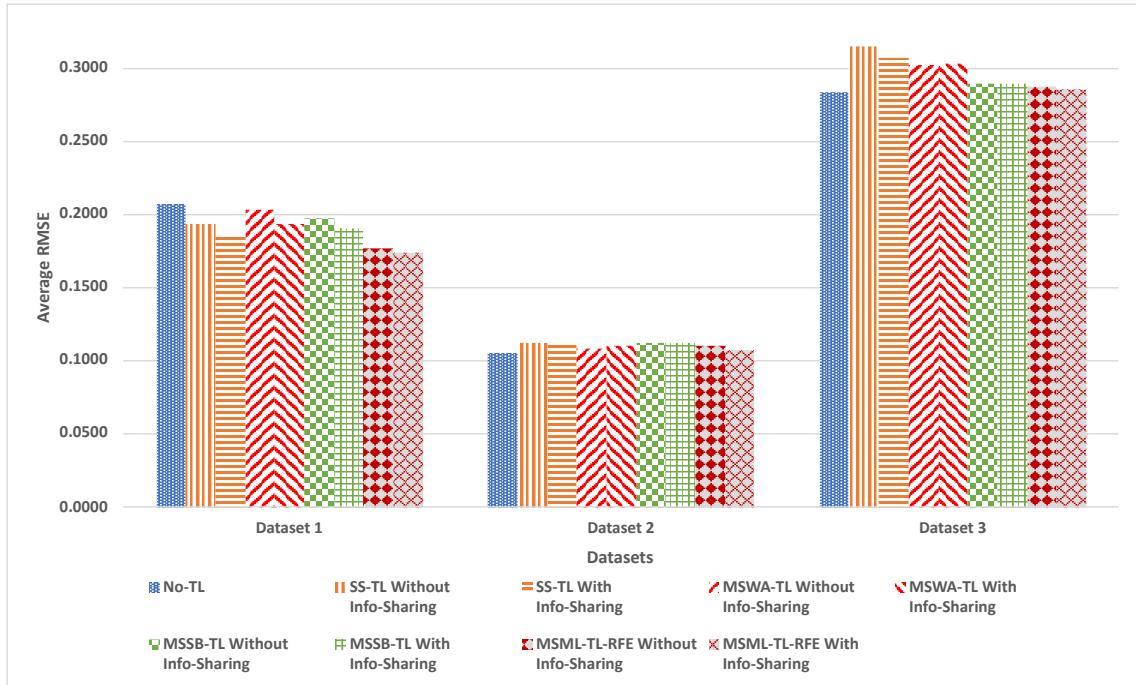


Figure 4.12: Average RMSE for Different Approaches using 3 Datasets, with and without Information Sharing

extreme changes in the data, which needs further future investigation. With the addition of multiple sources in the TL architectures, the average RMSE for each dataset gradually decreases. The overall effect can be observed from the average RMSE of the three datasets for all approaches, as presented in Table 4.8. Statistical tests were also carried out using the average RMSEs to demonstrate the significance of the TL approaches. In the next phase of the experimentation, the performance of the different TL approaches was investigated over two abstract SC scenarios.

Table 4.8: Average RMSE of Different Approaches, with and without Information Sharing

Forecasting Approaches With Limited Target Data	No-TL	SS-TL	MSWA-TL	MSSB-TL	MSML-TL-RFE	Mean RMSE
withOUT Info	0.1983	0.2068	0.2044	0.1996	0.1911	0.2005
WITH info	0.1983	0.2011	0.2021	0.1971	0.1889	0.1973

The information in Table 4.8 is used to visually represent the differences in average RMSE values for the different approaches, with and without information sharing which is presented in Figure 4.13. It enables comparison between the different datasets and helps

#### 4. A MULTI-SOURCE MULTI-LAYER-BASED TRANSFER LEARNING APPROACH FOR FORECASTING CUSTOMER DEMANDS OF NEWLY LAUNCHED PRODUCTS

---

identify trends or patterns in the accuracy of different approaches.

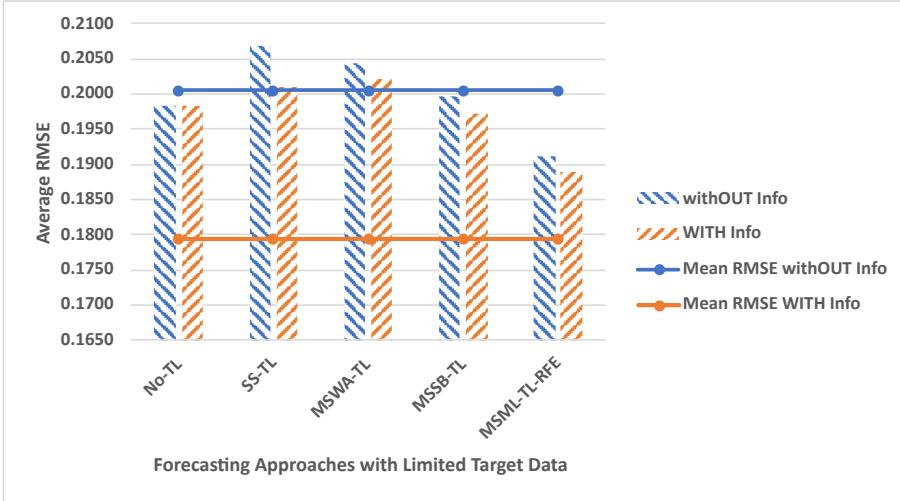


Figure 4.13: Average RMSE of Different Approaches, with and without Information Sharing

From Table 4.8 and Figure 4.13, it can be deduced that an average RMSE of 0.2068 is produced with the SS-TL system, which is the highest when no information has been shared by the retailers. Then, with the addition of numerous sources, the RMSE gradually decreases. The MSWA-TL approach produces an RMSE of 0.244 while the MSSB-TL approach is the next best in terms of an RMSE of 0.1996. The MSML-TL approach with RFE (MSML-TL-RFE) and Frozen Layers produces the lowest RMSE of 0.1911, which is the best among all approaches. The TL approaches are also compared against the non-TL approach which produced an RMSE of 0.1983.

When the stakeholders agree to share sales information with each other, closer sources can be identified for better knowledge harvesting. With more similar sources, all the previous 4 TL models are trained as before. The RMSE produced by each of the models is relatively lower than the previous set of RMSEs, where no information was shared. The SS-TL approach has an RMSE of 0.211, while the MSWA-TL technique has an RMSE of 0.2021. Both these two RMSEs are lower than their predecessors. The MSSB-TL and MSML-TL-RFE approaches also produced comparatively lower RMSE of 0.1971 and 0.1889, respectively. The same RMSE was reported with the No-TL approach as it has

no sharing of information. From observing the mean RMSE of the two SC scenarios, with and without information sharing, it is evident that sharing information among the retailers leads to more accurate predictions, regardless of the considered TL approach. The average RMSE without information sharing is 0.2005, while with sharing of information, it drops to 0.1973. Hence, sharing of information adds value to the knowledge acquired from closer sources, as it gives more freedom to the TL approaches to gain knowledge from a closer source.

#### **4.6.1.1 Statistical Analysis**

In this subsection, two statistical tests were conducted to establish the significance of the different TL approaches presented in this chapter. Using the RMSE obtained from the approaches, a Friedman Test was first carried out to find the rank, and then taking the best three approaches, a Wilcoxon Signed Rank Test was conducted to establish the significance of the TL frameworks. Both the two SC information-sharing concepts were taken into consideration and the tests were conducted on the overall RMSEs of each of the approaches on all 3 datasets. Table 4.9 shows the results of the mean rank of each method obtained from the Friedman Test.

Table 4.9: Friedman Test: Mean Rank

TL Approaches	Mean Rank: Combined dataset	Position (Combined Dataset)
No-TL	<b>2.93</b>	3
SS-TL	3.18	4
MSWA-TL	3.41	5
MSSB-TL	<b>2.88</b>	2
MSML-TL-RFE	<b>2.60</b>	1

It is evident from Table 4.9 that the best-performing TL Approach is the MSML-TL-RFE architecture which has the lowest mean rank of 2.60 (in bold). Apart from that, the next two best-performing approaches are MSSB-TL and No-TL which have mean rank values of 2.88 and 2.93. Taking the best three approaches into consideration, Wilcoxon Signed Rank Tests were carried out to find the significance of the TL systems. The results of the

---

*4. A MULTI-SOURCE MULTI-LAYER-BASED TRANSFER LEARNING APPROACH FOR FORECASTING CUSTOMER DEMANDS OF NEWLY LAUNCHED PRODUCTS*

---

Wilcoxon test are reported in Table 4.10.

Table 4.10: Results of the Wilcoxon test using RMSE-testing

TL Approaches	Better	Similar	Worst	p-value	Decision
MSSB-TL vs No-TL	150	0	150	0.38	$\approx$
MSML-TL-RFE vs No-TL	177	0	123	0.00	+
MSML-TL-RFE vs MSSB-TL	161	0	139	0.005	+

The Wilcoxon Signed Rank Test was conducted on the RMSEs of the best three approaches. Using a 5% significance level, one of three signs (+, – and  $\approx$ ) was assigned, where the “+” sign means the first algorithm is significantly better than the second one, “–” means that it is significantly worse and “ $\approx$ ” represents that there is no significant difference between these approaches. According to Table 4.10, the test between MSSB-TL and No-TL method is not statistically significant as it produced a p-value of 0.38. The other two tests conducted between the MSML-TL-RFE and No-TL Approaches, and with MSML-TL-RFE and MSSB-TL, produced p-values of 0.00 and 0.005 respectively which are much less than 0.05. Thus, the statistical test reveals that the proposed MSML-TL-RFE is statistically better against its two nearest competitive approaches when RMSE is the performance measure.

#### **4.6.2 Application of TL on the Designed Hierarchical SC Model**

From Subsection 4.6.1, it is quite clear that the best performing TL approach is the proposed MSML-TL-RFE system. The predictions from that best approach with predictions from two other approaches, namely the No-TL and SS-TL approaches, were considered for cost analysis while using the Hierarchical SC Model. Using these approaches, up to 5 days ahead of predictions were obtained for all three channels of one particular dataset (Dataset 1). It is worth noting that for each individual channel, the target domain item

or store was assumed as the newly launched product with limited data and then keeping that assumption fixed, similar items are identified using KNN as before. The predictions against each target domain data, for 3 different channels, were then fed into the SC system for cost analysis.

#### **4.6.2.1 Fixed Lead time with Variable Safety Stock**

In the SC model, two different aspects are considered. One with fixed or dynamic lead time, while the other is with variable safety stock. The safety stock is considered to vary in 4 steps: No safety stock, 95% to 99% and 95.9%. The analysis of 4 different safety stocks and the average cost is reported in Table 4.11 for a fixed lead time of 5 days.

Table 4.11: SC Cost (\$) for Fixed LT over 3 Channels using Dataset 1 with Information Sharing Concept

SC Cost (\$)	Channel 1 Fixed LT 5 days	Channel 2 Fixed LT 5 days	Channel 3 Fixed LT 5 days	Avg SC Cost
No-TL	32573.55	45940.51	40564.12	39692.73
SS-TL	32797.38	45823.47	40589.15	39736.67
MSML-TL-RFE	<b>32298.28</b>	<b>45180.57</b>	<b>39959.41</b>	<b>39146.09</b>

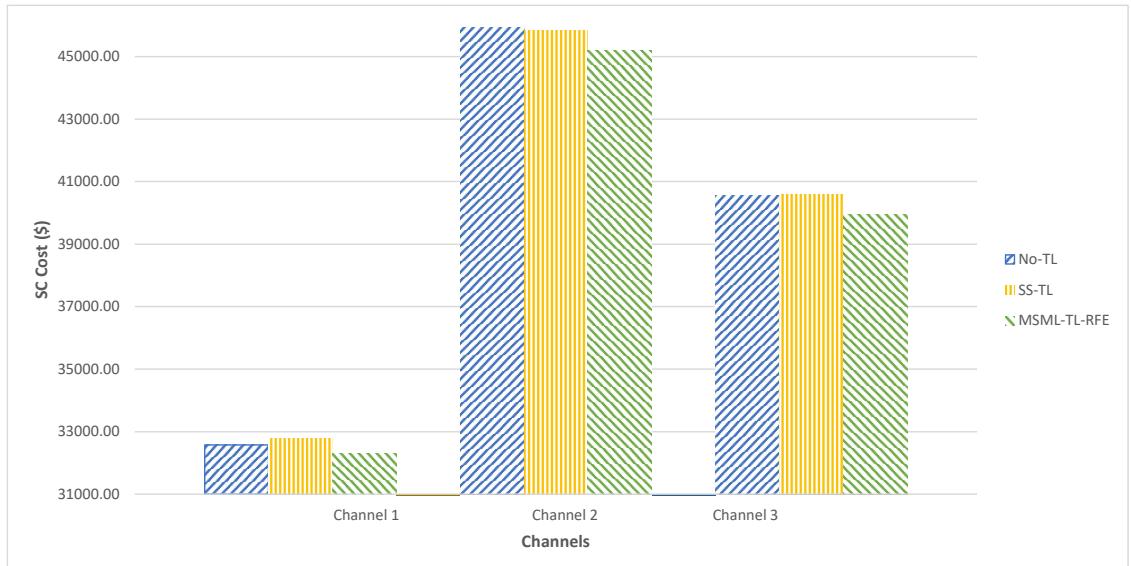


Figure 4.14: Supply Chain Cost (\$) over 3 Channels of SC having Fixed LT

#### *4. A MULTI-SOURCE MULTI-LAYER-BASED TRANSFER LEARNING APPROACH FOR FORECASTING CUSTOMER DEMANDS OF NEWLY LAUNCHED PRODUCTS*

---

From Table 4.11, it can be seen that the more accurate approach creates a lower SC cost over all three channels. As the SS-TL system is less accurate than the No-TL approach, it produces comparatively higher costs of \$32797.38 and \$40589.15 for Channel 1 and Channel 3 respectively, while it is slightly lower (\$45823.47) for Channel 2 i.e. Item 10 from Store 2 is considered as the target domain data. The MSML-TL-RFE method proposed in this chapter is the most accurate one and produces the lowest overall SC cost (presented in bold) of all three channels. The overall average SC cost for the MSML-TL-RFE approach is also the lowest (\$39146.09), if the average value is considered.

##### **4.6.2.2 Dynamic Lead time with Variable Safety Stock**

Changing the SC model to work in situations where the delivery lead time (LT) is dynamic (where it varied between 3 to 5 days), the same predictions from different models were fed into the SC models for cost calculations. This is often a more practical scenario, as because retailers and suppliers have some uncertainty in managing logistics. Thus such a type of cost calculation is done in the case of newly launched products with variable safety stock to minimise stock out and the result is presented in Table 4.12.

Table 4.12: SC Cost (\$) for Dynamic LT over 3 Channels using Dataset 1 with Information Sharing Concept

SC Cost (\$)	Channel 1 Dynamic LT 3+2 days	Channel 2 Dynamic LT 3+2 days	Channel 3 Dynamic LT 3+2 days	Avg Cost
No-TL	32945.41	46785.29	41037.82	40256.17
SS-TL	32738.96	46658.70	40844.37	40080.68
MSML-TL-RFE	<b>32692.13</b>	<b>46274.51</b>	<b>40762.72</b>	<b>39909.79</b>

Unlike Fixed LT, where the SC model considered all 5 days ahead of predicted data, here the model did not consider all 5 days of prediction every time, rather it considered only 3 days of LT and then there was a delay of an additional 1 to 2 days, mimicking the uncertainty in practical fields. It can be seen from the table that the SC model produces a lower cost for the SS-TL approach than the No-TL approach, because the SC system with safety stock in place, reduces the chance of stock out. Hence the average cost produced

by SS-TL over three channels of the dynamic SC model is \$40080.68, which is lower than the No-TL approach of \$40256.17. As before, the most accurate approach is MSML-TL-RFE and it consistently produces the lower SC cost (presented in bold), even in situations where the replenishment LT is dynamic. Its overall average SC cost of \$39909.79 is the lowest among all of the approaches presented in Table 4.12. Comparing Tables 4.11 and 4.12, the average cost produced by MSML-TL-RFE is relatively higher for dynamic LT, as the system has to incur some additional cost when the uncertainty in delivery time is considered, which is a more practical representation of a real system.

#### **4.6.3 Behavioural/Sensitivity Analysis**

From previous experimental analysis and statistical tests, it can be concluded that the MSML-TL-RFE approach having three sources with an information-sharing concept provides the best result. It was further tested by adding a few more sources. Since the best-performing TL system has the capability of learning from three sources, its sensitivity was further evaluated by the addition of a few more sources. The system was tested with 6 sources and also with 9 sources with the information-sharing concept enabled. The time needed and the accuracy of each of the approaches is reported in Table 4.13.

The experiment was also conducted with the No-TL and SS-TL approaches without sharing information as benchmark approaches for comparison. It can be seen from Table 4.13 that predictably learning from multiple sources takes a longer time as compared to learning from a single source. When it comes to accuracy, a larger number of sources slightly increases accuracy while taking a significant amount of extra time for the models to extract information from multiple sources. However when the number of sources is increased from 3 to 6, accuracy actually decreases, while increasing it further slightly improves accuracy, showing some indications of negative information transfer. However it takes significantly longer (3519 seconds) to train, while with 3 sources it takes only 1304 seconds with a more moderate accuracy of 5.668.

Another way of checking the sensitivity of the best TL approach (MSML-TL-RFE) is

**4. A MULTI-SOURCE MULTI-LAYER-BASED TRANSFER LEARNING APPROACH FOR FORECASTING CUSTOMER DEMANDS OF NEWLY LAUNCHED PRODUCTS**

---

Table 4.13: Time and Accuracy of Different Approaches for Dataset 1

	Time(s)	Accuracy (Reciprocal RMSE)
Expt 1 No-TL	49	4.8340
Expt 2 SS-TL w/o info sharing	458	5.1749
Expt 3 MSML-TL-RFE with info sharing having 3 Sources	1304	5.6668
Expt 4 MSML-TL-RFE with info sharing having 6 Sources	2487	5.5102
Expt 5 MSML-TL-RFE with info sharing having 9 Sources	3915	5.7538

by changing the type of pool of knowledge sources for transfer. In all of the previous experimentation, the 10th Item or the sales of the 10th store was assumed to be the target item for each channel, but in this section, the type of target item is treated as the determining factor to select potential sources from the pool of items or stores. Considering the same type of item or stores, information is now extracted from the same type or category of items sold or stores (including the 10th item of other channels) and the result of the analysis is presented in Table 4.14.

Table 4.14: Performance of MSML-TL-RFE Approach with Different Number of Similar Source Items from Same Group

	Without Info Sharing	With Info Sharing (3 Items/Stores from Same Group)	With Info Sharing (9 Items/Stores from Same Group)
Dataset 1	0.1763	0.1764	0.1722
Dataset 2	0.1099	0.1083	-
Dataset 3	0.2898	0.2814	0.2891
Mean RMSE	0.1920	0.1887	-

When no information is shared among the retailers, then at times it is difficult to find

items that are of a similar type, given the availability of limited information in Datasets 1 and 2. For Dataset 3, there are other criteria in it that were used to select the source items belonging to the type or category of different stores. In this section, for Dataset 3, source stores belonging to the same category as that of target Store 10 (which is Category 'a', as mentioned in Dataset 3) were identified to initiate the information transfer process. When the information sharing concept was taken into consideration, the 10<sup>th</sup> item of each store was selected as the source items for Dataset 1, since according to the dataset they have the same corresponding characteristics as Target Item 10, which means for Target Store 1 Item 10, the corresponding available sources are Store 2 Item 10, Store 3 Item 10, Store 4 Item 10, etc. Dataset 2 is about pasta of different brands and types. Item 10 of one brand of pasta corresponds to Item 10 of other brands (e.g. Brand 1 Item 10 corresponds to Item 10 of Brand 2 and Item 10 of Brand 3) and were considered as the same category items as they are of the same type. This means for Dataset 1, the same item 10 is being sold at different stores while for Dataset 2, it is the same pasta having different brands. Considering a real-life scenario, for Dataset 1, Item 10 may refer to the same jacket sold across different stores having different locations whereas, for Dataset 2, it could be pasta of different shapes and flavours such as spaghetti, rigatoni, or lasagne that can be produced by different food manufacturing companies, like Barilla, DeCecco or Rana. So if spaghetti is the target item, then only sales of spaghetti from different brand companies are considered for information transfer. For the 3rd dataset (Dataset 3), stores belonging to Category 'a' (since it matches with the sales pattern of the target Store 10) were taken into consideration while similar stores were identified for harvesting knowledge.

From Table 4.14, it can be seen that when trying to deal with items belonging to the same group or category with the assumption of information sharing, the MSML-TL-RFE approach produces lower RMSE than without information sharing and with 3 sources. When 9 same category sources are considered, there is inconclusive evidence of improvement as there is missing information for Dataset 2. In Dataset 2, only 4 Brand level information are available while it requires 9 Brand levels of information to carry out the experiment successfully, e.g. as discussed earlier, sales of spaghetti from four food manufacturing companies are available, but for successful experimentation, spaghetti sold by nine companies

are required. That is why some spaces are left blank (-). Thus in terms of average RMSE, if there is available information regarding same-category items, then those could be considered as more accurate sources of information, which could lead to better prediction in the target domain.

#### **4.6.4 Theoretical Derivation**

Considering the SC costs from Dataset 1, its corresponding accuracy and time for different approaches from Tables 4.11, 4.12 and 4.13, two practical suggestions can be made. SC costs from Tables 4.11 and 4.12 show that decision makers can select the appropriate channel and continue selling the recently launched product based on expected cost. In this case, for the same SC parameters, Channel 1 produces the lowest cost, which means that it is safe to carry out selling the new product, while Channel 3 produces a comparatively higher cost for the same product. Channel 2 is the most expensive and risky channel for business practitioners to carry out operations. Thus this approach can help managers to make an informed decision regarding the continuation of selling their newly launched product through different channels in the market, which is basically the 1st of the two theoretical contributions.

Apart from the SC costs, if the Time and Accuracy of different TL approaches, having a different number of sources of information presented in Table 4.13, are taken into consideration, then by the combination of all three factors together, another practical suggestion can be made. Figures 4.15 and 4.16 are plots of Cost vs Accuracy vs Time plotted using information from Tables 4.11, 4.12 and 4.13, respectively. Analysing Figure 4.13, it can be seen that the multi-source approaches were not limited to 3 sources, but further experimental analyses were conducted with a great number of sources with source number increasing to as many as 6 and 9.

From Figures 4.15 and 4.16, it is readily apparent that the No-TL approach produces higher cost with lower accuracy, but is fastest for both cases. As compared to the No-TL approach, SS-TL produces better accuracy and lower cost, while MSML-TL-RFE with

**4. A MULTI-SOURCE MULTI-LAYER-BASED TRANSFER LEARNING APPROACH FOR FORECASTING CUSTOMER DEMANDS OF NEWLY LAUNCHED PRODUCTS**

---

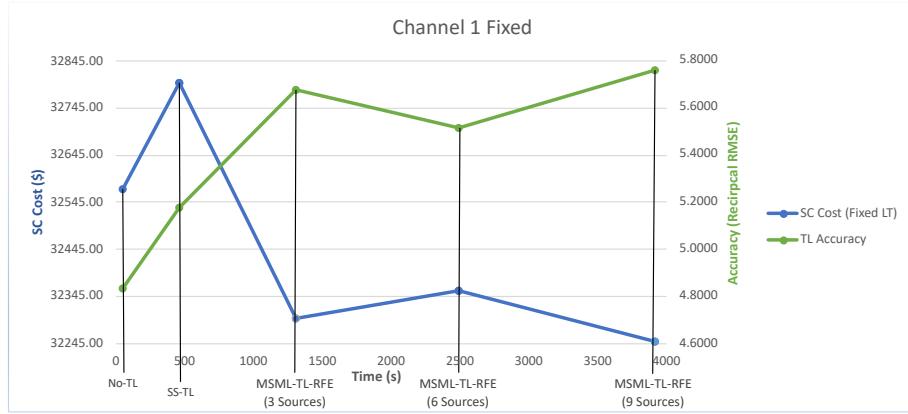


Figure 4.15: Supply Chain Cost (\$) vs Training Time vs Prediction Accuracy of Channel 1 of SC having Fixed LT

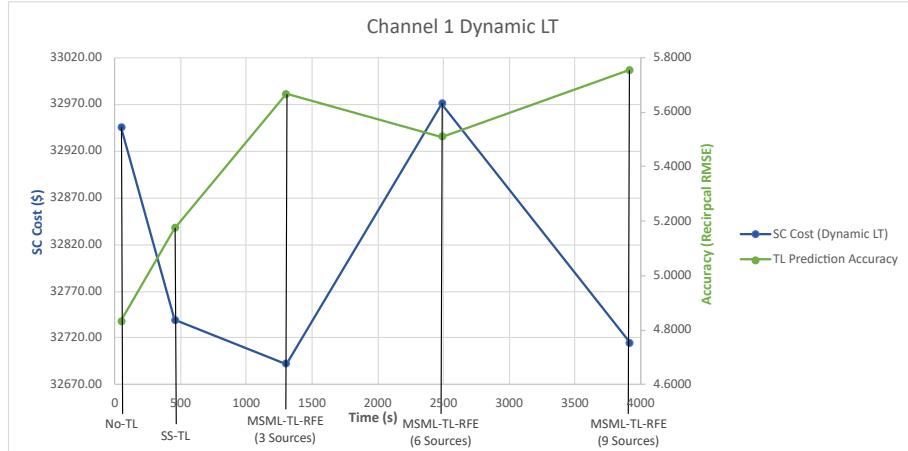


Figure 4.16: Supply Chain Cost (\$) vs Training Time vs Prediction Accuracy of Channel 1 of SC having Dynamic LT

3 sources produces significantly better accuracy with even lower cost, although it takes a considerable amount of time to produce results when compared to the previous two approaches. Increasing the source size to 6 and 9 slightly improves accuracy with lower supply chain cost, but also takes a significant amount of time to produce results, which may not be practical in some cases where it is difficult to collect that many sources for information transfer. Based on these findings, managers and practitioners can choose between different operating models. They can select the No-TL approach for faster but less accurate predictions or SS-TL for better predictions at the cost of extra time. If they have more time to process data, MSML-TL-RFE with 3 sources can provide much more

#### *4. A MULTI-SOURCE MULTI-LAYER-BASED TRANSFER LEARNING APPROACH FOR FORECASTING CUSTOMER DEMANDS OF NEWLY LAUNCHED PRODUCTS*

---

accurate predictions and lower supply chain costs. However, if they have a very large pool of data, they can identify more sources, which will take even more time to produce results with very little reduction in supply chain cost, as compared to selecting 3 sources for harvesting information.

##### **4.6.5 Managerial Implications**

The study related to sales predictions using AI with TL is especially beneficial for practitioners who often deal with limited data. Since the restricted data makes DL approaches less accurate, the TL approach can solve this problem by predicting short-term to mid-term fluctuations with better accuracy. This enables decision makers to see a better picture of how a newly launched product will be treated by customers in the future. They will be able to analyse the viability of a new business model with such cases. With the advances in AI with TL, they will have the advantage of calculating future expenses and adjusting their pricing policies and inventory management. They could also think of future investments with smaller risks while carrying out operations in multiple channels of an SC. Supply chain forecasting enhances operations by planning for unforeseen events, delays, and changes in demand. One may reduce costs, satisfy customer demand, and maintain the efficiency of a supply chain by leveraging data and other insights to make smarter decisions, even with little data in the target domain to be used to predict items. Organisations can make more profit by earning consumers' confidence in their newly launched item, and the overall productivity increases will ultimately create a positive impact on Gross Domestic Product (GDP).

### **4.7 Summary**

Limited data has always been a challenge for supply chain managers, especially when it comes to the decision-making process regarding whether to continue with a newly launched product or not. TL, a part of ML, can be used in such situations to predict the demand

---

*4. A MULTI-SOURCE MULTI-LAYER-BASED TRANSFER LEARNING APPROACH FOR FORECASTING CUSTOMER DEMANDS OF NEWLY LAUNCHED PRODUCTS*

---

for such a new product. This chapter harvests knowledge from similar products that have similar sales patterns while considering the short duration of the product in the market. Then using knowledge from similar sources, pre-trained models are re-trained in the target domain to make predictions about the newly launched product. Knowledge from a single source, as well as multiple sources have been investigated, and two different SC ideas were simulated, considering different information-sharing concepts. After extensively comparing the RMSEs of multiple TL approaches over 3 different datasets, and also through statistical tests, it has been shown that the MSML-TL-RFE approach was the best TL approach among five different approaches. Moreover, an SC model was used to demonstrate how the amount of cost for a business can change based on various approaches.

All approaches presented in this chapter, including the proposed multi-source multi-layer TL approach, were carried out with a fixed set of parameters after extensive parameter tuning. However, for each set of experiments, a grid search to find the best parameter setting could provide a better result, which could be investigated in the future. The addition of RFE to other TL approaches, such as LSTM, GRU or RNN-based TL approaches, along with the ones already mentioned in this chapter, could also be done to examine the performance of the TL models. Regarding the training time of the TL systems, a single model was trained with a single source at a time. This was also true for the multi-source approaches where each of the sources was selected sequentially to train a base model to obtain the pre-train models, which can be considered as a limitation of the proposed approach. In the future, if multiple models can be trained simultaneously, then the same accurate results could be obtained within a very short time.

# **Chapter 5**

## **Conclusion and Future Directions**

In this chapter, a brief overview of the research conducted in this thesis, along with its findings and conclusions, is presented. Additionally, potential avenues for future research are also outlined.

### **5.1 Summary of Accomplished Research**

The different approaches used in this thesis to predict multivariate time series data using DL are discussed in this chapter. The first approach proposed is a hybrid deep learning technique, termed as Poly-Linear Regression with Augmented Long Short Term Memory Neural Network (PLR-ALSTM-NN), which is designed to make more accurate long-term predictions. The approach was tested on various financial market datasets, and its performance was compared with other state-of-the-art approaches. The proposed technique is particularly useful for long-term financial investors who purchase stocks and sell them at a later date. After conducting extensive experimentation and comparing the performance of different DL approaches, it was found that different techniques produce different accuracies for different forecasting horizons, which is a kind of shortcoming for different DL approaches. To address this issue, an SBFA (Switching Based Forecasting Approach) ap-

---

## *5. CONCLUSION AND FUTURE DIRECTIONS*

proach was proposed, which is capable of making more accurate estimations of time series data for different forecasting horizons. The ensemble SBFA approach, containing different ML/DL approaches, was used to predict customer demand in various sales datasets. A supply chain model was developed to calculate the corresponding cost by comparing the predicted demand against the true demand. The benefit of a more accurate forecast is a reduction in supply chain costs.

However a shortage of data can be an issue when using DL techniques. According to the literature, DL performs well when there is a long chain of data, but it suffers when limited data is available for training. To address this issue, a TL approach was deployed to predict the demand for newly launched products, which is a common case for limited data. Available features in the dataset are used to identify similar source products using KNN. Once the sources are identified, only the critical features are chosen from each source using the RFE technique. These multiple sources with their important features are then used to pre-train a TL network called MSML-TL-RFE (Multi-source Multi-Layer TL Network with RFE). MSML-TL-RFE was tested on various datasets to demonstrate its efficacy and robustness. The predicted demand was further investigated using a two-echelon hierarchical supply chain model, where newly launched products are sold at the bottom of individual ends of the hierarchy. The accuracy and cost minimisation of the MSML-TL-RFE approach were superior to other existing TL approaches that were used for comparison in this thesis.

The following presents, in slightly greater detail, the key aspects of each technical chapter.

### **5.1.1 Efficient and Effective Predictions of Long-Term Time Series Data**

The study presented in Chapter 2 aimed to develop a long-term forecasting approach that would be useful for both stock market investors and supply chain practitioners. The proposed PLR-ALSTM-NN model was created by combining statistical regression analysis and SLSTM, an advanced DL model. Synthetic training data was also included to improve the model's generalization capabilities, resulting in improved accuracy in forecasting time

## *5. CONCLUSION AND FUTURE DIRECTIONS*

---

series data. The entire model was fine-tuned using an optimised mix of parameters, to obtain a good outcome. The approach was extensively validated using four stock market datasets and compared with various contemporary ML and DL techniques to demonstrate its acceptable performance in financial market engineering. Additionally, the proposed model was implemented using non-financial market data from a time series supply chain, to demonstrate its generic behavior in different sectors.

Observations derived from the implementation of PLR-ALSTM-NN in Chapter 2 revealed that the proposed model generates highly accurate predictions, particularly for long-term forecasting, surpassing the existing literature and establishing the model's robustness, while the forecasting accuracy of other approaches varied with varying forecasting horizons. The findings from this chapter would aid practitioners in making decisions based on real-life examples. Hence long-term stock market investors who adopt a buy-and-hold strategy would benefit from this approach by assessing the long-term prospects of markets and investing accordingly, which would have an overall positive impact on the economic growth of a country. While PLR-ALSTM-NN is superior for long-term prediction, it is not in making short-term predictions. This opens the opportunity to explore and develop other approaches that can make efficient and effective sales predictions in a wide range of forecasting horizons, which was further investigated in Chapter 3.

### **5.1.2 A Robust Hybrid Approach to Predict Sales Data over Varying Forecasting Horizons to Optimise Supply Chain Cost**

Chapter 3 aims to build on the previous chapter's work, which showed that the accuracy of various forecasting approaches varies with the length of the forecasting horizon. To address this issue, a new approach, called SBFA, was developed by combining different advanced forecasting methods and switching between them based on the lowest validation error for different datasets. The study involved two different datasets and validated SBFA by using it to predict reorder points for a two-echelon supply chain model with different attributes, such as fixed lead time, dynamic lead time, and variable safety stock. The predictions from

SBFA and other traditional/ML/DL forecasting approaches were compared by calculating the overall supply chain cost.

In Chapter 3, it was observed that deploying SBFA in various settings, such as static and dynamic lead times, and varied safety stock levels, can lead to lower overall supply chain cost through better prediction accuracy. The research also demonstrated a forecasting algorithm selection process based on these findings. The results of this chapter have practical implications for practitioners, as it was shown that the selection of an approach, including SBFA, should consider factors such as forecasting horizon, accuracy, time, and cost savings. Combining these factors can help practitioners choose the most suitable approach.

Following the discussion of training approaches in Chapters 2 and 3, where PLR-ASLTNN and SBFA were trained with abundant data. Chapter 4 investigated a knowledge transfer methodology for making sales/demand predictions of newly introduced market products with limited training data, where leveraging knowledge from similar existing market products becomes necessary.

### **5.1.3 A TL Approach to Maximise Knowledge Transfer Process with Limited Available Data**

The preceding two chapters focused on improving the accuracy of forecasting models using large amounts of data, but DL models can be limited by the availability of data. Chapter 4 introduces a TL approach called MSML-TL-RFE, which is designed to predict the sales of newly launched products, mimicking the concept of limited data. The chapter presented an extensive study of three datasets while introducing MSML-TL-RFE. The approach involved creating a pre-trained network by gathering knowledge from multiple similar source products with critical features. The chapter used a KNN algorithm to identify similar sources and considers all available features of the data. Additionally, the chapter demonstrated the effect of using knowledge from single sources versus multiple sources.

## *5. CONCLUSION AND FUTURE DIRECTIONS*

---

The study showed that MSML-TL-RFE effectively transfers knowledge from multiple  $S_D$  to  $T_D$ , allowing for more accurate predictions of sales patterns for newly launched products, when compared to other TL approaches. An extension to the study presented a hierarchical supply chain (SC) model with different attributes, to demonstrate the concept of information sharing among retailers. The study found that horizontal information sharing along the hierarchy improved the identification of source items and led to a better knowledge transfer process using MSML-TL-RFE. Similar to the findings in Chapter 3, it has also been found in Chapter 4 that better predictions led to lower supply chain costs. The study in Chapter 4 can benefit managers in decision making while dealing with insufficient data, by considering factors such as operating cost, time, and the accuracy of different TL approaches. It can also be used as a guideline for selecting the most appropriate business channel to carry out business operations while incurring a minimum cost.

### **5.2 Limitations of This Thesis**

Despite the significant contributions made in this thesis, there are a few limitations that need to be addressed in future research. The following points highlight some of these limitations:

- **Long-term Forecasting Approach:** The PLR-ALSTM-NN forecasting approach, as described in Chapter 2 of this thesis, was primarily tailored and optimised for financial datasets, given their unique characteristics and patterns. While its primary focus has been on financial data, the underlying principles and methodologies of the approach may offer potential applicability to other time-series datasets, such as those related to air pollution or weather. However, it's important to acknowledge that the effectiveness of the PLR-ALSTM-NN model on these alternative datasets could vary, and further research and adaptation would be necessary to fine-tune its performance in such contexts. Additionally, the hyperparameters of the studied

approaches were not fully optimised, and it is possible that further optimisation could improve prediction accuracy, although at the cost of increased training time.

- **SBFA:** While the hybrid SBFA presented in Chapter 3 showed promising results, it was limited to the supply chain domain and may not generalise well to other industries. Additionally, the current supply chain model used to validate the forecasting approach may not capture the full complexity and uncertainty of multi-echelon supply chains.
- **TL Approach:** Despite the great potential that the transfer learning approaches explored in Chapter 4 showed, they were only tested on retail sales data and may not generalise well to other domains or data sources. Moreover, the transfer learning approaches were based on the CNN architecture, and other neural network architectures remain unexplored. Additionally, the practical implications of information sharing in a hierarchical supply chain network were not fully investigated, and further research is needed to determine the most effective way to share information, while keeping confidential information private.

In summary, this thesis has achieved significant milestones in the domain of time series forecasting and predictive modelling. However, it is vital to acknowledge the overall inherent limitations that provide directions for future research endeavours. The predictive models developed in this thesis, while effective for time series data, may not readily extend to non-time series datasets or visual data, highlighting the need for specialized models in those domains. Furthermore, the use of publicly available datasets, although valuable for initial research, calls for future exploration with real-world datasets to assess model applicability and performance in practical scenarios. The choice of advanced ML and DL models was well-suited for single-modal datasets, but exploring reinforcement learning for multi-modal datasets to enhance sequential decision-making remains an avenue that is open for further investigation. Regrettably, time constraints prevented the incorporation of Federated Learning, leaving it as a promising avenue for privacy-preserving and decentralized predictive modelling. Finally, parameter exploration was bounded within certain limits, creating an opportunity for future research to delve deeper into parameter

## *5. CONCLUSION AND FUTURE DIRECTIONS*

---

optimisation and its impact on forecasting precision and computational efficiency. These limitations, rather than constraints, open the doors to new research prospects, enriching the ever-evolving field of predictive modelling and forecasting, which will be discussed more extensively in the next section.

### **5.3 Future Research Directions**

From the limitations highlighted in Section 5.2, numerous directions for future research arise from the work conducted in this study. Several potential future works have been identified and are described below.

#### **5.3.1 Future Work Related to Long-Term Forecasting Approach**

In the initial part of this thesis, a PLR-ALSTM forecasting approach is used on a financial dataset for making a long-horizon forecast. However it is worth exploring the combination of other deep learning approaches, such as CNNs, RNNs, and GRUs, with PLR in the future. Decision Tree-based approaches could also be experimented with and might lead to improved long-term predictions. The effectiveness of these approaches should be validated using a variety of time-series datasets, such as publicly available air pollution data, weather data, bitcoin sales data, and supply chain sales data, rather than relying solely on widely available financial market datasets. This would provide insight into how the different forecasting systems perform based on the characteristics of the data. The hyperparameters of the studied approaches, including the presented one, can be optimised using a grid search technique to increase prediction accuracy, although this may come at the cost of increased training time.

In addition, the existing method presented in Chapter 2, can be further extended by incorporating financial market modelling techniques to investigate the benefits of a better long-horizon forecasting approach for long-term investors. These models can also be applied to supply chains, helping practitioners and managers to make informed decisions and

reduce costs.

### **5.3.2 Future Work Related to SBFA**

Although a lot of experimental analysis has been conducted to determine the value of a hybrid SBFA in Chapter 3, there are still many avenues that have yet to be explored. One possibility is incorporating more state-of-the-art algorithms, such as GAN or transformer-based architectures, into the SBFA architecture, to improve accuracy. This would likely result in a longer training time for the entire network. However this issue can be addressed by using smart selection processes, such as meta-learning algorithms, decision trees, reinforcement learning, and genetic algorithms, to choose the best algorithms, based on the particular data's characteristics, features, memory consumption, and prediction horizon. This way, only a few algorithms would be trained instead of the entire pool, reducing the overall training time.

In addition to focusing on the prediction algorithms, the supply chain model used to validate the forecasting approach in terms of cost can also be updated to identify reductions in the bullwhip effect with more accurate forecasts. The model can also be extended to include more stakeholders, such as manufacturers and distributors, in a multi-echelon supply chain to make it more complex and capture uncertainty along the chain. The accuracy of the forecasts can also be used to assess and manage supply chain risk by incorporating relevant risk factors.

### **5.3.3 Future Work Related to TL Approach**

The transfer learning approaches explored in this thesis have opened up many avenues for future research. While the approaches investigated in this thesis only acquired knowledge from sales information from retail stores, their performance could be further evaluated using data from other sources, such as demographic data, market research reports, consumer feedback, and product reviews. With a large number of features in some data, unsuper-

## *5. CONCLUSION AND FUTURE DIRECTIONS*

---

vised clustering algorithms, such as the k-means clustering algorithm, can be applied to select the best source sales data. Having access to more information could significantly enhance a system's overall transfer learning ability. In this thesis, all the transfer learning approaches were based on the CNN architecture, but other neural networks, such as LSTM, RNN, GRU, and GAN, remain unexplored. A new transfer learning approach could be developed by combining these networks with CNN and other tree-based structures.

In a hierarchical supply chain network, information could be passed from different layers to pre-train a neural network. However businesses are often reluctant to share information with their competitors, making it an interesting area of research to determine what information to share and how much to share with other stakeholders, in order to maximise profit while keeping confidential information private.

The goal of minimising cost can be achieved by choosing the best transfer learning approach and pre-training it using various sources of information to make predictions in the target domain. This could provide deeper insights into the concept of information sharing while taking into consideration stakeholders' perspectives.

### **5.3.4 Reflection on the Use of Predictive Models in Pandemic Prediction and Management**

The research journey undertaken in this thesis, focused on predictive models for supply chain management, has also sparked contemplation about the broader implications and applications of predictive modelling, particularly in the context of global pandemics. While this study addressed sales forecasting and supply chain optimisation, the lessons learned from this research can be extended to the critical domain of pandemic prediction and management.

One of the overarching takeaways is the paramount importance of adaptability and versatility in predictive models. Just as this research aimed to develop models that could cater to varying risk profiles and forecasting horizons for supply chain practitioners, the

---

## *5. CONCLUSION AND FUTURE DIRECTIONS*

pandemic landscape presents a dynamic and evolving challenge. Predictive models used in the context of pandemics should be capable of accommodating changing data inputs, evolving virus strains, and shifting healthcare capacities.

Furthermore, the need for data is evident in both supply chain forecasting and pandemic prediction. In this work, the limitations imposed by data scarcity were addressed through techniques such as transfer learning and knowledge sharing. Similarly, in pandemic prediction, having access to real-time data on infection rates, vaccination progress, and healthcare resources is critical. Models should be designed to leverage every available data source, including demographic information, travel data, and genomic sequencing, to provide the most accurate predictions.

Another parallel lies in the significance of collaboration and information sharing. Just as supply chain stakeholders may be hesitant to share sensitive information, countries and healthcare systems can be reluctant to share data during a global health crisis. Balancing the need for information to make informed predictions with privacy and security concerns is an essential challenge.

Ultimately, predictive models in pandemic prediction and management must be designed with flexibility, adaptability, and data-driven insights in mind. They should account for various risk profiles, evolving conditions, and the critical need for data sharing. The research conducted in this thesis has reinforced the notion that predictive models can be invaluable tools for addressing complex challenges, be it in supply chain management or pandemic response, as long as they are designed to be robust, responsive, and collaborative.

# References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL <https://www.tensorflow.org/>. Software available from tensorflow.org.

Laith Abualigah, Raed Abu Zitar, Khaled H Almotairi, Ahmad MohdAziz Hussein, Mohamed Abd Elaziz, Mohammad Reza Nikoo, and Amir H Gandomi. Wind, solar, and photovoltaic renewable energy systems with and without energy storage optimization: a survey of advanced machine learning and deep learning techniques. *Energies*, 15(2): 578, 2022.

Luis Aburto and Richard Weber. Improved supply chain management based on hybrid demand forecasts. *Applied Soft Computing*, 7(1):136–144, 2007. ISSN 1568-4946. doi: <https://doi.org/10.1016/j.asoc.2005.06.001>. URL <https://www.sciencedirect.com/science/article/pii/S1568494605000311>.

Kahkashan Afrin, Bimal Nepal, and Leslie Monplaisir. A data-driven framework to new

- product demand prediction: Integrating product differentiation and transfer learning approach. *Expert Systems with Applications*, 108:246–257, 2018.
- Maxim Afteniy. Predicting time series with Transformer. Master’s thesis, Aalto University. School of Science, 2021. URL <http://urn.fi/URN:NBN:fi:aalto-202105236923>.
- Tanveer Ahmad, Hongyu Zhu, Dongdong Zhang, Rasikh Tariq, A Bassam, Fasee Ullah, Ahmed S AlGhamdi, and Sultan S Alshamrani. Energetics systems and artificial intelligence: applications of industry 4.0. *Energy Reports*, 8:334–361, 2022.
- Fahim Ahmed and Kyoung-Yun Kim. Recursive approach to combine expert knowledge and data-driven rsw weldability certification decision making process. *Robotics and Computer-Integrated Manufacturing*, 79:102428, 2023.
- Supriyo Ahmed, Ripon K Chakrabortty, Daryl L Essam, and Weiping Ding. Poly-linear regression with augmented long short term memory neural network: Predicting time series data. *Information Sciences*, 2022.
- Supriyo Ahmed, Ripon K Chakrabortty, Daryl L Essam, and Weiping Ding. A switching based forecasting approach (SBFA) for forecasting sales data in supply chains. *Neurocomputing*, Under review 2nd Round, submitted Oct. 2022.
- Supriyo Ahmed, Ripon K Chakrabortty, Daryl L Essam, and Weiping Ding. A multi-source multi-layer-based transfer learning approach for forecasting customer demands of newly launched products. *Expert Systems with Applications*, Under review, submitted July. 2023.
- Thomas Aichner and Valentin Santa. Demand forecasting methods and the potential of machine learning in the fmccg retail industry. In *Serving the Customer: The Role of Selling and Sales*, pages 215–252. Springer, 2023.
- Bashar Alhnaity, Stefanos Kollias, Georgios Leontidis, Shouyong Jiang, Bert Schamp, and Simon Pearson. An autoencoder wavelet based deep neural network with attention mechanism for multi-step prediction of plant growth. *Information Sciences*, 560:35–50, 2021.

Manar Alkhatib, Azza Abdel Monem, and Khaled Shaalan. Deep learning for arabic error detection and correction. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 19(5):1–13, 2020.

Khaled A Althelaya, El-Sayed M El-Alfy, and Salahadin Mohammed. Evaluation of bidirectional lstm for short-and long-term stock market prediction. In *2018 9th international conference on information and communication systems (ICICS)*, pages 151–156. IEEE, 2018.

R Ame Rayan, A Suruliandi, SP Raja, and H Benjamin Fredrick David. A survey on an analysis of big data open source datasets, techniques and tools for the prediction of corona virus disease. *Journal of Circuits, Systems and Computers*, 2023.

Mohammad Reza Amin-Naseri and B Rostami Tabar. Neural network approach to lumpy demand forecasting for spare parts in process industries. In *2008 International Conference on Computer and Communication Engineering*, pages 1378–1382. IEEE, 2008.

Halit Apaydin, Hajar Feizi, Mohammad Taghi Sattari, Muslume Sevba Colak, Shahaboddin Shamshirband, and Kwok-Wing Chau. Comparative analysis of recurrent neural network architectures for reservoir inflow forecasting. *Water*, 12(5):1500, 2020.

Marta Arias, Argimiro Arratia, and Ramon Xuriguera. Forecasting with twitter data. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(1):1–24, 2014.

Nimalan Arinaminpathy, Deepak Batra, Sunil Khaparde, Thongsuanmung Vualnam, Nilesh Maheshwari, Lokesh Sharma, Sreenivas A Nair, and Puneet Dewan. The number of privately treated tuberculosis cases in India: an estimation from drug sales data. *The Lancet Infectious Diseases*, 16(11):1255–1260, 2016.

J. Scott Armstrong. *Combining Forecasts*, pages 417–439. Springer US, Boston, MA, 2001. ISBN 978-0-306-47630-3. doi: 10.1007/978-0-306-47630-3\_19. URL [https://doi.org/10.1007/978-0-306-47630-3\\_19](https://doi.org/10.1007/978-0-306-47630-3_19).

Behzad Ashtari, Mohsen Alizadeh Bidgoli, Mojtaba Babaei, and Amir Ahmarinejad. A two-stage energy management framework for optimal scheduling of multi-microgrids

- with generation and demand forecasting. *Neural Computing and Applications*, pages 1–15, 2022.
- Matin N Ashtiani and Bijan Raahmei. News-based intelligent prediction of financial markets using text mining and machine learning: A systematic literature review. *Expert Systems with Applications*, page 119509, 2023.
- Mohsen Attaran and Bilge Gokhan Celik. Digital twin: Benefits, use cases, challenges, and opportunities. *Decision Analytics Journal*, page 100165, 2023.
- Michel Ballings, Dirk Van den Poel, Nathalie Hespeels, and Ruben Gryp. Evaluating multiple classifiers for stock price direction prediction. *Expert Systems with Applications*, 42(20):7046–7056, 2015.
- Wei Bao, Jun Yue, and Yulei Rao. A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PloS one*, 12(7), 2017.
- Bernard Baumohl. *The secrets of economic indicators: hidden clues to future economic trends and investment opportunities*. FT Press, 2012.
- Christopher Bennett, Rodney A Stewart, and Junwei Lu. Autoregressive with exogenous variables and neural network short-term load forecast models for residential low voltage distribution networks. *Energies*, 7(5):2938–2960, 2014.
- Mustapha Kamal Benramdane, Elena Kornyshova, Samia Bouzefrane, and Hubert Maupas. Supervised machine learning for matchmaking in digital business ecosystems and platforms. *Information Systems Frontiers*, pages 1–13, 2023.
- Nitin Bhardwaj et al. Time series prediction using hybrid arima-ann models for sugarcane. *International Journal of Plant & Soil Science*, 34(23):772–782, 2022.
- Jing Bi, Yongze Lin, Quanxi Dong, Haitao Yuan, and MengChu Zhou. Large-scale water quality prediction with integrated deep neural network. *Information Sciences*, 571:191–205, 2021.

Marko Bohanec, Mirjana Kljajić Borštnar, and Marko Robnik-Šikonja. Explaining machine learning models in sales predictions. *Expert Systems with Applications*, 71:416–428, 2017.

Halima Bousqaoui, Said Achhab, and Kawtar Tikito. Machine learning applications in supply chains: An emphasis on neural network applications. In *2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech)*, pages 1–7. IEEE, 2017.

Braulio Brunaud, José M Laínez-Aguirre, Jose M Pinto, and Ignacio E Grossmann. Inventory policies and safety stock optimization for supply chain planning. *AIChE Journal*, 65(1):99–112, 2019.

Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

Ayaz Hussain Bukhari, Muhammad Asif Zahoor Raja, Muhammad Sulaiman, Saeed Islam, Muhammad Shoaib, and Poom Kumam. Fractional neuro-sequential arfima-lstm for financial market forecasting. *IEEE Access*, 8:71326–71338, 2020.

Ümit Çavuş Büyüksahin and Şeyda Ertekin. Improving forecasting accuracy of time series data using a new arima-ann hybrid method and empirical mode decomposition. *Neurocomputing*, 361:151–163, 2019.

MA Castán-Lascorz, P Jiménez-Herrera, A Troncoso, and G Asencio-Cortés. A new hybrid method for predicting univariate and multivariate time series based on pattern forecasting. *Information Sciences*, 586:611–627, 2022.

Sungyong Chang, Saerom Lee, Daniel Levinthal, Hart Posen, Phanish Puranam, and Hyejin Youn. Artificial intelligence and the next frontier of organizational modeling.

*Academy of Management Proceedings*, 2019:12483, 08 2019. doi: 10.5465/AMBPP.2019.12483symposium.

Fahima Charef. Exchange rate forecasting: Nonlinear garch-nn modeling approach. *Annals of Data Science*, pages 1–11, 2023.

Charles W Chase Jr et al. Machine learning is changing demand forecasting. *The Journal of Business Forecasting*, 35(4):43, 2016.

Lujie Chen, Mengqi Jiang, Taiyu Li, Fu Jia, and Ming K Lim. Supply chain learning and performance: a meta-analysis. *International Journal of Operations & Production Management*, ahead-of-print(ahead-of-print), 2023.

Wei Chen, Manrui Jiang, Wei-Guo Zhang, and Zhensong Chen. A novel graph convolutional feature based convolutional neural network for stock trend prediction. *Information Sciences*, 556:67–94, 2021.

Yifei Chen and Zhihan Fu. Multi-step ahead forecasting of the energy consumed by the residential and commercial sectors in the united states based on a hybrid cnn-bilstm model. *Sustainability*, 15(3):1895, 2023.

Fangzheng Cheng, Tian Li, Yi-ming Wei, and Tijun Fan. The vec-nar model for short-term forecasting of oil prices. *Energy Economics*, 78:656–667, 2019.

Ching-Chin Chern, Wei-Yun Chen, and Bo Hsiao. A time series analysis approach to forecast price fluctuation points. *International Journal of Management Science and Engineering Management*, pages 1–11, 2023.

Mahima Choudhary, Saumya Jain, and Greeshma Arya. Classical models vs deep leaning: Time series analysis. In *Advancements in Interdisciplinary Research: First International Conference, AIR 2022, Prayagraj, India, May 6–7, 2022, Revised Selected Papers*, pages 276–286. Springer, 2023.

Francesco Colasanto, Luca Grilli, Domenico Santoro, and Giovanni Villani. Albertino for stock price prediction: a gibbs sampling approach. *Information Sciences*, 597:341–357, 2022.

Ana Corberán-Vallet, José D Bermúdez, and Enriqueta Vercher. Forecasting correlated time series with exponential smoothing models. *International Journal of Forecasting*, 27(2):252–265, 2011.

Alvaro HC Correia, Robert Peharz, and Cassio de Campos. Joints in random forests. *arXiv preprint arXiv:2006.14937*, 2020.

D Criado-Ramón, LGB Ruiz, and MC Pegalajar. Electric demand forecasting with neural networks and symbolic time series representations. *Applied Soft Computing*, 122:108871, 2022.

Vander Luiz Da Silva, João Luiz Kovaleski, and Regina Negri Pagani. Technology transfer in the supply chain oriented to industry 4.0: a literature review. *Technology Analysis & Strategic Management*, 31(5):546–562, 2019.

Saeid Sadeghi Darvazeh, Iman Raeesi Vanani, and Farzaneh Mansouri Musolu. Big data analytics and its applications in supply chain management. *New Trends in the Use of Artificial Intelligence for the Industry 4.0*, page 175, 2020.

Ivana Vívian De Carvalho and Neila Conceição Viana Cunha. Proposta de um modelo de transferência de tecnologia para as universidades públicas brasileiras. In *XV Congreso Latino-Iberoamericano de Gestión Tecnológica*, 2013.

Gabriella Dellino, Teresa Laudadio, Renato Mari, Nicola Mastronardi, and Carlo Meloni. A reliable decision support system for fresh food supply chain management. *International Journal of Production Research*, 56(4):1458–1485, 2018.

Zhuofu Deng, Chenxu Liu, and Zhiliang Zhu. Inter-hours rolling scheduling of behind-the-meter storage operating systems using electricity price forecasting based on deep convolutional neural network. *International Journal of Electrical Power & Energy Systems*, 125:106499, 2021.

Robin Devooght and Hugues Bersini. Long and short-term recommendations with recurrent neural networks. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*, pages 13–21, 2017.

- Luca Di Persio and Oleksandr Honchar. Artificial neural networks architectures for stock price prediction: Comparisons and applications. *International journal of circuits, systems and signal processing*, 10(2016):403–413, 2016.
- Stephen M Disney, Borja Ponte, and Xun Wang. Exploring the nonlinear dynamics of the lost-sales order-up-to policy. *International Journal of Production Research*, 59(19):5809–5830, 2021.
- LGBM Documentation—LGBM. Lightgbm. URL: <https://lightgbm.readthedocs.io>, 2022.
- XGBoost Documentation—XGBoost. Xgboost. URL: <https://xgboost.readthedocs.io/en/latest/>, 2022.
- Alexandre Dolgui, Dmitry Ivanov, and Maxim Rozhkov. Does the ripple effect influence the bullwhip effect? an integrated analysis of structural and operational dynamics in the supply chain. *International Journal of Production Research*, 58(5):1285–1301, 2020.
- Michael T Dugan, Keith A Shriver, and Peter A Silhan. How to forecast income statement items for auditing purposes. *The Journal of Business Forecasting*, 13(2):22, 1994.
- Hayana Dullah, Ali Najah Ahmed, Pavitra Kumar, and Ahmed Elshafie. Integrated non-linear autoregressive neural network and holt winters exponential smoothing for river streaming flow forecasting at aswan high. *Earth Science Informatics*, pages 1–14, 2022.
- Erol Egrioglu and Eren Bas. A new hybrid recurrent artificial neural network for time series forecasting. *Neural Computing and Applications*, pages 1–11, 2022.
- Ricardo Ernst and Bardia Kamrad. Estimating demand by using sales information: Inaccuracies encountered. *European journal of operational research*, 174(2):675–688, 2006.
- Asif Faisal, Md Kamruzzaman, Tan Yigitcanlar, and Graham Currie. Understanding autonomous vehicles. *Journal of transport and land use*, 12(1):45–72, 2019.
- Javad Feizabadi. Machine learning demand forecasting and supply chain performance. *International Journal of Logistics Research and Applications*, 25(2):119–142, 2022.

Zhong-kai Feng and Wen-jing Niu. Hybrid artificial neural network and cooperation search algorithm for nonlinear river flow time series forecasting in humid and semi-humid regions. *Knowledge-Based Systems*, 211:106580, 2021.

Emmanuel Senyo Fianu. Analyzing and forecasting multi-commodity prices using variants of mode decomposition-based extreme learning machine hybridization approach. *Forecasting*, 4(2):538–564, 2022.

Thomas Fischer and Christopher Krauss. Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2):654–669, 2018.

Luciano Floridi and Massimo Chiriatti. Gpt-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30(4):681–694, 2020.

W.R. Foster, F. Collopy, and L.H. Ungar. Neural network forecasting of short, noisy time series. *Computers & Chemical Engineering*, 16(4):293–297, 1992. ISSN 0098-1354. doi: [https://doi.org/10.1016/0098-1354\(92\)80049-F](https://doi.org/10.1016/0098-1354(92)80049-F). URL <https://www.sciencedirect.com/science/article/pii/009813549280049F>. Neural network applications in chemical engineering.

Jerome H Friedman. Greedy function approximation: A gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

Milton Friedman. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the american statistical association*, 32(200):675–701, 1937.

Edgar Fuller, Sampath Yerramalla, Bojan Cukic, and Srikanth Gururajan. An approach to predicting non-deterministic neural network behavior. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 5, pages 2921–2926. IEEE, 2005.

Xiang Gao, Junhao Wen, and Cheng Zhang. An improved random forest algorithm for predicting employee turnover. *Mathematical Problems in Engineering*, 2019, 2019.

Eric Octavio Mayoral Garinian, Tomas Eloy Salais Fierro, José Antonio Marmolejo Saucedo, and Roman Rodriguez Aguilar. Machine learning applications for demand driven in supply chain: Literature review. *Smart Applications with Advanced Machine Learning and Human-Centred Problem Design*, pages 763–772, 2023.

Yogesh Gautam. Transfer learning for covid-19 cases and deaths forecast using lstm network. *ISA transactions*, 124:41–56, 2022.

Aurélien Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems.* " O'Reilly Media, Inc.", 2019.

Michael Gilliland. *The business forecasting deal: Exposing myths, eliminating bad practices, providing practical solutions.* John Wiley & Sons, 2010.

João NC Gonçalves, M Sameiro Carvalho, and Paulo Cortez. Operations research models and methods for safety stock determination: A review. *Operations Research Perspectives*, page 100164, 2020.

Daniele Grattarola and Cesare Alippi. Graph neural networks in tensorflow and keras with spektral [application notes]. *IEEE Computational Intelligence Magazine*, 16(1):99–106, 2021.

Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural networks*, 18(5-6):602–610, 2005.

Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. Why do tree-based models still outperform deep learning on tabular data? *arXiv preprint arXiv:2207.08815*, 2022.

Qitao Gu and Qun Dai. A novel active multi-source transfer learning algorithm for time series forecasting. *Applied Intelligence*, 51(3):1326–1350, 2021.

Qitao Gu, Qun Dai, Huihui Yu, and Rui Ye. Integrating multi-source transfer learning, active learning and metric learning paradigms for time series prediction. *Applied Soft Computing*, 109:107583, 2021.

Shihao Gu, Bryan Kelly, and Dacheng Xiu. Empirical asset pricing via machine learning.

*The Review of Financial Studies*, 33(5):2223–2273, 2020.

Alev Taskin Gumus, Ali Fuat Guneri, and Fusun Ulengin. A new methodology for multi-echelon inventory management in stochastic and neuro-fuzzy environments. *International journal of production economics*, 128(1):248–260, 2010.

Xuesong Guo, Zhiping Yuan, and Bojing Tian. Supplier selection based on hierarchical potential support vector machine. *Expert Systems with Applications*, 36(3):6978–6985, 2009.

Zahra Hajirahimi and Mehdi Khashei. Hybrid structures in time series modeling and forecasting: A review. *Engineering Applications of Artificial Intelligence*, 86:83–106, 2019.

Yechan Han, Jaeyun Kim, and David Enke. A machine learning trading system for the stock market based on n-period min-max labeling using xgboost. *Expert Systems with Applications*, 211:118581, 2023.

Qi-Qiao He, Patrick Cheong-Iao Pang, and Yain-Whar Si. Transfer learning for financial time series forecasting. In *Pacific Rim International Conference on Artificial Intelligence*, pages 24–36. Springer, 2019.

Robin Hirt and Niklas KÄ1/4hl. Cognition in the era of smart service systems: inter-organizational analytics through meta and transfer learning. In *Proceedings of the 39th International Conference on Information Systems, ICIS 2018*, 2018.

Ming-Wei Hsu, Stefan Lessmann, Ming-Chien Sung, Tiejun Ma, and Johnnie EV Johnson. Bridging the divide in financial market forecasting: machine learners vs. financial economists. *Expert Systems with Applications*, 61:215–234, 2016.

Linqing Huang, Jinfu Fan, Wangbo Zhao, and Yang You. A new multi-source transfer learning method based on two-stage weighted fusion. *Knowledge-Based Systems*, 262: 110233, 2023.

Xiaoqiao Huang, Qiong Li, Yonghang Tai, Zaiqing Chen, Jun Liu, Junsheng Shi, and Wuming Liu. Time series forecasting for hourly photovoltaic power using conditional generative adversarial network and bi-lstm. *Energy*, 246:123403, 2022.

Jakob Huber and Heiner Stuckenschmidt. Daily retail demand forecasting using machine learning with emphasis on calendric special days. *International Journal of Forecasting*, 36(4):1420–1438, 2020.

Anas Iftikhar, Laura Purvis, Ilaria Giannoccaro, and Yingli Wang. The impact of supply chain complexities on supply chain resilience: the mediating effect of big data analytics. *Production Planning & Control*, pages 1–21, 2022.

Dmitry Ivanov and Alexandre Dolgui. A digital supply chain twin for managing the disruption risks and resilience in the era of industry 4.0. *Production Planning & Control*, pages 1–14, 2020.

Dmitry Ivanov, Suresh Sethi, Alexandre Dolgui, and Boris Sokolov. A survey on control theory applications to operational systems, supply chain management, and industry 4.0. *Annual Reviews in Control*, 46:134–147, 2018.

Dmitry Ivanov, Alexandre Dolgui, Ajay Das, and Boris Sokolov. Digital supply chain twins: Managing the ripple effect, resilience, and disruption risks by data-driven optimization, simulation, and visibility. In *Handbook of ripple effects in the supply chain*, pages 309–332. Springer, 2019a.

Dmitry Ivanov, Alexander Tsipoulanidis, and Jörn Schönberger. *Global supply chain and operations management: A decision-oriented introduction to the creation of value*, volume 2. Springer, 2019b.

Dmitry Ivanov, Jennifer Blackhurst, and Ajay Das. Supply chain resilience and its interplay with digital technologies: Making innovations work in emergency situations. *International journal of physical distribution & logistics management*, 2021.

Seyed Mohammad Jafar Jalali, Sajad Ahmadian, Abbas Khosravi, Miadreza Shafie-khah, Saeid Nahavandi, and João PS Catalão. A novel evolutionary-based deep convolutional

neural network model for intelligent load forecasting. *IEEE Transactions on Industrial Informatics*, 17(12):8243–8253, 2021.

Ebenezer Jangam, Aaron Antonio Dias Barreto, and Chandra Sekhara Rao Annavarapu. Automatic detection of covid-19 from chest ct scan and chest x-rays images using deep learning, transfer learning and stacking. *Applied Intelligence*, 52(2):2243–2259, 2022.

Wentao Jiang, Liwen Ling, Dabin Zhang, Ruibin Lin, and Liling Zeng. A time series forecasting model selection framework using cnn and data augmentation for small sample data. *Neural Processing Letters*, pages 1–28, 2023.

Yuan jiang Li, Yi Yang, Kai Zhu, and Jinglin Zhang. Clothing sale forecasting by a composite gru-prophet model with an attention mechanism. *IEEE Transactions on Industrial Informatics*, 2021.

Xuebo Jin, Xinghong Yu, Xiaoyi Wang, Yuting Bai, Tingli Su, and Jianlei Kong. Prediction for time series with cnn and lstm. In *Proceedings of the 11th International Conference on Modelling, Identification and Control (ICMIC2019)*, pages 631–641. Springer, 2020.

Ying Jin, Ming Gao, and Jixiang Yu. A transformer based sales prediction of smart container in new retail era. In *2021 5th International Conference on Deep Learning Technologies (ICDLT)*, pages 46–53, 2021.

Reuben Varghese Joseph, Anshuman Mohanty, Soumyae Tyagi, Shruti Mishra, Sandeep Kumar Satapathy, and Sachi Nandan Mohanty. A hybrid deep learning framework with cnn and bi-directional lstm for store item demand forecasting. *Computers and Electrical Engineering*, 103:108358, 2022.

Jianping Ju, Hong Zheng, Xiaohang Xu, Zhongyuan Guo, Zhaohui Zheng, and Mingyu Lin. Classification of jujube defects in small data sets based on transfer learning. *Neural Computing and Applications*, 34(5):3385–3398, 2022.

Kenneth B Kahn. Solving the problems of new product forecasting. *Business Horizons*, 57(5):607–615, 2014.

- Firuz Kamalov, Ikhlaas Gurrib, and Fadi Thabtah. Autoregressive and neural network models: A comparative study with linearly lagged series. In *2021 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, pages 175–180. IEEE, 2021.
- Amadu Fullah Kamara, Enhong Chen, and Zhen Pan. An ensemble of a boosted hybrid of deep learning models and technical analysis for forecasting stock prices. *Information Sciences*, 594:1–19, 2022.
- Sachin S Kamble, Angappa Gunasekaran, Vikas Kumar, Amine Belhadi, and Cyril Foropon. A machine learning based approach for predicting blockchain adoption in supply chain. *Technological Forecasting and Social Change*, 163:120465, 2021.
- Anirut Kantasa-Ard, Maroua Nouiri, Abdelghani Bekrar, Abdessamad Ait el Cadi, and Yves Sallez. Machine learning for demand forecasting in the physical internet: A case study of agricultural products in Thailand. *International Journal of Production Research*, 59(24):7491–7515, 2021.
- Tristan Karb, Niklas Kühl, Robin Hirt, and Varvara Glivici-Cotruta. A network-based transfer learning approach to improve sales forecasting of new products. *arXiv preprint arXiv:2005.06978*, 2020.
- Christos Katris. Forecasting the unemployment of med counties using time series and neural network models. *Journal of Statistical and Econometric Methods*, 8(2):37–49, 2019.
- Deeksha Kaul, Harika Raju, and BK Tripathy. Deep learning in healthcare. In *Deep Learning in Data Analytics*, pages 97–115. Springer, 2022.
- Fazil Kaytez. A hybrid approach based on autoregressive integrated moving average and least-square support vector machine for long-term forecasting of net electricity consumption. *Energy*, 197:117200, 2020.
- Yasaman Kazemi. Ai, big data & advanced analytics in the supply chain, 2019. URL

<https://www.forbes.com/sites/yasamankazemi/2019/01/29/ai-big-data-advanced-analytics-in-the-supply-chain/?sh=7c2e4df3244f>. Accessed: 2023-01-23.

Majd Kharfan, Vicky Wing Kei Chan, and Tugba Firdolas Efendigil. A data-driven forecasting approach for newly launched seasonal products by leveraging machine-learning approaches. *Annals of Operations Research*, 303(1):159–174, 2021.

Mehdi Khashei and Zahra Hajirahimi. A comparative study of series arima/mlp hybrid models for stock price forecasting. *Communications in Statistics-Simulation and Computation*, 48(9):2625–2640, 2019.

Daniel Kiefer, Florian Grimm, Markus Bauer, and Clemens Van Dinther. Demand forecasting intermittent and lumpy time series: Comparing statistical, machine learning and deep learning methods. In *54th Hawaii International Conference on System Sciences*, 2021.

Daniel Kiefer, Florian Grimm, Dinther Van, et al. Artificial intelligence in supply chain management: Investigation of transfer learning to improve demand forecasting of intermittent time series with deep learning. In *Proceedings of the 55th Hawaii International Conference on System Sciences*, 2022.

Zeynep Hilal Kilimci, A Okay Akyuz, Mitat Uysal, Selim Akyokus, M Ozan Uysal, Berna Atak Bulbul, and Mehmet Ali Ekmis. An improved demand forecasting model using deep learning approach and proposed decision integration strategy for supply chain. *Complexity*, 2019, 2019.

Shreya Kohli, Gracia Tabitha Godwin, and Siddhaling Urolagin. Sales prediction using linear and knn regression. In *Advances in Machine Learning and Computational Intelligence*, pages 321–329. Springer, 2021.

Sahar Koohfar, Wubeshet Woldemariam, and Amit Kumar. Prediction of electric vehicles charging demand: A transformer-based deep learning approach. *Sustainability*, 15(3):2105, 2023.

- Nikolaos Kourentzes. Intermittent demand forecasts with neural networks. *International Journal of Production Economics*, 143(1):198–206, 2013.
- Edward Kozłowski, Beata Kowalska, Dariusz Kowalski, and Dariusz Mazurkiewicz. Water demand forecasting by trend and harmonic analysis. *Archives of Civil and Mechanical Engineering*, 18(1):140–148, 2018.
- Urszula Kużelewska. Clustering algorithms for efficient neighbourhood identification in session-based recommender systems. In *International Conference on Dependability and Complex Systems*, pages 143–152. Springer, 2022.
- Henrika Langen and Martin Huber. How causal machine learning can leverage marketing strategies: Assessing and improving the performance of a coupon campaign. *Plos one*, 18(1):e0278937, 2023.
- Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.
- Miran Lee, Joo-Ho Lee, and Deok-Hwan Kim. Gender recognition using optimal gait feature based on recursive feature elimination in normal walking. *Expert Systems with Applications*, 189:116040, 2022.
- Yi-Shian Lee and Lee-Ing Tong. Forecasting time series using a methodology based on autoregressive integrated moving average and genetic programming. *Knowledge-Based Systems*, 24(1):66–72, 2011.
- Jiahong Li, Hui Bu, and Junjie Wu. Sentiment-aware stock market prediction: A deep learning method. In *2017 international conference on service systems and service management*, pages 1–6. IEEE, 2017.
- Na Li, Donald M Arnold, Douglas G Down, Rebecca Barty, John Blake, Fei Chiang, Tom Courtney, Marianne Waito, Rick Trifunov, and Nancy M Heddle. From demand forecasting to inventory ordering decisions for red blood cells through integrating machine learning, statistical modeling, and inventory optimization. *Transfusion*, 62(1):87–99, 2022a.

Shufei Li and Pai Zheng. Transfer learning for smart manufacturing: A stepwise survey.

*IFAC-PapersOnLine*, 53(5):37–42, 2020.

Weihua Li, Ruyi Huang, Jipu Li, Yixiao Liao, Zhuyun Chen, Guolin He, Ruqiang Yan, and Konstantinos Gryllias. A perspective survey on deep transfer learning for fault diagnosis in industrial scenarios: Theories, applications and challenges. *Mechanical Systems and Signal Processing*, 167:108487, 2022b.

Xi Li and Qian Liu. Strategic ignorance: Managing endogenous demand in a supply chain. *Omega*, 114:102729, 2023.

Yuanjiang Li, Yi Yang, Kai Zhu, and Jinglin Zhang. Clothing sale forecasting by a composite gru–prophet model with an attention mechanism. *IEEE Transactions on Industrial Informatics*, 17(12):8335–8344, 2021.

Qian Liu, Lifan Long, Qian Yang, Hong Peng, Jun Wang, and Xiaohui Luo. Lstm-snp: A long short-term memory model inspired from spiking neural p systems. *Knowledge-Based Systems*, 235:107656, 2022a.

Shimin Liu, Yuqian Lu, Pai Zheng, Hui Shen, and Jinsong Bao. Adaptive reconstruction of digital twins for machining systems: A transfer learning approach. *Robotics and Computer-Integrated Manufacturing*, 78:102390, 2022b.

Xiangjie Liu, Hao Zhang, Yuguang Niu, Deliang Zeng, Jizhen Liu, Xiaobing Kong, and Kwang Y Lee. Modeling of an ultra-supercritical boiler-turbine system with stacked denoising auto-encoder and long short-term memory network. *Information Sciences*, 525:134–152, 2020.

Ioannis E Livieris, Emmanuel Pintelas, and Panagiotis Pintelas. A cnn–lstm model for gold price time-series forecasting. *Neural computing and applications*, 32(23):17351–17360, 2020.

Francesco Lolli, Rita Gamberini, A Regattieri, Elia Balugani, T Gatos, and S Gucci. Single-hidden layer neural networks for forecasting intermittent demand. *International Journal of Production Economics*, 183:116–128, 2017.

Gloria Leticia López-Salazar, Rubén Molina-Sánchez, and Alejandra López-Salazar. Marketing strategy in msmes facing the covid-19 pandemic. In *Digital and Sustainable Transformations in a Post-COVID World: Economic, Social, and Environmental Challenges*, pages 329–345. Springer, 2023.

Wenjie Lu, Jiazheng Li, Jingyang Wang, and Lele Qin. A cnn-bilstm-am method for stock price prediction. *Neural Computing and Applications*, 33(10):4741–4753, 2021.

Xiaoyang Lu, Peijie Lin, Shuying Cheng, Yaohai Lin, Zhicong Chen, Lijun Wu, and Qianying Zheng. Fault diagnosis for photovoltaic array based on convolutional neural network and electrical time series graph. *Energy Conversion and Management*, 196:950–965, 2019.

Leanne Luce. Deep learning and demand forecasting. In *Artificial Intelligence for Fashion*, pages 155–166. Springer, 2019.

Xing Luo, Dongxiao Zhang, and Xu Zhu. Combining transfer learning and constrained long short-term memory for power generation forecasting of newly-constructed photovoltaic plants. *Renewable Energy*, 185:1062–1077, 2022.

Lingling Lv, Zongyu Wu, Lei Zhang, Brij B Gupta, and Zhihong Tian. An edge-ai based forecasting approach for improving smart microgrid efficiency. *IEEE Transactions on Industrial Informatics*, 18(11):7946–7954, 2022.

Zhengyu Ma, Chen Wang, and Zhongze Zhang. Deep learning algorithms for automotive spare parts demand forecasting. In *2021 International Conference on Computer Information Science and Artificial Intelligence (CISAI)*, pages 358–361. IEEE, 2021.

Amal Mahmoud and Ammar Mohammed. A survey on deep learning for time-series forecasting. In *Machine Learning and Big Data Analytics Paradigms: Analysis, Applications and Challenges*, pages 365–392. Springer, 2021.

Ritanjah Majhi, Ganapati Panda, Gadadhar Sahoo, Abhishek Panda, and Arvind Choubey. Prediction of s&p 500 and djia stock indices using particle swarm optimiza-

tion technique. In *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pages 1276–1282. IEEE, 2008.

Spyros Makridakis, Rob J Hyndman, and Fotios Petropoulos. Forecasting in social settings: The state of the art. *International Journal of Forecasting*, 36(1):15–28, 2020.

Prajowal Manandhar, Marouane Temimi, and Zeyar Aung. Short-term solar radiation forecast using total sky imager via transfer learning. *Energy Reports*, 9:819–828, 2023.

Paolo Mancuso, Veronica Piccialli, and Antonio M Sudoso. A machine learning approach for forecasting hierarchical time series. *Expert Systems with Applications*, 182:115102, 2021.

Sachin Kumar Mangla, Yiğit Kazançoğlu, Abdullah Yıldızbaşı, Cihat Öztürk, and Ahmet Çalık. A conceptual framework for blockchain-based sustainable supply chain and evaluating implementation barriers: A case of the tea supply chain. *Business Strategy and the Environment*, 2022.

Benjamin Maschler. A survey on deep industrial transfer learning in fault prognostics. *arXiv preprint arXiv:2301.01705*, 2023.

Ricardo P Masini, Marcelo C Medeiros, and Eduardo F Mendes. Machine learning advances for time series forecasting. *Journal of Economic Surveys*, 2021.

Sidra Mehtab and Jaydip Sen. Analysis and forecasting of financial time series using cnn and lstm-based deep learning models. In *Advances in Distributed Computing and Machine Learning*, pages 405–423. Springer, 2022.

D Dilkush Mihirsen, Josephine Taniha Joseph, and B Renisha. Time series analysis for supply chain planning in restaurants. In *2020 5th International Conference on Computing, Communication and Security (ICCCS)*, pages 1–5. IEEE, 2020.

Robin Mitra, Sarah F McGough, Tapabrata Chakraborti, Chris Holmes, Ryan Copping, Niels Hagenbuch, Stefanie Biedermann, Jack Noonan, Brieuc Lehmann, Aditi Shenvi, et al. Learning from data with structured missingness. *Nature Machine Intelligence*, 5(1):13–23, 2023.

- JP Morgan. Guide to the markets. *Market Insights, Brazil First quarter*, 2016.
- Amri Muhammin, Dedy Dwi Prastyo, and Henry Horng-Shing Lu. Forecasting with recurrent neural network in intermittent demand data. In *2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pages 802–809. IEEE, 2021.
- Khan Muhammad, Salman Khan, Javier Del Ser, and Victor Hugo C De Albuquerque. Deep learning for multigrade brain tumor classification in smart healthcare systems: A prospective survey. *IEEE Transactions on Neural Networks and Learning Systems*, 32(2):507–522, 2020.
- Thomas Musora, Z Chazuka, A Jaison, J Mapurisa, and J Kamusha. Sales forecasting of perishable products: A case study of a perishable orange drink. *Musora, T et. al (2023), Sales Forecasting of Perishable Products: A Case Study of a Perishable Orange Drink, Computer Science & Information Technology DOI*, 10, 2023.
- Paul Myerson. *The Art and Science of Demand and Supply Chain Planning in Today's Complex Global Economy*. CRC Press, 2023.
- Mojtaba Nabipour, Pooyan Nayyeri, Hamed Jabani, Amir Mosavi, Ely Salwana, et al. Deep learning for stock market prediction. *Entropy*, 22(8):840, 2020.
- Du Ni, Zhi Xiao, and Ming K Lim. A systematic review of the research trends of machine learning in supply chain management. *International Journal of Machine Learning and Cybernetics*, 11(7):1463–1482, 2020.
- Seyed Taghi Akhavan Niaki and Saeid Hoseinzade. Forecasting s&p 500 index using artificial neural networks and design of experiments. *Journal of Industrial Engineering International*, 9(1):1, 2013.
- Matthew Nicholson, Rahul Agrahari, Clare Conran, Haythem Assem, and John D Kelleher. The interaction of normalisation and clustering in sub-domain definition for multi-source transfer learning based time series anomaly detection. *Knowledge-Based Systems*, 257:109894, 2022.

Konstantinos Nikolopoulos. We need to talk about intermittent demand forecasting. *European Journal of Operational Research*, 291(2):549–559, 2021.

Shuteng Niu, Yongxin Liu, Jian Wang, and Houbing Song. A decade survey of transfer learning (2010–2020). *IEEE Transactions on Artificial Intelligence*, 1(2):151–166, 2020.

Yiyang Niu. Walmart sales forecasting using xgboost algorithm and feature engineering. In *2020 International Conference on Big Data & Artificial Intelligence & Software Engineering (ICBASE)*, pages 458–461. IEEE, 2020.

C. Olah. Understanding lstm networks. <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>, August 2015a.

Christopher Olah. Understanding lstm networks, 2015b. URL <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>.

Paulo Oliveira, Fátima Rodrigues, and Pedro Rangel Henriques. A formal definition of data quality problems. In *ICIQ*, 2005.

Intisar Omar, Muhammad Khan, and Andrew Starr. Suitability analysis of machine learning algorithms for crack growth prediction based on dynamic response data. *Sensors*, 23(3):1074, 2023.

Erik Otović, Marko Njirjak, Dario Jozinović, Goran Mauša, Alberto Michelini, and Ivan Stajduhar. Intra-domain and cross-domain transfer learning for time series data—how transferable are the features? *Knowledge-Based Systems*, 239:107976, 2022.

Shanoli Samui Pal and Samarjit Kar. Fuzzy transfer learning in time series forecasting for stock market prices. *Soft Computing*, 26(14):6941–6952, 2022.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

Xiongwen Pang, Yanqiang Zhou, Pan Wang, Weiwei Lin, and Victor Chang. An innovative neural network approach for stock market prediction. *The Journal of Supercomputing*, pages 1–21, 2018.

Bohdan M Pavlyshenko. Machine-learning models for sales time series forecasting. *Data*, 4(1):15, 2019.

Bohdan M Pavlyshenko. Analytics of business time series using machine learning and bayesian inference. *arXiv preprint arXiv:2205.12905*, 2022.

Gustavo Willam Pereira, Domingos Sárvio Magalhães Valente, Daniel Marçal de Queiroz, Nerilson Terra Santos, and Elpídio Inácio Fernandes-Filho. Soil mapping for precision agriculture using support vector machines combined with inverse distance weighting. *Precision Agriculture*, pages 1–16, 2022.

Marina Meireles Pereira and Enzo Morosini Frazzon. A data-driven approach to adaptive synchronization of demand and supply in omni-channel retail supply chains. *International Journal of Information Management*, 57:102165, 2021.

Rubén Pérez-Chacón, Gualberto Asencio-Cortés, Francisco Martínez-Álvarez, and A Troncoso. Big data time series forecasting based on pattern sequence similarity and its application to the electricity demand. *Information Sciences*, 540:160–174, 2020.

Anastasios Petropoulos, Vassilis Siakoulis, Konstantinos P Panousis, Loukas Papadoulas, and Sotirios Chatzis. Macroeconomic forecasting and sovereign risk assessment using deep learning techniques. *arXiv preprint arXiv:2301.09856*, 2023.

Hai Thanh Pham, Raffaele Testorelli, and Chiara Verbano. The impact of operational risk on performance in supply chains and the moderating role of integration. *Baltic Journal of Management*, 2023.

Christina Phillips. When Simulation Becomes Human Centric Analytics. In *Proceedings of the Operational Research Society Simulation Workshop 2021 (SW21)*, ed. by M. Fakhimi, D. Robertson, and T. Boness (Operational Research Society, 2021), 03 2021.

Christina Jane Phillips and Konstantinos Nikolopoulos. Forecast quality improvement with action research: A success story at pharmaco. *International Journal of Forecasting*, 35(1):129–143, 2019.

Tribikram Pradhan, Prashant Kumar, and Sukomal Pal. Claver: An integrated framework of convolutional layer, bidirectional lstm with attention mechanism based scholarly venue recommendation. *Information Sciences*, 559:212–235, 2021.

M Prakash, G Padmapriy, and M Vinoth Kumar. A review on machine learning big data using r. In *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, pages 1873–1877. IEEE, 2018.

Lutz Prechelt. Early stopping-but when? In *Neural Networks: Tricks of the trade*, pages 55–69. Springer, 1998.

Bagus Priambodo and Yuwan Jumaryadi. Time series traffic speed prediction using k-nearest neighbour based on similar traffic data. In *MATEC Web of Conferences*, volume 218, page 03021. EDP Sciences, 2018.

Luis Rabelo, Magdy Helal, and Chalermmon Lertpattarapong. Analysis of supply chains using system dynamics, neural nets, and eigenvalues. In *Proceedings of the 2004 Winter Simulation Conference, 2004.*, volume 2, pages 1136–1144. IEEE, 2004.

Rahul Rai, Manoj Kumar Tiwari, Dmitry Ivanov, and Alexandre Dolgui. Machine learning in manufacturing and industry 4.0 applications, 2021.

HV Ramachandra, G Balaraju, A Rajashekhar, and Harish Patil. Machine Learning Application for Black Friday Sales Prediction Framework. In *2021 International Conference on Emerging Smart Computing and Informatics (ESCI)*, pages 57–61. IEEE, 2021.

Milad Ramezankhani, Bryn Crawford, Apurva Narayan, Heinz Voggenreiter, Rudolf Seethaler, and Abbas S Milani. Making costly manufacturing smart with transfer learning under limited data: A case study on composites autoclave processing. *Journal of Manufacturing Systems*, 59:345–354, 2021.

Eduardo Ramos-Pérez, Pablo J Alonso-González, and José Javier Núñez-Velázquez. Multi-transformer: A new neural network-based architecture for forecasting s&p volatility. *Mathematics*, 9(15):1794, 2021.

V Ranganayaki, SN Deepa, and C Maheswari. Improved wind speed prediction using various neural network models. *International Journal of Electrical Engineering and Technology*, 11(2), 2020.

Prabod Rathnayaka, Harsha Moraliyage, Nishan Mills, Daswin De Silva, and Andrew Jennings. Specialist vs generalist: A transformer architecture for global forecasting energy time series. In *2022 15th International Conference on Human System Interaction (HSI)*, pages 1–5. IEEE, 2022.

Tobias Rebs, Marcus Brandenburg, and Stefan Seuring. System dynamics modeling for sustainable supply chain management: A literature review and systems thinking approach. *Journal of cleaner production*, 208:1265–1280, 2019.

Abderahman Rejeb, John G Keogh, and Horst Treiblmaier. Leveraging the internet of things and blockchain technology in supply chain management. *Future Internet*, 11(7):161, 2019.

Mauro Ribeiro, Katarina Grolinger, Hany F ElYamany, Wilson A Higashino, and Miriam AM Capretz. Transfer learning with seasonal and trend adjustment for cross-building energy forecasting. *Energy and Buildings*, 165:352–363, 2018.

Farnoush Ronaghi, Mohammad Salimibeni, Farnoosh Naderkhani, and Arash Mohammadi. Covid19-hpsmp: Covid-19 adopted hybrid and parallel deep information fusion framework for stock price movement prediction. *Expert Systems with Applications*, 187:115879, 2022.

Alaa Sagheer, Hala Hamdoun, and Hassan Youness. Deep lstm-based transfer learning approach for coherent forecasts in hierarchical time series. *Sensors*, 21(13):4379, 2021.

Saba Saleem, Javeria Amin, Muhammad Sharif, Muhammad Almas Anjum, Muhammad Iqbal, and Shui-Hua Wang. A deep network designed for segmentation and classification of leukemia using fusion of the transfer learning models. *Complex & Intelligent Systems*, 8(4):3105–3120, 2022.

Sreelekshmy Selvin, R Vinayakumar, EA Gopalakrishnan, Vijay Krishna Menon, and KP Soman. Stock price prediction using lstm, rnn and cnn-sliding window model. In *2017 international conference on advances in computing, communications and informatics (icacci)*, pages 1643–1647. IEEE, 2017.

Mahya Seyedian and Fereshteh Mafakheri. Predictive big data analytics for supply chain demand forecasting: methods, applications, and research opportunities. *Journal of Big Data*, 7(1):1–22, 2020.

Manu Sharma, Sunil Luthra, Sudhanshu Joshi, and Anil Kumar. Accelerating retail supply chain performance against pandemic disruption: Adopting resilient strategies to mitigate the long-term effects. *Journal of Enterprise Information Management*, 2021.

Rohit Sharma, Sachin S Kamble, Angappa Gunasekaran, Vikas Kumar, and Anil Kumar. A systematic literature review on machine learning applications for sustainable agriculture supply chain performance. *Computers & Operations Research*, 119:104926, 2020.

Zhenkun Shi, Sen Wang, Lin Yue, Lixin Pang, Xianglin Zuo, Wanli Zuo, and Xue Li. Deep dynamic imputation of clinical time series for mortality prediction. *Information Sciences*, 579:607–622, 2021.

Zhang Shilong et al. Machine learning model for sales forecasting by using xgboost. In *2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE)*, pages 480–483. IEEE, 2021.

Sima Siami-Namini, Neda Tavakoli, and Akbar Siami Namin. A comparison of arima and lstm in forecasting time series. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1394–1401. IEEE, 2018.

Shashi Pal Singh, Ajai Kumar, Hemant Darbari, Lenali Singh, Anshika Rastogi, and Shikha Jain. Machine translation using deep learning: An overview. In *2017 International Conference on Computer, Communications and Electronics (Comptelix)*, pages 162–167. IEEE, 2017.

- Nivethitha Somu, Gauthama Raman MR, and Krithi Ramamritham. A hybrid model for building energy consumption forecasting using long short term memory networks. *Applied Energy*, 261:114131, 2020.
- Alexander Spieske, Maximilian Gebhardt, Matthias Kopyto, and Hendrik Birkel. Improving resilience of the healthcare supply chain in a pandemic: Evidence from europe during the covid-19 crisis. *Journal of Purchasing and Supply Management*, 28(5):100748, 2022.
- Evangelos Spiliotis, Mahdi Abolghasemi, Rob J Hyndman, Fotios Petropoulos, and Vassilios Assimakopoulos. Hierarchical forecast reconciliation with machine learning. *Applied Soft Computing*, 112:107756, 2021.
- Aris A Syntetos, John E Boylan, and JD Croston. On the categorization of demand patterns. *Journal of the operational research society*, 56(5):495–503, 2005.
- Souhaib Ben Taieb and Rob J Hyndman. A gradient boosting approach to the kaggle load forecasting competition. *International journal of forecasting*, 30(2):382–394, 2014.
- Swee Chuan Tan and Jess Pei San Lau. Time series clustering: A superior alternative for market basket analysis. In *Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng-2013)*, pages 241–248. Springer, Singapore, 2014.
- Weng Chun Tan and Manjit Singh Sidhu. Review of rfid and iot integration in supply chain management. *Operations Research Perspectives*, page 100229, 2022.
- Tao Tao, Chen Sun, Zhaoyang Wu, Jian Yang, and Jing Wang. Deep neural network-based prediction and early warning of student grades and recommendations for similar learning approaches. *Applied Sciences*, 12(15):7733, 2022.
- Parvaneh Tavakol, Bijan Nahavandi, and Mahdi Homayounfar. Analyzing the drivers of bullwhip effect in pharmaceutical industry’s supply chain. *Journal of System Management (JSM)*, 9:1, 2023.
- Ankit Thakkar and Kinjal Chaudhari. A comprehensive survey on deep neural networks

for stock market: The need, challenges, and future directions. *Expert Systems with Applications*, page 114800, 2021.

Sébastien Thomassey. Sales forecasts in clothing industry: The key success factor of the supply chain management. *International Journal of Production Economics*, 128(2): 470–483, 2010.

Erfan Babaee Tirkolaee, Saeid Sadeghi, Farzaneh Mansoori Mooseloo, Hadi Rezaei Vandchali, and Samira Aeini. Application of machine learning in supply chain management: A comprehensive overview of the main areas. *Mathematical problems in engineering*, 2021, 2021.

Grigoris Tsoumakas. A survey of machine learning techniques for food sales prediction. *Artificial Intelligence Review*, 52(1):441–447, 2019.

Oluwatosin Umoren et al. The digital era and the rapid transformation in logistics and supply chain management structures. *TECHNOLOGY AND DEVELOPMENT IN SUPPLY CHAIN MANAGEMENT*, 2023.

Satish Vadlamani and Mayank Modashiy. Improving transportation planning using machine learning. In *Encyclopedia of Data Science and Machine Learning*, pages 3076–3088. IGI Global, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Venkataramana Veeramsetty, K Rajeshwar Reddy, M Santhosh, Arjun Mohnot, and Gaurav Singal. Short-term electric power load forecasting using random forest and gated recurrent unit. *Electrical Engineering*, 104(1):307–329, 2022.

Adrian Viellechner and Stefan Spinler. Novel data analytics meets conventional container shipping: Predicting delays by comparing various machine learning algorithms. In *Proceedings of the 53rd Hawaii International Conference on System Sciences*, 2020.

Marcus Voß, Christian Bender-Saebeckampf, and Sahin Albayrak. Residential short-term load forecasting using convolutional neural networks. In *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pages 1–6. IEEE, 2018.

Hadid Wahedi, Kacper Wrona, Mads Heltoft, Sarkaft Saleh, Thomas Roum Knudsen, Ulrik Bendixen, Izabela Nielsen, Subrata Saha, and Gregers Sandager Borup. Improving accuracy of time series forecasting by applying an arima-ann hybrid model. In *Advances in Production Management Systems. Smart Manufacturing and Logistics Systems: Turning Ideas into Action: IFIP WG 5.7 International Conference, APMS 2022, Gyeongju, South Korea, September 25–29, 2022, Proceedings, Part I*, pages 3–10. Springer, 2022.

Guangrun Wang, Liang Lin, Rongcong Chen, Guangcong Wang, and Jiqi Zhang. Joint learning of neural transfer and architecture adaptation for image recognition. *IEEE Transactions on Neural Networks and Learning Systems*, 33(10):5401–5415, 2021a.

HZ Wang, GB Wang, GQ Li, JC Peng, and YT Liu. Deep belief network based deterministic and probabilistic wind speed forecasting approach. *Applied Energy*, 182:80–93, 2016.

Jilong Wang, Rui Li, Renfa Li, Bin Fu, and Danny Z Chen. Hmckrautoencoder: An interpretable deep learning framework for time series analysis. *IEEE Transactions on Emerging Topics in Computing*, 10(1):99–111, 2022.

Jing Wang and Ling Luo. A forecasting method of multi-category product sales: analysis and application. *Management System Engineering*, 2(1):2, 2023.

Jinjiang Wang, Yulin Ma, Laibin Zhang, Robert X Gao, and Dazhong Wu. Deep learning for smart manufacturing: Methods and applications. *Journal of manufacturing systems*, 48:144–156, 2018.

Wei-Tsong Wang, Ying-Lien Lin, and Ting-Jun Chen. Exploring the effects of relationship quality and c-commerce behavior on firms' dynamic capability and c-commerce perfor-

mance in the supply chain management context. *Decision Support Systems*, 164:113865, 2023a.

Yixuan Wang, Yang Xu, Rui Yan, and Huajin Tang. Deep spiking neural networks with binary weights for object recognition. *IEEE Transactions on Cognitive and Developmental Systems*, 13(3):514–523, 2021b. doi: 10.1109/TCDS.2020.2971655.

Yu Wang, Ruonan Liu, Di Lin, Dongyue Chen, Ping Li, Qinghua Hu, and CL Philip Chen. Coarse-to-fine: Progressive knowledge transfer-based multitask convolutional neural network for intelligent large-scale fault diagnosis. *IEEE Transactions on Neural Networks and Learning Systems*, 34(2):761–774, 2023b. doi: 10.1109/TNNLS.2021.3100928.

Yujie Wang, Hui Liu, Qiang Guo, Shenxiang Xie, and Xiaofeng Zhang. Stock volatility prediction by hybrid neural network. *IEEE Access*, 7:154524–154534, 2019.

Zheng Wang, Qingxiu Liu, Hansi Chen, and Xuening Chu. A deformable cnn-dlstm based transfer learning method for fault diagnosis of rolling bearing under multiple working conditions. *International Journal of Production Research*, 59(16):4811–4825, 2021c.

Zhipeng Wang, Xuejun Liu, Jian Yu, Haizhou Wu, and Hongqiang Lyu. A general deep transfer learning framework for predicting the flow field of airfoils with small data. *Computers & Fluids*, 251:105738, 2023c.

Manuel Weber, Maximilian Auch, Christoph Doblander, Peter Mandl, and Hans-Arno Jacobsen. Transfer learning with time series data: a systematic mapping study. *Ieee Access*, 9:165409–165432, 2021.

M Weller and SF Crone. Supply chain forecasting: Best practices & benchmarking study. lancaster university, 2012.

Haoran Wen, Yang Du, Xiaoyang Chen, Enggee Lim, Huiqing Wen, Lin Jiang, and Wei Xiang. Deep learning based multistep solar forecasting for pv ramp-rate control using sky images. *IEEE Transactions on Industrial Informatics*, 17(2):1397–1406, 2020.

- Tingyu Weng, Wenyang Liu, and Jun Xiao. Supply chain sales forecasting based on lightgbm and lstm combination model. *Industrial Management & Data Systems*, 2019.
- Marc Wiedenmann and Andreas Größler. The impact of digital technologies on operational causes of the bullwhip effect—a literature review. *Procedia CIRP*, 81:552–557, 2019.
- Dingming Wu, Xiaolong Wang, and Shaocong Wu. Jointly modeling transfer learning of industrial chain information and deep learning for stock prediction. *Expert Systems with Applications*, 191:116257, 2022.
- Peng Xiong, Yonxin Zhu, Zhanrui Sun, Zihao Cao, Menglin Wang, Yu Zheng, Junjie Hou, Tian Huang, and Zhiqiang Que. Application of transfer learning in continuous time series for anomaly detection in commercial aircraft flight data. In *2018 IEEE International Conference on Smart Cloud (SmartCloud)*, pages 13–18. IEEE, 2018.
- Wei Yan, Junliang He, and Amy JC Trappey. Risk-aware supply chain intelligence: Ai-enabled supply chain and logistics management considering risk mitigation, 2019.
- Haolin Yang and Kristen R Schell. Real-time electricity price forecasting of wind farms with deep neural network transfer learning and hybrid datasets. *Applied Energy*, 299:117242, 2021.
- Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.
- Rui Ye and Qun Dai. Implementing transfer learning across different datasets for time series forecasting. *Pattern Recognition*, 109:107617, 2021.
- Lili Yin, Benling Li, Peng Li, and Rubo Zhang. Research on stock trend prediction method based on optimized random forest. *CAAI Transactions on Intelligence Technology*, 2021.
- Afshin Yousefi, Ayub Rahimzadeh, and Alireza Moradi. A new hybrid prediction reduces the bullwhip effect of demand in a three-level supply chain. *Journal of Modern Processes in Manufacturing and Production*, 7(3):45–58, 2018.

Ariele Zanfei, Bruno M Brentan, Andrea Menapace, Maurizio Righetti, and Manuel Herrera. Graph convolutional recurrent neural networks for water demand forecasting. *Water Resources Research*, 58(7):e2022WR032299, 2022.

George Zerveas, Srideepika Jayaraman, Dhaval Patel, Anuradha Bhamidipaty, and Carsten Eickhoff. A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2114–2124, 2021.

Biao Zhang, Chao Song, Xuchu Jiang, and Ying Li. Electricity price forecast based on the stl-tcn-nbeats model. *Heliyon*, page e13029, 2023.

Jun Zhang, Lan Li, and Wei Chen. Predicting stock price using two-stage machine learning techniques. *Computational Economics*, pages 1–25, 2020.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657, 2015.

Xiangrui Zhang, Chunyue Song, Jun Zhao, Zuhua Xu, and Xiaogang Deng. Deep sub-domain learning adaptation network: A sensor fault-tolerant soft sensor for industrial processes. *IEEE Transactions on Neural Networks and Learning Systems*, 2022a.

Xiaorui Zhang, Jie Zhou, Wei Sun, and Sunil Kumar Jha. A lightweight cnn based on transfer learning for covid-19 diagnosis. *Computers, Materials and Continua*, pages 1123–1137, 2022b.

Ke Zhao, Feng Jia, and Haidong Shao. A novel conditional weighting transfer wasserstein auto-encoder for rolling bearing fault diagnosis with multi-source domains. *Knowledge-Based Systems*, 262:110203, 2023.

Yiheng Zheng and Yisong Li. Unmanned retail's distribution strategy based on sales forecasting. In *2018 8th international conference on logistics, informatics and service sciences (LISS)*, pages 1–5. IEEE, 2018.

- Xiao Zhong and David Enke. Forecasting daily stock market return using dimensionality reduction. *Expert Systems with Applications*, 67:126–139, 2017.
- Siyu Zhou, Lin Zhou, Mingxuan Mao, and Xinze Xi. Transfer learning for photovoltaic power forecasting with long short-term memory neural network. In *2020 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 125–132. IEEE, 2020.
- Xu Zhou, Yong Zhang, Zhao Li, Xing Wang, Juan Zhao, and Zhao Zhang. Large-scale cellular traffic prediction based on graph convolutional networks with transfer learning. *Neural Computing and Applications*, pages 1–11, 2022.
- Xiaodan Zhu, Anh Ninh, Hui Zhao, and Zhenming Liu. Demand forecasting with supply-chain information and machine learning: Evidence in the pharmaceutical industry. *Production and Operations Management*, 30(9):3231–3252, 2021.
- Xiaotian Zhuang, Ying Yu, and Aihui Chen. A combined forecasting method for intermittent demand using the automotive aftermarket data. *Data Science and Management*, 5(2):43–56, 2022.