AI Aided Tools for Fresh Produce Yield and Price Forecasting: Deep Learning Approaches

by

Mohita Chaudhary

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2021

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

Following publications have resulted from the work presented in the thesis:

1. M. Chaudhary, M.S.Gastli, L.Nassar and F.Karray."Transfer Learning Application for Berries Yield Forecasting using Deep Learning" (Accepted to IEEE International Joint Conference on Neural Networks 2021)

2. M. Chaudhary, M.S.Gastli, L.Nassar and F.Karray."Deep Learning Approaches for Forecasting Strawberry Yields and Prices Using Satellite Images and Station-Based Soil Parameters." (Accepted to Association for the Advancement of Artificial Intelligence Spring Symposium (AAAI-MAKE) 2021)

3. M. Saad, M. Chaudhary, L.Nassar, F.Karray and V. Gaudet. "Versatile Deep Learning Based Application for Time Series Imputation." (Accepted to IEEE International Joint Conference on Neural Networks 2021)

4. M.Saad, M.Chaudhary, F.Karray and V. Gaudet. "Machine Learning Based Approaches for Imputation in Time Series Data and their Impact on Forecasting." (Accepted to IEEE International Conference on Systems, Man and Cybernetics 2020)

5. L.Nassar, M.Saad, I. E. Okwuchi, M.Chaudhary, F.Karray and K. Ponnambalam. "Imputation Impact on Strawberry Yield and Farm Price Prediction Using Deep Learning." (Accepted to IEEE International Conference on Systems, Man and Cybernetics 2020)

6. M.Chaudhary, M.Saad, L.Nassar and F.Karray . "Evaluation of Imputation Models Based on the Enhancement to Yield Forecasting." (Submitted to IEEE International Conference on Systems, Man and Cybernetics 2021)

7. M.Chaudhary, L.Nassar and F.Karray . "Deep Learning Approach for Forecasting Apple Yield using Soil Parameters." (Submitted to IEEE International Conference on Systems, Man and Cybernetics 2021)

Abstract

It is important to have an accurate estimate of the yields and prices of fresh produce (FP) for maintaining an effective Fresh Produce Supply Chain Management (FSCM). Since, the FP comprises of the perishable goods, it is cumbersome to manage and keep a track of logistics, which makes it important to have an estimate of the FP yield to have a better management of the supply and demand. In addition, having a reliable estimate of the FP prices helps the food company to bid the right price to the wholesalers. This prevents the food company from bidding unreasonable price and incurring any loss. Computational tools for forecasting yields and prices for fresh produce have been based on conventional machine learning approaches or time series modeling. These approaches can neither effectively capture the complex relationships between the inputs and the outputs to the models nor can they handle large datasets. To overcome such drawbacks, Deep Learning (DL) based approaches are proposed in this work for forecasting the yields and prices of FP. Soil and weather parameters of counties across California are used to forecast the yields and prices of FP like berries and apples.

Choosing the most effective input parameters for forecasting strawberry yields and prices is investigated. The set of parameters used for this investigation are soil parameters alone and soil parameters along with the weather parameters. For this forecasting, the ensemble of two DL models is used namely, Convolutional Neural Networks and Long Short Term Memory with Attention (Att-CNN-LSTM) and Convolutional LSTM with Attention (Att-ConvLSTM). It is found that using soil and weather parameters together gives better forecasting results than using soil or weather parameters alone. Also, various compound DL models like Att-CNN-LSTM, Att-ConvLSTM, Temporal Convolutional Network (TCN) and SeriesNet with Gated Recurrent Unit (SeriesNet-GRU) are tested for forecasting, to determine the best performing DL model. It is found that the ensemble of two compound DL models Att-CNN-LSTM and SeriesNet-GRU gives the best forecasting results with an improvement of around 7% in the value of Aggregated Measure (AGM) than the component compound DL models. It also outperforms the previous work done in literature with an improvement of around 14% in the value of AGM. The effect of using soil input parameters on yield forecasting is further studied. To study the effect of static soil parameters on forecasting performance, the compound DL model SeriesNet with GRU is used to forecast the annual apple yield using the static and dynamic soil parameters. The county level annual apple yield forecast, using both static and dynamic parameters together, proves to give promising results, it reduces the forecasting AGM by around 34% compared to the case of excluding the static parameter and only using the dynamic parameters set. It is also found that, on using an augmented training set to train the DL model improves the AGM value by around 12% on testing with the non-augmented test set.

To generalize the findings, transfer learning technique is utilized amongst the yield forecasting models of the similar crops. To overcome the computational complexity of retraining DL yield forecasting models for each type of FP, it is necessary to have a generalization of the models' application to similar FP with minimal retraining. Two berries are considered in this work, California strawberries and raspberries which have similar yield, since the two follow similar time series on the basis of a number of parameters such as lag, seasonality and trend. The voting regressor ensemble of two compound DL models Att-CNN-LSTM and SeriesNet with GRU is used. First, the proposed DL model is trained using station-based soil data input mapped to the strawberry yield as output. The weights obtained from this learning are transferred to the raspberry yield forecasting ensemble model with minimal retraining. It is found that the transfer learning gives comparable results to training from scratch and reduces the processing time by half.

Acknowledgements

Firstly, I would like to express my gratitude to my supervisor Professor Fakhri Karray. Under his guidance, I was able to get involved in fields of Deep Learning, Machine Learning and Time Series Forecasting. My Master's under Professor Fakhri Karray was filled with new experiences, lessons, support, and encouragement. I could not have accomplished this thesis without his valuable insights and feedbacks. I also thank him for giving me the opportunity to work on this research project. I would especially like to thank Dr. Lobna Nassar for her help and guidance with the experiment ideas, documentation, discussions and revisions throughout my master's. I also thank all my colleagues and team members: Mohamed Sadok Gastli and Muhammad Saad. I would also like to acknowledge their contributions: Mohamed Sadok Gastli's contribution with feature extraction from satellite images and Muhammad Saad's help with data imputation.

I also wish to thank the members of my thesis committee, Professor Kumaraswamy Ponnambalam and Professor Apurva Narayan for taking the time to provide an in-depth review of the thesis and provide very helpful feedback. Your efforts and support are deeply appreciated.

I acknowledge and am thankful for the financial support from Loblaw Companies Limited, Natural Sciences and Engineering Research Council, University of Waterloo and Electrical and Computer Engineering department.

I thank and dedicate this work to my father Mr Dinesh Chaudhary, my mother Mrs Sangeeta Chaudhary, my siblings Sujay Chaudhary and Mahima Chaudhary for their unconditional love, constant support and motivation that allowed me to pursue graduate studies. I would also like to thank my friends who brought immeasurable positive energy and enthusiasm in my life.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

R^2 Coefficient of Determination 30

AGM Aggregated Measure 32

AI Artificial Intelligence 2

ANN Artificial Neural Network 32

CIMIS California Irrigation Management Information System 36

CNN Convolutional Neural Network 28

CP-ANN CounterPropagation Artificial Neural Networks 33

CRD Crop Reporting Districts 40

DL Deep Learning 2

FP Fresh Produce 1

FSCM Fresh Produce Supply Chain Management 5

GADF Gramian Angular Difference Fields 14, 44

GAF Gramian Angular Field 14, 44

GASF Gramian Angular Summation Fields 14, 44

GRU Gated Recurrent Unit 10

ICA Independent Component Analysis 13

LDA Linear Discriminant Analysis 13

LLE Locally linear embedding 14

LOCF Last Observation Carried Forward 40

LSTM Long Short Term Memory 18

MAE Mean Absolute Error 29

MDS Metric Multidimensional Scaling 14

ML Machine Learning 2

MSE Mean Squared Error 29

MTF Markov Transition Fields 14

NOCB Next Observation Carried Backwards 41

PCA Principal Component Analysis 13

PDSI Palmer Drought Severity Index 38

PLS Partial Least Squares 32

RMSE Root Mean Squared Error 29

SDE Semidefinite Embedding 14

SKNs Supervised Kohonen Networks 33

t-SNE t-Distributed Stochastic Neighbor Embedding 14

UAV Unmanned Aerial Vehicle 32

USDA United States Department of Agriculture 3

USDA-WSS United States Department of Agriculture: Web Soil Survey 36

Chapter 1

Introduction

1.1 Problem Definition

Food wastage, or food shrinkage is a global ongoing issue with at least one third of all the globally produced food getting wasted or lost in the supply chain every year. In Canada, around $31 billion worth of food is wasted every year, with an average food loss of around 5% experienced by retailers in the Fresh Produce (FP) alone. The food business considers the food wastage as a cost of doing business but in reality it is a business opportunity. It is estimated that with every 1% reduction in food waste the revenue can be elevated by 4% [5] and not many opportunities offer this much of benefit. The food shrinkage can be attributed to the poor planning and mismanagement of Supply Chain. The food supply mainly depends on the demand for FP the distribution centers are expected fulfil the demand. However, making supply decisions are highly dependent on the unstable and fluctuating procurement prices. The procurement prices are also driven by the anticipated crop yields. In order to earn good revenue and avoid food shrinkage, the food companies are required to properly time and price the food demand.

The food pricing is driven by various factors related to supply and demand. The food supply or the yield depends mainly on the weather, type of soil, irrigation, quality of chemicals like fertilizers and pesticides used, technology deployed for cultivation etc. region wise and the demand is driven by the economic state of the place, the culture, weather and environment of that area. With the alarming climate change and globalization these factors are becoming even more uncertain. To understand the complex relationship between supply and demand driven by numerous factors and procure the right price is a cumbersome process. The FP business requires deployment of advanced technological tools to fulfil this challenging task of procuring right price and estimating right yield of the FP.

1.2 Motivation

Efficient procurement performance is vital to the food company's produce department performance, and to the overall profitability of the food company and its brand image. The category managers and buyers work together to forecast store level produce needs and to procure the right amount of fresh produce (fruits and vegetables). Category managers are responsible for planning sales and merchandising for the stores and work with buyers to foresee the produce needs for the food company [105]. The fresh procurement is a process of solicitation in which the food company makes a price bid to the distributor or wholesaler who then makes a decision to accept or reject the offer. Having a prior knowledge and estimate of the price which the distributors might be expecting can help the food company to make a fair bid and avoid over payment.

The Fresh Produce Supply Chain Industry which has traditionally been a comparatively low-tech, now the suppliers and retailers are increasingly turning towards technology to address fresh food supply chain challenges, including waste reduction, improving inventory management and bidding fair FP prices. Food companies believe AI's strongest potential to improve supply chain management is with the quality and speed of planning insights. To ease the strenuous process of FP procurement and get a prior estimate of FP price and yield, the food companies these days are banking on the new technological AI based tools for FP price and yield forecasting. The FP price forecasting for a long time has been done by estimating the future price values based on the historical price data. This process is not very efficient since it fails to utilize various other factors governing the FP pricing. The use of powerful ML and DL models combined with all the essential data affecting the price, like weather data, soil data, yield data etc. to forecast the FP procurement price is beneficial for the food company, farmers, wholesalers, distributors, retailers and consumers and hence beneficial for the society. The food company in Canada which has a year round revenue in billions with a revenue of $111.6 billion in 2018 [2], this industry accounts for high volume of transactions, even a little improvement in the every FP transaction would lead to benefits of millions of dollar every year for Canada.

The successful accomplishment of AI based tools is owed to the existence of a huge amount of data, the efficient AI models and easier access to computational resources, leading to the development of reliable decision tools providing with almost precise and reliable estimates of prices and yields. Using automated AI tools in food industry can eliminate the inefficient methods of estimating the prices and can boost the revenues of the food industry. It can also lead to an affordable pricing of food products for the benefit of consumers and lead to lesser food wastage.

1.3 Scope of Work

This work focuses on the development and implementation of Deep Learning models to accomplish the complex task of forecasting the price and yield of the FP commodities. Various models have been developed and used for forecasting price and yield using the weather data and the soil data for training purpose. The effect of different types of soil parameters have been studied on the yields of FP. The generalization of the models' application to other fresh produces have been achieved by implementing transfer learning with minimal retraining among the similar type of FP. The crops considered for this study are strawberries, raspberries and apples grown in California, United States. California is chosen since it is the leading state producer for strawberries and raspberries in the United States [1]. As of the survey from USDA for year 2017-2019 [12] California has over 13,000 bearing acres of apples with a yield value of around 24,000 pounds/acre throughout California's growing regions. A broad evaluation of these models was done using several metrics and the best models were identified.

1.4 Objective

The objective of this work is to build and test deep learning based forecasting models to obtain efficient procurement offer price for the bilateral transaction of the food company and an efficient estimate of the FP yield.

To reach the main objective, the following tasks are carried out.

(a) Data collection, Data Preprocessing and Feature Selection.

(b) Choosing the most efficient effective input parameters for yield forecasting.

(c) Building DL models and training them on the collected data for forecasting the price and yield.

(d) Choosing the best model based on model evaluation using various metrics.

(e) Performing transfer learning amongst the forecasting models of similar crops to achieve a generalized application.

(f) A forecasting web application combining all the work and findings in the thesis.

1.5 Thesis Organization

This thesis constitutes five chapters. The present chapter introduces and describes the challenges faced in the process of fresh produce procurement. It also throws light on the importance of efficient procurement of prices and how it is essential to the efficient supply chain management. This chapter also describes the scope and objective of the work. The Chapter 2 covers the literature and background of the problem. In Chapter 3, the proposed solution is described includes proposed models and proposed evaluation metric. Chapter 4 provides details of the various experiments carried out such as FP yield forecasting, FP price forecasting and oil price forecasting. Chapter 5 consists the details of the forecasting web application developed. Finally, Chapter 6 summarizes the major findings derived from this research work.

Chapter 2

Background and Literature

2.1 Fresh Produce Procurement

The Fresh Produce Supply Chain Management (FSCM) is a critical process due to the high risk of markdowns and wastage, making it highly important to estimate the demand and replenish in accordance with the demand. The planning process for fresh produce must be quite granular to capture any change in demand. The supply chain for fresh products must be agile enough to adjust to the changes in demand. Owing to this instability of demand and supply in the fresh produce market, deciding the farmers' price is quite a crucial task. One of the major concerns of farmers' these days, is the uncertainty associated with agricultural prices and markets, i.e., variations in global market conditions can lead to abrupt fluctuations in prices of agricultural produce at a local level [29]. The inability to decide the correct price for the FP commodity might lead to financial losses. The FP prices are strongly affected by the FP yields determining the supply and hence the availability of accurate yields values is also quite crucial.

Fresh produce industry typically has been a comparatively low-tech industry, but nowadays the suppliers and retailers are increasingly turning to technology to solve the fresh food supply chain challenges including waste reduction, deciding correct price, determining the yield and improving the inventory management. The retailers believe in Artificial Intelligence (AI) potential to enhance the supply chain management due to the quality and speed of planning insights [3]. AI when combined with right data can not only improve the management of the fresh food supply chain, but also can optimize the reduction of waste associated with overstocking and under-stocking. It can enhance the delivered product freshness and streamline inventory management.

Food loss is a serious issue in the current scenario which can be avoided by having an effective Fresh Produce Supply Chain Management. The FP loss at the retail and consumer levels represents resources invested in food production, like the water used for irrigation, land, labor cost, energy, agricultural chemicals like fertilizer and pesticides and other inputs to produce the food which does not meet its intended purpose of feeding people causing financial loss and hunger [37]. The United Nations World Food Programme in 2020 highlights the prevalence of undernourishment in around world between 2017-2019 and states that if the current trends continue, then the number of hungry people will reach 840 million by 2030 [11]. The presence of the Covid-19 pandemic might further elevate this number. Therefore, United Nations has included ending hunger as one of the major goals in the 2030 Agenda for Sustainable Development to promote food security [10]. The ability to reliably estimate the crop yields using AI forecasting models might help in overcoming these issues. Knowing the estimates of yield can help in understanding the amount of supply of food and hence, help in maintaining the balance and distribution of the supply fulfilling the demands across the world, preventing hunger, and enabling a sustainable development.

To enable the procurement professionals to negotiate the price of FP commodity, an amalgamation of market knowledge and intelligence is highly required to provide an estimate to the otherwise confidential business cost and pricing details. To improve transparency, the food industry should share the data. The confidential data containing the information on the historical procurement prices if provided by the Food Company and used together with the free source publicly available climate data can be used to train, validate and test the AI models and give an estimate of the price.

2.2 Time Series Modelling

A time series or time stamped data is a sequence of data recorded in successive order in a time frame. A time series can be a record of any variable changing periodically with any time frequency; every second, hourly, daily, weekly or yearly. Time series analysis is mainly done to understand the underlying structure and function that is responsible for producing the observations. Understanding the underlying mechanism enables the development of a mathematical model for the time series which explains the data in such a way that allows for prediction, monitoring, or control to occur [68]. The time series forecasting is widely used in economics and business. Furthermore, monitoring of ambient conditions of an input or output is common in science and industry while quality control is used in computer science and communications.

Time series analysis is not only useful for tracking how a particular variable changes over time but also used to scrutinize the changes associated with the chosen variable with respect to the shifts in other variables over the same time period [71]. The major goals of time series analysis are: descriptive analysis which identifies patterns in correlated data trends and seasonal variation, explanatory analysis which includes understanding and modeling the data, forecasting which enables prediction of short-term trends from previous patterns, intervention analysis which helps in analyzing how a single event changes the time series, and quality control in which deviations of a specified size indicates a problem. The focus of this thesis is on the time series forecasting. Time Series forecasting is a process of predicting the future values in a time series by analyzing the historical values and past patterns. Time Series forecasting involves analyzing the trend, seasonality, and cyclic fluctuations in a time series then applying a suitable method for forecasting. The very common statistical methods used for time series forecasting involve ARIMA, exponential smoothing, VARMAX, SARIMA, ... etc. [73, 42, 25]. These methods provide complementary approaches to the problem. The major difference between modeling data using statistical methods and deep learning methods is that statistical methods account for the fact that data points taken over time have an internal structure like auto-correlation, trend, cyclic frequency or seasonal variation which should be considered.

2.2.1 Time Series Types

There are mainly two categories of time series: univariate time series and multivariate time series. The characteristics of each type are detailed in this section.

2.2.1.1 Univariate Time Series

The univariate time series is the one that consists of a single observation recorded sequentially over equal time intervals. Contrary to other areas of statistics the univariate time series modelling contains lag values of itself as independent variables. These lag variables can play the role of independent variables as in multiple regression. An instance of a univariate time series model is Autoregressive Integrated Moving Average (ARIMA) [33]. Univariate methods include time series forecasting methods which rely on the historical data to predict the future [53]. These models include naive forecasting methods, which assume that the next observation is similar to the previous one and another basic method which assumes that the future observation is equal to the current observation plus a basis value. A univariate time series may possess the following characteristics like auto-correlation, seasonality, trend and stationarity. Auto-correlation is the Pearson correlation of a signal with a delayed copy of itself as a function of delay. It can be defined as the similarity between observations as a function of the time lag between them. Stationarity is a characteristic occurring when the statistical properties do not change over time. This type of time series has constant mean and variance which is independent of time. The properties of a stationary time series do not depend on time. A time series with a trend or seasonality is non-stationary. Stochastic time series can be both: stationary e.g. the auto-regressive (AR), moving average (MA), and auto-regressive moving average (ARMA) [101] and the non-stationary like the generalized autoregressive conditional heteroskedastic (GARCH) and the auto-regressive integrated moving average (ARIMA) [63, 64].

2.2.1.2 Multivariate Time Series

he multivariate time series model is an extension of the univariate time series case and involves two or more input variables. It is not only limited to the past information of one variable but also incorporates the past of other variables as well. The multivariate analysis constitutes several related time series being observed simultaneously over time, instead of observing a single series as in the univariate case [93]. This analysis emerged in quest of studying the inter-relationship amongst multiple time series variables. These relationships are mostly examined through the consideration of the correlated structures among the component series. A Vector Autoregression (VAR) model is a generalization of the univariate autoregressive model for forecasting a vector of time series [26]. This method consists of one equation per variable in the system. The right-hand side of each equation includes a constant and lags of all of the variables in the system. Following are the equations for a two variable (2 dimensional) VAR with one lag VAR(1):

$$y_{1,t} = c_1 + \phi_{11,1} y_{1,t-1} + \phi_{12,1} y_{2,t-1} + e_{1,t} \tag{2.1}$$

$$y_{2,t} = c_1 + \phi_{21,1} y_{1,t-1} + \phi_{22,1} y_{2,t-1} + e_{2,t} \tag{2.2}$$

where $e_{1,t}$ and $e_{2,t}$ are the white noise processes that may be contemporaneously correlated. The coefficient $\phi_{ii,L}$ captures the influence of the Lth lag of variable y_i on itself, while the coefficient $\phi_{ij,L}$ captures the influence of the Lth lag of variable y_j on y_i. The multivariate time-series models involve a large number of unknown parameters which might introduce non linearities and elevate complexity. One of the easiest solution would be extending univariate non-linear models to the multivariate analysis. It is difficult to conclude that which is the best approach without experimentation.

2.2.2 Components of Time Series

The value of an observation in a time series are dependent on various forces which are known as the components of a time series. The four main categories of the components of time series are trend, seasonal variations, cyclic variations and random variations. Out of these seasonal and cyclic variations are the periodic changes or short-term fluctuations.

2.2.2.1 Seasonal variations

The seasonal variations are the rhythmic forces which act in a regular and periodic manner over a span of less than a year [78]. They have a almost similar, repetitive pattern during the period of twelve months in a year. This sort of variation is observed in a time series if the data is recorded on hourly, daily, weekly, quarterly, or monthly basis. These variations are mostly the outcome of some natural forces or man-made conventions. The changing seasons and climatic conditions play a vital role in seasonal variations. For example, as crop yield depends on seasons, during the monsoon the sale of umbrella increases and during the summer the sale of air-conditioner increases. The man-made factors like festivals, trending fashions, etc also effect the time series [87].

2.2.2.2 Cyclic Fluctuations

The cyclicity are components which tend to repeat themselves over a certain period of time. They occur in a regular spasmodic manner. The variations in a time series which act over a span of more than one year are known as the cyclic variations (seasonal variations are within a year). This oscillatory movement has a time period of more than a year. One complete period is a cycle. This cyclic movement is also referred to as the 'Business Cycle'. This cycle can be divided into four-phase cycle covering the phases of prosperity, recession, depression, and recovery. The cyclic variation which are regular are not periodic. The rise and the fall in any business depend upon the economic forces as well as the interaction between them [71].

2.2.2.3 Secular Trend

The trend depicts the general nature of the data to increase or decrease during a stretch of time. A trend is a general long-term average tendency. Moreover, it is not always important that the increase or decrease is in the similar direction over the given period of time. The tendencies might increase, decrease or remain stable in different stretches of time. Although, the overall trend of the time series should either be upward, downward, or stable. A few examples of time series with trends are the population curve, the food production, number of industries in a sector etc. The trend can be a general systematic linear or nonlinear component of a time series that changes over time and does not repeat [68].

2.2.2.4 Irregular Variations

Another factor contributing to the variation in the time series under examination are these random and irregular factors. These variations are completely irregular or random. The fluctuations caused by these random factors are erratic, unforeseen, uncontrollable and unpredictable. The example of such forces are natural calamities, some sudden event in a time frame which causes random behavior of the variable under study [123].

2.3 Data Preprocessing

2.3.1 Data Imputation

It is normal for a time series to have missing values and it is important to fill these missing values before fitting any model for forecasting or prediction. Time series imputation is a challenging task due to the existence of non-linear dependencies between current and past values. Missing values imputation or interpolation is a progressive research area with various existing methods to deal with it. The most trivial method to tackle this issue is to discard incomplete or empty records. Common methods include the list wise deletion (complete-case analysis)[121], removing all instances with at least one missing value as well as the pairwise deletion (available-case analysis) which uses cases that contain some missing data [76]. These methodologies introduce bias in the data and reduce the amount of data fostering incorrect forecasting. Moreover, methods like linear interpolation are considered primitive [62]. Although, this method fits a smooth curve to the given dataset and fills the missing values using local interpolations, this method fails to take into consideration the dependencies of features over time. Statistical methods like SARIMA, ARIMA [135, 88] are auto-regressive and do not perform well when the time series contains consecutive missing values.

Recurrent Neural Networks (RNN) have also been used to impute the missing values [47, 100]. A modified version of GRU, GRU-D is proposed in [41] which imputes the missing values in a health care time series data. It assumes that missing value can be represented as a combination of the last available value and the global mean. The GRU-D model has aided in harnessing the power of the RNNs and explanation of the missing patterns in a time series. LIME-RNN is another state of the art algorithm which uses residual networks and graph-based temporal dependency in imputation. This model gives a good performance on various datasets [125]. It introduces a linear memory vector called the residual sum vector (RSV) and integrates it over previous hidden states of the RNN, to fill the missing values.

2.3.2 Exponential Smoothing

Data smoothing is performed on a time series to remove the noise and outliers from it, which enables important patterns to clearly standout. Data smoothing also helps in predicting trends clearly and it also helps in taking the account of seasonality. The data smoothing models include moving average methods. Although data smoothing might help in predicting certain trends but it also inherently leads to loss of information. There are different methods to perform data smoothing like the randomization method which uses random walk [127], moving average method [126] , or performing exponential smoothing techniques [77].

Exponential smoothing assigns exponentially decreasing weights from latest to the oldest observations. This simply means that the older the data the lesser is the weight or importance given to that data and the newer data is assigned more weight since it is more relevant. The smoothing parameters α denotes the weights for observations. Exponential smoothing is mostly used for short term forecasts since longer term forecasts using exponential smoothing can be quite unreliable. The Simple exponential smoothing [81] uses a weighted moving average which has exponentially decreasing weights. The double exponential smoothing (Holt's trend) [63] is suited for tackling the data that shows trends, compared to the single procedure. The triple exponential smoothing (Multiplicative Holt-Winters) [39] is suitable for time series with both trend and seasonality. In this thesis, the multiplicative Holt-Winters method of exponential smoothing is used due to the existing seasonality in the data.

2.3.3 Feature Selection

In this work, the yield and price are forecasted using the weather and soil data as input. The problem with soil and weather data is the presence of highly correlated parameters which are often redundant. It is important to omit highly correlated features; this process is known as feature selection. Feature selection involves selecting and excluding given features without processing them [108]. There are various approaches for feature selection like removing the parameters with high number of missing values, unfortunately this is not preferable as it may lead to the loss of essential information. But if the features consist of missing value beyond a specific threshold, they are preferred to be dropped. Features with low variance can also be dropped since they do not add essential information to the model. This can be done by using Variance Threshold which is a part of sklearn's feature selection module [119]. Using this method, the features with variance below a threshold or features with similar values and zero variance can be eliminated. The correlation between features can be determined by plotting the correlation matrix and the features which are highly correlated, positively or negatively, can be dropped. Univariate feature selection can be used, it works by selecting the features on the basis of some univariate statistical tests. The sklearn's SelectKBest [36] is used to select the top number of features to keep. It uses statistical tests to select features having the highest correlation to the target. Recursive feature selection can be used for eliminating the least important features. It continues recursively until the specified number of features are chosen. Recursive elimination can be used with any model that assigns weights to features [155]. The models like Random Forest Classifier [107] and Random Forest Regressor [98] can be used for feature selection using SelectFromModel [84]. Other methods include: Forward feature selection which starts with a full feature set then greedily removes features one at a time. Backward feature selection [103] which starts with an empty set and greedily adds or removes features one at a time. Backward stepwise elimination which starts with a full feature set then greedily adds or removes features one at a time. Random mutation which starts with a feature set containing randomly selected features then adds or removes randomly selected features one at a time and stops after a given number of iterations [108].

2.3.4 Sliding Window Method

A time series data consists of series of values which keep changing with respect to time. To perform supervised learning using time series data, it is essential to convert this data from a series form to a supervised form, which can be easily fed into a DL model. This is achieved by using the sliding window method. The sliding window method is performed to use the previous time steps to predict the future time steps. It is referred to as the lag method in statistics [139]. The lag method is the one that offsets or lags a time series such that the lagged values are aligned with the actual time series. The lags can be shifted by any number of units, which simply controls the length of the backshift. Lags are very essential in any time series analysis due to the existence of autocorrelation, which is a tendency for the values within a time series to be correlated with historical values of itself. Autocorrelation helps in identifying patterns within the time series, which helps in determining seasonality, the tendency for patterns to repeat at periodic frequencies. eriodic frequencies.

2.3.5 Dimensionality Reduction

Dimensionality Reduction is different from Feature Selection as it transforms the data unlike the feature selection which is simple elimination of redundant features. There are various methods used for dimensionality reduction like Linear Discriminant Analysis (LDA)in which d linear combinations of the n input features, where d < n, and the linear combinations are produced to be uncorrelated to maximize class separation [82]. These discriminant functions become the new basis for the dataset. All the numeric features in the dataset are projected onto these linear discriminant functions moving the dataset from the n dimensionality to the m dimensionality. In LDA, the maximum number of reduced dimensions is equivalent to the number of classes minus one. The assumption behind LDA is that the target classes have a multivariate normal distribution with similar variance but different mean for every class. Principal Component Analysis (PCA) is also widely used for the purpose of dimensionality reduction [75, 86]. PCA is a statistical method in which the n numeric dimensions of dataset are orthogonally transformed into a new set of n dimensions referred to as Principal Components. The first principal component depicts the highest variability of the data or variance. Every succeeding principal component has the constraint that it is orthogonal, which means uncorrelated to the preceding principal components and has the highest succeeding variance. Given a set of data on n dimensions, PCA aims to find a linear subspace of d dimension lower than n (d < n) such that the data points lie mainly on this linear subspace. The reduced subspace attempts to maintain most of the variability of the data. The PCA transformation is sensitive to the scaling of original features and it is important to normalize the data before applying PCA. Independent Component Analysis (ICA) is also one of the commonly used dimensionality reduction techniques [51]. It has its basis in information-theory. ICA is different from PCA as PCA looks for uncorrelated principal components whereas ICA looks for independent components. Uncorrelated variables are those with no linear relation among them, whereas independent variables those ones whose values don't depend on one another and thus form independent components in ICA. Autoencoders can also be used for dimensionality reduction. It is a neural network in which the output units are equal to n, the number of input units [129]. It has at least one hidden layer with d number of nodes such that $d < n$. They are trained using the backpropagation algorithm and the input vector is reproduced on to the output layer. The n features in the data are reduced, using the output of the hidden layer, to represent the input vector. The autoencoder consists of both an encoder from the input layer to the hidden layer and a decoder: The encoder compresses the n dimensions of the input dataset into d dimensional space. The decoder expands the data vector from d dimensional space into the original n dimensional dataset. It is used to bring back the data to original value. There are other dimensionality techniques which handle non-linear patterns in datasets like Kernel PCA [150] which finds the principal components that are nonlinearly related to the input space by performing PCA in the space produced by the nonlinear mapping, in the cases where the low-dimensional latent structure is easier to discover. Approaches like Locally linear embedding (LLE) [130], Laplacian Eigen Maps [31], Metric Multidimensional Scaling (MDS)[52], Isomap [28], Semidefinite Embedding (SDE) [148] and t-Distributed Stochastic Neighbor Embedding (t-SNE) [141].

2.3.6 Gramian Angular Field Imaging

The researchers proposed a novel framework for encoding time series as images after being inspired by the huge success of DL in computer vision [147]. The time series are encoded into various forms of images like the Gramian Angular Summation Fields (GASF), Gramian Angular Difference Fields (GADF) and Markov Transition Fields (MTF). Converting time series into images helps in using the computer vision algorithms on the data. Using Gramian Angular Fields for time series forecasting has given very promising results [147, 74, 30]. The Gramian Angular Field (GAF) represents the time series in a polar coordinate system instead of representing in a Cartesian coordinate system. In a Gramian matrix every element is the cosine of the summation of angles. Suppose there is a time series $X = x_1, ..., x_n$ for some n values, all the values are rescaled between [-1,1] or [0,1]. The rescaled time series $\tilde{X}$ can be represented in polar coordinates by encoding the value as the angular cosine and the time stamp as the radius using the following mathematical equation:

$$\begin{cases} \phi = arcross(\tilde{x}_i), & -1 \leq \tilde{x}_i \leq 1, \tilde{x}_i \in \tilde{X} \\ & r = \frac{t_i}{N}, t_i \in \mathbb{N} \end{cases}$$

where, t_i is the time stamp, N is a constant factor to regularize the polar coordinate system, which is a novel way to understand time series. Visually, as the time increases the values in the time series warp among different angular points on the spanning circles similar to water rippling. The equation above is bijective since $\cos(\phi)$ is monotonic when $\phi \in [0, \pi]$. For every unique time series the proposed map produces unique result in the polar coordinate system with a unique inverse map. The polar coordinates also preserve the temporal relations unlike the Cartesian coordinate.

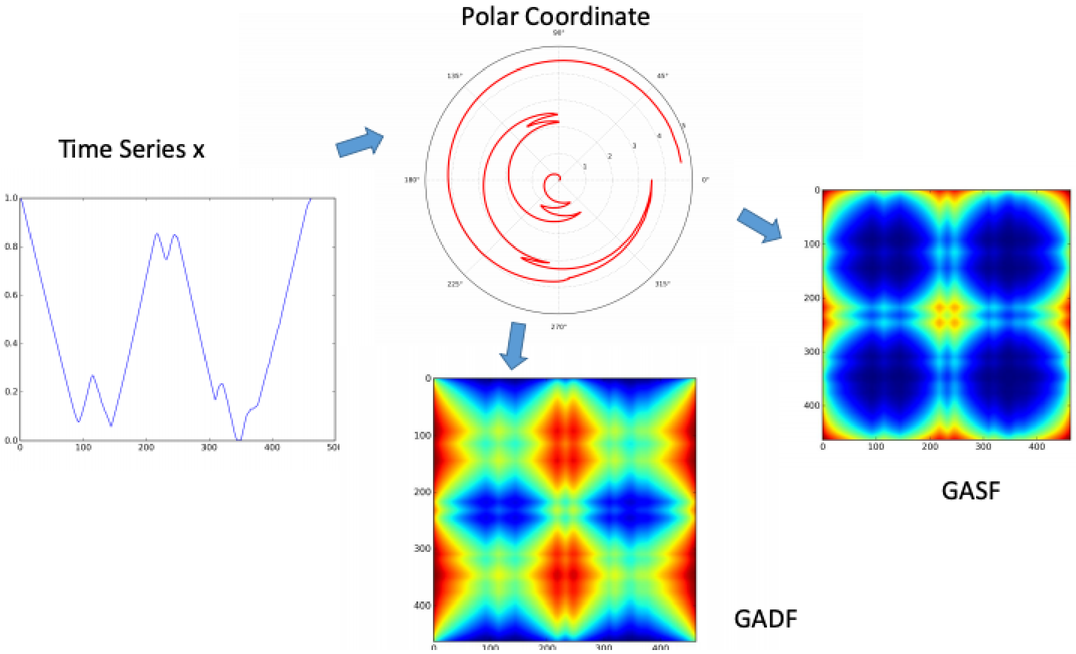


Figure 2.1: The Proposed Encoding Map of Gramian Angular Fields [147]

After rescaling and transforming the time series into polar coordinates, the angular perspective can be exploited by considering the trigonometric sum and difference between every point to find the temporal correlation within different time intervals. The GASF and GADF can be defined as follows:

$$GASF = [\cos(\phi_i + \phi_j)] = \tilde{X}'.\tilde{X} - \sqrt{I - \tilde{X}^{2\prime}}.\sqrt{I - \tilde{X}^2} \tag{2.3}$$

$$GADF = [\sin(\phi_i - \phi_j)] = \sqrt{I - \tilde{X}^{2\prime}}.\tilde{X}' - \tilde{X}.\sqrt{I - \tilde{X}^2} \tag{2.4}$$

where, I is the identity matrix and post transformation of the time series into the polar coordinate system the time series at each time step is take as a 1-D metric space. The pipeline for GAFs generation is shown in Figure 2.1. The figure depicts a sequence of the rescaled time series of 'Fish' dataset. The time series is transformed into a polar coordinate system and GASF/GADF is calculated. The size of the GAFs is sometimes large because the size of matrix becomes nxn when the size of time series sequence is n and to deaql with it the Piecewise Aggregation Approximation (PAA) [91] is used to smooth the time series, this process doesn't affect the trend and preserves it.

2.4 Deep Learning Models

Deep Learning is a subset of Artificial Intelligence which imitates the functioning of the human brain by processing the data and creating patterns for making future decisions accordingly like detecting objects in images, recognizing speech, translating languages, and making decisions [94, 49]. Deep learning has networks which do unsupervised learning from unstructured or unlabeled data. Deep Learning, also referred to as deep structured learning or differential programming, can be used to help detect fraud or money laundering. Deep learning has evolved with the digital era where a huge amount of data in every possible form, from all across the globe, became readily available. This data is referred to as big data and is derived from sources like social media, internet search engines, e-commerce platforms, and online cinemas. Big data is easily accessible and can be shared across cloud computing platforms. In its crude form, big data is unstructured hence it is beyond human comprehension to directly deduce results or extract relevant information from it. Here comes the role of AI tools and its subsets like Deep Learning and Machine Learning which have the potential to unravel the immense information hidden in the big data. Representation learning is a set of methods that allows a machine to be fed with raw data and to automatically discover the representations needed for detection or classification [67, 32]. There are numerous types of deep learning networks which are used these days like the Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), Deep Belief Networks (DBN), and Artificial Neural Networks (ANNs) which have been applied to numerous fields on the basis of application and specialty of each network.

2.4.1 Recurrent Neural Network (RNN)

The RNNs are similar to traditional time series models as they are both able to model time dependent relationships in the data. Every node in a given layer in RNN is connected with a directed, one-way, connection to every other node in the next successive layer. Every node has a time varying activation function and each connection has a modifiable real-valued weight. Nodes are categorized as either input nodes which receive data from outside the network, the output nodes which yield the results, or hidden nodes which modify the data from the input layer to the output layer. The nodes with recurrent edges receive input from the current data value (x_t) and also from h_{t-1} which is the hidden node value in the network's previous state at time t. The Figure 2.2 shows the rolled structure of RNN.

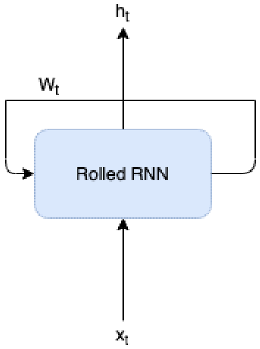


Figure 2.2: Structure of RNN

The output $\hat{y}_t$ is transferred to hidden node at h_t at time t. This means that the input x_{t-1} at time $t-1$ influences the output $\hat{y}_t$ at time t and later by way of recurrent connections [99]. The following equations represent the workflow of RNNs:

$$h_t = \sigma(W_{hx}x_t + W_{hh}h_{t-1} + b_h) \tag{2.5}$$

$$\hat{y}_t = softmax(W_{yh}x_t + b_y) \tag{2.6}$$

σ is the sigmoid function, W_{hx} are weights between the input and the hidden layer and W_{hh} are weights between the hidden layer and itself at adjacent time step. The vectors b_h and b_y are bias parameters which help neural networks to learn the offset. The output $\hat{y}_t$ is the predicted value of the sequence.

2.4.2 Long Short Term Memory (LSTM)

The Long Short Term Memory are developed mainly developed to handle the problem of vanishing gradients in the Recurrent Neural Networks [143]. LSTMs are quite similar to RNNs with hidden layers except that each node in the network is replaced by a memory cell. The memory cells ensure that the gradient passes across many time steps without vanishing or exploding by having certain nodes with self-connected recurrent edge of a fixed weight one. The LSTMs mainly have 3 gates, the input gate, forget gate and output gate [66]. A typical structure of an LSTM is shown in Figure 2.3

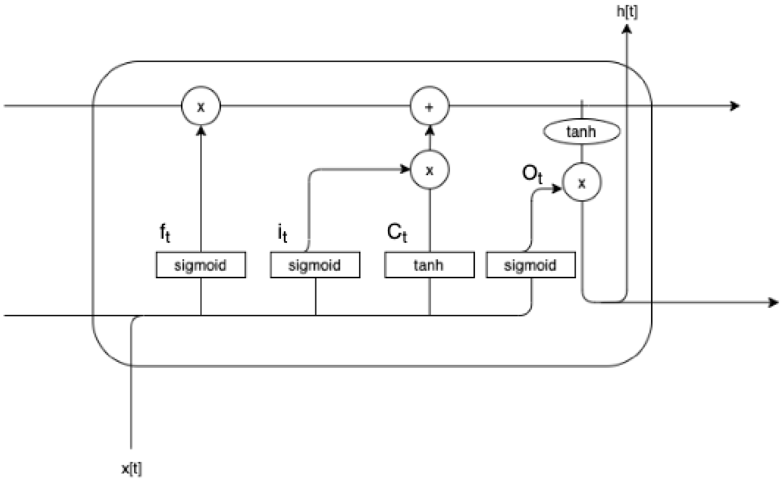


Figure 2.3: Structure of LSTM

The input gate discovers which values from the input should be used to modify the memory. Sigmoid activation function transforms the values to be in range 0 to 1 and tanh function gives weights between -1 and 1 to the input values depending on their level of importance. The working of input can be defined in following equations:

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{2.7}$$

$$C_t^{`} = \tanh(W_c[h_{t-1}, x_t] + b_c) \tag{2.8}$$

where, b is the bias, W are the weights, the cell state at time stamp t is $C_t^{`}$ and h_{t-1} is the hidden state of the previous LSTM block at time stamp $t-1$. The forget gate (f_t) are used to eliminate the contents in the internal state of LSTM and were introduced by were introduced by Gers et al. [65]. This gate takes in both the current input and values from the previous hidden state.

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{2.9}$$

The output gate produces the output value o_t which is ultimately used to produce h_t by the memory cell by multiplying o_t with the value of the internal states The internal state must be first run through a tanh activation function, since it provides the output of each cell which has the same dynamic range as an ordinary tanh hidden unit. This is contrary to the other neural network which use rectified linear units (RELU), which are easier to train and have a greater dynamic range.

$$O_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{2.10}$$

$$C_t = f_t * C_{t-1} + i_t * C_t^{'} \tag{2.11}$$

$$h_t = O_t * \tanh(C_t) \tag{2.12}$$

where, C_t is the hidden state of the cell at time t.

2.4.3 Gated Recurrent Unit (GRU)

GRU is a modified version of the RNNs and are designed to solve the problem of vanishing gradients which is encountered by the RNNs [48] & [55]. GRUs and LSTMs are similar [61] since both of them are designed to overcome the short term memory issues faced by RNNs and in many scenarios give equally good results; the only difference is that the GRUs use hidden states instead of using the cell state or memory for transferring the information. To solve the problem of vanishing gradients, GRU uses two gates, a reset gate and an update gate, which are the two vectors responsible for deciding what information needs to be passed on to the output. These gates can be trained to keep information from the long past through time without removing it or eliminating information which is irrelevant to the prediction. The function of the update gate is quite similar to the forget gate and input gates of LSTM, whereas the reset gate decides how much past information needs to be forgotten. Owing to the fewer tensor operations, the GRUs are faster than LSTMs. Figure 2.4 shows the structure of the GRU.

On the other hand, LSTMs can easily perform unbounded counting which helps in generalizing far beyond the training set, while the GRUs cannot therefore it is proven that LSTMs are strictly stronger than GRUs [149]. The mathematical representation of GRU is as follows:

$$z_t = \sigma_g(W_z x_t + U_z h_{t-1} + b_z) \tag{2.13}$$

$$r_t = \sigma_g(W_r x_t + U_r h_{t-1} + b_r) \tag{2.14}$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tanh(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h) \tag{2.15}$$

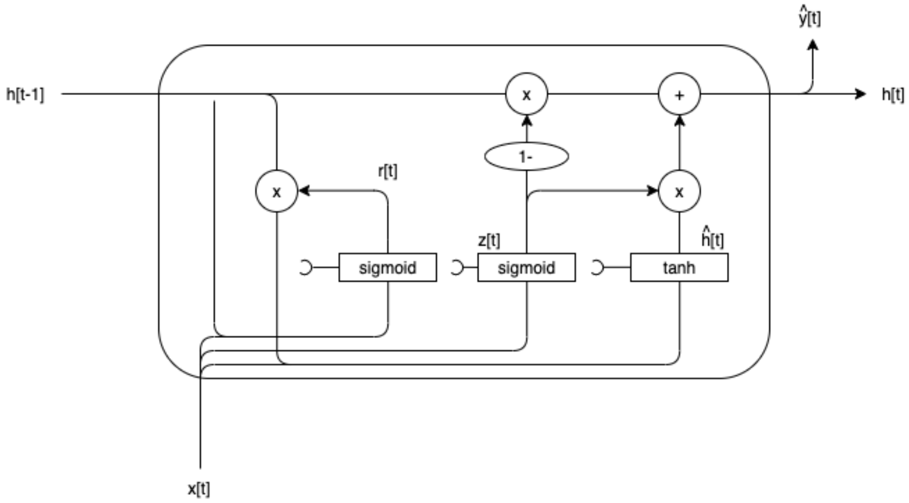


Figure 2.4: Structure of GRU

Where x_t is the input vector, z_t is the update gate vector, r_t is the reset gate and h_t is the output vector vector. σg is the element-wise sigmoid activation function applied individually to every element of the vector and W, U are parameter matrices and b is a parameter vector and $\odot$ is the Hadamard product. The sigmoid and tanh functions here are performed individually on all the elements of the vectors therefore the outputs of the equations are also vectors.

2.4.4 Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNN) or ConvNet are widely applied to image recognition and analysis. CNNs use shared weights and translation invariance, therefore they are also referred to as shift invariant or space invariant networks. If every pixel in an image is moved the same number of times in the same direction, the image would still retains its original property and is recognized as the same thing this is known as Translation invariance. For a time series data this implies that the patterns in sequence would be recognized irrespective of where in the sequence these patterns appear [14]. CNNs are used in a variety of tasks like the recommender systems [140], image detection [145], natural language processing [50], time series modelling [137],.. etc. The CNNs are simply the artificial neural networks, that in at least one of their layers, use convolution in place of general matrix multiplication [67]. CNNs have an inputlayer, an output layer, as well as several hidden layers. The hidden layers of a CNN consist of convolutional layers which convolve with a dot product. CNNs tackle overfitting in the fully connected networks by using the method of weight sharing and this is how CNNs provide a regularized version of fully connected networks [20]. The CNNs most commonly use the ReLU activation function. They also have other layers like pooling layers, fully connected layers and normalization layers. The Figure 2.5 represents the architecture of the CNN.

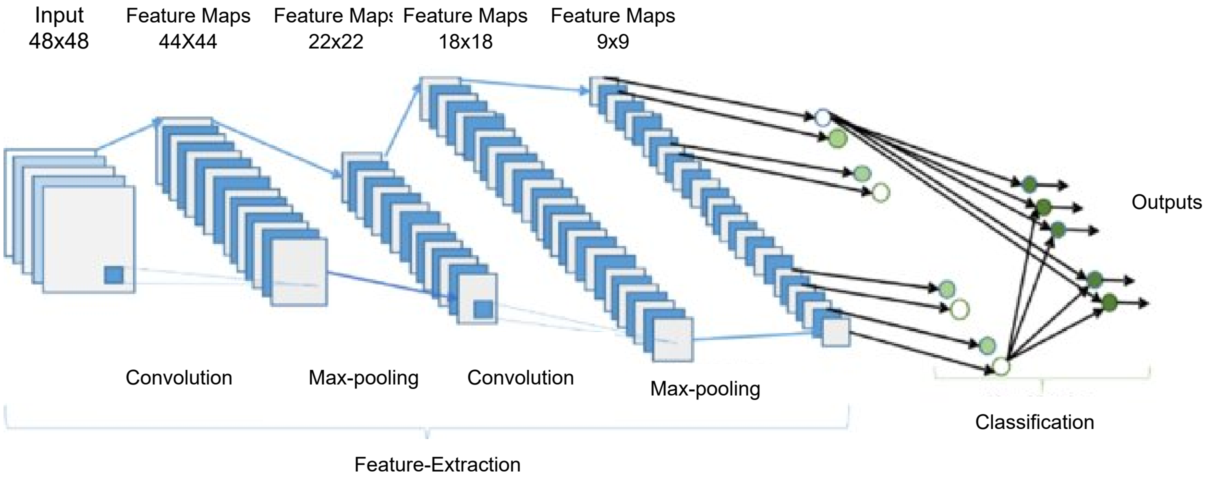


Figure 2.5: Architercture of Convolutional Neural Network [23]

The convolutional layer has convolutional kernels defined by a width and height, input and output channels, convolution filter depth which is equal to the depth of the input feature map. For the time series, the kernel is passed over the one dimensional input sequences created using a sliding window. The sliding window converts a time series to a supervised learning problem. The input sequence shape is converted to a tensor (number of sample sequence) x (length of each sample sequence) x (1) x (1). On performing convolution, the sequence becomes a feature map with dimensions (number of sample sequences) x (length of feature map) x (height of feature map (1)) x (channels of feature map (1)). It is to be noted that this is a 1D convolution. The filter applies a generic nonlinear transformation on the time series and the output is another modified time series. The same weights in convolution filter are used across all time steps which makes CNN learn weights that are invariant across time unlike in ANNs. The pooling layers can be local pooling layers or global pooling layers and are used for the purpose of dimensionality reduction. It is done by combining several outputs to create a single output. The global pooling acts on all the neurons of the convolutional layer whereas the local pooling combines small clusters. The pooling can also be max pooling or average pooling which output the maximum value or average value of the cluster respectively.

2.4.5 Convolutional LSTM (ConvLSTM)

Convolutional LSTMs (ConvLSTMs) are created to combine the plus points of both the LSTMs which is to tackle the temporal relationship and CNNs which is to tackle the spatial relationship. The ConvLSTM can tackle the spatiotemporal relationships in any data [133]. The spatial relationships in any time series data refers to the patterns that exist based on the location of one data point relative to other points whereas, the temporal relationships are patterns represented by a function of the sequential time based order of the data points. The ConvLSTMs model the spatio-temporal structures by encoding the spatial information into tensors and overcoming the limitation of the vector-variate representations as encountered in the LSTMs, where no track is kept of the spatial information isn't kept track of [146]. The inputs $x_1, .., x_t$, cell outputs $c_1, .., c_t$, the hidden states $h_1, .., h_t$ and gates i_t, f_t, g_t, o_t are the 3- D tensors in $\mathbb{R}^{pXmXn}$, where p dimension depicts the number of measurements or feature maps and 'm x n' depicts the spatial dimension; see Figure 2.6

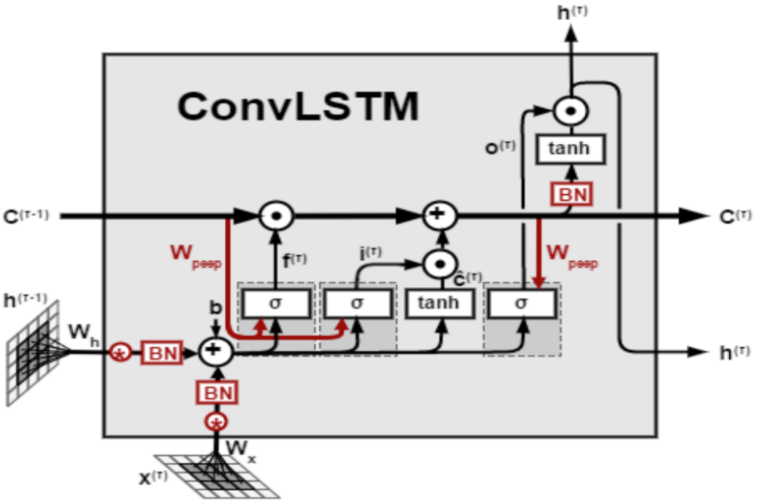


Figure 2.6: A ConvLSTM Cell [7]

In this research work, to tackle the time series data the first dimension 'p' is the number of input samples, the second dimension 'm' is the sequence length of each input sample and the third dimension 'n' is taken as 1. We can visualize the inputs as the vectors standing on the spatial grid. The ConvLSTM helps in finding the future state of certain cells in the m x n grid by utilizing the inputs and past states of the neighbours of the current point. This is done by using convolution operators in the state-to-state and input-to-state transitions. ConvLSTM follows the encoder-decoder recurrent neural network architecture as proposed in [136] and it is also used in video prediction in [114] The mathematical equations representing the working of ConvLSTM are as follows:

$$g_t = \tanh\left(W_{xg} * x_t + W_{hg} * h_{t-1} + b_g\right) \tag{2.16}$$

$$i_t = \sigma(W_{xi} * x_t + W_{hi} * h_{t-1} + W_{ci} \odot c_{t-1} + b_i) \tag{2.17}$$

$$f_t = \sigma(W_{xf} * x_t + W_{hf} * h_{t-1} + W_{cf} \odot c_{t-1} + b_f) \tag{2.18}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \tag{2.19}$$

$$o_t = \sigma(W_{xo} * x_t + W_{ho} * h_{t-1} + W_{co} \odot c_t + b_o) \tag{2.20}$$

$$h_t = o_t \odot \tanh\left(c_t\right. \tag{2.21}$$

The ConvLSTM cell with a larger transitional kernel captures faster motions whereas ConvLSTM with smaller kernel captures slower motions [133]. In above equations, $*$ is the convolution operator, σ is the element-wise sigmoid activation function and $\odot$ is the Hadamard product. The use of the input gate vector i_t, forget gate vector f_t, output gate vector o_t, and input-modulation gate vector g_t controls information flow across the memory cell vector c_t. Thus, preventing the problem of vanishing gradients by getting trapped in the memotry. The sigmoid and tanh activations also work upon the components of vectors individually.

2.4.6 Temporal Convolutional Networks(TCN)

Usually, the sequence modeling is dealt with using recurrent networks but the recent work indicates that the convolutional networks can outperform the recurrent networks on various DL problems like machine translation, audio synthesis,...etc. It is concluded that the convolutional neural networks must be regarded as the starting point for any sequence modelling task and the common association between sequence modelling and recurrent networks should be reconsidered. A generic network of Temporal Convolutional Network is used to represent the convolutional networks. This architecture is derived from the recent research and is kept simple; it combines the advantages of modern convolutional architectures. When compared to the canonical recurrent architectures, like GRUs and LSTMs, the TCNs outperform them across sequence of modelling tasks [27]. The recurrent networks have the memory retention characteristics, and the TCNs exhibit substantially longer memory and are suitable for capturing longer history [97]. The distinct characteristics of TCNs are that the convolutions are causal ensuring that there is no information leakage from future to past also it can take any past sequence of a given length and map it to future sequence of same length like the RNNs. The TCNs help to build long effective history sizes using a very deep networks augmented with residual layers and dilated convolutions. Moreover, the TCN is much simpler than WaveNet [113] as it has no skip connections across layers, conditioning, context stacking, or gated activations.

Sequence Modelling As the name suggests sequence modelling helps in modelling an input sequence $x_o, ..., x_t$ and predicts an output sequence $y_o, ..., y_t$ for every time sequence t. To predict the output y_t for some time t, the key constraint is that all the previously observed values of inputs $x_o, ..., x_t$ are used. Formally, a sequence modeling network is any function $f : \mathcal{X}^{T+1} \longrightarrow \mathcal{Y}^{T+1}$ that produces the following mapping:

$$\hat{y}_o, ..., \hat{y}_T = f(x_o, ..., x_T) \tag{2.22}$$

The Equation 2.22 satisfies the causal constraint which shows that the current output values do not depend on future values. TThe sequence modelling follows the autoregressive prediction in which the future is predicted based on the past by shifting the target output to the input shifted by one time step.

Causal Convolutions TCNs are based mainly on two principles: First, the network produces an output sequence with the same length as the input sequence. Second, there is no leakage from the future into the past. In order to achieve the first principle, the TCN uses a 1D fully convolutional network architecture (FCN) [131],in which every hidden layer has same length as the input layer; to maintain the same length, zero padding is added with length of one minus the kernel size . For the second principle, TCN uses causal convolutions, where an output at time t is convolved only with elements from time t and elements in the previous layer. Thus, TCN comprises of 1D FCN and causal convolutions. One of the disadvantages of this network is that to achieve a long effective history an extremely deep network is needed along with large filters which were not feasible initially.

Dilated Convolutions A basic causal convolution can look back at past values with size linear in the depth of the network thus, making it challenging to apply the causal convolution on sequence tasks requiring longer history. This can be dealt with employing

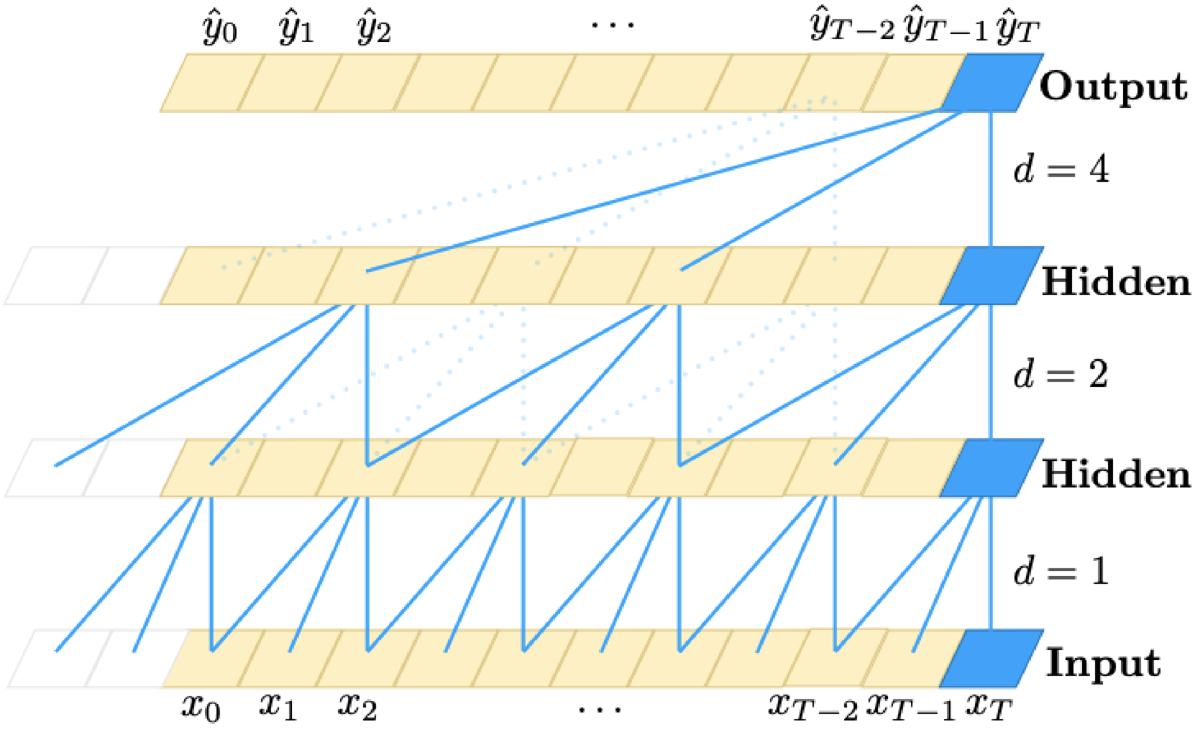


Figure 2.7: Dilated Causal Convolution [27]

the dilated convolutions which enable a large receptive field [113, 158]. Figure 2.7 represents a dilated causal convolution which has dilation factors d = 1, 2, 4 and filter size k = 3. The entire input sequence is captured by the receptive field. Thus, for a 1-D sequence input $x \in \mathbb{R}^n$ and filter $f : 0, ..., -1 \longrightarrow \mathbb{R}$. The dilated convolution operation 'F' on element 's' can be represented mathematically as follows:

$$F(s) = (x *_d f)(s) = \sum_{i=0}^{k-1} f(i).x_{s-d.i} \tag{2.23}$$

Where k is the filter size, d is the dilation factor and s-d.i captures the direction of the past. Thus, dilation introduces a constant time step between two adjacent filters.

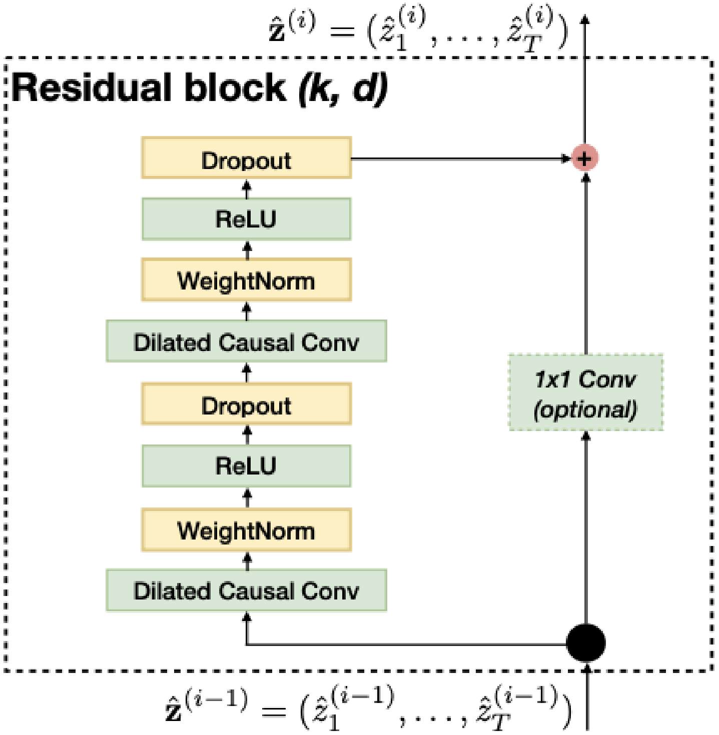

$\hat{\mathbf{z}}^{(i)} = (\hat{z}_1^{(i)}, \ldots, \hat{z}_T^{(i)})$
Residual block (k, d)
Dropout
ReLU
WeightNorm
Dilated Causal Conv
Dropout
ReLU
WeightNorm
Dilated Causal Conv
1x1 Conv (optional)
$\hat{\mathbf{z}}^{(i-1)} = (\hat{z}_1^{(i-1)}, \ldots, \hat{z}_T^{(i-1)})$

Figure 2.8: Residual Block [27]

Residual Connections [69] A residual block contains a series of transformations the output of which is added to the input of the block. Let $\mathcal{F}$ be the transformation and x be the input then the output is represented as in Equation 2.24.

$$o = Activation(x + \mathcal{F}(x)) \tag{2.24}$$

This allows the network to learn the modifications to find the mapping instead of the entire transformation, which has proved to benefit the deep learning networks. The TCN's receptive field depends on the depth of the network and filter size and dilation factor, the stabilization of the deeper networks becomes essential. Figure 2.8 shows a TCN residual block, 1x1 convolution is added when residual input and output have different dimensions.The residual block consists of two layers of dilated causal convolution and non linearity to tackle which ReLU activation is used. In addition, spatial dropout is added after the dilated convolutions for regularization. In TCN the input and output can have different widths unlike the ResNet where the input is added to the output of the residual

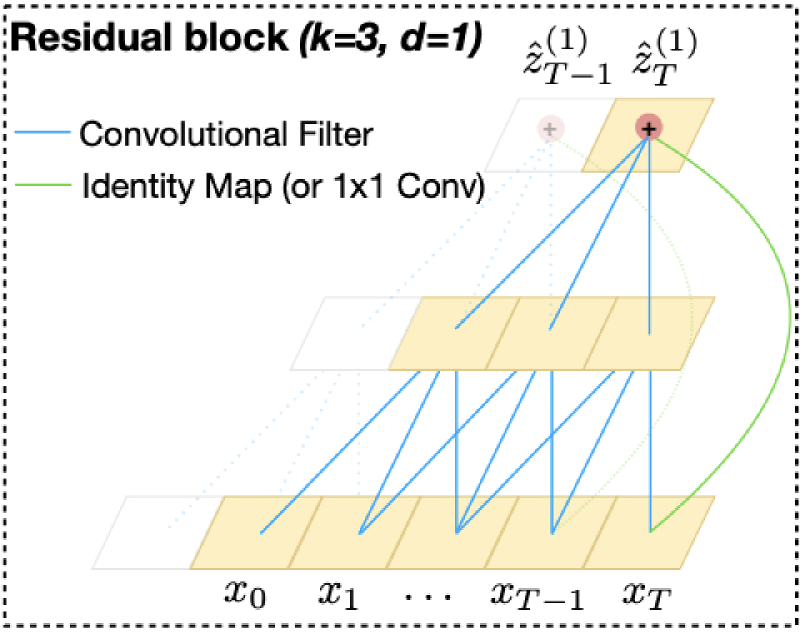


Figure 2.9: An Example of Residual Connection. [27]

function. To ensure different input output width, an additional 1x1 convolution is used to have tensors of same shape for element wise addition. Figure 2.9 shows an example of residual connection in a TCN where the blue lines represent filters in the residual function and the green lines are for identity mappings.

2.4.7 SeriesNet

The conventional time series forecasting models are not competent to adequately extract the essential sequence of data features and often give results with poor accuracy. To tackle this issue a novel forecasting architecture of SeriesNet is introduced in [132].

Typically, SeriesNet consists of two networks, an LSTM network and a dilated causal convolution network. The LSTM network aims to learn integrated features and to reduce dimensionality of multi-conditional data. The dilated convolution handles the loss of resolution or coverage due to the down-sampling operation in image semantic segmentation [156, 44, 45]. The multi-scale contextual information is aggregated systematically using

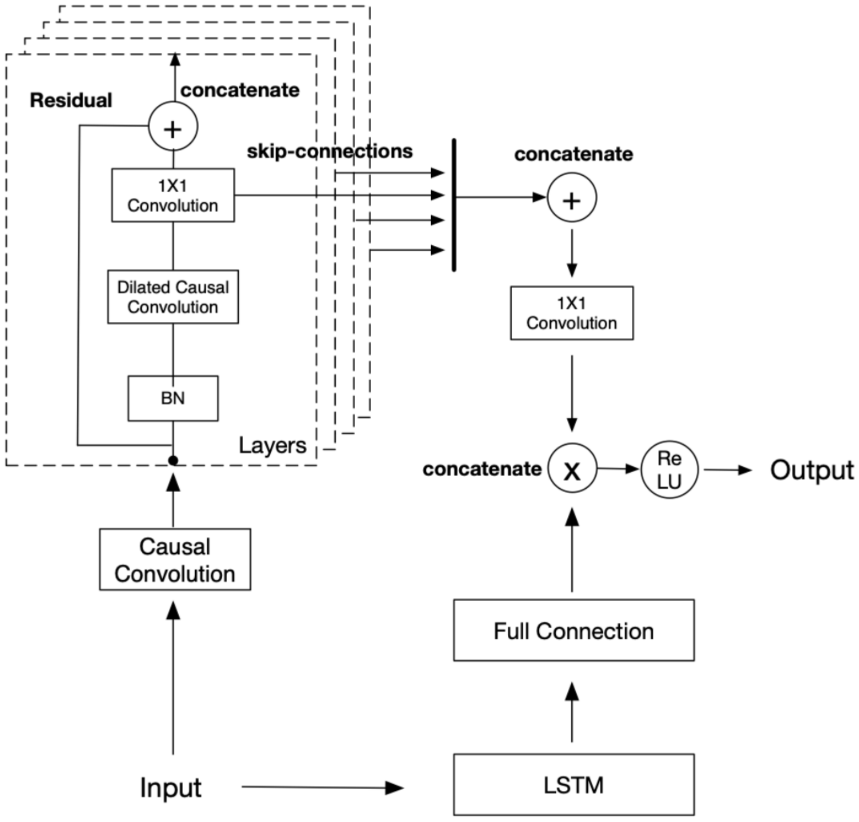


Figure 2.10: The Architecture of SeriesNet [132].

the dilated convolutions which improves the accuracy of image recognition. The causal convolution [113] ensures that the convolution kernel of CNN performs the convolution operations in exact time sequence and that the convolution kernel can only read the current and historical information. This approach helps in giving a higher predictive accuracy as compared to the other models that use fixed time intervals since the combined result obtained from both networks help the models to learn multilevel and multi-range features from time se-ries data. Moreover, this model uses batch normalization as well as residual learning to improve the overall generalization. The architecture of the network is illustrated in Figure 2.10. In this work, the LSTM network of the SeriesNet in [132] is replaced by a GRU network with two layers of GRU and an attention module in between.

2.4.8 Attention Mechanism

The self attention module approaches the different positions of a single sequence and then finds the representation of the encountered sequence. It enables the deep learning model to focus on the important details of the input and reject the unnecessary information. The attention layer is applied to the top of every unit of the sequence and it uses the additive attention in the applied model. The attention function maps a set of key-value pairs and query to an output. The keys, values, queries and outputs are all considered as vectors [142]. A feed forward network is utilized by additive attention to calculate the compatibility function [159]. The equations below show the working:

$$h_{t,t'} = \tanh\left(x_t^T W_t + x_{t'}^T W_x + b_t\right) \tag{2.25}$$

$$e_{t,t'} = \sigma(W_a h_{t,t'} + b_a) \tag{2.26}$$

$$a_t = \mathrm{softmax}(e_t) \tag{2.27}$$

$$l_t = \sum_{t'} a_{t,t'} x_{t'} \tag{2.28}$$

where σ is the element wise sigmoid function and W_x and W_t are the weight matrices corresponding to x_t^T and $x_{t'}^T$. The W_a is the weight matrix corresponding to the non-linear combination of W_x and W_t, b_t and b_a are the bias vectors [142]. Equation (2.28) depicts how the attention value l_t is calculated. To find the value of attention, the probability distribution a_t and compatibility score $e_{t,t'}$ should be found first. The compatibility score is calculated using the hidden representation $h_{t,t'}$ of x_t^T and $x_{t'}^T$. Introducing the attention layer is proven to be very useful since it improves the performance of the deep learning models. The addition of the attention layer improves the performance in problems such as yield prediction of fresh produce, natural language processing, and healthcare interactions [54, 38, 159, 142]. The attention layer is used in the proposed model, it is added after the 1-D convolutional layer after dilated convolution operations and also between the two layers in the GRU network.

2.5 Performance Metrics

It is highly essential to choose the correct performance metric to evaluate the Deep Learning models. The metrics influence how the performance of machine learning algorithms is measured and their comparison. The metrics which are most commonly used are the Mean Squared Error (MSE), Root Mean Square Error (RMSE), and Mean Absolute Error (MAE)

[21, 80, 124] . R^2 [22] is used to measure the degree of correlation between the predicted and actual values.

2.5.1 Mean Squared Error (MSE)

The Mean Squared Error is one of the most commonly used metrics and it works by taking the average of the squared differences between actual and predicted values; taking the square difference highly penalizes large errors. This metric takes the square difference and therefore, it highly penalizes large errors. This metric is preferred mainly because it is differentiable and hence it can be optimized. The only disadvantage of MSE is that it is not robust to outliers [80]. Mathematically, it is defined as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2.29}$$

where, y_i is the actual value whereas, $\hat{y}_i$ is the predicted value and n is the sample size.

2.5.2 Root Mean Squared Error (RMSE)

RMSE is square root of the average squared difference between the actual and predicted values. The square root brings the squared error back to the scale of the target. RMSE is a better option when large errors are not desirable. It is calculated as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{2.30}$$

where, y_i is the actual value whereas, $\hat{y}_i$ is the predicted value and n is the sample size.

2.5.3 Mean Absolute Error (MAE)

The Mean Absolute Error calculates the absolute difference between the actual and predicted values and then calculates the average difference or error. In some cases, MAE is preferred over MSE or RMSE since it is robust towards the outliers. The only disadvantage of MAE is that it is non- differentiable [152]. It is given as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{2.31}$$

where n is the total number of observations, y_i is the true value for observation i, and $\hat{y}_i$ is the predicted value at observation i.

2.5.4 Coefficient of Determination (R^2)

The R-Squared coefficient, or the coefficient of determination, represents the percentage of the variance in the dependent variable that is explained by the independent variable. Mathematically, it is defined as:

$$R^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2} \tag{2.32}$$

where y_i is the true value for observation i, $\hat{y}_i$ is the predicted value at observation i, and $\bar{y}$ is the mean value of the observations. Also, $\sum (y_i - \hat{y}_i)^2$ is the residual sum of squares and $\sum(y_i - \bar{y})^2$ is the total sum of squares. MAE and MSE give absolute values which may not intuitively depict how well a model performs. R^2 is a statistical term which is used to describe the portion of the variance in the dependent variable that can be predicted using the independent variable. It helps in measuring how well observed outcomes are replicated by the model, based on the proportion of total variation of outcomes explained by the model [60, 57]. R^2 is used to measure the goodness of fit of a model for any linear least square regression model which has an intercept, the value of R^2 ideally lies between 0 and 1 [95]. However, sometimes negative values can occur depending on the exact mathematical definition of R^2 being used and the type of model being fitted. Occurrence of negative value means that the mean of the data set fits the dependent variables better than the values provided by the model which signifies a complete lack of fit. This occurs when an unsuitable model is picked to solve a particular regression problem. The best results occur when the predicted values by the model are the same as the actual values which yields an R^2 of 1. A constant value model which derives no information about the independent variables and predicts the mean will always result in an R^2 of 0. When a model performs well the variation in the data is properly captured and the residual sum of squares obtained is low and R^2 value close to 1 is attained. However, if the variation in the data is not properly captured by the model, then the residual sum of squares will be high and a value of R^2 will be closer to zero. The drawbacks of R^2 are: as the number of explanatory variables increase, its value increases. It does not account for collinearity. It does not tell if enough data points are used.

2.5.5 Aggregated Measure (AGM)

The AGM incorporates the information captured by the metrics like RMSE, MAE and R^2 into a single metric to simplify the process of choosing the best performing model [102, 110]. AGM is negatively oriented which means that lower scores demonstrate better performance; it is mathematically defined in (2.33).

$$AGM = \frac{RMSE + MAE}{2} \times (1 - R^2) \tag{2.33}$$

2.6 Related Study

Crop yield forecasting is an essential task for making decisions related to crop management, future market, and crop pricing at both regional and country levels. There are various image based methods in precision agriculture which are adopted for apple yield forecasting like in [24], the apple yield is estimated by obtaining the images of the individual apple trees using unmanned aerial vehicle (UAV) and then a Convolutional Neural Network is trained to detect the count of apples on the orthomosaic built from images taken by UAV. In [43], the images of the Gala trees are captured from an orchard near Bonn, Germany and then the apple yield is forecasted by performing the image analysis and using the tree canopy features as input to the Artificial Neural Networks (ANNs). These methods are very different from the proposed forecasting method where there is no need to access the apple orchards and capture images of the trees. Using the weather and soil parameters, the apple yield can be forecasted for a wider area since there is no need for visiting each individual orchard to obtain images.

Previously, multiple methods have been used for yield forecasting like the method of Partial Least Squares (PLS) and ANN as used in [120]. It is concluded that they are good for short and long term crop price forecasting. The strawberry yield forecast is evaluated using the predictive principal component regression with a single layer neural network and generic random forest (RF) in [104] & [118]. In [40], they use the Bayesian Network approach to predict the corn yield for the 99 counties in Iowa. On the other hand, nondeep Machine Learning (ML) models such as ANNs are used in [90, 89, 56] for forecasting as well as simple DL ones like the Long Short Term Memory Networks (LSTM) in [85] & [72]. In [102], the strawberry yields and prices are predicted using various DL compound models like ConvLSTM, CNN-LSTM, CNN-LSTM-GRU with attention layer along with DL ensemble models. The models are trained using the dynamic weather parameters. The DL models are recommended over the nondeep ML and non-ML models for forecasting. In

[117], counter-propagation artificial neural networks (CP-ANN) and Supervised Kohonen Networks (SKNs) are used to predict the wheat yields after being trained using a found input set of influential soil quality parameters obtained from advanced sensing techniques like soil spectroscopy. In [58], Neural Networks and statistical methods are used for site specific yield prediction while the soil quality parameters are used as input. In this work, the static soil quality parameters as well as the dynamic ones are used to forecast the apple yield. Deep learning imputation techniques in [128] & [110] are used to fill the missing values encountered in the utilized time series. In addition, transfer learning applications are well established within the deep learning literature. In the fresh produce field, the authors of [144] successfully transferred the learning of a forecasting model trained on annual yield data based in Argentina to another model built for annual yield forecasting in Brazil. They discuss the possibility of extending their transfer learning application to cover other crops as well.

2.7 Chapter Summary

This chapter starts with a Fresh Produce Procurement process section covering the challenges which are currently faced in Fresh Produce Supply Chain Management. This is followed by a Time Series Modelling section which starts by discussing the types of time series, multivariate or univariate, then the components of any time series, like trend, seasonal variations, cyclic variations and random variations. The Data Preprocessing section includes methods and work done in Data Imputation, Feature Selection, conversion of time series to supervised learning form and finally dimensionality reduction techniques. Then the state-of-the-art deep learning models which are used in literature for forecasting purposes are descried. The Evaluation metrics section lists metrics commonly used in literature along with the pros and cons of each. Finally, the Related Work section covers the work done in the field of forecasting.

This research work aims to achieve the following which has not been tackled in any of the discussed works of literature:

(a) Advanced deep learning models which have mostly been used for natural language processing or image video analysis are modified to work with the time series problem being solved in this thesis.

(b) Despite using either the dynamic parameters or the static soil quality parameters, none of the reviewed approaches analyze or gauge the effect of adding the static parameters to the dynamic ones on forecasting performance when both parameter types are concurrently used.

(c) The commonly used methods for yield prediction are survey based which is a stereotypical method of sending enumerators in field visits to important production areas. This is a cumbersome process compared to just using the station-based data of soil or weather parameters to forecast the yield.

(d) The method of transfer learning is used to transfer and share the learning amongst similar crops. Despite the deployment of Transfer Learning in many fields, not much is found on its application in the domain of crop yield/price prediction.

Chapter 3

Proposed Solution

Various aspects need to be tackled to solve the problem of Fresh Produce forecasting. The proposed solution to deal with this problem comprises two stages:

- Data collection and preprocessing: sources for the right input data needed for the modelling purpose are found. The missing values should be handled correctly, this is critical because right imputation methods must be selected else complete elimination of missing values can add bias to the forecasting model. The correct lag time for each considered FP is decided.
- Forecasting models and model tuning: Various forecasting models are developed, these are compound DL models. The optimal parameters are chosen using hyper-parameter tuning.

Further details of the two stages are discussed in this section.

3.1 Datasets and Preprocessing

Various Datasets are used in this research work to forecast the Fresh Produce yield and price. These datasets include soil, weather, FP price and FP yield data.

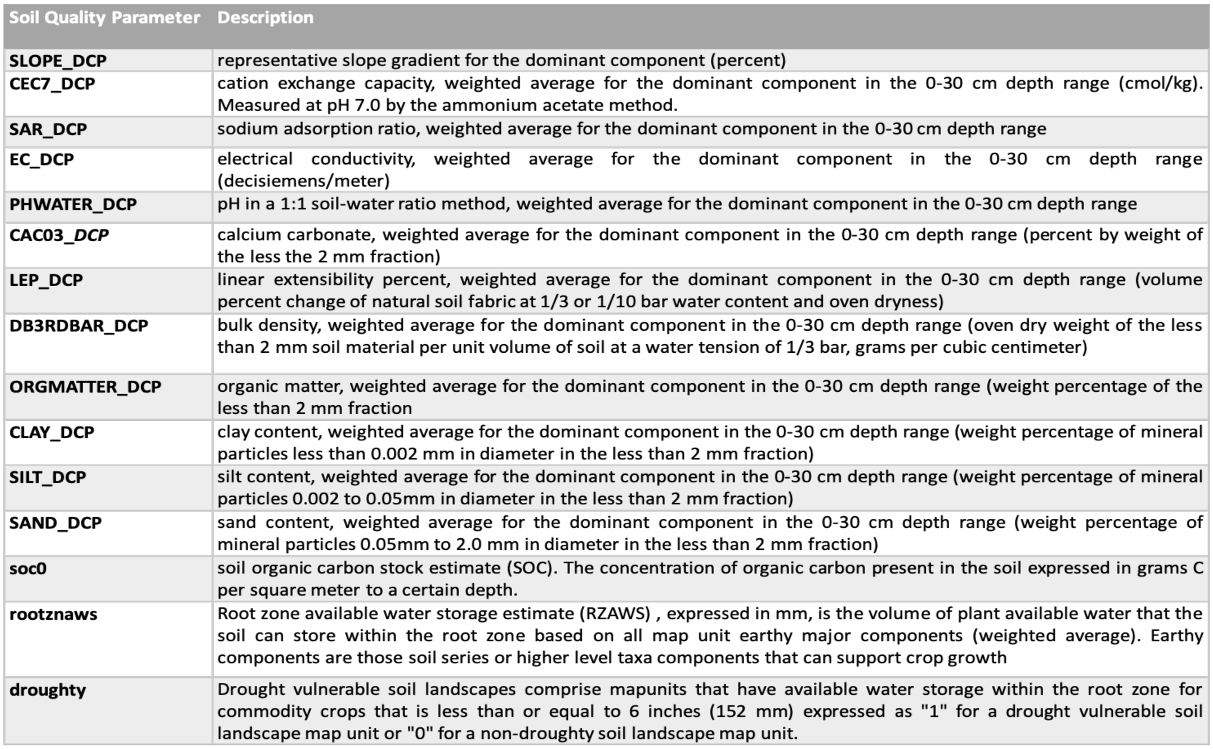
3.1.1 California Weather Data

The California weather data is obtained from California Irrigation Management Information System (CIMIS) [4]. TThis weather data consists of the thirteen weather parameters: the daily value of evapotranspiration rate (ETo), precipitation, solar radiation, dew point, air temperature, vapor pressure, relative humidity, wind speed, and soil temp parameters. This data is available at the county level and can be obtained for every county in California.

3.1.2 California County Level Soil Quality Static Data

The static soil parameters vary per county but are constant within each county. The static soil parameters are collected from the United States Department of Agriculture: Web Soil Survey (USDA-WSS) [9]. The USDA site has a detailed record for each county and the soil quality parameters are available at the map unit level for every county.

Table 3.1: Details of Static Soil Parameters



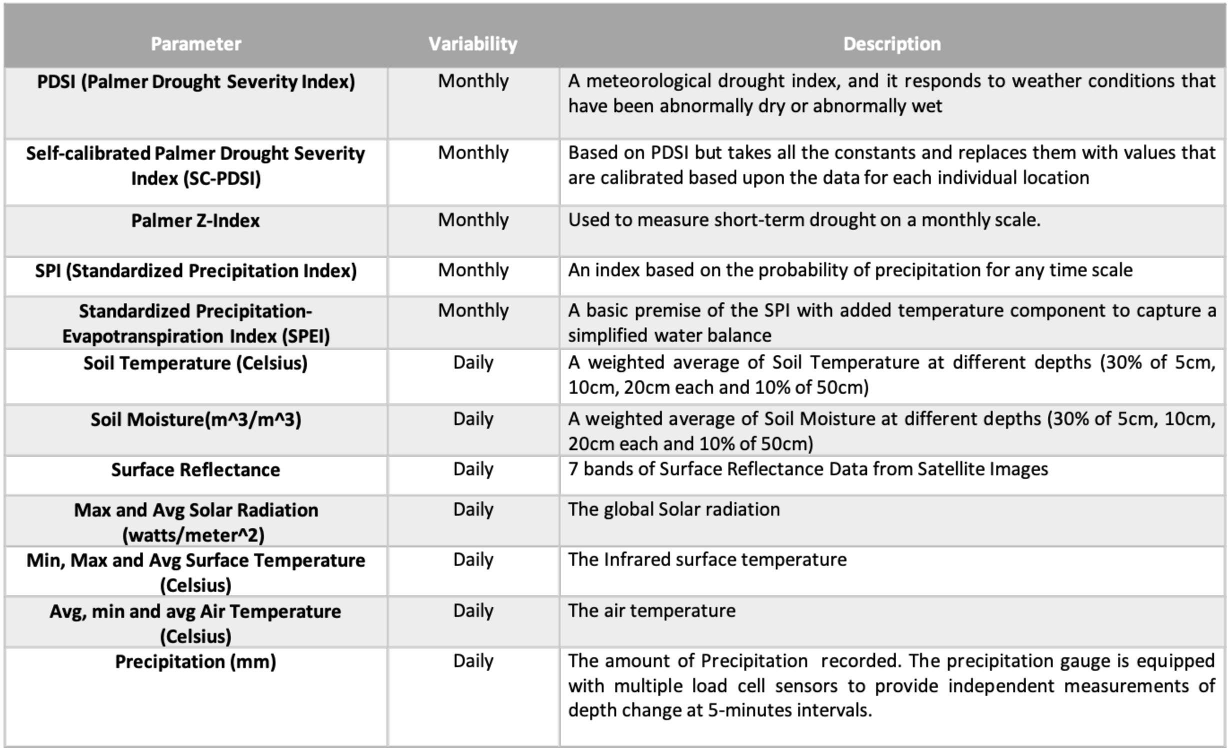
Soil Quality Parameter	Description
SLOPE_DCP	representative slope gradient for the dominant component (percent)
CEC7_DCP	cation exchange capacity, weighted average for the dominant component in the 0-30 cm depth range (cmol/kg). Measured at pH 7.0 by the ammonium acetate method.
SAR_DCP	sodium adsorption ratio, weighted average for the dominant component in the 0-30 cm depth range
EC_DCP	electrical conductivity, weighted average for the dominant component in the 0-30 cm depth range (decisiemens/meter)
PHWATER_DCP	pH in a 1:1 soil-water ratio method, weighted average for the dominant component in the 0-30 cm depth range
CAC03_*DCP*	calcium carbonate, weighted average for the dominant component in the 0-30 cm depth range (percent by weight of the less the 2 mm fraction)
LEP_DCP	linear extensibility percent, weighted average for the dominant component in the 0-30 cm depth range (volume percent change of natural soil fabric at 1/3 or 1/10 bar water content and oven dryness)
DB3RDBAR_DCP	bulk density, weighted average for the dominant component in the 0-30 cm depth range (oven dry weight of the less than 2 mm soil material per unit volume of soil at a water tension of 1/3 bar, grams per cubic centimeter)
ORGMATTER_DCP	organic matter, weighted average for the dominant component in the 0-30 cm depth range (weight percentage of the less than 2 mm fraction
CLAY_DCP	clay content, weighted average for the dominant component in the 0-30 cm depth range (weight percentage of mineral particles less than 0.002 mm in diameter in the less than 2 mm fraction)
SILT_DCP	silt content, weighted average for the dominant component in the 0-30 cm depth range (weight percentage of mineral particles 0.002 to 0.05mm in diameter in the less than 2 mm fraction)
SAND_DCP	sand content, weighted average for the dominant component in the 0-30 cm depth range (weight percentage of mineral particles 0.05mm to 2.0 mm in diameter in the less than 2 mm fraction)
soc0	soil organic carbon stock estimate (SOC). The concentration of organic carbon present in the soil expressed in grams C per square meter to a certain depth.
rootznaws	Root zone available water storage estimate (RZAWS) , expressed in mm, is the volume of plant available water that the soil can store within the root zone based on all map unit earthy major components (weighted average). Earthy components are those soil series or higher level taxa components that can support crop growth
droughty	Drought vulnerable soil landscapes comprise mapunits that have available water storage within the root zone for commodity crops that is less than or equal to 6 inches (152 mm) expressed as "1" for a drought vulnerable soil landscape map unit or "0" for a non-droughty soil landscape map unit.

The values for the map unit comprising of the farmlands are averaged to get the soil quality values per county. 15 soil quality parameters are considered including: root zone available water (rootznaws), drought vulnerability (droughty), Soil Organic Carbon Stock (SOC), sand content, silt content, clay content, organic matter, bulk density, calcium carbonate, water PH, linear extensibility percent, slope, cation exchange capacity, sodium adsorption ratio and electrical conductivity. The details of these parameters are given in Table 3.1.

3.1.3 California County Level Soil Dynamic Data

The dynamic parameters varying on daily and monthly basis within each county are collected from the National Oceanic and Atmospheric Administration website [17] and The National Drought Mitigation Center; Drought Risk Atlas [8] respectively.

Table 3.2: Details of Dynamic Soil Parameters



Parameter	Variability	Description
PDSI (Palmer Drought Severity Index)	Monthly	A meteorological drought index, and it responds to weather conditions that have been abnormally dry or abnormally wet
Self-calibrated Palmer Drought Severity Index (SC-PDSI)	Monthly	Based on PDSI but takes all the constants and replaces them with values that are calibrated based upon the data for each individual location
Palmer Z-Index	Monthly	Used to measure short-term drought on a monthly scale.
SPI (Standardized Precipitation Index)	Monthly	An index based on the probability of precipitation for any time scale
Standardized Precipitation-Evapotranspiration Index (SPEI)	Monthly	A basic premise of the SPI with added temperature component to capture a simplified water balance
Soil Temperature (Celsius)	Daily	A weighted average of Soil Temperature at different depths (30% of 5cm, 10cm, 20cm each and 10% of 50cm)
Soil Moisture(m^3/m^3)	Daily	A weighted average of Soil Moisture at different depths (30% of 5cm, 10cm, 20cm each and 10% of 50cm)
Surface Reflectance	Daily	7 bands of Surface Reflectance Data from Satellite Images
Max and Avg Solar Radiation (watts/meter^2)	Daily	The global Solar radiation
Min, Max and Avg Surface Temperature (Celsius)	Daily	The Infrared surface temperature
Avg, min and avg Air Temperature (Celsius)	Daily	The air temperature
Precipitation (mm)	Daily	The amount of Precipitation recorded. The precipitation gauge is equipped with multiple load cell sensors to provide independent measurements of depth change at 5-minutes intervals.

There are various parameters related to the soil which affect the yields and prices of fresh produce namely: soil moisture, soil temperature, solar radiation, surface temperature, PDSI (Palmer Drought Severity Index), Palmer Z-Index [16, 18],... etc. The details of these parameters are give in Table 3.2. These monthly parameters remain same for all the counties within a Crop Reporting District.

3.1.4 Daily Berry Yield and Price Data

The berries used for this research are the strawberries and raspberries. The berry yield and farm-gate price data is extracted from the California Strawberry Commission website [15]. Both the daily and weekly values for price and yield are available on this website. There are a few missing records with missing yield or price data that are entirely dismissed without interpolation in few experiments for the sake of comparison, otherwise, for all other experiments, the daily values are imputed using advanced deep learning imputation techniques. Figure 3.1, Figure 3.2 and Figure 3.3 show the strawberry yields, raspberry yields and strawberry prices respectively after imputing their missing values.

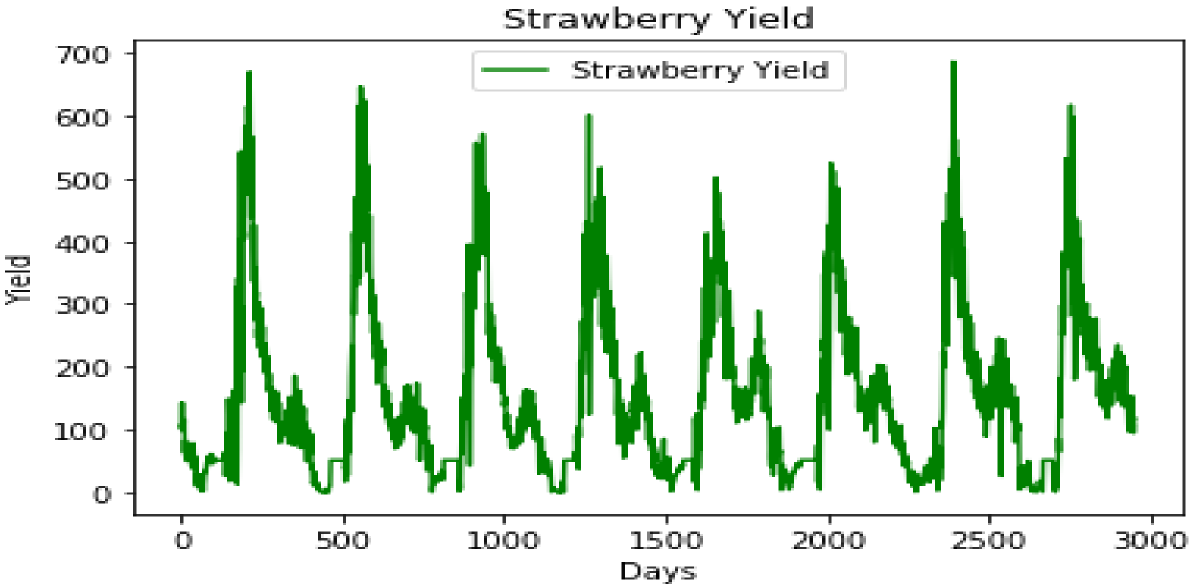


Figure 3.1: Imputed Strawberry Yields

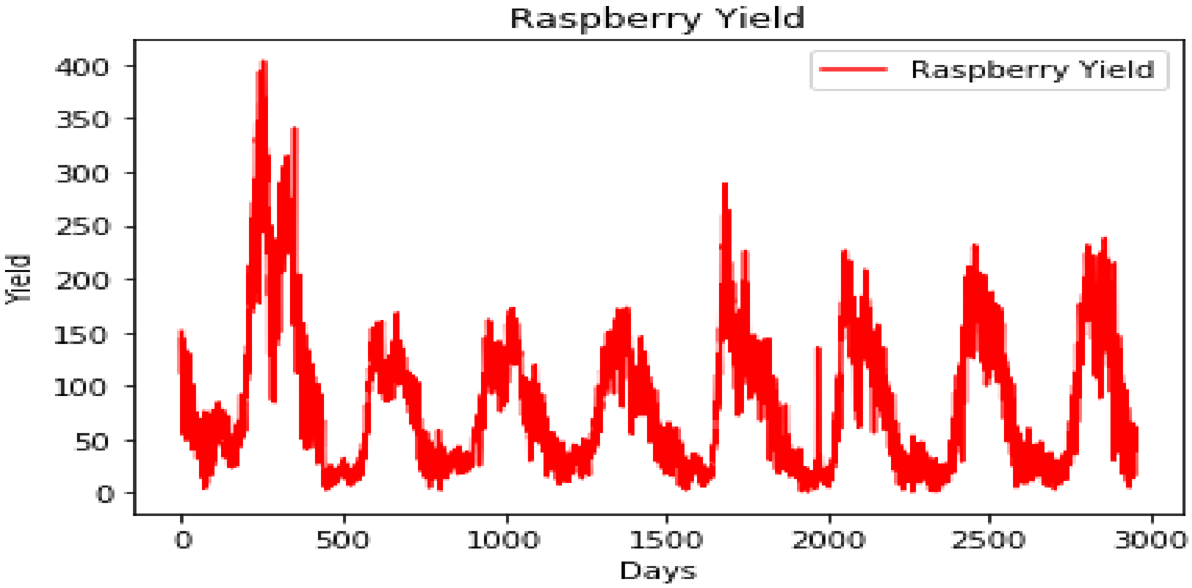


Figure 3.2: Imputed Raspberry Yields

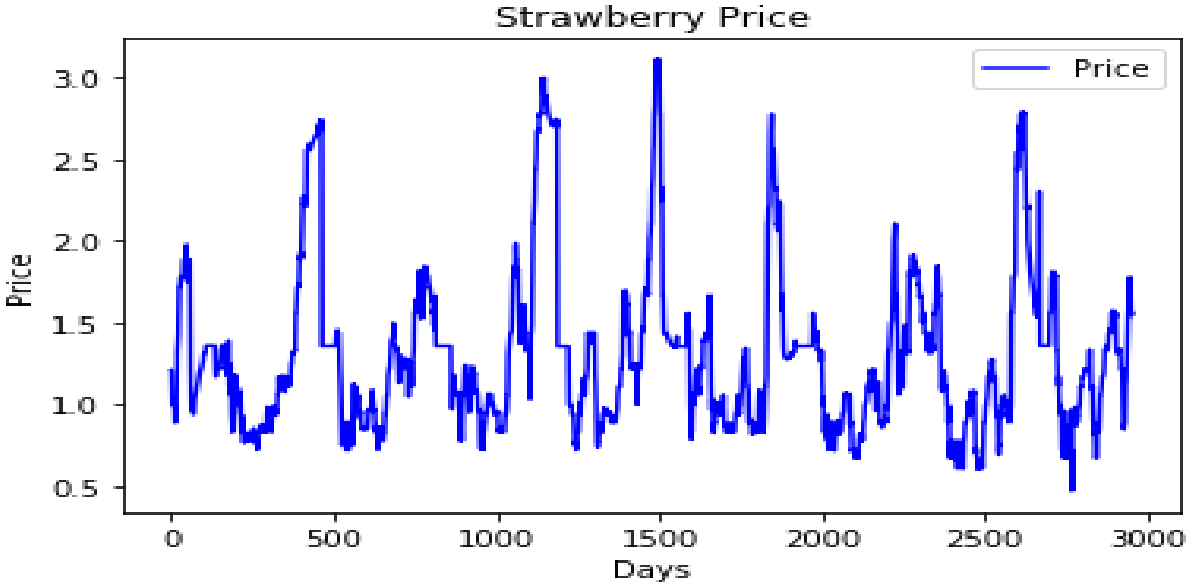


Figure 3.3: Imputed Strawberry Prices

3.1.5 Yearly California County Level Apple Yield Data

The yearly apple yield data is collected from the United States Department of Agriculture, National Agricultural Statistics Survey (USDA-NASS) [13]. This data is collected for 15 counties across 6 Crop Reporting Districts (CRD) in California. The six CRDs are North Coast, Central Coast, Sacramento Valley, San Joaquin Valley, Sierra Nevada and Southern California. Figure 3.4 shows highlights the selected 15 counties across 6 CRDs.

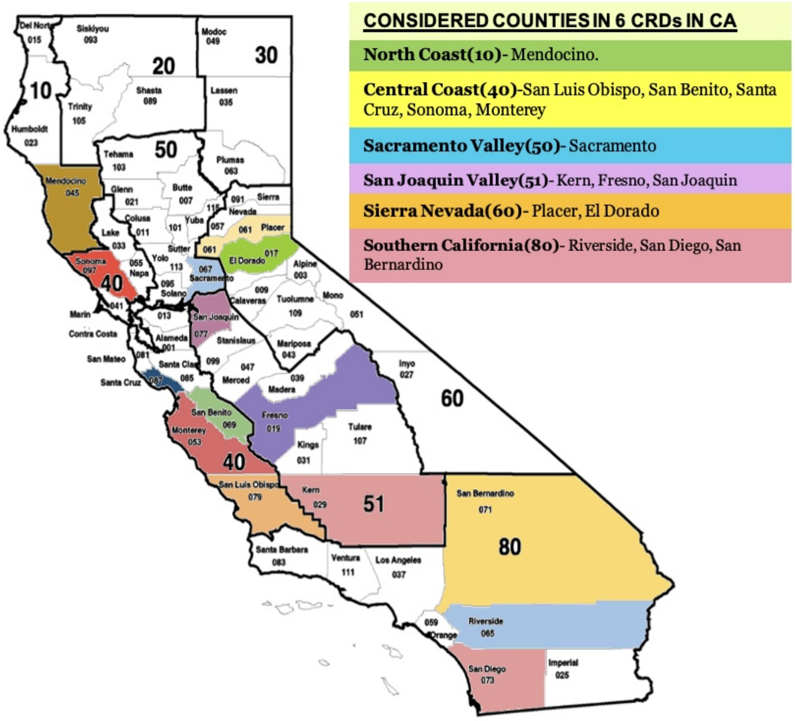


Figure 3.4: Selected 15 Counties in 6 Crop Reporting Districts

3.1.6 Bidirectional Imputation

The method of Bi-directional imputation is used to impute the missing values in the soil data. This method is a primitive method of filling the missing values in a time series dataset inspired by the Last Observation Carried Forward (LOCF) and Next Observation

Carried Backwards (NOCB) methods [134]. This method first considers the frequency of the data values , whether yearly, monthly, ... etc. Whenever a missing value is encountered, the value occurring in a similar month in the preceding year is used, or carried forward, to impute the encountered missing value. If that last value is missing, then the value of the following year at the same point of time is used, or carried backward, to impute the encountered missing value in the currently considered year.

3.1.7 Data Augmentation

Since the yield data is annual, the number of resulting records is low and hence the need for data augmentation to increase the number of records used for training the DL model which should enhance the forecasting results. From the USDA site, the apple yield data of 37 years, from 1982 to 2018, is extracted for 15 counties, out of which the first 33 years data is used for training and the last 4 years data is used for testing. This splitting takes place before any preprocessing of the data and all the experiments performed with or without augmentation are tested using this non-augmented test dataset comprised of the annual yield values of 15 counties for the last four years from 2015 to 2018. Hence, the total number of available training samples is low; 495 which is 33 x 15. The data augmentation is performed by finding all possible combinations of two counties un-ordered tuples within the same CRD and averaging each pair input and output values to create a new data point [59]. This increases the sample size up to 19008 samples. The augmentation method, followed to generate new samples, is sound since the PDSI value remains the same within a CRD and the other parameters are simply averaged values for the county. Hence, those newly generated points can be used to represent the missing farmland data points [85]; averaging two data points resembles data interpolation for a hypothetical farmland or county in the CRD whose data is missing or unavailable.

3.1.8 Feature Selection and Dimensionality Reduction

Choosing the most influential parameters is a major challenge due to the high correlation amongst those parameters. Hence, the Random Forest feature selection method in Python with scikit-learn [119] is used for the purpose of feature selection. A lag of the past 20 weeks, i.e., 140 days of soil parameters values is found to affect the yields forecasting and prices values 5 weeks ahead. After normalizing the data, the Principal Component Analysis (PCA) [19] is applied and the first n components with the maximum proportion of variance (around 95%) are chosen to train and test the forecasting model along with the corresponding yields or prices output.

3.2 Deep Learning Models

This section describes the details of the deep learning models which are used for various forecasting experiments conducted in this thesis. The deep learning models are chosen for forecasting since, they outperform the conventional machine learning models. Moreover, the deep learning models help in capturing the complex relationship between the input soil and weather parameters and the output yield and price values. The statistical and machine learning models are unable to capture such complex relations. Another reason of using deep learning models lies in their ability to handle large datasets. The input data used for forecasting is huge and contains lots of input parameters. The deep learning models effectively process huge input data to establish complex relationship between the input and output. In this work, mainly the compound deep learning models are used.

3.2.1 Compound Deep Learning Models

This section describes the compound deep learning models used for the experiments performed in the thesis. The compound deep learning models are a combination of two or more deep learning models. The compound deep learning models experimentally perform better than using individual component deep learning models [111]. Using combinations of CNNs and RNNs help in capturing the complex temporal and spatial information in the data. This section discusses the structure and design of the following compound deep learning models used for experiments:

- SeriesNet with GRU and Attention: This compound model consists of SeriesNet comprised of dilated causal convolutional network, GRU which is modified version of RNNs and Attention layer.
- Att-CNN-LSTM: This model contains combinations of convolutional and LSTM layers with Attention layer.
- TCN on Time Series Encoded as Images: This model used the deep learning model TCN on the encoded images of the times series using Gramian Angular Field.

3.2.1.1 SeriesNet with GRU and Attention

The conventional architecture of SeriesNet consists of mainly two networks. One network is of dilated causal convolutional network and the other can be one from a variety of networks

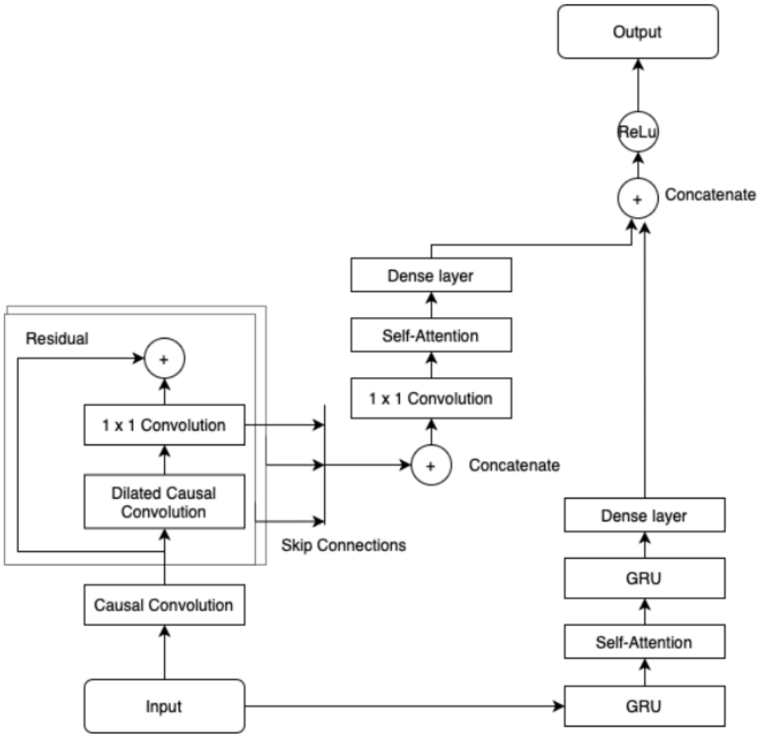


Figure 3.5: SeriesNet with GRU and Attention

mostly similar to the RNNs, like LSTMs, GRUs, ... etc. In the proposed model, the distinguishing factor is the added attention layer which highly enhances the forecasting accuracy. The first network of the dilated causal convolutional network contains seven layers of dilated convolutional operations and a 1D convolution layer. 32 filters are used in the dilated convolutional operation and the dilation rate is increased by 2 times for each layer. The output of the 1D convolutional layer is passed through the attention layer which is then flattened and fed to a single neuron dense layer. In the second network, GRU is used together with the dilated causal convolution. The introduced GRU is a two-layer network where in between an attention module is added. The output at the end is flattened and fed to a single neuron dense layer. The outputs from both networks are concatenated and passed through ReLU activation function. Adam [92] is used as an optimizer and the Mean Squared Error loss function is used. The architecture of the model can be summarized in Fig 3.5. Instead of using the LSTM as proposed in the SeriesNet model in [132], the GRU network is used. The number of layers in the network can be altered as per the requirement.

3.2.1.2 Att-CNN-LSTM

This compound model consists of convolutional layers, LSTM layers and attention layer, starting with four layers of 1-dimensional Convolutional layers with 100 filters, kernel size of 3, padding causal, 1 stride and activation function Relu. This is followed by three LSTM layers, each containing 100 units and Relu activation function. This is followed by a self-attention layer having sigmoid activation function. Then there are four dense layers 128, 64, 32, 16 and 1 neurons in the networks. The loss function used is MSE and Adam optimizer. Figure 3.6 represents the architecture of the network.

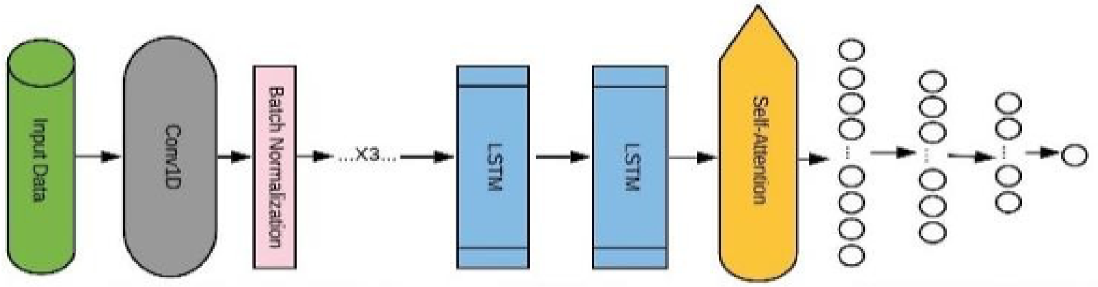


Figure 3.6: Architecture of CNN-LSTM with Attention

3.2.1.3 TCN on Time Series Encoded as Images

In this approach, the input to the Temporal Convolutional Network is the time series encoded into images using the Gramian Angular Fields as discussed in Section 2.3.6 as inputs. The encoding using Gramian Angular Field preserves the temporal dependency through the r coordinate and the whole encoding is bijective which also proves to be helpful [6]. The time increases as the position is moved from top-left to bottom-right and thus the time dimension is encoded into the geometry of the matrix. This is how the Gram Matrix helps in preserving the temporal dependency. Figure 3.7 represents the process of encoding time series into a Gramian image. Figure 3.8 represents the images obtained when a row of soil parameters is encoded into images using GAF, here summation image is obtained from GASF and difference image is obtained from GADF.

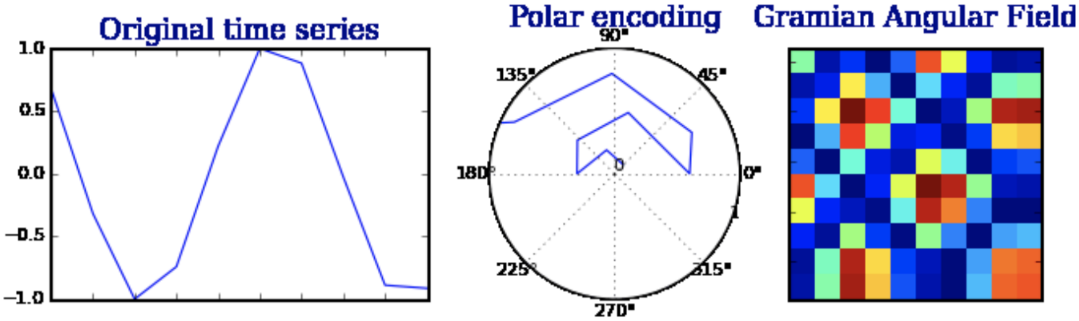


Figure 3.7: Steps for Conversion Encoding Time Series into Image[6].

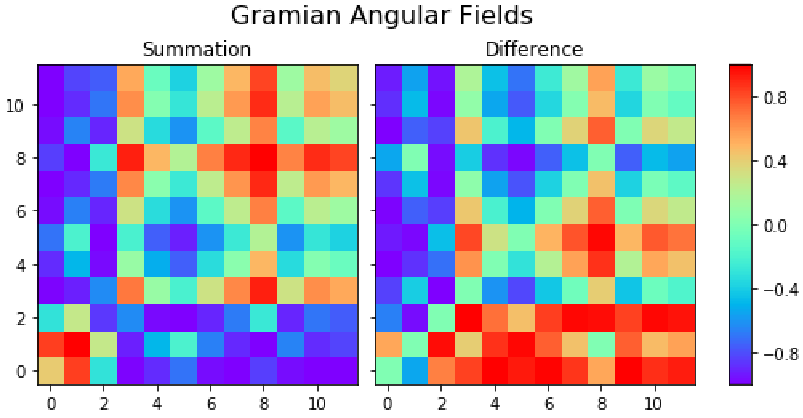


Figure 3.8: Encoded Soil Parameters into Image

TCNs can be preferred over the GRUs and LSTMs since, they have longer memory as compared to RNNs of same length [70, 96]. The convolutional layers promote parallelism, they have flexible receptive size and have stable gradients. The model used in this work has 6 residual blocks 64 filters with a kernel size of 2, with the dilations [1, 2, 4, 8, 16, 32], relu activation and causal padding are used. Figure 3.9 represents a stack of dilated causal convolutional layers.

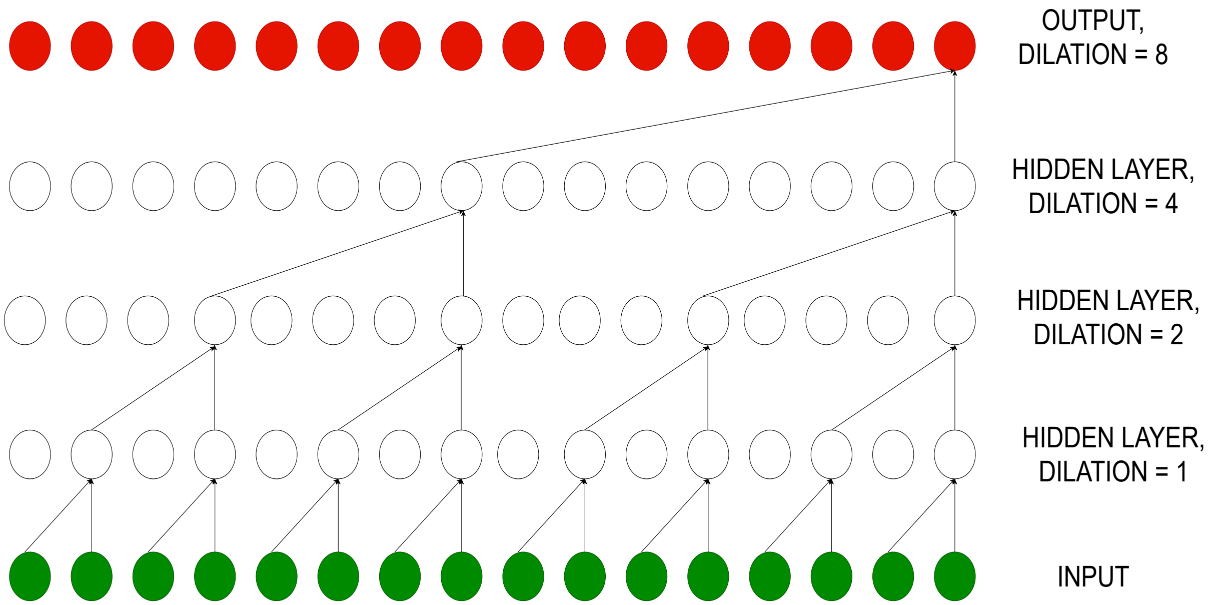


Figure 3.9: Visualization of a Stack of Dilated Causal Convolutional Layers with Filter Size 2 and Dilation [1,2,4,8].

3.2.2 Ensemble Learning

In the field of machine learning, ensemble techniques utilize a combination of multiple learning algorithms to obtain better forecasting performance than could be obtained from any of the individual learning algorithms alone [122, 160]. A machine learning ensemble consists of only a concrete finite set of alternative models, but typically allows for much more flexible structure to exist among those alternatives. Ensemble learning is a method using which multiple models are strategically generated and the results from them are combined to solve a computational problem [115]. Ensemble learning is primarily used to improve the performance of a model, such as classification, prediction, function approximation models, or reduce the likelihood of an unwanted selection of a poor one. Moreover, ensemble learning can be applied to assigning confidence to the decision made by the model, selecting optimal features, data fusion, incremental learning, non-stationary learning, and error-correcting. The two ensemble techniques considered are:

(a) **Stacking Ensemble (SE):** Stacking or stacked generalization involves training a learning algorithm to combine the predictions of several other learning algorithms.

The first step is to train all of the other algorithms using the available data, then a combiner algorithm is trained to make a final prediction using all the predictions of the other algorithms as additional inputs. In practice, a logistic regression model is often used as the combiner for classification and linear regression is used for regression. Stacking typically yields performance better than any single one of the trained models [109]. It has been successfully used on both supervised learning tasks [34].

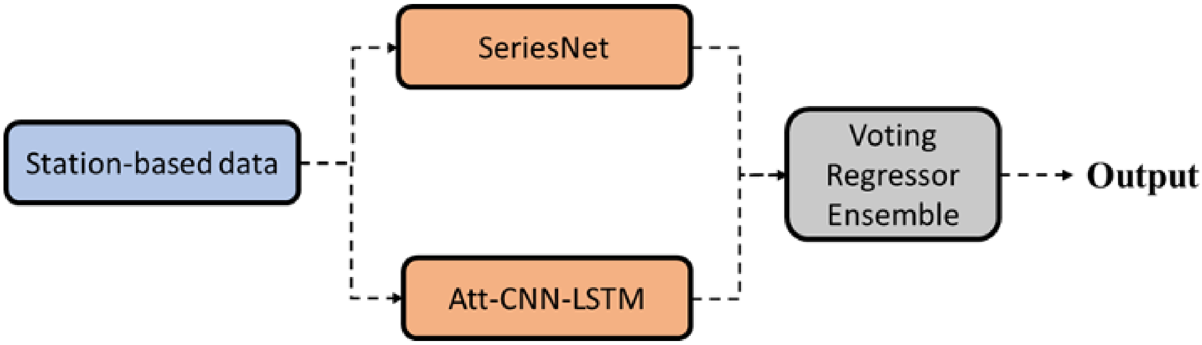


Figure 3.10: Voting Regressor Ensemble for SeriesNet and Att-CNN-LSTM

(b) **Voting Regressor Ensemble (VR):** This ensemble technique is a simple but very effective one. For classification problems, it works by selecting the majority vote after every individual algorithm makes a forecast (hard voting) or averages the forecast probabilities of all the algorithms and picks the class with the highest average probability (soft voting). In regression problems, it works similar to soft voting by averaging the forecasts of the individual algorithms to come up with a final forecast [138, 112]. Since, the problems tackled in this thesis is a regression problems, the voting regressor ensemble of Series-Net and Att-CNN-LSTM takes the output forecast from each model and then ensembles it as shown in Figure 3.10. This helps in adding the learning of both the models in one final forecast value and thus helps in better forecast.

3.3 Hyperparameter Tuning and deciding FP Lag

Tuning the hyperparameters for a deep learning model is an essential part of modelling. The first step for tuning is to set the right learning rate, which is the size of the step which the algorithm follows as it discovers the global minimum [154, 157]. Selecting a very large value for learning rate might oscillate the model from one local minimum to the other and if the learning rate is too low then the convergence to the global minimum will take a lot of time. Hence, it is necessary to find an optimal learning rate to form a generalizable model which fits well to the data. Overfitting is a situation when the data has high variance and the model tends to memorize the data pattern, the model fits well over the train data but doesn't perform well for unseen test data because it is not generalizable. When the dataset is too small with high variance, the overfitting causes the model to be too complex with very low bias. The problem of overfitting can be dealt with using regularization. Conversely, in the case of underfitting, the opposite of overfitting, the model is unable to fit the data well, it neither fits the training nor the test set, i.e., not generalizable. Thus, the ideal case is to find hyper-parameters with which the model neither underfits nor overfits the data and be generalizable. To find these best parameters the following steps are taken:

(a) While training the model, the callbacks has been used in which model checkpoint is added. This checkpoint tracks the weights corresponding to the best validation loss, i.e., least error. The patience, which is the number of tracking epochs with increasing loss, is set to 30 to avoid overfitting which is commonly encountered due to high number of epochs.

(b) To determine the optimal learning rate, layers in model, optimizer choice, the loss function and regularization parameter the method of hit and trial is used, which is having multiple manual iteration till the best set is discovered.

The lag for the strawberry as well as raspberry is chosen to be 20 weeks which is equal to 140 days. The lag value depicts how many past values are auto correlated with the present value. 140 days lag depicts that the present value to be forecasted, e.g., price or yield, depends on the past 140 values of the weather and soil parameters.

3.4 Chapter Summary

The deployed datasets, preprocessing, proposed models, and tuning are the four main topics covered in this chapter.

(a) The datasets used in this work are described: the California weather data along with the dynamic and static soil data. The yield and price data for strawberry and raspberry along with the apple yearly yield data.

(b) The preprocessing methods are explained, like the feature selection, data augmentation and dimensionality reduction.

(c) The details of the proposed models are provided: the proposed compound deep Learning models which are as solution for the problem.

(d) A process is provided for hyper-parameter tuning and deciding the correct FP lag.

Chapter 4

Experiments and Analysis

The Chapter 3 covers the solution proposed to forecast the yield and price of fresh produce, with the details of the input data, output data and the models used. This chapter describes details on the various experiments conducted in this thesis using the models and the datasets described in the previous chapter. The experiments are divided into the following four sections:

1. **Soil and Weather Parameters' Effect on Yield Forecasting**: The experiments under this section are performed to find the best set of input parameters for accurate yield forecasting.

2. **Soil and Weather Parameters' Effect on Price Forecasting**: The experiments under this section are performed to find the best set of input parameters for precise price forecasting.

3. **Finding the best performing Deep Learning Model for Price and Yield Forecasting**: Two experiments are performed in this section to find the best performing compound DL model for yield and price forecasting with soil moisture and temperature as the input parameters.

4. **Soil Parameters' Effect on Yield Forecasting**: The experiments under this section are performed to find the effect of using different types of soil parameters like the dynamic and static soil parameters on yield forecasting across varied counties in California and to find the best performing models.

5. **Transfer Learning for Yield Forecasting**: The experiments under this section are performed to analyse how effective is transfer learning amongst the crops of similar

nature. The transfer learning reduces the computational complexity incurred while retraining the models.

4.1 Soil and Weather Parameters' Effect on Yield Forecasting

In this section experiments are conducted to forecast the strawberry yield using the soil and weather parameters. Three experiments are conducted in which soil parameters, soil and weather parameters together and just soil moisture and temperature, are used to forecast the strawberry yield. The proposed deep learning model for forecasting strawberry yields and prices is tested in Santa Barbara county, California. The main objective behind these experiments is to determine the best set of parameters and best deep learning model required to forecast strawberry yield. The experiments discussed in Section 4.1.1 and 4.1.2 are performed to determine which set of parameters give better yield forecasting results..

For the datasets, the dynamic soil data as discussed in Section 3.1.3 is used while the weather data consisting of the thirteen weather parameters which are the daily value of evapotranspiration rate (ETo), precipitation, solar radiation, dew point, air temperature, vapor pressure, relative humidity, wind speed, and soil temp parameters is used as discussed in Section 3.1.1, and the yield data in Section 3.1.4 is used as labels.In this experiment set, only one county, which is Santa Barbara in California, is considered therefore the static soil parameters are omitted since they are constant per county. A lag of the past 20 weeks, i.e., 140 days, for all considered parameters is found to affect the yields forecasting values 5 weeks ahead [102]. For the preprocessing needed, first the data is normalized using the Principal Component Analysis [19]and then the parameters with maximum proportion of variance are chosen to train and test the forecasting model along with the corresponding yields or prices output. For a few experiments, particularly the one using the TCN model, the data is further transformed using Gramian Angular field. The total number of samples is 2,812 from year 2011 to 2019 out of which 80% are used for training and 20% for testing. The description of experiments is in the following subsections.

4.1.1 Yield Forecasting Using Soil Parameters

In this experiment the soil parameters alone are used to forecast the yield and the results are reported based on all deployed performance measures for 5 weeks ahead. The dynamic soil data described in Chapter 3 is used in this experiment. The soil data is first preprocessed, the preprocessing steps include first creating a window of 20 weeks of data to predict the yield 35 days ahead and hence the prediction is for 5 weeks ahead. Thus, each single parameter had value for 140 days (20x7days), which naturally resulted into a lot of input parameters and thus high dimensionality. To select the best parameters and reduce the dimensionality of the data, the Principal Component Analysis is applied and the parameters accounting to 90% variance in the data are chosen.

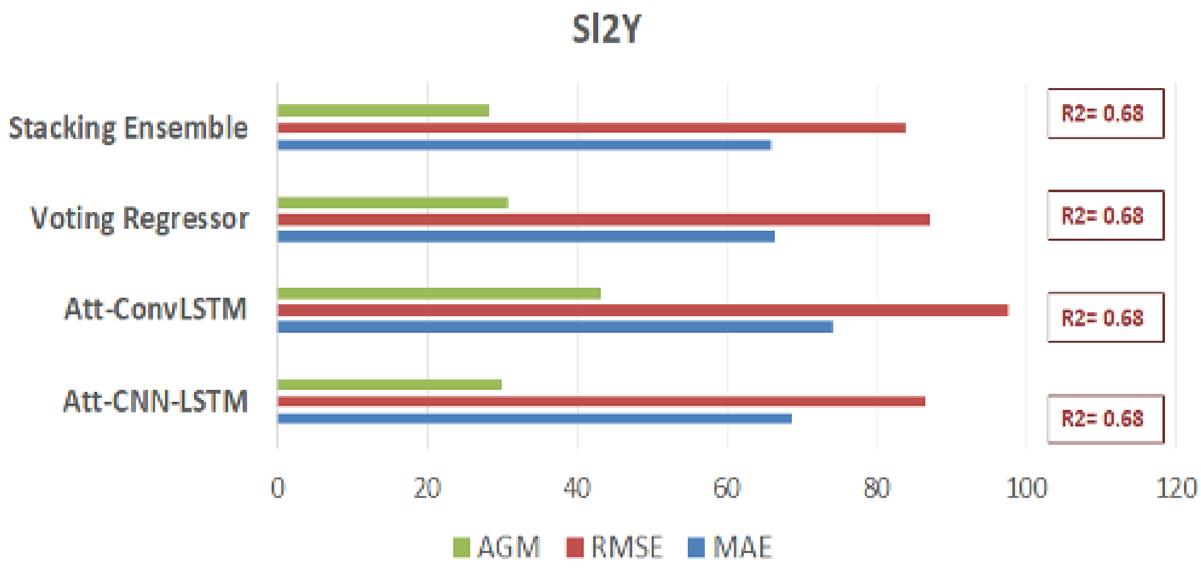


Figure 4.1: Results of Yield Forecasting Using Soil Parameters

The preprocessed data is then fed into two models Att-CNN-LSTM and Att-Conv-LSTM. Finally, the stacking ensemble and voting regressor are used to ensemble the results from these two models. Figure 4.1 presents the results of these models, with the stacking ensemble performing the best with the least forecasting errors. Figure 4.2 shows the final forecasted values for days between 17-12-2017 and 03-07-2019.

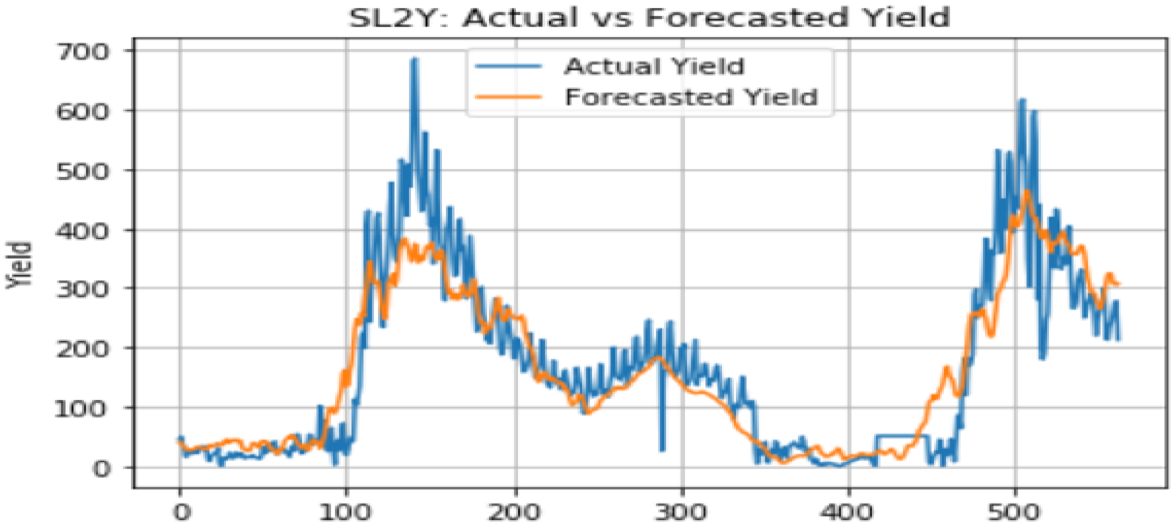


Figure 4.2: Actual vs Forecasted Yields Using Soil Parameters

4.1.2 Yield Forecasting Using Soil and Weather Parameters

In this experiment, the weather parameters are used along with the soil parameters to forecast the yield. The thirteen weather parameters described in Chapter 3 along with the dynamic soil parameters are used to train the two compound models Att-CNN-LSTM and Att-ConvLSTM. The test set is then used for forecasting and the obtained results are shown in Figure 4.3. It is observed that overall, the soil and weather parameters together perform better than using just the soil parameters. The outputs from both compound models are ensembled using the voting regressor and the stacking ensemble; results of both ensembles are depicted in Figure 4.3. It is observed that the stacking ensemble performs better than the voting regressor. Figure 4.4 shows the final forecasted values for days between 17-12-2017 and 03-07-2019.

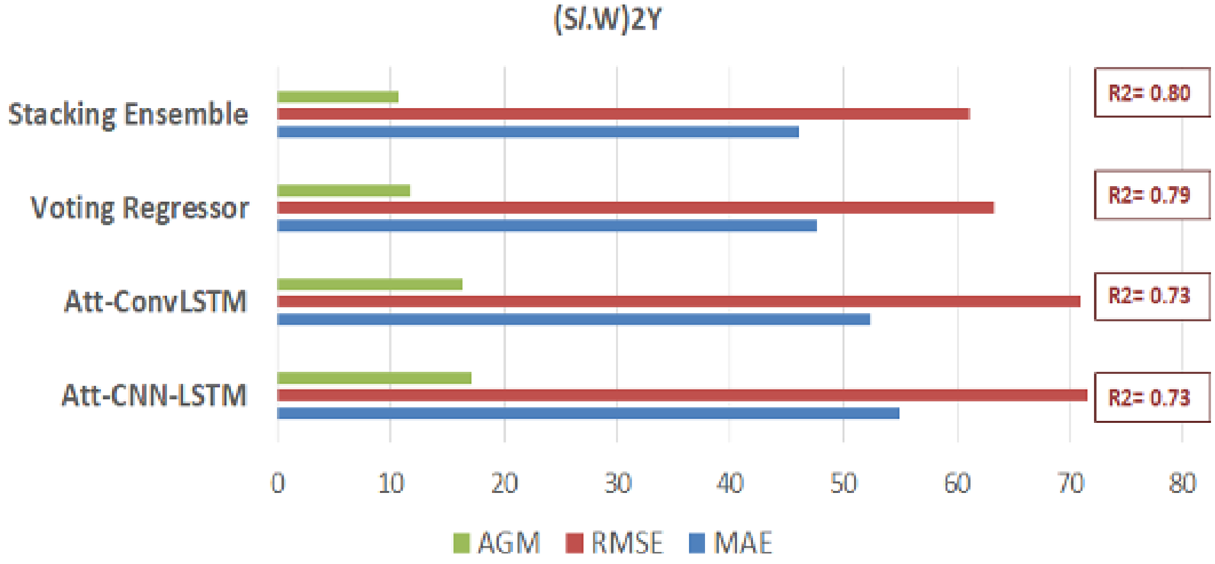


Figure 4.3: Results of Yield Forecasting Using Soil and Weather Parameters

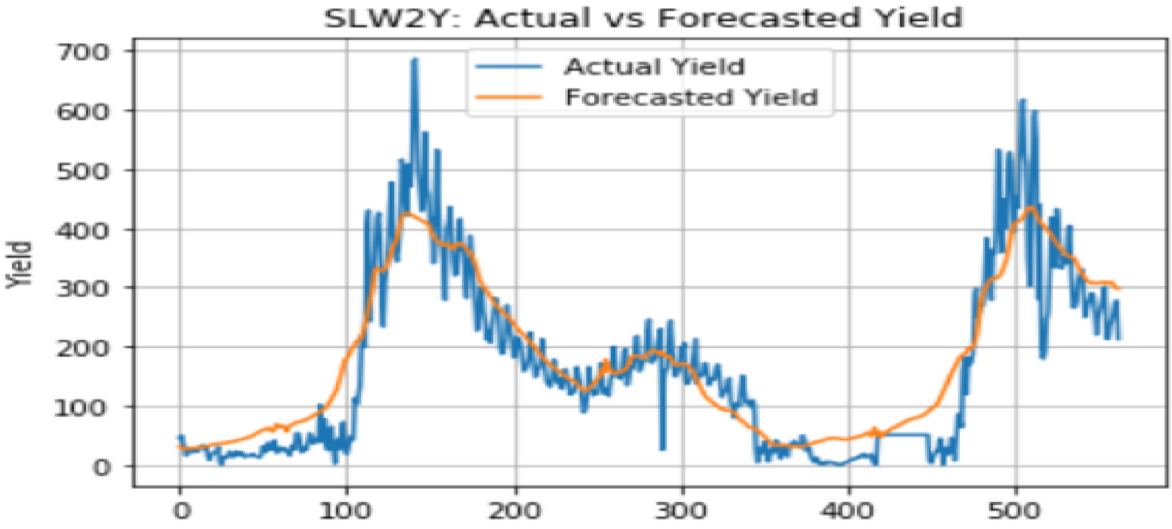


Figure 4.4: Actual vs Forecasted Yields Using Soil and Weather Parameters

4.1.3 Inferences and Conclusion

Following conclusions are drawn from the experiments conducted in the Section 4.1:

- It is inferred that better results are obtained when using both soil and weather parameters discussed in Section 4.1.1 than using the soil parameters alone as discussed in Section 4.1.2 or weather parameters alone as in [111].
- It is also observed that the stacking ensemble of the component deep learning models Att-ConvLSTM and Att-CNN-LSTM gives the best yield forecasting results for both experiments in Section 4.1.1 and 4.1.2.

4.2 Soil and Weather Parameters' Effect on Price Forecasting

Having an estimate of the prices of the fresh produce commodities helps food companies to bid a fair price to the distributors. To forecast an FP price close to the actual, deep Learning models should be trained using an effective set of parameters that affect the price. In this section, experiments are conducted to forecast the strawberry price using varying soil and weather parameter sets to discover the most effective parameters. Two experiments are conducted to forecast the strawberry price for Santa Barbara County of California; In the first, the soil parameters are used alone. In the second, weather parameters are added to the soil ones.

The dynamic soil data as discussed in Section 3.1.3, the weather data as discussed in Section 3.1.1 and the price data as discussed in Section 3.1.4. The data preprocessing for the inputs is done as described in Section 4.1. Following experiments have been conducted to forecast the strawberry yield using the soil and weather parameters.

4.2.1 Price Forecasting Using Soil Parameters

In this experiment the soil parameters alone are used to forecast the price and the results are reported based on all deployed performance measures for 5 weeks ahead. The dynamic soil data described in Chapter 3 are used in this experiment. The soil data is first preprocessed, the preprocessing steps include first creating a window of 20 weeks of data to predict the yield 35 days ahead, hence the prediction is for 5 weeks ahead yield. Thus, each single parameter has values for 140 days, 20 weeks x 7days, which naturally results into a lot of input parameters and thus high dimensionality. To pick the best parameters and to reduce the dimensionality of the data Principal Component Analysis is applied and the parameters accounting to 90% variance in the data are chosen. The price is also preprocessed using the exponential smoothing [79] to smooth out any spikes. The exponential smoothing is mostly used for data which has seasonality. Since, the price data had seasonality exponential smoothing is used.

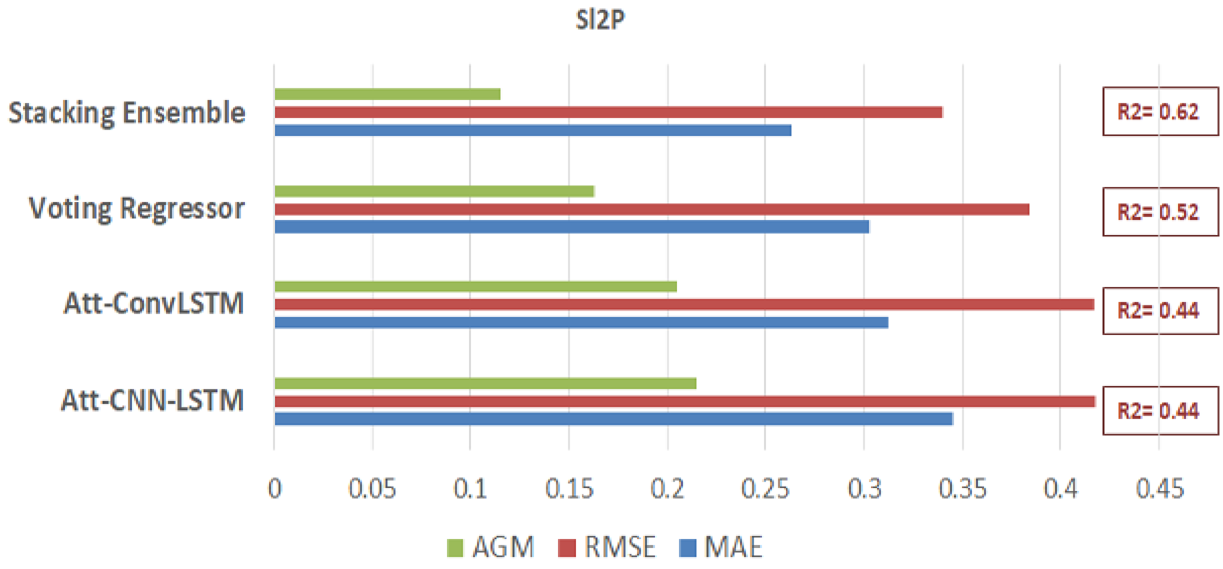


Figure 4.5: Results of Price Forecasting Using Soil and Weather Parameters

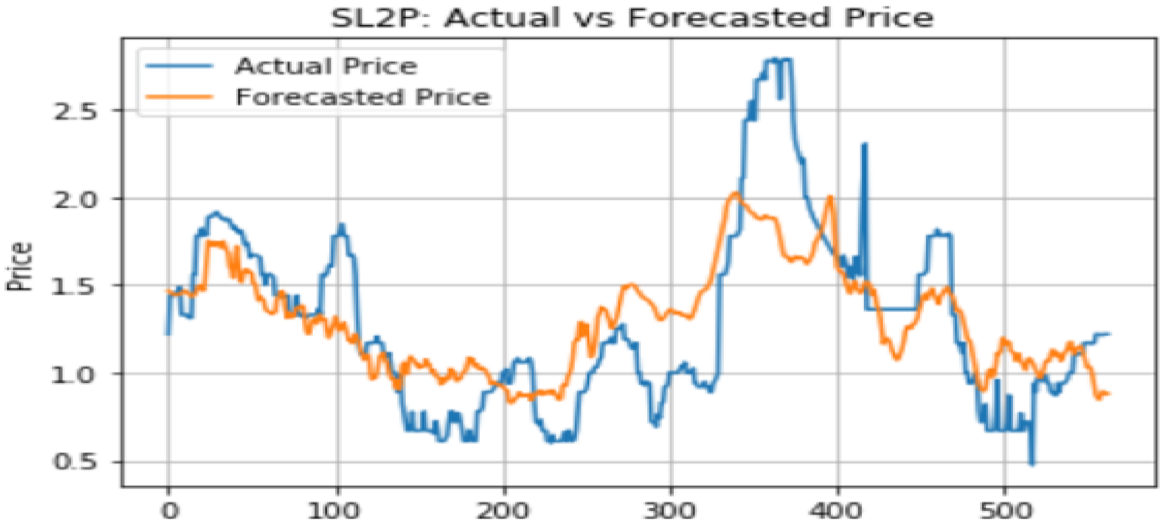


Figure 4.6: Actual vs Forecasted Prices Using Soil Parameters

The preprocessed data is then fed into two models Att-CNN-LSTM and Att-Conv-LSTM and then finally stacking ensemble and Voting Regressor are used to ensemble the results from these two models. As shown in the results in Figure 4.5, the Stacking Ensemble is the best performing with the least forecasting errors. Figure 4.6 shows the final forecasted values for days between 17-12-2017 and 03-07-2019.

4.2.2 Price Forecasting Using Soil and Weather Parameters

In this experiment the weather parameters are used again along with the soil parameters to forecast the price. The thirteen weather parameters described in Chapter 3 along with the dynamic soil parameters are fed into the two compound models Att-CNN-LSTM and Att-ConvLSTM to give the results shown in Figure 4.7. It is observed that, overall, the soil and weather parameters together perform better than just using the soil parameters as inputs to the model. The voting regressor ensemble and stacking ensemble is performed on the outputs from both the compound models and it is observed that Stacking Ensemble performs better than Voting Regressor. Figure 4.8 below shows the final forecasted values for days between 17-12-2017 and 03-07-2019. The soil and weather parameters together perform better than using the soil parameters alone.

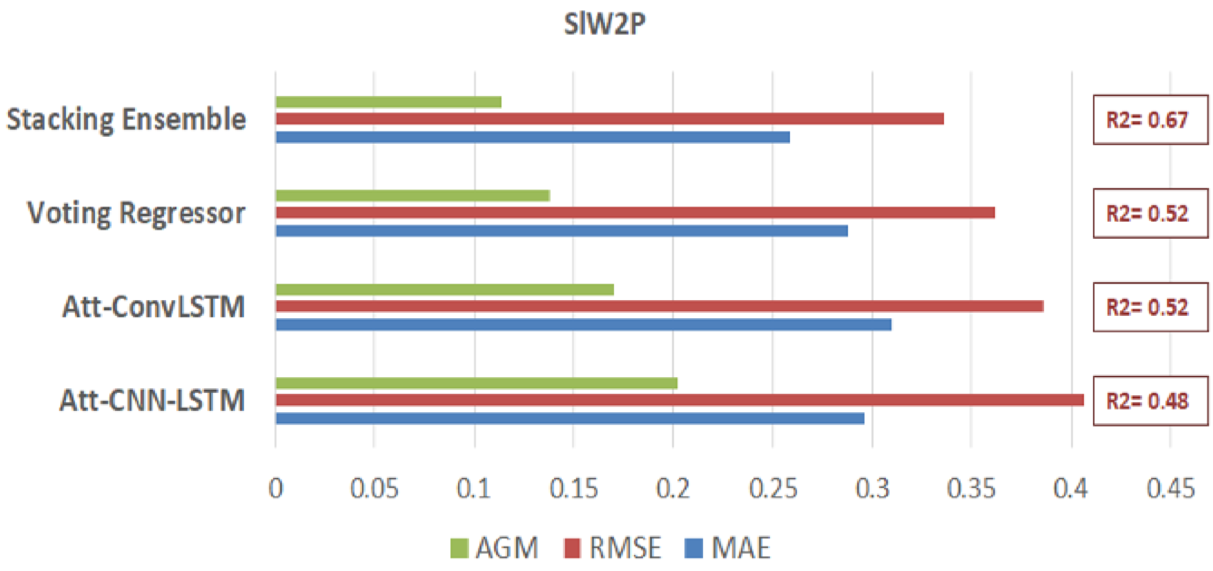


Figure 4.7: Results of Price Forecasting Using Soil and Weather Parameters

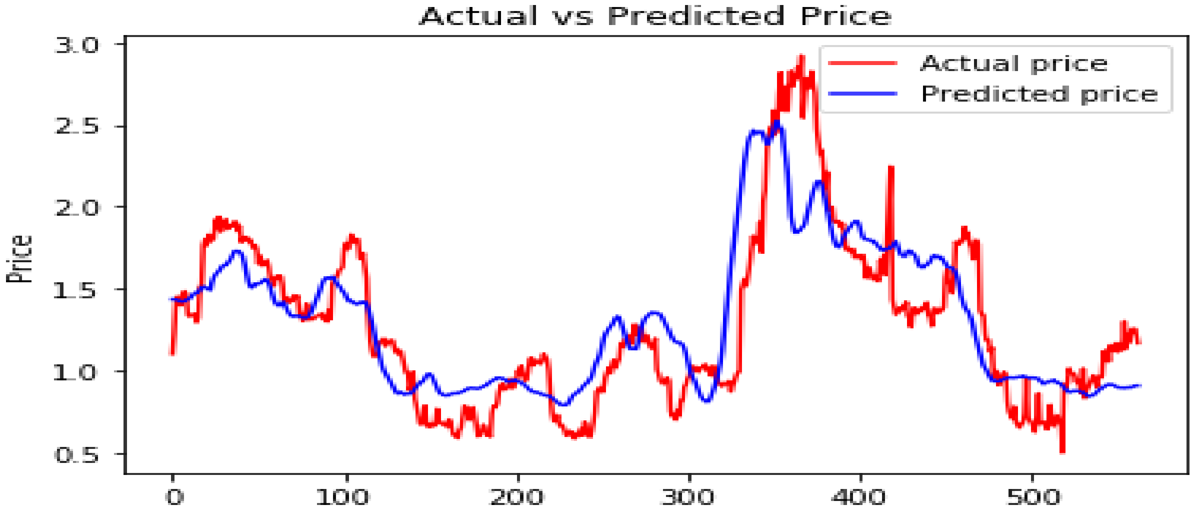


Figure 4.8: Actual vs Forecasted Price Using Soil and Weather Parameters

4.2.3 Inferences and Conclusion

- It is inferred that better results are obtained on using both soil and weather parameters discussed in Section 4.2.1 than using the soil parameters alone as discussed in 4.2.2 or weather parameters alone as in [111] for forecasting the price values.

- The stacking ensemble of the component deep learning models Att-ConvLSTM and Att-CNN-LSTM gives the best price forecasting results for both experiments in Section 4.2.1 and 4.2.2.

4.3 Finding the Best Performing Deep Learning Model for Price and Yield Forecasting

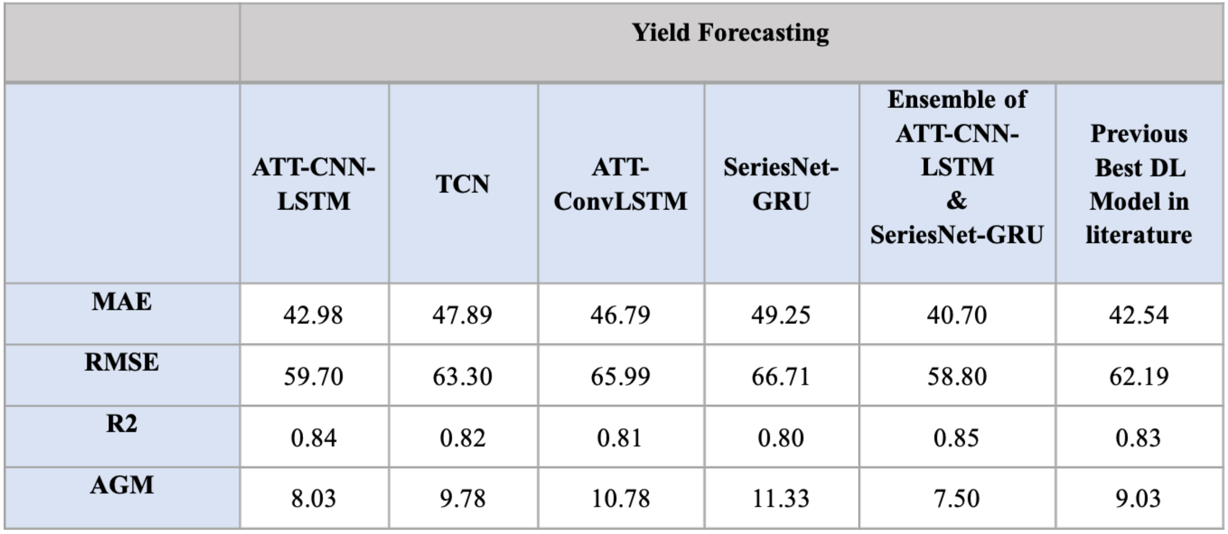
There are two experiments described in this section, both experiment use soil moisture and temperature parameters as inputs; the first experiment is strawberry yield forecasting and the second is strawberry price forecasting. These experiments are performed to find the best performing DL model for forecasting price and yield.

Various models are used in both the experiments to find the best possible forecasting results. The compound models used for experimentation are Att-CNN-LSTM, TCN, Att-ConvLSTM, SeriesNet with GRU, all these models are trained using the soil moisture and temperature data as input and the corresponding yield as output. There exists a lot of correlation amongst the soil and weather parameters therefore, choosing the most influential parameters is a major challenge. The Random Forest feature selection method in Python with scikit-learn is used and the soil moisture and temperature are selected accordingly as the most effective parameters. Another reason for using soil moisture and temperature is that these parameters can easily be extracted using the satellite imagery. Thus, out of the entire set of parameters only soil moisture and temperature is used as inputs to the models. For the TCNs, the time series input data is first transformed into encoded images using the Gramian Angular Fields and then inputted into the TCNs.

4.3.1 Yield Forecasting Using Soil Moisture and Temperature Parameters

This experiment is performed to find the best performing model for yield forecasting. The output yield from the compound DL models is ensembled using the voting regressor to find the best combination. It is found that the forecasted value of yield obtained from the voting regressor ensemble of SeriesNet-GRU and Att-CNN-LSTM models gives the best forecasting results.

Table 4.1: Evaluation Metric Values for all the Models Used for Yield Forecasting Using Soil Moisture and Temperature



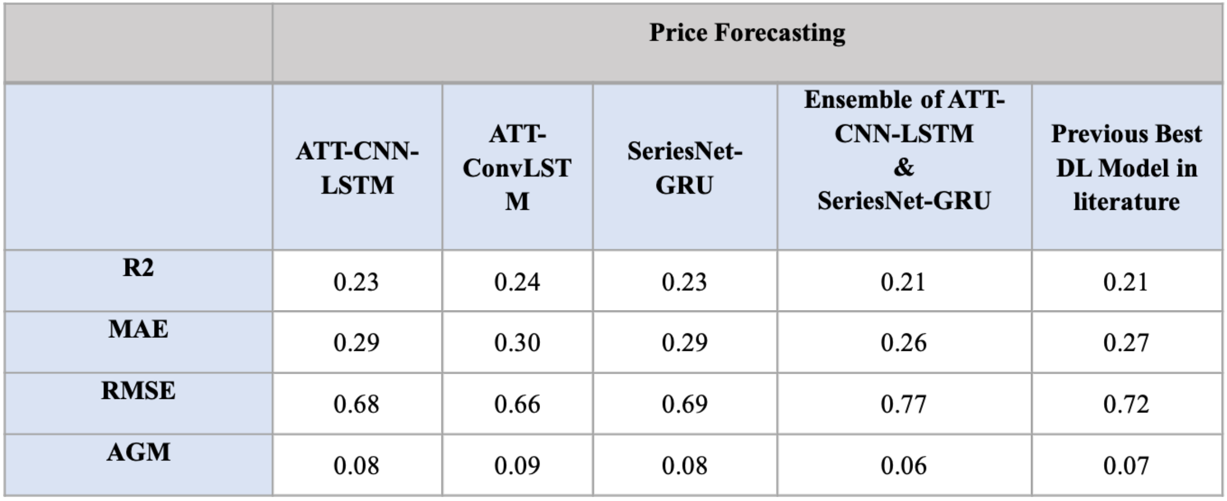
	Yield Forecasting					
	ATT-CNN-LSTM	TCN	ATT-ConvLSTM	SeriesNet-GRU	Ensemble of ATT-CNN-LSTM & SeriesNet-GRU	Previous Best DL Model in literature
MAE	42.98	47.89	46.79	49.25	40.70	42.54
RMSE	59.70	63.30	65.99	66.71	58.80	62.19
R2	0.84	0.82	0.81	0.80	0.85	0.83
AGM	8.03	9.78	10.78	11.33	7.50	9.03

Based on the experiment results listed in Table 4.1 , it is evident that the ensemble of the two compound models, SeriesNet-GRU and Att-CNN-LSTM, gives better results than each of the two individual models as well as literature results found in [111]. The possible reason behind the ensemble performing better lies in the fact that the ensemble captures the trends predicted by the component models. So, if one model fails to capture some trend the other might and the ensemble gives the value which is a combination of output from the component models.

4.3.2 Price Forecasting Using Soil Moisture and Temperature Parameters

This experiment is performed to find the best performing model for price forecasting. Various compound DL models are used for forecasting and the forecasting results from all possible combinations of models are ensembled to find that the ensembled forecasting values of Att-CNN-LSTM and SeriesNet-GRU outperform the other models.

Table 4.2: Evaluation Metric Values for all the Models Used for Price Forecasting Using Soil Moisture and Temperature



	Price Forecasting				
	ATT-CNN-LSTM	ATT-ConvLSTM	SeriesNet-GRU	Ensemble of ATT-CNN-LSTM & SeriesNet-GRU	Previous Best DL Model in literature
R2	0.23	0.24	0.23	0.21	0.21
MAE	0.29	0.30	0.29	0.26	0.27
RMSE	0.68	0.66	0.69	0.77	0.72
AGM	0.08	0.09	0.08	0.06	0.07

Table 4.2 contains the result of the experiment, we can clearly see that the Ensemble of the two compound models gives better result than the two individual models. It also outperforms the literature results found in [111]. The final forecasted value of price are the ensemble values from two models Att-CNN-LSTM and SeriesNet. The Voting Regressor is used for this purpose.

4.3.3 Inferences and Conclusion

- For the experiment in Section 4.3.1 it is observed that the yield value obtained using the Voting Regressor on the results of the two compound models SeriesNet-GRU and Att-CNN-LSTM is better that obtained by each of the individual models

- The AGM of the ensemble is 7% less than the best performing component model. The AGM value obtained in Section 4.3.1 experiment shows a 17% improvement than the AGM value obtained using the literature methods proposed in [111].

- For experiment in Section 4.3.2 it is observed that the price value obtained using the Voting Regressor on the results of the two compound models SeriesNet-GRU and Att-CNN-LSTM is better than the individual component compound model.

- A 25% improvement in AGM value is observed as compared to the best performing compound model. In Experiment in Section 4.3.2 a 14% improvement in AGM value is observed as compared to using the methods proposed in literature in [111].

4.4 Soil Parameters' Effect on Yield Forecasting

The main objective of the set of experiments in this section is to study the effect of considering the static soil parameters in addition to the dynamic ones on the performance of yield forecasting. To analyze the effect of using these static soil quality parameters, constant per county, with the frequently changing dynamic ones, four experiments are conducted in this section for forecasting apple yield; two with only dynamic parameters and two with both dynamic and static.

For the deployed dataset and preprocessing, the dynamic and static soil parameters are used as inputs for training the proposed DL model with the corresponding yearly apple yield specific to each county as the output. This data is collected for 15 counties in 6 Crop Reporting Districts as shown in Figure 3.4. The yearly apple yield data is collected from the United States Department of Agriculture, National Agricultural Statistics Survey (USDA-NASS) [13]. In total, 30 input parameters are considered including 15 static and 15 dynamic parameters. There is a total of 30 input variables to the proposed model: 15 static and 15 dynamic. Unfortunately, these variables are found to be highly correlated while the best scenario for training the DL model is to have input parameters that are least correlated with each other yet highly correlated with the output. Hence, to find this type of parameters, the random forest feature selection method in Python with scikit-learn [119] is used beside calculating the parameters' correlation and eliminating those parameters that are highly correlated. Based on these two methods the most effective eight parameters are selected; 4 dynamic parameters, which are solar radiation, soil temperature, soil moisture and PDSI, and four static parameters, which are the root zone available water (rootznaws), drought vulnerability (droughty), bulk density and organic matter. The apples in California are harvested around October, they grow during the summer season and ripen around the fall season [106]]. Hence, the growing period of apples is from May to October; 6 months or 184 days. The input to the utilized model is the daily value of all the parameters for a specific county within those six months per year. The output is the apple yield value for the corresponding county for that year. The static parameters remain constant everyday per county while the dynamic parameters vary daily or monthly per county over the six months. PDSI is a monthly varying parameter, hence, to have daily records the same value is repeated 30 or 31 times depending on the number of days in each month. Whereas the same values of the static parameters are repeated 184 times since they remain constant within 6 months. Therefore, the input to the model is in the form of a 3-D tensor equivalent to (8 x 184 x sample count); where (8 x 184) is the number of parameters or columns while sample count is the number of samples or rows in the data set [85]. Thus, there is a need for the dimensionality reduction of the dataset due to the high number of input parameters, 8 x 184 =1472 parameters. After normalizing the dataset, the Principal Component Analysis (PCA) [19] is used and the parameters contributing to maximum proportion of variance are chosen to train and test the model as input along with the corresponding apple yield output.

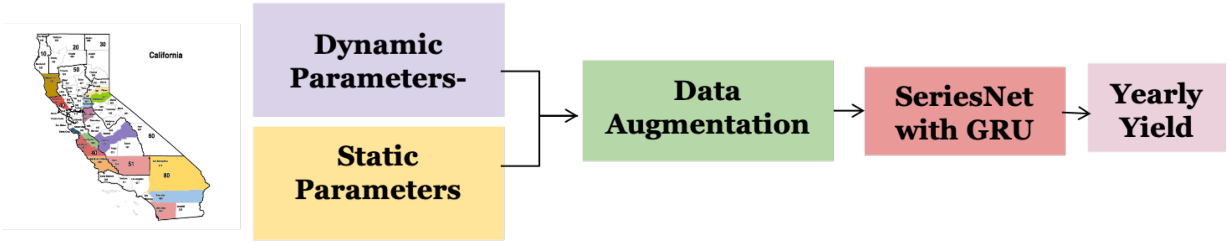


Figure 4.9: Chart for Experiment Using Static and Dynamic Parameters with Data Augmentation.

Four datasets are used as input, one per experiment, along with their corresponding yearly apple yield as output to train and test the forecasting DL model; SeriesNet with GRU and Attention. The four data sets have different shapes; two have the dynamic parameters with and without augmentation and the other two have both dynamic and static soil parameters with and without augmentation. Since only the annual yield values are forecasted, data augmentation is necessary for increasing the sample space to obtain better training results. The proposed model is trained for 200 epochs using each of the four data sets; Adam is the optimizer with a learning rate of 0.001 and the mean squared error is the loss function. The testing is conducted using 4 years of counties yield data; from year 2015 to 2018. Finally, the forecasting performance is measured for each of these data sets. Figure 4.9 shows the chart of the fourth experiment with the winning performance, based on the results listed in Table 4.3, in which the augmented dynamic and static parameters are used. Following experiments are performed to study the Soil Parameters' Effect on Yield Forecasting

4.4.1 Forecasting Yield Using Dynamic Parameters

In this experiment, the four selected dynamic parameters, solar radiation, soil temperature, soil moisture and PDSI across 184 days, are first fed into the PCA model for dimensionality reduction. The resulting first 47 parameters, with the maximum proportion of variance, are used for training the DL forecasting model along with the corresponding yield. Thus, the input shape of the training set is (495, 47).

4.4.2 Forecasting Yield Using Dynamic and Static parameters

In this experiment, all the eight selected parameters, both dynamic as well as static, including solar radiation, soil temperature, soil moisture, PDSI, rootznaws, droughty, bulk density and organic matter across 184 days are used. After the dimensionality reduction using PCA, the first 30 parameters with the maximum proportion of variance are used for training; the input shape of the training set is (495, 30).

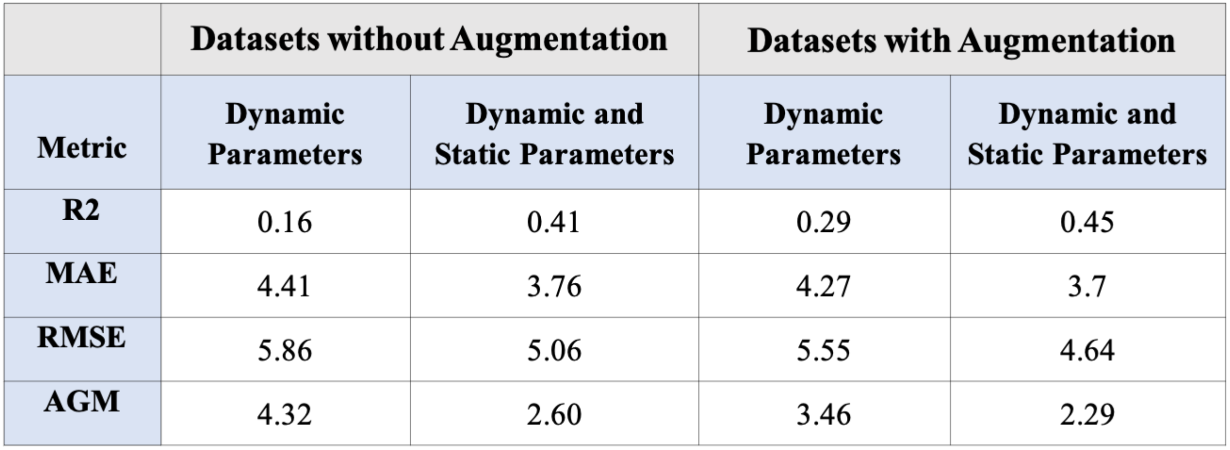
4.4.3 Forecasting Yield Using Augmented Dynamic parameters

In this experiment, the samples after augmentation are used along with the four selected dynamic parameters across 184 days. This data is fed into the PCA model and the first 40 parameters with the maximum proportion of variance are utilized in the DL model training; the input shape of the training set is (19008, 40).

4.4.4 Forecasting Yield Using Augmented Dynamic and Static Parameters

The samples after augmentation are used along with the eight selected dynamic and static parameters across 184 days. This data is fed into the PCA model and the first 24 parameters with the highest proportion of variance are used for training the model; the input shape of the training set is (19008, 24).

Table 4.3: Evaluation Metric Values for all the Four Experiments



	Datasets without Augmentation		Datasets with Augmentation	
Metric	Dynamic Parameters	Dynamic and Static Parameters	Dynamic Parameters	Dynamic and Static Parameters
R2	0.16	0.41	0.29	0.45
MAE	4.41	3.76	4.27	3.7
RMSE	5.86	5.06	5.55	4.64
AGM	4.32	2.60	3.46	2.29

4.4.5 Inferences and Conclusion

- Table 4.3 shows the numeric values of the different evaluation metrics, MAE, RMSE, R^2 score and AGM, for all the four conducted experiments. Table 4.3 results, with both augmented and nonaugmented datasets, show that adding the static soil parameters to the dataset reduces the forecasting AGM by around 34% compared to the case of excluding the static parameters; only using the dynamic parameters set.

- On the other hand, using the augmented training set to train the DL model improves the AGM value by 12% when tested with the nonaugmented test set as input.

- In addition, it is visually evident from Figure 4.10 that the forecasted yield curve fits better to the actual yield curve in the experiments using both static and dynamic soil parameters.

- It can also be seen that the forecasted values after augmentation become closer to the actual ones. From the obtained visual and tabular results, it is deduced that considering the static and dynamic parameters together enhances the forecasting performance over the case of solely relying on the dynamic ones while using augmentation further improves the obtained forecasts.

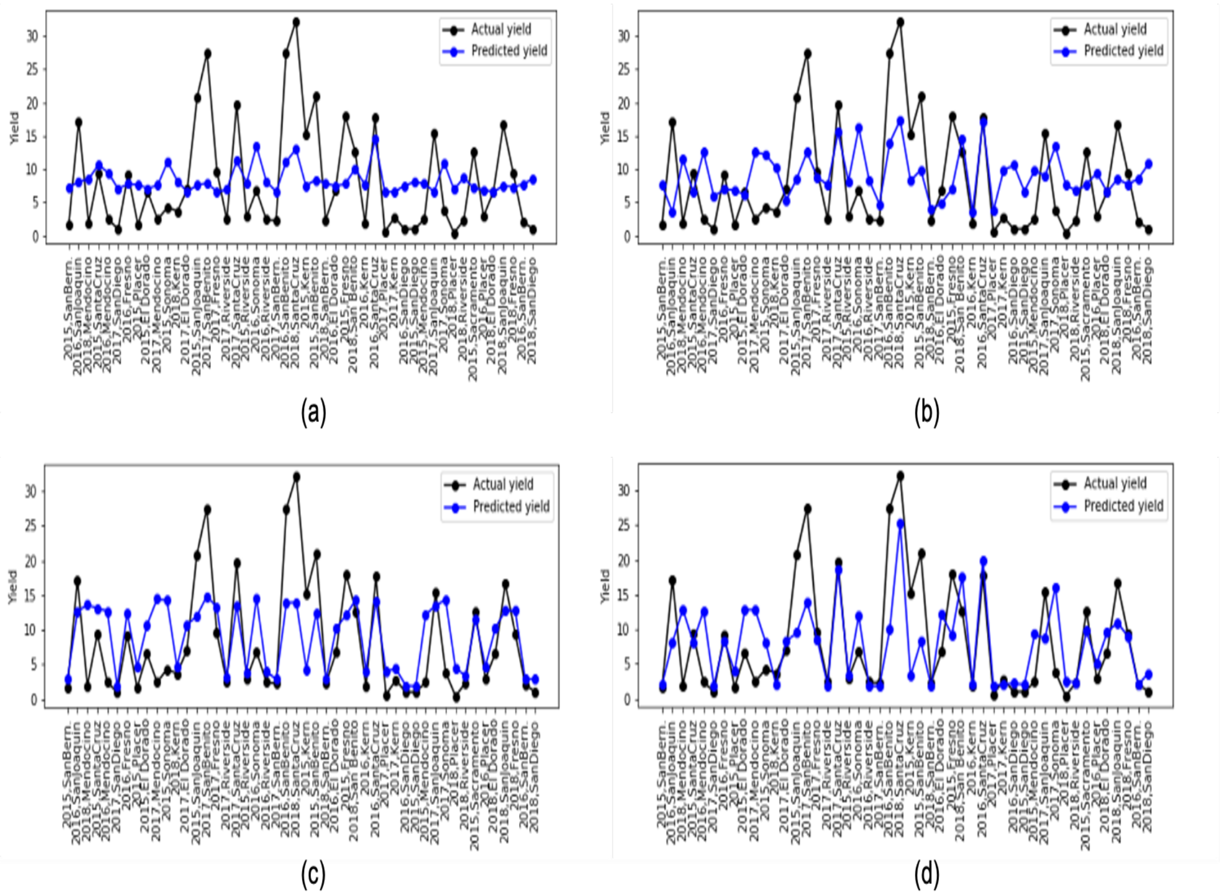


Figure 4.10: Four forecasted yield sets by the DL Model versus the actual yield values with nonaugmented test sets as input. The DL Model is trained in each of the four cases by a different training input set: (a) The Nonaugmented Dynamic Soil Parameters. (b) The Augmented Dynamic Soil Parameters. (c) The Nonaugmented Dynamic and Static Soil Parameters. (d) The Augmented Dynamic and Static Soil Parameters.

4.5 Transfer Learning for Yield Forecasting

To overcome the computational complexity of re training deep learning yield forecasting models for each type of fresh produce, it is necessary to have a generalization of the models' application to similar FP which is investigated in the set of experiments presented in this section. This generalization can be done by transferring the learning among similar FP with minimal retraining. Hence, it is decided to use Transfer Learning (TL) amongst berries which are similar in nature. First, the proposed DL model is trained using station-based data as inputs mapped to the strawberry yield as output. The weights obtained from this learning are transferred to the raspberry yield forecasting model since raspberry and strawberry yields are similar and are both planted in California. The proposed station-based ensemble model, ATT-CNN-LSTM-SeriesNet Ens, is an ensemble of two models: SeriesNet with Gated Recurrent Unit (GRU) and Convolutional Neural Network LSTM with Attention layer (Att-CNN-LSTM); trained and tested using station-based data as input and the corresponding strawberry yields as output. The weights obtained are transferred to the raspberry yield forecasting ensemble model with minimal retraining.

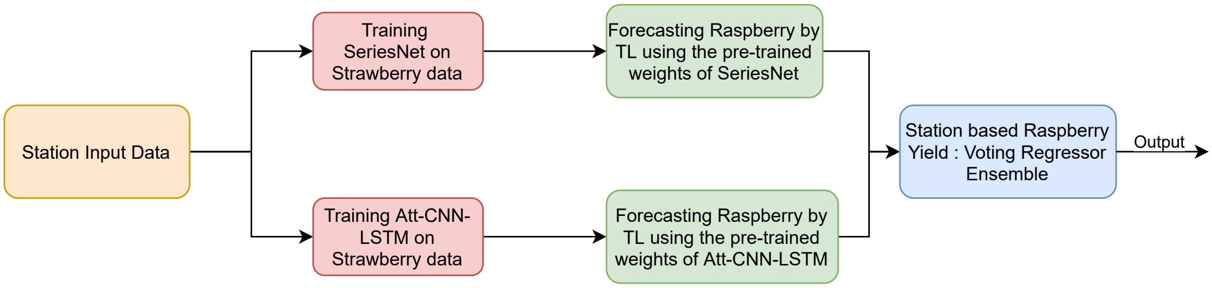

Station Input Data
Training SeriesNet on Strawberry data
Training Att-CNN-LSTM on Strawberry data
Forecasting Raspberry by TL using the pre-trained weights of SeriesNet
Forecasting Raspberry by TL using the pre-trained weights of Att-CNN-LSTM
Station based Raspberry Yield : Voting Regressor Ensemble
Output

Figure 4.11: Block Diagram of the Proposed Solution

For the deployed dataset and preprocessing, the station based soil data is obtained from the website of the National Oceanic and Atmospheric Administration [17], whereas the yield data for the strawberries and the raspberries is obtained from the website of the California Strawberry Commission [9]. Two input parameters are used for forecasting; soil moisture and soil temperature. The daily varying soil moisture and temperature values are considered for 140 days, which adds up to 2 parameters x 140 days = 280 input parameters. The preprocessing is performed as discussed in Section 4.1. After performing normalization of the input values and applying the Principal Component Analysis (PCA) [19] for dimensionality reduction, it is found that the first 36 parameters give the maximum proportion of variance. Therefore, these 36 input parameters are used for testing and training the station based models along with their corresponding yield values as output.

The preprocessed strawberry yield data is inputted into two compound DL models: Att-CNN-LSTM and SeriesNet with GRU. The weights of these two component models are then saved to perform the transfer learning to the raspberry yield forecasting DL models. Figure 4.11 shows the experimental setup.

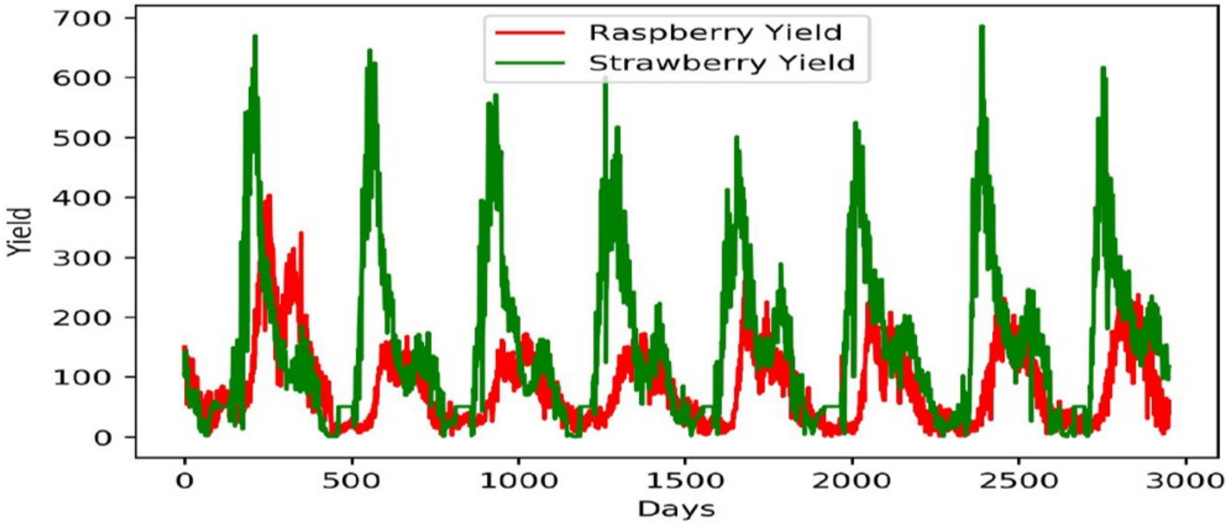


Figure 4.12: Plot for Strawberry and Raspberry Yields

From Figure 4.12, it can be seen that despite the fact that the strawberry yields are much more than raspberry yield, they still follow almost similar seasonality. Moreover, there are eight peaks for both strawberry and raspberry yields in a period of eight years. Similarity is decided based on having similar lag, similar seasonality, similar trend,... etc, [83] in their yield values across the time.

4.5.1 Using Pretrained Model with No Learning

In this experiment, the weights obtained for the DL models SeriesNet GRU and Att-CNN-LSTM by training them using the strawberry data are directly used as is to predict the raspberry yield without any retraining. Figure 4.13 represents the forecasted raspberry yield versus the original strawberry yield. Since, the weights are trained using the strawberry yield and strawberry yield is higher than the raspberry yield thus, the model predicts the raspberry yield value much greater than the actual raspberry yield value. Although the transfer learning follows the seasonality in the time series, it cannot predict the peaks, range, of yield

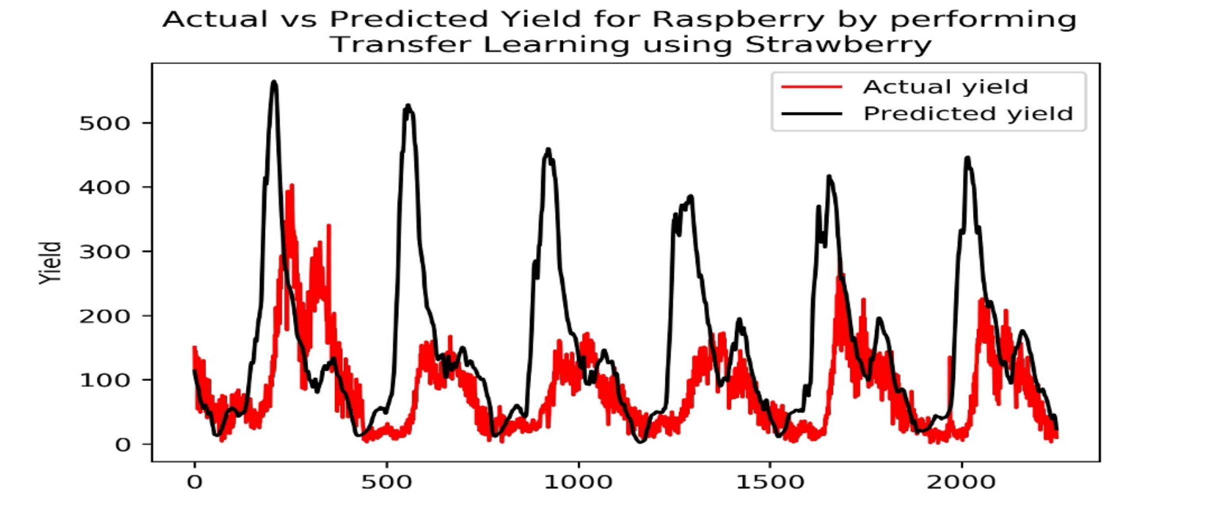


Figure 4.13: Actual vs Forecasted Raspberry Yield Using Pretrained Model on Strawberry Data with No Learning.

4.5.2 Using Pretrained Model with Minimal Learning

In this experiment, the weights obtained for the DL models, SeriesNet GRU and Att-CNN-LSTM, by training them on the strawberry data are used to predict the raspberry yield with some minimal retraining. For the transfer learning to the raspberry yield forecasting model, the base models Att-CNN-LSTM and SeriesNet with GRU are loaded with the pre-trained weights obtained from training them with the station-based input soil data along with the corresponding strawberry yield output. For the feature extraction from the raspberry yield data in transfer learning, additional dense layers with ReLU activation function are added atop the base models. The base models loaded with the pre-trained weights are frozen and only the added dense layers on the top are trained using the raspberry yield data. Finally, two sets of raspberry yield forecasts are obtained from the transfer learning performed using the two base models Att-CNN-LSTM and SeriesNet with GRU. The resulting forecasts are then combined using a voting regressor ensemble to give the final raspberry forecasted yield from the station-based soil data.

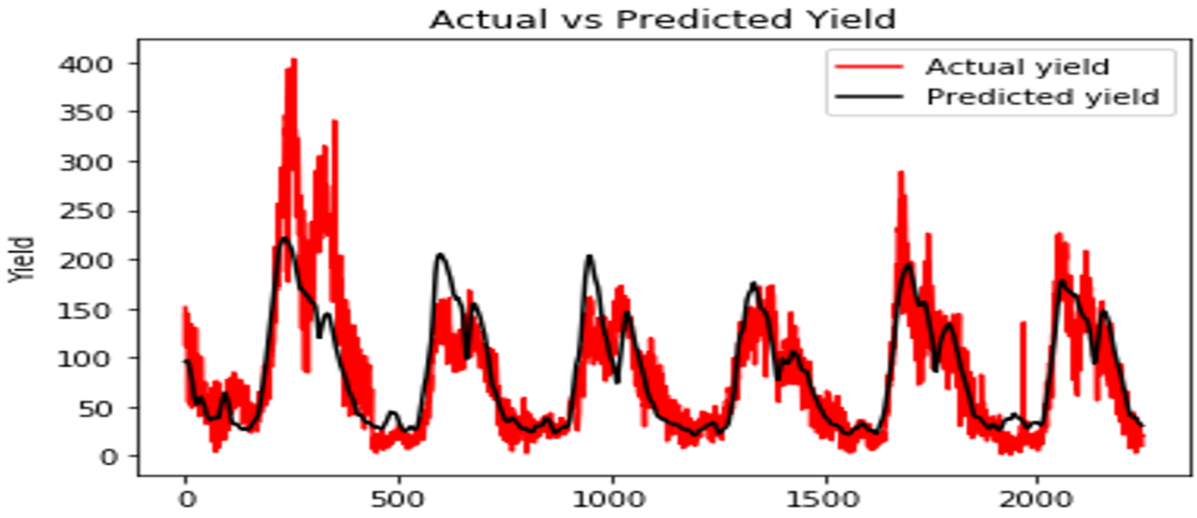


Figure 4.14: Actual vs. Forecasted Raspberry Yield Using Pretrained Model on Strawberry Data with Minimal Learning.

Figure 4.14 represents the predicted raspberry yield vs the original raspberry yield. Here the predicted yield closely follows the actual yield value.

4.5.3 Training Without any Pretrained Models

In this experiment, the DL models, SeriesNet GRU and Att-CNN-LSTM, are trained on the raspberry data from scratch without any transfer learning. The output from the DL models is then ensembled to give a final forecasted raspberry yield value. Figure 4.15 represents the predicted raspberry yield vs the original raspberry yield. The model performs well, and predicted yield follows actual yield.

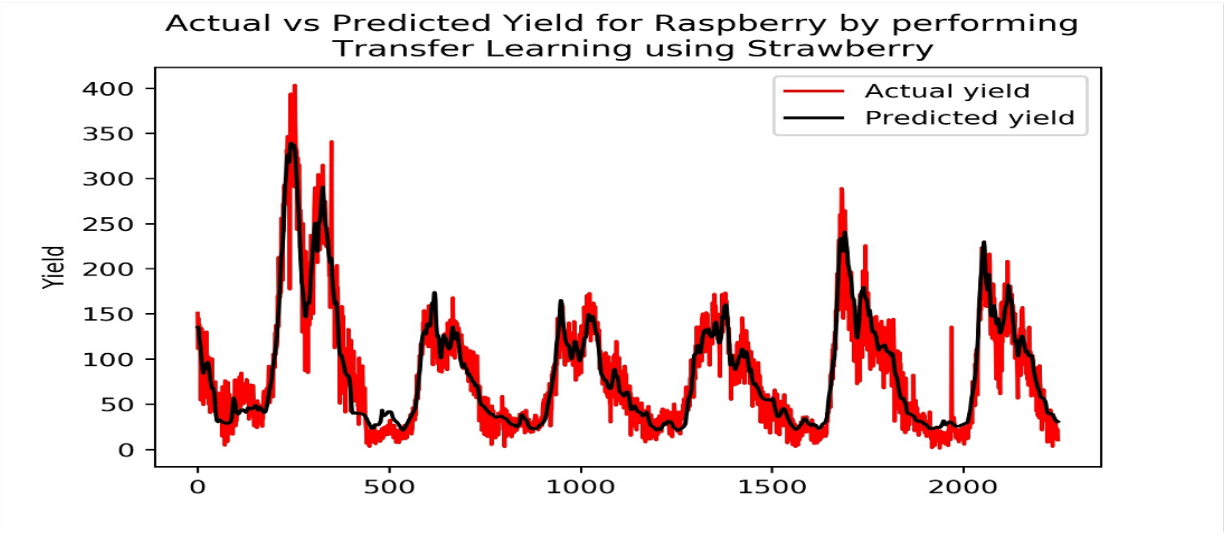


Figure 4.15: Actual vs Forecasted Raspberry Yield Without TL.

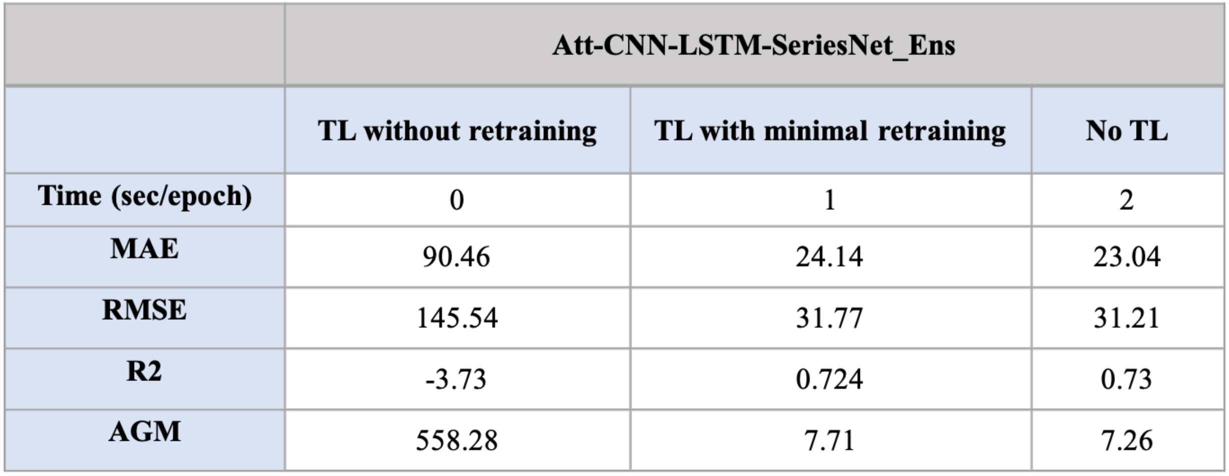
4.5.4 Inferences and Conclusion

Table 4.4 shows the results obtained on conducting various experiments to forecast the raspberry yield using TL. The following conclusions can be drawn from Table 4.4.

- The generalization of the deep learning forecasting models to forecast yields of other similar crops is investigated to promote the reusability of the pre-trained models and save the computational cost.

- The station-based data is used to train the ATT- CNN-LSTM-SeriesNet_Ens model with strawberry yields as the output. The learning from these models is then transferred to the raspberry yield forecasting model and the final forecasted raspberry yield is obtained by assembling the raspberry yield forecasted values obtained from both mentioned models.

Table 4.4: Results of Using Att-CNN-LSTM-SeriesNet_Ens to Forecast Raspberry Yield With and Without TL



	Att-CNN-LSTM-SeriesNet_Ens		
	TL without retraining	**TL with minimal retraining**	**No TL**
Time (sec/epoch)	0	1	2
MAE	90.46	24.14	23.04
RMSE	145.54	31.77	31.21
R2	-3.73	0.724	0.73
AGM	558.28	7.71	7.26

- It is found that using TL with the voting regressor reduces the processing time by 50% compared to the case of training it without TL. This proves that using transfer learning across similar crops is more efficient than performing complete retraining especially with large-scale datasets.

- The AGM value obtained from TL with minimal retraining is comparable to the AGM value obtained without performing TL. The AGM value obtained by no TL is just around 5% less than the AGM value obtained by TL with minimal retraining.

4.6 Chapter Summary

Four sets of experiments are conducted in this chapter:

- The first experiment deals with the forecasting of strawberry yield for Santa Barbara, California using various soil and weather parameters. The sub-experiments of this particular experiment are yield forecasting using the dynamic soil parameters alone, then yield forecasting using the dynamic soil and weather parameters and finally yield forecasting using soil moisture and temperature. The first two experiments are conducted to find the best set of parameters and it is found that soil and weather parameters together give better forecasting results than using the soil or weather parameters alone. The third experiment is performed to find the best performing model and it found that the voting regressor ensemble of Att-CNN-LSTM and SeriesNet-GRU gives the best forecasting results.

- The second experiment deals with the strawberry price forecasting for Santa Barbara, California. The sub-experiments include price forecasting using the dynamic soil parameters, forecasting using soil and weather parameters and finally strawberry price forecasting using soil moisture and soil temperature. The first and second experiment are conducted to find the best set of input parameters to forecast the prices and it is found that soil and weather parameters together give the best forecasting results. The third experiment is conducted to find the best performing DL model using the soil moisture and temperature as inputs to all the models.

- The third experiment set explores how using the different types of soil parameters affect the yield forecasting. Annual apple yield is forecasted using two different types of soil parameters, static and dynamic, across 15 counties in California. It is concluded that using the static and dynamic parameters together give better forecasting results.

- The fourth experiment set evaluates using transfer learning for yield forecasting for sake of generalization. This TL is conducted amongst similar types of berries, mainly raspberry and strawberry. Similarity is decided based on having similar lag, similar seasonality, similar trend,... etc, [83] in their yield values across the time. It is found that transfer learning gives results comparable to the forecasting results obtained when training from scratch while reducing the processing time .

Chapter 5

Forecasting Web Application

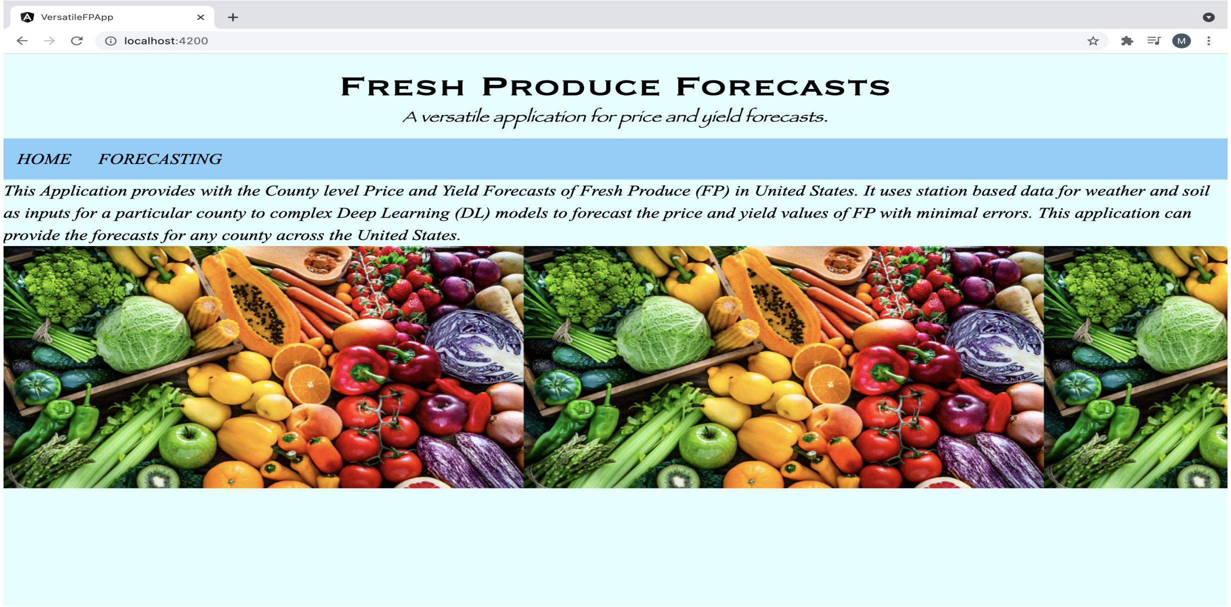
The work done in the thesis revolves around building deep learning models for forecasting the yield and prices of fresh produce. The work also aims to extend the learning to similar FP by performing transfer learning to avoid training from scratch. To enable the end-users to benefit from such models and facilitate end-to-end deployment of these models, a versatile fresh produce Time Series Forecasting web application is developed for forecasting price and yield of any fresh produce across any region by using the soil and weather parameters of that area as inputs to the DL models, using Angular CLI for the client side development and Flask framework at the back-end.

The purpose of the web application is to enable clients to use state-of-the-art deep learning models for forecasting the yield and price values of any desired fresh produce and in any region. This web application by default offers the clients options to choose from strawberries and raspberries for forecasting the price and yield of strawberry and just yield for raspberry. For counties it offers Santa Maria and Oxnard in California. If the client wants to have forecast for some other FP, then an option to upload the yield/price values will pop up. Similarly, if the user wants prediction for some other region which is not in drop down menu, then the user needs to upload the soil temperature and soil moisture data file for that region. If the user selects from the existing options of the FP and counties no retraining is needed, the pre-trained model is used for forecasting the results. In addition, if the FP whose forecast is needed is similar to an already available fresh produce then the training from scratch is not required and Transfer learning is performed.

The models are updated using latest weather, soil and FP price/yield data for the available counties. In the backend the latest soil and weather data for Santa Maria and Oxnard is downloaded automatically and after the period of lag which is 20 weeks in our case (for all berries), the model is is retrained using this new data. The user needs to provide the yield/price files for counties other than Santa Maria and if the crop is other than strawberry. The user also needs to provide the soil moisture and temperature data if the county is other than Oxnard or Santa Maria. Figure 5.1, Figure 5.2 and Figure 5.3 are screenshots of the developed web application on forecasting. Figure 5.1 depicts the home page of the application. Figure 5.2 depicts the forecasting web-page in which the user needs to enter the following details related to the forecast:

- **Forecast Type:** For this option the user needs to choose either *Price* option for price forecast or *Yield* option if the yield forecast is needed .
- **Horizon:** This option is a drop-down menu where the users need to choose how far ahead is the needed forecast: 1 day ahead, one week ahead, or two, three, four, five weeks ahead. 6 different sets of models are trained for different horizons. For each model the inputs are mapped to the output for the horizon ahead.
- **Fresh Produce:** In this menu, the user needs to select which fresh produce they are interested in, the available options are strawberry. If the user intends to find the forecast for some other FP then the user must upload the yield or price data file for that FP.
- **County:** In this menu the user needs to choose the county for which they looking for forecasts, the available options being Santa Maria. If the user wants to forecast for some other county, then a file containing soil moisture and temperature of that place needs to be uploaded.
- **Start Date:** This option is to select the starting date of the range of dates for forecast.
- **End Date:** This option lets the user select the ending date of the range for forecast.

If the forecast is just required for a specific date, then start date and end date should be the same. On clicking the save button the details are transferred to the backend where the forecasting using deep learning models is executed and the forecast values as well as their chart are displayed as shown in Figure 5.3. The generated chart depicts the y-axis forecasted yield/price for any date on the x-axis. The unit for price value is USD and for the yield is pounds/acre. The chart gives the client a clear view of the future trends.



FRESH PRODUCE FORECASTS

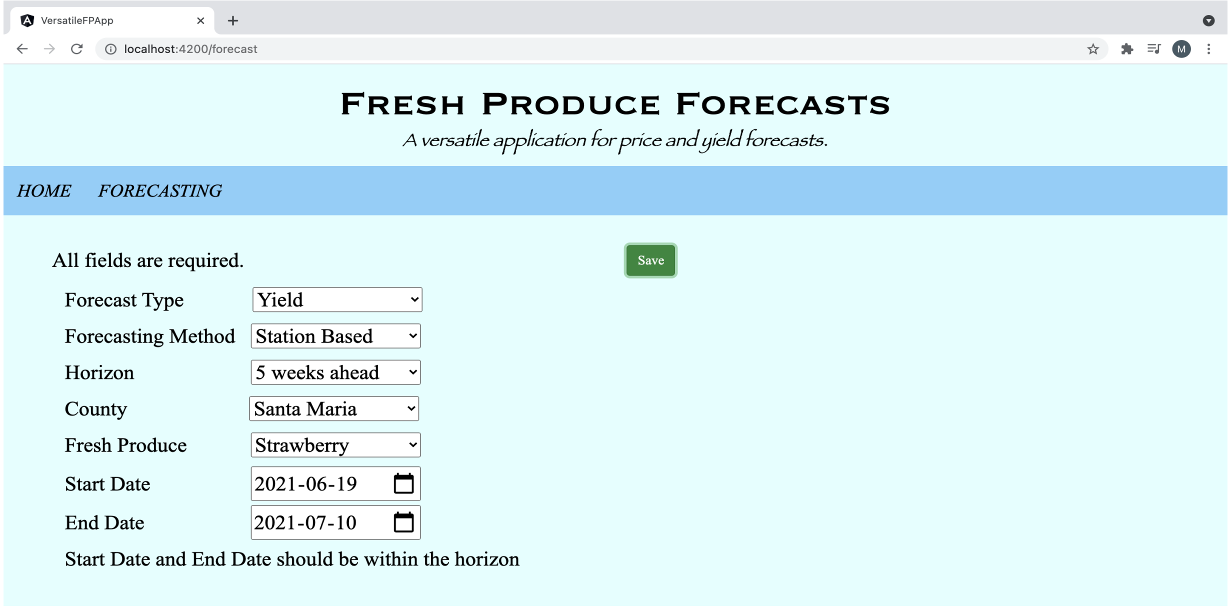
A versatile application for price and yield forecasts.

This Application provides with the County level Price and Yield Forecasts of Fresh Produce (FP) in United States. It uses station based data for weather and soil as inputs for a particular county to complex Deep Learning (DL) models to forecast the price and yield values of FP with minimal errors. This application can provide the forecasts for any county across the United States.

Figure 5.1: The Home-Page of the Application

5.1 Forecasting Strawberry Yield and Price

The deep learning models used for forecasting the strawberry yield and price are the ensemble of two compound DL models: Att-CNN-LSTM and SeriesNet with GRU. The application does not require any input from the user if the fresh produce or the county already exist in the list. The application takes in the input from the user when the county or the FP is not in the drop-down menu. The user needs to upload the price or yield values of the FP if the FP is not in the list. If the county is not in the list, the user needs to upload the weather/soil parameter data of that specific county. After obtaining these files, similarity is checked and if the uploaded data is similar to the studied one then transfer learning is performed and forecasts are provided. If the data is not similar to any of the studied FPs and counties then data-preprocessing is conducted in the backend to the inputted files, a lag is calculated and finally retraining is recommended with a suggested lag. On the other hand, if the user chooses from the available options, then no training is performed and the forecast is made using the pre-trained model for the desired dates. Figure 5.2 and 5.3 depict the results when the user fills the options which are available in the form already i.e. Strawberry as the fresh produce and Santa Maria as the county. The Download Forecasted Value Button helps in downloading the forecasted values.



Fresh Produce Forecasts

A versatile application for price and yield forecasts.

HOME *FORECASTING*

All fields are required. Save

Forecast Type: Yield

Forecasting Method: Station Based

Horizon: 5 weeks ahead

County: Santa Maria

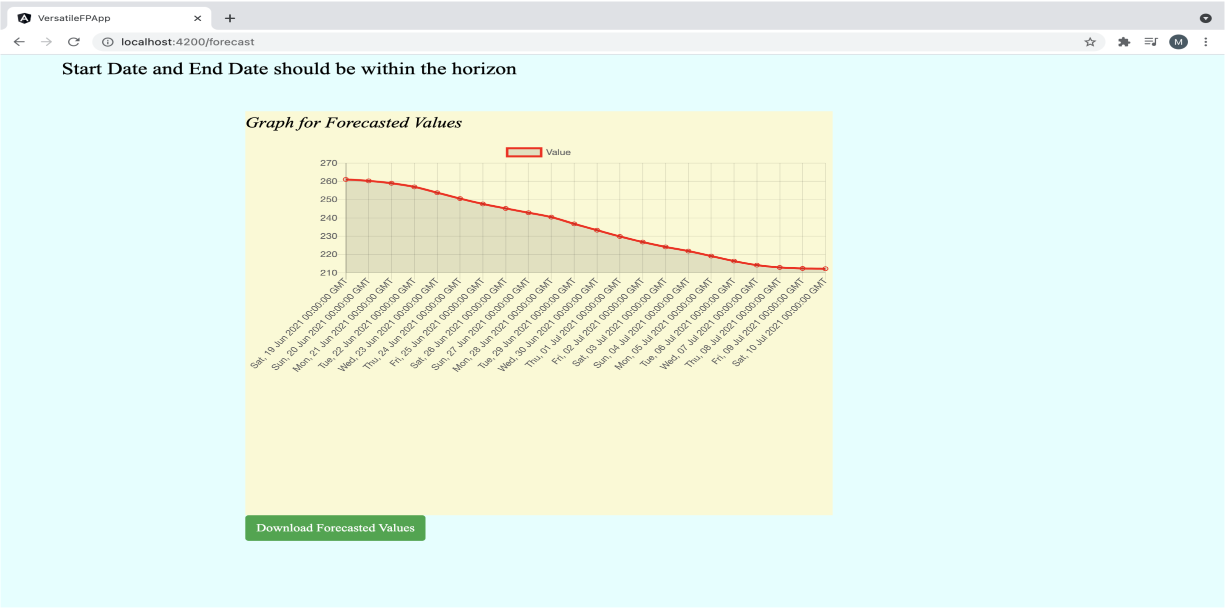
Fresh Produce: Strawberry

Start Date: 2021-06-19

End Date: 2021-07-10

Start Date and End Date should be within the horizon

Figure 5.2: Sample of Filled Fields for Yield Forecasting


VersatileFPApp
localhost:4200/forecast
Value
270
260
250
240
230
220
210
Sat, 19 Jun 2021 00:00:00 GMT
Sun, 20 Jun 2021 00:00:00 GMT
Mon, 21 Jun 2021 00:00:00 GMT
Tue, 22 Jun 2021 00:00:00 GMT
Wed, 23 Jun 2021 00:00:00 GMT
Thu, 24 Jun 2021 00:00:00 GMT
Fri, 25 Jun 2021 00:00:00 GMT
Sat, 26 Jun 2021 00:00:00 GMT
Sun, 27 Jun 2021 00:00:00 GMT
Mon, 28 Jun 2021 00:00:00 GMT
Tue, 29 Jun 2021 00:00:00 GMT
Wed, 30 Jun 2021 00:00:00 GMT
Thu, 01 Jul 2021 00:00:00 GMT
Fri, 02 Jul 2021 00:00:00 GMT
Sat, 03 Jul 2021 00:00:00 GMT
Sun, 04 Jul 2021 00:00:00 GMT
Mon, 05 Jul 2021 00:00:00 GMT
Tue, 06 Jul 2021 00:00:00 GMT
Wed, 07 Jul 2021 00:00:00 GMT
Thu, 08 Jul 2021 00:00:00 GMT
Fri, 09 Jul 2021 00:00:00 GMT
Sat, 10 Jul 2021 00:00:00 GMT
Download Forecasted Values

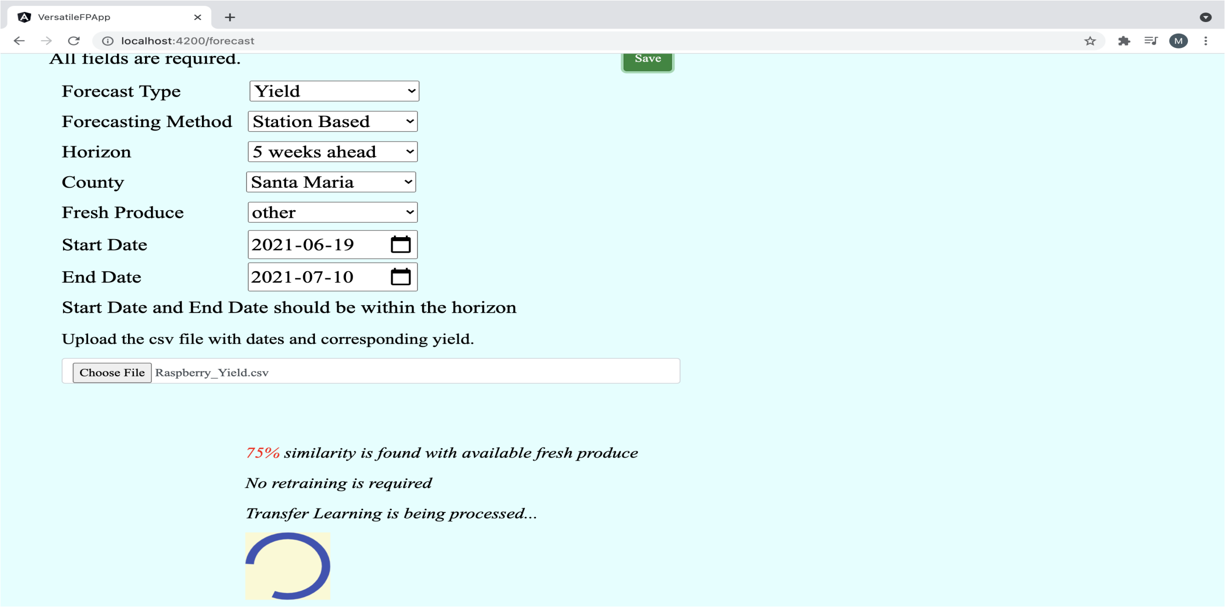
Start Date and End Date should be within the horizon

Graph for Forecasted Values

Figure 5.3: The Forecasted Yield Values as per Selected Options

5.2 Forecasting Yield and Price for Similar or Dissimilar FP.

For the scenario when the user wants the forecast for the counties provided in the dropdown list; Santa Maria but the fresh produce is outside the list then the user must upload the file with labels of yield or price for that fresh produce depending on the required forecast type. After obtaining the labels, the first step is to check similarity between the uploaded labels and the already existing fresh produce in the list. The similarity between two FP is checked, if two FP are of similar nature then there is no need of retraining the model with the new uploaded files rather transfer learning is used with little fine tuning and this saves time and leads to efficient forecasting. Figure 5.4 shows that when the user selects other FP and county is Santa Maria, the user is asked to upload the file of the fresh produce. After, selecting all the fields in form and submitting them, similarity is checked between the uploaded FP file and strawberry, if the two FP are similar then the user is prompted with percentage of similarity and transfer learning is performed in backend. Finally, the graph of the forecasted values is displayed as shown in Figure 5.5.



All fields are required.

Forecast Type	Yield
Forecasting Method	Station Based
Horizon	5 weeks ahead
County	Santa Maria
Fresh Produce	other
Start Date	2021-06-19
End Date	2021-07-10

Start Date and End Date should be within the horizon

Upload the csv file with dates and corresponding yield.

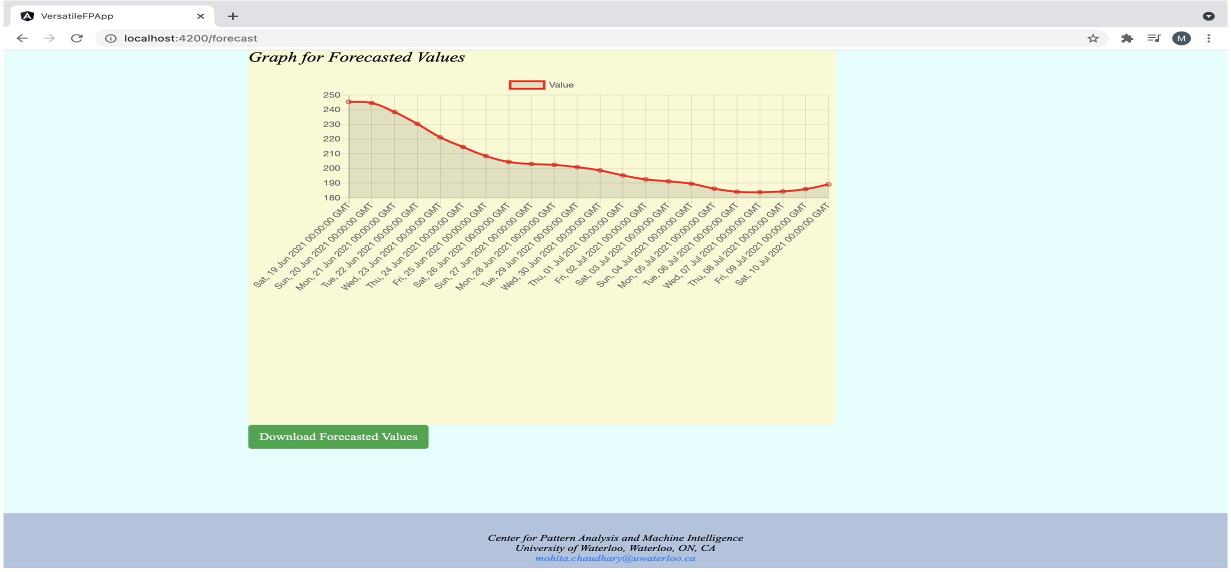
Raspberry_Yield.csv

75% similarity is found with available fresh produce

No retraining is required

Transfer Learning is being processed...

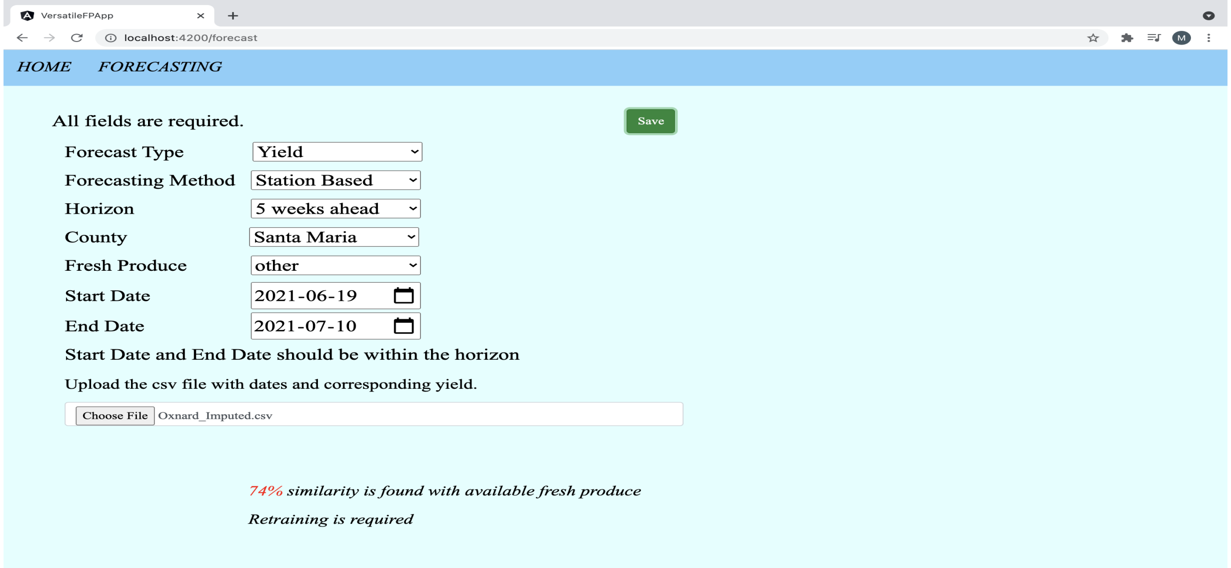
Figure 5.4: Sample of Filled Fields for Yield Forecasting when FP is not in List and is Similar


VersatileFPApp
localhost:4200/forecast
Graph for Forecasted Values
Value
250
240
230
220
210
200
190
180
Sat, 19 Jun 2021 00:00:00 GMT
Sun, 20 Jun 2021 00:00:00 GMT
Mon, 21 Jun 2021 00:00:00 GMT
Tue, 22 Jun 2021 00:00:00 GMT
Wed, 23 Jun 2021 00:00:00 GMT
Thu, 24 Jun 2021 00:00:00 GMT
Fri, 25 Jun 2021 00:00:00 GMT
Sat, 26 Jun 2021 00:00:00 GMT
Sun, 27 Jun 2021 00:00:00 GMT
Mon, 28 Jun 2021 00:00:00 GMT
Tue, 29 Jun 2021 00:00:00 GMT
Wed, 30 Jun 2021 00:00:00 GMT
Thu, 01 Jul 2021 00:00:00 GMT
Fri, 02 Jul 2021 00:00:00 GMT
Sat, 03 Jul 2021 00:00:00 GMT
Sun, 04 Jul 2021 00:00:00 GMT
Mon, 05 Jul 2021 00:00:00 GMT
Tue, 06 Jul 2021 00:00:00 GMT
Wed, 07 Jul 2021 00:00:00 GMT
Thu, 08 Jul 2021 00:00:00 GMT
Fri, 09 Jul 2021 00:00:00 GMT
Sat, 10 Jul 2021 00:00:00 GMT
Download Forecasted Values

Center for Pattern Analysis and Machine Intelligence
University of Waterloo, Waterloo, ON, CA
mohita.chaudhary@uwaterloo.ca

Figure 5.5: Forecasted Yield as per Selected Options when FP is not in List and is Similar

When the uploaded FP is not similar to strawberry, the user is prompted with the percentage similarity and "Retraining is recommended" is displayed, as shown in Figure 5.6.



All fields are required.

Forecast Type	Yield
Forecasting Method	Station Based
Horizon	5 weeks ahead
County	Santa Maria
Fresh Produce	other
Start Date	2021-06-19
End Date	2021-07-10

Start Date and End Date should be within the horizon

Upload the csv file with dates and corresponding yield.

Oxnard_Imputed.csv

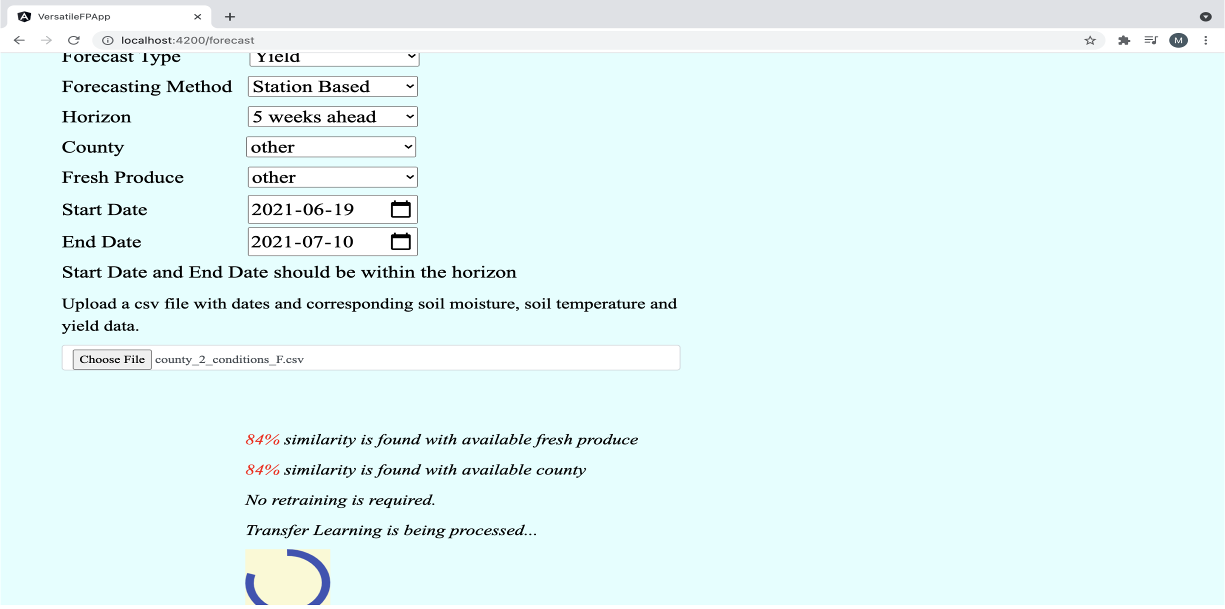
74% similarity is found with available fresh produce

Retraining is required

Figure 5.6: The Application Output when the Uploaded FP is Dissimilar

5.3 Forecasting Yield and Price for Similar or Dissimilar FP and county

For the scenario, when the user wants the forecast for the county which is not in the list. The user is asked to upload the .csv file with the soil moisture, soil temperature conditions and the FP yield/price values of that county as shown in Figure 5.7. If both the soil parameters and the label; yield/price time series are similar to that of Santa Maria and strawberry transfer learning is performed. Figure 5.8 shows the information about the percentage similarity of the uploaded time series; the features, soil moisture and soil temperature and the yield with the existing time series available in backend. A graph of the forecasted values is also displayed, which can be downloaded by clicking on the download button.



Forecast Type: Yield
Forecasting Method: Station Based
Horizon: 5 weeks ahead
County: other
Fresh Produce: other
Start Date: 2021-06-19
End Date: 2021-07-10

Start Date and End Date should be within the horizon

Upload a csv file with dates and corresponding soil moisture, soil temperature and yield data.

Choose File county_2_conditions_F.csv

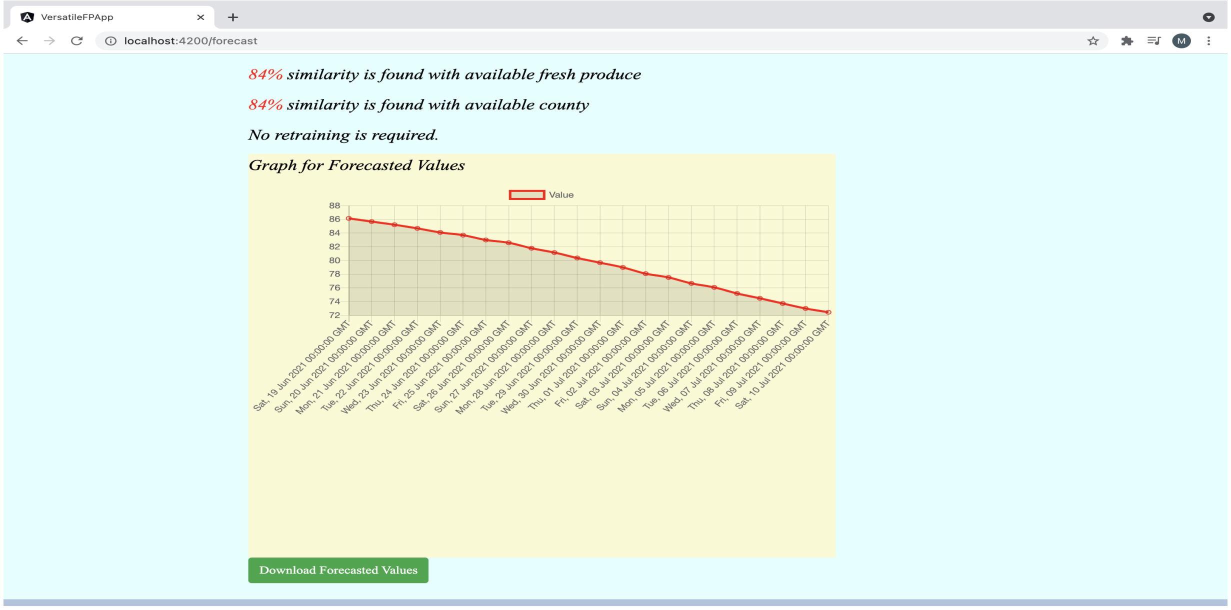
84% similarity is found with available fresh produce

84% similarity is found with available county

No retraining is required.

Transfer Learning is being processed...

Figure 5.7: Sample of Filled Fields for Yield Forecasting when both FP and County are not in List and are Similar to Available FP and County Data


VersatileFPApp
localhost:4200/forecast

84% similarity is found with available fresh produce

84% similarity is found with available county

No retraining is required.

Graph for Forecasted Values

Value
88
86
84
82
80
78
76
74
72
Sat, 19 Jun 2021 00:00:00 GMT
Sun, 20 Jun 2021 00:00:00 GMT
Mon, 21 Jun 2021 00:00:00 GMT
Tue, 22 Jun 2021 00:00:00 GMT
Wed, 23 Jun 2021 00:00:00 GMT
Thu, 24 Jun 2021 00:00:00 GMT
Fri, 25 Jun 2021 00:00:00 GMT
Sat, 26 Jun 2021 00:00:00 GMT
Sun, 27 Jun 2021 00:00:00 GMT
Mon, 28 Jun 2021 00:00:00 GMT
Tue, 29 Jun 2021 00:00:00 GMT
Wed, 30 Jun 2021 00:00:00 GMT
Thu, 01 Jul 2021 00:00:00 GMT
Fri, 02 Jul 2021 00:00:00 GMT
Sat, 03 Jul 2021 00:00:00 GMT
Sun, 04 Jul 2021 00:00:00 GMT
Mon, 05 Jul 2021 00:00:00 GMT
Tue, 06 Jul 2021 00:00:00 GMT
Wed, 07 Jul 2021 00:00:00 GMT
Thu, 08 Jul 2021 00:00:00 GMT
Fri, 09 Jul 2021 00:00:00 GMT
Sat, 10 Jul 2021 00:00:00 GMT
Download Forecasted Values

Figure 5.8: The Forecasted Yield Values as per Selected Options when both FP and County are not from List and are Similar to Available FP and County Data

Explanation of the flow-chart. Figure 5.9 depicts the flowchart of the application.

- The flowchart shows that application starts with the home page which has option for forecasting on selecting which the user is directed to a form.

- This form needs the user to enter forecast type, horizon, time period, fresh produce and county.

- If the FP and county are in the menu then the forecast is displayed immediately.

- If FP or county are not in list, then it is checked whether or not the FP is in list. If FP is in list then control moves to next step, if it is not in list then the user must upload the historical price/yield file for FP and then it moves to next step.

- The next step is to check if the county is in the list or not. If the county is in list then control moves to next step, if it is not then the user is supposed to upload the file for soil moisture and temperature data of that county as well as the file historical price/yield of that county and then move to next step.

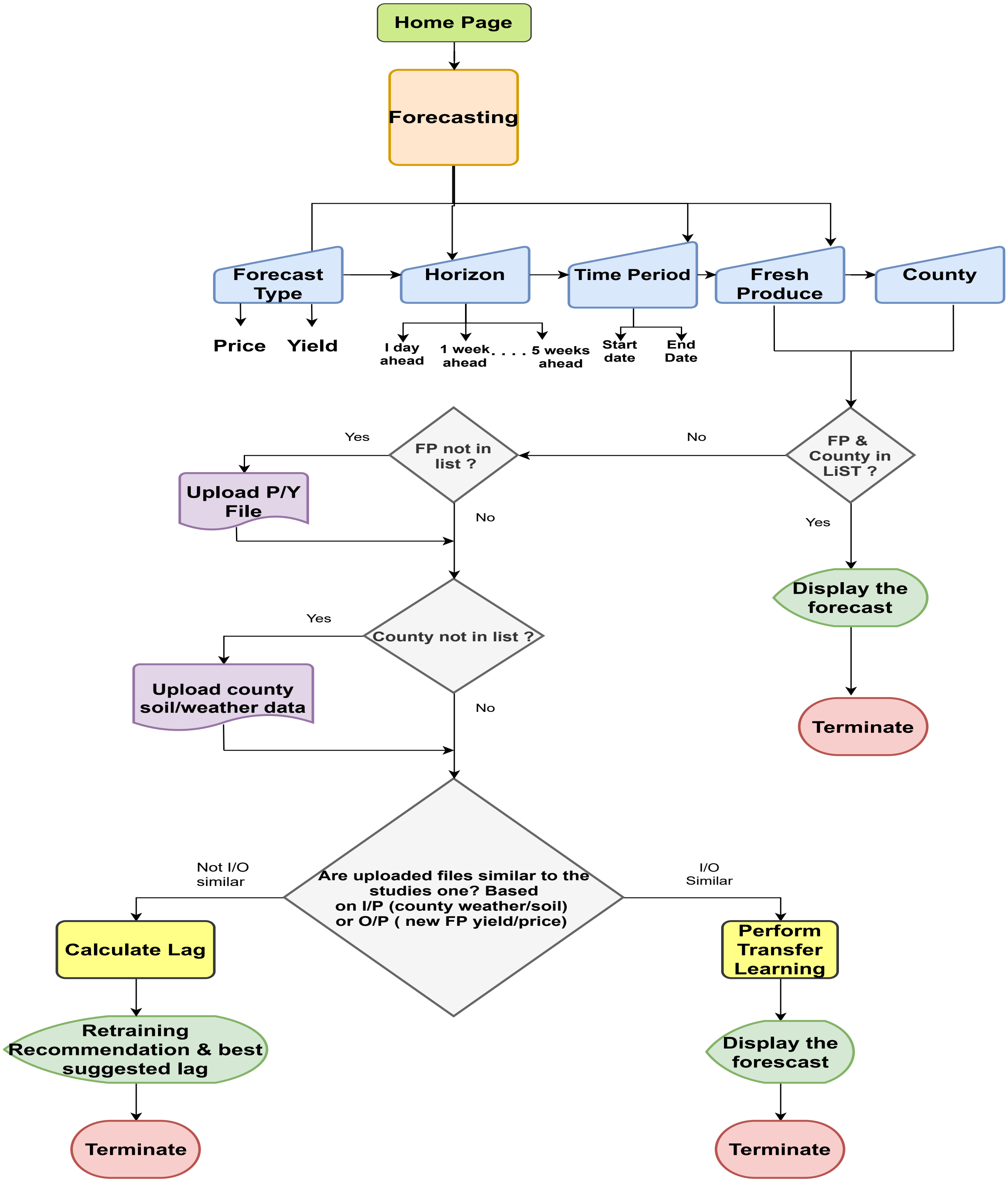


Figure 5.9: The Flow-Chart of Designed Application.

- After obtaining the files for county and FP, similarity check is performed. If the uploaded files are similar to the studied one then transfer learning is performed and forecast is displayed.
- If the uploaded files are not similar to the studied one then the lag is calculated and retraining is recommended.

5.4 Chapter Summary

This chapter explains the working of the application which is designed to forecast FP yield or price using the proposed models. The main points covered in this chapter can be summarized as follows:

- The chapter begins by stressing the need for the web application and lists the information the client needs to provide or feed into the application to get the forecasts.
- Snippets of the web application are provided, including the home page, web form needed to be filled and results page.
- The application flow-chart is provided to show the application input, execution paths based on the forecasting requests made by the user and the corresponding output.
- The chapter covers how transfer learning is performed when two time series are similar and hence reducing computational complexity required for forecasting in Section 5.3.
- The methodology used for forecasting is elaborated in Section 5.1.

Chapter 6

Conclusion and Future Work

The main objective of this work is to build state-of-the-art deep learning models for forecasting the yield and price for Fresh Produce. Most of the work which done in literature revolves around other agri-produce yield modelling and not much is done to tackle the perishable fresh produce items like the ones tack led in this work; strawberries, raspberries and apples. The models built in this work are capable enough to capture the information from the soil and weather data to forecast the yield and price values of the fresh produce. The deep learning models efficiently capture the trends in the data and precisely forecast the yield/price values. This thesis mainly explores the soil data and how it effects the yield and price forecasting, it analyses the effect of considering static and dynamic soil parameters on yield and price forecasting across various counties in California. The station based data is captured by installing the sensors in the fields which is a cumbersome process. This work uses the station-based soil and weather data for counties to forecast the yield/price of Fresh Produce.

The fresh produce selected for forecasting in this work are mainly the ones which have very short shelf life and come under the category of perishable goods. This thesis also proposes transfer learning amongst the crops which have a similar kind of yield and price time series for a particular region. It is found that transfer learning for similar fresh produce like strawberries and raspberries gives quite good results which is comparable to the results obtained by training the models from scratch. This work proposes state-of-the-art DL models which take as input the soil and weather data for forecasting which is quite a cost effective method to forecast the yield compared to the primitive survey-based method where surveyors are sent on the fields to find the estimate of yield for census purpose. In addition, literature suggests methods where drones have been installed in the fields to capture and estimate the apple yield using the pictures captured by them. The proposed method is quite effortless as compared to these primitive approaches.

Detailed experiments are carried out using various compound DL models Att-CNN-LSTM, SeriesNet with GRU and Attention, as well as TCNs on four major application areas: strawberry yield modelling using various soil and weather parameters as discussed in Section 4.1 where it is concluded that soil and weather parameters together give better results than using soil or weather parameters alone. Strawberry price modelling using soil and weather parameters as discussed in Section 4.2 where it is again found that using weather and soil parameters together gives better results compared to using individual parameters. Analysis of the effect of considering static and dynamic soil parameters on apple yield forecasting as discussed in Section 4.4 where it is concluded that using the static and dynamic soil parameters together gives better results than using the dynamic parameters alone. Generalization to other FPs, transfer learning for yield forecasting is discussed in Section 4.5 where it is concluded that transfer learning for fresh produce with similar time series gives comparable results to training from scratch and reduces the computational complexity incurred during the process of training the deep learning models.

The potential future work which can be done to take this work forward can be summarized as follows:

- Considering other external factors affecting the fresh produce yield like irrigation, fertilizers and pesticides added.
- The work done in this thesis considers the counties in California, United States. The possible future work should be to extend this work to other places as well.
- The experiment conducted to test transfer learning amongst similar crops considered berries. This work can be extended to other similar kind of fresh produce as well.
- Clusters of similar types of fresh produce can be identified and transfer learning amongst similar clusters can be used for forecasting without training from scratch promoting re-usability and reducing the computational complexity as well.
- Forecasting the farm-gate prices as a function of yield should be considered to be able to transfer the forecasted yield into price.
- This work uses the station-based data provided by the official agricultural websites of the United States. The station-based data might not be available for all the areas around the world and therefore, a more reliable source of data should be considered like the satellite-imagery.

References

[1] Agricultural marketing resource center. https://www.agmrc.org/commodities-products/fruits. accessed 27.01.2021.

[2] Annual revenue of the food manufacturing industry in canada from 2012 to 2018, https://www.statista.com/statistics/734870/annual-revenue-of-food-manufacture-in-canada/statisticContainer. accessed 18.02.2021.

[3] Artificial intelligence and the fresh food supply chain, https://insidebigdata.com/2019/04/01/artificial-intelligence-and-the-fresh-food-supply-chain/. accessed 18.02.2021.

[4] California irrigation management information system (cimis), https://cimis.water.ca.gov/WSNReportCriteria.aspx. accessed 10 January 2021.

[5] Canadian grocer, https://publications.virtualpaper.com/canadian-grocer/cg02rw/12/. accessed 18.02.2021.

[6] Encoding time series as images, https://medium.com/analytics-vidhya/encoding-time-series-as-images-b043becbdbf3. accessed 18.02.2021.

[7] An introduction to convlstm, https://medium.com/neuronio/an-introduction-to-convlstm-55c9025563a7. accessed 10 January 2021.

[8] The national drought mitigation center, drought risk atlas, https://droughtatlas.unl.edu/Data/Climate.aspx. accessed 10 January 2021.

[9] Unite states department of agriculture, web soil survey (usda-wss), url = ”https://websoilsurvey.sc.egov.usda.gov/App/WebSoilSurvey.aspx. accessed 10 January 2021.

[10] United nations general assembly, transforming our world: The 2030 agenda forsustainable development, https://sustainabledevelopment.un.org/post2015/transformingourworld/publication. accessed 18.02.2021.

[11] United nations world food programme, https://www.wfp.org/publications/hunger-map-2020. accessed 18.02.2021.

[12] United states department of agriculture, non citrus fruits and nuts 2019 summary. https://downloads.usda.library.cornell.edu/usda-esmis/files/zs25x846c/0g3551329/qj72pt50f/ncit0520.pdf. accessed 10 January 2021.

[13] United states department of agriculture,national agricultural statistics survey (usda-nass), https://www.nass.usda.gov/Statistics$_byState/California/Publications/AgC$ $omm/index.php$. accessed 10 January 2021.

[14] “translation invariance” http://www.cogsci.ucsd.edu/ rik/courses/readings/plunkett97-RIEg/chapter7.pdf. accessed 18.02.2021.

[15] The california strawberry commission website, https://www.calstrawberry.com/en-us/, n.d. accessed 27.11.2020.

[16] Details on the soil parameters, https://www1.ncdc.noaa.gov/pub/data/uscrn/products/hourly02/README.txt, n.d. accessed 27.11.2020.

[17] National Oceanic and Atmospheric Administrations, https://www.noaa.gov/, n.d. accessed 27.11.2020.

[18] Palmer drought severity index and palmer z-index, http://www.worldwindsinc.com/palmer.htm, n.d. accessed 27.11.2020.

[19] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley Interdisciplinary Reviews: computational statistics*, 2(4):433–459, 2010.

[20] Hamed Habibi Aghdam and Elnaz Jahani Heravi. Guide to convolutional neural networks. *New York, NY: Springer*, 10:978–973, 2017.

[21] Nesreen K Ahmed, Amir F Atiya, Neamat El Gayar, and Hisham El-Shishiny. An empirical comparison of machine learning models for time series forecasting. *Econometric Reviews*, 29(5-6):594–621, 2010.

[22] David LJ Alexander, Alexander Tropsha, and David A Winkler. Beware of r 2: simple, unambiguous assessment of the prediction accuracy of qsar and qspr models. *Journal of Chemical Information and Modeling*, 55(7):1316–1322, 2015.

[23] Md Zahangir Alom, Tarek M Taha, Chris Yakopcic, Stefan Westberg, Paheding Sidike, Mst Shamima Nasrin, Mahmudul Hasan, Brian C Van Essen, Abdul AS Awwal, and Vijayan K Asari. A state-of-the-art survey on deep learning theory and architectures. *Electronics*, 8(3):292, 2019.

[24] Orly Enrique Apolo-Apolo, Manuel Pérez-Ruiz, Jorge Martínez-Guanter, and João Valente. A cloud-based environment for generating yield estimation maps from apple orchards using uav imagery and a deep learning technique. *Frontiers in Plant Science*, 11:1086, 2020.

[25] George Athanasopoulos and Rob J Hyndman. Modelling and forecasting australian domestic tourism. *Tourism Management*, 29(1):19–31, 2008.

[26] George Athanasopoulos, Don Stephen Poskitt, and Farshid Vahid. Two canonical varma forms: Scalar component models vis-à-vis the echelon form. *Econometric Reviews*, 31(1):60–83, 2012.

[27] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

[28] Mukund Balasubramanian, Eric L Schwartz, Joshua B Tenenbaum, Vin de Silva, and John C Langford. The isomap algorithm and topological stability. *Science*, 295(5552):7–7, 2002.

[29] N Barik. Analysis of interventions addressing farmer distress in rajasthan. *Rajasthan Priorities, Copenhagen Consensus Center*, 2018.

[30] Silvio Barra, Salvatore Mario Carta, Andrea Corriga, Alessandro Sebastian Podda, and Diego Reforgiato Recupero. Deep learning and time series-to-image encoding for financial forecasting. *IEEE/CAA Journal of Automatica Sinica*, 7(3):683–692, 2020.

[31] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[32] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

[33] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[34] Leo Breiman. Stacked regressions. *Machine Learning*, 24(1):49–64, 1996.

[35] Denny Britz. Recurrent neural networks tutorial. *URL: http://www. wildml. com/2015/09/recurrentneural-networks-tutorialpart-1-introduction-to-rnns/[accessed June 28, 2019]*, 2015.

[36] 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.

[37] Jean C Buzby, Jeffrey Hyman, Hayden Stewart, and Hodan F Wells. The value of retail-and consumer-level fruit and vegetable losses in the united states. *Journal of Consumer Affairs*, 45(3):492–515, 2011.

[38] Ruichu Cai, Binjun Zhu, Lei Ji, Tianyong Hao, Jun Yan, and Wenyin Liu. An cnn-lstm attention approach to understanding user query intent from online health communities. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 430–437. IEEE, 2017.

[39] Chris Chatfield and Mohammed Yar. Prediction intervals for multiplicative holt-winters. *International Journal of Forecasting*, 7(1):31–37, 1991.

[40] Vikas Chawla, Hsiang Sing Naik, Adedotun Akintayo, Dermot Hayes, Patrick Schnable, Baskar Ganapathysubramanian, and Soumik Sarkar. A bayesian network approach to county-level corn yield prediction using historical data and expert knowledge. *arXiv preprint arXiv:1608.05127*, 2016.

[41] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific Reports*, 8(1):1–12, 2018.

[42] Peng Chen, Aichen Niu, Duanyang Liu, Wei Jiang, and Bin Ma. Time series forecasting of temperatures using sarima: An example from nanjing. In *IOP Conference Series: Materials Science and Engineering*, volume 394, page 052024. IOP Publishing, 2018.

[43] Hong Cheng, Lutz Damerow, Yurui Sun, and Michael Blanke. Early yield prediction using image analysis of apple fruit and tree canopy features with neural networks. *Journal of Imaging*, 3(1):6, 2017.

[44] Yepeng Cheng, Zuren Liu, and Yasuhiko Morimoto. Attention-based seriesnet: An attention-based hybrid neural network model for conditional time series forecasting. *Information*, 11(6):305, 2020.

[45] Yepeng Cheng and Yasuhiko Morimoto. Triple-stage attention-based multiple parallel connection hybrid neural network model for conditional time series forecasting. *IEEE Access*, 9:29165–29179, 2021.

[46] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[47] Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F Stewart, and Jimeng Sun. Doctor ai: Predicting clinical events via recurrent neural networks. In *Machine learning for healthcare conference*, pages 301–318. PMLR, 2016.

[48] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[49] Dan Ciregan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3642–3649. IEEE, 2012.

[50] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167, 2008.

[51] Pierre Comon. Independent component analysis, a new concept? *Signal Processing*, 36(3):287–314, 1994.

[52] Michael AA Cox and Trevor F Cox. Multidimensional scaling. In *Handbook of data visualization*, pages 315–347. Springer, 2008.

[53] Jesús Crespo Cuaresma, Jaroslava Hlouskova, Stephan Kossmeier, and Michael Obersteiner. Forecasting electricity spot-prices using linear univariate time-series models. *Applied Energy*, 77(1):87–106, 2004.

[54] Sandya De Alwis, Yishuo Zhang, Myung Na, and Gang Li. Duo attention with deep learning on tomato yield prediction and factor interpretation. In *Pacific Rim International Conference on Artificial Intelligence*, pages 704–715. Springer, 2019.

[55] Rahul Dey and Fathi M Salemt. Gate-variants of gated recurrent unit (gru) neural networks. In *2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS)*, pages 1597–1600. IEEE, 2017.

[56] Philip Doganis, Alex Alexandridis, Panagiotis Patrinos, and Haralambos Sarimveis. Time series sales forecasting for short shelf-life food products based on artificial neural networks and evolutionary computing. *Journal of Food Engineering*, 75(2):196–204, 2006.

[57] Norman R Draper and Harry Smith. *Applied regression analysis*, volume 326. John Wiley & Sons, 1998.

[58] Scott T Drummond, Kenneth A Sudduth, Anupam Joshi, Stuart J Birrell, and Newell R Kitchen. Statistical and neural methods for site–specific yield prediction. *Transactions of the ASAE*, 46(1):5, 2003.

[59] Germain Forestier, François Petitjean, Hoang Anh Dau, Geoffrey I Webb, and Eamonn Keogh. Generating synthetic time series to augment sparse datasets. In *2017 IEEE international conference on data mining (ICDM)*, pages 865–870. IEEE, 2017.

[60] RA Fox. Principles and procedures of statistics with special reference to the biological sciences, 1961.

[61] Rui Fu, Zuo Zhang, and Li Li. Using lstm and gru neural network methods for traffic flow prediction. In *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, pages 324–328. IEEE, 2016.

[62] David S Fung. Methods for the estimation of missing values in time series. 2006.

[63] Everette S Gardner Jr. Exponential smoothing: The state of the art. *Journal of Forecasting*, 4(1):1–28, 1985.

[64] Everette S Gardner Jr. Exponential smoothing: The state of the art—part ii. *International Journal of Forecasting*, 22(4):637–666, 2006.

[65] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. 1999.

[66] Felix A Gers, Nicol N Schraudolph, and Jürgen Schmidhuber. Learning precise timing with lstm recurrent networks. *Journal of Machine Learning Research*, 3(Aug):115–143, 2002.

[67] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[68] Clive William John Granger and Michael John Morris. Time series modelling and interpretation. *Journal of the Royal Statistical Society: Series A (General)*, 139(2):246–257, 1976.

[69] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[70] Pradeep Hewage, Ardhendu Behera, Marcello Trovati, Ella Pereira, Morteza Ghahremani, Francesco Palmieri, and Yonghuai Liu. Temporal convolutional neural (tcn) network for an effective weather forecasting using time-series data from the local weather station. *Soft Computing*, 24(21):16453–16482, 2020.

[71] Keith W Hipel and A Ian McLeod. *Time series modelling of water resources and environmental systems*. Elsevier, 1994.

[72] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[73] Rune Höglund and Ralf Östermark. Modelling varmax-processes by extended sample autocorrelation and linear regression techniques. In *Proceedings of the Third Finnish-Soviet Symposium on Probability Theory and Mathematical Statistics, Turku, Finland, August 13–16, 1991*, pages 68–85. De Gruyter, 2020.

[74] Ying-Yi Hong, John Joel F Martinez, and Arnel C Fajardo. Day-ahead solar irradiation forecasting utilizing gramian angular field and convolutional long short-term memory. *IEEE Access*, 8:18741–18753, 2020.

[75] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417, 1933.

[76] Ming Hua and Jian Pei. Cleaning disguised missing data: a heuristic approach. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 950–958, 2007.

[77] J Stuart Hunter. The exponentially weighted moving average. *Journal of Quality Technology*, 18(4):203–210, 1986.

[78] Svend Hylleberg. *Modelling seasonality.* Oxford University Press, 1992.

[79] Rob Hyndman, Anne B Koehler, J Keith Ord, and Ralph D Snyder. *Forecasting with exponential smoothing: the state space approach.* Springer Science & Business Media, 2008.

[80] Rob J Hyndman and Anne B Koehler. Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4):679–688, 2006.

[81] Rob J Hyndman, Anne B Koehler, Ralph D Snyder, and Simone Grose. A state space framework for automatic forecasting using exponential smoothing methods. *International Journal of Forecasting*, 18(3):439–454, 2002.

[82] Alan Julian Izenman. Linear discriminant analysis. In *Modern multivariate statistical techniques*, pages 237–280. Springer, 2013.

[83] Fatemeh Jafari, Lobna Nassar, and Fakhri Karray. Time series similarity analysis framework in fresh produce yield forecast domain. In *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 3599–3605, 2021. submitted to IEEE.

[84] Jitendra Kumar Jaiswal and Rita Samikannu. Application of random forest algorithm on feature subset selection and classification and regression. In *2017 World Congress on Computing and Communication Technologies (WCCCT)*, pages 65–68. IEEE, 2017.

[85] Zehui Jiang, Chao Liu, Nathan P Hendricks, Baskar Ganapathysubramanian, Dermot J Hayes, and Soumik Sarkar. Predicting county level corn yields using deep long short term memory models. *arXiv preprint arXiv:1805.12044*, 2018.

[86] Ian T Jolliffe. Principal components in regression analysis. In *Principal component analysis*, pages 129–155. Springer, 1986.

[87] Per Jonsson and Lars Eklundh. Seasonality extraction by function fitting to time-series of satellite sensor data. *IEEE transactions on Geoscience and Remote Sensing*, 40(8):1824–1832, 2002.

[88] WL Junger and A Ponce De Leon. Imputation of missing data in time series for air pollutants. *Atmospheric Environment*, 102:96–104, 2015.

[89] Fakhreddine Karray, Fakhreddine O Karray, and Clarence W De Silva. *Soft computing and intelligent systems design: theory, tools, and applications*. Pearson Education, 2004.

[90] Monisha Kaul, Robert L Hill, and Charles Walthall. Artificial neural networks for corn and soybean yield prediction. *Agricultural Systems*, 85(1):1–18, 2005.

[91] Eamonn J Keogh and Michael J Pazzani. Scaling up dynamic time warping for datamining applications. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 285–289, 2000.

[92] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[93] Nikolaos Kourentzes and Fotios Petropoulos. Forecasting with multivariate temporal aggregation: The case of promotional modelling. *International Journal of Production Economics*, 181:145–153, 2016.

[94] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing systems*, 25:1097–1105, 2012.

[95] Tarald O Kvålseth. Cautionary note about r 2. *The American Statistician*, 39(4):279–285, 1985.

[96] Pedro Lara-Benítez, Manuel Carranza-García, José M Luna-Romera, and José C Riquelme. Temporal convolutional networks applied to energy-related time series forecasting. *Applied Sciences*, 10(7):2322, 2020.

[97] Colin Lea, Michael D Flynn, Rene Vidal, Austin Reiter, and Gregory D Hager. Temporal convolutional networks for action segmentation and detection. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 156–165, 2017.

[98] Andy Liaw, Matthew Wiener, et al. Classification and regression by randomforest. *R news*, 2(3):18–22, 2002.

[99] Zachary C Lipton, John Berkowitz, and Charles Elkan. A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*, 2015.

[100] Zachary C Lipton, David Kale, and Randall Wetzel. Directly modeling missing data in sequences with rnns: Improved classification of clinical time series. In *Machine learning for healthcare conference*, pages 253–270. PMLR, 2016.

[101] Lon-Mu Liu, Siddhartha Bhattacharyya, Stanley L Sclove, Rong Chen, and William J Lattyak. Data mining on time series: an illustration using fast-food restaurant franchise data. *Computational Statistics & Data Analysis*, 37(4):455–476, 2001.

[102] M.Saad F.Karray K. Ponnambalam P. Agarwal L.Nassar, I.Okwucchi. Prediction of strawberry yield and farm price utilizing deep learning. *International Joint Conference on Neural Networks (IJCNN), paper in press*, 2020.

[103] Kezhi Z Mao. Orthogonal forward selection and backward elimination algorithms for feature subset selection. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(1):629–634, 2004.

[104] Mahesh L Maskey, Tapan B Pathak, and Surendra K Dara. Weather based strawberry yield forecasts at field scale using statistical and machine learning models. *Atmosphere*, 10(7):378, 2019.

[105] Edward W McLaughlin. *Produce Industry Procurement: Changing Preferences and Practices*. Charles H. Dyson School of Applied Economics and Management College of . . . , 2015.

[106] R Melnico. Crop profile for apples in california, 1999. " Center for Integrated Pest Management, North Carolina State University".

[107] Bjoern H Menze, B Michael Kelm, Ralf Masuch, Uwe Himmelreich, Peter Bachert, Wolfgang Petrich, and Fred A Hamprecht. A comparison of random forest and its gini importance with standard chemometric methods for the feature selection and classification of spectral data. *BMC Bioinformatics*, 10(1):1–16, 2009.

[108] Dunja Mladenić. Feature selection for dimensionality reduction. In *International Statistical and Optimization Perspectives Workshop" Subspace, Latent Structure and Feature Selection"*, pages 84–102. Springer, 2005.

[109] Jihoon Moon, Seungwon Jung, Jehyeok Rew, Seungmin Rho, and Eenjun Hwang. Combination of short-term load forecasting models based on a stacking ensemble approach. *Energy and Buildings*, 216:109921, 2020.

[110] Lobna Nassar, Muhammad Saad, Ifeanyi Emmanuel Okwuchi, Mohita Chaudhary, Fakhri Karray, and Kumaraswamy Ponnambalam. Imputation impact on strawberry yield and farm price prediction using deep learning. In *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 3599–3605. IEEE, 2020.

[111] Ifeanyi Okwuchi. Machine learning based models for fresh produce yield and price forecasting for strawberry fruit. Master's thesis, University of Waterloo, 2020.

[112] Ifeanyi Okwuchi, Lobna Nassar, Fakhri Karray, and Kumaraswamy Ponnambalam. Deep learning ensemble based model for time series forecasting across multiple applications. In *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 3077–3083. IEEE, 2020.

[113] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[114] Aaron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. Conditional image generation with pixelcnn decoders. *arXiv preprint arXiv:1606.05328*, 2016.

[115] David Opitz and Richard Maclin. Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence research*, 11:169–198, 1999.

[116] The pandas development team. pandas-dev/pandas: Pandas, February 2020.

[117] Xanthoula Eirini Pantazi, Dimitrios Moshou, Thomas Alexandridis, Rebecca L Whetton, and Abdul Mounem Mouazen. Wheat yield prediction using machine learning and advanced sensing techniques. *Computers and Electronics in Agriculture*, 121:57–65, 2016.

[118] Tapan B Pathak, Surendra K Dara, and Andre Biscaro. Evaluating correlations and development of meteorology based yield forecasting model for strawberry. *Advances in Meteorology*, 2016, 2016.

[119] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[120] Yung-Hsing Peng, Chin-Shun Hsu, and Po-Chuang Huang. Developing crop price forecasting service using open data from taiwan markets. In *2015 Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, pages 172–175. IEEE, 2015.

[121] James L Peugh and Craig K Enders. Missing data in educational research: A review of reporting practices and suggestions for improvement. *Review of Educational Research*, 74(4):525–556, 2004.

[122] R Polikar. Ensemble learning in ensemble machine learning: Methods and applications; zhang, c., ma, y., eds, 2012.

[123] A Provenzale, Leonard A Smith, R Vio, and G Murante. Distinguishing between low-dimensional dynamics and randomness in measured time series. *Physica D: Nonlinear Phenomena*, 58(1-4):31–49, 1992.

[124] Piotr Przymus, Youssef Hmamouche, Alain Casali, and Lotfi Lakhal. Improving multivariate time series forecasting with random walks with restarts on causality graphs. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 924–931. IEEE, 2017.

[125] Yao Qin, Dongjin Song, Haifeng Chen, Wei Cheng, Guofei Jiang, and Garrison Cottrell. A dual-stage attention-based recurrent neural network for time series prediction. *arXiv preprint arXiv:1704.02971*, 2017.

[126] Aistis Raudys, Vaidotas Lenčiauskas, and Edmundas Malčius. Moving averages for financial data smoothing. In *International Conference on Information and Software Technologies*, pages 34–45. Springer, 2013.

[127] Kenneth Rogoff. Traded goods consumption smoothing and the random walk behavior of the real exchange rate. *NBER Working Paper*, (w4119), 1992.

[128] Muhammad Saad, Mohita Chaudhary, Fakhri Karray, and Vincent Gaudet. Machine learning based approaches for imputation in time series data and their impact on forecasting. In *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2621–2627. IEEE, 2020.

[129] Mayu Sakurada and Takehisa Yairi. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, pages 4–11, 2014.

[130] Lawrence K Saul and Sam T Roweis. An introduction to locally linear embedding. *unpublished. Available at: http://www. cs. toronto. edu/˜ roweis/lle/publications. html*, 2000.

[131] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. 2016.

[132] Zhipeng Shen, Yuanming Zhang, Jiawei Lu, Jun Xu, and Gang Xiao. SeriesNet: A Generative Time Series Forecasting Model. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2018.

[133] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *arXiv preprint arXiv:1506.04214*, 2015.

[134] Da vid L Streiner. The case of the missing data: methods of dealing with dropouts and other research vagaries. *The Canadian Journal of Psychiatry*, 47(1):70–77, 2002.

[135] K Sutiene, Gytis Vilutis, and D Sandonavicius. Forecasting of grid job waiting time from imputed time series. *Elektronika ir Elektrotechnika*, 114(8):101–106, 2011.

[136] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *arXiv preprint arXiv:1409.3215*, 2014.

[137] Avraam Tsantekidis, Nikolaos Passalis, Anastasios Tefas, Juho Kanniainen, Moncef Gabbouj, and Alexandros Iosifidis. Forecasting stock prices from the limit order book using convolutional neural networks. In *2017 IEEE 19th Conference on Business Informatics (CBI)*, volume 1, pages 7–12. IEEE, 2017.

[138] P Tuppad, KR Douglas-Mankin, T Lee, R Srinivasan, and JG Arnold. Soil and water assessment tool (swat) hydrologic/water quality model: Extended capability and wider adoption. *Transactions of the ASABE*, 54(5):1677–1684, 2011.

[139] Peter Turchin and Andrew D Taylor. Complex dynamics in ecological time series. *Ecology*, 73(1):289–305, 1992.

[140] Aäron Van Den Oord, Sander Dieleman, and Benjamin Schrauwen. Deep content-based music recommendation. In *Neural Information Processing Systems Conference (NIPS 2013)*, volume 26. Neural Information Processing Systems Foundation (NIPS), 2013.

[141] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(11), 2008.

[142] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[143] Michael Wand, Jan Koutník, and Jürgen Schmidhuber. Lipreading with long short-term memory. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6115–6119. IEEE, 2016.

[144] Anna X. Wang, Caelin Tran, Nikhil Desai, David Lobell, and Stefano Ermon. Deep transfer learning for crop yield prediction with remote sensing data. In *Proceedings of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies*, COMPASS '18, New York, NY, USA, 2018. Association for Computing Machinery.

[145] Jiao Wang and Yanzhu Hu. An improved enhancement algorithm based on cnn applicable for weak contrast images. *IEEE Access*, 8:8459–8476, 2020.

[146] Yunbo Wang, Mingsheng Long, Jianmin Wang, Zhifeng Gao, and Philip S Yu. Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 879–888, 2017.

[147] Zhiguang Wang and Tim Oates. Imaging time-series to improve classification and imputation. *arXiv preprint arXiv:1506.00327*, 2015.

[148] Kilian Q Weinberger, Fei Sha, and Lawrence K Saul. Learning a kernel matrix for nonlinear dimensionality reduction. In *Proceedings of the twenty-first international conference on Machine learning*, page 106, 2004.

[149] Gail Weiss, Yoav Goldberg, and Eran Yahav. On the practical computational power of finite precision rnns for language recognition. *arXiv preprint arXiv:1805.04908*, 2018.

[150] Christopher KI Williams. On a connection between kernel pca and metric multidimensional scaling. *Machine Learning*, 46(1):11–19, 2002.

[151] JR Williams, JG Arnold, JR Kiniry, PW Gassman, and CH Green. History of model development at temple, texas. *Hydrological Sciences Journal*, 53(5):948–960, 2008.

[152] Cort J Willmott and Kenji Matsuura. Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. *Climate Research*, 30(1):79–82, 2005.

[153] David H Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992.

[154] Yanzhao Wu, Ling Liu, Juhyun Bae, Ka-Ho Chow, Arun Iyengar, Calton Pu, Wenqi Wei, Lei Yu, and Qi Zhang. Demystifying learning rate policies for high accuracy training of deep neural networks. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 1971–1980. IEEE, 2019.

[155] Ke Yan and David Zhang. Feature selection and analysis on correlated gas sensor data with recursive feature elimination. *Sensors and Actuators B: Chemical*, 212:353–363, 2015.

[156] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[157] Ameema Zainab, Ali Ghrayeb, Mahdi Houchati, Shady S Refaat, and Haitham Abu-Rub. Performance evaluation of tree-based models for big data load forecasting using randomized hyperparameter tuning. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 5332–5339. IEEE, 2020.

[158] Xin Zhang and Jiali You. A gated dilated causal convolution based encoder-decoder for network traffic forecasting. *IEEE Access*, 8:6087–6097, 2020.

[159] Yangsen Zhang, Jia Zheng, Yuru Jiang, Gaijuan Huang, and Ruoyu Chen. A text sentiment classification modeling method based on coordinated cnn-lstm-attention model. *Chinese Journal of Electronics*, 28(1):120–126, 2019.

[160] Zhi-Hua Zhou, Jianxin Wu, and Wei Tang. Ensembling neural networks: many could be better than all. *Artificial Intelligence*, 137(1-2):239–263, 2002.