RECOMMENDATION SYSTEM FOR HORTICULTURAL COMMODITIES

Mr. Lukman Adlin Harahap

A Dissertation Submitted in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy in Computer Science and Information Technology
Department of Mathematics and Computer Science
FACULTY OF SCIENCE
Chulalongkorn University
Academic Year 2020

ระบบการแนะนำสำหรับโภคภัณฑ์พืชสวน

นายลุคมัน อัดลิน ฮาราฮับ

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรดุษฎีบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2563

Thesis Title	RECOMMENDATION SYSTEM FOR HORTICULTURAL COMMODITIES
By	Mr. Lukman Adlin Harahap
Field of Study	Computer Science and Information Technology
Thesis Advisor	Associate Professor RAJALIDA LIPIKORN, Ph.D.

Accepted by the FACULTY OF SCIENCE, Chulalongkorn University in Partial Fulfillment of the Requirement for the Doctor of Philosophy

.... Dean of the FACULTY OF SCIENCE

(Professor POLKIT SANGVANICH, Ph.D.)

DISSERTATION COMMITTEE

.... Chairman

(Associate Professor PATTARASINEE BHATTARAKOSOL, Ph.D.)

.... Thesis Advisor

(Associate Professor RAJALIDA LIPIKORN, Ph.D.)

.... Examiner

(Associate Professor NAGUL COOHAROJANANONE, Ph.D.)

.... Examiner

(Assistant Professor SUKREE SINTHUPINYO, Ph.D.)

.... External Examiner

(Associate Professor Panjai Tantatsanawong, Ph.D.)

ลุคมัน อัดลิน ฮาราฮับ : ระบบการแนะนำสำหรับโภคภัณฑ์พืชสวน. (RECOMMENDATION SYSTEM FOR HORTICULTURAL COMMODITIES) อ.ที่ปรึกษาหลัก : รศ. ดร.รัชลิดา ลิปิกรณ์

โดยปกติแล้วโภคภัณฑ์พืชสวนจะมีราคาขึ้นลงที่ไม่แน่นอนอันเนื่องมาจากธรรมชาติของพืชสวน ฤดูกาลและภูมิอากาศเป็นปัจจัยหลักที่ทำให้ราคาโภคภัณฑ์พืชสวนไม่แน่นอน ความไม่เสถียรของราคาทำให้การวางแผนการเพาะปลูกพืชสวนทำได้ยาก เกษตรกรท้องถิ่นส่วนใหญ่สามารถวางแผนการเพาะปลูกได้โดยสัญชาติญาณที่มาจากประสบการณ์ แต่การวางแผนการเพาะปลูกไม่ใช่เรื่องง่ายสำหรับเกษตรกรมือใหม่หรือผู้ที่ไม่มีประสบการณ์ ดังนั้นระบบการแนะนำที่นำเสนอนี้จะสามารถช่วยเกษตรกรมือใหม่ในการจัดกำหนดการการเพาะปลูกพืชสวน ระบบที่นำเสนอนี้ประกอบไปด้วย 3 ส่วน ได้แก่ การทำนายราคา การแนะนำโภคภัณฑ์ และการจัดกำหนดการการเพาะปลูก ลูกผสมของแอลเอสทีเอ็มและเอสทีดี-แอลโออีเอสเอสถูกนำมาใช้ในแบบจำลองทำนายราคาที่สามารถทำนายราคาล่วงหน้าแบบหลายขั้นซึ่งมีความถูกต้องที่ยอมรับได้ ราคาที่ทำนายได้จะถูกนำมาใช้ร่วมกับช่วงเวลาที่เกษตรกรต้องการจะวางแผนการเพาะปลูกเพื่อแนะนำพืชสวนที่ควรจะปลูกในช่วงเวลานั้น และเมื่อเกษตรกรระบุทำเลที่ตั้งของสถานที่เพาะปลูก ระบบจะนำเสนอกำหนดการการเพาะปลูกพืชสวนที่เหมาะสมตั้งแต่เริ่มการเพาะปลูกไปจนถึงช่วงเวลาเก็บเกี่ยวพืชสวน

สาขาวิชา	วิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ	ลายมือชื่อนิสิต ....
ปีการศึกษา	2563	ลายมือชื่อ อ.ที่ปรึกษาหลัก ....

5973101023 : MAJOR COMPUTER SCIENCE AND INFORMATION TECHNOLOGY
KEYWORD: Horticultural commodities, Multistep price prediction, Long Short-Term Memory Neural Network, Recommendation System, Seasonal and Trend Decompostion, Cultivation scheduling

Lukman Adlin Harahap : RECOMMENDATION SYSTEM FOR HORTICULTURAL COMMODITIES. Advisor: Assoc. Prof. RAJALIDA LIPIKORN, Ph.D.

Horticultural commodities commonly have fluctuating prices due to their nature. Seasonality and climate are the main factors that cause their prices to fluctuate. Price instability causes a planning on horticultural cultivation to become difficult. Local farmers would intuitively know the planning by their experience, but this might be too complicated for those farmers who are new or have no experience. Thus, the proposed recommendation system would be able to help the new farmers to set a schedule for horticultural cultivation. The proposed recommendation system consists of three phases: price prediction, commodity recommendation, and cultivation scheduling. A hybrid of Long Short-Term Memory Neural Network (LSTM) with Seasonal and Trend Decomposition based on LOESS (STD-LOESS) are used as price prediction model. The proposed model can provide multistep price prediction with acceptable accuracy. The predicted prices are used with the preferred cultivation period to find the most suitable commodities to be cultivated for that period. Finally, the cultivation schedule with the best starting time and harvesting time for suitable commodities based on seasonality, price, cultivation location and production index is returned as the result, thus farmers would be able to decide when to start the cultivation and when to harvest the commodities.

Field of Study:	Computer Science and Information Technology	Student's Signature ....
Academic Year:	2020	Advisor's Signature ....

ACKNOWLEDGEMENTS

My utmost respect and gratitude to my advisor Associate Professor Rajalida Lipikorn, for her unceasing support and advices that kept me going until my research was finally completed and written into this dissertation. I would also like to thank the commitee : Associate Professor Pattarasinee Bhattarakosol, Associate Professor Nagul Cooharojananone, Associate Professor Panjai Tantatsanawong and Assistant Professor Sukree Sinthupinyo for their suggestion and guidance on the writing of my dissertation that have been very helpful and valuable.

Finally, I would like to thank Indonesia Endowment Fund For Education (LPDP) for providing scholarship on my Ph.D study, Thai Meteorological Department, Department of Internal Trade of Thailand, and Office of Agricultural Economics of Thailand for providing data required in my research.

Lukman Adlin Harahap

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Page

Chapter 1

Introduction

Rationale

Horticultural products are agricultural products such as flowers, vegetables, and fruits. Agricultural activities take time to proceed thus it is necessary to have a foresight on the prices of the products ahead of time. Prices of agricultural products are commonly fluctuating over time, thus planning for cultivation of the products is hard to do. The experienced farmers can plan by intuition alone without having to rely on any foresight of future prices of the products, but this is hard for new farmers. Moreover, climate change can also affect the prices of horticultural products to some degrees, this adds the complexity to decision making in horticultural product cultivation. Thus, a reliable prediction system for horticultural product prices would help in this matter since prediction with small error will help in decision making regarding commodity cultivation or commodity selection to maximize the profit of the farmers.

Problem Formulation

1. What is the most suitable method to predict the future prices of agricultural products based on climate parameters and prices of horticultural products?
2. How to select agricultural products to cultivate in a specified period of time based on the predicted prices?
3. How can we schedule the commodity cultivation according to the specified period of time?

Objectives

To develop the price prediction system for horticultural products and the cultivation scheduling system based on the specified time period, cultivation period of each product and the predicted price of each product.

Scope

1 Horticultural product data that are used in the prediction consist of prices of the products in a period of year 2008 to 2018.

2 Climate data that are used in the prediction consist of temperature, rainfall, wind speed, humidity, and day light duration in a period of year 2008 to 2018.

3 The data of horticultural product prices are limited to the prices of horticultural products in Bangkok Province.

Research Methodology

1 Reviewing literatures related to horticulture products, theories and research on price prediction methods and agricultural scheduling for horticultural products.

2 Formulating problems regarding the price prediction system and cultivation scheduling for horticultural products.

3 Gathering horticultural product prices and climate data from agricultural areas of Thailand from Department of Internal Trade, Office of Agricultural Economics and Department of Meteorology of Thailand. All data are expected to contain information of the location, and to be time series data in the period of five to ten years.

4 Developing a price prediction system which can predict the prices of horticultural products with acceptable accuracy.

5 Developing a horticultural product recommendation and cultivation scheduling system based on predicted prices and a specified time period such that the farmers will gain the optimal profit.

6 Summarizing the results and writing dissertation.

Expected Outcomes

Obtain a system that can predict the prices of horticultural products in an acceptable accuracy and a recommendation system that can suggest which commodities to be cultivated and a cultivation schedule for horticultural products.

Chapter 2

Literature Review

Price Prediction in Agriculture

Several price prediction methods have been applied to agricultural products. They can be categorized as one of these categories which are time series mathematical modeling, machine learning, and mixed method.

For time series mathematical modeling methods, the auto-regressive integrated moving average model (ARIMA) was used to predict cotton prices in India[1], in this research data from AGMARKNET website for a period of 11 years were used to forecast the prices, however there is no evaluation of model performance despite the claim that it has been successfully used to forecast future monthly price, whereas the generalized autoregressive conditional heteroscedastic (GARCH) and exponential GARCH (EGARCH) was used to predict three prices: domestic and international edible oil price indices, and international cotton price 'Cotlook A' index [2] which is the international prices of raw cotton, gathered from commodity price bulletin released by United Nations Convention of Trade and Development (UNCTAD). The data consisted of 360 data points over the period of April 1982 to March 2012 where 346 data points were used for modelling and the remaining 14 points were used for prediction. It was concluded that EGARCH was more effective in forecasting than ARIMA and GARCH; furthermore, it is applicable as well for other commodity price forecasting. Another traditional method such as multiple linear regression model was also used to predict the price of Chinese corn based on big data [3], data of corn from United States Department of Agriculture (USDA) and International Monetary Fund (IMF) China were used. It was assumed that the prices of corn depended on supply and demand of corn and the prediction yielded the coefficient of determination (R^2) of 0.939. Another research further explored the performance of variants of GARCH models in assessing volatility of agricultural product prices in Indonesian spot market, it was mentioned that there are five models of GARCH, which are ARCH, GARCH, GARCH-M, TGARCH and EGARCH [4]. As a conclusion, EGARCH is the best model to assess volatility in almost all the commodities except CPO, which is best assessed using TGARCH. Similar work called Realized GARCH were also conducted for other commodities, which has shown better performance in comparison with EGARCH [5]. There is also a case where ARIMA and GARCH were used together in assessing price fluctuation risk of harvested dry grain in Pemalang, Indonesia, the combination of these models can be used as a risk measurement on longer investment time for farmers [6].

With the advance of computing power, several machine learning methods have been used for prediction, from the simplest artificial neural network (ANN) such as the use of backpropagation neural network (BPNN) to predict vegetable prices in India [7]. In the research, three years of monthly and weekly data of tomato prices in India from 2009 to 2011 were used to forecast the prices of next week and next month, BPNN model was configured to have 5 and 4 hidden nodes for monthly and weekly models, optimization target was set to 0.001. The model returned absolute errors of 10% and 25% for monthly and weekly price forecast, respectively. The use of least square support vector regression (LSSVR) to predict prices of vegetable products in China [8] was proposed based on SVR, the model transformed complex optimization problems to easily solvable linear equations. It was used to forecast the prices of fruit in the period of January 2013 to December 2015, using data from the period of January 2003 to December 2012. Beside fruits price data, sunshine duration data from October 2002 to December 2015 were also used, the model used radial basis function as kernel with gamma value of 1.2, penalty coefficient of 1.8, and error tolerance value of 0.001. As a result, the model could perform well with the average error (MAE), the normalized mean square error (NMSE) and the root mean square error (RMSE) of 0.1062, 0.1624, 0.1318, consecutively. Other models such as long short term memory (LSTM) model was used to predict market prices of fresh produces in Taiwan [9]. The performance of conventional models such as ANN, ARIMA, SVM and gradient boosting (GB) were compared to that of deep learning models such as LSTM, Convolutional Neural Network (CNN)-LSTM, and the addition of attention (ATT) to CNN-LSTM, the results in forecasting several commodities of horticulture, such as cabbage, cauliflower, bokchoi, watermelon and strawberry, showed that the integration of ATT-CNN-LSTM generally outperformed other models. However, the forecast was only limited to 20-day steps ahead of prediction and other affecting factors beside the prices were not considered as input to the model. BPNN is a general machine learning method that has been used as the baseline model for comparison to newer models' performance. The use of BPNN in agricultural price prediction has been used in a wide range, aside from the aforementioned, it was used to predict Shiitake mushroom (*Lentinus edodes*) price, the results showed that it had good performance on simulation, but it performed poorly on prediction [10]. In another work, BPNN was regarded as superior to Extreme Learning Machine (ELM) and SVR in performance [11], it was considered as the optimal single model based on its performance; however, the main focus of this work was on a model selection framework for agricultural product price prediction rather than the performance of the mentioned model. BPNN was further developed to attain better generalization, one of the efforts is to use more hidden layers instead of singular layer, such configuration is called Deep Neural Network (DNN); however, the simple architecture may not used as many layers as the recent models, such as Recurrent Neural Network (RNN), LSTM and CNN. The simple DNN has been applied to stock price predictions in many literatures, one of them was a stock trading system based on DNN, which claimed to have 70.83% profitable trades with profit factor of 18.67 [12]. The performance of DNN was supported by another paper that used DNN for stock price prediction [13], later on LSTM and CNN have been found to have overwhelming capabilities on time series prediction including price prediction. Thus, DNN should be a viable method for agricultural product price prediction which has similar volatility nature to that of the stock prices. As previously mentioned, LSTM is another potential method for agricultural price prediction, a recent research used LSTM to predict tomato price in India with observation lag size of 12 and the prediction of daily price have an accuracy of 96.43%. Unfortunately, the authors did not compare the performance to other models [14]. Another research on Indian local regulated market, called Mandi, used LSTM to predict the prices of onion and potato for 30-day period; however, LSTM performance was not as good as that of the multi variate model. Thus it was ruled out from detecting price anomalies in Mandis [15]. However, the performance of LSTM was compared to mathematical model such as ARIMA and the results showed that LSTM can perform time series forecast better by reducing error rates up to 84-87 percent [16]. Another research compared SVR to ARIMA in prediction performance on Thailand’s agricultural commodities such as cassava, sugarcane and coffee, it was shown that SVR couldn’t surpass performance of ARIMA by itself, but when it was combined with Maximal Overlap Discrete Wavelet Transform (MODWT), it outperformed ARIMA in predicting the prices of all the commodities [17]. Most of these machine learning models outperform prediction results of time series mathematical models, in most scenarios.

For mixed models, most of them are a mix of multiple models with either a time series mathematical model or a machine learning model. The recent research predicts prices of grape using a combination of multiple linear regression (MLR) and SVR with empirical model decomposition (EMD) method [18]. The prices of grape from January 2011 to December 2018 were used to predict prices of July 2011 to December 2018, the results revealed that the model outperformed other models; however, no other factors were considered as input except the prices of grape. Quantile regression (QR) and radial basis function neural network (RBFNN) were combined to predict the price of soybean in China [19]. The prices of soybean from May 2015 to December 2015 were predicted using several economic parameters such as global and local soybean productivities, import volumes, local demand of soybean, consumer price index, consumer confidence index, money supply, and imported soybean prices from January 2010 to April 2015. The model was able to forecast with MAPE of 1.11, but it was stated that certain variables were more important than others depended on the quantile of price levels, thus the use of economic variables could not be considered as general. A combination of seasonal trend decomposition using LOESS (STL) and extreme machine learning (ELM) were used to forecast vegetable prices in China [20], in this research, price of cabbage, pepper, cucumber, and green bean. The tomato prices of March 2010 to April 2014 were predicted using the model trained with prices from the period of January 2002 to February 2010 which was basically analogue to [18]. The prices were decomposed using STL and then the price components were fed into ELM model, as a result, STL-ELM model generally outperformed predictions for all of the commodities when compared to other models such as time delay neural network (TDNN), singular ELM, SVR, seasonal auto-regressive integrated moving average (SARIMA) and SARIMA with Kalman filter (KF). There are also a comparison between a combination of ARIMA-GARCH and ARIMA-ANN in predicting potato prices in India [21], the models were trained using data from period of January 2005 to May 2016 to predict monthly prices of June 2016 to May 2017. The prices were decomposed using the method proposed by Zhang [22] where prices were decomposed into linear and nonlinear elements. The results showed that ARIMA-ANN model outperformed other models, such as ARIMA, GARCH and ARIMA-GARCH models. Price Estimation for Crops using the Application of Deep Learning (PECAD), which relied on a mix of DNN, Generalized Linear Models (GLM), and Temporal Convolutional Network (TCN) was proposed where DNN was treated as deep component and GLM was treated as a wide component while TCN output was used as input to GLM. The PECAD is claimed to outperform other standard models, including TCN and LSTM [23]. Other research used a compound of several deep learning models such as LSTM, CNN, GRU, and attention model to predict strawberry yield and farm price, and the results show that the compound of LSTM and CNN with attention model outperforms all other deep learning model compounds [24].

There is another research that combined ARIMA and SVR to forecast short time price of garlic and the results reveal that a combination of ARIMA and SVR outperforms the prediction of respective singular models [25]. The recent review paper shows that the trend for agricultural price forecasting method is the mixed model, it is suggested that the application of the model should also incorporate price influencing factors, while seasonal adjustment model can help with difficult seasonal forecasting task, and finally a hybrid optimization algorithm is expected to improve prediction performance [26].

Recommendation system can be considered as a decision support system in agriculture. The sole purpose of a decision support system (DSS) is to assist decision making in many different fields, thus a DSS can come in any form according to the given function including a recommendation system. In agriculture, a DSS can provide services for many aspects of agriculture, for example, a DSS for irrigation scheduling that can be configured and used in many different locations and conditions without rebuilding a new DSS [26]. The DSS can provide helpful information for practical use in agricultural water utilization and allocation. Moreover, harvest scheduling could also be solved using a DSS as a DSS was used to optimize sugarcane cultivation for commercial growers in Africa by mapping the harvesting time problem onto the traveling salesmen problem [27]. The system was proven to be better than traditional random harvesting time over a period of 2 years. Another research in Europe attempted to build a Systems Dynamics (SD) model to help in decision making process [27]. The proposed SD model was able to provide simulations of different policies and the system was simple enough to be used by a wide range of users. Additionally, the system was capable of assessing strategic changes in sugar beet processing. Decision making in agricultural activities was largely done by experience, thus a DSS could support a decision making that is clear on what relevant information was taken into consideration, and also a retractable and simple decision making process[28]. Another paper also suggested that the use of weather and climate information in agricultural decision making was underutilized [29] due to low accuracy of weather forecast, and there were greater concerns on non-weather factors such as market volatility and regulation changes. In this research, a recommendation system was developed upon the principles of DSS.

Chapter 3

Theoretical Background

Data Preprocessing Methods

Data can exist in a wide range of values for different parameters, disparity between these values can cause a problem in training a neural network. The asymmetry of data also could affect the training outcome of a neural network, it can be in a form of missing data and outliers. These problems can be tackled by preprocessing the data before using them to train neural networks.

According to Han et al., [30], data preprocessing generally undergoes these steps; data cleaning, data integration, data reduction, and data transformation and discretization. In data cleaning, missing data and outlier will be handled. In data integration, since data from multiple sources will be used together in training neural network, data integration will be performed in order to make sure there is no data redundancy. Data reduction can help if data have dimensionality problem, and there are numerous methods to perform data reduction, such as principal component analysis (PCA) and wavelet transform. Then the last step is data transformation which transforms data of different parameters with different ranges and different units to have values that are in the same range and also discretize data to meet the need of a model used in training process. The transformation can be Min-max, z-Score or decimal normalization.

Emerging data preprocessing techniques that have been used in conjunction with neural network especially in time series prediction models are data decomposition techniques. These techniques will decompose time series data into parts, one of the techniques that was used in this research is seasonal trend decomposition using LOESS (STL)

STL is a decomposition technique developed by Cleveland in 1990[31]. Basically, STL will decompose time series data into seasonal, trend and residue component. The decomposition method consists of the iteration of two loops, namely inner loop and outer loop. The inner loop undergoes six steps, which are detrending, seasonal smoothing, low pass filtering of the smoothed seasonality, detrending of the smoothed seasonality, de-seasonalizing and the last step is trend smoothing. Outer loop is basically a subtraction of trend and seasonal components from the original time series, resulted in residual. If there are large values within residual component, these values will be treated as outliers and will be given weight so that they would be reduced in the next iteration. Thus, after the iteration has completed it will result in three components as aforementioned. In mathematical terms, the relationship of the components to the original time series is illustrated in Eq. (1) where X_t is the original time series, S_t is seasonal component, *Tt* is trend component and *Rt* is residual component at time t. An example of decomposed time series can be observed in Figure 1 which displays decomposed radish price time series data in Jin et al.'s price prediction using combination of STL and LSTM [32].

$$X_t = S_t + T_t + R_t \qquad \dots (1)$$

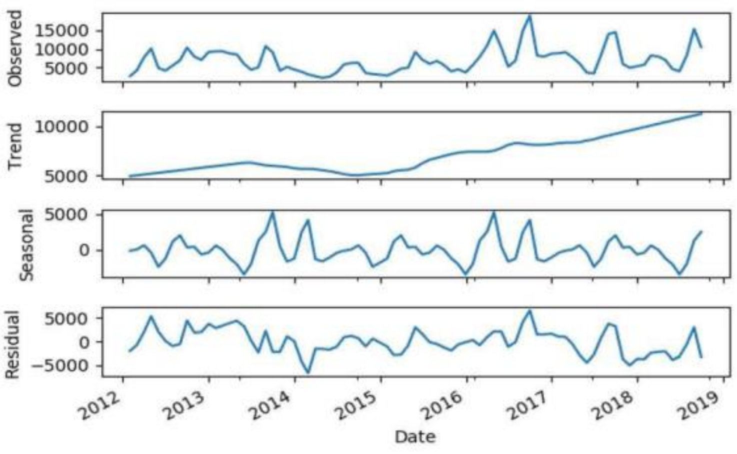


Figure 1.The decomposed radish price using STL

Source: [32]

Machine Learning Techniques

The rise of computing power has given rise of machine learning as well, various techniques were employed to perform prediction in various fields. In this research, several machine learning techniques were employed to predict horticultural product prices, namely SVR, BPNN, DNN, and LSTM.

Support Vector Machine

Support vector regression (SVR) was proposed the first time by Drucker et al., in 1997[33]. Support vector machine (SVM) is considered as a good tool for classification task and SVR is a modification of SVM to tackle regression task. Both SVM and SVR map variables to a high dimensional space using separation planes which are called kernels. Boser et al. stated that in SVR, data are mapped onto a high dimensional space and the regression is operated in the space [34]. In general, SVR is represented in Eq. (2) to predict the value of y_p, given a vector $x^{(p)}$ where (α_i^*, α_i) are Lagrange multipliers, v_i^t is the t-th support vector which corresponds to non-zero α_i^*or α_i , N is the number of Lagrange multiplier pairs which denotes the number of support vectors, and b is a threshold value [35]. Additionally, there is a maximum of *2N + 1* pairs of α_i^*, α_i and b values as stated by Drucker, et al [33].

$$y_p = \sum_{i=1}^{N} (\alpha_i^* - \alpha_i)(v_i^t x^{(p)} + 1)^p + b \qquad \ldots (2)$$

The performance of SVR technique relies on the tuning of hyperparameter values which are the cost of error (*C*), the width of loss function (ε), and selected kernel, as stated by Chu et al. [18]. There are three kernels that are commonly used, namely linear, polynomial, and radial basis function (RBF) kernels.

The value of C represents the degree of error and sometimes, it is also called regularization parameter. In concept, the value of C is used to determine how much deviation will be tolerated from the value of ε, and it is simple to say that the larger value of C will allow less error, while the smaller value will allow more error[36]. In practice, this is not always the case, thus the fine-tuning of C value has to be performed to improve the performance of SVM or SVR as C value was optimized by using teacher-learning based optimization by Das and Padhy [37].

The value of ε determines the width of the ε-insensitive zone. The larger value of ε will allow the use of more support vectors, and vice versa. Thus, ε value is needed to be fine-tuned according to the complexity nature of the problem[38]. Logically, more complex problems will require larger value of ε, but again, this has not always been the case, thus it is needed to be fine-tuned.

The most common and basic kernels in SVM and SVR are linear, polynomial, and RBF kernel. Kernel is used to transform input into a higher dimensional space. Generally, a kernel works as illustrated in Eq. (3), where x_1 and x_2 represent the input to be transformed, $K(x_1, x_2)$ is a kernel used to transform the input, and $\Phi(x_1)$ and $\Phi(x_2)$ represent the transformed input. The mathematical representations of kernels are shown in Table 1, where x and x_i represent the vectors to be transformed, T denotes a transpose operation, d is the degree of polynomial kernel and γ is the RBF kernel function parameter [39].

$$\langle x_1 \cdot x_2 \rangle \leftarrow K(x_1, x_2) = \langle \Phi(x_1) \cdot \Phi(x_2) \rangle \qquad \ldots (3)$$

Table 1. SVM Kernels

Kernel	Mathematical Representation
Linear	$K(x, x_i) = x \cdot x^T$
Polynomial	$K(x, x_i) = \left(1 + x \cdot x_i^T\right)^d$
RBF	$K(x, x_i) = e^{-\gamma \lVert x - x_i \rVert^2}$

Backpropagation Neural Network

Backpropagation neural network (BPNN) was first proposed by Rumelhart et al. in the year 1986 [40], it was meant to be the solution for multilayer perceptron training in feed forward neural network (NN). As other models of NN, BPNN also has similar problems to be considered; initial values, overfitting, input scaling, network structure, and the existence of multiple minima [41]. However, BPNN is still being used as a benchmark to other developing models.

The concept of BPNN is to adjust the weights based on the errors of predicted results from the desired values. The input of forward pass can be expressed as Eq. (4) which illustrates the relationship between total input of x_j connected to node j, output y_i and a vector of weights w_{ji}. The output y_j, which is a real value, can be calculated by Eq. (5). The aim is to minimize the total error E, which is defined as Eq. (6) where c is an index over input and output pairs, j is an index of output node, y_{jc} is the actual value of pairs c and output j, and d is the desired state.

$$x_j = \sum_i y_i w_{ji} \qquad \ldots (4)$$

$$y_j = \frac{1}{1+e^{-x_j}} \qquad \ldots (5)$$

$$E = \frac{1}{2}\sum_c \sum_j \left(y_{j,c-} d_{j,c}\right)^2 \qquad \ldots (6)$$

The backward pass is a bit more complicated than forward pass, it starts with the calculation of $\partial E/\partial y$ of a node in a layer given the value of the node before it, the value itself can be calculated using Eq.(7) where y_i is the output value of node i, x_j is the value of input of node j, and w_{ij} is the weight assigned for the connection from node j to node i. The value of $\partial E/\partial w$ is also calculated using Eq.(8). This process is performed recursively for all nodes that exist in layers. The next step is to adjust the weights, a simple gradient descent is usually used to change the weight value proportional to total $\partial E/\partial w$, as in Eq.(9), the value of t is increased by 1 every sweep of calculation through the nodes. These steps are performed repeatedly until the desired accuracy is attained.

$$\partial E/_{\partial y_i} = \sum_j \partial E/_{\partial x_j} \cdot w_{ji} \qquad \ldots (7)$$

$$\partial E/_{\partial w_{ji}} = \sum_j \partial E/_{\partial x_j} \cdot y_i \qquad \ldots (8)$$

$$\Delta w = -\varepsilon\, \partial E/_{\partial w}\,(t) + \alpha \Delta w(t-1) \qquad \ldots (9)$$

Hastie et al. stated some problems that BPNN have [41]. The most common problems are the number of hidden nodes and the existence of multiple minima. Hence the general approach to tackle the problems is through experimentation by either trying out different numbers of hidden nodes or experimenting with the number of iterations for weight adjustments, also known as epochs.

BPNN can be graphically illustrated as a network containing an input layer, a hidden layer, and an output layer. Depending on the task, it could have different sets of configurations, this is usually known as a neural network architecture. Common illustration of simple BPNN architecture is shown in Figure 2 where a connection between nodes represents the weight assigned to each pair.

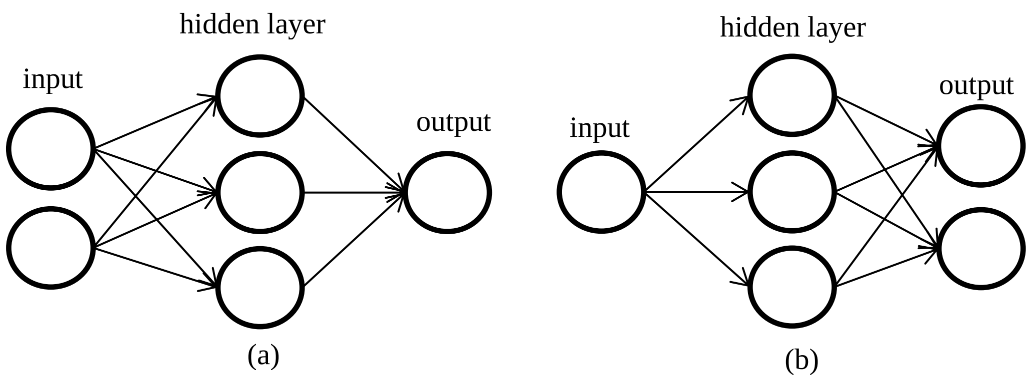


Figure 2 (a) A BPNN architecture with: (a) 2 input nodes, 3 hidden-layer nodes and one output node (b) one input node, 3 hidden-layer nodes and 2 output nodes

Choice of activation function and learning algorithm can also affect the performance of BPNN in terms of convergence speed. Sometimes, learning algorithm is also called an optimizer. One of the most commonly known learning algorithms is stochastic gradient descent (SGD), which has been used in BPNN since 1993 [42]. SGD and its variations have been widely used for training NN models, such as adaptive gradient (Adagrad)[43], root mean square propagation (RMSprop)[44], adaptive learning rate (Adadelta)[45], and also adaptive moment estimation (Adam)[46]. Jindal et al. suggested the use of Adam because it has better performance than RMSprop and Adadelta [47].

An activation function determines the output of a node. Some of basic activation functions are step or threshold function, piecewise linear function, logistic sigmoid function, land hyperbolic tangent sigmoid function[48]. These activation functions can be expressed by mathematical expressions as shown in Table 2.

Table 2. Basic activation functions

Activation function	Mathematical expression
Step or Threshold function	$\varphi(v) = \begin{cases} 1, v \geq 0 \\ 0, v < 0 \end{cases}$

Activation function	Mathematical expression
Piecewise linear function	$\varphi(v) = \begin{cases} 1, v \geq 1 \\ v, 0 < v < 1 \\ 0, v \leq 0 \end{cases}$
Sigmoid function (logistic)	$\varphi(v) = \dfrac{1}{1+e^{-v}}$
Sigmoid function (hyperbolic tangent)	$\varphi(v) = \tanh(v)$

Deep Neural Network

Deep neural network (DNN) is a brainchild of deep learning, a term introduced by Rina Dechter in 1986[49]. Back then, the term had not been associated closely to NN until Cybenko gave a mathematical proof of its use in neural network terms, stating that a continuous function can be approximated by a neural network with continuous sigmoidal nonlinearity[50]. The concept got into practice when Hornik proposed a multilayer feedforward network that was proven to be a universal estimator when emphasizing on the number of hidden nodes. He further remarked that different activation functions may also perform differently [51]. Since then, DNN has been used for approximation in many fields of study from general science to economics.

A DNN can be simply illustrated as a BPNN with more layers as it was stated by Chung et al.[52], though it may not always been the case. Lu et al. explored the interchangeability of deep networks and shallow networks, as the conclusion, it is possible but with the cost of substantial larger number of hidden nodes in shallow networks [53]. Bengio suggested that a deeper network may not yield better performance than that of a shallow one since training of deep network has always been difficult, many approaches have been introduced to tackle the problems [54].

DNN is analogous to BPNN, except there is a multiple layer of hidden nodes, a DNN model, θ, can be characterized by Eq. (10) where W is the weight matrix of each layer thus W = $\{W^{(1)}, W^{(2)}, \ldots, W^{(L)}\}$, L is the number of hidden layers, b is the bias vector of each layer, thus b = $\left\{b^{(1)}, b^{(2)}, \dots, b^{(L)}\right\}$, and $\sigma(x)$ is the nonlinear activation function. A DNN model uses a similar activation function as BPNN, and so does the learning algorithm.

$$\theta = \{W, b, \sigma(x)\} \quad \dots (10)$$

An input vector x_t is transformed through hidden layers by applying the following transformations in Eq. (11) – Eq. (13) where $N^{(l)}$, $W^{(l)}$ and $b^{(l)}$ are the numbers of hidden nodes, the weight matrix and the bias vector of layer l, consecutively and y_i is the output value of the i-th hidden node of the same layer, the value of y_i is calculated from the value of y_i from the previous layer that is converted to z_i using the activation function in Eq.(13). Softmax is used to determine the probability of class s for input vector of x_t using Eq. (14) for the final layer output which can be converted into any kind of output value using a function that corresponds to a specific problem, such as linear function which is generally used for regression problem since the output needs to be represented in a range of [-∞,∞] [55].

$$z^0 = x_t \quad \dots (11)$$

$$y_i^{(l+1)} = \sum_{j=1}^{N(l)} W_{ij}^{(l)} z_j^{(l)} + b_i^{(l)} \quad \dots (12)$$

$$z_i^{(l+1)} = \sigma\left(y_i^{(l+1)}\right) \quad \dots (13)$$

$$p(s|x_t) = softmax(x_t) = \frac{exp\left(w_s\ y^L\right)}{\sum_{n=1}^{N(L)}\left(w_n\ y^{(L)}\right)} \quad \dots (14)$$

A DNN can be fine-tuned just the same way a BPNN with additional part of network depth. The configuration of network size depends on each problem since some considerations are needed to design a DNN model for a particular problem, such as the nature of data, the existence of biases, and nonlinearities. A special care is needed when designing the size of the network since larger network can lead to overfitting despite the regularization and smaller network might perform inadequately[56]. Graphical representation of a common simple DNN can be seen in Figure 3.

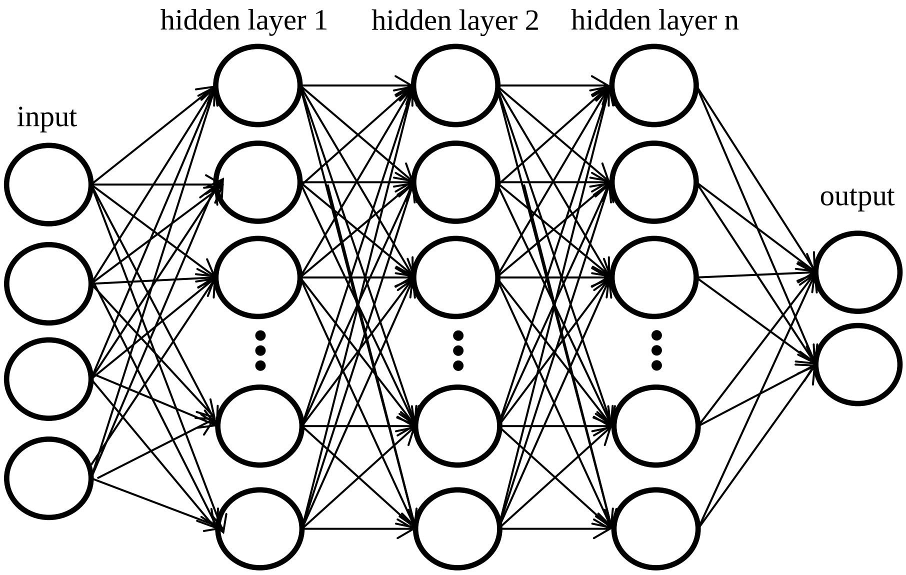


Figure 3. A common simple DNN architecture

Long Short-Term Memory Neural Network

Long-short term memory neural network (LSTMNN or LSTM) was proposed to solve some problems of recurrent neural network (RNN) which was derived from Rumelhart's work in 1986. [41]. A simple RNN is basically a BPNN where the output from previous time step is used as input to the current one. The simplest and common graphical representation of RNN is shown in Figure 4. RNN has a remarkable performance on various tasks regarding sequential data, from time series regression to DNA sequencing[57]. RNN has a problem of vanishing gradient which causes the RNN training hard to converge and a problem of exploding gradient. LSTM came as a solution to the convergence problem and gradient clipping came as a solution to solve the exploding gradient problem [58].

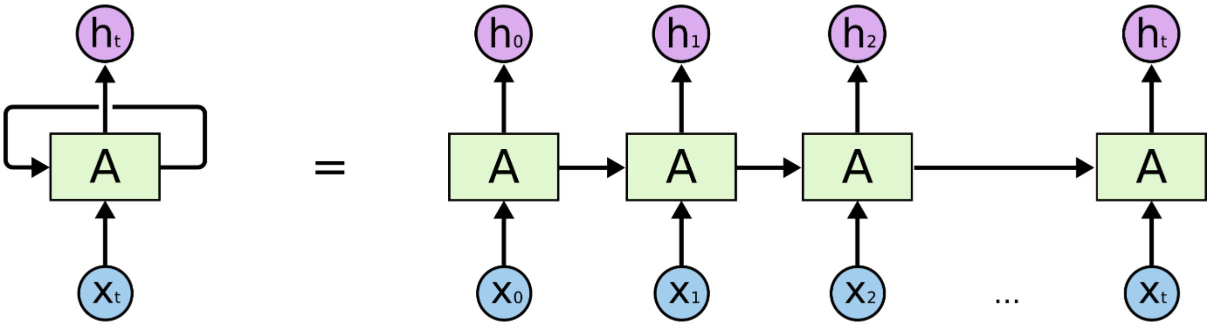


Figure 4. An unrolled RNN
Source: [59]

LSTM introduced the use of three gates; input, output and forget gates which are used to modify a memory cell state by resetting, writing or reading from it over time. The mathematical representations of an LSTM are as follows:

$$f_t = \sigma(W_{fh}h_{t-1} + W_{fx}x_t + b_f) \qquad \dots (15)$$

$$i_t = \sigma(W_{ih}h_{t-1} + W_{ix}x_t + b_i) \qquad \dots (16)$$

$$o_t = \sigma(W_{oh}h_{t-1} + W_{ox}x_t + b_o) \qquad \dots (17)$$

$$\check{c}_t = tanh(W_{ch}h_{t-1} + W_{cx}x_t + b_c) \qquad \dots (18)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \check{c}_t \qquad \dots (19)$$

$$h_t = o_t \odot tanh(c_t) \qquad \dots (20)$$

At time *t*, a memory cell c_t will be written, reset or read based on the signals in input gate i_t, forget gate f_t, and output gate o_t. These gates are actually similar to those of RNN but using sigmoid activation function instead of hyperbolic tangent (tanh) activation function, thus, it will only give value of 0 or 1 to signal the activity on the gate whether it is fully off (0) or fully on (1). An updated value $\check{c}_t$ will be combined with previous memory time step memory c_{t-1} state value based on f_t and i_t, and to be finally applied as a new hidden state h_t, along with weighting elements *W*[58]. A graphical representation of a simple LSTM is shown in Figure 5.

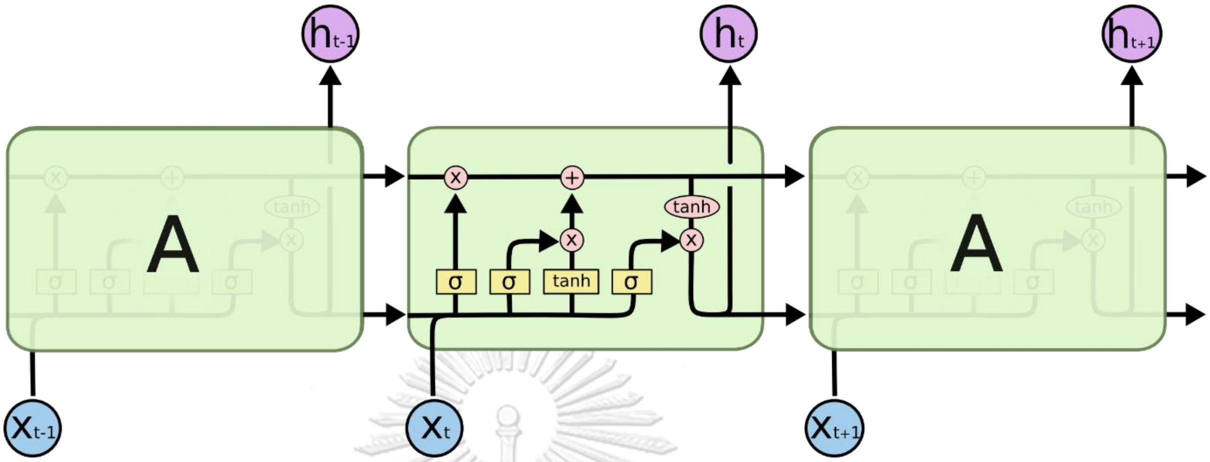


Figure 5. A simple LSTM cell unit
Source: [59]

Abbasimehr et al. compiled a list of hyper parameters that can be used to fine-tune an LSTM model which includes the lag size, the number of hidden layers, the number of neurons, the dropout rate, the learning rate, the batch size and the epoch size[60]. A lag size is a selection of past values to be used in the future prediction[61]. In some papers, a lag size can also be defined as time window size [62], [63], [64] and [65]. A suitable selection of a lag size can improve the performance of an LSTM model since a lag captures temporal behavior of time series data. Neurons in an LSTM model is also known as LSTM cell as mentioned in [66] and [67]. Too few neurons will give poor learning ability of the model, whereas too many neurons might lead to overfitting.

Decision Support System Design

DSS was pioneered by Ralph H. Sprague Jr. on developing a framework for a DSS in 1980 which became a foundation of his book entitled “Building Effective Decision Support Systems” published in 1992 [68]. In early 1990s, the term DSS was replaced by the term “business intelligence” which envelopes the term for applications, technologies, and process for gathering, storing, accessing, and analyzing data to help making business decision [69]. Thus, whenever there is a business decision to be made, a DSS can help in making a better decision which is also applicable to the field of agriculture. A Sprague's typical model of DSS contains three major parts which are model base management system, database management system, and user interface. The main feature of a model base management system is to provide data analysis which can be manifested in the form of machine learning model in the current state of the art model. A human decision making has several problems, namely cognitive limitation, cost of assistance, temporal constraint, collaboration constraint, and also low trust [70] which can be solved by a DSS.

DSS design has three phases: data/theory interaction, simulation/theory interaction, and decision/design interaction [71]. In phase I, data are matched with theory, this is the phase where testing is sometimes performed before design in order to ensure which theory would fit the available data or what data would be required to fit certain theory. In phase II, the theory is then applied to data that results as simulation-like activities where the output is used for further analysis. In phase III, the interpretation of output from previous phase is constituted, resulting in the decision making.

A database in DSS can range from a spreadsheet file or a Database Management System (DBMS) depending on the scale of DSS. If a security of data is the utmost important then a DBMS is a requirement. Nowadays, the operations of DBMS have been shifted to cloud computing instead of a dedicated hardware to handle data storage and data manipulation; for example, a cloud-based Digital Farm Management System uses a cloud storage to store data to allow the DSS to access data from anywhere using web-browser interface while retaining the service quality and transparency [72].

Chapter 4

Methodology

System Design

The proposed system was designed to assist the decision making regarding cultivation of horticultural products which returns the suggestion of which products to be cultivated in a given time period and a cultivation schedule of the products. The recommendation is made based on predicted price using climate parameters, product prices, and other input such as the location of cultivation and the start and the end of cultivation period. The proposed system thus consists of three phases as follows:

1. Price prediction

   Price prediction is performed by using machine learning methods. In this research, several machine learning methods are used in the prediction, namely SVR, BPNN, DNN, and LSTM. The most suitable method with the highest accuracy is then optimized to be used in price prediction. The prediction is based upon the product prices alongside climate parameters.

2. Commodity recommendation

   Based upon the prediction made in phase one, the best periods to cultivate products are determined according to their prices, seasonality, location, and productivity index. In this phase, inputs from a user, which are cultivation location, starting month and ending month of cultivation, are required.

3. Cultivation scheduling

   Recommended cultivation periods from phase two are then used to set up the cultivation schedule for the selected period according to the user's input. This phase returns the cultivation schedule that is the most suitable schedule and yields the highest profit.

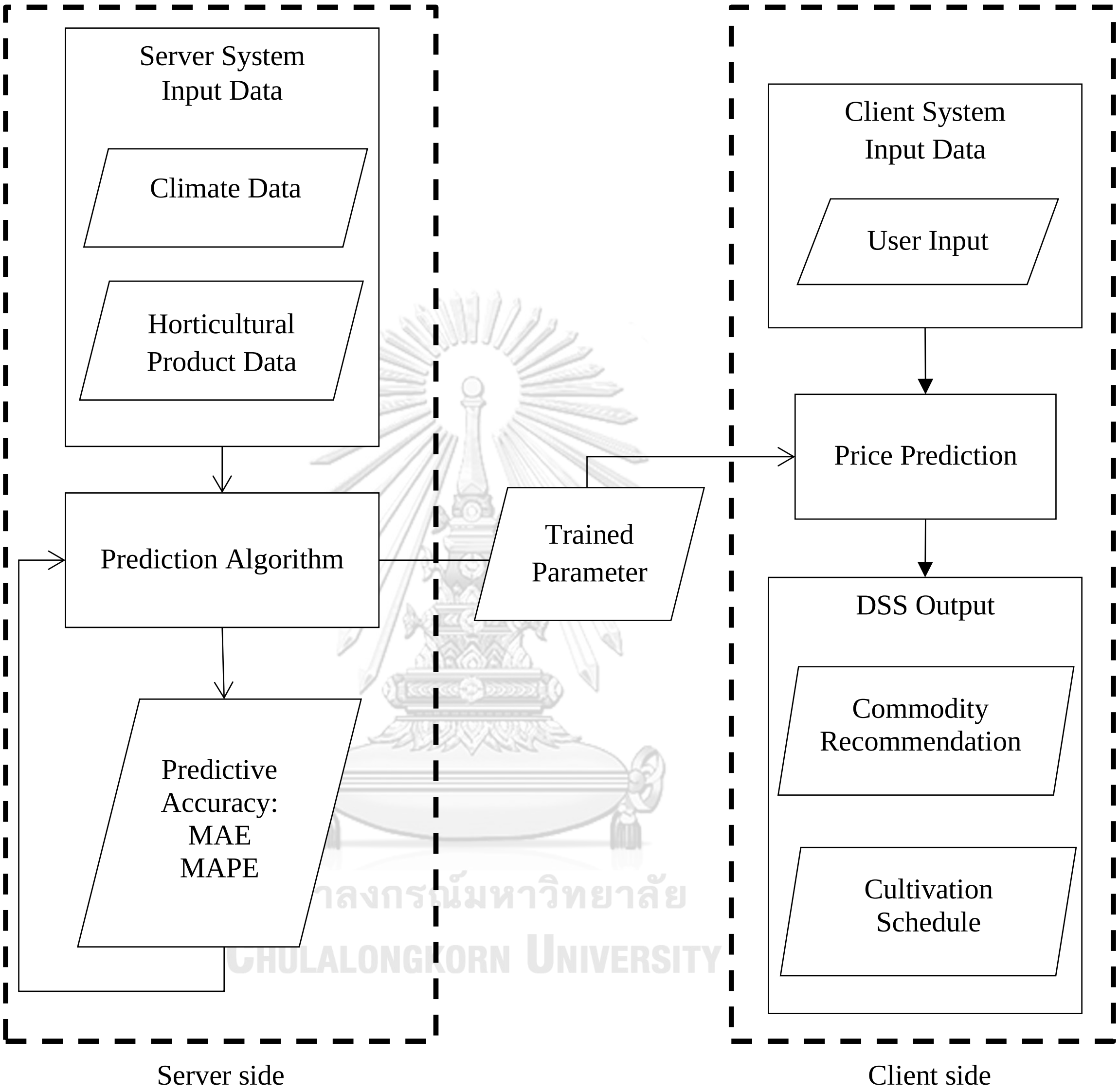


Figure 6. The proposed system diagram

The proposed system was designed to be executed on Keras[73] framework which provides the checkpoint system that saves the weights of the trained model into a file which can be uploaded onto other computer systems with the same model configuration.

The proposed system consists of two subsystems: the server side and the client side as illustrated in Figure 6. The server side is responsible for the price prediction because the training of the model with new data can be very time consuming and requires rather high computing power thus it needs computer system with better specification, the server-side subsystem output will be the trained parameters of the selected model, these trained parameters are usually small in size and can be transferred to any other computer systems very quickly and easily. This is enabled by Keras checkpointing system that allows the weights of trained model to be saved as .h5 files. The server-side subsystem is designed to work periodically or continuously to update these weight files every time new data are available. The client-side subsystem is responsible for commodity recommendation and cultivation scheduling, this subsystem gives recommended commodities along with recommended cultivation time and a cultivation schedule based on recommended commodities.

The inputs of the system are separated into two parts: input for the server side and input for the client side. Server-side subsystem does not require user input, the inputs to this subsystem are the prices of commodities and climate parameters which are mainly used to train models to predict the prices as described in Data Source section. The inputs to the client-side subsystem are user inputs which consist of the location of cultivation, starting and ending of cultivation period. The process of how the inputs are used to generate commodity recommendation and cultivation schedule are described in Commodity Recommendation and Scheduling System section.

Data Preprocessing

Data preprocessing in this research follows the process of Han et al. [30] which includes data cleaning, data integration, and data transformation. More details about the input data, data management system, and additional data preprocessing are discussed in the following subsections.

Data Source

There are two sources of data in this research, meteorological data and horticulture product data, the details are as described below.

Meteorological Data

Meteorological data were obtained directly from the office of Thai Meteorological Department, Bangkok. The data contain four climate parameters from 33 selected agro weather stations across Thailand; they are temperature (in ºC), rainfall (in mm), wind speed (in knots), and daylight duration (in hours). The data were collected from January 1st, 2008 to December 31st, 2018 which account for 4093 samples of data for each climate parameter. The list of 34 stations is shown in Table 3. The Agroclimate stations are installed in plantation areas, thus it would reflect the changes of weather with related to agricultural production. The final values of these climate parameters are the average values of climate parameters from the stations, the stations that did not have complete data were omitted entirely. The stations that were omitted included station no 328301, 373301, 376301, and 429301.

Table 3. Weather station list

No	Weather Station Number and Location
1.	303301-Chiang Rai SMA, Chiang Rai Province
2.	328301-Lampang SMA, Lampang Province
3.	331301-Nan SMA, Nan Province
4.	353301-Loei SMA, Loei Province
5.	356301-Sakon Nakhon SMA, Sakon Nakhon Province
6.	357301-Nakhon Phanom SMA, Nakhon Phanom Province
7.	373301-Si Samrong SMA, Sukhothai Province
8.	376301-Doi Muser SMA, Tak Province
9.	381301-Tha Phra SMA, Khon Kaen Province
10.	386301-Phichit SMA, Phichit Province
11.	400301-Tak Fah SMA, Nakhon Sawan Province
12.	402301-Chai Nat SMA, Chainat Province
13.	405301-Roi Et SMA, Roi Et Province
14.	407301-Ubon Ratchathani SMA, Ubon Ratchathani Province
15.	409301-Sisaket, Sisaket Province
16.	415301-Phra Nakhon Si Ayutthaya SMA, Phra Nakhon Si Ayutthaya Province
17.	419301-Pathum Thani SMA, Pathum Thani Province

No	Weather Station Number and Location
18.	423301-Chachoengsao, Chachoengsao Province
19.	424301-Ratchaburi, Ratchaburi Province
20.	425301-U Thong SMA, Suphan Buri Province
21.	429301-Samut Prakan, Samut Prakan Province
22.	431301-Pak Chong SMA, Nakhon Ratchasima Province
23.	432301-Surin SMA, Surin Province
24.	451301-Nakhon Pathom, Nakhon Pathom Province
25.	455301-Bangkok Bangna SMA, Bangkok
26.	478301-Huai Pong SMA, Rayong Province
27.	480301-Flipper SMA, Chanthaburi Province
28.	500301-Nong Phlap SMA, Changwat Prachuap Khiri Khan
29.	517301-SWS SMA. Chumphon Province
30.	551301-Surat Thani SMA, Surat Thani Province
31.	552301-Nakhon Si Thammarat SMA, Nakhon Si Thammarat Province
32.	560301-Phatthalung SMA, Phatthalung Province
33.	568301-Kho Hong SMA, Songkhla Province
34	581301-Yala SMA, Yala Province

Horticultural Product Data

The horticultural product prices were taken from the Department of Internal Trade website [74]. The price data were collected daily from January 1st, 2008 to December 31st, 2018. They include the prices of several kinds of vegetables, food plants and fruits. All prices are in Thai Baht. The sample sizes of some products are not the same because of their seasonality as can be seen in Table 4. Price data were not recorded during weekends or some national holidays, thus the missing prices were omitted. Moreover, the prices of some products such as raw mango and ripe mango were also missing due to their seasonality which can only be cultivated in certain months of the year.

Table 4. Sample size of horticultural product data

No.	Product Name	Sample Size
1.	Watermelon	2676
2.	Raw Mango	810
3.	Ripe Mango	1145
4.	Pineapple	2676
5.	Fresh Chilies	2676

No.	Product Name	Sample Size
6.	Mixed Cabbage	2676
7.	Large Mixed Tomato	2676
8.	Ginger	2676
9.	Cucumber	2676
10.	Chinese Kale	2676
11.	Chao Phraya Lettuce	2676
12.	Celery	2676
13.	Yardlong Bean	2676
14.	Spring Onion	2676
15.	Mixed Lettuce	2676
16.	Tamarind	2675
17.	Medium Garlic	2675
18.	A Grade Potato	2676
19.	Shallot	2675
20.	O Grade Potato	2676
21.	Dried Garlic	2610

Data Management System

Input data for the server-side subsystem consisting of climate data and price data, as described in previous section, are fed into prediction algorithm of the model as illustrated in Figure 6. The input data consist of 21 products, and four climate parameters. The data are stored in four files: three files for prices of three product categories: fruits, food plants and vegetables, and one file for climate parameter data.

In machine learning, the common data processing package is Pandas package, this package is based on McKinney's paper[75], which was developed into a package by Pandas developer team[76]. Pandas has versatile support of data files, powerful data query and providing statistic calculations and basic visualization [77].

Data Normalization

Since each parameter has different unit and different range, it is necessary to normalize the data of each parameter to avoid prediction errors caused by this problem. The normalization is performed using MinMax normalization that maps a value into a value in the range of the specified minimum and maximum values [78] using Eq. … (21)

$$v' = \left(\frac{v - min_p}{max_p} \times \left(newmax_p - newmin_p\right)\right) + newmin_p \qquad \dots (21)$$

where v is the original value, v' is the normalized value, p is the dataset of each parameter, min_p is the minimum value of each parameter, max_p is the maximum value of each parameter, $newmin_p$ is the specified minimum value and $newmax_p$ is the specified maximum value. In this research, all data are mapped into the value in the range of [0, 1].

Seasonal and Trend Decomposition using LOESS

STL decomposition in the system is performed using *statsmodels* package which was developed by Seabold et al.[79]. *Statsmodels* are capable of performing estimation of various statistical models while also providing support in conducting statistical tests and statistical data exploration. The decomposed prices of the horticultural products are used later in optimization of prevailing model in price prediction.

Machine Learning Model Configuration

In order to deliver optimal performance, machine learning models need to be fine-tuned for its hyperparameters as described in the following sections. All of the machine learning models in the system were built using *Keras* package on python 3.7, with additional packages such as *scikit-learn* [80] for data preprocessing, *matplotlib* [81] for graph displaying and also *numpy* for additional mathematical computation [82].

The data used in each machine learning model are split in the same proportion. The data set of normalized data after cleaning are split into 70-30 where 70% of data are used for training and 30% of data are used for testing. The testing set contains data which have not been exposed to machine learning models, thus a prediction using testing data can provide an insight of how the trained model would respond to real world data. A special treatment must be taken on LSTM model since it requires a time window as additional input, the conversion from a regular data set into data set with time window input is performed after data splitting, thus the values in training set and testing set do not intersect in any way. More on how the data are transformed can be observed in Long Short-Term Memory Neural Network section.

Support Vector Machine for Regression

The initial setting of support vector machine (SVM) is configured by using tolerance value (ε) of 0.003, cache size of 200, and maximum iteration value of -1 which signifies that there is no limit on the number of iterations and they are default setting of SVR in *scikit-learn* package. SVR is optimized to obtain the best prediction performance by varying the values of C and the kernels. The values of C are set to the multiples of 5 in the range of 5 to 250, and the selected kernels are linear kernel and RBF kernel. The optimization was performed by using *GridSearchCV* in python which is a package in *scikit-learn*. *GridSearchCV* is basically an exhaustive search on a set range of values of hyper parameters in a SVM or SVR model.

The input data used in prediction by SVM are a combination of climate values and the prices to predict the next day price as illustrated in Eq.(22) and Eq.(23) where t_i, r_i, w_i, s_i are temperature, rainfall, wind speed and daylight duration of the *i*-th time step, and p_{i+1} is the next time step product price value.

$$input_i = \{t_i, r_i, w_i, s_i\} \quad \dots (22)$$

$$output_{i+1} = p_{i+1} \quad \dots (23)$$

Backpropagation Neural Network

BPNN model has initial setting architecture of four input nodes, 16 hidden layer nodes, and one output node, using 1000 epochs (iterations), batch size of 16, mean squared error loss function and Adam as learning algorithm. The initial number of hidden layer nodes is based on the size of input. The input data and output of BPNN are the same as in SVR. BPNN model is fine-tuned by varying the number of nodes in hidden layer by the values of 2, 4, 8, 16, 32, 64, 128, 256, 512, and 1024 nodes, which are the factors of 2 in accordance with the size of input which is the factor of 2 as well.

Deep Neural Network

Configuration of DNN model is similar to that of BPNN model, except more hidden layers are used instead of a single layer, thus DNN model in this system has an initial setting architecture of 4 input nodes, 16 nodes in the first hidden layer, 16 nodes in the second hidden layer, 8 nodes in the third hidden layer and one output node. The initial number of nodes in the hidden layers are also chosen with similar consideration to that of BPNN model. The rest of hyper parameters are the same as in BPNN model. The input data and output in DNN model are also the same as that in SVR model. DNN model is fine-tuned by varying the number of nodes in three hidden layers by the values of 2, 4, 8, 16, 32, 64, 128, 256, 512, and 1024 nodes, which is based on similar consideration to that of BPNN model.

Long Short-Term Memory Neural Network

LSTM requires different set of input data as opposed to SVR, BPNN and LSTM. To function properly, a LSTM needs to use a sequence of previous time step values. Thus, a modification to the input data needs to be performed to Eq.(22) as in Eq.(24), where $p_{i-n}, p_{i-n+1}, p_{i-n+2}, \dots, p_i$ represent the price of the *i*-th minuses *n* time window size, up to the *i*-th time step price, while the output is the same as Eq.(23).

$$input_i = \{t_i, r_i, w_i, s_i, p_{i-n}, p_{i-n+1}, p_{i-n+2}, \dots, p_i\} \qquad \dots (24)$$

The definition of *n* is very important since it helps to capture changes of values in the predicted time series, as suggested by Chung et al.[62]. Thus, it is reasonable to fine-tune the performance of LSTM model by varying the time window sizes using values of 2, 4, 8, 16, 32, 64, 128, and 256.

LSTM model uses one LSTM layer with cell size of 4 and one output node, while the size of input node corresponds to the size of n. The choice of the cell size also uses similar consideration as that of BPNN model, although in practice, the size of input to LSTM model is in accordance with time window size. The model is trained using 100 epochs and batch size of 1, where the loss function and learning algorithm are similar to that of BPNN and DNN models. The batch size of 1 is also the result from an optimization of LSTM model in forecasting demand research, thus the use of the value can be justified [60]. The use of 100 epochs is based on initial training on LSTM model, where there is no significant change of error beyond 100 epochs.

Performance Measurement

The use of different machine learning techniques needs to be assessed in performance and the most common performance measurements are MAE and MAPE which signify the error of the system. MAE and MAPE are calculated using Eq. (25) and Eq. (26) where y^i is the *i*-th actual value, *n* is the number of test data and y_i' is the *i*-th predicted value:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - y_i'| \quad \dots (25)$$

$$MAPE = \left(\frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - y_i'|}{y_i}\right) \times 100 \quad \dots (26)$$

Commodity Recommendation and Scheduling System

After the most suitable machine learning model on price prediction is determined, the next two phases of commodity recommendation and its scheduling can be implemented based on several assumptions and limitation which are as follows:

1. The area or the size of the land is not taken into consideration. The prediction is performed based on the product prices and climate parameters without considering the irrigation system and soil fertility.

2. Economic parameters in cultivation process, such as production cost, labor cost, investment size, tools used, and infrastructure are not in the scope of the recommendation system. In the recommendation system, predicted prices of the respective products are the sole factor.
3. Products that are out of season are excluded from recommendation. Since seasonality may cause unsuitable climate for the out of season products to be grown, the product qualities would be below standard.
4. The productivity factor of each product is not to be considered since the data are not available. If such data exist, they would be another valuable parameter in price prediction along with market demand.

The algorithm for commodity recommendation is rather simple and straight forward as can be observed from the pseudocode below where stmo, hvmo and year are starting month, harvesting month and the cultivation period.

```
Let stmo, hvmo and year as user input
For all commodities:
    Predict price for testing data set
    Split predicted price for selected year to monthly prices
    For each months in range[stmo:hvmo]:
        If commodity in season:
            Get monthly average price
            If the price is better than previous price:
                Store price as best price at corresponding month
            Otherwise:
                Next month
        Otherwise:
            Next month
Return best prices and recommended starting and harvesting month for each commodity
```

Additional input for the recommendation model can increase the ability to generate better recommendation. The production index and cultivation location are used for additional input to the recommendation model. Monthly production index values of January 2008 to December 2018 period were taken directly from Office of Agricultural Economics, the values were normalized using MinMax normalization for each year and the average for each month were used as the calibrating weight, these inputs were used to select which month will be more appropriate to harvest the commodity. However, there is a limit for the use of these inputs since production index data are available for only four commodities, which are shallot, garlic, potato and onion. Table 5 displays the monthly average of production index values, for both real and normalized values. The average values in normalized value column were multiplied by the predicted prices of the corresponding months, thus the month with the highest value is recommended for harvesting month.

Cultivation location was selected based on production volume of commodities in corresponding area as production index, these values were normalized using MinMax normalization as weights. The data that were used to calibrate cultivation location were taken from Agricultural Statistics of Thailand Yearbooks of 2012 to 2019 that was released by Office of Agricultural Economics in their website [83]. The yearbooks contain yearly production volume data for the same commodities as in production index for respective province where they were produced. The yearly average production value for each location was taken and normalized using MinMax normalization and was used as weighting factor on input of cultivation location. Table 6 shows the production location and the yearly average production volume values for shallot. The values in the normalized mean column were used to determine if the location in the input can be recommended for cultivation of the corresponding commodities. In location selection, the threshold value was used instead of multiplication with predicted prices, thus the only locations that are listed in cultivation location and having normalized average values are recommended as cultivation location. However, the available data for production location were limited to previously mentioned commodities.

The recommendation model with additional input uses a slightly modified original algorithm, which can be observed in the algorithm below, with additional inputs of location and threshold value. The parts that were modified from the original algorithm were printed in bold.

```
Let loc, thr, stmo, hvmo and year as user input
For all commodities:
    If loc is in loc list AND commodity mean normalized value >= thr:
        Predict price for testing data set
        Split predicted price for selected year to monthly prices
        For each months in range[stmo:hvmo]:
            If commodity in season:
                Get monthly average price
                Multiply average price with normalized production index value
                If the price is better than previous price:
                    Store price as best price at corresponding month
                Otherwise:
                    Next month
            Otherwise:
                Next month
    Otherwise:
        Next commodity
Return best prices and recommended starting and harvesting month for each commodity
```

Table 5. Monthly production index values for shallot

Month	Monthly Mean	Normalized Mean
1	144.0526272	0.656727
2	139.1455027	0.607574
3	223.5677198	0.969474
4	84.02872946	0.350111
5	15.9327734	0.064961
6	0.227680569	0
7	31.6973065	0.129057
8	29.47954382	0.128673
9	21.41302236	0.095993
10	31.53686246	0.133231
11	63.36049817	0.263204
12	101.4514326	0.456212

Table 6. Yearly production volumes for shallot

Cultivation Location	Mean	Normalized Mean
Chiang Rai	1979.375	0.034013
Phayao	13566.63	0.2439
Lampang	915.375	0.01474
Lamphun	13384.88	0.240608
Chiang Mai	26539.25	0.478882
Mae Hong Son	1208.125	0.020043
Sukothai	1493.125	0.025205
Tak	1039.125	0.016982
Nan	303.125	0.00365
Uttaradit	15440.13	0.277836
Phetchabun	8743	0.156527
Yasothon	2007.875	0.034529
Ubon Ratchatani	794.75	0.012555
Si Sa Ket	55308.63	1
Surin	750.875	0.01176
Buri Ram	2665	0.046432
Chayaphum	3121.375	0.054699
Other	101.625	0

The output of the second phase is a recommendation of starting month and harvesting month of each product that is in season, for both the original and the model with additional input. This output is then converted to a cultivation schedule in the last phase with several assumptions as follows:

1. There is only one product to cultivate at a time since the size of the land is not being considered, the mixed farming is unfeasible.
2. A starting month of the next product can overlap with a harvesting month of previous product since the start of cultivation process can usually be done right after harvesting for most of the products.

3. There is no repeated cultivation of the same product in the specified time period, this is due the fact that the commodity recommendation is based on the best price in a specified period, thus this case can rarely happen.

The scheduling phase is based on the output from the recommendation phase. To put it simply, given the starting month and harvesting month in a certain year, the best time to start cultivation and harvesting time for each in season product are based on the results from the recommendation phase. Thus, some of the products may have overlapped starting and harvesting month in the specified year. The logical way is to provide a sequence of cultivation for the recommended products which enables a user to select any sequence that would fit his preference. The algorithm of scheduling can be observed in the pseudocode below.

```
Let recommended commodities for specified stmo, hvmo and year as input
Let cultivation schedule as a list of sequence
For each commodities:
   Get pstmo and phvmo, add commodity into temporary sequence, remove from
commodity pool
   While (commodity pool ≠ []) or (pstmo ≠ stmo):
      Look for next product where pstmo = previous product phvmo
      Get pstmo and phvmo, add commodity into temporary sequence, remove from pool
   add final sequence into cultivation schedule
return final cultivation schedule
```


The output of the scheduling phase consists of all the possible cultivation schedules for each recommended product thus it could result in several alternatives having common ending sequence or starting sequence. As the result of the third assumption, these sequences will have singular starting month and harvesting month which also provides the idle months when there is no cultivation activity for several alternatives.

Chapter 5

Results

Data management system

As described in Data Management System section in Chapter 4, the use of data management system (DMS) is required to manage data in price prediction. Three MS Excel files that are used to store data have structures as follows:

1. The data of climate parameters are stored in a single MS Excel file containing date, temperature, rainfall, wind speed, and daylight duration for 4092 days. These data are stored in a form of daily average values of all stations thus they may contain missing values and recorded errors.
2. Products regarded as fruits are stored in one file with one product per worksheet in MS Excel. The products are watermelon, raw mango, ripe mango and pineapple. These data are stored in their original values which were retained from Department of Internal Trade, Thailand [74].
3. Products regarded as vegetables are stored in a similar way as those categorized as fruits. The products are fresh chilies, mixed cabbage, mixed large tomato, ginger, cucumber, Chinese kale, lettuce, mixed lettuce, celery, Yardlong beans, and spring onions.
4. Products regarded as food plants are also stored in a similar way as those of fruits. The products are tamarind, medium garlic, shallot, dried garlic, grade a and grade o potato.

Data preparation using DMS is illustrated in Figure 7. For example, Chinese kale is taken as a case; the 2676 rows of price data of Chinese kale are first loaded, along with 4051 rows of climate parameter data. These data sets are then combined into a single data set using date as a reference. However, climate data were reduced from 4092 rows of raw data due to incomplete data during loading process. The combined data set is then screened for NA values, zero value for the average temperature and zero value for the product price, resulted in a cleaned data set of 2669 rows.

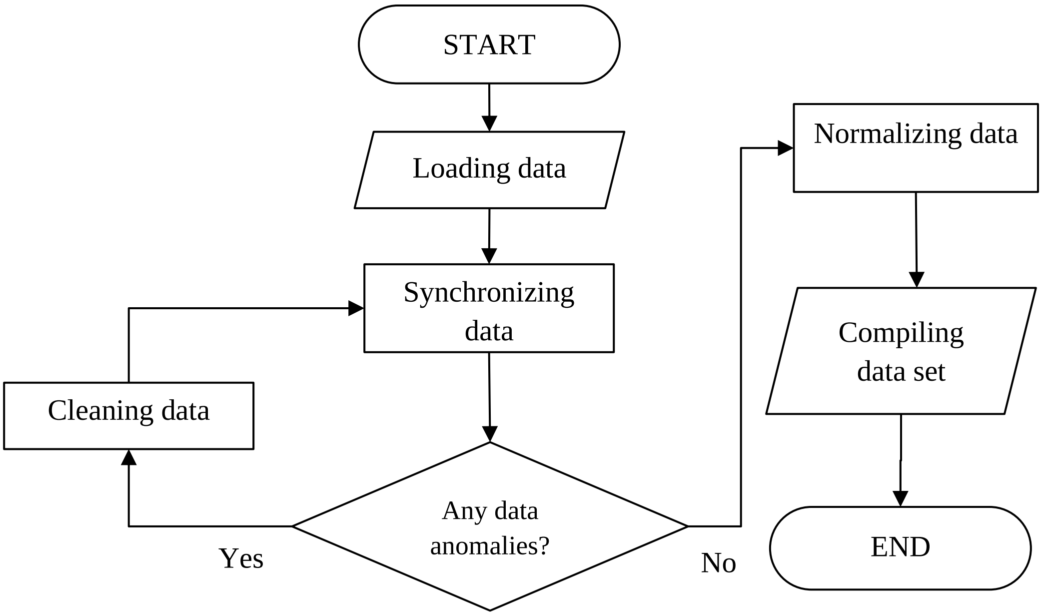


Figure 7. Data management system workflow

Beside using DBMS to normalize data and convert data back to their original values, it also provides additional function such as displaying price graph, as displayed in Figure 8.

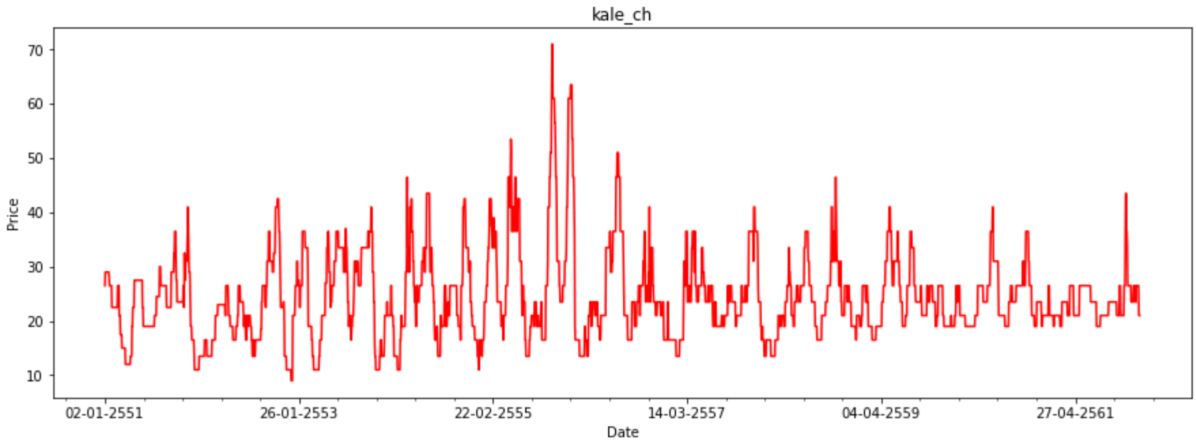


Figure 8. Chinese kale prices

Model performance comparison

The performances of several machine learning techniques are evaluated by two data sets. One data set is a large data set of Chinese kales that contains only a few missing data and another small data set of raw mangos that contains more missing data. Chinese kales are year-round commodities which are available throughout the year, while raw mangoes are seasonal fruits that are available in some certain months only. Figure 9 shows raw mango prices before preprocessing where the prices are equal to zero in some periods. The raw data set originally contains 2100 data before preprocessing compared to 809 data after preprocessing as shown in Figure 10.

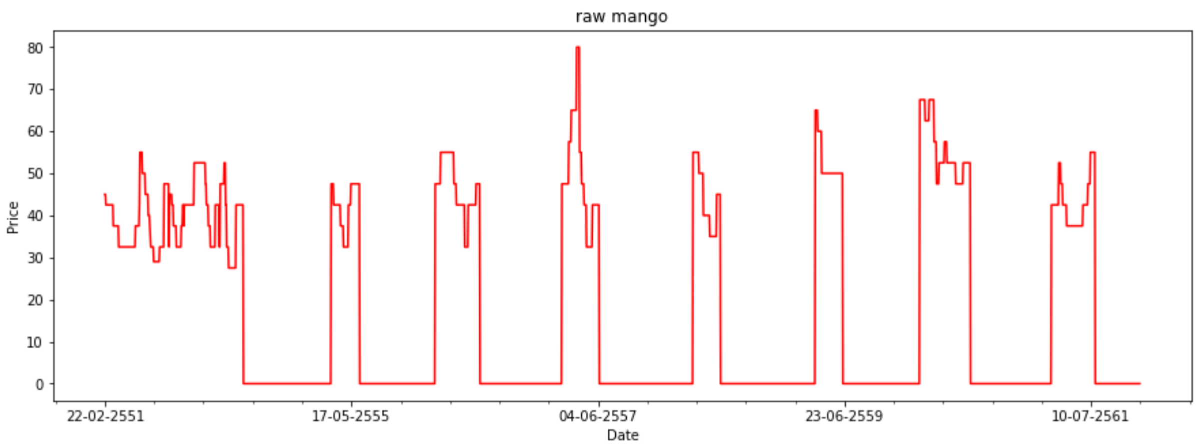


Figure 9. Raw mango price data before preprocessing

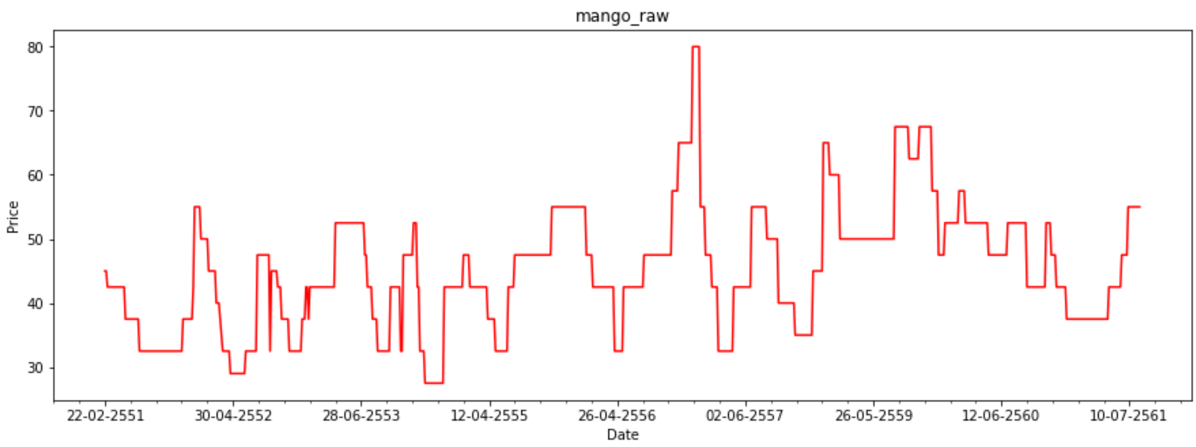


Figure 10. Raw mango price data after preprocessing

The performances of SVM, BPNN, DNN, and LSTM models are compared based on the prediction of price for the next time step (next day price). For Chinese kale data set, SVR performance yields MAE of 3.76 and MAPE of 15.52. After optimization using *GridSearchCV* with the setting using linear kernel and C value of 40, the performance does not change which implies that this is the optimal performance for SVR. On the other hand, BPNN performance yields MAE of 4.22 and MAPE of 27.22 with the initial settings and after optimizing the setting by using two nodes in a hidden layer, it gives better performance with MAE of 3.85 and MAPE of 16.10. For DNN model, the performance yields MAE of 4.51 and MAPE of 18.76 with the initial settings but after optimization the setting by using four nodes in each hidden layer, it gives better performance with MAE of 3.80 and MAPE of 15.43. The training performance can be observed in Table 7 and it can be observed that LSTM model outperforms all other models even without optimization with MAE of 0.74 and MAPE of 2.98.

Table 7. Performance comparison on Chinese kale data set

Performance Measure	Model							
	BPNN		DNN		SVR		LSTM	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing
MAE	6.48	3.85	6.34	3.80	6.47	3.76	1.28	0.74
MAPE	29.22	16.10	27.93	15.43	28.56	15.52	5.17	2.98

In Figure 11, it can be observed that predicted Chinese kale prices throughout the same period resulted from prediction by BPNN, DNN, SVR and LSTM models. It is also observable that the predicted prices that were resulted from LSTM model are tied to those of the real values.

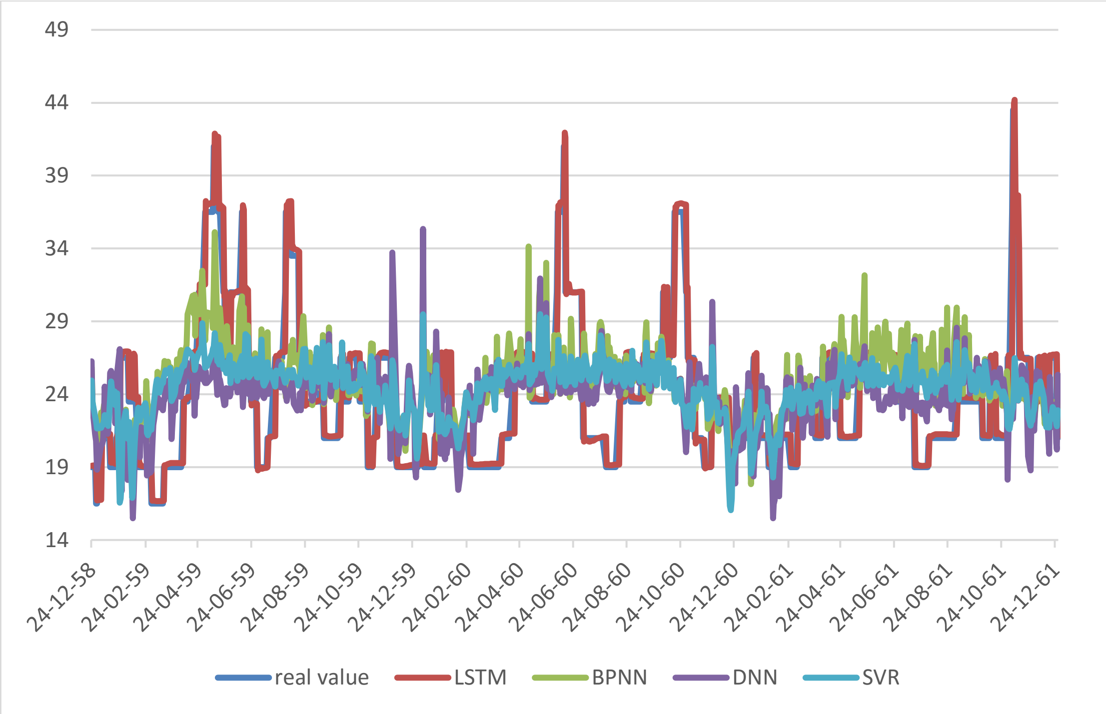


Figure 11. Chinese kale price prediction

For raw mango data set, SVM model with the initial setting yields performance with MAE of 9.22 and MAPE of 17.44 and after optimization using RBF kernel and C value of 5, it yields better performance with MAE value of 9.15 and MAPE value of 17.37. For BPNN model with initial setting, its performance yields MAE of 9.42 and MAPE of 17.97 and after optimization using eight nodes in the hidden layer, its performance yields MAE of 8.69 and MAPE of 16.55. For DNN model with the initial setting, its performance returns MAE of 10.60 and MAPE of 20.41 and after optimization using the configuration of 4-8-4 nodes in the first to the third hidden layers, it yields better performance with MAE of 7.66 and MAPE of 14.63. For raw mango data set, the results are similar to Chinese kale data set, LSTM model also outperforms all other models even without any optimization. The complete performance measurement can be observed in Table 8.

Table 8. Performance comparison on raw mango data set

Performance Measure	Model							
	BPNN		DNN		SVR		LSTM	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing
MAE	5.38	8.69	6.19	7.66	5.76	9.15	1.22	0.77
MAPE	12.81	16.55	15.03	14.63	13.59	17.37	2.95	1.55

In Figure 12, it can be observed that the raw mango price prediction from BPNN, DNN, SVR and LSTM model also show distinctively different performance. It can simply be seen that as in Chinese kale price prediction, the predicted prices of raw mango that were obtained from LSTM model are close to those of the real values.

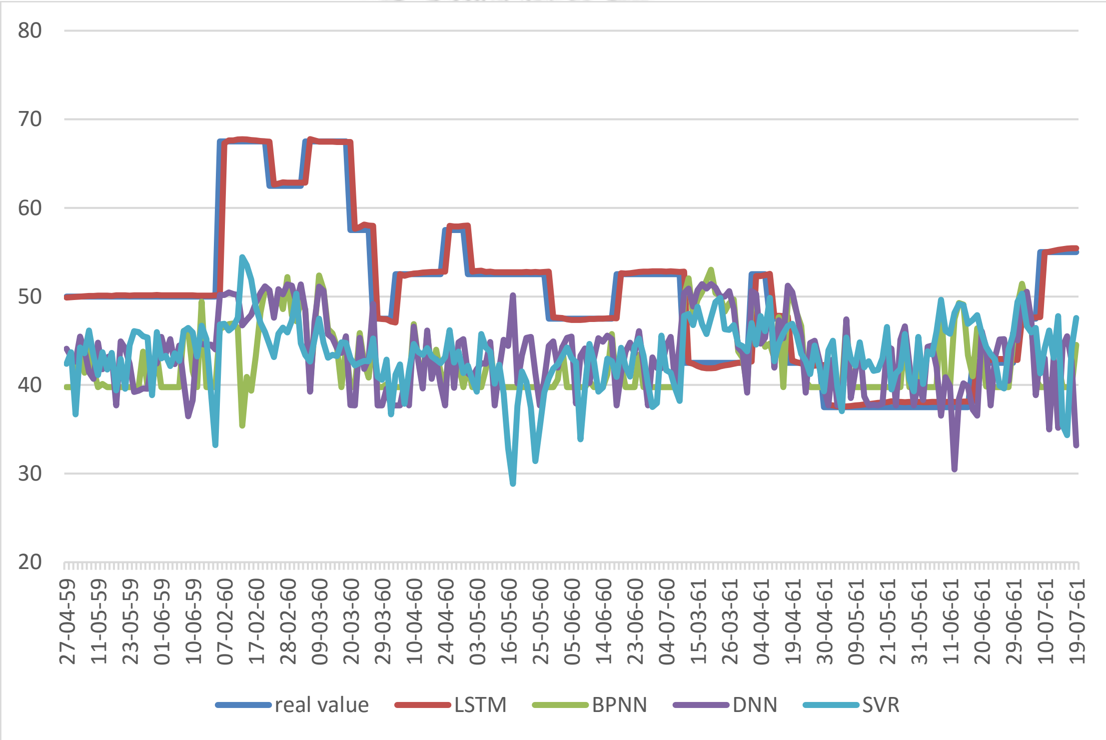


Figure 12. Raw mango price prediction graph

Multistep prediction using LSTM model

Performance of LSTM model in comparison with other models has been shown as superior for one day ahead prediction but such prediction is not very useful for farmers since it usually takes months to cultivate most of the products, especially perennial products which take several months to cultivate, thus a multiple time step ahead prediction would be more helpful. Several papers on other fields also used LSTM model to predict multistep ahead, for instance, one and 24-step ahead prediction were performed to predict electricity prices in Europe [84], one and seven-step ahead of daily data prediction were performed to predict stock prices [65], time steps of 24, 48, 72, 96, 120, 144, and 168 were done to predict PM 2.5 concentration using hourly data [85] and 2, 4, 8, 6, 8, and 10 time step ahead prediction were made to predict sea surface temperatures [86]. Thus, the same method can be deemed as viable to be done as well on predicting commodities price using LSTM model.

Input data for multistep ahead prediction are the same as those of a single step but the output is different, thus the output can be expressed as in Eq.(27), where k is the number of time steps(days).

$$output_{i+1} = p_{i+\mathrm{k}} \qquad \text{… (27)}$$

Predictions for 30, 60, 90, 120, 150, 180, 210, 240, 270, 300, 330 and 360 time-step ahead are performed using LSTM model. Since the prices were not recorded in the weekends, an offset of 8 days per month needs to be accounted for the time steps, thus the time steps are adjusted to 22, 44, 66, 88, 110, 132, 154, 176, 198, 220, 242, and 264 time steps. Table 9 shows the prediction results for Chinese kale and raw mango data sets; however, prediction for 330 and 360 days ahead for raw mango data set cannot be performed due to lack of data. The prediction performances for Chinese kale vary among different time steps but not much difference, whereas the prediction performances for raw mango vary with high fluctuations; for example, prediction performance for 30 days ahead results in MAE of 12.63 and MAPE of 31.05. The same high error also occurs in the prediction for 240 days ahead with MAE of 11.37 and MAPE of 36.57 and the prediction for 300 days ahead also exhibits the same high error with MAE of 11.49 and MAPE of 26.4 while the performances for other time steps are almost similar to those of Chinese kale.

Table 9. Multistep prediction performance using LSTM model

Dataset	Model	Multi-Day Ahead											
	Performance	30	60	90	120	150	180	210	240	270	300	330	360
Chinese	MAE	4.36	4.11	3.86	3.57	3.82	4.54	3.72	3.21	3.49	3.58	4.37	4.12
kale	MAPE	16.47	16.16	15.07	16.1	15.45	17.03	14.64	13.51	13.95	14.81	16.19	15.66
raw	MAE	12.63	7.2	6.76	8	6.73	5.28	6.44	11.37	5.86	11.49	-	-
mango	MAPE	31.05	15.64	15.7	16.2	13.94	11.96	13.96	36.57	12.99	26.4	-	-

Optimization of LSTM prediction performance

As mentioned before, LSM model can be optimized in order to improve its performance, and one of the suggested ways is to modify the time window size of input data. An optimization is performed for each product by varying time window size and the performance of single time step ahead prediction for each data set is shown in Table 10. It can be inferred that each product requires different time window size.

Table 10. Optimized LSTM performance for each horticultural product

Product Name	Time Window Size	MAE	MAPE
Watermelon	4	0.05	0.20
Raw Mango	16	0.77	1.56
Ripe Mango	8	1.78	1.93
Pineapple	2	0.09	0.30
Fresh Chilies	64	0.06	1.16
Mixed Cabbage	256	0.23	1.39
Large Mixed Tomato	16	0.53	2.23
Ginger	16	0.20	0.33
Cucumber	32	0.76	3.23
Chinese Kale	64	0.74	2.98
Chao Phraya Lettuce	256	0.41	1.92
Celery	64	0.57	6.57
Yardlong Beans	256	2.24	6.19
Spring Onion	8	0.29	3.72
Mixed Lettuce	128	3.66	9.12
Tamarind	128	1.13	1.24
Medium Garlic	4	0.77	0.58
A Grade Potato	8	0.12	0.38

Product Name	Time Window Size	MAE	MAPE
Shallot	128	0.66	1.00
O Grade Potato	4	0.22	0.59
Dried Garlic	2	1.43	1.49

Multistep prediction for selected products using different time window sizes can be observed in Table 11. The selected products are pineapple, spring onion, raw mango, Chinese kale and Chao Phraya lettuce for time window size of 2, 8, 16, 64 and 256, respectively. It can be observed that the multistep prediction performs differently on different products. It can also be observed that different time window sizes can affect the performance of multistep prediction as can be compared between the predictions of Chinese kale and raw mango in Table 9 and those in Table 11.

Table 11. Multistep prediction for selected products

Product Name		Prediction Step											
		22	44	66	88	110	132	154	176	198	220	242	264
Pineapple	MAE	1.88	2.23	2.36	3.06	3.64	3.79	12.26	15.2	14.39	13.47	20.03	16.32
	MAPE	6.42	8.02	9.05	11.53	14.08	15.47	47.48	59.65	57.6	55.04	82.08	68.4
Spring Onion	MAE	1.59	1.7	1.72	1.83	1.96	1.74	1.77	1.84	1.86	1.7	1.89	1.74
	MAPE	18.76	19.42	19.6	20.85	22.58	19.51	18.98	21.14	21.25	19.45	21.01	20.52
Raw Mango	MAE	12.63	7.2	6.76	8	6.73	5.28	6.44	11.37	5.86	11.49	-	-
	MAPE	24.73	14.17	13.95	18.33	15.93	12.05	15.54	24.66	12.33	20.89	-	-
Chinese Kale	MAE	3.83	3.85	5.02	3.57	3.75	3.64	3.64	4.12	3.68	3.73	6.53	6.56
	MAPE	15.97	16.62	20.11	14.7	15.68	14.73	15.08	17.12	16.2	16.45	28.78	28.7
Chao Phraya Lettuce	MAE	3.53	3.1	3.22	3.16	3.05	3.52	2.95	3.25	3.04	3.16	3.44	3.53
	MAPE	17.09	14.77	15.56	16	15.49	18.05	14.11	16.71	15.5	16.28	18.18	18.49

Prediction performance of LSTM model can also be enhanced by using STL decomposition on prices before inputting them into LSTM model. STL decomposes prices into seasonal, trend, and residual components as demonstrated in Figure 13 (a) and (b). The characteristics of components depend on the time step used when decomposing the prices, larger time step results in smoother components, while smaller time step results in more fluctuation on the components. Since smoother component is preferred to be input into LSTM model, thus the STL period for raw mango and ripe mango is set to 66 days which equals to 90-day period to anticipate three months of mango season, while the rest of the products are considered as year-round commodities thus using 264 days which equals to 360-day period.

For STL-LSTM model, product prices are first decomposed into three components, namely X_s, X_t and X_r which represent seasonal, trend, and residual components. Data of each component are used to predict the value of that component for a given time step, resulted in Y_s, Y_t and Y_r which represent predicted values for seasonal, trend and residual components. The values of these three components are then combined to form the predicted price, Y, as described in Eq.(28).

$$Y = Y_s + Y_t + Y_r \qquad \dots (28)$$

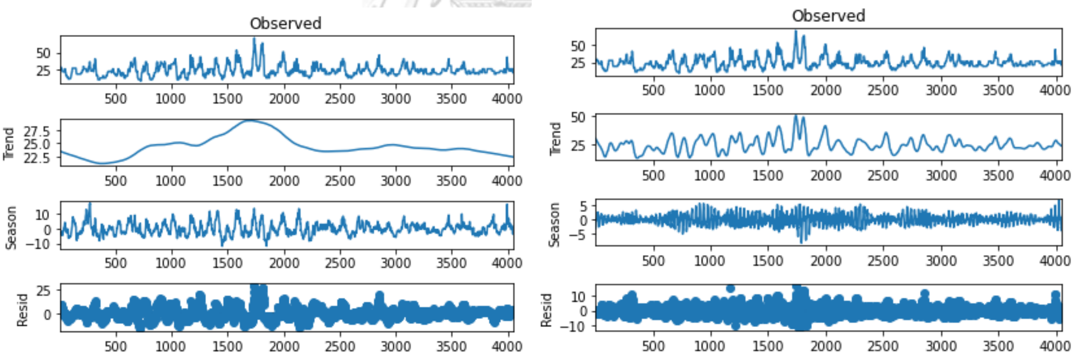


Figure 13. Price decomposition for Chinese kale data set; (a) with time step of 264, (b) with time step of 22.

Multistep prediction performance for the selected products using STL-LSTM model can be observed in Table 12.

Table 12. Multistep prediction performance using STL-LSTM Model

Product Name		Prediction Step											
		22	44	66	88	110	132	154	176	198	220	242	264
Pineapple	MAE	1	2.08	2.69	2.99	3.41	3.67	4.01	5.96	5.03	6.57	7.96	7.69
	MAPE	3.98	8.46	11.17	12.33	14.05	15.34	16.72	25.17	21.49	28.03	34.5	33.29

Product Name		Prediction Step											
		22	44	66	88	110	132	154	176	198	220	242	264
Spring	MAE	1.53	1.65	1.79	1.86	1.84	1.88	1.82	1.71	1.77	1.79	1.71	1.47
Onion	MAPE	19.29	21.56	23.18	23.73	23.2	23.71	21.07	19.59	20.85	19.98	19.37	16.69
Raw Mango	MAE	6.12	7.49	6.44	8.35	5.2	7.55	7.97	6.01	6.46	5.24	-	-
	MAPE	12.93	15.22	13.88	17.28	12.07	18.44	16.93	14.33	13.88	9.53	-	-
Chinese	MAE	4.69	4.61	4.45	3.7	4.09	4.04	4.16	3.41	3.33	3.35	3.58	2.8
Kale	MAPE	20.26	19.12	16.77	14.99	17.5	17.57	17.41	14.62	14.35	14.89	15.32	11.82
Chao Phraya	MAE	3.2	2.76	2.61	2.87	3.05	3.23	3.14	2.79	3.47	3.1	3.67	3.52
Lettuce	MAPE	15.1	13.07	12.7	14.94	15.61	17.04	16.4	14.86	18.1	15.83	19.32	18.35

Commodity recommendation and scheduling

After determining that the best machine learning model for price prediction is STL-LSTM model, the next phase is the product recommendation and scheduling. In order to recommend which products should be cultivated during the specified period, three input data including the starting month, the harvesting month, and the chosen year are needed to be specified. Since data were collected up until the year 2018, the proposed recommendation and scheduling systems are tested by using data from 2008 to 2017 to give a recommendation and schedule for the year 2018. Three periods are selected to simulate different seasons of commodities, there are some commodities that have harvest season in the first half of the year, and there are also some other commodities that have harvest season in the second half of the year, while a period of one year is also taken to show how the recommendation system would respond to different length of period, hence the selected periods are January to June, June to December and June to December.

Commodity recommendation

As described in Commodity Recommendation and Scheduling System section, seasonality of commodities is taken into consideration for selecting the recommended commodity, while price prediction for each product also uses the respective time window size as shown in Table 10. Each product needs a specific size of time window to give the optimal prediction. The seasons of the products are shown in Table 13 where the year-round harvest season means the respective products do not have specific harvest season, thus they can be cultivated anytime throughout the year and the numbers in harvest season column indicates the month of the season as 01 refers to January and 12 refers to December.

Table 13. Seasonality of Commodities

Commodity Name	Cultivation Period (days)	Harvest Season
Watermelon	80	Year-round
Raw Mango	3650	04,05,06,07
Ripe Mango	3650	04,05,06,07
Pineapple	630	04,05,06,07,12,01
Fresh Chilies	89	Year-round
Mixed Cabbage	100	01,02,03,04
Large Mixed Tomato	70	Year-round
Ginger	225	Year-round
Cucumber	62.5	Year-round
Chinese Kale	70	Year-round
Chao Phraya Lettuce	50	Year-round
Celery	115	Year-round
Yardlong Beans	80	Year-round
Spring Onion	56	Year-round
Mixed Lettuce	50	Year-round
Tamarind	2737.5	12,01
Medium Garlic	147	03,04,05,06
A grade Potato	75	07,08,12
Shallot	89	02,03
O grade Potato	75	07,08,12
Dried Garlic	147	03,04,05,06

*Gathered from various sources.

Table 14, Table 15, and Table 16 provide commodity recommendation for three different periods: January to June, June to December, and January to December, respectively. It can be observed that different selection of starting month and harvesting month in the same year could recommend different products. For example, if the specified period is January-June then one of the recommended products is watermelon, and the suggested schedule is to start cultivating in April and harvesting in June with the expected average price of 26.31 baht per kg. On the other hand, if the specified period is June-December then one of the recommended products is also watermelon but the suggested schedule is to start cultivating in October and harvesting in December with the average price of 29.31 baht per kg.

Seasonal effect also can be observed form the same couple of periods, where mixed cabbage, medium garlic, shallot, and dried garlic are recommended for the period of January-June but are not recommended for period of June-December, while none of the recommended commodities in both periods is recommended for the period of January-December. This difference could be extended into comparison with longer duration. For example, Fresh chilies is recommended to be cultivated for all periods, but the starting month and harvesting month are different according to the specified period. The starting month and harvesting month for January-June period is March through May, whereas the starting month and harvesting month for June-December period is September through November and the starting month and harvesting month for January-December period is May through July.

Table 14. Commodity Recommendation for January to June

No.	product	starting month	harvesting month	exp avg price
0	Watermelon	4	6	26.31212425
1	Fresh Chilies	3	5	5.0571208
2	Mixed Cabbage	3	6	19.70629311
3	Large Mixed Tomato	4	6	22.26581764
4	Cucumber	2	4	20.82969666
5	Chinese Kale	3	5	22.49029541
6	Chao Phraya Lettuce	3	4	20.3427906
7	Celery	2	5	9.1550951
8	Yardlong Beans	4	6	38.28850555
9	Spring Onion	5	6	7.716771603
10	Mixed Lettuce	3	4	36.51749039
11	Medium Garlic	1	5	118.9671097
12	Shallot	4	6	44.61709595
13	Dried Garlic	1	5	88.86936188

Table 15.Commodity Recommendation for June to December

No.	product	starting month	harvesting month	exp avg price
0	Watermelon	10	12	29.31071663
1	Fresh Chilies	9	11	5.067181587
2	Large Mixed Tomato	9	11	24.39205742
3	Cucumber	6	8	20.40604591
4	Chinese Kale	9	11	21.3671875
5	Chao Phraya Lettuce	10	11	18.66643333
6	Celery	6	9	7.322007656
7	Yardlong Beans	7	9	41.4286499
8	Spring Onion	6	7	7.666098118
9	Mixed Lettuce	10	11	37.89729691
10	A grade Potato	10	12	32.03812027
11	O grade Potato	10	12	37.01141357

Table 16. Commodity Recommendation for January to December

No	product	starting month	harvesting month	exp avg price
0	Watermelon	10	12	29.31071663
1	Fresh Chilies	5	7	5.111023903
2	Mixed Cabbage	3	6	19.70629311
3	Large Mixed Tomato	9	11	24.39205742
4	Ginger	4	11	60.83574295
5	Cucumber	2	4	20.82969666
6	Chinese Kale	3	5	22.49029541
7	Chao Phraya Lettuce	3	4	20.3427906
8	Celery	2	5	9.1550951
9	Yardlong Beans	7	9	41.4286499
10	Spring Onion	5	6	7.716771603
11	Mixed Lettuce	10	11	37.89729691
12	Medium Garlic	1	5	118.9671097
13	A grade Potato	10	12	32.03812027
14	Shallot	9	11	69.18836212
15	O grade Potato	10	12	37.01141357
16	Dried Garlic	6	10	89.90684509

Bangkok, Chiang Mai, and Chiang Rai were selected as input for commodity recommendation, the threshold value was set to 0.2, and the period was from January to

December. Figure 14 shows that there is no recommendable commodity for cultivation area of Bangkok since the period is for the whole year and the recommendation would not differ even different period is selected.

```
In [2]: r1 = recommender('Bangkok', 1,12,2561)
no recommendable commodity
```

Figure 14. Commodity recommendation result for Bangkok

Table 17 shows the result of commodity recommendation system for Chiang Mai cultivation location as input. The starting month, harvesting month and expected average price in the table indicate the same use to that of the original recommendation system.

Table 17. Recommended commodities for Chiang Mai, January to December

No	product	starting month	harvesting month	exp avg price
1	Spring Onion	2	3	6.857649
2	Medium Garlic	8	12	104.9174
3	Dried Garlic	8	12	83.59029
4	O grade Potato	1	3	33.51736
5	A grade Potato	1	3	28.98498
6	Shallot	1	3	42.04671

Table 18 shows the result of commodity recommendation system for Chiang Rai cultivation location as input. The starting month, harvesting month, and the expected average price used in the table are analogue to that of the original recommendation system.

Table 18. Recommended commodities for Chiang Rai, January to December

No	product	starting month	harvesting month	exp avg price
1	O grade Potato	1	3	33.5173645
2	A grade Potato	1	3	28.98498154

It is notable that the results of recommendation system using additional input as shown in Table 17 and Table 18 with different input of cultivation locations do affect the recommended commodities. For example, spring onion, medium garlic and dried garlic are recommended to be cultivated in Chiang Mai, while they are not recommended in Chiang Rai, and none of the commodities is recommended to be cultivated in Bangkok, as can be seen in Figure 14.

Commodity scheduling

Scheduling is also evaluated for three periods as in recommendation. The commodity scheduling for January-June, June-December, and January-December periods can be seen in Table 19, Table 20, and Table 21, respectively. The scheduling provides several alternative sequences for cultivation together with the period of idle months when no cultivation takes place. For example, sequence No. 1 in Table 19 suggests that a farmer should start the cultivation with fresh chilies and proceed with spring onion. According to Table 14, we can see that fresh chilies are recommended to be cultivated from March to May, while spring onion is recommended to be cultivated from May to June. Thus, for the period of January-June, there will be no cultivation in January and February, hence there are two idle months in the sequence.

Table 19. Cultivation schedule for the period of January-June

No.	Cultivation sequence	Starting month	Ending month	Idle month
1	Fresh Chilies - Spring Onion	3	6	2
2	Chinese Kale - Spring Onion	3	6	2
3	Celery - Spring Onion	2	6	1
4	Medium Garlic - Spring Onion	1	6	0
5	Dried Garlic - Spring Onion	1	6	0
6	Chao Phraya Lettuce - Watermelon	3	6	2
7	Mixed Lettuce - Watermelon	3	6	2
8	Cucumber - Watermelon	2	6	1
9	Chao Phraya Lettuce - Large Mixed Tomato	3	6	2
10	Mixed Lettuce - Large Mixed Tomato	3	6	2
11	Cucumber - Large Mixed Tomato	2	6	1

No.	Cultivation sequence	Starting month	Ending month	Idle month
12	Chao Phraya Lettuce - Yardlong Beans	3	6	2
13	Mixed Lettuce - Yardlong Beans	3	6	2
14	Cucumber - Yardlong Beans	2	6	1
15	Chao Phraya Lettuce - Shallot	3	6	2
16	Mixed Lettuce - Shallot	3	6	2
17	Cucumber - Shallot	2	6	1

Table 20. Cultivation schedule for the period of June- December

No.	Cultivation sequence	Starting month	Ending month	Idle month
1	Yardlong Beans - Fresh Chilies	7	11	2
2	Spring Onion - Yardlong Beans - Fresh Chilies	6	11	1
3	Celery - Fresh Chilies	6	11	1
4	Yardlong Beans - Large Mixed Tomato	7	11	2
5	Spring Onion - Yardlong Beans - Large Mixed Tomato	6	11	1
6	Celery - Large Mixed Tomato	6	11	1
7	Yardlong Beans - Chinese Kale	7	11	2
8	Spring Onion - Yardlong Beans - Chinese Kale	6	11	1
9	Celery - Chinese Kale	6	11	1
10	Spring Onion - Yardlong Beans	6	9	3

Table 21. Cultivation schedule for the period of January-December

No.	Cultivation sequence	Starting month	Ending month	Idle month
1	Dried Garlic - Watermelon	6	12	5
2	Spring Onion - Dried Garlic - Watermelon	5	12	4
3	Chinese Kale - Spring Onion - Dried Garlic - Watermelon	3	12	2
4	Celery - Spring Onion - Dried Garlic - Watermelon	2	12	1
5	Medium Garlic - Spring Onion - Dried Garlic - Watermelon	1	12	0
6	Mixed Cabbage - Dried Garlic - Watermelon	3	12	2
7	Dried Garlic - A grade Potato	6	12	5
8	Spring Onion - Dried Garlic - A grade Potato	5	12	4
9	Chinese Kale - Spring Onion - Dried Garlic - A grade Potato	3	12	2
10	Celery - Spring Onion - Dried Garlic - A grade Potato	2	12	1
11	Medium Garlic - Spring Onion - Dried Garlic - A grade Potato	1	12	0
12	Mixed Cabbage - Dried Garlic - A grade Potato	3	12	2
13	Dried Garlic - O grade Potato	6	12	5

No.	Cultivation sequence	Starting month	Ending month	Idle month
14	Spring Onion - Dried Garlic - O grade Potato	5	12	4
15	Chinese Kale - Spring Onion - Dried Garlic - O grade Potato	3	12	2
16	Celery - Spring Onion - Dried Garlic - O grade Potato	2	12	1
17	Medium Garlic - Spring Onion - Dried Garlic - O grade Potato	1	12	0
18	Mixed Cabbage - Dried Garlic - O grade Potato	3	12	2
19	Dried Garlic - Mixed Lettuce	6	11	6
20	Spring Onion - Dried Garlic - Mixed Lettuce	5	11	5
21	Chinese Kale - Spring Onion - Dried Garlic - Mixed Lettuce	3	11	3
22	Celery - Spring Onion - Dried Garlic - Mixed Lettuce	2	11	2
23	Medium Garlic - Spring Onion - Dried Garlic - Mixed Lettuce	1	11	1
24	Mixed Cabbage - Dried Garlic - Mixed Lettuce	3	11	3
25	Yardlong Beans - Large Mixed Tomato	7	11	7
26	Fresh Chilies - Yardlong Beans - Large Mixed Tomato	5	11	5
27	Chinese Kale - Fresh Chilies - Yardlong Beans - Large Mixed Tomato	3	11	3
28	Celery - Fresh Chilies - Yardlong Beans - Large Mixed Tomato	2	11	2
29	Medium Garlic - Fresh Chilies - Yardlong Beans - Large Mixed Tomato	1	11	1
30	Yardlong Beans - Shallot	7	11	7
31	Fresh Chilies - Yardlong Beans - Shallot	5	11	5
32	Chinese Kale - Fresh Chilies - Yardlong Beans - Shallot	3	11	3
33	Celery - Fresh Chilies - Yardlong Beans - Shallot	2	11	2
34	Medium Garlic - Fresh Chilies - Yardlong Beans - Shallot	1	11	1
35	Chao Phraya Lettuce - Ginger	3	11	3
36	Cucumber - Ginger	2	11	2
37	Spring Onion - Dried Garlic	5	10	6
38	Chinese Kale - Spring Onion - Dried Garlic	3	10	4
39	Celery - Spring Onion - Dried Garlic	2	10	3
40	Medium Garlic - Spring Onion - Dried Garlic	1	10	2
41	Mixed Cabbage - Dried Garlic	3	10	4
42	Fresh Chilies - Yardlong Beans	5	9	7
43	Chinese Kale - Fresh Chilies - Yardlong Beans	3	9	5
44	Celery - Fresh Chilies - Yardlong Beans	2	9	4
45	Medium Garlic - Fresh Chilies - Yardlong Beans	1	9	3
46	Chinese Kale - Fresh Chilies	3	7	7
47	Celery - Fresh Chilies	2	7	6
48	Medium Garlic - Fresh Chilies	1	7	5
49	Chinese Kale - Spring Onion	3	6	8
50	Celery - Spring Onion	2	6	7

No.	Cultivation sequence	Starting month	Ending month	Idle month
51	Medium Garlic - Spring Onion	1	6	6

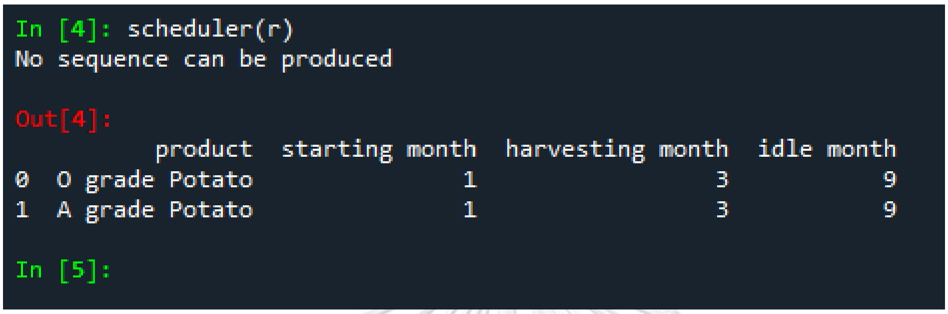
In Table 22 and Table 23, cultivation schedules for recommended commodities with additional input of location and threshold value can be observed. It is notable that the cultivation scheduling system can provide the same result as those from the original commodity recommendation. Starting month, ending month and idle month can be interpreted in a similar way as those of Table 19, Table 20 and Table 21. However, the cultivation schedule for Bangkok as cultivation location could not be processed since there is no commodity recommended. Also, there is no sequence that can be produced from recommended commodities for Chiang Rai cultivation location since the starting month and harvesting month of each commodity are the same. Figure 15 shows the output of cultivation scheduling system which states that no cultivation sequence could be produced using the input of recommended commodities for Chiang Rai location.

Table 22. Cultivation scheduling for recommendation result with Chiang Mai as location input, in period of January to December

No.	Cultivation sequence	Starting month	Ending month	Idle month
1	O grade Potato - Medium Garlic	1	12	5
2	A grade Potato - Medium Garlic	1	12	5
3	Shallot - Medium Garlic	1	12	5
4	O grade Potato - Dried Garlic	1	12	5
5	A grade Potato - Dried Garlic	1	12	5
6	Shallot - Dried Garlic	1	12	5
7	Spring Onion - Medium Garlic	2	12	6
8	Spring Onion - Dried Garlic	2	12	6

Table 23. Cultivation scheduling for recommendation result with Chiang Rai as location input, in period of January to December

No.	Cultivation sequence	Starting month	Ending month	Idle month
1	O grade Potato	1	3	9
2	A grade Potato	1	3	9



```
In [4]: scheduler(r)
No sequence can be produced

Out[4]:
          product  starting month  harvesting month  idle month
0  O grade Potato               1                 3           9
1  A grade Potato               1                 3           9

In [5]:
```

Figure 15. Scheduling system output for Chiang Rai as location input

Chapter 6

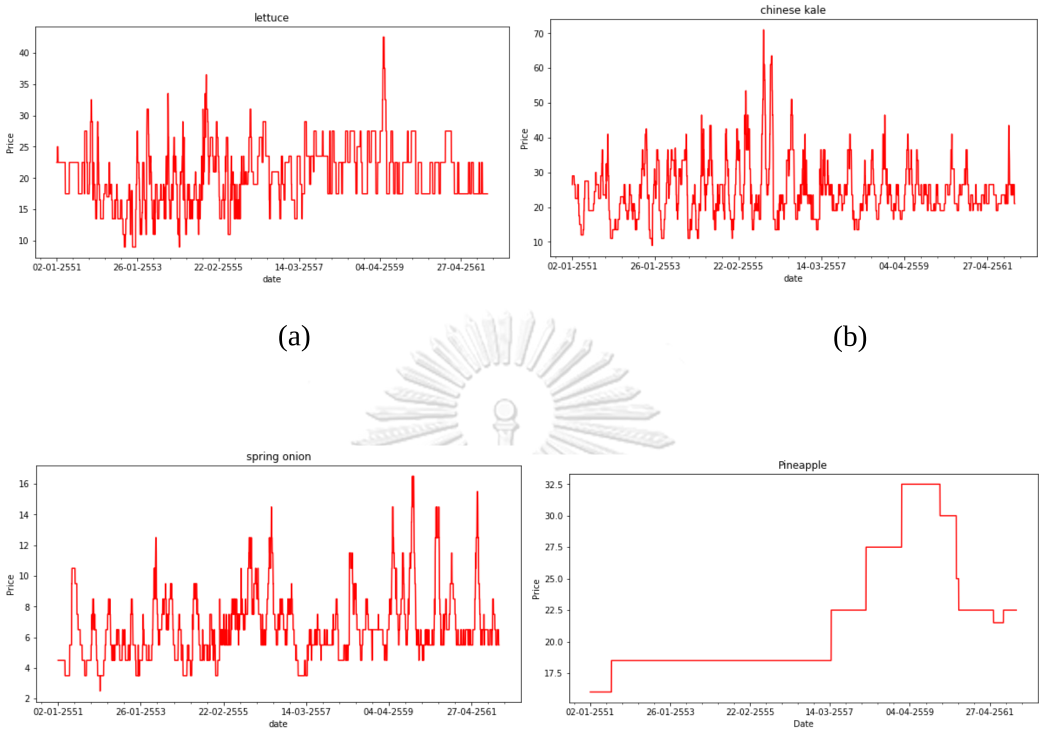
Discussions

In Multistep prediction using LSTM model section, the comparisons among different machine learning models are conducted, and the results generally show that LSTM outperforms other models for single time step prediction. It is also noticeable that LSTM has resilience in dealing with overfitting from using raw mango and Chinese kale data sets to represent small and large data sets. It can be observed from Table 7 and Table 8 that the training and testing performances of all models with a large data set, such as Chinese kale, show good generalization where testing performance is better than training performance, while testing performances of BPNN, DNN and SVR models are lower than those of the training for a smaller data set such as raw mango which is a sign of overfitting. However, overfitting problem is not experienced by LSTM model as can be seen from the results of large data sets.

In Table 10, optimization of LSTM model shows that different data sets require different window sizes to obtain best performance. Larger time window size does not affect prediction performance directly since it can be observed that only 6 out of 21 products would require time window size beyond 64. The problem might be confined to the nature of product price itself. Figure 16 displays prices of several products in the range of 10 years in conjunction to Table 10 which reveal that more fluctuated product prices require larger window sizes; as examples, Chao Phraya lettuce, Chinese kale, spring onion and pineapple would require time window sizes of 256, 64, 8 and 2.

Table 9 displays price prediction performances of LSTM for Chinese kale and raw mango with initial setting, the average values of MAE and MAPE for Chinese kale are 3.9 and 15.42, respectively, while the average values for raw mango are 8.18 and 19.44. Roughly, these average values are not much different, but the instability of prediction performances must be taken into account. For Chinese kale, the standard deviations of MAE and MAPE are 0.41

and 1.01, while those for raw mango are 8.18 and 8.65, respectively. The standard deviations of MAE and MAPE both reflect the stability of prediction performance.



(a) (b)

(c) (d)

Figure 16. Product prices for: (a) lettuce, (b) Chinese kale, (c) spring onion, and (d) pineapple

The optimized LSTM model using suggested time window size as shown in Table 11 shows that error tends to increase when step size increases in terms of both MAE and MAPE. As an example, for pineapple data set, the prediction for 22 steps ahead has MAE and MAPE values of 1.88 and 6.42, these values increase significantly to 16.32 and 68.4 for 264 steps ahead. An increase of error tends to be similar in other products except raw mango which exhibits an anomaly. The values of MAE and MAPE of raw mango for 22 steps ahead are 12.63 and 24.73, while the values are decreased to 11.49 and 20.89 for 220 steps ahead prediction.

There is no prediction performance value for 242 and 264 steps ahead due to lack of data. Prediction performances for Chao Phraya lettuce seem to be stable throughout all of the multiple steps with not much change, for example, MAE and MAPE values for 22 steps ahead are 3.53 and 17.09, while these values stay around 3.53 and 18.49 for 264 steps ahead. Evaluating from these errors in various time steps, it can be inferred that some products can only be predicted accurately for certain time steps ahead.

Further analysis shows that smaller time window size indicates seasonality of commodity, which can be observed from seasonal component of the decomposed price using STL. For an instance, in Figure 17, pineapple (pineapple) and dried garlic (dried garlic) have the same time window size of 2, it can be observed from the figure that obvious patterns are repeated between peak values of seasonal components which are marked by x tick mark.

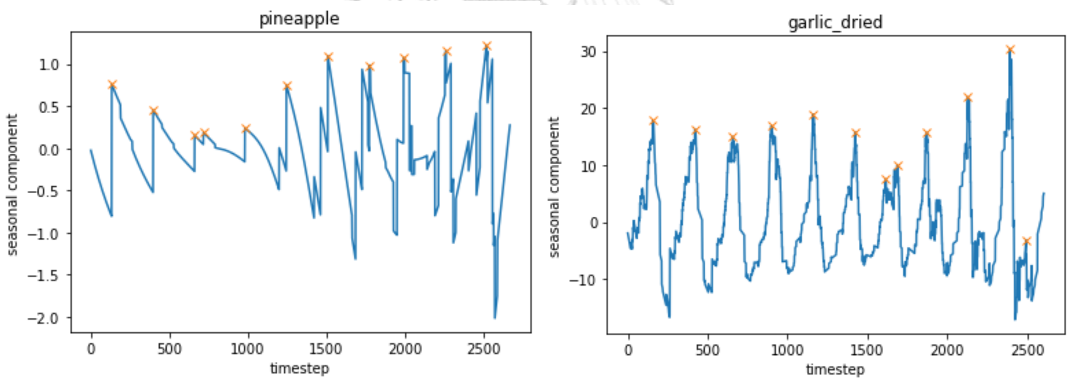


Figure 17. Seasonal component for pineapple and dried garlic

Accordingly, in Table 24 and Table 25, it can be seen that the peak seasonal component values are recurring around the same months of the year, except for pineapple whose pattern shows that there is a one month shift each year. This indicates the seasonality that is apparent in both commodity prices.

Table 24. Seasonal component peak values for dried garlic

Timestep	Date	Price	Seasonal
157	26-11-51	42	17.80475
420	05-01-53	115	16.34436
656	23-12-53	135	14.98941
901	05-01-55	105	16.84765
1160	24-01-56	88	18.92364
1421	21-02-57	123.5	15.79516
1614	08-12-57	71	7.518625
1685	24-03-58	81	9.987764
1872	30-12-58	75	15.67626
2128	19-01-60	185	21.91398
2392	16-02-61	100	30.33368
2491	16-07-61	80	-3.11033

Table 25. Seasonal component peak values for pineapple

Timestep	Date	Price	Seasonal
133	22-07-51	18.5	0.759364
397	26-08-52	18.5	0.452889
661	27-09-53	18.5	0.153946
717	17-12-53	18.5	0.18621
981	26-01-55	18.5	0.239238
1245	21-02-56	18.5	0.751491
1509	26-03-57	22.5	1.097995
1773	30-04-58	27.5	0.98573
1991	18-03-59	32.5	1.072769
2261	02-05-60	30	1.159756
2518	22-05-61	22.5	1.219274

Using the same analysis, it could be seen that larger time window size shows irregularity of seasonal component which indicates that there is no apparent pattern that recurs between the peaks. Figure 18 displays seasonal components for mixed cabbage (cabbage_mix), Chao Phraya lettuce (lettuce), and yardlong beans which require time window size of 256 to yield best prediction using LSTM model. It is clear that seasonal component of the commodities has no clear recurring pattern between the peaks of seasonal component values.

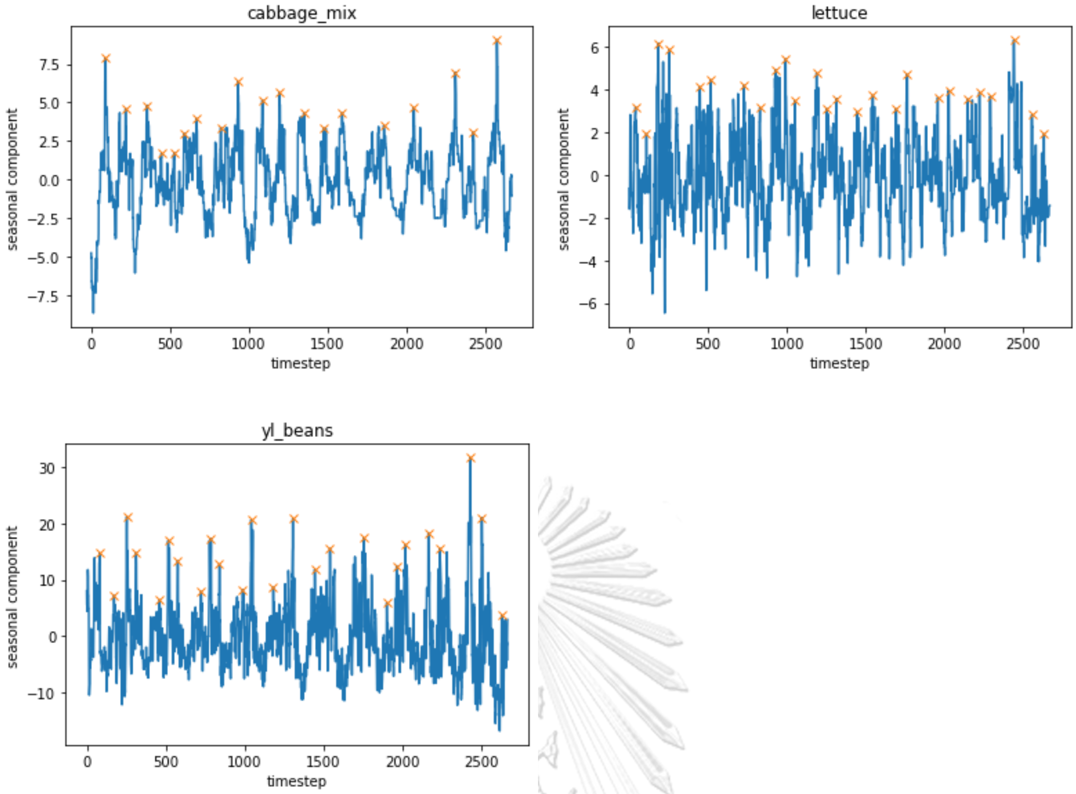


Figure 18. Seasonal component for mixed cabbage, Chao Phraya lettuce, and yardlong beans

Table 26, Table 27, and Table 28 show peak values of seasonal component for mixed cabbage, Chao Phraya lettuce, and yardlong beans, consecutively, it can be seen that the commodities have different behavior from that of previous commodities, which require a lot smaller time window size. The seasonal component peak values occur in different month of the year, although for some period it shows a regularity of this occurrence, but for some other period, the month could change drastically. It can also be seen that between peaks, the fluctuation of seasonal component value is so high that it forms no clear pattern. It could be seen that the commodities that require larger time window size do not exhibit some form of seasonality, thus LSTM model requires larger time window size to capture these changes better.

Table 26. Seasonal component peak values for mixed cabbage

Timestep	Date	Price	Seasonal
89	16-05-51	27.5	7.866447
219	21-11-51	29	4.594124
356	24-06-52	21	4.815356
445	04-11-52	14.5	1.688842
527	05-03-53	21	1.753394
592	16-06-53	26.5	2.991621
667	05-10-53	21	3.970653
825	02-06-54	23.5	3.366562
931	11-11-54	26.5	6.356072
1089	06-07-55	26.5	5.144311
1195	11-12-55	26.5	5.633818
1348	30-07-56	19	4.31795
1476	06-02-57	11	3.331807
1592	01-08-57	22.5	4.319389
1856	03-09-58	19	3.519352
2044	13-06-59	31	4.655052
2310	13-07-60	19	6.963077
2423	28-12-60	21	3.071486
2574	14-08-61	26.5	9.050292

Table 27. Seasonal component peak values for Chao Phraya lettuce

Timestep	Date	Price	Seasonal
39	27-02-51	22.5	3.175343
108	13-06-51	22.5	1.905498
184	02-10-51	27.5	6.153768
253	15-01-52	29	5.850519
448	09-11-52	19	4.103488
517	18-02-53	11	4.437375
727	04-01-54	26.5	4.183066
827	06-06-54	16.5	3.134129
926	01-11-54	36.5	4.921357
991	09-02-55	29	5.419007
1051	14-05-55	26.5	3.455804
1187	27-11-55	21	4.760329
1253	06-03-56	23.5	3.090279
1315	11-06-56	23.5	3.56149
1446	23-12-56	23.5	2.938382
1543	22-05-57	29	3.708711
1690	24-12-57	27.5	3.07509

Timestep	Date	Price	Seasonal
1759	07-04-58	22.5	4.685255
1958	01-02-59	27.5	3.613833
2028	19-05-59	42.5	3.938724
2146	09-11-59	22.5	3.52978
2222	02-03-60	27.5	3.841323
2292	16-06-60	17.5	3.664126
2441	25-01-61	27.5	6.302849
2556	16-07-61	22.5	2.797684
2631	05-11-61	22.5	1.899722

Table 28. Seasonal component peak values for yardlong beans

Timestep	Date	Price	Seasonal
81	02-05-51	52.5	14.71226
174	18-09-51	42.5	7.165034
254	16-01-52	67.5	21.22917
313	21-04-52	27.5	14.76837
461	26-11-52	42.5	6.369155
518	19-02-53	23.5	17.02495
577	25-05-53	107.5	13.33532
725	29-12-53	43.5	7.872647
788	31-03-54	71	17.29818
841	24-06-54	21	12.7263
989	07-02-55	46.5	8.224543
1043	30-04-55	73.5	20.67742
1183	21-11-55	41	8.661968
1307	30-05-56	63.5	20.86867
1446	23-12-56	51	11.82242
1542	21-05-57	46.5	15.53145
1756	01-04-58	36.5	17.42572
1903	10-11-58	31	5.849941
1965	10-02-59	71	12.46908
2020	04-05-59	111	16.20919
2167	09-12-59	46.5	18.35034
2240	28-03-60	41	15.56001
2431	11-01-61	81	31.74029
2504	01-05-61	71	21.01055
2632	06-11-61	46.5	3.650141

The use of STL-decomposed values on LSTM shows improvement on some of the selected products. For example, pineapple price prediction performance for 22 steps ahead has MAE and MAPE values of 1 and 3.98, which is an improvement from the performance of the optimized LSTM model. A significant improvement has been achieved on prediction performance for 264 steps ahead with MAE and MAPE values of 7.62 and 33.29. A similar improvement can also be seen in prediction performance of raw mango prices with MAE and MAPE of 6.12 and 12.93 for 22 steps ahead, while the values for 220 steps prediction are 5.24 and 9.53.

However, the improvement does not apply to all the selected products, some of the multistep predictions have poorer performance when compared to that of optimized LSTM model. For example, the MAE and MAPE values of 22 steps for spring onion are 1.53 and 19.29 which are higher than those of the optimized LSTM with values of 1.59 and 18.76 but for 264 steps ahead, the optimized LSTM shows an improvement with MAE and MAPE values of 1.47 and 16.69 as opposed to prior values of 1.74 and 20.52. Thus, a thorough analysis on the improvement of prediction performance using STL-LSTM model for the selected products are conducted, and the results are shown in Table 29. All values in Table 18 reflect percentage of improvement and impairment of prediction performances where the impaired prediction performances are printed in bold and given negative sign.

Table 29. Performance comparison between LSTM and STL-LSTM Model

Product Name		Prediction Step											
		22	44	66	88	110	132	154	176	198	220	242	264
Pineapple	MAE	46.81	6.726	**-14**	2.288	6.319	3.166	67.29	60.79	65.05	51.22	60.26	52.88
	MAPE	38.01	**-5.49**	**-23.4**	**-6.94**	0.213	0.84	64.79	57.8	62.69	49.07	57.97	51.33
Spring Onion	MAE	3.774	2.941	**-4.07**	**-1.64**	6.122	**-8.05**	**-2.82**	7.065	4.839	**-5.29**	9.524	15.52
	MAPE	**-2.83**	**-11**	**-18.3**	**-13.8**	**-2.75**	**-21.5**	**-11**	7.332	1.882	**-2.72**	7.806	18.66
Raw Mango	MAE	51.54	**-4.03**	4.734	**-4.38**	22.73	**-43**	**-23.8**	47.14	**-10.2**	54.4	-	-
	MAPE	47.72	**-7.41**	0.502	5.728	24.23	**-53**	**-8.94**	41.89	**-12.6**	54.38	-	-
Chinese Kale	MAE	**-22.5**	**-19.7**	11.35	**-3.64**	**-9.07**	**-11**	**-14.3**	17.23	9.511	10.19	45.18	57.32
	MAPE	**-26.9**	**-15**	16.61	**-1.97**	**-11.6**	**-19.3**	**-15.5**	14.6	11.42	9.483	46.77	58.82
Chao Phraya Lettuce	MAE	9.348	10.97	18.94	9.177	0	8.239	**-6.44**	14.15	**-14.1**	1.899	**-6.69**	0.283
	MAPE	11.64	11.51	18.38	6.625	**-0.77**	5.596	**-16.2**	11.07	**-16.8**	2.764	**-6.27**	0.757

From Table 14, Table 15, and Table 16, it can be noticed that some of the commodities have never been recommended regardless of time, or even the cultivation period, examples of these products are raw mango, ripe mango, pineapple, and tamarind. Even though these commodities are in season for the sample periods, they have never been made into recommendation because their cultivation times exceed one year, i.e., 3650, 3650, 630 and 2737.5 days for raw mango, ripe mango, pineapple, and tamarind, respectively. The commodities whose cultivation time exceeds one year are called perennial crop, whereas other commodities whose cultivation time is within one year can be considered as annual crop. Thus, it can be concluded that the algorithm is not suitable for perennial crop recommendation.

Using Bangkok, Chiang Mai and Chiang Rai as location input to recommendation system resulted in commodity recommendation that can be seen in Table 17 and Table 18, while for Bangkok location input, there is no recommendable input, as can be seen from the system output in Figure 14. This is due to the fact that Bangkok is not in the production location list that was used to calibrate the recommendation system. The recommended commodities for Chiang Mai and Chiang Rai differ since the normalized average value of Chiang Rai for respective commodities is below the threshold value of 0.2, as can be seen in Table 30. The limited availability of data impaired the ability of recommendation system to recommend different commodities. If more data can be gathered in the future, the system is expected to be able to provide variety of commodities based on production location input.

Table 30. Actual mean and normalized mean values of commodity production volumes for Chiang Rai cultivation location

Cultivation Location	Mean	Normalized Mean
Shallot	1979.375	0.034013
Garlic	2454.00	0.074086
Onion	4606.875	0.112268
Potato	16743.500	0.420323

The scheduling for sample periods can be seen in Table 19, Table 20, and Table 21. It can be noticed that the longer the period is, the more options of sequence are given. The total numbers of suggested sequences for periods of January-June, June-December, and January-December are 17,10 and 51 sequences, consecutively. Aside from giving a foresight to a user of the cultivation period for each product in a specified period, it also provides several alternative sequences with the period of idle months where the value of zero means there is no idle month in a sequence. There are two sequences with zero idle month for period of January-June and three sequences for period of January-December, while there is no sequence with zero idle month for period of June-December, instead the minimum idle month for this period is one idle month.

Cultivation scheduling using output from recommendation system with additional input resulted in different schedules for each cultivation location input; however, the limit of variability of recommended commodities causes the scheduling to be somehow limited as well. For instance, a schedule for Bangkok cultivation location cannot be produced since there is no recommended commodity to cultivate in Bangkok, and cultivation schedule for Chiang Rai cultivation location resulted in no cultivation sequence to be produced since there is only one product that was recommended by the recommender system. The availability of data can help improve the scheduling system to be able to provide cultivation schedule for various commodities.

The scheduling can give foresight to a user on planning in advance of what commodities to be cultivated in the specified period which could save time and money on cultivation activities. However, planning of cultivation activities is not as simple as assumed, different commodities may require different cares in the cultivation field, this would require a lot more effort which involves experts from other fields. This topic can be our future work that is an extension of this research.

Chapter 7
Conclusions

Several models have been used for price prediction of horticultural products and the best model is LSTM model which is then optimized by selecting the appropriate time window size for each product. The optimized model is evaluated for its ability to predict several time steps ahead and the prediction results are quite outstanding in most of the selected products, while there is an exception for some products which might result from seasonality of the products.

Further optimization of LSTM performance is proposed by using STL to decompose prices of the products into three components and using them as input data for LSTM. The optimization shows improvement in most of the time steps for each product, but also failure in some time steps. However, the use of STL-decomposed price values can improve the performance of previously optimized LSTM with appropriate time window size.

Next, the recommendation is able to provide a list of recommended commodities to cultivate and suggest when to start cultivation and harvest together with the average prices of the products. However, the recommendation is not suitable for perennial crop due to the cultivation time that exceeds a period of one year. Additional input of cultivation location provides more insight on commodities that were recommended by the system, but the availability of data seems to limit the ability of the system to recommend more commodities.

Finally, the scheduling based on recommendation is able to provide a suggested list of cultivation sequences for recommended commodities. The scheduling is expected to give foresight to a user on planning in advance for the cultivation of recommended commodities according to the selected sequence. However, the versatility of the scheduling system depends largely on the performance of recommendation system, thus, if more data are available, the cultivation scheduling system will be able to provide variety of cultivation schedules.

As a conclusion, the recommendation system still has many more interesting topics for further research and development, such as a larger scale project which involves experts from other fields might deliver a system that can fulfill the need of new farmers.

REFERENCES

[1] A. Darekar and A. A. Reddy, "Cotton Price Forecasting in Major Producing States," *Economic Affairs*, vol. 62, no. 3, pp. 373-378, 2017.
[2] L. Achal, J. G. K., P. R. K., and G. Bishal, "Modelling and Forecasting of Price Volatility: An Application of GARCH and EGARCH Models," *Agricultural Economics Research Review*, vol. 28, no. 1, pp. 73 - 82, 2015.
[3] Y. Ge and H. Wu, "Prediction of corn price fluctuation based on multiple linear regression analysis model under big data," *Neural Computing and Applications*, vol. 32, no. 22, pp. 16843-16855, 2020/11/01 2020.
[4] S. E. Hatane, "The Predictability of GARCH-Type Models on the Returns Volatility of Primary Indonesian Exported Agricultural Commodities," *JURNAL AKUNTANSI DAN KEUANGAN*, vol. 12, no. 2, pp. 87-97, 2011.
[5] W. HUANG, Z. HUANG, M. MATEI, and T. WANG, "PRICE VOLATILITY FORECAST FOR AGRICULTURAL COMMODITY FUTURES: THE ROLE OF HIGH FREQUENCY DATA," *Romanian Journal of Economic Forecasting*, vol. 4, 2012.
[6] R. Rahmawati, A. Rusgiyono, A. Hoyyi, and D. A. I. Maruddani, "GARCH-family for measuring price fluctuation risk of harvested dry grain in Pemalang district," *IOP Conf. Series: Journal of Physics: Conf. Series*, vol. 1217, no. 012092, 2019.
[7] G. M. Nasira and N. Hemageetha, "Vegetable price prediction using data mining classification technique," in *International Conference on Pattern Recognition, Informatics and Medical Engineering (PRIME-2012)*, 2012, pp. 99-102.
[8] W. Shengwei, L. Yanni, Z. Jiayu, and L. Jiajia, "Agricultural price fluctuation model based on SVR," in *2017 9th International Conference on Modelling, Identification and Control (ICMIC)*, 2017, pp. 545-550.
[9] L. Nassar, I. E. Okwuchi, M. Saad, F. Karray, and K. Ponnambalam, "Deep Learning Based Approach for Fresh Produce Market Price Prediction," in *2020 International Joint Conference on Neural Networks (IJCNN)*, 2020, pp. 1-7.
[10] C. Luo, Q. Wei, L. Zhou, J. Zhang, and S. Sun, "Prediction of Vegetable Price Based on Neural Network and Genetic Algorithm," Berlin, Heidelberg, 2011, pp. 672-681: Springer Berlin Heidelberg.
[11] D. Zhang, S. Chen, L. Liwen, and Q. Xia, "Forecasting Agricultural Commodity Prices Using Model Selection Framework With Time Series Features and Forecast Horizons," *IEEE Access*, vol. 8, pp. 28197-28209, 2020.
[12] B. X. Yong, M. R. Abdul Rahim, and A. S. Abdullah, "A Stock Market Trading System Using Deep Neural Network," in *Modeling, Design and Simulation of Systems*, Singapore, 2017, pp. 356-364: Springer Singapore.
[13] M. Dixon, D. Klabjan, and J. H. Bang, "Classification-based financial markets prediction using deep neural networks," *Algorithmic Finance*, vol. 6, pp. 67-77, 2017.
[14] N. Prakash and G. Sumaiya Farzana, "Short Term Price Forecasting of Horticultural Crops Using Long Short Term Memory Neural Network," in *Advances in Computational and Bio-Engineering*, Cham, 2020, pp. 111-118: Springer International Publishing.
[15] L. Madaan, A. Sharma, P. Khandelwal, S. Goel, P. Singla, and A. Seth, "Price forecasting & anomaly detection for agricultural commodities in India," presented at the Proceedings of the 2nd ACM SIGCAS Conference on Computing and Sustainable Societies, Accra, Ghana, 2019. Available: https://doi.org/10.1145/3314344.3332488
[16] S. Siami-Namini, N. Tavakoli, and A. S. Namin, "A Comparison of ARIMA and LSTM in Forecasting Time Series," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2018, pp. 1394-1401.

[17] T. Sujjaviriyasup, "Predicting prices of agricultural commodities in Thailand using combined approach emphasizing on data pre-processing technique," *Songklanakarin Journal of Science Technology,* vol. 40, no. 1, pp. 75-78, 2016.

[18] X. Chu, Y. Li, D. Tian, J. Feng, and W. Mu, "An optimized hybrid model based on artificial intelligence for grape price forecasting," *British Food Journal,* vol. 121, no. 12, pp. 3247-3265, 2019.

[19] D. Zhang, G. Zang, J. Li, K. Ma, and H. Liu, "Prediction of soybean price in China using QR-RBF neural network model," *Computers and Electronics in Agriculture,* vol. 154, pp. 10-17, 2018/11/01/ 2018.

[20] T. Xiong, C. Li, and Y. Bao, "Seasonal forecasting of agricultural commodity price using a hybrid STL and ELM method: Evidence from the vegetable market in China," *Neurocomputing,* vol. 275, pp. 2831-2844, 2018/01/31/ 2018.

[21] D. Mitra and R. K. Paul, "Hybrid time-series models for forecasting agricultural commodity prices," *Model Assisted Statistics and Applications,* vol. 12, pp. 255-264, 2017.

[22] G. P. Zhang, "Time series forecasting using a hybrid ARIMA and neural network model," *Neurocomputing,* vol. 50, pp. 159-175, 2003/01/01/ 2003.

[23] H. Guo, A. Woodruff, and A. Yadav, "Improving Lives of Indebted Farmers Using Deep Learning: Predicting Agricultural Produce Prices Using Convolutional Neural Networks," *Proceedings of the AAAI Conference on Artificial Intelligence,* vol. 34, no. 08, pp. 13294-13299, 04/03 2020.

[24] L. Nassar, I. E. Okwuchi, M. Saad, F. Karray, K. Ponnambalam, and P. Agrawal, "Prediction of Strawberry Yield and Farm Price Utilizing Deep Learning," in *2020 International Joint Conference on Neural Networks (IJCNN)*, 2020, pp. 1-7.

[25] B. Wang *et al.*, "Research on Hybrid Model of Garlic Short-term Price Forecasting based on Big Data," *Computers, Materials \& Continua,* vol. 57, no. 2, pp. 283--296, 2018.

[26] L. Wang, J. Feng, X. Sui, X. Chu, and W. Mu, "Agricultural product price forecasting methods: research advances and trend," *British Food Journal,* vol. 122, no. 7, pp. 2121-2138, 2020.

[27] Č. Rozman, A. Škraba, K. Pažek, and M. Kljajić, "The Development of Sugar Beet Production and Processing Simulation Model – a System Dynamics Approach to Support Decision-Making Processes," *Organizacija,* vol. 47, no. 2, pp. 99-105, 2014.

[28] C. Daydé, S. Couture, F. Garcia, and R. Martin-Clouaire, "Investigating Operational Decision-Making in Agriculture," in *7th International Congress on Environmental Modelling and Software Proceeding*, San Diego, California, 2014, vol. 21.

[29] A. S. Mase and L. S. Prokopy, "Unrealized Potential: A Review of Perceptions and Use of Weather and Climate Information in Agricultural Decision Making," (in English), *Weather, Climate, and Society,* vol. 6, no. 1, pp. 47-61, 01 Jan. 2014 2014.

[30] J. Han, M. Kamber, and J. Pei, "3 - Data Preprocessing," in *Data Mining (Third Edition)*, J. Han, M. Kamber, and J. Pei, Eds. Boston: Morgan Kaufmann, 2012, pp. 83-124.

[31] R. B. Cleveland, W. S. Cleveland, J. E. McRae, and I. Terpenning, "STL : A Seasonal-Trend Decomposition Procedure Based on Loess," *Journal of Official Statistics,* vol. 6, no. 1, pp. 3-73, 1990.

[32] D. Jin, H. Yin, Y. Gu, and S. J. Yoo, "Forecasting of Vegetable Prices using STL-LSTM Method," in *Proceedings of 2019 6th International Conference on Systems and Informatics (ICSAI)*, Shanghai, China, 2020: IEEE.

[33] H. Drucker, C. J. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," *Neural Information Processing Systems,* vol. 9, pp. 155-161, 1997.

[34] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," presented at the Proceedings of the fifth annual workshop on Computational learning theory, Pittsburgh, Pennsylvania, USA, 1992. Available: https://doi.org/10.1145/130385.130401

[35] V. N. Vapnik, "The Nature of Statistical Learning Theory," ed: Springer New York, 1995.

[36] V. Cherkassky and Y. Ma, "Practical selection of SVM parameters and noise estimation for SVM regression," *Neural Networks*, vol. 17, no. 1, pp. 113-126, 2004/01/01/ 2004.

[37] S. P. Das and S. Padhy, "A novel hybrid model using teaching–learning-based optimization and a support vector machine for commodity futures index forecasting," *International Journal of Machine Learning and Cybernetics*, vol. 9, no. 1, pp. 97-111, 2018.

[38] V. Cherkassky and F. M. Mulier, *Learning from Data: Concepts, Theory, and Methods*. Wiley, 2007.

[39] A. Patle and D. S. Chouhan, "SVM kernel functions for classification," in *2013 International Conference on Advances in Technology and Engineering (ICATE)*, 2013, pp. 1-9.

[40] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533-536, 1986/10/01 1986.

[41] T. Hastie, R. Tibshirani, and J. Friedman, "The Elements of Statistical Learning," in *Springer Series in Statistics*, ed: Springer New York, 2009.

[42] S.-i. Amari, "Backpropagation and stochastic gradient descent method," *Neurocomputing*, vol. 5, no. 4, pp. 185-196, 1993/06/01/ 1993.

[43] J. Duchi, E. Hazan, and Y. Singer, "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization," *J. Mach. Learn. Res.*, vol. 12, no. null, pp. 2121–2159, 2011.

[44] D. Gutierrez. (2017, March 5th). *RMSprop Optimization Algorithm for Gradient Descent with Neural Networks*. Available: https://insidebigdata.com/2017/09/24/rmsprop-optimization-algorithm-gradient-descent-neural-networks/

[45] M. D. Zeiler. Adadelta: An Adaptive Learning Rate Method [Online]. Available: https://arxiv.org/abs/1212.5701

[46] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proceedings of 3rd International Conference on Learning Representations 2015*, California, USA, 2015.

[47] H. Jindal, N. Sardana, and R. Mehta, "Analyzing Performance of Deep Learning Techniques for Web Navigation Prediction," *Procedia Computer Science*, vol. 167, pp. 1739-1748, 2020/01/01/ 2020.

[48] "Appendix A - Artificial neural networks," in *Neural Networks Modeling and Control*, E. N. Sanchez, J. D. Rios, A. Y. Alanis, N. Arana-Daniel, and C. Lopez-Franco, Eds.: Academic Press, 2020, pp. 117-124.

[49] R. Dechter, "Learning while searching in constraint-satisfaction-problems," presented at the Proceedings of the Fifth AAAI National Conference on Artificial Intelligence, Philadelphia, Pennsylvania, 1986.

[50] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals and Systems*, vol. 2, no. 4, pp. 303-314, 1989/12/01 1989.

[51] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Networks*, vol. 4, no. 2, pp. 251-257, 1991/01/01/ 1991.

[52] H. Chung, S. J. Lee, and J. G. Park, "Deep neural network using trainable activation functions," in *2016 International Joint Conference on Neural Networks (IJCNN)*, 2016, pp. 348-352: IEEE.

[53] Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang, "The expressive power of neural networks: a view from the width," presented at the Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, California, USA, 2017.

[54] Y. Bengio, "Learning Deep Architectures for AI," *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, 2009.

[55] J. Feng and S. Lu, "Performance Analysis of Various Activation Functions in Artificial Neural Networks," *Journal of Physics: Conference Series*, vol. 1237, p. 022030, 2019/06 2019.

[56] S. Theodoridis, "Chapter 18 - Neural Networks and Deep Learning," in *Machine Learning (Second Edition)*, S. Theodoridis, Ed.: Academic Press, 2020, pp. 901-1038.

[57] T. M. Navamani, "Chapter 7 - Efficient Deep Learning Approaches for Health Informatics," in *Deep Learning and Parallel Computing Environment for Bioengineering Systems*, A. K. Sangaiah, Ed.: Academic Press, 2019, pp. 123-137.

[58] R. DiPietro and G. D. Hager, "Chapter 21 - Deep learning: RNNs and LSTM," in *Handbook of Medical Image Computing and Computer Assisted Intervention*, S. K. Zhou, D. Rueckert, and G. Fichtinger, Eds.: Academic Press, 2020, pp. 503-519.

[59] C. Olah. (2015, March 3rd). *Understanding LSTM Networks*. Available: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[60] H. Abbasimehr, M. Shabani, and M. Yousefi, "An optimized model using LSTM network for demand forecasting," *Computers & Industrial Engineering*, vol. 143, p. 106435, 2020/05/01/ 2020.

[61] G. H. T. Ribeiro, P. S. G. d. M. Neto, G. D. C. Cavalcanti, and I. R. Tsang, "Lag selection for time series forecasting using Particle Swarm Optimization," in *The 2011 International Joint Conference on Neural Networks*, 2011, pp. 2437-2444.

[62] H. Chung and K. S. Shin, "Genetic Algorithm-Optimized Long Short-Term Memory Network for Stock Market Prediction," *Sustainability*, vol. 10, no. 10, 2018.

[63] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85-117, 2015/01/01/ 2015.

[64] F. A. Gers, D. Eck, and J. Schmidhuber, "Applying LSTM to Time Series Predictable through Time-Window Approaches," Berlin, Heidelberg, 2001, pp. 669-676: Springer Berlin Heidelberg.

[65] D. Shah, W. Campbell, and F. H. Zulkernine, "A Comparative Study of LSTM and DNN for Stock Market Forecasting," in *2018 IEEE International Conference on Big Data (Big Data)*, 2018, pp. 4148-4155.

[66] F. Harrou, Y. Sun, A. S. Hering, M. Madakyaru, and A. Dairi, "Chapter 7 - Unsupervised recurrent deep learning scheme for process monitoring," in *Statistical Process Monitoring Using Advanced Data-Driven and Deep Learning Approaches*, F. Harrou, Y. Sun, A. S. Hering, M. Madakyaru, and A. Dairi, Eds.: Elsevier, 2021, pp. 225-253.

[67] J. Chebeir, H. Asala, V. Manee, I. Gupta, and J. Romagnoli, "Comprehensive Investment Framework for Optimal Exploitation of Liquid-Rich Unconventional Reservoirs and Natural Gas Distribution," in *Computer Aided Chemical Engineering*, vol. 44, M. R. Eden, M. G. Ierapetritou, and G. P. Towler, Eds.: Elsevier, 2018, pp. 961-966.

[68] R. H. Sprague, "A Framework for the Development of Decision Support Systems," *MIS Quarterly*, vol. 4, no. 4, pp. 1-26, 1980.

[69] H. J. Watson, "Revisiting ralph sprague's framework for developing decision support systems," *Communications of the Association for Information Systems*, vol. 42, pp. 363 - 385, 2018.

[70] F. G. Filip, "Decision support and control for large-scale complex systems," *Annual Reviews in Control*, vol. 32, no. 1, pp. 61-70, 2008/04/01/ 2008.

[71] R. Klashner and S. Sabet, "A DSS Design Model for complex problems: Lessons from mission critical infrastructure," *Decision Support Systems*, vol. 43, no. 3, pp. 990-1013, 2007/04/01/ 2007.

[72] F. Yang, K. Wang, Y. Han, and Z. Qiao, "A Cloud-Based Digital Farm Management System for Vegetable Production Process Management and Quality Traceability," *Sustainability*, vol. 10, no. 11, p. 4007, 2018.

[73] keras-team. (23 February). *Deep Learning for humans*. Available: https://github.com/keras-team/keras

[74] Department of Internal Trade Thailand. (10 July). *Retail And Wholesale Prices Of Agricultural Products*. Available: https://pricelist.dit.go.th/main_price.php

[75] W. McKinney, "Data Structures for Statistical Computing in Python," in *Scipy 2010 Python for Scientific Computing Conference*, Austin, Texas, 2010.

[76] The pandas development team, "pandas-dev/pandas: Pandas," ed: Zenodo, 2020.

[77] I. Stančin and A. Jović, "An overview and comparison of free Python libraries for data mining and big data analysis," in *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2019, pp. 977-982.

[78] L. A. Shalabi, Z. Shaaban, and B. Kasasbeh, "Data Mining: A Preprocessing Engine," *Journal of Computer Science* vol. 2, no. 9, pp. 735-739,, 2006.

[79] Seabold, Skipper, and J. Perktold, "statsmodels: Econometric and statistical modeling with python," in *Proceedings of the 9th Python in Science Conference*, 2010.

[80] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825-2830, 2011.

[81] J. D. Hunter, "Matplotlib: A 2D Graphics Environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90-95, 2007.

[82] C. R. Harris *et al.*, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357-362, 2020/09/01 2020.

[83] Office of Agricultural Economics. (2021, 10 Jun 2021). *Agricultural Statistics of Thailand*. Available: http://www.oae.go.th/view/1/%E0%B9%80%E0%B8%AD%E0%B8%81%E0%B8%AA%E0%B8%B2%E0%B8%A3%E0%B9%80%E0%B8%9C%E0%B8%A2%E0%B9%81%E0%B8%9E%E0%B8%A3%E0%B9%88%E0%B8%A2%E0%B9%89%E0%B8%AD%E0%B8%99%E0%B8%AB%E0%B8%A5%E0%B8%B1%E0%B8%87/TH-TH

[84] L. Peng, S. Liu, R. Liu, and L. Wang, "Effective long short-term memory with differential evolution algorithm for electricity price prediction," *Energy*, vol. 162, pp. 1301-1314, 2018/11/01/ 2018.

[85] B. Zhang, H. Zhang, G. Zhao, and J. Lian, "Constructing a PM2.5 concentration prediction model by combining auto-encoder with Bi-LSTM neural networks," *Environmental Modelling & Software*, vol. 124, p. 104600, 2020/02/01/ 2020.

[86] C. Xiao, N. Chen, C. Hue, K. Wang, J. Gong, and Z. Chen, "Short and mid-term sea surface temperature prediction using time-series satellite data and LSTM-AdaBoost combination approach," *Remote Sensing of Environment* vol. 233, 2019.

VITA

NAME	Lukman Adlin Harahap
DATE OF BIRTH	9 June 1982
PLACE OF BIRTH	Medan
INSTITUTIONS ATTENDED	Faculty of Agriculture, University of Sumatera Utara, Indonesia (Bachelor) Faculty of Agricultural Engineering, Bogor Institute of Agriculture, Indonesia (Master)
HOME ADDRESS	Monard Mansion 104/18 Soi Pracha Songkhro 23, Pracha Songkhro Road, Din Daeng, Bangkok 10400 Telephone Number 0647613665