Forecasting Agricultural-Problems' Trends using Deep Learning

Samarth Godara[a,b,*], Durga Toshniwal[b]

[a]Division of Computer Applications, Indian Agricultural Statistics Research Institute, New Delhi, India
[b]Department of Computer Science and Engineering, Indian Institute of Technology Roorkee, Uttarakhand, India

Abstract

Agriculture plays a vital role in the global economy, provides the primary source of food, livelihood, and employment to the populations. Therefore, we are in a continuous quest to develop innovative approaches with the intention to boost agricultural productivity. For planning agricultural-policies, organizing farmers' training, gaining agricultural product-based market insights, and executing strategic marketing actions, government officials, policymakers, and private organizations need to gain awareness regarding farmers' problems. However, today, there is a need of such a robust system that can be used to collect and analyze Spatio-temporal information regarding the problems faced by farmers on a large scale. This article describes a new approach that uses high-end artificial intelligence-based techniques to forecast the trends in agricultural problems using farmers' helpline data. The dataset utilized in this work is accumulated from the "Kisan Call Center", a farmers' helpline center administered by the Ministry of Agriculture, Government of India, from March 2013 to March 2020. Moreover, we take data corresponding to the top-five rice-producing states of India (Uttar Pradesh, Punjab, Bihar, West Bengal, and Andhra Pradesh) as case studies to inspect the performances of the proposed framework with four different forecasting periods (1, 7, 15, and 30-days forecasting). Later, we compare the forecasting potential of four different Machine Learning and Deep Learning-based forecasting techniques, i.e., Support Vector Regression, Multi-layer Perceptron, Long Short-Term Memory Networks, and Gated Recurrent Units using three different performance measures (Mean Squared Error, Mean Absolute Error, and Correlation Coefficient). From the experimental results, we found that the proposed framework is useful for forecasting trends in farmers' problems, furthermore, we identify various other potential applications of the presented work. Finally, we conclude with some possible future developments in the proposed approach.

Keywords: Farmers' Query, Machine Learning in Agriculture, Deep Learning in Agriculture, Forecasting agricultural-problems' trends, Kisan Call Center, Deep Learning based forecasting

1. Introduction

Today, the growing world population has become a notable concern worldwide due to the deficiency in agricultural resources. Researchers estimate that it is required to enhance the world's agricultural yield by 1.5 times in the following thirty years ([1]). Nonetheless, a considerable fraction of the population in developing countries like India is entirely reliant on agriculture; this makes it a vital component for the nation's economic development. Figure 1 gives statistics of the worldwide production (tonnes/hectare) of Maize, Rice, Wheat and Soy since 1950 ([2]). Although it seems that global crop-production has been consistently increasing for the past few decades, it is estimated that the production-requirement will surpass crop-production in the future. This requires the policymakers

*Corresponding author
Email addresses: sgodara@cs.iitr.ac.in (Samarth Godara), durgatoshniwal@gmail.com (Durga Toshniwal)

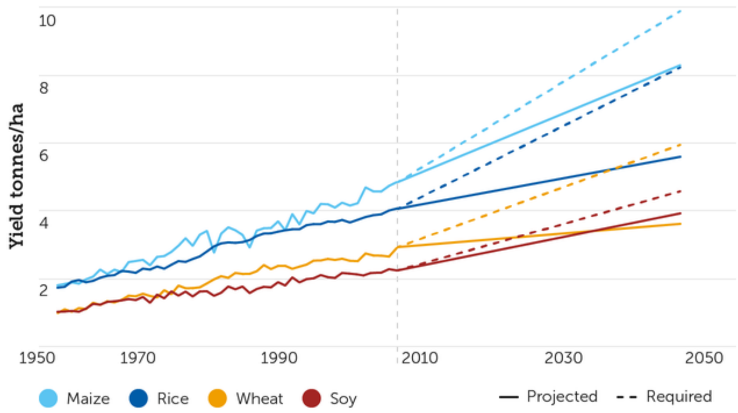


Figure 1: Global crop production with estimation of required production.

and the agriculture-product based companies to introduce better agri-policies and agri-technologies to enhance the crop production.

For the betterment of the current agricultural scenario, it is required to have advanced tools and techniques to analyze the agricultural problems faced by the farmers. A fact to be noticed regarding the agriculture-based analysis is that it is primarily dependent on the crop-seasonality. Generally, farmers face problems (or they tend to buy or sell agricultural products) according to the crop season. Hence temporal-analysis is an essential aspect in gaining insights for both, the policymakers and the agri-product based companies.

The traditional methods (manual surveys, personal/group interviews) used to gain such inferences on a large scale are not only expensive to execute, but these also may take days if not weeks to complete and extract insights. Nonetheless, another challenging characteristic of this kind of analysis is to obtain updated and authentic information regarding farmers' problems. Considering the case where the policymakers are unable to get a reasonable estimate of the farmers' problems, it can possibly lead to the introduction of inaccurate policies and can result in an unoptimized usage of agricultural resources.

In a country like India, with around 118 million farmers, it is a difficult task to obtain information about their current problems altogether. Hence, such countries are still striving to use the technological developments to obtain updates regarding the difficulties encountered by the farmers. Due to rural areas' lack of technological progressions, collecting information regarding farmers' problems is mostly done manually, either through surveys ([3]) or campaigns ([4]). These methods do not only produce out-dated results, but the gathered data can also lead to incorrect decisions due to the tendency of human flaws.

As an alternative to the existing methods of collecting information and analyzing the farmers' problems, we propose a potential solution by using their telephonic queries' records. We accumulate the call-record data from the Kisan Call Center's (KCC) official website, a free helpline service for Indian farmers ([5]). The call accounts of this service are managed by the "Kisaan Knowledge Management System" (KKMS), working under the Ministry of Agriculture, Government of India. In this study, the data-insights are extracted in the form of forecasted problems' trends corresponding to the target-crop. As India is one of the leading producers of rice, we use the same as our target-crop in the presented work. Furthermore, to analyze the trend in rice-related problems and to compare the performances of various forecasting techniques, we chose the five top-most rice-producing states of India as five different case studies ([6]). Farmer's query-data collected in this research is from March 2013 to March 2020, corresponding to the states, Uttar Pradesh, Punjab, Bihar, West Bengal, and Andhra Pradesh. For our objective, we chose the four most extensively used Machine learning (ML) and Deep Learning (DL) techniques, (i.e., Support Vector Regression (SVR), Multi-layer Perceptron (MLP), Long Short-Term Memory Networks (LSTM), and Gated Recurrent Units (GRU)) to develop the forecasting models. Additionally, to compare their performances over four different forecasting periods (1, 7, 15, and 30-days forecasting), three metrics are taken into account: Mean Squared Error, Mean Absolute Error, and Correlation Coefficient. Finally, all the developed models are examined against the testing data, and models are ranked according to the obtained error rates. This helped in an in-depth understanding of the forecasting models' performance and ensured their reliability. A few of the notable research contributions of the proposed work are as follows:

- Introducing the concept of utilizing farmers' helpline data to forecast the trend in agricultural-problems.
- Extracting the telephonic records to formulate a time-series problem for forecasting spatial-specific agricultural trends.
- Five different case studies are examined to compare the models' performances.
- Two ML techniques (SVR and MLP) and two DL techniques (LSTM and GRU) are compared in the presented work.
- Models developed and tested with four different forecasting periods (1, 7, 15, and 30-days forecasting) using three different performance metrics.

The rest of this article is organized as follows: Section 2 delivers the related literature. Section 3 explains the prerequisite concepts in detail. The full description of the practiced methodology is given in Section 4. Experiments and results are addressed and elaborated in Section 5. Discussion and conclusion about the presented research are demonstrated in Sections 6 and 7, respectively.

2. Literature Survey

Presently, about 70% of the Indian population still depends primarily on agriculture ([7]). Therefore, monitoring and investigation of improvements in agriculture has continuously been a focus for the Indian Government. In India, the Ministry of Statistics and Programme Implementation is in charge of designing and conducting surveys in various sectors including education, healthcare, and agriculture. Moreover, the countrywide surveys' monitoring is administered by the National Sample Survey Office (NSSO) ([8]). Although it has been proved that the computer-administered surveys have many benefits ([9]), NSSO still utilizes the process of paper-based surveys to gather countrywide information. Besides this, the government surveys are mostly designed to gain information concerning only the agricultural yields, and not the agricultural problems. Similarly, to gain market insights, the traditional methods of executing market research include focus group interaction, depth interview (face to face, telephonic), questionnaire-based survey (manual, online), and public behavior observation ([10]). Nevertheless, in our literature review we found that there are only a few studies that concentrate on the analysis of farmers' problems.

In other parts of the world, researchers have developed various systems to gather information regarding the agricultural practices on a large scale ([11], [12], [13]). Still, since most of these methods utilize smart devices with a necessary connection to the internet, it is not feasible to execute them in developing nations. Several other types of research use IoT technology to accumulate the data instantly from the fields ([14], [15]). However, drawback of such systems is that it requires the investment of new devices, which makes it an infeasible alternative for gaining data of a vast area.

The Indian government has always desired to direct its resources to harness ICT's potential in Agriculture ([5]). A step in that direction, the scheme of KCC was introduced in January 2004 by the Ministry of Agriculture and Farmers Welfare, Government of India. The scheme intends to solve farmers' questions from all over the country through phone calls, in the farmers' regional language. The launched helpline assistance is accessible in 22 local languages, every day of the week, from 6.00 AM to 10.00 PM. The call centers work in 21 diverse locations in India, reaching every State and Union Territory (UT) of the country. Farmers require to dial the toll-free number 1800-180-1551 to reach the helpline center. Queries made by the farmers are answered by the KCC agents known as Farm Tele Advisors (FTAs). FTAs are the graduates or above (i.e., Post-Graduates or Doctorates) in agriculture or allied sectors, including Bio-Technology, Animal Husbandry, Fisheries, Horticulture, Aquaculture, Poultry, Bee-Keeping, Agricultural Engineering, Agricultural Marketing, Sericulture, etc. Every phone call made to the KCC centers is logged by the FTAs with the data regarding the farmer's location, time and date of query, the complete question and its response in textual form, query category, crop, etc. Query call reports are managed and made publicly accessible by the Ministry of Agriculture, Government of India, through the KKMS. The publicly-available reports can be downloaded from the KKMS servers in .json file format; here, individual files are stored every month, corresponding to each state/UT.

Earlier, many researchers have used information from helpline centers to analyze and obtain knowledge regarding public behavior. [16] developed an algorithm based on the fuzzy decision tree (FDT), in 2010, to obtain the useful information and make appropriate decisions automatically from the voluminous telephone call detail records. Later, [17] analyzed data from 111 helpline centers affiliated with "Child Helpline International" to discover the differences between the reasons for children's phone-calls from different parts of the world, and the noticeable shifts in the 10-year period. In 2018, through the analysis of the Hong Kong suicide-helpline records, [18] developed an early suicide warning system based on the characteristics of men who used the suicide helpline services. Moreover, [19] used the data from "Perinatal Anxiety & Depression Australia National Helpline" to describe the characteristics of women having severe mental health issues.

As the call records from the farmers' helpline centers can point to several valuable insights, with the use of different data mining and ML procedures, researchers have already begun deriving knowledge from it. Various studies have been done examining the KCC's impact ([20], [21], [22], [23]) and farmers' attitude towards it ([24], [25], [26]). [27] analyzed three years of data (2015-17) from the farmers' helpline centers using Hadoop based MapReduce techniques to extract insights such as the crops that have been investigated the most by Indian farmers and the time during which the highest number of calls are made. Moreover, in the study, the researchers clustered similar inquiries using Natural Language Processing (NLP) to know the type of questions that are frequently asked by the farmers. In other work, [28] used NLP on a comparable dataset to develop a model to produce query responses for the textual-formatted queries. The scholars utilized Latent Dirichlet allocation (LDA) and Latent Semantic Indexing (LSI) in the pipeline to the term-frequency-inverse document frequency (TF-IDF) model for this objective. A significant weakness of such a system is that it can only process textual information, whereas the KCC is an audio phone-call helpline service. Therefore, it is not reasonable to integrate such models with the KCC as they are. A comparable work was produced by [29], where the researchers developed a model to get the query's information based on the relationship between the sentences of the questions and later obtained the best workable response according to the associations. In this study, TF-IDF was adopted to find related queries. [30] proposed a model to extract insights utilizing six years of the farmers' helpline data (2013-19). The gained inference was in the form of association rules, extracted using the Apriori algorithm. Later, the insights were filtered out using the TOPSIS technique.

Although most of the existing studies do obtain knowledge from the helpline data, those researches primarily concentrate on improving the current KCC model. Only a few of the researches provide insights to be used by the agriculture policymakers. Nevertheless, none of the existing works have examined the data from the time-series perspective. Our intention with the present work is to introduce a framework that can benefit agricultural policymakers and market analysts to obtain essential insights in terms of Spatio-temporal trends in agricultural-problems corresponding to the target crops. The deduced insights are useful in strategist planning of resource utilization in this sector. For our objective, we use four of the widely used ML and DL techniques, SVR, MLP ([31]), LSTM ([32]), and GRU ([33]). Researchers have been using these methods in various fields including healthcare ([34], [35]), financial forecasting ([34], [36]), energy demand forecasting ([37], [38]), NLP ([39], [40]) and environmental forecastings ([34], [41], [42]).

3. Prerequisites

3.1. Support Vector Regression

Support Vector Machines (SVMs) were presented by Cortes and Vapnik ([43]) as a set of supervised learning techniques for classification and regression. Since then, SVMs have been considered one of the most powerful tools in ML as SVMs have lower susceptibility to local minima and higher resistance to increased model complexity.

Vapnik's ε-tube support vector regression (ε-SVR) is the standard formulation of SVM regression ([44]). ε-SVR performs linear regression (equation (1)) in the high-dimensional feature space generated by a kernel function using ε-insensitive loss. Simultaneously, it tries to decrease the model's complexity by minimizing its coefficients, i.e., weight vector method (equation (3)). The goal of ε-SVR is to obtain a function $f(x)$ (equation (1)) that has a maximum deviation from the training data, and yet, it should be as flat as possible.

$$f(x) = \sum_{n=1}^{l} w_i K(x) + b \tag{1}$$

Where l denotes the number of the training data samples, x is the p-dimensional input vector, and K is the kernel function. Here, K is used to transform the input space into a high-dimensional feature space. This transformation makes it possible to apply the linear SVR algorithm on the non-linear input ([45]). In this study, we used the Radial Basis function (RBF) (equation (2)) as a kernel function.

$$K(x) = e^{-\gamma||x-t||^2} \tag{2}$$

Here, γ is a constant parameter, and t is the center point. The SVR algorithm seeks a small weight vector w, the primary factor behind flatness of function $f(x)$. It is done simply by minimizing the following risk function :

$$\min_{w,b,\xi_i,\xi_i^*} \frac{1}{2}||w||^2 + C\sum_{n=1}^{N}(\xi_i + \xi_i^*) \tag{3}$$

$$
\textit{subject to} \begin{cases} y_i - \langle w, K(x_i) \rangle + b \le \varepsilon + \xi_i^* \\ \langle w, K(x_i) \rangle + b - y_i \le \varepsilon + \xi_i \\ \xi_i^*, \xi_i \ge 0 \end{cases}
$$

Where ξ_i is the lower training error (ξ_i^* is the upper) subject to the ε-insensitive tube $|y - \langle w, K(x) \rangle + b)| \le \varepsilon$, and C is the regularization constant.

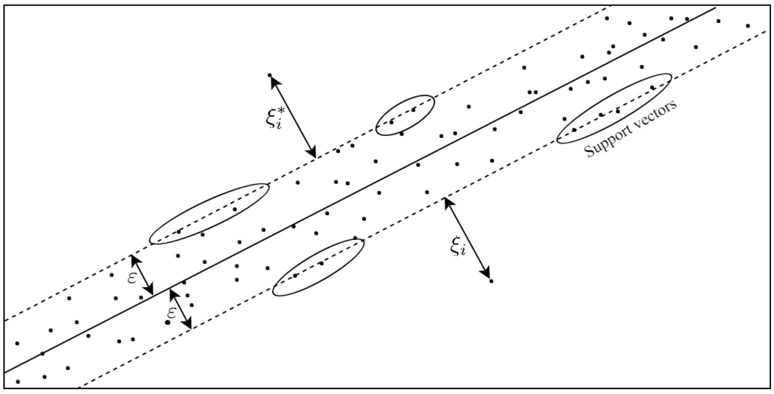


Figure 2: A schematic representation of the ε-tube (linear SVR).

As shown in the figure 2, the value of ε defines the radius of the tube; a smaller value shows a higher sensitivity towards errors, and it also affects the number of support vectors and the solution sparsity. The ε-insensitive loss function used by SVR penalizes predictions that are farther than ε from the desired output. To adopt a soft-margin approach, slack variables ξ_i, ξ_i^* are utilized to guard against outlier data points. These variables decide the number of points that are allowed outside the ε-tube. The vectors x_i corresponding to non-zero w_i are called "support vectors". This SVR minimization problem is then transformed to the dual optimization problem to calculate the weight w. The transformed problem is a constrained quadratic problem and can be easily solved by quadratic programming ([46]).

3.2. Multi-layer Perceptron

The theory of Artificial Neural Networks (ANNs) was initially given in 1943 ([47]). Later, the first practical ANN was introduced in 1958 ([48]). A standard MLP consists of a network of perceptrons (linear classifiers) organized in layers as shown in 3 ([49]).

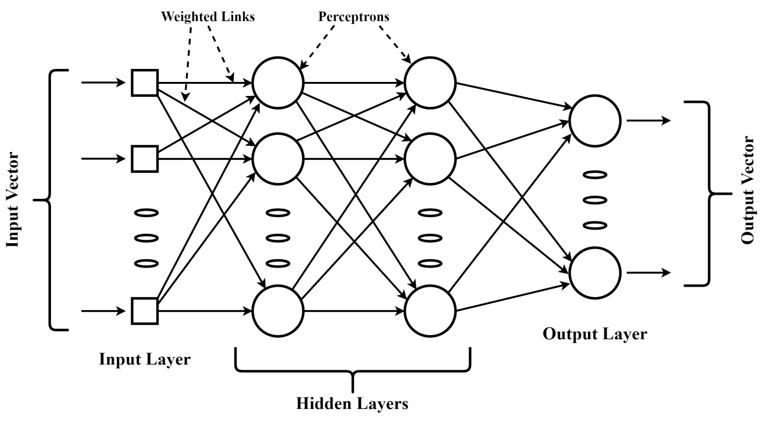


Figure 3: Architecture of an MLP with two hidden layers.

There are three types of layers present in any MLP: an input layer, one or more hidden layers, and an output layer. The hidden layers process and transfer the information from the input layer to the output layer; no calculation is involved in the input layer.

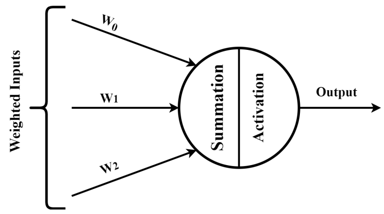


Figure 4: A schematic representation of a Perceptron.

Every perceptron j present in the network sums its input signals x_i after multiplying them by their respective connection weights w_{ji}. It then applies an activation function to the resultant and passes the output to the next layer as shown in figure 4. The working of a perceptron can be mathematically described by equation (4).

$$y_j = \psi\left(\sum_{i=1}^{u} w_{ji}x_i\right) \tag{4}$$

Where ψ is the activation function utilizing the weighted summations of the inputs and u represents the number of nodes in the previous layer. Some of the common activation functions used by the model developers are; sigmoid function (equation (5)) , hyperbolic tangent function ([50]) and ReLU ([51]). In this study, the sigmoid function (equation (5)) is used as activation function in the MLPs.

$$\psi(x) = \frac{1}{1+e^{-x}} \tag{5}$$

At first, each layer of the MLP is initialized with random weights. The output of the model is then optimized with the help of a loss function (equation (6)), which is used to keep track of the model's error rate during the training process. A commonly used loss function in the training process of MLP is the "mean squared error" loss function, mathematical notation is given in equation (6).

$$E = \frac{1}{l}\sum_{j=1}^{l}(y_{dj} - y_j)^2 \tag{6}$$

Where y_{dj} is the desired output, y_j is the actual output, and l is the number of samples that have been tested. The process of feeding the input vector in the first (input) layer of the model, and getting the output from the last (output) layer after calculating the results is called "forward propagation". In this process, the information is transferred from the input layer towards the output layer through the hidden layers.

After the forward propagation, weights of each layer are tuned using the derivative of the error rate corresponding to the layer's output. This process is called "backward propagation", as the information (error rate) is transferred from the output layer back towards the input layer. The weights of each layer are updated according to the equation (7).

$$W \leftarrow W - \eta\frac{\partial E}{\partial W} \tag{7}$$

Where, η is the learning rate. The weights in the MLP-based models are tuned through iterations of the forward and backward propagation. After the model is trained, the prediction is done using only the forward propagation.

3.3. Long Short-Term Memory (LSTM) Network

LSTM Network was introduced by Hochreiter and Schmidhuber in 1997 ([52]). Unlike the standard perceptrons, LSTM units have a cyclic construction and can pass information not only to the next layer but also among units of the same layer. When the input data is a time series, the model can be expanded into a line of connected LSTM units as shown in figure 5.

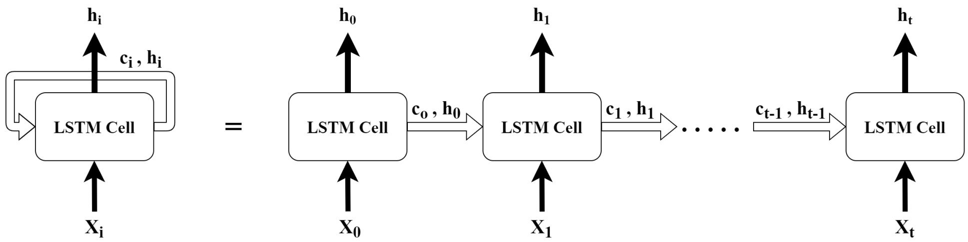


Figure 5: Sequential processing in an LSTM Network.

In LSTMs, the ability to learn long-term dependencies and memorize information for prolonged periods is acquired by the use of gates. A gate of an LSTM unit represents a set of matrix operations, which includes individual weights and biases. The gating mechanism of the unit includes mainly three types of gates: input, forget, and output gate (figure 6). The unit also includes state boxes that receive the inputs from the previous unit in the sequence. Each unit transfers two states to the next unit, i.e., the cell state (c_t) and the hidden state (h_t). Following points describe the working of each gate in detail :

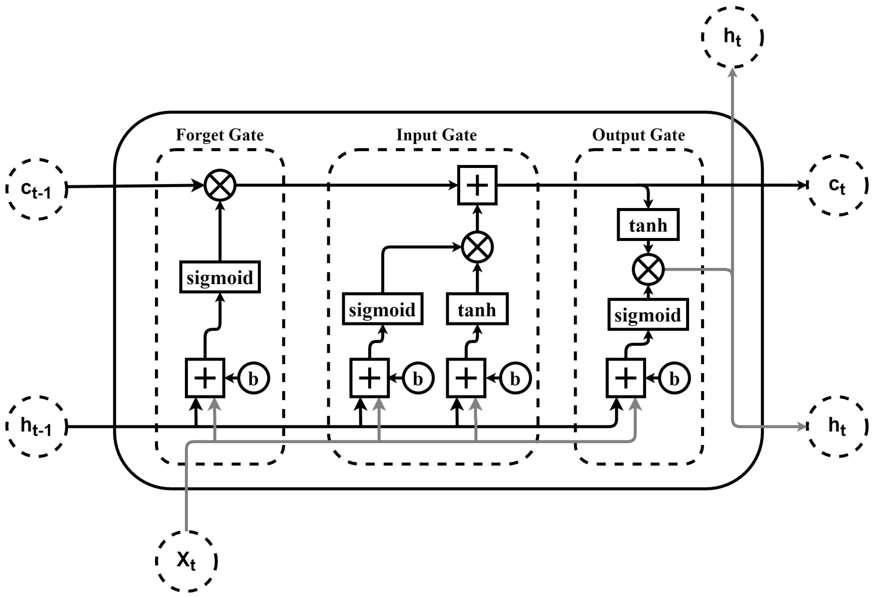


Figure 6: Structure of an LSTM unit.

1. *Forget Gate:* The first action that an LSTM unit executes is to identify and omit the trivial information received from the previous unit. This operation is carried out by the forget gate, and its functionality is mathematically expressed by equation (8).

$$f_t = \sigma(W_f[h_{t-1}, X_t] + b_f) \tag{8}$$

2. *Input Gate:* Updations in the received cell state (c_{t-1}) is done by the input gate of the LSTM unit, this whole operation can be divided into three parts :

    (a) Sigmoid layer : In this step, the gate decides whether the new information should be appended or ignored (equation (9)).

    $$i_t = \sigma(W_i[h_{t-1}, X_t] + b_i) \tag{9}$$

    (b) Tanh layer : Here, weights are given to the passed values by determining their level of importance (equation (10)).

    $$N_t = tanh(W_n[h_{t-1}, X_t] + b_n) \tag{10}$$

    (c) At last, the new cell state (c_t) gets calculated by equation (11).

    $$c_t = c_{t-1} * f_t + N_i * i_t \tag{11}$$

    In equations (9),(10) and (11), c_{t-1} and c_t are the cell states at time $(t-1)$ and t, while W_i, W_n, and b_i, b_n are the weight matrices and biases of the two layers of the input gate.

3. *Output Gate:* Working of the output gate can be partitioned into the following two steps :

    (a) A sigmoid layer (equation (12)) decides which elements of c_t get filtered out and which parts of it make it to the output.

    $$O_t = \sigma(W_o[h_{t-1}, X_t] + b_o), \tag{12}$$

    Here, W_o and b_o are the weight matrix and bias, respectively, of the output gate.

    (b) The gate output h_t is calculated by the equation (13).

    $$h_t = O_t \ tanh(c_t) \tag{13}$$

    Here h_t serves two purposes, the output of the LSTM unit, as well as the hidden state which gets passed to the next unit.

3.4. Gated Recurrent Unit Network

Although the LSTMs show better accuracy than that of the previously explained methods, training LSTM models is computationally very expensive. An evolution of LSTM, GRU was proposed by [53] to solve this problem. As described in section 3.3, an LSTM unit is composed of three gates, whereas, a GRU unit has only two gates in it, i.e., the update gate and reset gate. The structure of a GRU unit is described in figure 7, the remainder of this subsection explains the two gates of a GRU unit.

1. *Reset Gate:* Working of this gate can be mathematically described by equation (14).

    $$r_t = \sigma(W_r[h_{t-1}, X_t] + b_r) \tag{14}$$

    Where, r_t, W_r and b_r are the output vector, the weight and bias of the gate, respectively.

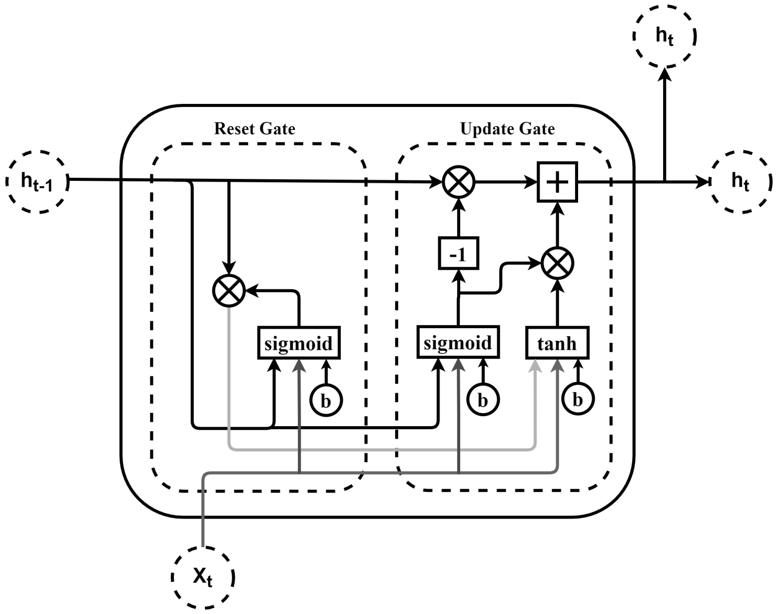


Figure 7: Structure of a GRU unit.

2. *Update Gate:* Like the reset gate, the update gate determines the level of updation which is to be done in the received data (equation (15)). The output (h_t) of the unit is a linear interpolation between element-wise multiplication of h_{t-1} and update gate output z_t, and the element-wise multiplication of $\hat{h}$ and $(1-z_t)$ (equation (17)). Where, $\hat{h_t}$ is calculated by using the reset gate output and the current input as shown in equation (16).

$$z_t = \sigma(W_z[h_{t-1}, X_t] + b_z) \tag{15}$$

$$\hat{h_t} = tanh(W_h[r_t * h_{t-1}, X_t] + b_h) \tag{16}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \hat{h_t} \tag{17}$$

Here, W_z, W_h, b_z and b_h are the weights and biases used in the update gate.

4. Methodology

The methodology used for the development of the forecasting models can be divided into five phases: Data Collection, Data Pre-processing (DP), Model Compilation, Model Training and Model evaluation (figure 8). In the first phase, we fetch the query-data files of the target states. These files can be downloaded in .json format from the official website of the farmers' help-line center ([5]). Next, DP is employed for cleaning and merging the downloaded files, removing irrelevant data, inserting some attributes, and at last, transforming the data into time-series. In the Model Compilation phase, ML and DL-based models are complied with different architecture styles, depending on the type of forecasting to be done (1, 7, 15, or 30 days). Later, after splitting the complete dataset into two parts (i.e., training data and testing data), the models learn on the training data. Finally, the trained models are evaluated on the testing data. For the performance evaluation of the models, we use three metrics, i.e., Mean squared Error (MSE), Mean absolute error (MAE), and Correlation coefficient (R). The remainder of this section elaborates on each of the above mentioned phases in detail.

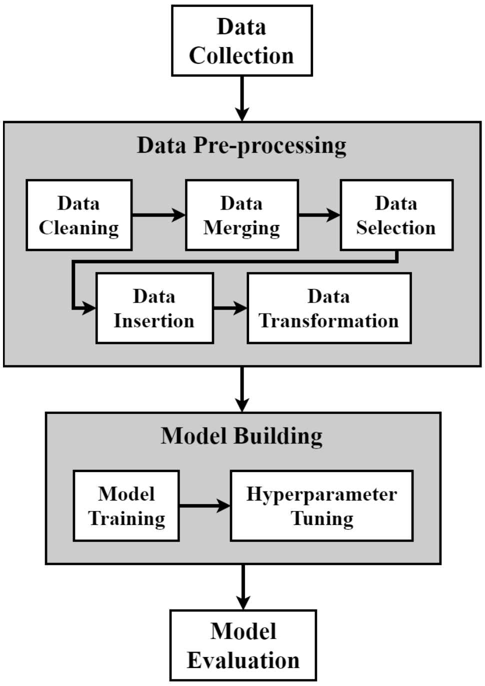


Figure 8: Block diagram of the methodology

4.1. Data Collection

The information regarding the KCC-helpline phone calls are publically available and can be retrieved from KKMS's official website ([5]). Every month, the KKMS server saves one file corresponding to each state, and these can be downloaded in .json format. For this study, a total of 414 such files were downloaded from the KKMS server, which contained 13,46,184 queries' records. A simple web-server crawler automated the downloading process, which was explicitly developed for this study. The fetched data consists of the queries from March 2013 to March 2020, corresponding to the top five rice-producing states, including West Bengal, Uttar Pradesh, Punjab, Bihar, and Andhra Pradesh. Attributes of the tuples present in the data are described in Table 1; all attributes are in textual format.

4.2. Data Pre-processing (DP)

DP is a data mining method that is used to reconstruct the raw data in a useful and effective form. DP steps used in this study involve data cleaning, merging, selection, insertion, and transformation; the following points explains these steps in brief.

Data Cleaning. In the first step of DP, we use data cleaning to remove unexpected, erroneous tuple entries from the files. In this step, we eliminate all characters from the files, excluding the alphabets (a-z, A-Z), numerical digits (0-9), and some special symbols, including the comma, space, and hyphen.

Data Merging. For this study, a total of 414 files were downloaded from the KKMS server. These files represent the query data from March 2013 to March 2020 corresponding to India's top five rice-producing states (West Bengal, Uttar Pradesh, Punjab, Bihar, and Andhra Pradesh). After removing the erroneous characters from the files, the next step is to combine all of the files' records into five separate .csv files (each file corresponding to each of the states mentioned above). A separate database file makes it more accessible to manage and execute operations on

Table 1: Attributes' description of the downloaded data

Attribute Name	Sample values	Description
QueryText	asked about reddening of leaves, control of tsetse fly, asked about sucking pest, leaf webber	Query (in text) made by the farmer
KccAns	spray micromax, spraying of carbaryle 3 grms/lit water suggest to spray chloripyriphos, carbaril 2g/l	Query response (in text) by the KCC
QueryType	Plant Protection, Water Management, Field Preparation, Agriculture Mechanization	Query's type
Category	Cereals	Category of the query
Crop	Rice (Paddy)	Target crop of the query
Sector	AGRICULTURE	Target farming-sector of the query
Season	RABI, JAYAD, KHARIF	Cropping season of the year
CreatedOn	2017-06-14 18:22:00.000	Year, month, date and time of the query
StateName	UTTAR PRADESH, WEST BENGAL, ANDHRA PRADESH, BIHAR, PUNJAB	State of the farmer
DistrictName	Barnala, Bathinda, Faridkot, Fatehgarh Sahib, Fazilka, Ferozepur	District of the farmer
BlockName	Jandiala Guru, Majitha, Rayya, Tarsikka, Verka	Block of the farmer

the dataset. Table 2 gives the number of queries obtained corresponding to each of the states (the number of queries present in each of the merged files).

Data Selection. To obtain the time-series information from the data, the output dataset of this step includes only the "CreatedOn" attribute. Therefore, after this operation, the dataset consists of only a list of dates at which the queries were made.

Data Insertion. The "CreatedOn" attribute includes information about the day, month, year, and time of the telephonic query altogether in textual form. Since the study only needs the date-part of this attribute, in this step, we replaced "CreatedOn" with three new attributes, i.e., "day", "month", and "year".

Table 2: Number of queries corresponding to the Indian states
(from March 2013 to March 2020)

State	No. of queries
Uttar Pradesh	5,99,348
Punjab	4,54,697
Bihar	69,724
West Bengal	1,49,048
Andhra Pradesh	73,367

Data Transformation. Until this phase, the dataset contains information regarding every query call, but for our objective, we needed time-series data to train the forecasting models. For this transformation, we grouped the queries based on their date information and later counted the number of queries present in each group. This gives the number of queries made each day from March, 2013 to March, 2020.

4.3. Model Building

The process of building the forecasting models can be divided into two parts, Model training and Hyperparameter tuning.

1. *Model Training:* After transformation of the dataset into time-series, it is divided into two parts, i.e., the training set and testing set (in ratio 70:30, respectively). Next, the datasets' values are normalized to the range of 0 to 1 using equation 18.

$$V_i = \frac{U_i - min(U)}{max(U) - min(U)} \tag{18}$$

   Where, U is the input number set, V_i is the normalized value corresponding to U_i. Next, the training and testing samples are created, each sample consists of input x and the corresponding output y. Here, x consists of 365 data points corresponding to the past 1-year data, and y includes the next data points in the series. The number of data points in y corresponds to the forecasting days the model is getting trained on (1, 7, 15, or 30). While creating the samples, the values from the datasets are picked using the sliding window of $|x| + |y|$ data points. All the MLP, LSTM, and GRU-based models are trained using the "early stopping" technique, which means that the algorithm stops iterating when there seems to be no improvement in the models' performance.

2. *Hyperparameter Tuning:* Remainder of this section elaborates on the hyperparameters used for building various forecasting models, all hyper-parameters are tuned using the random search approach ([54]). We have evaluated the training accuracy of the models on different set of randomly chosen parameters. The final set of parameters are decided based on the best accuracy achieved. For this study, we developed models for India's top 5 rice-producing states, furthermore, for each state, models are developed for four different forecasting periods (1, 7, 15, and 30-days forecasting) as shown in figure (9). Moreover, each type of forecasting is done through four different algorithms (SVR, MLP, LSTM, and GRU). Therefore, in the proposed work, overall 80 forecasting models (4 algorithms $\times$ 4 forecasting ranges $\times$ 5 states) are built and compared.

   (a) *SVR Hyperparameters :* Table 3 gives the details of hyperparameters used in each of the SVR models. Each entry of the table describes a vector consisting of 3 values $\langle c, t, e\rangle$, the first element c represents the value of regularization parameter, t signifies tolerance for stopping criterion, and e describes the radius of $\epsilon - SVR$ tube. RBF is used as kernel in all the SVR models, with γ set to $\frac{1}{n_f \times v}$. Where, n_f represents the number of features and v represents the variance of the input dataset.

   (b) *MLP Hyperparameters :* Table 4 gives the architecture details of all the MLP models formulated in this study. Each entry of the table represents a vector to describe the corresponding model's architecture. The elements of the vectors represent the number of perceptrons present in the corresponding layer of the model. For example, vector $\langle l_1, l_2, l_3\rangle$, in this model, the first (input) layer contains l_1, second (hidden) layer contains l_2, and third (output) layer contains l_3 number of perceptrons. It is to be noted that all

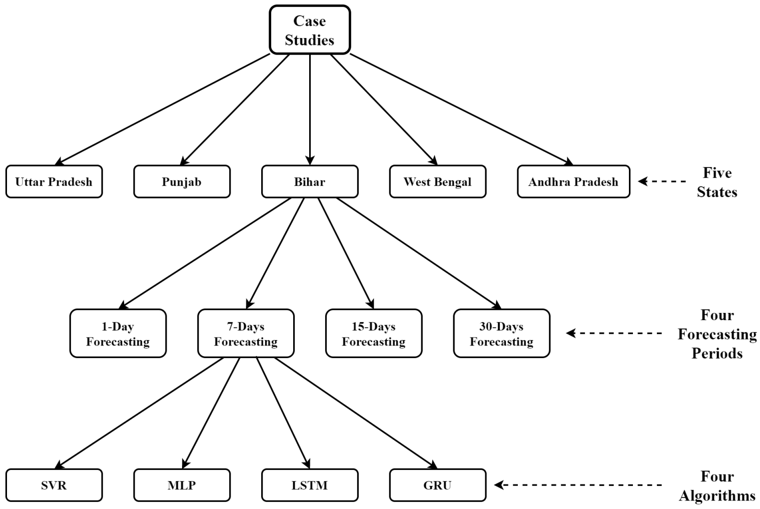


Figure 9: Models developed for various case studies in the proposed work

Table 3: Hyperparameters of the SVR Models

State	**1 Day forecasting**	**7 Days forecasting**	**15 Days forecasting**	**30 Days forecasting**
Uttar Pradesh	$\langle 0.782, 1e-8, 1e-5 \rangle$	$\langle 1, 1e-5, 0.03 \rangle$	$\langle 1, 1e-3, 0.05 \rangle$	$\langle 1, 1e-5, 0.09 \rangle$
Punjab	$\langle 0.2, 1e-5, 1e-6 \rangle$	$\langle 5, 1e-3, 0.1 \rangle$	$\langle 9, 1e-3, 0.1 \rangle$	$\langle 15, 1e-3, 0.1 \rangle$
Bihar	$\langle 0.15, 1e-8, 1e-5 \rangle$	$\langle 1.5, 1e-5, 0.001 \rangle$	$\langle 1, 1e-3, 0.05 \rangle$	$\langle 1, 1e-5, 0.09 \rangle$
West Bengal	$\langle 0.8, 1e-7, 0.001 \rangle$	$\langle 2, 1e-5, 0.1 \rangle$	$\langle 10, 1e-3, 0.1 \rangle$	$\langle 3, 1e-1, 0.1 \rangle$
Andhra Pradesh	$\langle 0.055, 1e-5, 0.001 \rangle$	$\langle 1, 1e-5, 0.03 \rangle$	$\langle 1.2, 1e-3, 0.05 \rangle$	$\langle 1, 1e-5, 0.09 \rangle$

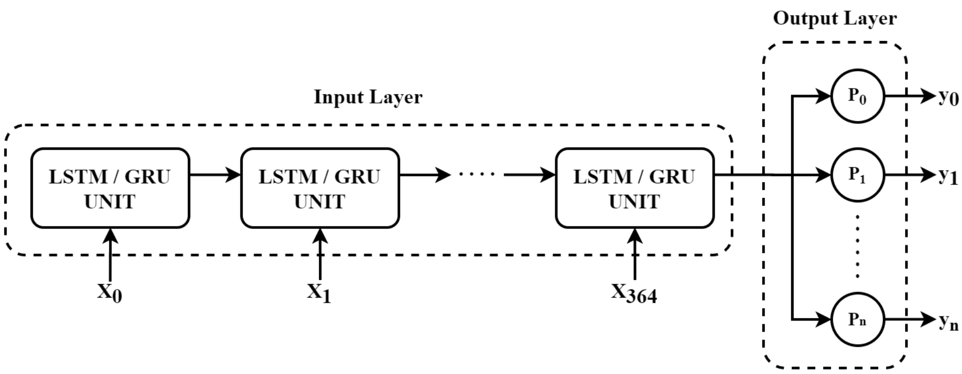


Figure 10: Architecture of the LSTM/GRU-based models

the layers in the MLP models use "ReLU" as activation function, except the output layer, which uses "sigmoid" for its activation.

(c) *LSTM & GRU Hyperparameters :* All the LSTM and GRU-based models developed in the study have a similar architecture (figure 10). The first layer consists of 365 LSTM/GRU units with "ReLU" activation, followed by the output layer of perceptrons with the "sigmoid" activation. The number of perceptrons in the output layer is equal to the number of days in the forecasting period. For example, models that are developed for one-day forecasting has one perceptron in their output layer, and so on.

Table 4: Hyperparameters of the MLP models

State	**1 Day forecasting**	**7 Days forecasting**
Uttar Pradesh	$\langle 365, 2500, 2500, 1 \rangle$	$\langle 365, 2500, 2500, 7 \rangle$
Punjab	$\langle 365, 2500, 3500, 3500, 3500, 2500, 1 \rangle$	$\langle 365, 2500, 3500, 3500, 2500, 7 \rangle$
Bihar	$\langle 365, 2500, 3500, 2500, 1 \rangle$	$\langle 365, 2500, 3500, 2500, 7 \rangle$
West Bengal	$\langle 365, 2500, 2500, 2500, 1 \rangle$	$\langle 365, 2500, 2500, 2500, 7 \rangle$
Andhra Pradesh	$\langle 365, 2500, 3500, 2500, 1 \rangle$	$\langle 365, 2500, 3500, 2500, 7 \rangle$
	15 Days forecasting	**30 Days forecasting**
Uttar Pradesh	$\langle 365, 2500, 2500, 15 \rangle$	$\langle 365, 2500, 2500, 30 \rangle$
Punjab	$\langle 365, 2500, 3500, 2500, 15 \rangle$	$\langle 365, 2500, 3500, 2500, 30 \rangle$
Bihar	$\langle 365, 2500, 3500, 2500, 15 \rangle$	$\langle 365, 2500, 3500, 2500, 30 \rangle$
West Bengal	$\langle 365, 2500, 500, 500, 2500, 15 \rangle$	$\langle 365, 2500, 500, 500, 2500, 30 \rangle$
Andhra Pradesh	$\langle 365, 2500, 3500, 2500, 15 \rangle$	$\langle 365, 2500, 3500, 2500, 30 \rangle$

4.4. Model Evaluation

Evaluation of the developed models is done on the testing data, which consists of data from the last two years out of seven years of the complete dataset. A window of 365 data points is taken as input, and the subsequent points are considered for comparing the output. The window is shifted by the number of forecasting data points, depending on the model's forecasting type (1, 7, 15, or 30 days forecasting). Models are evaluated on the bases of MSE (equation 19), MAE (equation 20), and R (equation 21).

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{19}$$

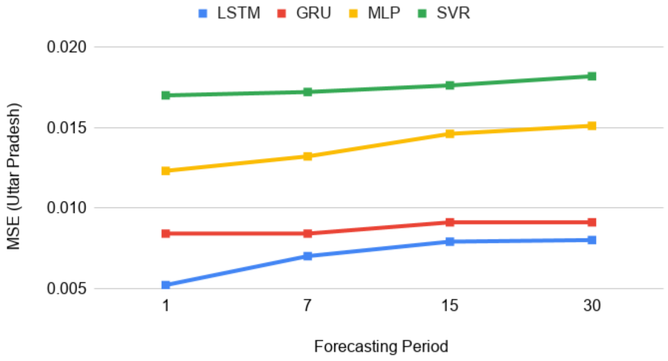
$$MAE = \frac{1}{n}\sum_{i=1}^{n}|Y_i - \hat{Y}_i| \tag{20}$$

$$R = \frac{n(\sum Y\hat{Y}) - (\sum Y)(\sum \hat{Y})}{\sqrt{\left[n\sum Y^2 - \left(\sum Y\right)^2\right]\left[n\sum \hat{Y}^2 - \left(\sum \hat{Y}\right)^2\right]}} \tag{21}$$

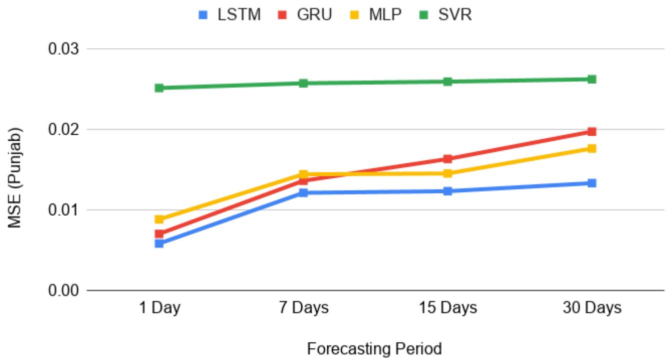
Where, n is the number of output data point, $\hat{Y}$ is the output of the forecasting model, and Y is the ground truth. The following section explains the evaluation results in detail.

5. Experiments and results

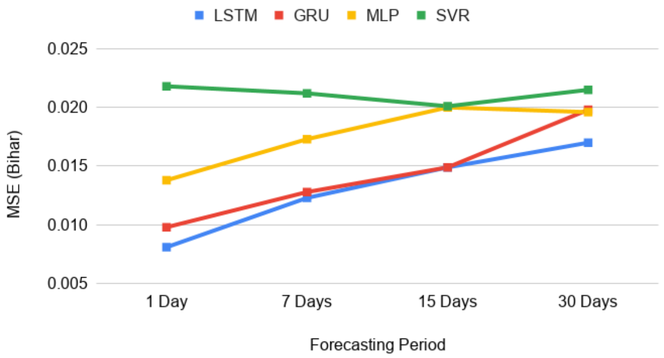
The research work done in this study is carried out on a system with 8GB RAM, 500GB SSD, 4 Cores Intel i5-1035G1 CPU @ 1.20GHz, 2GB NVIDIA Geforce MX250 Graphics card, with the operating system Window 10, and python 3.7 interpreter to develop the source code. Remainder of this section gives the metric-wise performance comparison of each forecasting model, followed by the output of the best models of each case study.



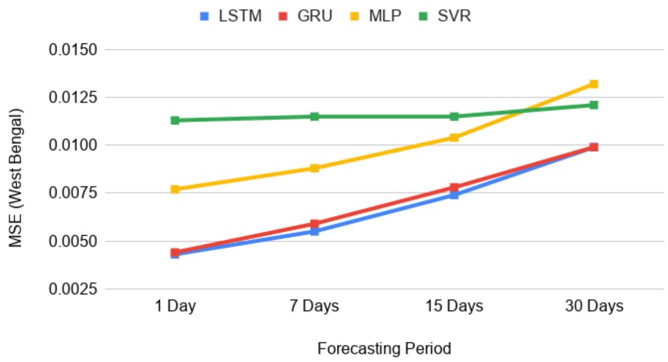
(a) MSE performance of Uttar Pradesh's forecasting models



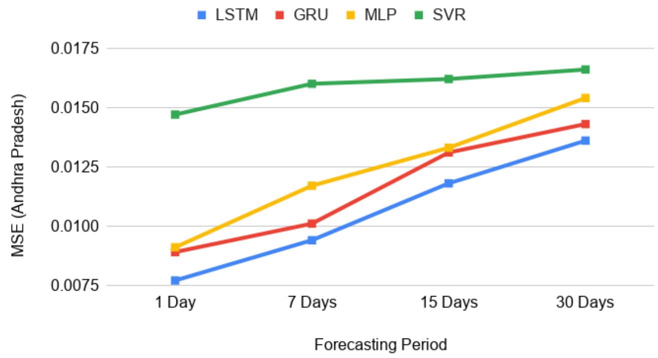
(b) MSE performance of Punjab's forecasting models



(c) MSE performance of Bihar's forecasting models



(d) MSE performance of West Bengal's forecasting models



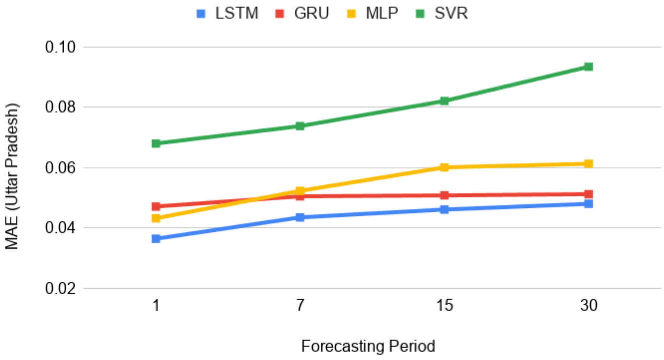
(e) MSE performance of Andhra Pradesh's forecasting models

Figure 11: State-wise MSE comparison of the forecasting models

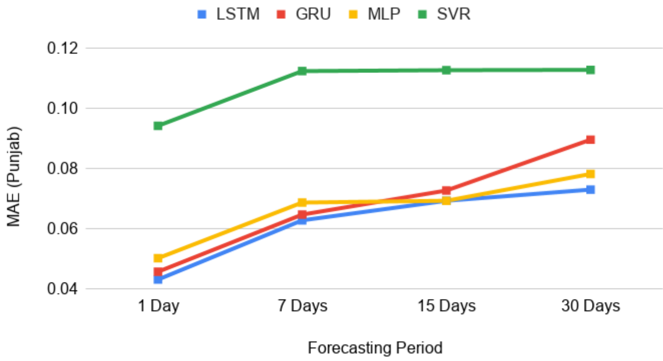
5.1. MSE comparison:

In the presented work, 16 models are trained on the query-count time-series data corresponding to each of the five states. The models are divided into four types based on the forecasting period (1, 7, 15, and 30-days forecasting). Each forecasting type is executed with four different techniques (SVR, MLP, LSTM, and GRU-based models). Figure 11 gives the MSE values achieved by each of the 80 models developed in this study, each sub-figure here represents values corresponding to a particular state. From the graphs, it can be easily observed that the MSE rate of each model declined with the decrement of the forecasting time. As it is evident that the error rate tends to increase with the increment in the forecasting period.

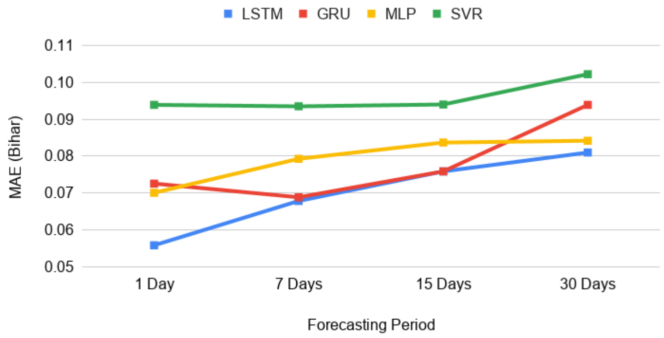
From figure 11, it can be summarized that LSTM-based forecasting models have the least MSE values in every case. Moreover, in some cases GRU-based models showed comparable performance (figure 11c, 11d) to that of the LSTM-based models. In some instances, MLP-based models outperformed GRU-based models (figure 11b, 11c), but overall MLP-based models' performance is noted to be lower than the GRU-based models. On the other hand, in



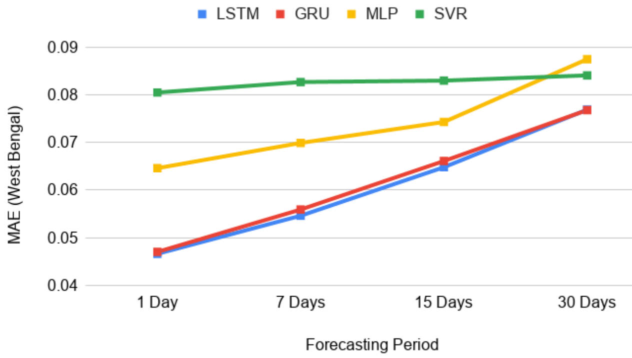
(a) MAE performance of Uttar Pradesh's forecasting models



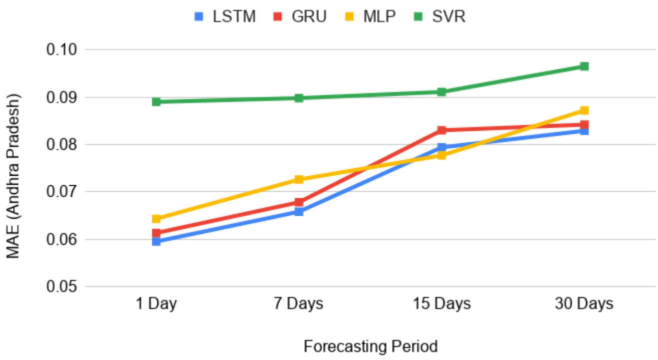
(b) MAE performance of Punjab's forecasting models



(c) MAE performance of Bihar's forecasting models



(d) MAE performance of West Bengal's forecasting models



(e) MAE performance of Andhra Pradesh's forecasting models

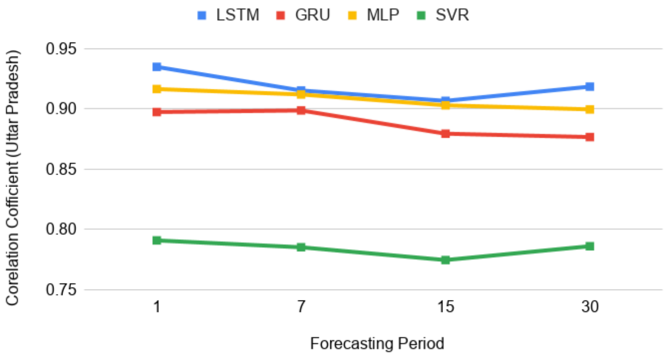
Figure 12: State-wise MAE comparison of the forecasting models

most cases the SVR-based models' performance was observed to be the lowest among all the models in terms of MSE. Altogether, the models' ranking in decreasing order of MSE performance is as follows : LSTM, GRU, MLP, and SVR.

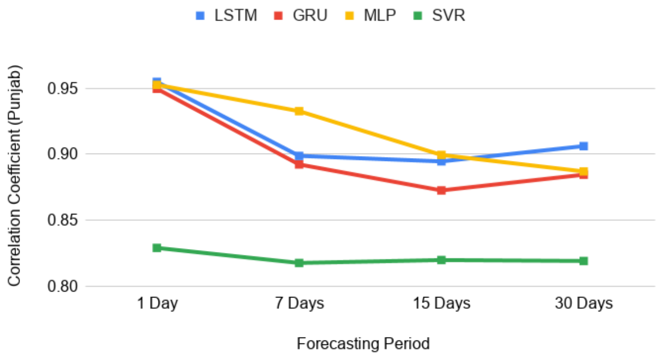
5.2. MAE comparison:

As discussed above, the error rates of all the models tend to escalate with the increase of the forecasting period. Therefore, the graphs in figure 12 show increment in MAE rates when the forecasting period is lengthened. The comparison of models' performance in terms of MAE is similar to that of MSE. In almost all cases, LSTM-based models performed the best among all other models. Furthermore, in several cases it was observed that GRU-based models' performance is comparable to that of LSTM-based model, particularly in the West Bengal state (figure 12d). Hence, the GRU-based models are noted to be at the second place after LSTM-based models in terms of MAE.

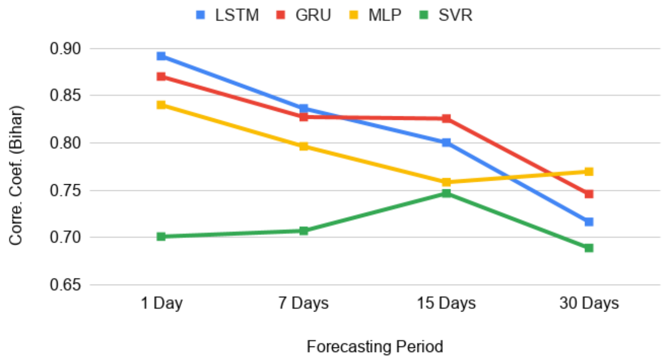
Moreover, although in some cases it was noted that the MLP-based models outperformed GRU-based models



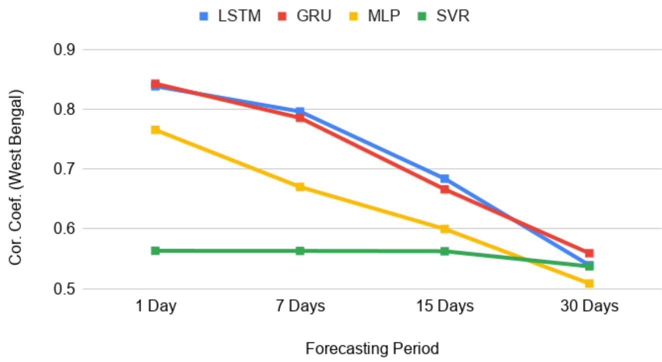
(a) R performance of Uttar Pradesh's forecasting models



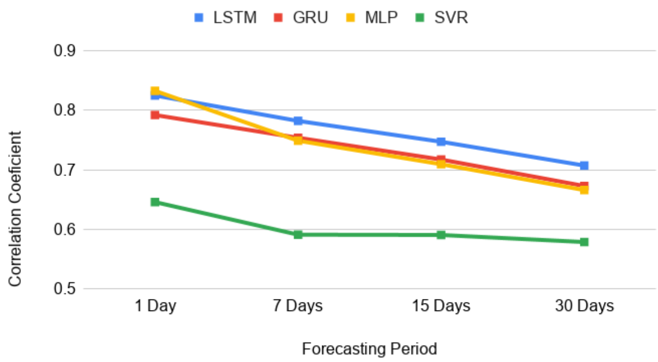
(b) R performance of Punjab's forecasting models



(c) R performance of Bihar's forecasting models



(d) R performance of West Bengal's forecasting models



(e) R performance of Andhra Pradesh's forecasting models

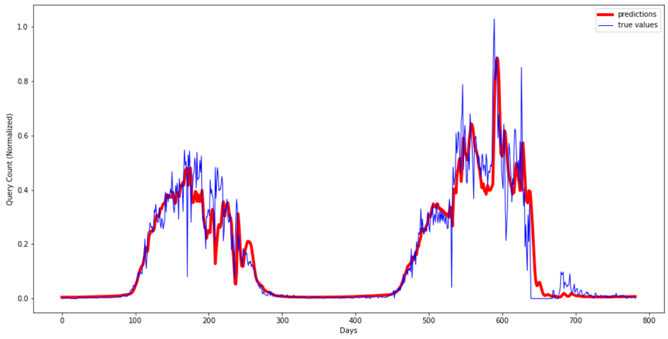
Figure 13: State-wise R comparison of the forecasting models

by a small margin, still, on average the MLP-based models' performance was found to be lower than GRU-based models. On the other hand, MAE values of the SVR-based models are noted to be the highest among all the models, except in one instance. In the case of West Bengal 30-days forecasting (figure 12d), the MLP-based model's MAE value is found to be highest among all the other models. Therefore, the models' ranking in decreasing order of MAE performance is same as the MSE-based ranking, which is as follows : LSTM, GRU, MLP, and SVR.

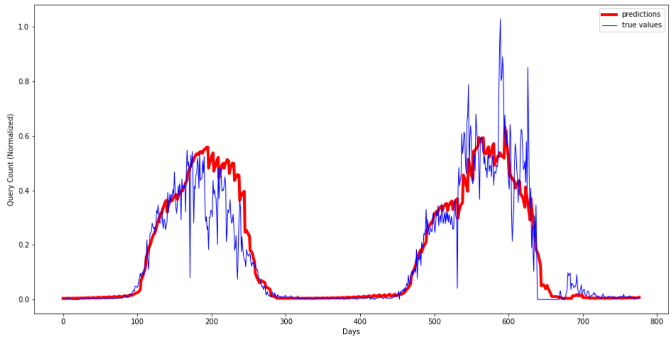
5.3. R comparison:

Unlike the observations from figures 11 and 12, the figure 13 shows that the R value tends to decrease with the increment in forecasting time. Here, R is used as an indication that the output of the forecasting model positively correlates with the ground truth. Therefore, we have given it less priority while ranking the models on their overall performance.

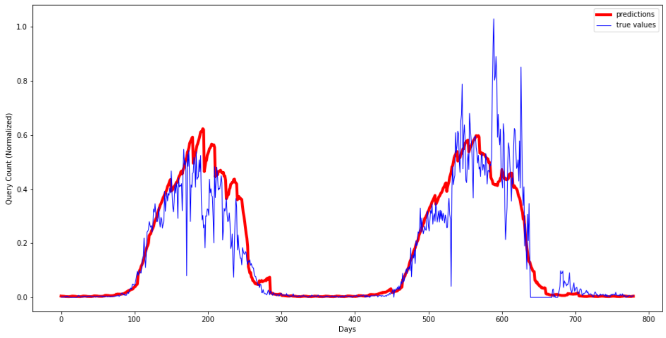
In terms of R, it is observed that LSTM-based models are at the first place in most cases. Whereas, in a few



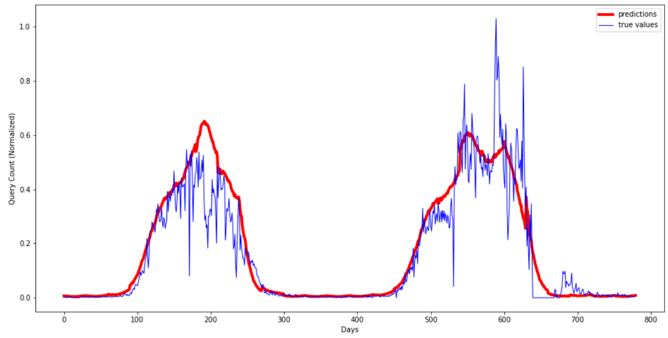
(a) LSTM-based model's 1 day forecasting predictions



(b) LSTM-based model's 7 days forecasting predictions



(c) LSTM-based model's 15 days forecasting predictions



(d) LSTM-based model's 30 days forecasting predictions

Figure 14: Uttar Pradesh forecasting models' output

cases, other models have shown the best performance. In the case of Bihar's 15 and 30-days forecasting (figure 13c), GRU-based models outperformed LSTM-based models. Whereas, in some cases, MLP-based models gave the best results among all the models, for example, in the case of Punjab's 7-days forecasting (figure 13b), Bihar's 30-days forecasting (figure 13c), and Andhra Pradesh's 1-day forecasting (figure 13e). But, in most cases, the MLP-based models are noted to be at the third place. Furthermore, the performance of SVR-based models is observed to be lowest of all the models, except in the case of West Bengal's 30-days forecasting (figure 13d), where the MLP-based model's performance was noted to be lower than the other models. Ranking of the models in terms of R seems to be same as the ranking in other metrics.

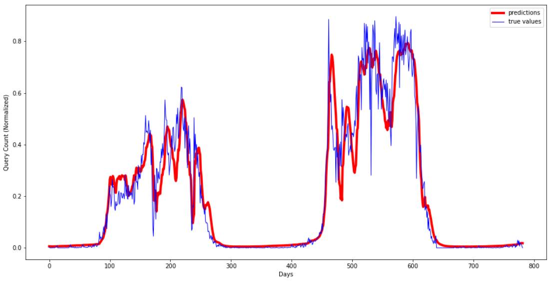
5.4. Forecasting Output

Figure 14, 15, and 16 illustrates the output of the LSTM-based forecasting models developed for the state of Uttar Pradesh, Punjab, and Bihar; these graphs present the one-step predictions of the LSTM-based models. The models' predictions are shown in red colour, whereas the ground truth is shown in blue. Since the testing data consists of 30% of the complete dataset, these graphs represent data of approximately two years (January 2018 - March 2020).

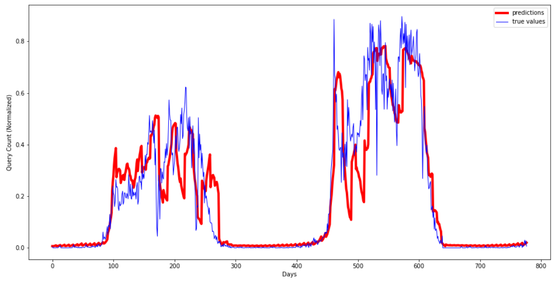
Two peaks in each of the graphs can be observed in the figures 14, 15, and 16; here, each peak represents one rice season. This is because rice is grown once in a year (Kharif season) in these three states. Whereas, the figures 17 and 18 represents the models' output of the states West Bengal and Andhra Pradesh. Here, there are no clear peaks in the graphs because rice is grown in two seasons (Rabi and Kharif season) in these two states.

6. Discussion

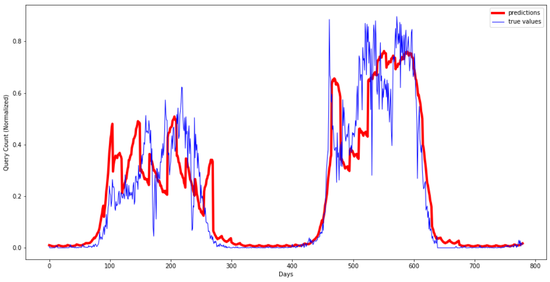
Both public and private sector organizations have given high priority to the development of modern techniques for a better understanding of the on-going agricultural practices worldwide. Understanding the trend in farmers'



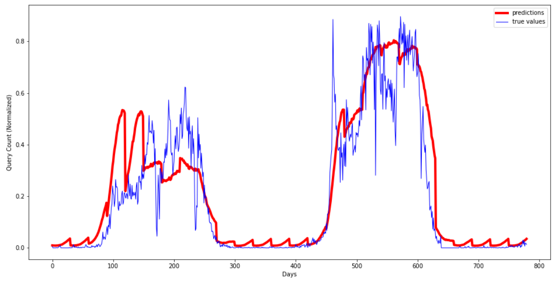
(a) LSTM-based model's 1 day forecasting results



(b) LSTM-based model's 7 days forecasting results

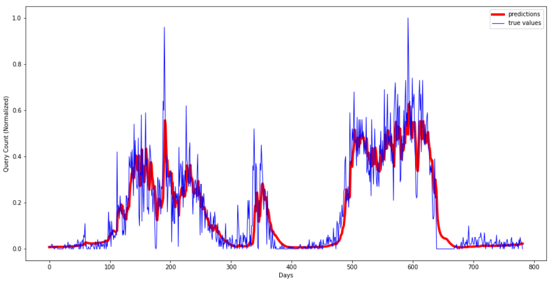


(c) LSTM-based model's 15 days forecasting results

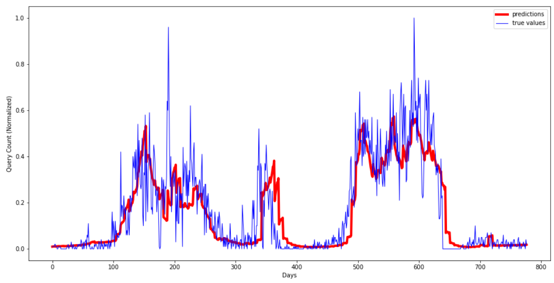


(d) LSTM-based model's 30 days forecasting results

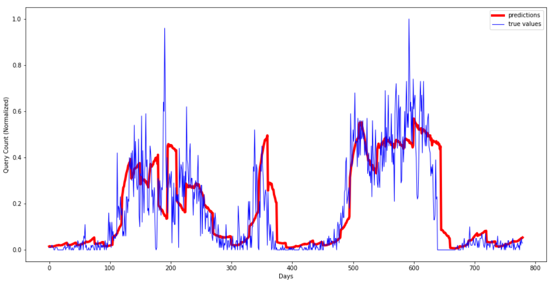
Figure 15: Punjab forecasting models' output



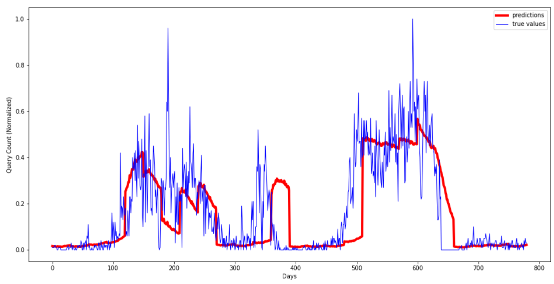
(a) LSTM-based model's 1 day forecasting results



(b) LSTM-based model's 7 days forecasting results

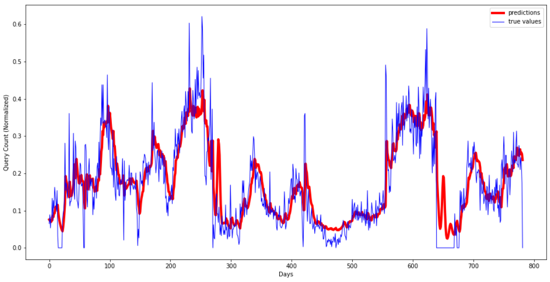


(c) LSTM-based model's 15 days forecasting results

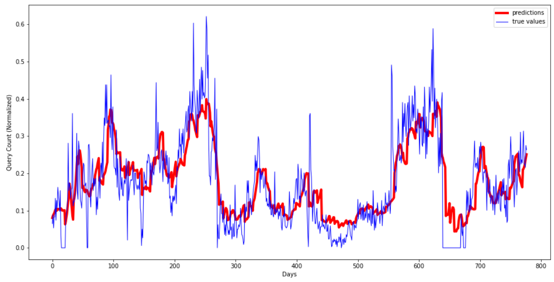


(d) LSTM-based model's 30 days forecasting results

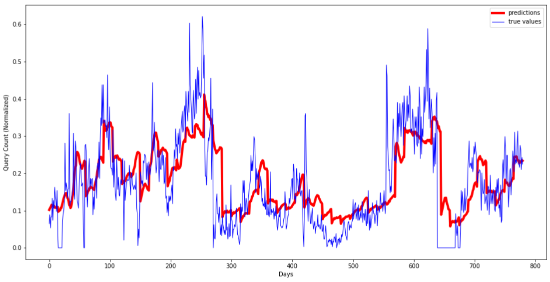
Figure 16: Bihar forecasting models' output



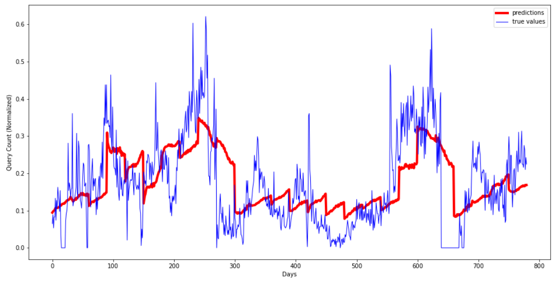
(a) LSTM-based model's 1 day forecasting results



(b) LSTM-based model's 7 days forecasting results

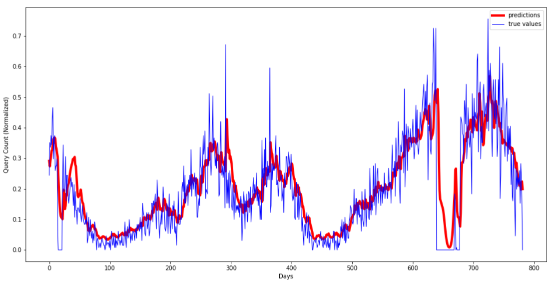


(c) LSTM-based model's 15 days forecasting results

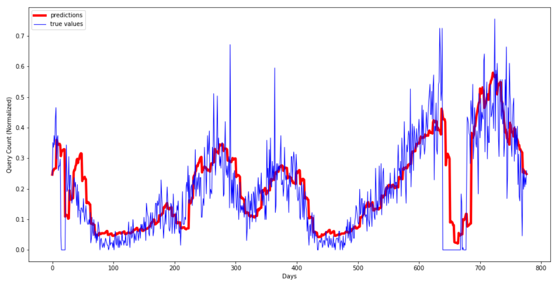


(d) LSTM-based model's 30 days forecasting results

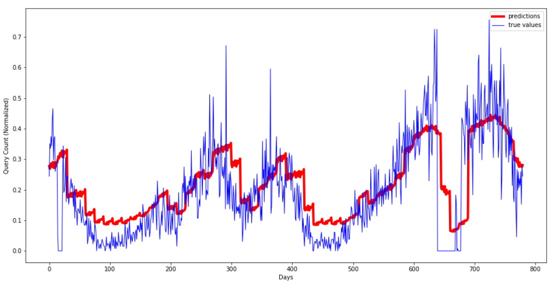
Figure 17: West Bengal forecasting models' output



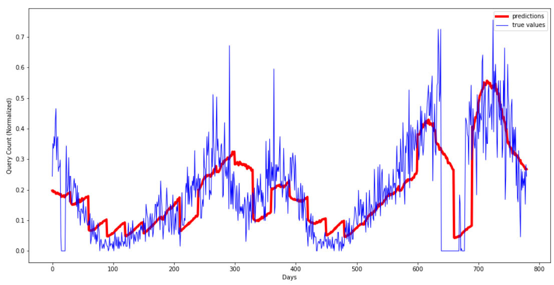
(a) LSTM-based model's 1 day forecasting results



(b) LSTM-based model's 7 days forecasting results



(c) LSTM-based model's 15 days forecasting results



(d) LSTM-based model's 30 days forecasting results

Figure 18: Andhra Pradesh forecasting models' output

problems can provide valuable insights, which is very useful for the government authorities to design and implement agricultural policies. This knowledge is also beneficial for private agriculture-related organizations to get insight into the agriculture-based product market. Existing practices executed by the organizations to obtain information regarding farmers' problems not only lead to out-dated conclusions, but can also result in inaccurate outcomes.

As a solution to this scenario, in this article, we propose to use the data records corresponding to the farmers' query calls from various helpline centers located in 22 cities all over India. Existing studies in the literature are either directed towards automating the query answering system, or extracting patterns from the helpline dataset. Whereas, our proposed approach aims to introduce a framework to forecast the trends in agricultural-problems using the farmers' query dataset.

To test the performance of the proposed framework, we used data from the five topmost rice-producing states of India (Uttar Pradesh, Punjab, Bihar, West Bengal, and Andhra Pradesh). Data from each state is used to develop four types of forecasting models, categorized based on their forecasting period, i.e., 1, 7, 15, and 30-days forecasting. Each type of forecasting is done using four different techniques (SVR, MLP, LSTM, and GRU). Altogether, in this study, we developed and tested 80 forecasting models. The optimized hyperparameters used for development of the models in this study are discussed in section 4.3.

After analyzing the developed models' performances, we observed that the LSTM-based models demonstrated the best results in every case-study in terms of MSE. Similar results are observed in terms of MAE, but, in one case, MLP-based model outperformed LSTM-based model (Andhra Pradesh, 15-days forecasting). In this particular case, the performances of both the models are comparable. Moreover, in terms of the correlation coefficient, we found that, in some instances, even though LSTM-based models achieved the best MSE results, the models' output is not as strongly correlated as that of the other models. Since all the models are developed using MSE as their loss function, out of the three metrics used for comparing the models' performances (MSE, MAE, and R), R is just an indication that the models' output correlates with the ground truth. Hence, the highest priority is given to the MSE while comparing the models' output. The comparison results show that the DL models, especially the LSTM-based models, can be highly effective in forecasting time-series related to agricultural problems. Below are some of the potential applications of the proposed research work :

- Prediction systems to forecast the trend in farmers' problems.
- Recommender systems for suggesting the time of agriculture-related training to the farmers, providing them knowledge at the right time of the year.
- Prediction systems to forecast the trend in crops' popularity.
- Market-trend prediction system for forecasting market demand for agricultural technologies, including agricultural machinery, nutrients, insecticides, pesticides, etc.
- Recommender systems for organizing marketing strategies of agriculture related products.
- Frameworks to study the various factors (like climate, soil type, etc.) which contribute to the strong association among the agricultural problems.

7. Conclusion

Due to the low literacy rate and lack of technology in many developing countries, a robust system is in demand for analyzing farmers' problems. The information regarding the agricultural problems that reaches the government officials and private organizations through manual surveys is neither updated and is also more prone to errors. Our contribution to this situation is to introduce a new way to gain insights regarding site-specific agricultural-problems and help decision-makers to forecast its trend using the query-data of farmers' helpline service. In this context, the proposed framework uses ML/DL techniques to generate predictions from the given data. We evaluated the performances of all the developed models over five different case studies. Furthermore, to have an in-depth comparison of the models, we used three performance measures to evaluate the forecasting models' results. Altogether, in this work, we developed a total of 80 forecasting models and examined their predictions' potential. From the comparison results, we observed that for our problem, the ranking of the techniques in decreasing order of accuracy is LSTM, GRU, MLP, and SVR. Nevertheless, the obtained results show that the proposed framework is useful in forecasting the trend in all the specified periods. This type of forecasting is beneficial for adopting appropriate strategies and policies to improve agricultural practices in any developing country. The developed framework can be used to construct various types of expert systems including recommender systems, prediction models, and warning systems. Conclusively, in future work, the proposed approach will be expanded with other data preprocessing techniques such as data encoder-decoder to enhance the feature extraction functionality of the framework. Moreover, models will be developed using more complex DL techniques such as attention-based models to perform forecasting for extended periods.

References

[1] M. S. U. E. George Silva, Feeding the world in 2050 and beyond (2018).
URL https://www.canr.msu.edu/news/feeding-the-world-in-2050-and-beyond-part-1

[2] R. Bailey, Disrupting dinner? Food for the future (2017).
URL https://hoffmanncentre.chathamhouse.org/article/disrupting-dinner-food-for-the-future/

[3] M.O.S.P.I., Agricultural Surveys, Government of India (2019).
URL http://mospi.nic.in/agriculture-surveys

[4] I. M.A.F.W., Past Events of KVKs (2019).
URL https://kvk.icar.gov.in/past_events_kvk.aspx

[5] K. G.O.I., Kisan Call Center (2020).
URL https://dackkms.gov.in/

[6] N. H. Board, State-wise rice production in India (2020).
URL http://agriexchange.apeda.gov.in/India%20Production/India_Productions.aspx?cat=Agri&hscode=1011

[7] F. U.N., India at a glance (2019).
URL http://www.fao.org/india/fao-in-india/india-at-a-glance/en/

[8] M.O.S.P.I., National Sample Survey Office (NSSO) (2019).
URL http://mospi.nic.in/NSSOa

[9] P. Rosenfeld, S. Booth-Kewley, J. E. Edwards, Computer-administered surveys in organizational settings: Alternatives, advantages, and applications, American Behavioral Scientist 36 (4) (1993) 485–511.

[10] P. N. Hague, N. Hague, C.-A. Morgan, Market research in practice: a guide to the basics, Kogan Page Publishers, 2004.

[11] S. C. Zipper, Agricultural research using social media data, Agronomy Journal 110 (1) (2018) 349–358.

[12] S. Wolfert, L. Ge, C. Verdouw, M.-J. Bogaardt, Big data in smart farming–a review, Agricultural Systems 153 (2017) 69–80.

[13] A. Kamilaris, A. Kartakoullis, F. X. Prenafeta-Boldú, A review on the practice of big data analysis in agriculture, Computers and Electronics in Agriculture 143 (2017) 23–37.

[14] J. Muangprathub, N. Boonnam, S. Kajornkasirat, N. Lekbangpong, A. Wanichsombat, P. Nillaor, Iot and agriculture data analysis for smart farm, Computers and electronics in agriculture 156 (2019) 467–474.

[15] C. Brewster, I. Roussaki, N. Kalatzis, K. Doolin, K. Ellis, Iot in agriculture: Designing a europe-wide large-scale pilot, IEEE communications magazine 55 (9) (2017) 26–33.

[16] L. Ding, J. Gu, Y. Wang, J. Wu, analysis of telephone call detail records based on fuzzy decision tree, in: International Conference on Forensics in Telecommunications, Information, and Multimedia, Springer, 2010, pp. 301–311.

[17] R. G. Fukkink, S. Bruns, R. Ligtvoet, Voices of children from around the globe; an international analysis of children's issues at child helplines, Children & Society 30 (6) (2016) 510–519.

[18] C.-H. Chan, H.-K. Wong, P. S.-F. Yip, Exploring the use of telephone helpline pertaining to older adult suicide prevention: A hong kong experience, Journal of affective disorders 236 (2018) 75–79.

[19] T. Shafiei, L. J. Biggs, R. Small, H. L. McLachlan, D. A. Forster, Characteristics of women calling the panda perinatal anxiety & depression australia national helpline: a cross-sectional study, Archives of women's mental health 21 (6) (2018) 801–812.

[20] P. Jaisridhar, Impact of kisan call center on technological adoption among dairy farmers of tamilnadu., Ph.D. thesis, NDRI, Karnal (2013).

[21] S. Kavitha, A. Nallusamy, A study on socio economics characteristics of kisan call centre beneficiaries and non-beneficiaries in mahaboobnagar district of telangana, Journal of Pharmacognosy and Phytochemistry 8 (3) (2019) 4660–4663.

[22] K. Chachra, G. Seelam, H. Singh, M. Sarkar, A. Jain, A. Jain, The impact of kisan call centers on the farming sector, in: Environmental and Agricultural Informatics: Concepts, Methodologies, Tools, and Applications, IGI Global, 2020, pp. 66–78.

[23] B. S. Behera, T. Das, K. Jishnu, R. Behera, A. Behera, S. Jena, E-governance mediated agriculture for sustainable life in india, Procedia Computer Science 48 (2015) 623–629.

[24] M. Gandhi, A. Pallae, A. Srinivas, et al., A scale to measure the attitude of farmers towards kisan call center, Interaction 31 (2) (2013) 56–59.

[25] V. S. Parmar, Knowledge and attitude of farmers towards the use of kisan call center in sount gujarat region, Ph.D. thesis, Department of Agricultural Extension, N.M. College of Agriculture.

[26] V. Sudharani, A. Pallae, A. Srinivas, et al., A scale to measure the attitude of farmers towards kisan call center-development and standardization, Interaction 31 (1) (2013) 77–80.

[27] V. K. Viswanath, C. G. V. Madhuri, C. Raviteja, S. Saravanan, M. Venugopalan, Hadoop and natural language processing based analysis on kisan call center (kcc) data, in: 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE, 2018, pp. 1142–1151.

[28] S. K. Mohapatra, A. Upadhyay, Query answering for kisan call centerwith lda/lsi, in: 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), IEEE, 2018, pp. 711–716.

[29] S. K. Mohapatra, A. Upadhyay, Using tf-idf on kisan call centre dataset for obtaining query answers, in: 2018 International Conference on Communication, Computing and Internet of Things (IC3IoT), IEEE, 2018, pp. 479–482.

[30] S. Godara, D. Toshniwal, Sequential pattern mining combined multi-criteria decision-making for farmers' queries characterization, Computers and Electronics in Agriculture 173 (2020) 105448.

[31] M. I. Rapoo, E. Munapo, M. M. Chanza, O. S. Ewemooje, Modelling and forecasting portfolio inflows: A comparative study of support vector regression, artificial neural networks, and structural var models, in: Handbook of Research on Smart Technology Models for Business and Industry, IGI Global, 2020, pp. 329–350.

[32] A. Sherstinsky, Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network, Physica D: Nonlinear Phenomena 404 (2020) 132306.

[33] A. Zhang, Z. C. Lipton, M. Li, A. J. Smola, Dive into deep learning, Unpublished Draft. Retrieved 19 (2019) 2019.

[34] P. Naraei, A. Abhari, A. Sadeghian, Application of multilayer perceptron neural networks and support vector machines in classification of healthcare data, in: 2016 Future Technologies Conference (FTC), IEEE, 2016, pp. 848–852.

[35] T. Razzaghi, O. Roderick, I. Safro, N. Marko, Multilevel weighted support vector machine for classification on healthcare data with missing values, PloS one 11 (5) (2016) e0155119.

[36] B. M. Henrique, V. A. Sobreiro, H. Kimura, Stock price prediction using support vector regression on daily and up to the minute prices, The Journal of finance and data science 4 (3) (2018) 183–201.

[37] Y. Chen, P. Xu, Y. Chu, W. Li, Y. Wu, L. Ni, Y. Bao, K. Wang, Short-term electrical load forecasting using the support vector regression (svr) model to calculate the demand response baseline for office buildings, Applied Energy 195 (2017) 659–670.

[38] J. Bedi, D. Toshniwal, Deep learning framework to forecast electricity demand, Applied energy 238 (2019) 1312–1326.

[39] Z. Zhang, D. Robinson, J. Tepper, Detecting hate speech on twitter using a convolution-gru based deep neural network, in: European semantic web conference, Springer, 2018, pp. 745–760.

[40] S. Cascianelli, G. Costante, T. A. Ciarfuglia, P. Valigi, M. L. Fravolini, Full-gru natural language video description for service robotics applications, IEEE Robotics and Automation Letters 3 (2) (2018) 841–848.

[41] A. D. Mehr, V. Nourani, V. K. Khosrowshahi, M. A. Ghorbani, A hybrid support vector regression–firefly model for monthly rainfall forecasting, International Journal of Environmental Science and Technology 16 (1) (2019) 335–346.

[42] B. T. Pham, D. T. Bui, I. Prakash, M. Dholakia, Hybrid integration of multilayer perceptron neural networks and machine learning ensembles for landslide susceptibility assessment at himalayan area (india) using gis, Catena 149 (2017) 52–63.

[43] C. Cortes, V. Vapnik, Support-vector networks machine learning (pp. 237–297), vol. 20 (1995).

[44] H. Drucker, C. J. Burges, L. Kaufman, A. J. Smola, V. Vapnik, Support vector regression machines, in: Advances in neural information processing systems, 1997, pp. 155–161.

[45] A. J. Smola, B. Schölkopf, A tutorial on support vector regression, Statistics and computing 14 (3) (2004) 199–222.

[46] V. Vapnik, S. E. Golowich, A. J. Smola, Support vector method for function approximation, regression estimation and signal processing, in: Advances in neural information processing systems, 1997, pp. 281–287.

[47] W. S. McCulloch, W. Pitts, A logical calculus of the ideas immanent in nervous activity, The bulletin of mathematical biophysics 5 (4) (1943) 115–133.

[48] F. Rosenblatt, The perceptron: a probabilistic model for information storage and organization in the brain., Psychological review 65 (6) (1958) 386.

[49] A. J. Shepherd, Second-order methods for neural networks: Fast and reliable training methods for multi-layer perceptrons, Springer Science & Business Media, 2012.

[50] I. Isa, Z. Saad, S. Omar, M. Osman, K. Ahmad, H. M. Sakim, Suitable mlp network activation functions for breast cancer and thyroid disease detection, in: 2010 Second International Conference on Computational Intelligence, Modelling and Simulation, IEEE, 2010, pp. 39–44.

[51] Y. Li, Y. Yuan, Convergence analysis of two-layer neural networks with relu activation, in: Advances in neural information processing systems, 2017, pp. 597–607.

[52] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (8) (1997) 1735–1780.

[53] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using rnn encoder-decoder for statistical machine translation, arXiv preprint arXiv:1406.1078 (2014).

[54] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization, The Journal of Machine Learning Research 13 (1) (2012) 281–305.