Estimating Coconut Yield Production using Hyperparameter Tuning of Long Short-Term Memory

Model for Estimating Coconut Yield Production

Niranjan Shadaksharappa Jayanna, Raviprakash Madenur Lingaraju
Computer Science & Engineering, Kalpataru Institute of Technology, Tiptur, India
Visvesvaraya Technological University, Belagavi, India

***Abstract*—Coconut production is one of the significant and main sources of revenue in India. In this research, an Auto-Regressive Integrated Moving Average (ARIMA)-Improved Sine Cosine Algorithm (ISCA) with Long Short-Term Memory (LSTM) is proposed for coconut yield production using time series data. It is used for converting non-stationary data to stationary time series data by applying differences. The Holt-Winter Seasonal Method is the Exponential Smoothing variations utilized for seasonal data. The time-series data are given as the input to the LSTM classifier to classify the yield production and the LSTM model is tuned by hyperparameter using Improved Sine Cosine Algorithm (ISCA). In basic SCA, parameter setting and search precision are crucial and the modified SCA improves the coverage speed and search precision of the algorithm. The model's performance is estimated by utilizing R2, Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Square Error (RMSE) with date on yield from 2011-2021 by categorizing yearly production into 120 records and eight million nuts. The outcomes display that the LSTM-ISCA offers values of 0.38, 0.126, 0.049 and 0.221 for R2, MAE, MSE and RMSE metrics, which offer a precise yield production when related to other models.**

Keywords—Auto-regressive integrated moving average; coconut yield production; improved sine cosine algorithm; long short-term memory; time series

I. Introduction

Coconut is one of the most important and multi-use everlasting crops that is mostly used to produce hair oil, cosmetics, soaps and other products [1]. In India, out of all states, southern regions offer huge production of coconut, and Kerala has the largest area for coconut production. Coconut production is determined by many factors [2]. Coconuts are categorized into three types, intermediate, dwarf and tall based on their height and behavior. Coconut farmers always face price risk that is determined by coconut retention and the strength of demand in the market [3]. Coconut yield is a difficult quantitative feature that differs with environmental factors, age, variety, and communication between environmental factors and variety [4]. Mature nuts are considered as yield because the solid endosperm content in nuts is used for yield production. Additionally, the temperature is important for the growth and development of nut yield of coconuts; high temperature reduces the root growth [5].

Coconut palms play an important role in humans as it meets the social, economic and cultural requirements. The tall type of nut production signifies the produced number of coconuts under agronomic environments [6]. Subsequently, the enhanced productivity in coconut yield results in proper resource usage and the evaluation of coconut enhance the yield production [7]. The accurate quantification of coconut yields is critical for disasters such as drought, which give rise to a variety of crop growth, leading to different levels of coconut yield reduction [8]. It is important to carry out sequential time series coconut growth monitoring over the main growth and development period [9]. The Auto-Regressive Integrated Moving Average (ARIMA) model is utilized for forecasting future outcomes based on the past data points influencing the data points. Hyperparameter tuning is used in the LSTM model for performance and accuracy improvement, resulting in a more robust performance as compared to other deep learning models. The Improved Sine-Cosine Algorithm (ISCA) is an optimization algorithm that mainly depends on the sine and cosine operators so as to improve the performance of the LSTM model. The contributions of LSTM-ISCA are specified as follows:

- The Linear Interpolation is utilized for preprocessing which fills missing values of the dataset in districts of Kerala.
- ARIMA model is used for converting stationary time series data from non-stationary data by applying differences and then the exponential Smoothing of Holt-Winter Seasonal Method is used for seasonal data.
- The LSTM-ISCA model is employed for coconut yield production, wherein the LSTM model is tuned by hyper-parameters using Improved Sine Cosine Algorithm, and the performance is evaluated by using R2, MSE, RMSE, and MAE.

The rest of this paper is organized as follows: The literature review for coconut yield production is described in Section II. The detailed description of the proposed method is illustrated in Section III, while the obtained results of the proposed model are described in Section IV, and lastly, the conclusion of this research is described in Section V.

II. Literature Review

Novarianto [10] implemented an aerial photography method by using drones to establish the classification of local tall coconut production, and coconut diversity in Taliabu Regency, Taliabu Island and North Maluku Province. The drone technology increased the data efficiency by combining the classical sampling population in local tall coconut trees. This drone technology surrounded the data of plant types, total areas, number of areas planted with coconut, and characteristics like the number of coconuts, and the height of the coconut palm. The coconut production information was used to develop a coconut rehabilitation and replanting approach. The yield population density of each area was determined by different factors such as the number of palms dying, and the pests and diseases that are not suitable for coconut production.

Samarakoon et al. [11] developed a different coconut plantation using Ordinary Least Square (OLR) estimation and Quantile Regression (QR) approaches to estimate the production function in Cobb Douglas functional form. OLR estimation was an appropriate method for conditional mean model estimation, while the QR provides an equally appropriate method for the conditional quantile function. In OLS estimation, the amount of bearing palms has a beneficial effect when the input variables are considered without delay on coconut production. In comparison, QR provides significant information that enables suitable policies for the use of inputs in coconut production. In this model, the long-term viable option is to increase coconut production and the productivity of coconut lands.

Pramono and Arifin [12] proposed a dry coconut for copra production using a conventional method named fuzzy logic control. This conventional method was used for coconut drying in sunlight with weather conditions. In the electronic system, the fuzzy logic method used an AC to AC voltage circuit and control system using the ARM STM32F4VG microcontroller. This model consumed only 18 hours for drying and the quality of the product was assured in both laboratory testing and physical conditions. Drying using direct sunlight produced copra that was polluted with microbes and dust. Apart from the drying technique, the copra was never absolute which resulting in greater capacity of moisture.

Hadi [13] developed a quantitative with regression analysis as a casual estimation tool between variables in the Kalimantan region. This model was used to examine the risks in coconut production which affected the economic value, selling value, and income. The output showed that the risk in the coconut business is the reduction in the number of trees because of the translation of mining and plantations. Additionally, the operational costs are not equitable to the selling price. Moreover, the household coconut farmer's production risks impact the level of productivity caused in every harvest season.

Karunakaran and Narmadha [14] implemented a Quin-decadal Analysis for coconut production and growth performance in the global scenario. This model considered three methods: compound growth rate for growth models, Coppock's instability index for instability measures, and the decomposition analysis for yield production. In this model, the low-production country was correctly developed as acceptable harvesting was not carried out and the fallen nuts were not considered. Hence, high attention is needed for the states to encourage and fascinate the farmers in coconut production by retrieving modern technologies.

Das et al. [15] developed six multivariate techniques for coconut yield production using weather-based linear and nonlinear models in west coastal areas of India. Weather indices were created by using collected values for rainfall and average values for other parameters monthly, such as solar radiation, relative humidity, min and max temperature, and wind speed. Various linear and non-linear models were applied to the model for coconut yield production using input as weather indices monthly. The model showed that the frequency of weighted weather was higher, as opposed to simple weather indices.

Paudel et al. [16] suggested a crop yield production by MARS Crop Yield Forecasting System (MCYFS) data in the European Commission. Machine learning was an independent source of data collected and combined for a crop model to build a large-scale crop yield prediction baseline. The baseline had growth in fit-for-determination optimizations and overall design values. Certain countries and crop data required huge preprocessing to fit the baseline requirements which were not instigated for big data analysis.

Hebbar et al. [17] developed a MaxEnt model for coconut yield production in India under climate change scenarios. The developed model was used to estimate bioclimatic variables for defining specific cultivation and forecast region's climate, which was possibly appropriate in the future weather conditions. Based on the present cultivation region and weather change prediction from seven ensembles of Global Circulation Models, the prediction of climatic suitability was done through MaxEnt model. In climate suitability from moderate to high, high to low, and low to inappropriate under upcoming climate. In this model, coconut production and the productivity of coconut lands is increased to enhance the coconut production.

Tuckeldoe et al. [18] introduced a coconut yield and nutritional quality prediction using sterilized growth media. The introduced model was used to define the coconut's effects on biochemical constituents of the varieties cultivated under various environments. The developed model considered two seasons in 2021 and 2022 for cultivating coconuts on fertigated land. Drying using direct sunlight produced copra that is polluted with microbes and dust, and apart from the drying technique, the copra was never absolute, thereby resulting in having a greater moisture capacity.

Pham et al. [19] developed coconut mesocarp-based lignocellulosic waste as cellulase production subtract from huge promising multienzyme-producing bacillus FW2 without pretreatments. Particularly, extremophilic bacteria played a significant part in biorefinery because of huge score catalytic enzymes which were under harsh environment situations. The developed model estimated the capability to generate and isolate from food waste. It continued to discover functional candidate which enabled low expensive production, and was perfect for clean environment and waste management. This model showed that higher frequency of weighted weather when compared to simple weather indices.

III. PROPOSED METHOD

The proposed method is built for forecasting yield of coconut production. ARIMA and Holt-Winters seasonal methods are mainly used to predict data as per seasonality and trend. The LSTM model is tuned by hyperparameters using Improved Sine Cosine Algorithm because the ISCA takes the data point's non-linear dependence, which produces more favorable for time-series forecasting. Fig. 1 represents the block diagram of the proposed coconut yield prediction.

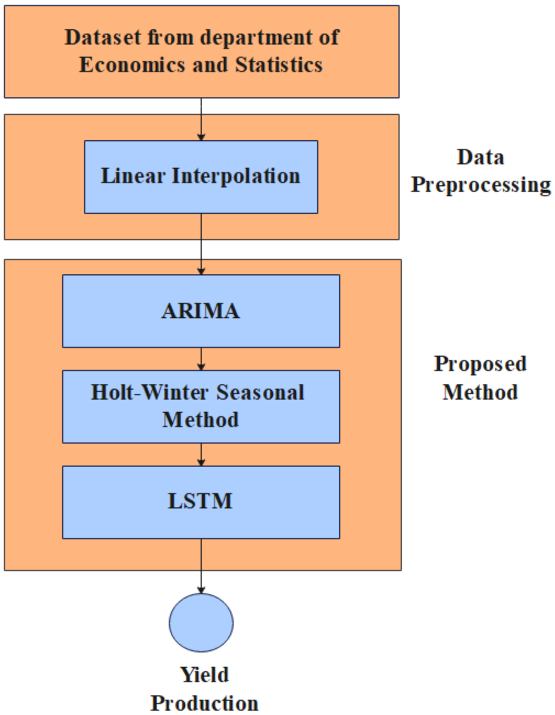

Fig. 1. Proposed method for Coconut yield production.

A. Dataset

Coconut is one of the important cultivation crops in Kerala, covering about 39% of the area grown in the state. The dataset is collected from Kerala's Department of Economics and Statistics, and consists of monthly coconut yield from 10 districts like Alappuzha, Ernakulam, Kozhikode, Palakkad, Kollam, Kasaragod, Idukki, Wayanad, Kottayam and Thrissur from 2011-2021 with yearly production of 120 records and 8 million nuts. The coconut is cultivated three months once and the 10-year annual production is segregated into 1200 records. However, the risk factors include climate change effects, diseases and pest, soil health, increased vulnerability, and water pollution and contamination.

B. Linear Interpolation

In Kerala, some districts contain missing values and these missing values are filled by using the Linear Interpolation method. This method is a mathematical procedure that measures new datapoints based on the previous data in traditional lines for enhancing the order as previous value. If it is time-series data, it contains missing scores, Interruption is a method applied to fill the missing scores in time-series data. This function is shown in Eq. (1),

$$f(X) = f(x_0) + \frac{f(x_1)-f(x_0)}{x_1-x_0}(x - x_0) \quad (1)$$

where, x x_0 and x_1 are independent variables, and $f(x)$ is a dependent for independent variable x.

C. Auto Regressive Integrated Moving Average

The Auto-Regressive Integrated Moving Average (ARIMA) model is used for converting the non-stationary time series data to stationery time data by applying differences, while also defining the present time series values using the previous predicted values and errors. ARIMA consists of three parts, denoted as ARIMA (p, d, q). Where, p is the amount of AR terms, d is the number of differences required to make the time-series stationery, and q is the amount of MA terms. The auto-regressive integrated moving average is shown in Eq. (2),

- AR: It observes a score that depends on the last lagged value.
- Integrated: The difference between observed and last lagged value.
- MA: A last lagged error values are the predictor values.

$$\emptyset(B)(1-B)^d Y_t = \theta(B)\varepsilon_t \quad (2)$$

where, $\emptyset$ is the autoregressive parameter, d is the degree of differencing parameter, B is the backshift operator, θ is a parameter of moving average and ε_t is white noise.

In time series models, the ARIMA model performs better than the deterministic growth model for short-term prediction. A better ARIMA model for time series prediction is fitted using the Box-Jenkins technique is shown in Eq. (3),

$$X_t = c + \emptyset_1 X_{t-1} + \cdots + \emptyset_p X_{t-p} + \theta_1 e_{t-1} + \cdots + \theta_q e_{t-q} + e_t \quad (3)$$

where,

- X_t = the variable that will be explained in time t;
- c = constant or intercept;
- $\emptyset$ = coefficient of each parameter p;
- θ = coefficient of each parameter q;
- e_t = errors in time t.

D. Holt-Winters Seasonal Method

The Holt-Winters seasonal method is also known as the triple exponential smoothing which forecasts both trend and seasonality. This method itself combines three simple components, which are smoothing methods.

- Simple Exponential Smoothing (SES): The method cannot be used with seasonality, trend, or both series. The SES method guesses that there is no change in the level of the time-series.
- Holt-Exponential Smoothing (HES): The method is used only with trend components and not with seasonal data.
- Winter's Exponential Smoothing (WES): The method is used for both trend component and seasonality data.

1) Holt-Winter's additive method: When the seasonal variations are not exactly close along the series, the addition method is employed. For subtracting the seasonal component, the obtained series scale and series of the level equation are seasonally altered and determined in accurate terms. The mathematical form of the additive method is shown in Eq. (4), (5), (6) and (7),

$$\text{Overall Equation: } \hat{y}_{t+h} = l_t + hb_t + s_{t+h-m} \quad (4)$$

$$\text{Level Equation: } l_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(l_{t-1} + b_{t-1}) \quad (5)$$

$$\text{Trend Equation: } b_t = \beta(l_t - l_{t-1}) + (1-\beta)\, b_{t-1} \quad (6)$$

$$\text{Seasonality Equation: } s_t = \gamma(y_t - l_{t-1} - b_{t-1}) + (1-\gamma)\, s_{t-m} \quad (7)$$

where, m is time series seasonality, s_t is the seasonal forecast component, s_{t-m} is the forecast for the previous season and y is the seasonal component smoothing factor $(0 \leq y \leq 1-\alpha)$.

2) Holt-winter's multiplicative method: When changing the seasonal variations proportion to the series level, the method uses the multiplication method. This series is seasonally altered by dividing the seasonal component and determined in relative terms. The mathematical form of the multiplicative method is shown in Eq. (8), (9), (10) and (11),

$$\text{Overall Equation: } \hat{y}_{t+h} = (l_t + hb_t)s_{t+h-m} \quad (8)$$

$$\text{Level Equation: } l_t = \alpha \frac{y_t}{s_{t-m}} + (1-\alpha)(l_{t-1} + b_{t-1}) \quad (9)$$

$$\text{Trend Equation: } b_t = \beta(l_t - l_{t-1}) + (1-\beta)\, b_{t-1} \quad (10)$$

$$\text{Seasonality Equation: } s_t = \gamma \frac{y_t}{l_{t-1} + b_{t-1}} + (1-\gamma)\, s_{t-m} \quad (11)$$

Where, m is time series seasonality, s_t is the seasonal forecast component, s_{t-m} is the forecast for the previous season and y is the seasonal component smoothing factor.

E. Long Short-Term Memory (LSTM)

The Long Short-Term Memory (LSTM) is one of the most advanced models to forecast time series. A hyperparameter is a variable chosen before optimizing the real model's parameter. In deep learning, there are more hyperparameters like neurons, hidden layers, and learning rate. As a human, we have a hard time handling and visualizing multi-dimensional spaces. Hyperparameter is a procedure of enhancing the performance of the model by designating hyperparameters precise integration. In LSTM, the parameters like timesteps, amount of layers and hidden neurons at every LSTM is improved by hyperparameter. In deep LSTM, the layers lead to less convergence and overfitting, while the hidden layer works equally to the LSTM. Generally, previous historical data is required for predicting time-series data, but the conventional neural network takes current input data into account. LSTM embraces the historical data as well as the past data for an extensive period.

The LSTM has different memory cells in the hidden layer that are simplified through various gates like input, forget and output gate. It is utilized for determining data that is required to be stored. The cell state transfers data from one to another layer. The forgot gate is a primary gate that enables the transfer of required information by the cell state. In this gate, there are two steps of pointwise multiplication and sigmoid layer. The mathematical formula of the forget-gate f_t is shown in Eq. (12),

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right) \quad (12)$$

where, f_t is denoted as the current forget gate with the value being between 0 and 1 and is managed by the sigmoid function σ. The weight matrices are W_f and U_f, b_f is the value of bias, x_t is the value of input, and h_{t-1} is the previous data used for forget gate value calculation.

The next stage is the input gate which is used for processing the data. The mathematical form of the input gate is shown in Eq. (13),

$$i_t = \sigma(W_i X_t + U_i h_{t-1} + b_i) \quad (13)$$

where, i_t is the sigmoid function result and it controls which data to be stored in the memory cell. W_i and U_i are the weight matrices and b_i value is of the bias. These are the parameters adjusted in this input gate.

The last stage is the output gate which determines the actual values of the hidden layers, and the mathematical form is shown in Eq. (14),

$$o_t = \sigma(W_o X_t + U_o h_{t-1} + b_o) \quad (14)$$

Here, o_t is the sigmoid function, the weight matrices are W_o and U_o, while b_o is the value of bias.

1) Sine-Cosine Algorithm (SCA): The Sine-cosine algorithm is a population-based optimization algorithm for improving the movement of an agent toward the best solution, which is determined by the sine and cosine operators. The equation of this operator is used to fluctuate towards the optimal solution to identify the optimal solutions. In this search area, the random variables are used to find the possible global optima. To achieve optimal global solutions, SCA is proven to be more effective than other population-based algorithms. When the sine and cosine functions return values lesser than one or greater than one, the different occasions on the search space are determined.

a) Initial population: For filtering the current best solution, the Sine-Cosine algorithm works with the population. The numerical formula of the initial population is shown in Eq. (15),

$$X(i,j) = X(min,j) + random\,(0,1) * \left(X(max,j) - X(min,j)\right) \quad (15)$$

where, $X(min,j)$ and $X(max,j)$ are lower and upper limits of individuals on dimension j , R is the random number between $(0,1)$. Sine and Cosine function's range is shown in Eq. (16),

$$r_i = a\left(1 - \frac{t}{T}\right) \quad (16)$$

where, t indicates the present iteration, T is the maximum number of iterations, and a is the constant variable. In this equation, r_1 decreases linearly from a to 0.

b) Updating position: Sine and Cosine function's range are shown in Eq. (16) and the updated new candidate solutions are shown in Eq. (19). The current population's best candidate solutions and the best objective value are also updated. The numerical formula for updating positions for both sine and cosine are shown in Eq. (17) and Eq. (18),

$$P_i^{t+1} = P_i^t + r_1 \, sin\,(r_2) * |r_3 P_{best}^t - P_i^t| \quad (17)$$

$$P_i^{t+1} = P_i^t + r_1 \, cos(r_2) * |r_3 P_{best}^t - P_i^t| \quad (18)$$

where, P_i^t is the present candidate position at the t th iteration in the ith dimension and P_{best}^t is the best candidate optimal position at the t th iteration in the i th dimension. r_1, r_2, r_3 and r_4 are the random agents. The $(*)$ is the sign of multiplication. The updated new candidate solutions are shown in Eq. (19),

$$P_i^{t+1} = \begin{cases} P_i^t + r_1 * sin\,(r_2) * |r_3 P_{best}^t - P_i^t|, r_4 < 0.5 \\ P_i^t + r_1 * cos(r_2) * |r_3 P_{best}^t - P_i^t|, r_4 \geq 0.5 \end{cases} \quad (19)$$

where, r_1 is the parameter that determines the search space's next region and increases the investigation of the search space for a higher value, r_2 determines the movement direction and how long the movement should go towards or away from P_{best}^t. r_3 controls the current movement destination effects. To increase the solution diversity values of r_1, r_2 and r_3, they are updated at each iteration, while r_4 is used to switch between cosine and sine functions.

c) Fitness function: To improve the model accuracy, the parameters of LSTM such as the number of hidden neurons (HN), the learning rate (α) and the dropout value (DV) are used for the fitness function. The Sine Cosine optimization algorithm optimizes these three parameters $\theta = \{HN, \alpha, DV\}$ of the LSTM model. The fitness function is set between the predicted and observed values as the RMSE. The eqs. (17) and (18) are merged in (21). The fitness function is shown in (20) and (21),

$$min\, f_{RMSE} = \frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2 \quad (20)$$

$$\text{Subject to} \begin{cases} \hat{y}_i = f_{LSTM}\,(HN, \alpha, DV) \\ HN^{min} \leq HN_{max} \leq HN^{max} \\ \alpha^{min} \leq \alpha \leq \alpha^{max} \\ DV^{min} \leq DV \leq DV^{max} \end{cases} \quad (21)$$

Here, $\hat{y}_i$ is the predicted values and y_i is the observed value. HN^{min}, α^{min}, DV^{min} and HN^{max}, α^{max} , DV^{max} are the upper and lower bounds of the three decision variables.

2) Modified sine-cosine algorithm: In a simple SCA, r_1 is a control parameter which manages the transformation of algorithm from global exploration to local improvement. The local improvement capability of the algorithm is improved by a smaller value r_1 . The global searching capability of algorithm is developed by a larger value r_1. Thus, to manage global exploration and local improvement ability, r_1 is used as a linear decreasing technique of Eq. (16). Based on Eq. (19), the linear and exponential reducing inertia weight and conversion parameter which are utilized in this method, are seen to perform commendably in the ability of the local improvement and global exploration of the model. The updated equation of the individual is shown in Eq. (22), Eq. (23) and Eq. (24),

$$P_{i,j}^{t+1} = \begin{cases} \omega(t) * P_{i,j}^t + r_1 * sin\,(r_2) * \left|r_3 P_{best,j}^t - P_{i,j}^t\right|, r_4 < 0.5 \\ \omega(t) * P_{i,j}^t + r_1 * cos(r_2) * \left|r_3 P_{best,j}^t - P_{i,j}^t\right|, r_4 \geq 0.5 \end{cases} \quad (22)$$

$$\omega(t) = \omega_{max} - (\omega_{max} - \omega_{min}) * \frac{t}{T} \quad (23)$$

$$r_1(t) = a.\, e^{\frac{t}{T}} \quad (24)$$

where, t indicates the present iteration, T is the highest number of iterations and a is the constant variable. $P_{best,j}^t$ is jth individual dimension value at iteration t , $P_{i,j}^t$ is the j th individual dimension score of the i of present iteration, ω_{min} and ω_{max} are the minimum and maximum weights of inertia. The linear reducing inertia weight and exponential reducing conversion parameter are utilized in this work for enhancing the coverage speed and for the searching precision of the algorithm.

a) Neighborhood search of the optimal individual: In a simple SCA, new individuals updating process search directions are done by individuals optimal in the population. The entire algorithm is converted into early convergence when the global optimum individual falls into the local optimum. For reducing the algorithm probability of getting into local optimum, individuals near optimal solution are utilized as a guiding role. Here, the current optimal individuals are replaced through a random individual near to the optimal solution for guiding the algorithm search, thereby improving the algorithm's probability of coming out into the local optimum. The optimal individual neighborhood search is shown in Eq. (25),

$$P_{i,j}^{t+1} = \begin{cases} \omega(t) * P_{i,j}^t + r_1 * sin(r_2) * \\ \left|r_3 P_{best,j}^t * (1 - \lambda * unifrnd(-1,1)) - P_{i,j}^t\right|, r_4 < 0.5 \\ \omega(t) * P_{i,j}^t + r_1 * cos(r_2) * \\ \left|r_3 P_{best,j}^t * (1 - \lambda * unifrnd(-1,1)) - P_{i,j}^t\right|, r_4 \geq 0.5 \end{cases} \quad (25)$$

where, $unifrnd(-1,1)$ is the number of uniform distributions within $(-1,1)$, and λ is distribution coefficient.

b) Greedy levy mutation: In simple Sine-Cosine algorithm, the whole population search direction is controlled by the optimal individuals. For the further prevention of traditional SCA, low efficiency is eliminated and getting into the local optimum in later period. Here, the mutation operation helps an individual to come out of the position of optimal score in population which is previously searched to maintain population diversity. The mutation methodology is shown in Eq. (26),

$$P_{best,j}^{t+1} = P_{best,j}^{t} + \theta(j) * levy * P_{best,j}^{t} \quad (26)$$

where, $\theta(j)$ is the self-adapting variation coefficient, $levy$ is a random number which accepts the levy distribution and $P_{best,j}^{t}$ is the jth individual dimension value at iteration t.

For simple sine-cosine algorithm, creating the initial population time complexity is $O(n)$, performing sine-cosine operation time complexity is $O(T * n * d)$, and the processing of cross-border is $O(T * n)$. So, the simple sine-cosine algorithm's time complexity is $O(n) + O(T * n) + O(T * n * d)$. In the modified sine-cosine algorithm, creating the initial population time complexity is $O(n)$ for calculating $\omega(t)$ time complexity, and r_1 is $O(2 * T)$ for performing sine-cosine operation with time complexity $O(T * n * d)$, the processing of cross-border is $O(T * n)$ and the greedy levy mutation's time complexity is $O(T * d * n)$. So, the modified sine-cosine algorithm's time complexity is $O(n) + O(2 * T) + O(T * n * d) + O(T * n) + O(T * d * n) = O(n) + O\left((n + 2) + T\right) + O(2 * T * d * n)$. So, the modified sine-cosine algorithm's time complexity is lesser than the simple sine-cosine algorithm.

IV. Experimental Results

The LSTM-ISCA is simulated using the tool Python 3.10 on system configuration of windows 10 OS, 16GB RAM and intel i7 processor. The R2, MAE, MSE and RMSE are considered to estimate LSTM-ISCA the performance which are estimated from forecasted yields of each year for all districts. They are mathematically shown in Eq. (27), (28), (29) and (30), respectively:

$$R^2 = 1 - \sum_{i=1}^{n} \left(\frac{(y_i - y_{di})^2}{(y_{di} - y_m)^2}\right) \quad (27)$$

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - y_{di}| \quad (28)$$

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - y_{di})^2 \quad (29)$$

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - y_{di})^2} \quad (30)$$

where, n, y_i, y_{di} and y_m are the number of points, predicted score, real score, and mean of real score.

A. Quantitative Analysis

The performance of LSTM-ISCA is explained in this section, with respect to achievable sum rate. Table I demonstrates the results of the LSTM model, whereas Table II shows the results of ISCA, and Table III exhibits the outcomes of LSTM-ISCA. The LSTM and LSTM-SCA models are compared with LSTM-ISCA model that achieves better performance as compared to other models.

In Table I, the effectiveness of the LSTM model is validated and the experimentation is performed with various deep learning techniques: Convolutional Neural Network (CNN), Generative Adversarial Network (GAN), Recurrent Neural Network (RNN), and Deep Neural Network (DNN). The LSTM model achieves 0.43, 0.131, 0.054 and 0.232 of R2, MAE, MSE and RMSE, respectively. A graphical representation of the LSTM model is represented in Fig. 2.

TABLE I. Quantitative Analysis of Various Classifiers

Methods	MSE	R2	RMSE	MAE
CNN	0.062	0.49	0.248	0.139
GAN	0.055	0.44	0.234	0.132
RNN	0.058	0.46	0.240	0.135
DNN	0.059	0.47	0.242	0.136
LSTM	0.054	0.43	0.232	0.131

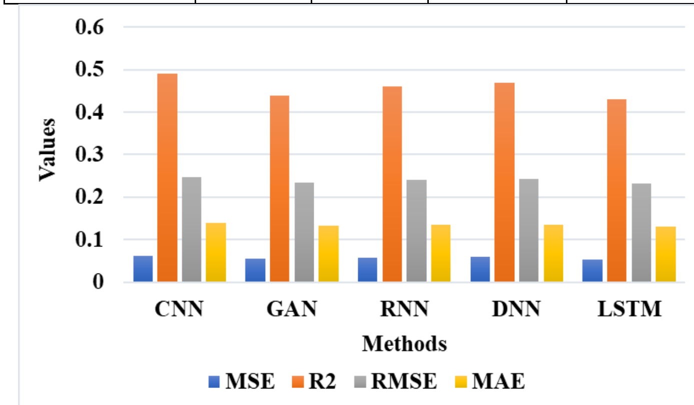

Fig. 2. Performance of various classifiers.

TABLE II. Quantitative Analysis of Various Optimization

Methods	MSE	R2	RMSE	MAE
PSO	0.056	0.44	0.236	0.133
WOA	0.054	0.43	0.232	0.131
SGD	0.053	0.41	0.230	0.130
SCA	0.051	0.40	0.225	0.128
ISCA	0.049	0.38	0.221	0.126

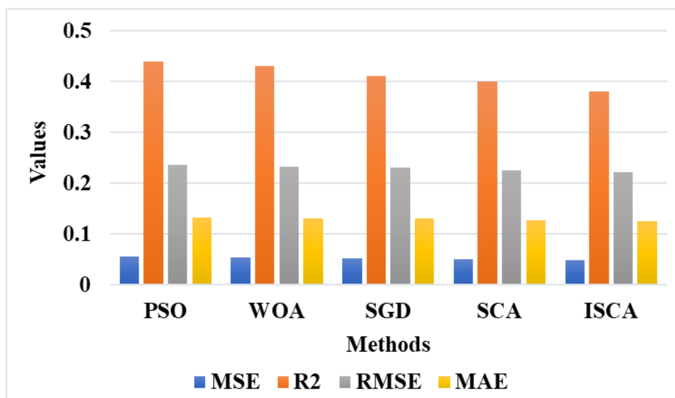

Fig. 3. Performance of various optimizations.

In Table II, the ISCA's effectiveness is validated and experimentation is performed with various optimization algorithms like Particle Swarm Optimization (PSO), Stochastic Gradient Descent (SDG), Whale Optimization Algorithm (WOA), and SCA. The ISCA achieves 0.38, 0.126, 0.049 and 0.221 of R2, MAE, MSE and RMSE. Fig. 3 represents the graphical representation of the ISCA model.

TABLE III. QUANTITATIVE ANALYSIS OF THE PROPOSED LSTM-ISCA MODEL

Methods	MSE	R2	RMSE	MAE
LSTM	0.054	0.43	0.232	0.131
LSTM-SCA	0.051	0.40	0.225	0.128
LSTM-ISCA	0.049	0.38	0.221	0.126

In Table III, the effectiveness of LSTM with ISCA model is validated and the experimental is performed with LSTM and LSTM-SCA. The LSTM-ISCA model achieves better performance when compared to other models. The LSTM-ISCA model achieves 0.38, 0.126, 0.049 and 0.221 of R2, MAE, MSE and RMSE. A graphical representation of LSTM-ISCA model is shown in Fig. 4.

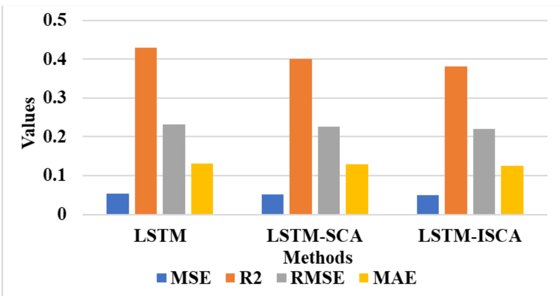

Fig. 4. Performance of the proposed LSTM-ISCA model.

B. Comparative Analysis

The comparison of LSTM-ISCA is demonstrated in this section with attainable sum rate. The previous researches like [17-20] are utilized for estimating the proposed model's efficiency as shown in Table IV. Iniyan and Jebakumar [20] considered 456 samples across 105 different areas and the experimental analysis showed that the presented model attained 7.431, 121.123, and 11.005 of MAE, MSE and RMSE, correspondingly. Oikonomidis et al. [21] took into account 25345 samples across 9 different states, where the experimental results confirmed that the presented obtained 0.87, 0.199, 0.071, and 0.266 of R2, MAE, MSE and RMSE, correspondingly. Joshua et al. [22] examined 280 samples across 50 fields and the model attained 0.986, 0.129, 0.052, and 0.229 of R2, MAE, MSE and RMSE. Therefore, it is seen that the LSTM-ISCA model achieves superior performance, as opposed to other models.

TABLE IV. EXPERIMENTAL RESULTS OF LSTM-ISCA MODEL

Author	No. of data	R2	MAE	MSE	RMSE
Iniyan and Jebakumar [20]	456 samples across 105 different areas	-	7.431	121.123	11.005
Oikonomidis et al. [21]	25345 samples across 9 different states	0.87	0.199	0.071	0.266
Joshua et al. [22]	280 samples across 50 fields	0.986	0.129	0.052	0.229
Proposed LSTM-ISCA	84000 samples across 10 different districts	0.38	0.126	0.049	0.221

C. Discussion

The advantages of the LSTM-ISCA and limitations of the existing researches are deliberated in this section. The Iniyan and Jebakumar [20] considered only 456 samples across 105 areas which achieved 7.431 of MAE due to the limited samples. Oikonomidis et al. [21] considered 25343 samples across nine states, achieving 0.199 of MAE due to the long-term viable option increasing the coconut production, as well as the productivity of coconut lands. Joshua et al. [22] considered 280 samples across 590 fields, achieving 0.129 of MAE because the crop data requiring huge preprocessing to fit the baseline requirements. To overcome these issues, this research proposes LSTM-ISCA. The time-series data are given as the input to the LSTM classifier to classify the yield production and the LSTM model is tuned by hyperparameter using ISCA. The ISCA improves the coverage speed and searching precision of the algorithm.

V. CONCLUSION

In this research, a LSTM-ISCA is proposed for effective coconut yield production. The proposed model comprises multiple objective functions such as linear interpolation and Holt's Winter Seasonal method for effective classification. ARIMA model is used for converting stationary time series data from non-stationary data, by applying differences and predicted values from the previous historical data. The Holt-Winters seasonal method is introduced for the exponential smoothing of the seasonal data. Then LSTM is tuned by the ISCA which uses an exponential convergence strategy and linear decreasing inertia weight, thereby improving the convergence speed and algorithm's search precision. From the performance analysis, it is concluded that the LSTM-ISCA provides better performance than other models. In the future, Bayesian optimization can be used for hyperparameter tuning over the ISCA algorithm to enhance the prediction performance.

REFERENCES

[1] T.D. Nuwarapaksha, U.N. Rajapaksha, J. Ekanayake, S.A. Weerasooriya, and A.J. Atapattu, "Exploring the Economic Viability of Integrating Jamnapari Goat into Underutilized Pastures under Coconut Cultivations in Coconut Research Institute, Sri Lanka," *Biology and Life Sciences Forum*, vol. 27, no. 1, p. 27, 2023.

[2] C.B. MacEachern, T.J. Esau, A.W. Schumann, P.J. Hennessy, and Q.U. Zaman, "Detection of fruit maturity stage and yield estimation in wild blueberry using deep learning convolutional neural networks," *Smart Agric. Technol.*, vol. 3, p. 100099, 2023.

[3] I.Y. Dharmegowda, L.M. Muniyappa, P. Siddalingaiah, A.B. Suresh, M.P. Gowdru Chandrashekarappa, and C. Prakash, "MgO nano-catalyzed biodiesel production from waste coconut oil and fish oil using response surface methodology and grasshopper optimization," *Sustainability*, vol. 14, no. 18, p. 11132, 2022.

[4] C.R.K. Samarasinghe, M.K. Meegahakumbura, D.P. Kumarathunge, H.D.M.A.C. Dissanayaka, P.R. Weerasinghe, and L. Perera, "Genotypic selection approach made successful advancement in developing drought tolerance in perennial tree crop coconut," *Sci. Hortic.*, vol. 287, p. 110220, 2021.

[5] A.F. Cardoso, W.X. Ferreira, G.L.S. de Castro, P.M.P. Lins, M.A.S. Dos Santos, and G.B. da Silva, "Cost Reduction in the Production of Green Dwarf Coconut Palm Seedlings Biostimulated with Bacillus cereus," Indian J. Microbiol., 2024.

[6] A.M. Lad, K.M. Bharathi, B.A. Saravanan, and R. Karthik, "Factors affecting agriculture and estimation of crop yield using supervised learning algorithms," *Materials Today: Proceedings*, vol. 62, pp. 4629-4634, 2022.

[7] T. Prasert, and V. Rungreunganun, "Thai Coconut Price Forecasting using Arima Model," *International Journal of Advanced Research in Engineering and Technology (IJARET)*, vol. 12, no. 1, pp. 950–961, 2021.

[8] N. Bakhashwain, and A. Sagheer, "Online tuning of hyperparameters in deep LSTM for time series applications," *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 1, pp. 212–220, 2021.

[9] T. Peng, C. Zhang, J. Zhou, and M.S. Nazir, "An integrated framework of Bi-directional long-short term memory (BiLSTM) based on sine cosine algorithm for hourly solar radiation forecasting," *Energy*, vol. 221, p. 119887, 2021.

[10] H. Novarianto, "Estimating Coconut Production and Productivity of Local Tall in Taliabu Island Using Drone and Sampling Population," *Coconut Research & Development*, vol. 38, pp. 22–29, 2022.

[11] S.M.M. Samarakoon, L.H.P. Gunaratne, and H.L.J. Weerahewa, "Determinants of Coconut Production in Large Scale Coconut Plantations in Sri Lanka: A Quantile Regression Approach," *Sri Lankan Journal of Agricultural Economics*, vol. 21, no. 1, pp. 1–16, 2020.

[12] M.F.B. Pramono, and B.Z. Arifin, "Design of Dryer Coconut for Copra Production Using Fuzzy Logic Control," In: *2020 International Conference on ICT for Smart Society (ICISS)*, Bandung, Indonesia, pp. 1–7, 2020. doi: 10.1109/ICISS50791.2020.9307589.

[13] M.N. Hadi, "Implementation of Traditional Risk Management as Loss Prevention in Coconut Production Results," *AKADEMIK: Jurnal Mahasiswa Ekonomi & Bisnis*, vol. 2, no. 2, pp. 92–102, 2022.

[14] K.R. Karunakaran, and N. Narmadha, "Growth Performance of Coconut Production in Global Scenario: A Quin-decadal Analysis," *Journal of Experimental Agriculture International*, vol. 44, no. 11, pp. 7–15, 2022.

[15] B. Das, B. Nair, V. Arunachalam, K. V. Reddy, P. Venkatesh, D. Chakraborty, and S. Desai, "Comparative evaluation of linear and nonlinear weather-based models for coconut yield prediction in the west coast of India," *Int. J. Biometeorol.*, vol. 64, no. 7, pp. 1111–1123, 2020.

[16] K.B. Hebbar, P.S. Abhin, V. Sanjo Jose, P. Neethu, A. Santhosh, S. Shil, and P.V. Prasad, "Predicting the potential suitable climate for coconut (Cocos nucifera L.) cultivation in India under climate change scenarios using the MaxEnt model," *Plants*, vol. 11, no. 6, p. 731, 2022.

[17] R.B. Tuckeldoe, M.K. Maluleke, and P.J.S.R. Adriaanse, "The effect of coconut coir substrate on the yield and nutritional quality of sweet peppers (Capsicum annuum) varieties," *Sci. Rep.*, vol. 13, no. 1, p. 2742, 2023.

[18] V.H.T. Pham, J. Kim, J. Shim, S. Chang, and W. Chung, "Coconut mesocarp-based lignocellulosic waste as a substrate for cellulase production from high promising multienzyme-producing Bacillus amyloliquefaciens FW2 without pretreatments," *Microorganisms*, vol. 10, no. 2, p. 327, 2022.

[19] D. Paudel, H. Boogaard, A. de Wit, S. Janssen, S. Osinga, C. Pylianidis, and I.N. Athanasiadis, "Machine learning for large-scale crop yield forecasting," *Agric. Syst.*, vol. 187, p. 103016, 2021.

[20] S. Iniyan, and R. Jebakumar, "Mutual information feature selection (MIFS) based crop yield prediction on corn and soybean crops using multilayer stacked ensemble regression (MSER)," *Wireless Pers. Commun.*, vol. 126, no. 3, pp. 1935–1964, 2022.

[21] A. Oikonomidis, C. Catal, and A. Kassahun, "Hybrid deep learning-based models for crop yield prediction," *Applied Artificial Intelligence*, vol. 36, no. 1, p. 2031822, 2022.

[22] V. Joshua, S.M. Priyadharson, and R. Kannadasan, "Exploration of Machine Learning Approaches for Paddy Yield Prediction in Eastern Part of Tamilnadu," *Agronomy*, vol. 11, no. 10, p. 2068, 2021.