ISSN: 1803-3814 (Printed), 2571-3701 (Online)
https://doi.org/10.13164/mendel.2023.2.295

Exploring Hybrid Models For Short-Term Local Weather Forecasting in IoT Environment

Toai Kim Tran[1,2,✉], Roman Senkerik[2,✉], Vo Thi Xuan Hanh[1], Vo Minh Huan[1], Adam Ulrich[3], Marek Musil[3], and Ivan Zelinka[2]

[1]Ho Chi Minh University of Technology and Education, Vietnam
[2]VSB-Technical University of Ostrava, Ostrava-Poruba, Czech Republic
[3]Tomas Bata University, Zlin, Czech Republic

toaitk@hcmute.edu.vn✉, roman.senkerik@vsb.cz ✉, hanhvtx@hcmute.edu.vn, huanvm@hcmute.edu.vn, ivan.zelinka@vsb.cz

Abstract

This paper explores using and hybridizing simple prediction models to maximize the accuracy of local weather prediction while maintaining low computational effort and the need to process and acquire large volumes of data. A hybrid RF-LSTM model is proposed and evaluated in this research paper for the task of short-term local weather forecasting. The local weather stations are built within an acceptable radius of the measured area and are designed to provide a short period of forecasting - usually within one hour. The lack of local weather data might be problematic for an accurate short-term valuable prediction in sustainable applications like agriculture, transportation, energy management, and daily life. Weather forecasting is not trivial because of the non-linear nature of time series. Thus, traditional forecasting methods cannot predict the weather accurately. The advantage of the ARIMA model lies in forecasting the linear part, while the SVR model indicates the non-linear characteristic of the weather data. Both non-linear and linear approaches can represent the combined model. The hybrid ARIMA-SVR model strengthens the matched points of the ARIMA model and the SVR model in weather forecasting. The LSTM and random forest are both popular algorithms used for regression problems. LSTM is more suitable for tasks involving sequential data with long-term dependencies. Random Forest leverages the wisdom of crowds by combining multiple decision trees, providing robust predictions, and reducing overfitting. Hybrid Random forest-LSTM potentially leverages the robustness and feature importance of Random Forest along with the ability of LSTM to capture sequential dependencies. The comparison results show that the hybrid RF-LSTM model reduces the forecasting errors in metrics of MAE, R-squared, and RMSE. The proposed hybrid model can also capture the actual temperature trend in its prediction performance, which makes it even more relevant for many other possible decision-making steps in sustainable applications. Furthermore, this paper also proposes the design of a weather station based on a real-time edge IoT system. The RF-LSTM leverages the parallelized characteristics of each decision tree in the forest to accelerate the training process and faster inferences. Thus, the hybrid RF-LSTM model offers advantages in terms of faster execution speed and computational efficiency in both PC and Raspberry Pi boards. However, the RF-LSTM consumes the highest peak memory usage due to being a combination of two different models.

Keywords: *Weather Forecast, Short-term Prediction, Machine Learning, ARIMA, SVR, Hybrid ARIMA-SVR, LSTM, Random Forest, Hybrid RF-LSTM, Weather Station Design, IoT.*

Received: 16 November 2023
Accepted: 14 December 2023
Online: 14 December 2023
Published: 20 December 2023

1 Introduction

Weather forecasting is an important activity that predicts the weather conditions in a particular location at a specific time. If we know the weather conditions in the next few hours, we can suitably decide on activities and precautions. The weather forecasting system predicts the atmosphere condition at a given location and time, usually based on the principle of a precipitation radar, weather satellite, or sensor system [45, 35, 28, 23]. The precipitation radar sends out radio waves. The radio wave is reflected from raindrops or liquids in the air environment [45, 35], and the receiver captures the reflection. However, the received signal might be less accurate because of abnormal feedback from various obstacles. Moreover, the temperature, humidity, and solar radiation are different from various conditions of the atmosphere, ocean, and land [28]. The weather radar shows the location of rainfall at a specific time. The satellite uses a camera to get images of the earth's weather [23]. These systems can measure the temperature of the earth's surface. The approaches to collecting surface air temperature and humidity data are limited to a large area on this planet because of the sparse weather stations.

The long-term forecast systems use all the methods mentioned above to gather data for at least twenty-four hours [46]. The weather station based on sensors can be installed widely at a given location to collect quantitative data about the current state of the atmosphere. However, the weather stations are located very far, from 2 to 3 km [46, 18]. Thus, it is limited by high-resolution temporal and spatial measurements. Additionally, the collected data is unsuitable for local sites that aim to observe the weather in a short period. Moreover, the weather data belongs to national weather officers and is owned by the government or private companies. Therefore, it is also incorporated into numerical prediction models to improve forecast accuracy.

There are many algorithms for weather forecasting so far. Data mining is a prevalent technique for weather forecasting [22, 34, 42]. It allows appropriate predictions to make decisions. The data mining techniques have been applied to various published types of research [7]. The data mining was used to classify and predict whether the weather would be sunny, rainy, or cloudy on a specific day [12, 22, 34, 42]. Naïve Bay and Chi-square algorithms were utilized for the classification methods [5]. The data mining techniques have taken the parameters of the current outlook, temperature, humidity, and wind conditions to predict the weather after analyzing these data in the database. In Ref. [19], the SPRINT algorithm was based on the principle of the decision tree. This methodology is used to compare the historical data and to analyze the relation of climate parameters such as temperature, wind speed, and humidity. The long short-term memory (LSTM) method is used to forecast weather status to harvest renewable energy. The model is employed in the EDGE platform [9]. The problem of weather forecasting lies in the non-linear behavior of nature, climate change, and frequent weather changes. The ARIMA model, Artificial Neural Networks (ANN), and Support Vector Machines (SVM) or Support vector regression (SVR) have been used to predict the temperature, wind, and humidity [13, 1, 30, 27]. Researchers also use a back-propagation neural network technique to forecast the temperature for a future time and given location [3]. The model proves more efficient than numerical differentiation. In [20], an ensemble neural network creating multiple neural network models improves the accuracy of the prediction. Each neural network model uses a different architecture (consisting of different numbers of hidden layers and neurons). A model can be discarded if it does not achieve an accuracy rate.

The short-term weather forecast in a local area is necessary for several sustainable applications, such as agriculture, transportation, and solar power management, as well as our daily life [11, 48, 8]. The short-term local forecast system is responsible for forecasting an hour or less in a short range that covers an acceptable radius. For example, a short-term local weather forecast system implements air humidity and temperature prediction that impacts agriculture to avoid water waste. The development of edge computing platforms which is the joint utilization of the Internet of Things and machine learning provides a significant improvement in the sustainability of the weather system. The short-time weather system helps to monitor and compute the system weather status timely and locally [10, 43]. Weather forecast accuracy is essential for irrigation systems that make actions for pumping water to the ground. Knowing whether it rains in the next hour for such systems is crucial. The accuracy of forecasting helps the system save power and resources when deciding whether to pump the water or not. We strive to catch credible signs of rain to take proper action in the next few hours. Making an exact prediction is one of the significant challenges that it is difficult to forecast in advance. The main problems in weather forecasting are permanent changes in the environment; abrupt changes in the environment are tied to several problems, such as accurate forecasting, lack of methods in big data, and real-time processing [12].

Collecting the dataset of short-term weather forecasts is usually a problem in a local area. The local dataset is often interpolated from large-area observations in resolutions of several kilometers. Various techniques have been developed such as PRISM, Daymet, GridMET, and CHRITS using techniques of the statistical model or machine learning-based model to perform this interpolation [43]. In this research paper, weather station hardware is designed to collect the temperature and humidity of the environment. By doing so, the dataset is very precise in the local sites. A few researchers localized dataset points by combining sensors and satellite imagery [43, 44], which may be quite complicated in real applications. An integrated database was created by combining remote sensing data and data collected by ground weather stations. The remote sensing data was collected by satellite images.

In paper [6], the research compared the forecasting performance of ARIMA and ANN in wind speed forecasting on the South Coast of Oaxaca, Mexico. The ARIMA model mostly showed better and more accurate forecasts than the ANN model. Researchers in [39, 4] discussed that the ANN and ARIMA models achieved good forecasting performance in several real-world applications and time series prediction. The experimental results revealed that the ARIMA model performs better forecasting linear time series, while the ANN showed better forecasting of non-linear time series. Weather prediction is a non-linear task due to humidity, wind speed, sea level, and air density changes. The SVM/SVR and ANN are used for robust weather prediction purposes. ARIMA model is one of the most widely used linear models in time series forecasting. However, the ARIMA model cannot easily capture non-linear patterns. The SVM/SVR model has been applied to solve non-linear regression estimation. Thus, a hybrid method combining the strengths of the ARIMA model and the SVM/SVR model has been investigated in various fields such as stock price prediction [24, 32], the remaining service life of aircraft engines prediction [31], and the $PM_{2 \cdot 5}$ concentrations time series data set (a heterogeneous data set mixed one-dimension series data) [41], temperature [29]. The researchers also publish the combination methods that were used to forecast climate change by machine learning methods such as support vector regression, random forest, and K-nearest neighbors [36]. It is known that the LSTM model has become popular to be used to predict the time series data [37, 16, 2]. The LSTM model is used to predict time-series temperature data to accurately sense the changes occurring in temperature levels [47]. A convolution neural network (CNN) and long short-term memory (LSTM) are integrated into a network model for hourly temperature prediction. The CNN reduces the dimensionality of the time-series data, while LSTM captures the long-term memory of the massive temperature time-series data [17]. The Random Forest algorithm is a type of machine learning algorithm that combines multiple decision trees to predict outcomes [14]. Each decision tree provides a classification, and the algorithm selects the classification with the most votes across all trees in the forest. To make a prediction, the algorithm calculates the average or mean of the outputs from the different trees. Random Forest is suitable for solving both regression and classification problems, and it is often used to address complex problems that are difficult to solve with a single decision tree [26]. Additionally, according to the statistical parameters and sensitivity analysis, the Random Forest method was found to be more effective than the M5P model in predicting the California Bearing Ratio (CBR) value [38].

This proposed research combines the LSTM with the random forest model. A hybrid Random forest-LSTM model to predict the short-time local weather by leveraging the strengths of both algorithms. The RF is good at handling complex relationships and non-linear patterns, while LSTMs are designed to capture sequential dependencies and temporal patterns. The diversity introduced by combining an RF and an LSTM model can help mitigate the individual weaknesses of each model. By combining these strengths, we can potentially handle a wider range of patterns in your time series data. The observed weather data is obtained from weather stations designed as sensor nodes to collect the atmosphere data each hour. By doing so, the proposed two-stage approach can minimize the input dimension. The hybrid random forest and LSTM demonstrate that the proposed model outperforms other models such as the LSTM, random forest, SVR, ARIMA, and hybrid SVR-ARIMA. It also demonstrates a performance improvement compared to traditional model selection methods which are LSTM, ARIMA, and random forest. The hybrid random forest-LSTM model has not previously been applied to forecasting the short-time local weather. The accuracy of the predictive model was assessed using three continuous error measurement metrics: Mean Absolute Error (MAE), Maximum Error, and Root Mean Square Error (RMSE) [14]. These metrics are commonly used to evaluate the accuracy of predictive models. As the number of training data increased, the model's performance improved, resulting in smaller errors between the predicted values and the actual values. This performance improvement indicates that the model became more accurate in making predictions, by considering these metrics. Therefore, it can be inferred that the predictive model demonstrated better performance with an increase in training data, leading to improved accuracy in its forecasts [14]. The use of standard statistical parameters implies that various metrics were employed to evaluate the model's performance, such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), R-squared value, or other appropriate evaluation metrics for regression tasks. By comparing these metrics between the models, it can be determined which model performs better.

This paper investigates the data type mentioned above and uses an analysis performed by the hybrid model. Section 2 offers the background of the ARIMA, SVM/SVR, Random forest, LSTM, and hybrid models which are used to design the IoT edge system in a weather forecasting application. Section 3 provides the designed weather station hardware. The next step of this research is presented by validation results, which prove our comparison results and evaluation of the accuracy of prediction algorithms in Section 4. Finally, Section 5 proposes a conclusion and discussion.

2 Models in Weather Forecasting

Figure 1 shows the flowchart of the weather data forecast using various models such as ARIMA, SVM/SVR, hybrid ARIMA-SVR, LSTM, RF, and hybrid RF-LSTM model. Firstly, the weather data is collected by sensors and saved to a CSV file. The preprocessing step of the data is used to simplify the prediction problem and to increase the model's accuracy. Then, the models are applied to training data. Finally, accuracy/quality measurements for results are calculated as the mean error between test values and real data.

This section briefly describes the most important basic facts and mathematical background of the components forming the hybrid SVM/SVR-ARIMA and hybrid RF-LSTM model.

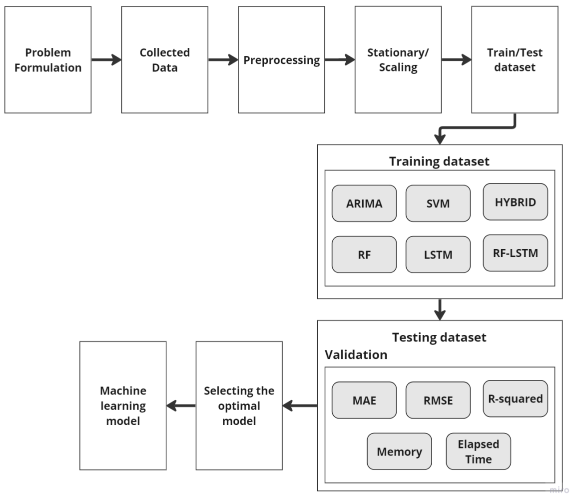

Figure 1: The flow diagram of the selecting optimal models.

2.1 ARIMA Model

ARIMA model is very often used for analyzing and forecasting time series data. ARIMA stands for AutoRegressive Integrated Moving Average and is a generalization of simpler AutoRegressive Moving Average with the notion of integration. Two linear time series models are used widely, such as Autoregression (AR) and Moving Average (MA). ARIMA mathematical model is a combination of AR (p), Integration, and MA(q) models [1]. ARIMA model has their strengths in capturing linear patterns, and it can be useful for short- to medium-term forecasting. However, they might struggle with complex non-linear patterns or longer-term forecasts. Depending on the nature of time-series data, the ARIMA model might need to combine with other techniques to improve performance.

2.2 Support Vector Regression Model

The support Vector Machine is a technique used to overcome classification problems and regression estimation. Support vector regression (SVR) is applied to solve the regression problem. The SVR is a method based on artificial intelligence to improve forecasting accuracy. The SVR attempts to minimize the generalization error boundary to achieve generalized performance. The SVR creates a decision boundary that separates n-dimensional space into classes so that we can put new data points into the correct category in the future. The computation of the SVR is based on the linear regression function in a high-dimensional feature space where input data is mapped through a non-linear process [1].

The major limitation of the ARIMA model as mentioned earlier is not considering the load factors with non-linear patterns. The idea of the SVR algorithm is to find a hyperplane f(x) with a specific deviation (ε) from the input training in the form of an equation (1).

$$f(x) = y = \omega \cdot x + b \tag{1}$$

The optimal problem in SVR is to find ω and b such that the margin reaches the maximum value at input training to the $f(x)$. The regression problem is transformed into an optimization function (2).

$$\min \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{m}(\xi_i + \xi_i^*) \tag{2}$$

With constrain conditions of optimization function (3).

$$\begin{cases} y_i - \omega^T x_i - b \le \varepsilon + \xi_i \\ \omega^T x_i + b - y_i \le \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \ge 0, i = 1, \ldots, m \end{cases} \tag{3}$$

Where C is the parameter determining penalty degree and $C > 0$, ω is the weight, b is the mapping parameter, ε is the loss function, ($\varepsilon > 0$). The above hyperplane determination is assumed under ideal conditions when the input training has a margin of less than or equal to ε. Therefore, in the case of datasets with confusing points, these will not satisfy the above conditions, and no solution to the problem will be found. For those cases, we need to use slack variables $\xi_i \ge 0$. The slack variables present the distance from the actual values to the corresponding boundary values. The slack variables ξ_i and ξ_i^*, correspond to upper and lower deviations, respectively. When the data problem is non-linear, we have to use the kernel that maps the data to a more dimensional space to represent the data in an easier computational form. Calculating each data point otherwise takes more memory and time in higher dimensional space. To make this calculation more accessible, we use kernel functions. The SVR model uses the radial basis (kernel) function (RBF) in the form of an equation (4) [33].

$$K(x, y) = e^{-y\|x-y\|^2} \tag{4}$$

2.3 The Hybrid ARIMA-SVM/SVR Model

The different hybrid prediction models have been studied extensively in many various types of research [32, 31, 25]. A hybrid prediction model can cope with both linear and non-linear predictions, which is a good choice for weather or financial market predictions. The hybrid model (Z_t) can be represented as in (5).

$$Z_t = Y_t + N_t \tag{5}$$

Where Y_t is the linear part, N_t is the non-linear part. Both Y_t and N_t are predicted from the datasets. Consequently, ε_t represents the error at time t obtained from the linear model (6).

$$\varepsilon_t = Z_t - \tilde{Y}_t \tag{6}$$

Where $\tilde{Y}_t$ is the predicted data from the linear model at time t. These errors will be predicted from the non-linear model (SVR) and can be expressed as (7).

$$\varepsilon_t = f(\varepsilon_{t-1}, \varepsilon_{t-2}, \ldots, \varepsilon_{t-n}) + \Delta_t \tag{7}$$

Where f is the non-linear function generated by the SVR model and Δ_t is the random error. Finally, the model combines both the linear and the non-linear function, as demonstrated in the equation (8).

$$\tilde{Z}_t = \tilde{Y}_t + \tilde{N}_t \quad (8)$$

Where $\tilde{N}_t$ is the predicted result from a non-linear model. In the proposed hybrid model, the ARIMA model will handle the linear part, and the SVR model will handle the non-linear part. The ARIMA model is used to filter the linear patterns of the dataset. The error terms of the ARIMA model are applied to the SVR model in the hybrid model.

2.4 The LSTM Model

Long Short-Term Memory (LSTM) is an improvement of the Recurrent Neural Networks that allows learning of long correlations. It is designed to remember information over extended periods, making it suitable for tasks involving sequences, such as natural language processing, speech recognition, and time series analysis. The LSTM is designed to address the vanishing gradient problem that occurs in traditional RNNs, allowing them to effectively capture long-term dependencies in sequential data [37, 16].

The main components of an LSTM cell are as follows: Input Gate, Forget Gate, Cell State, Output Gate, and Hidden State. The Input Gate controls the flow of new information into the cell. It decides how much of the new input should be stored in the cell state. The Forget Gate determines what information to discard from the cell state. It selectively removes irrelevant or outdated information from the previous cell state. The Cell State acts as a conveyor belt, transporting relevant information across time steps. It retains useful information and passes it through the sequence. The Output Gate decides how much of the cell state should be used to generate the output or hidden state for the current time step. The Hidden State is also known as the output state. It represents the information that is being propagated to the subsequent LSTM cell or used for the final prediction. The LSTM cell's internal operations can be described using mathematical equations, which govern the flow of information and control the gating mechanisms. It is important to note that the equations can vary slightly depending on the specific LSTM variant being used. During training, an LSTM network learns to adjust the weights and biases of these gates and memory cell operations through backpropagation. LSTM networks have proven to be very effective in handling long-range dependencies in sequential data, and have become an essential tool in various machine learning and deep learning applications [37, 16].

2.5 The Random Forest Model

The random forest algorithm includes many decision trees, each tree is built using the Decision Tree algorithm on different data sets and using different subsets of features [21]. The prediction results of the random forest algorithm are aggregated from the decision trees. The random forest algorithm consists of many decision trees, each decision tree has random elements, taking random data and random attributes to build a decision tree. Since each decision tree in the random forest algorithm does not use all the training data, nor does it use all the features of the data to build the tree, each tree may make a bad prediction. However, the result of the random forest algorithm is aggregated from many decision trees, so the information from the trees complements each other, leading to the model with good prediction results. The key advantage of the random forest is less susceptible to overfitting than individual decision trees, making it more robust and generalizable to new data. It provides a feature importance score, indicating the relative importance of each feature in the prediction process. Random forests can efficiently handle large datasets and high-dimensional feature spaces. The construction of individual decision trees can be parallelized, leading to faster training times on multicore processors. Random Forest has become a popular and powerful algorithm in machine learning, and it has been successfully applied to various real-world problems such as classification, regression, and feature selection tasks.

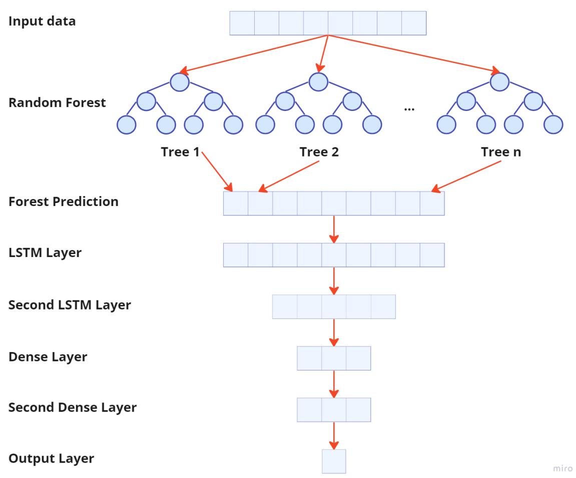

Figure 2: The diagram of the hybrid RF-LSTM model.

2.6 The Hybrid Random Forest-LSTM Model

Combining LSTM (Long Short-Term Memory) and random forest is a hybrid approach that leverages the strengths of both techniques. The idea is to use random forest for capturing features in sequential data and then use LSTM to make predictions based on the extracted features from the random forest model.

Figure 2 shows the diagram of the hybrid RF-LSTM model.

The sequential data is fed into the random forest model, which is designed to capture complex patterns and non-linear relationships. The random forest model uses the lagged features as inputs and the target variable as the output features. The LSTM excels at modeling sequential dependencies. The LSTM generates a hidden state (output state) at each time step of features, which represents the extracted features. For each sequence in the dataset, the LSTM model is used to obtain the hidden states for all time steps. The hid-

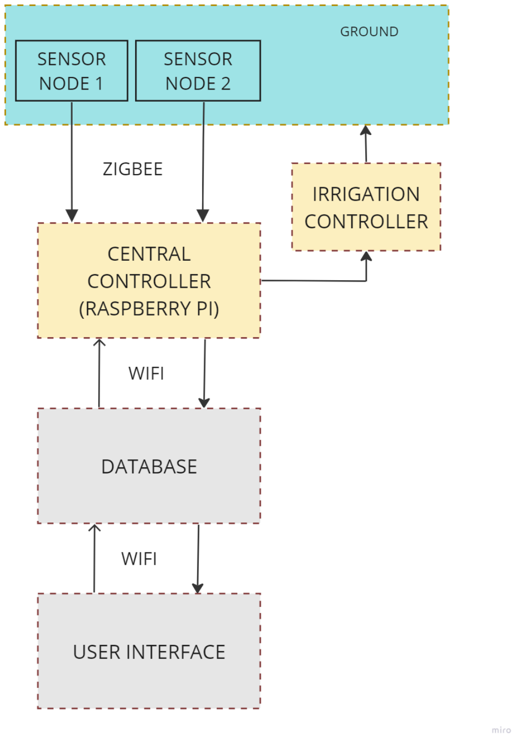

Figure 3: The hardware diagram.

den states act as feature vectors, which capture the learned representations of the sequential data. The LSTM model can effectively capture temporal patterns and dependencies in the sequential data, which helps improve the overall generalization of the hybrid model. LSTM learns meaningful features from the sequential data, which can enhance the feature representation for Random Forest, potentially leading to better performance. The hybrid model is less likely to overfit compared to using a standalone LSTM or Random Forest model, as both components contribute to reducing the risk of overfitting. The RF captures short-term series data well, while LSTM captures long-term trends. The hybrid model can provide accurate predictions across different time horizons. Each model has its limitations. RF might struggle with non-linear sequences, while LSTM might overfit on smaller datasets. A hybrid approach can mitigate these weaknesses and lead to a more robust model that generalizes better to new data. However, it's important to note that building and tuning such hybrid models may require additional computational resources and careful parameter tuning to achieve optimal performance.

3 Weather Station Hardware Design

As shown in Figure 3, the system includes Sensor Nodes that measure the temperature and send data to the Central Controller. The Central Controller uses the machine learning algorithm to train temperature data and forecast the successive temperature data. The suitable tasks are sent back to the Irrigation controller that performs actions such as controlling irrigation. The Central Controller also sends the forecasted data features to the database. The user gets forecasted data for display and interacts with users.

Each block in this diagram takes responsibility as follows. The Central Controller gets data from Sensor Nodes through the Zigbee transmission protocols and sends both real and forecasted data to the database block. Meanwhile, the Central Controller will transmit the tasks to the Irrigation Controller Block. The data transmission protocol is an intermediate point for receiving data from the Sensor Nodes and transferring data to the Central Controller. Sensor Node 1 and Sensor Node 2 handle the obtained data from sensor modules. The Irrigation Controller Block turns on and turns off the irrigation valve at different times. The database Block stores manages, and evaluates the data. The user interface displays real and forecasted data for supporting interaction. The device uses a DHT22 module with a temperature accuracy of +/- 0.5^oC. The system uses the Zigbee module to transmit the data between the Central Controller and the Sensor Nodes.

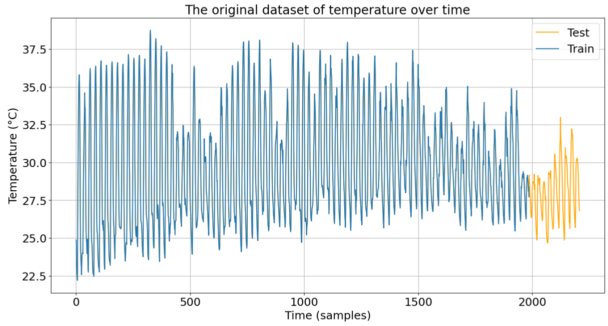

Figure 4: The original dataset divided into training set (90%) and testing set (10%).

4 Simulation Result

To obtain the dataset, temperature data was measured in Ho Chi Minh City, Vietnam. The obtained information is sampled each hour including 2209 data samples. This dataset is separated into two parts: the training data with 90% dataset, and the test data with 10% dataset as shown in Figure 4. The training data is used to fit the model with different parameters. The test data is used to evaluate the final model fit on the training data.

The accuracy metrics are evaluated through MAE (Mean Absolution Error), R^2 (R-squared), and RMSE (Root Mean Square Error) in the equations (9), (10), and (11). These criteria help overview the results of the models. The MAE measures the mean difference between predicted and actual values, while the RMSE measures the mean squared difference. The R^2 indicates the proportion of the variance in the dependent variable that can be explained by the independent variables in the model. By using all three of these criteria, the research has the metrics to choose which model is best for the problem as well as the need to adjust the

Table 1: Results of p-value index and ADF Statistic.

Test Data	ADF Test
Original Data	-3.271274 (0.016212)
1st difference operator	-14.877671 (0.000)
Log transformed data	-3.259819 (0.016773)
1st difference log operator	-14.894931 (0.000)

Table 2: Parameters of C and γ for hybrid ARIMA model.

	q=1	q=2	q=3	q=4	q=5
p=1	0.596	0.578	0.552	0.544	0.555
p=2	0.592	0.535	0.565	0.568	0.576
p=3	0.593	0.592	0.569	0.588	0.584
p=4	0.591	0.592	0.594	0.593	0.580
p=5	0.590	0.591	0.588	0.598	0.562

model and enhance its predictive ability.

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |d_t - z_t| \tag{9}$$

$$R^2 = 1 - \frac{SSR}{SST} \tag{10}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(d_t - z_t)^2} \tag{11}$$

Where N is the amount of predicted data. d_t is the actual value at time t. z_t is the expected value at time t. The SSR is short for Sum of Squared Residuals which is the sum of the squared differences between the predicted values and the actual values. The SST (Total Sum of Squares) is the sum of the squared differences between the actual values and the mean of the actual values. The stationary is an essential requirement when building models, especially ARIMA. Table 1 shows the results of the stationary analysis based on the Augmented Dickey-Fuller test (ADF). The results show that both the original data and the log-transformed data do not meet the stationary requirements. However, after performing the first difference for both the original data and the log-transformed data, both achieve stationary. The p-values are shown in parentheses. If p-values are less than 0.01 (critical value), stationary data is proved. The ADF statistic value is used to evaluate the stationary time series data. If the ADF statistic value is less than the critical value, the data series is stationary, and vice versa, if the ADF statistic value is greater than the critical Values value, the data series is Non-stationary.

4.1 Model Evaluation Comparison

4.1.1 ARIMA Model

The ARIMA model requires fine-tuning of the parameters *p, d, and q*. The MAE accuracy is shown in Table 2 for all three adjustable parameters *p, q, and d*, where parameter d has the following options $d = 0$, $d = 1$, and $d = 2$. For each d parameter, we created a table where the X-axis is the q parameter; the Y-axis is the p parameter. The higher MAE means that the model with these parameters basically does not work well. The goal of this optimizing process is to find an optimal set of parameters that achieves the best model accuracy. Table 2 shows the designed value, $d = 1$. This is selected because temperature prediction often tends to change steadily and does not tend to increase or decrease rapidly. Thus, using the ARIMA model with simple difference step *(d = 1)* will suffice to deal with this stability. Also, we can reduce the possibility of information loss in the data by applying the simple difference step [24]. As a result, with the lowest MAE =0.326, the ARIMA model parameters are achieved such as *p=1, d=1, and q=4* as shown in Table 2.

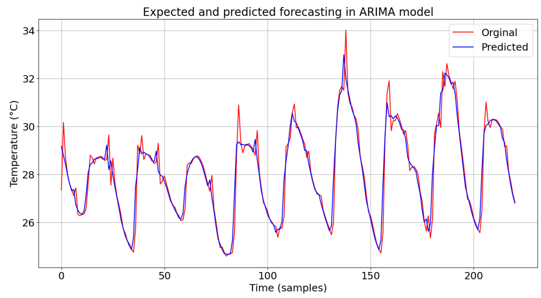

Figure 5: Comparison results between actual and predicted data (temperature) for ARIMA model.

Figure 5 illustrates the prediction results of temperature by the ARIMA model. When new data is available, the ARIMA model is re-trained [40]. As a result, the accuracy is generally quite good, only deviating at certain times.

4.1.2 Suport Vector Regression Model

As mentioned in the previous sections, the SVR model uses the RBF kernel, including C and γ (gamma) parameters that need to be calibrated. It is similar to the ARIMA model that needs to be adjusted by *p, d, and q* parameters. The selection of SVR parameters has a crucial role regarding the non-linear characteristic of the model and improving the model's accuracy. The best parameters giving the lowest MAE value are selected for the final model.

Two parameters, C and γ, are adjusted based on the dataset with the best accuracy results. The less erroneous results are used for the best model. The SVR algorithm uses RBF kernel with many different parameters such as $C = 1, 10, 100, 1000$ and $\gamma = 0.001, 0.01, 0.05, 0.1$. To find the best parameter, we calculate the accuracy between the predicted data and test data according to many different C and γ parameters by Grid Search method [15]. When applying the parameters to the experiment, the algorithm chooses the best C and γ parameters. Table 3 shows the dependencies of MAE value on the γ (gamma) and the various C values. The parameter set is selected *C=10* and γ =0.05 to achieve the highest accuracy of the SVR model which achieves the best MAE =0.46 as shown in Table 3.

Table 3: Parameters of C and γ for SVR model.

	C=1e1	C=1e2	C=1e3	C=1
γ =5e-2	0.597	0.680	0.771	1.023
γ =1e-1	0.626	0.760	0.813	1.141
γ =1e-2	0.957	0.776	0.758	1.309
γ =1e-3	2.302	2.501	2.550	2.042

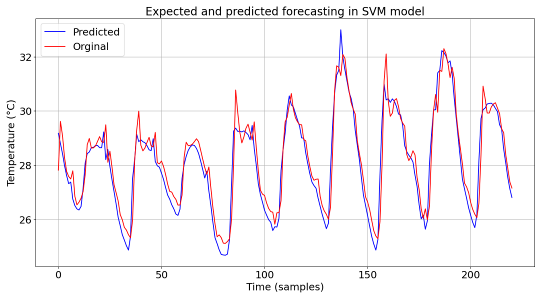

Figure 6: Comparison results between actual and predicted data (temperature) for SVR model.

The prediction results of the SVR model are depicted in Figure 6 after the fine-tuning parameter experiments. Figure 6 presents the result of the SVR model's temperature prediction compared to the actual temperature. As seen in the figure, there is a significant gap that gradually widens over time; this suggests the presence of heteroscedasticity within SVR, which degrades the model's performance.

In terms of the ARIMA's temperature prediction, as seen in Figure 5, although there is still a significant difference between the predicted and actual temperature, it can be seen that the gap between the predicted and the actual temperature gradually decreases as time progresses. This may indicate a higher accuracy of the ARIMA forecast compared to the SVR which the ARIMA model MAE = 0.326 is compared to the SVR model MAE = 0.46; however, the prediction accuracy is still unsatisfactory given the wide gap throughout most of the time.

4.1.3 Hybrid ARMIA-SVR Model

In the hybrid model, the pipeline is as follows. Time series data is used to fit the ARIMA model. Then, the residuals are used to evaluate the SVR model. The SVR model uses RBF kernel with specific parameters set up to cope with the non-linear data. Table 4 shows the investigation of parameter setup for the hybrid model. The linear part (ARIMA) uses the fixed parameters *(p, d, q)* based on the best MAE as mentioned in Table 2. Then, the non-linear part uses the residuals to calibrate C and γ parameters for the SVR model. As shown in Table 4, the C=*3* and γ=1e−7 achieve the highest accuracy for the lowest MAE = 0.32.

The temperature prediction results are obtained after the fine-tuning step of model parameters and shown in Figures 7. Figure 7 shows the comparison result of the hybrid SVR-ARIMA model between the actual temperature and predicted temperature in the test dataset. The gap becomes better when the model indicates higher accuracy compared to a single ARIMA only or single SVR only.

Table 4: Parameters of C and γ for hybrid ARIMA-SVM model.

	C=1	C=3	C=1e2	C=1e3	C=1e5
γ=1e-1	1.213	1.557	2.620	2.943	2.981
γ=1e-4	0.982	0.982	1.393	1.844	2.743
γ=1e-6	0.614	0.670	0.711	1.244	1.733
γ=1e-7	0.656	0.545	0.882	0.791	1.609
γ=1e-8	0.473	0.625	0.599	1.000	0.634

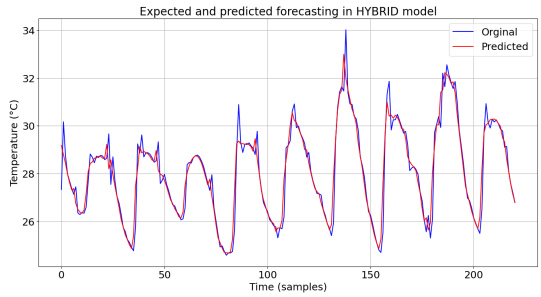

Figure 7: Comparison results between actual and predicted data (temperature) for hybrid SVM-ARIMA model.

4.1.4 LSTM Model

The LSTM model is an improvement of the regression network that allows learning of long correlations that are constrained by the traditional regression network. Like ARIMA and SVR, the LSTM model also requires hyperparameters to fit the model. The architecture of the LSTM model consists of the two LSTM layers which have 128 and 64 neurons, and two dense layers which have 25 neurons and 1 neuron, respectively.

The number of consecutive data points used as input to predict the next data point or a future sequence of data points is 5 (window size =5). The LSTM model chooses a loss function suitable for prediction problems which is Mean Squared Error (MSE) for regression problems. The optimization algorithm adjusts the weight of the model including Adam. The Adam algorithm optimizes the weights and biases to find the values of those parameters so that the Loss function reaches the minimum value. Tanh activation is used with 50 epochs. Figure 8 shows the comparison results between actual and predicted temperature for the LSTM in the test dataset by the above-mentioned LSTM architecture. Three accuracy metrics MAE, R-squared, and RMSE are used to evaluate mentioned in more detail in Table 6.

4.1.5 Random Forest Model

Random forest (RF) is a widely used model in supervised learning, especially in value prediction problems. The RF model does not require defining special parameters. Instead, it uses a set of decision trees to make the final prediction. The RF model-building process consists of randomly selecting samples and features from the data set to generate each decision tree. Each decision tree is trained based on part of the data and part

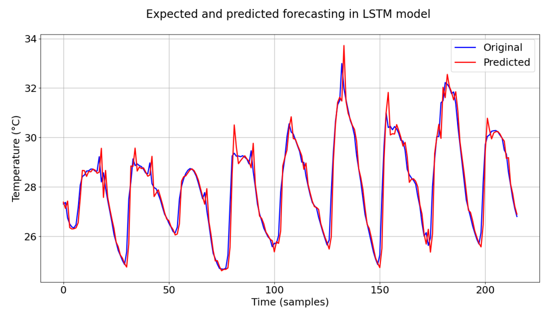

Figure 8: Comparison results between actual and predicted data (temperature) for LSTM model.

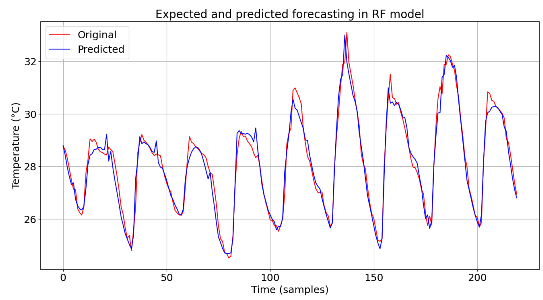

Figure 9: Comparison results between actual and predicted data (temperature) for random forest model.

of randomly selected features. Then the final prediction will be the sum of the predictions of each decision tree. The number of trees in the forest is 500. The bootstrap samples are used when building trees. The random state is 42 which controls both the randomness of the bootstrapping of the samples used when building trees and sampling of the features to consider when looking for the best split at each node. The minimum number of samples at the leaf node is required to be one. The maximum depth of the tree is 10. The minimum number of samples required to split an internal node is two as shown in Table 5.

Figure 9 shows the comparison results between actual and predicted temperature for the random forest model in the test dataset. Random forest algorithms operate based on assemble learning. The RF forms several decision trees on different subsets of a given dataset. The larger the number of trees, the higher the accuracy of the model. The random forest algorithm creates various decision trees. Each tree is constructed by a decision tree algorithm on different data sets and using different subsets of features. The prediction results are aggregated from the decision trees. Each decision tree has random elements, taking random data and random attributes to build a decision tree. As a result, the information from the trees complements each other, leading to the model with good prediction results. Three accuracy metrics MAE, R-squared, and RMSE are used to evaluate mentioned in more detail in Table 6.

Table 5: Random forest parameters.

Parameter	Description
n_estimators = 500	The number of decision trees.
max_depth = 10	The maximum depth of the decision tree, this parameter determines the number of classes of the decision tree and affects the model's ability to learn complex rules.
min_samples_split = 2	Minimum number of samples required to split a node in the tree.
min_samples_leaf = 1	Minimum number of samples required to form a leaf when constructing a decision tree.
random_state = 42	Used to set the initial value of the random number generator, this random number generator is used to generate decision trees in RF.

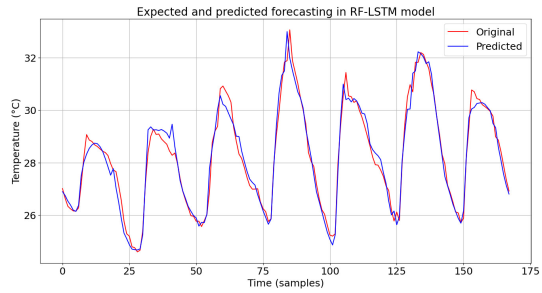

Figure 10: Comparison results between actual and predicted data (temperature) for hybrid LSTM and random forest model.

Table 6: Accuracy comparison among various models.

	ARIMA	SVR	Hybrid ARIMA-SVR	LSTM	RF	Hybrid RF-LSTM
MAE	0.326	0.430	0.320	0.313	0.279	0.269
RMSE	0.544	0.565	0.545	0.513	0.380	0.358
R-squared	0.910	0.904	0.910	0.922	0.956	0.961

4.1.6 Hybrid Random Forest and LSTM Model

After training the model with the random forest network, it continues to go through the LSTM network as shown in Figure 2. Because the feature has been extracted through the random forest network, the LSTM network only needs to train a few epochs, the results are quite good. Here, the LSTM network is formed from two LSTM layers with several neurons of 128 and 64 and two dense layers with 25 neurons and 1 neuron which is similar to a single LSTM model as mentioned above. The number of trees in the forest is 500 which is similar to a single random forest as mentioned above.

Figure 10 shows the hybrid RF-LSTM's temperature predictions. The RF-LSTM model achieves the best prediction performance out of the five investigated models as its predicted temperature not only indicates a considerably smaller difference than the actual temperature but also can capture the actual temperature trend.

Based on the summary table 6, it can be commented that the RF-LSTM model shows the best results among the evaluated models. The RF-LSTM model has the best MAE, RMSE, and R-squared metrics, showing more accurate and stable predictions than the RF model. The RF model also shows good results, although it is not superior to the hybrid RF-LSTM. The LSTM and hybrid SVR-ARIMA models also performed relatively well, however, it not able to beat RF and RF-LSTM. Although the ARIMA and SVR models have not yet performed as well as other models, they can still be useful in special situations or when the data has obvious non-linear characteristics.

In more detail, Table 6 shows the cumulative results

Table 7: Elapsed time comparison on PC and Raspberry PI by performing the training and testing step.

	ARIMA	SVR	Hybrid ARIMA-SVR	LSTM	RF	Hybrid RF-LSTM
Elapsed Time on PC (s)	128	183	136	88.2	32.1	42.7
Elapsed Time on PI (s)	143	336	152	95.2	45.14	78.8

Table 8: Peak memory usage comparison by performing the training and testing step.

	ARIMA	SVR	Hybrid ARIMA-SVR	LSTM	RF	Hybrid RF-LSTM
Memory Usage (MB)	280	114	233	469	188	533

of the six models. The results are as expected, given the conditions, the appropriate setup, and the progress of the experiment so far. As shown in Table 6, the hybrid RF-LSTM model is the most accurate in terms of MAE, R-squared, and RMSE prediction quality metrics. Table 6 shows the SVR has the lowest accuracy. Both hybrid SVR-ARIMA, ARIMA, random forest, and LSTM share similar values of MAE, Rsquared, and RMSE; both SVR, ARIMA, RF, and LSTM have their respective values ranging from 0.279 to 0.326 for MAE, RMSE is from 0.38 to 0.544, and Rsquared is around 0.920, from 0.904 to 0.956. These values, combined with the temperature prediction results shown in Figures 5, 6, 7, 8, and 9 further confirm the unsatisfactory performance of both SVR, ARIMA, hybrid ARIMA-SVR, LSTM, and RF models, if they are used separately in the selected prediction task. For the hybrid RF-LSTM model, the values of MAE, R-squared, and RMSE are much smaller than the others as their values are 0.269, 0.358 for MAE, RMSE, and 0.961 for R-squared, respectively. This validation result, combined with the prediction performance in Figure 10, suggests the superior prediction performance of the hybrid RF-LSTM model in short-term local weather forecasting.

In addition to accuracy, other aspects such as processing speed and memory usage should be considered. The model deployment is measured on a PC with configuration Intel(R) Xeon(R) CPU at 2.20GHz, 12 Gb RAM; and on a Raspberry PI board with configuration Broadcom BCM2711, SoC 64-bit quad-core ARM Cortex-A72, 4 GB RAM. Table 7 shows the elapsed time comparison on PC and Raspberry PI by performing the training and testing step comparing the execution speed of the models on PC and Raspberry Pi. The elapsed time is an important metric for evaluating the efficiency and performance of various processes and systems. The RF shows the fastest elapsed time which is 32.1s on PC and 45.14s on PI. This is because the RF can be easily parallelized since each decision tree in the forest can be trained independently. This allows for efficient utilization of multiple CPU cores and accelerates the training process. Each decision tree can independently predict the target variable, and the final prediction can be aggregated quickly. The RF is faster to train and inference compared to the LSTM. Thus, the hybrid RF-LSTM can adjust the relative contribution of RF and LSTM based on their computational requirements while still benefiting from the sequential pattern capture of the LSTM and faster execution speed time of RF. The elapsed time of RF-LSTM is 42.7s on PC and 78.8s on PI.

Table 8 shows peak memory usage comparison by performing the training and testing step. Peak memory usage refers to the maximum amount of computer memory (RAM) that a process consumes during its execution. Memory usage is a critical metric for assessing the efficiency and resource requirements of the model. Monitoring peak memory usage helps ensure that models are using memory resources efficiently and that they do not exceed available memory limits, which could lead to performance issues or even crashes. It is realized that the hybrid RF-LSTM utilizes the highest available system resources that are as large as 533 MB. This is the RF-LSTM that performs two combination models, however, it is still executable on Raspberry hardware as an edge device.

The RF-LSTM model is capable of handling nonlinear and heterogeneous features in the data. It is also capable of capturing data patterns over time and processing long-term state information. Therefore, this model achieves the highest accuracy compared to other models. However, the choice of the best model depends on the specific requirements and criteria of the problem. If execution speed is the top factor, RF is the more suitable choice.

In summary, based on the comparison results of the criteria MAE, RMSE, R2, execution speed, and peak memory usage, it can be found that the RF-LSTM model is the best choice. This model has the best performance in predicting, with high accuracy and faster execution speed, Although RF-LSTM uses more peak memory due to the combination of two different models, it also brings benefits it's worth making up for that much memory usage. The RF-LSTM model is the best choice based on the comparison results of the evaluation criteria and is suitable for the temperature data set needed for highly accurate results.

5 Conclusion

The short-term local weather forecast system has various applications in transportation, agriculture, power management, and daily life. The advantage of such a system is that it facilitates decision-making more effectively in a short period (hour) at the local site.

This paper explores using and hybridizing simple prediction models to maximize accuracy while maintaining low computational effort and the need to process and acquire large volumes of data. Furthermore, this paper also proposes the design of a weather station based on a real-time IoT system. The weather device is set up at the local Ho Chi Minh City site.

The obtained data prove that the hybrid model significantly improves prediction performance over individual models. The hybrid RF-LSTM model is constructed by combining the advantages and optimal parameters of the random forest model and the LSTM model to forecast weather for each hour. From the comparison result, both SVR, ARIMA, hybrid

ARIMA-SVR, LSTM, and RF show unsatisfactory prediction performance; their significant divergence of predicted temperature and actual temperature, combined with rather high values of accuracy indicators, suggests that by using these models individually, the prediction accuracy is severely limited. This research overcomes this fact by combining both random forest and LSTM models. The empirical results show that the hourly predicted temperature optimized by the hybrid model shows the best accuracy in metrics of MAE, R-squared, and RMSE, which suggests a significant reduction in forecasting errors. Leveraging the random forest which is faster in training and inference, the RF-LSTM makes predictions on new data faster and more efficient. The hybrid approach can offer advantages in terms of elapsed time and computational efficiency although it has the highest peak memory usage. Additionally, while the difference between the proposed hybrid model's predicted air temperature and the actual air temperature is small, the hybrid model can also capture the actual temperature trend in its prediction performance, which makes it even more relevant for many other possible decision-making steps in the applications.

Acknowledgement: Supported by Internal Grant Agency of Tomas Bata University under project no. IGA/CebiaTech/2023/004, and by the resources of A.I.Lab at the Faculty of Applied Informatics, Tomas Bata University in Zlin, Czech Republic. Further supported by the European Union under the REFRESH — Research Excellence For Region Sustainability and High-tech Industries project number CZ.10.03.01/00/22-003/0000048 via the Operational Programme Just Transition. The following grants are also acknowledged for the financial support provided for this research: grant of SGS No. SP2023/050, VŠB-Technical University of Ostrava, Czech Republic.

References

[1] Adhikari, R., and Agrawal, R. K. An introductory study on time series modeling and forecasting. *arXiv preprint arXiv:1302.6613* (2013).

[2] Amami, R., et al. A robust voice pathology detection system based on the combined bilstm–cnn architecture. *MENDEL 29*, 2 (2023), 202–210.

[3] Baboo, S. S., and Shereef, I. K. An efficient weather forecasting system using artificial neural network. *International journal of environmental science and development 1*, 4 (2010), 321.

[4] Babu, C. N., and Reddy, B. E. A moving-average filter based hybrid arima–ann model for forecasting time series data. *Applied Soft Computing 23* (2014), 27–38.

[5] Biswas, M., Dhoom, T., and Barua, S. Weather forecast prediction: an integrated approach for analyzing and measuring weather data. *International Journal of Computer Applications 182*, 34 (2018), 20–24.

[6] Cadenas, E., and Rivera, W. Wind speed forecasting in the south coast of oaxaca, mexico. *Renewable energy 32*, 12 (2007), 2116–2128.

[7] Chang, W. C., and Sangodiah, A. Automated semantic annotation deploying machine learning approaches: A systematic review. *MENDEL 29*, 2 (2023), 111–130.

[8] Chen, L., and Lai, X. Comparison between arima and ann models used in short-term wind speed forecasting. In *Asia-Pacific power and energy engineering conference* (2011), pp. 1–4.

[9] Chih, H.-C., et al. Implementation of edge computing platform in feeder terminal unit for smart applications in distribution networks with distributed renewable energies. *Sustainability 14*, 20 (2022), 13042.

[10] Codeluppi, G., Davoli, L., and Ferrari, G. Forecasting air temperature on edge devices with embedded ai. *Sensors 21*, 12 (2021), 3973.

[11] Corne, D., et al. Accurate localized short term weather prediction for renewables planning. In *2014 IEEE symposium on computational intelligence applications in smart grid (CIASG)* (2014), IEEE, pp. 1–8.

[12] Cuzzocrea, A., Gaber, M. M., Fadda, E., and Grasso, G. M. An innovative framework for supporting big atmospheric data analytics via clustering-based spatio-temporal analysis. *Journal of Ambient Intelligence and Humanized Computing 10* (2019), 3383–3398.

[13] Dhoot, R., Agrawal, S., and Kumar, M. S. Implementation and analysis of arima model and kalman filter for weather forecasting in spark computing environment. In *Proceedings of the 3rd ICCCT* (2019), IEEE, pp. 105–112.

[14] Fieri, B., and Suhartono, D. Offensive language detection using soft voting ensemble model. *MENDEL 29*, 1 (2023), 1–6.

[15] Fuadah, Y. N., Pramudito, M. A., and Lim, K. M. An optimal approach for heart sound classification using grid search in hyperparameter optimization of machine learning. *Bioengineering 10*, 1 (2022), 45.

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

[17] Hou, J., Wang, Y., Hou, B., Zhou, J., and Tian, Q. Spatial simulation and prediction of air temperature based on cnn-lstm. *Applied Artificial Intelligence 37*, 1 (2023), 2166235.

[18] Kanagaraj, E., Kamarudin, L., Zakaria, A., Gunasagaran, R., and Shakaff, A. Cloud-based remote environmental monitoring system with distributed wsn weather stations. In *2015 IEEE SENSORS* (2015), IEEE, pp. 1–4.

[19] Krishnaveni, N., and Padma, A. Weather forecast prediction and analysis using sprint algorithm. *Journal of Ambient Intelligence and humanized computing 12* (2021), 4901–4909.

[20] Kung, H.-Y., Kuo, T.-H., Chen, C.-H., and Tsai, P.-Y. Accuracy analysis mechanism for agriculture data using the ensemble neural network method. *Sustainability 8*, 8 (2016), 735.

[21] Liaw, A., Wiener, M., et al. Classification and regression by randomforest. *R news 2*, 3 (2002), 18–22.

[22] Mahmood, M. R., et al. A novel approach for weather prediction using forecasting analysis and data mining techniques. In *Proceedings of the 7th ICIECE* (2019), Springer, pp. 479–489.

[23] Manandhar, S., Lee, Y. H., and Meng, Y. S. Gps-pwv based improved long-term rainfall prediction algorithm for tropical regions. *Remote Sensing 11*, 22 (2019), 2643.

[24] Mei, W., Xu, P., Liu, R., and Liu, J. Stock price prediction based on arima-svm model. In *International Conference on Big Data and Artificial Intelligence* (2018), p. 4.

[25] Mei, W., Xu, P., Liu, R., and Liu, J. Stock price prediction based on arima-svm model. In *International Conference on Big Data and Artificial Intelligence* (2018), p. 4.

[26] Mendoza Uribe, I. Predictive model of the enso phenomenon based on regression trees. *MENDEL 29*, 1 (2023), 7–14.

[27] Mohd-Safar, N. Z., Ndzi, D., Kagalidis, I., Yang, Y., and Zakaria, A. Short-term localized weather forecasting by using different artificial neural network algorithm in tropical climate. In *Proceedings of SAI Intelligent Systems Conference* (2018), Springer, pp. 463–476.

[28] Nashwan, M. S., Shahid, S., and Wang, X. Assessment of satellite-based precipitation measurement products over the hot desert climate of egypt. *Remote Sensing 11*, 5 (2019), 555.

[29] Nawi, W., et al. Improved of forecasting sea surface temperature based on hybrid arima and support vector machines models. *Malaysian Journal of Fundamental and Applied Sciences 17*, 5 (2021), 609–620.

[30] Nurunnahar, S., et al. A short term wind speed forecasting using svr and bp-ann: A comparative analysis. In *20th International Conference of Computer and Information Technology (ICCIT)* (2017), IEEE, pp. 1–6.

[31] Ordóñez, C., et al. A hybrid arima–svm model for the study of the remaining useful life of aircraft engines. *Journal of Computational and Applied Mathematics 346* (2019), 184–191.

[32] Pai, P.-F., and Lin, C.-S. A hybrid arima and support vector machines model in stock price forecasting. *Omega 33*, 6 (2005), 497–505.

[33] Prachyachuwong, K., and Vateekul, P. Stock trend prediction using deep learning approach on technical indicator and industrial specific information. *Information 12*, 6 (2021), 250.

[34] Rasel, R. I., Sultana, N., and Meesad, P. An application of data mining and machine learning for weather forecasting. In *Recent Advances in Information and Communication Technology 2017: Proceedings of the 13th International Conference on Computing and Information Technology (IC2IT)* (2018), Springer, pp. 169–178.

[35] Reyniers, M. *Quantitative precipitation forecasts based on radar observations: Principles, algorithms and operational systems.* Institut Royal Météorologique de Belgique Brussel, Belgium, 2008.

[36] Rezapour, S., et al. Forecasting rainfed agricultural production in arid and semi-arid lands using learning machine methods: A case study. *Sustainability 13*, 9 (2021), 4607.

[37] Shahi, T. B., et al. Stock price forecasting with deep learning: A comparative study. *Mathematics 8*, 9 (2020), 1441.

[38] Suthar, M., and Aggarwal, P. Modeling CBR value using RF and M5P techniques. *MENDEL 25*, 1 (2019), 73–78.

[39] Tektaş, M. Weather forecasting using anfis and arima models. *Environmental Research, Engineering and Management 51*, 1 (2010), 5–10.

[40] Toai, T. K., et al. Arima for short-term and lstm for long-term in daily bitcoin price prediction. In *International Conference on Artificial Intelligence and Soft Computing* (2022), Springer, pp. 131–143.

[41] Wang, P., Zhang, H., Qin, Z., and Zhang, G. A novel hybrid-garch model based on arima and svm for pm2. 5 concentrations forecasting. *Atmospheric Pollution Research 8*, 5 (2017), 850–860.

[42] Wang, Z., and Mujib, A. M. The weather forecast using data mining research based on cloud computing. In *Journal of Physics: Conference Series* (2017), vol. 910, IOP Publishing, p. 012020.

[43] Wardana, I. N. K., Gardner, J. W., and Fahmy, S. A. Optimising deep learning at the edge for accurate hourly air quality prediction. *Sensors 21*, 4 (2021), 1064.

[44] Wilson, B., et al. High-resolution estimation of monthly air temperature from joint modeling of in situ measurements and gridded temperature data. *Climate 10*, 3 (2022), 47.

[45] Wilson, J. W., and Brandes, E. A. Radar measurement of rainfall—a summary. *Bulletin of the American Meteorological Society 60*, 9 (1979), 1048–1060.

[46] Yonekura, K., Hattori, H., and Suzuki, T. Short-term local weather forecast using dense weather station by deep neural network. In *2018 IEEE international conference on big data (big data)* (2018), IEEE, pp. 1683–1690.

[47] Zhang, K., Huo, X., and Shao, K. Temperature time series prediction model based on time series decomposition and bi-lstm network. *Mathematics 11*, 9 (2023), 2060.

[48] Zhang, Y., and Hanby, V. I. Short-term prediction of weather parameters using online weather forecasts. *Proceedings: Building Simulation 2007* (2007).