Artificial intelligence-based weather prediction framework using neural networks

Keshav Kaushik[1], Gunjan Chhabra[2,3], Salil Bharany[4], Ateeq Ur Rehman[5], Habib Hamam[6,7,8,9]
[1]Amity School of Engineering and Technology, Amity University Punjab, Mohali, India
[2]Department of CSE, Graphic Era Hill University, Dehradun, India
[3]Graphic Era Deemed to be University, Dehradun, India
[4]Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab, India
[5]School of Computing, Gachon University, Seongnam, Republic of Korea
[6]Faculty of Engineering, Uni de Moncton, Moncton, Canada
[7]School of Electrical Engineering, University of Johannesburg, South Africa
[8]Hodmas University College, Taleh Area, Mogadishu, Banadir, Somalia
[9]Bridges for Academic Excellence, Spectrum, Tunis, Tunisia

Article Info

Article history:

Received Mar 23, 2024
Revised Aug 7, 2024
Accepted Aug 25, 2024

Keywords:

Artificial intelligence
Internet of things
Neural networks
Smart computing devices
Weather prediction

ABSTRACT

For humans, weather prediction is vital in making rational everyday choices and avoiding risk. Accurate weather forecasting is regarded as one of the world's most difficult issues. New weather forecasting, unlike conventional techniques, relies on a mixture of computer simulations, observation (via balloons and satellites), and information of patterns and trends (via local weather analysts and weather stations). Predictions are rendered with fair precision using such techniques. Prediction algorithms based on complicated formulas run the majority of computational models used for prediction. This paper highlights the prediction of weather with the artificial neural networks (ANN) using the latest available smart computing devices. To assess the effectiveness of the model, comparison research is conducted with the other existing models in the same area. The result demonstrates that our approach is better in comparison to other similar research and products. The comparative analysis has been undergone which confirms the superiority of our proposed techniques with an accuracy of 90.4%.

Corresponding Author:

Ateeq Ur Rehman
School of Computing, Gachon University
Seongnam-13120, Republic of Korea
Email: 202411144@gachon.ac.kr

1. INTRODUCTION

The integration of internet of things (IoT) devices in weather prediction models highlights the necessity for robust security measures to safeguard against potential threats [1]. A smart computing device is a context-aware electronic device consisting of autonomous computation and dataset sharing through wired or wireless connections with other devices. Context perception, autonomous computation, and networking are the three key features of a smart computer. This description adheres to the IoT core concept, which is that each and every 'thing' may be a part of it. If sensors are introduced, a small amount of processing power, and network access to a chair, it could become a smart chair. Radar and light detection and ranging (LIDAR) instruments, as well as microphones, GPS receivers, and cameras, are both possible forms of information for context-aware computation. A context-aware machine may collect data from these and other channels and react using pre-determined guidelines or computational intelligence. The core feature of autonomous computation is that it involves a computer or a group of computers executing activities independently of the user. For starters, our phones can make recommendations dependent on our position or the weather. For this simple mission, a smartphone must be self-contained and make decisions based on background data. Smart systems, including smart houses and houses, can be used to add information to both the physical environment and therefore can help simplify systems and practices. They can help increase productivity and maximize activities in almost every sector, from industrial automation to healthcare. Smart devices can run at the channel's surface or on very limited nodes, and because they are small, they are efficient enough to handle information without having to transfer it to the server. Sensors, refrigerators, wearables, and container shipping are among the devices that can operate autonomous workforces. There is a certain drawback associated with these smart computing devices like business categories that are rapidly maturing, Factionalism competition between devices is being created because of digital convergence. In certain places, the demand is flooded. Supportive relationship times on non-subsidized devices like tablets and PCs, while there is no ground breaking technology here, just enticing product enhancements. Survey results by US competitors LLC were mixed [2], and a drop in PCs and associated sockets, I/O interfaces, and separable, discrete connection items will be required to counterbalance a decline in economically healthy, expanding markets. Some firms may have been able to, should have been able to, or will be required to extend their product lines to include micro-packaging-related technologies required for future interconnect applications. Some have downplayed the importance of these high-volume, low-cost markets, sometimes to their detriment. Hostile vehicle mitigation (HVM) provides the cornerstone for low-cost manufacturing, allowing various spinoffs to higher-valued, more specialized applications. The worldwide estimation of smart computing devices [3] is shown in Figure 1.

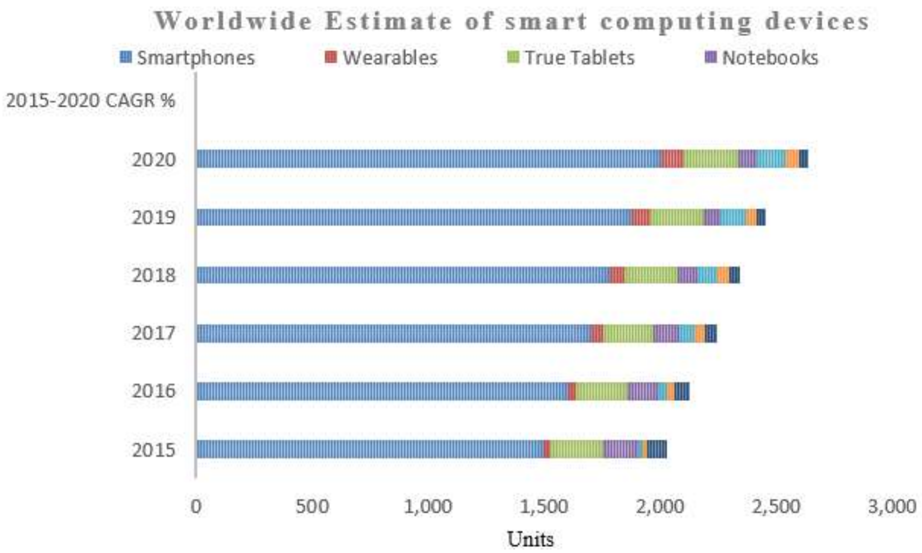

Figure 1. Worldwide estimate of smart computing devices (in millions)

The standard approach to weather prediction is synoptic weather forecasting, which is the first process. Until the late 1950s, this was the dominant mode of communication. The term 'synoptic' refers to the detection of various weather components at a given time. To a weather forecaster, a weather forecast that represents weather patterns at a certain time is a synoptic chart. Every day, a meteorological centre creates a sequence of synoptic graphs to get an overall picture of the evolving weather trend. Weather predictions are built on the foundation of such synoptic maps. To make a prediction, numerical weather prediction utilizes the computational power of computers. Weather predictions models are sophisticated computer systems that operate on supercomputers that make forecasts for a variety of environmental factors such as temperature, pressure, wind, and rainfall. A meteorologist looks at how well the computer's future features can work to create the climate for the day. Statistical methods rely on weather data from the past, predicting that the future will be close to the past. The key aim of researching historical weather data is to decide which elements of the weather are strong predictors of future activities. Artificial neural network (ANN) was created in 1986 and has attracted a lot of interest from researchers because it can solve complex non-linearity challenges better than traditional approaches. Over the past five years, ANN has developed dramatically. Since the climate dataset is non-linear, weather prediction can be done using an ANN. Many meteorological technologies, including weather predictions, have used machine learning. They have been discovered to be more effective than any conventional expert scheme. Some of the world's most advanced computers make today's weather forecasts. Each second, the gigantic machines do millions of calculations to solve equations that predict temperature, winds, precipitation, and other environmental events. Even for the most advanced computers are put to the test when speed and accuracy are required in a forecast. It is possible that the world will take a drastically different direction. Artificial intelligence (AI) models offer potential to provide faster, more accurate weather forecasts, as shown in a collaboration between the University of Washington and Microsoft Research. UW News, 2020 [4]. In order to integrate successful AI methods with weather forecasting, the researchers placed the six sides of a cube onto the globe Earth and then flattened out the cube's six faces, like in an architecture paper model. Because of their particular significance in the weather, the authors handled the polar faces differently as a means to enhance prediction accuracy. The authors then put their model to the test by estimating the worldwide height of the 500 hectopascal pressure every 12 hours for a year. WeatherBench is a benchmark test for weather forecasts based on data. One of the top performers in that forecasting test, which was made for three-day estimates, was this new model. The authors note that more information would be needed for the data-driven model before it could start competing with current operational forecasts, but the idea has promise as a fresh approach to producing weather forecasts, particularly given the growing amount of historical forecasts and weather findings. Understanding the economic impact of severe weather events emphasizes the need for accurate weather prediction to mitigate financial risks [5]. The success of machine learning in developing a COVID-19 warning system [6] showcases the potential of similar approaches in weather prediction. Advanced prediction such as multiclass prediction has been also used [7], [8].

Key aspects and novel elements: this study introduces a unique approach to weather prediction. It employs ANN to analyze and forecast weather patterns. The innovative aspect lies in the utilization of smart computing devices for data collection and analysis. These devices, equipped with various sensors, gather real-time environmental data. This data forms the foundation for the neural network's training and prediction processes. The framework differs from traditional weather prediction methods by it is reliance on direct data input from IoT-enabled devices, enhancing prediction accuracy. Another novel element is the integration of machine learning algorithms with traditional weather forecasting techniques. This combination offers a more dynamic analysis of weather patterns. The study also explores the impact of climate change on prediction models. It acknowledges the increasing unpredictability of weather patterns and adapts it is methodology accordingly. The research aims to improve prediction accuracy and reliability, focusing on next-day weather forecasts. It highlights the significance of advanced computing and artificial intelligence in addressing complex environmental challenges. This approach represents a forward-thinking contribution to the field of weather prediction, promising enhanced accuracy and efficiency in forecasts.

2. RELATED LITERATURE

2.1. Current research

The authors examined 312 papers and journal publications from IEEE, ACM, and Elsevier as well as other referenced journals that have been published during 2018 to date. The publications were categorised based on research evaluations, keywords, and findings to match the pertinent work and findings with this research. The authors narrowed down the relevant literature using the selection methods indicated below to arrive at the final 31 studies shown in Figure 2. The final 31 studies that were chosen were grouped as shown in Table 1 for works that closely matched the metadata, synopsis, and keywords like weather prediction, prediction models, artificial intelligence, smart computing devices, machine learning, and IoT. An overall distribution ratio of between 16 and 25% was indicated by the categorization.

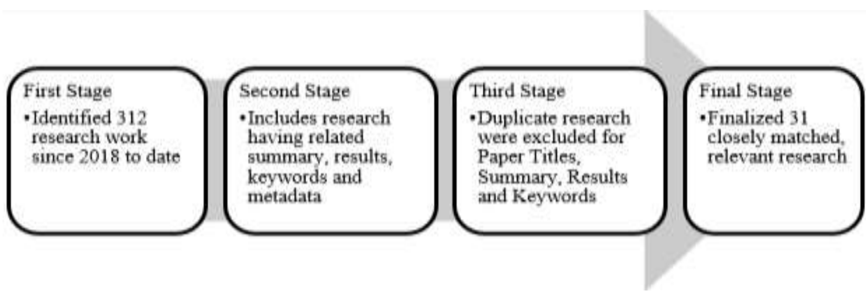

Figure 2. Research selection methodology

Table 1. Research papers and subcategories

Classification	First stage	Second stage	Third stage	Final stage	Breakup (%)
Weather prediction	65	42	16	6	20.83
Artificial intelligence	77	49	19	8	24.68
Machine learning	64	41	16	6	20.51
Smart devices	51	33	13	5	16.35
Internet of things	55	35	14	5	17.63
	312	200	78	31	

Several key sectors, such as agriculture, rely on weather forecasting for production. These forecasts have an impact on a country's economy and people's lives, even as they have become less effective and more chaotic as the climate has changed at a rapid pace. Singh *et al.* [9] created and presented a low-cost, portable weather prediction system to tackle these challenges. The proposed approach provided a more accurate and dependable alternative for forecasting weather in remote places. To anticipate meteorological conditions, the authors used machine learning and data analytics algorithms such as the random forest (RF) classification. Vasantha *et al.* [10] used machine learning and data analytics to provide an imitation model for predicting varied climate conditions across the Indian subcontinent. The standard wellspring of data for directed taking in will be compiled from Indian government datasets and the machine learning data bank.

Heatstroke is disastrous for a few countries, in particular because of climate change. Residents of different places are not advised of the dangers of coming to their areas because they are told the city's average temperature, even though the temperature fluctuates at different heights and over short distances. Durrani *et al.* [11], proposed a smart weather station to monitor and anticipate meteorological data as well as send out rapid alerts to residents in various places. Climate change uncertainty makes decisions about food production and cost, as well as procurement costs and quantities, increasingly difficult. For the distribution in Canada [12], provided artificial intelligence-based forecasting models that were trained, tested, and compared for predicting future daily offer prices using mutual transactions.

Jadhav *et al.* [13], suggested a simple, easy-to-use methodology for accurately predicting agricultural prices in the face of changing meteorological conditions. Using data such as rainfall, month, wholesale price index, and year, machine learning decision tree regression techniques were used to predict crop price. The model supplied farmers with a high-level forecast that included weather forecasting, crop and fertilizer recommendations, and cooperation and chat portals for shopping and guiding farmers.

Hatim *et al.* [14], reviewed the literature on some of the strategies employed by various scholars to anticipate precipitation using AI. In recent decades, weather forecasting has grown in importance as a scientific topic. In most cases, the researchers sought to create a linear relationship between the meteorological conditions provided and the requested data. Nevertheless, after the advent of non-linear weather conditions, the emphasis has turned to non-linear meteorological data prediction. To estimate ice-growth-induced productivity losses, Scher and Molinder [15], proposed a statistical technique based on random forest regression. It anticipates relative producing power loss ahead of time using regional weather predictions and on-site data to improve projections for the next day's power generation.

Pavuluri *et al.* [16], anticipated changes in temperature in specific places. The AI-based algorithms calculated the mean values, median values, confidence values, and probability, as well as highlighting the differences between plots of all three algorithms, among other things. Finally, the authors projected whether the temperature would rise or fall and if it would rain or not, using these algorithms. The dataset is entirely dependent on the weather of a certain location and includes a few objects such as the year, month, and temperature, as well as forecasted values. The authors of [17]-[19] offered techniques for farmers to capture the yield of their crops before cultivation and so assist them in making the appropriate decisions in the field of agriculture. The machine learning algorithm can then be distributed after implementation with web-based visual software that is simple to use. The farmer is given access to the results obtained.

The urban heat island effect is well-known for making cities warmer than their rural environs. Predicting geographical distribution is a step toward implementing the Smart City idea of providing an early warning system for vulnerable groups. Vulova *et al.* [20], the spatial dispersion of urban conditions has been constrained thus far by the poor spatial resolution of traditional data sources. Volunteered geographic information gives alternative data with greater spatial density, with citizen weather stations continually monitoring this in hundreds or thousands of sites throughout a single city. The author suggested a machine-learning-based technique for forecasting power consumption using meteorological data. The proposed technique can anticipate energy usage with a correlation coefficient of 75.7% using a dataset of observations from domestic appliances spanning 350 days in combination with meteorological conditions.

Brahim [21], projected long short-term memory for index sequence and grid data that contained temporal and spatial interdependencies. The anticipated index may be determined by projecting output onto the anomalous pattern. The authors pre-processed data using discrete wavelet transform and proposed two models for index data and grid data, respectively, to improve prediction accuracy, particularly for severe occurrences. Sharma *et al.* [22] demonstrates how a Smart Gateway may be used to collect sensor data from networks installed in a greenhouse, which can then be reinforced with a neural network prediction model to estimate the greenhouse's interior air temperature. With an R2 score of 0.965 and a root mean square error (RMSE) of 1.50 °C, a mean absolute percentage error of 4.91%, and a mean absolute percentage error of 4.91%, the neural network prediction system can estimate greenhouse air temperature.

Urban flash floods [23] are becoming more common because of rapid urbanization, climate change, and heavy rainfall. It's critical to predict when a flood will come so that the damage it causes can be minimized. An urban flash flood occurs in a city in a very short period, as the name implies. Rani *et al.* [24] developed a theoretical model that considers the parameters that cause urban flash floods and predicts the event in advance. Data syntheses are used to verify the model's soundness, and the findings are validated using an ANN.

2.2. Research gap

The exploration of smart computing devices in weather prediction has shown significant promise. However, there remains a clear gap in the integration of these technologies with advanced security measures and machine learning techniques tailored for meteorological applications. Previous works have successfully applied machine learning in various domains, yet their potential in enhancing weather forecasting through secure and efficient data collection and analysis has not been fully realized. This gap is evident in the lack of comprehensive frameworks that leverage IoT for real-time data gathering while ensuring the robustness and reliability of the predictive models against cyber threats. Additionally, the potential for machine learning models to improve accuracy in weather forecasts, considering the vast and varied data generated by IoT devices, has not been fully explored. Our contribution aims to bridge this gap by developing a novel approach that combines secure, smart computing infrastructures with advanced machine learning algorithms for weather prediction. This initiative not only addresses the current limitations but also sets a new direction for research in the intersection of cybersecurity, IoT, and meteorology.

3. RESEARCH METHOD

The climate is defined by physicists as a complicated system. While there are several interpretations, this instance can be deemed complicated enough to be unsolvable analytically. This may appear disappointing, but it prepares the way for a diverse set of numerical methods aimed at addressing climate change. AI and machine learning algorithms are undoubtedly a part of the recent computational advances. The problem at hand involves utilizing standard machine-learning methods to estimate the average temperature. The following is a step-by-step approach to implementing the suggested model with Python on a real-world dataset.

The researchers used the Australian government's WeatherPerth.csv dataset. This archive contains nearly nine years of daily weather data from a Perth weather station. The last column indicates if rain fell the next day. This is a value we'd like to forecast. As a result, we're dealing with a supervised learning problem in machine learning terms. These are our training examples, and they are divided into two categories: whether or not it rained the next day. This study provides a data model that can forecast whether it will rain tomorrow based on today's weather circumstances, even if those variables are not identical to those experienced previously. There are numerous approaches to construct such a model, and the ANN model is used in this study as illustrated in Figure 3 for the proposed research methodology model.

3.1. Step 1: loading the dataset

In step 1, the authors utilized python programming to complete the implementation phase, which included loading the necessary libraries and dataset. The libraries utilized were some of the most well-known for data processing, plotting, and mathematical calculations (Pandas, NumPy and Matplotlib). Then there are others for advanced data visualization (such as folium) and some that are ARIMA model specialized libraries (like stats models). The file weatherPerth.csv covers the weather conditions in Perth. After loading the CSV file, the data is pre-processed to eliminate any unnecessary variables. For efficient data processing, unwanted variables such as date, risk MM, and location are eliminated from the data frame. Following that, the Boolean variables are given a value of 0 or 1, and missing values are removed. After that, wind characteristics are provided cardinal direction in radians. The pseudo-code for that is given:

```
Start
for var in wind_attributes:
    df[var]=df[var].map(wind_angles)
    df[var+'_cos']=np.cos(df[var])
```

```
    df[var+'_sin']=np.sin(df[var])
  df=df.drop(columns=var)
End
```

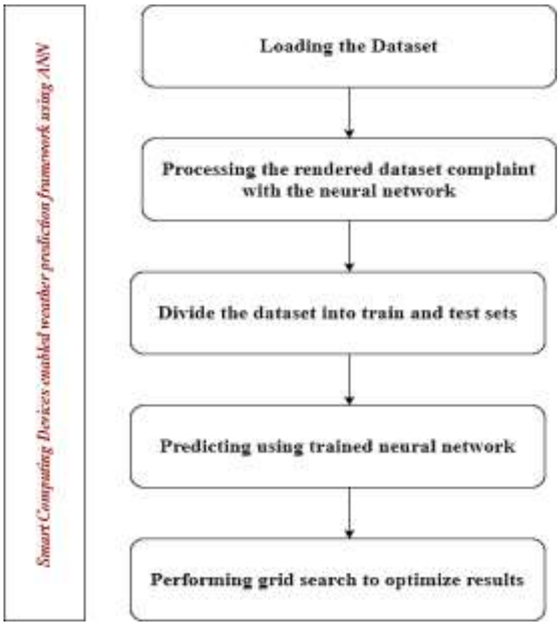

Figure 3. Proposed weather prediction framework

As shown in Figure 4, a simple three-layer neural network is built as the input layer, hidden layer, and output layer. Each node in the next layer to the right is linked to the nodes in the previous layer. Weather conditions should be plugged into the input layer, such that these values flow forward into the network along with the arrows, according to the authors. The weight of each arrow doubles the value it conveys. Incoming values activate the nodes in the buried layer, and this activation is transmitted to the next layer. It is the output layer node that indicates whether or not it will rain tomorrow. The network believes it will rain if its activation is close to one. If it's close to zero, the network believes it won't.

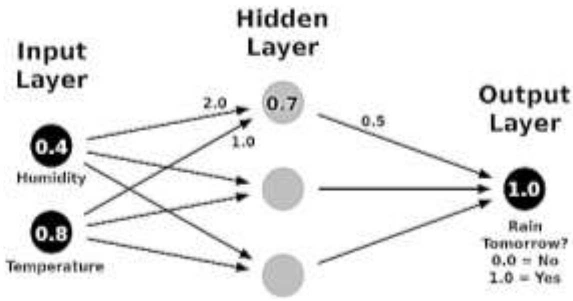

Figure 4. Artificial neural network for weather prediction

3.2. Step 2: processing the rendered dataset complaint with the neural network

In this step, humidity and temperature are considered with only two nodes for the input layer, so this step utilizes two columns from the selected dataset. The authors initially normalize these values first, so all data is in the same range for this to work. This is one reason for having small values like 0.4 and 0.8. These values are fed forward to the next layer along with the arrows. So, focusing on this top node in the hidden layer, which determines the input for the next layer. It is 0.4 from the humidity node times two, and then add the 0.8 from the temperature note that is multiplied by 1. The total input would be 1.6. This goes into an activation function inside the node that determines what the output of the note will be. The output here is 0.7. Therefore, this is the output that is sent to the next layer, multiplying it by 0.5. In this case, with all of the inputs from the hidden layer, the output node is activated to 1.0. This means that the network believes that it will rain tomorrow. Neural network model proposed in this research will include many more connections so that we may use all of the columns from our data. The difficult element is getting the network to suit the data. This entails adjusting all the weights and activation functions within the nodes such that the output provides the correct response. Figure 5 depicts the evolution of the neural network's activation function. The sigmoid function is used for the activation, and it behaves like the sine function.

Pandas is used in this study to convert the chosen dataset and make it suitable with the input layer of the suggested neural network model. First, extraneous columns and duplicate information are deleted, such as the RiskMM property, which is not part of the weather conditions. Because the numerical data in our input layer cannot be used by the study, the easiest solution is to just erase them. By convention, the dropna method removes any row with an empty value. The following phase is concerned with Boolean characteristics. These are all the columns in the dataset that can only have one of two values. Because RainToday and RainTomorrow just have yes or no values, they can be readily mapped from one to zero.

The authors mapped the word yes to the value one and no to zero. For wind direction data, these are encoded as cardinal directions shown in Figure 5 like east-by-southeast or southwest. The small numbers for these neural networks represent these cardinal directions. However, the ordering is preserved by assigning some values like-zero to north, 0.1 to north-by-northeast, and work around this circle: 0, 0.1, 0.2, 0.3... By the time this reaches north-by-northwest, the authors assign the largest number, and yet the smallest number would be assigned to north. Therefore, that is not quite right. The two neighbours should have very similar values to feed into the neural network. Therefore, the research uses sine and cosine of the angles to preserve the cyclic data as illustrated in Figure 6.

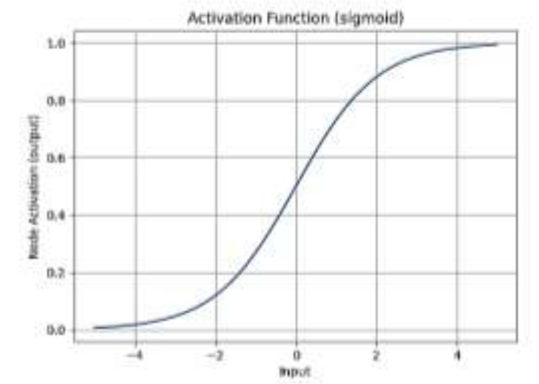

Figure 5. Behaviour of node activation function

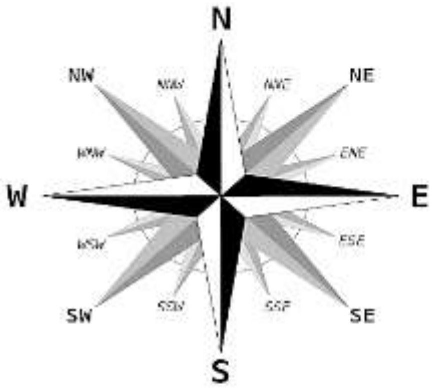

Figure 6. Cardinal directions

3.3. Step 3: divide the dataset into test and train sets

In this step, the authors split the dataset into training and testing sets and normalize it. But before performing that step, the target class RainTomorrow is separated, which is the one value to predict, from all the other columns of data. The target class belongs to the output side of our neural network, not the input layer. The idea is to select rows of data at random and set them aside for later. These are not considered at all while training the neural network. When referring to them later, it is the same as finding examples of weather conditions not seen before. The proposed model can accurately predict whether it rains the next day. sklearn has a convenient function called train_test_split. The pseudo-code for the division of the dataset into test and train is given:

```
Begin
Train_set, Test_set, train_y, test_y=testtrain_splitt(
    X,
    y,
    testsize=0.33,
    randomstate=0
)
print('Train_set', Train_set.shape)
print('Test_set', Test_set.shape)
       End
```

The output of this step is - X_train (2026, 23), X_test (999, 23). Thereafter, scaling of data is done before training the neural network. The pseudo-code for scaling is as follows:

```
Begin
Set_scaler=StandardScaler()
scaler.fit(Train_set)
Train_set=scaler.transform(Train_set)
Test_set=scaler.transform(Test_set)
End
```

Now, the calculation of scaling factors to multiply the values with, so these lie in the same range of small values. Nor that the research does not refer to the testing set yet, so the factors are calculated based on the training set only and then apply these factors to both sets. The authors used the StandardScaler object from the sklearn library and fit the dataset. This ensures the calculation of the adjustment required so that each column of data will have a mean of zero and a standard deviation of one. In the next step, the authors trained the neural network on our training set and determined the accuracy to make predictions.

3.4. Step 4: predicting using trained neural network

In this step, the proposed neural network is trained using an multi-layer perceptron (MLPClassifier). The target class is predicted using the MLPClassifier with the following pseudo-code:

```
Begin
classifier=MLPClassifier(
    hidden_layer_sizes=(50,50),
    iteration_max=500,
    set_ randomstate=0
)
classifier.fit(Train_set, train_y)
End
```

X has all the 23 attributes and is split into two-thirds, one-third subsets. In addition, y has only one value, the target class. To train a neural network, each row of the training data is plugged into the input layer, one-row-at-a-time, and check if the output is correct. If the dataset mentions there was rain tomorrow, then the network's output to be one, otherwise considered zero. Therefore, when the network output is wrong, the authors adjusted the weights and biases a little bit in the direction to make it right. The larger the network, the more intricate the interactions between the input and output may be captured. However, if this is done excessively, the model may overfit the data for training to catch too many tiny features. Then it would generalize extremely well to weather circumstances it had never experienced before. It is always a good idea to set the set_randomstate and consider the maximum number of epochs that it is allowed to take is 500. This neural network (pseudo code given below) can predict whether it will rain tomorrow in Perth with 89.38% accuracy. In the next step, the results are improved by changing the layout of the neural network.

```
Predicting_y=classifier.predict(Test_set)
print(accuracy_score(test_y, Predicting_y))
```

3.5. Step 5: performing grid search to optimize the results

In this final step, the proposed neural network architecture is optimized to improve accuracy. In the previous step, the model had taken two hidden layers of 50 nodes each. But this did not obtain the best possible prediction accuracy from the proposed model. Therefore, the authors decided to try a variety of network layouts to see which one performs well. This task can be automated using the GridSearchCV function of the sklearn library. Therefore, the authors specified the parameters to vary for the proposed model, and this function will take care of evaluating the performance. The authors performed one layout with just a single hidden layer with two nodes, a single hidden layer with 10 nodes, and then tried with the original layout: two layers of 50 nodes each. A neural network with all the parameters that are not going to be varying during this test, the authors increased the maximum number of epochs to 2,000 for this to work and set the random_state to zero again. The pseudo-code for the same is given:

```
print(gridsearch.cv_results_['params'])
print(gridsearch.cv_results_['mean_test_score'])
best_nn=gridsearch.best_estimator_
y_pred=best_nn.predict(Test_set)
print(accuracy_score(test_y, Predicting_y))
```

4. RESULTS AND DISCUSSION

Initially, the data is read from the dataset using the .csv file, the authors utilized Google collaborator for adding the .csv file. The dataset is analysed first to check the redundancy and any anomalies. The missing values from the dataset were dropped to make the dataset fit for creating the weather prediction framework. To train a neural network, the authors plugged each row of the training dataset into the input layer, one-row-at-a-time, and checked if the output is correct. If the data indicates there will be rain tomorrow, the model needs to have the network's output to be one or zero otherwise. Therefore, when the network output is wrong, the authors adjust the weights and biases a little bit in the direction to make it right. The focus is for the best fit over all the dataset, not just for one row. So, completing a single run through all our data is called an epoch of training. The model carries out many epochs for the network to be trained. The larger our network, the more intricate interactions between input and output may be captured. However, if this is done to an excessive degree, the model may overfit the training dataset and catch too many small features. Then it won't be able to generalize about weather situations it hasn't encountered before. The optimal network layout is chosen once the model has been trained. the pseudo-code for the same is given:

```
Start
parameters=
{
   'hidden_layer_sizes':
    (
       (2,), (10,), (50,50),
    )
}
nn=MLPClassifier(max_iter=2000, random_state=0)
gridsearch=GridSearchCV(nn, parameters, cv=3)
           gridsearch.fit(Train_set, train_y)
End
```

Here in the above pseudo-code, we have used the multi-layer perceptron classifier and used the GridSearch CV method to take the parameters. Thereafter, the grid results are displayed using the pseudo code given:

```
print(gridsearch.cv_results_['params'])
                  print(gridsearch.cv_results_['mean_test_score'])
```

the accuracy of various models came out to be:

```
[{'hidden_layer_sizes': (2,)}, {'hidden_layer_sizes': (10,)}, {'hidden_layer_sizes': (50, 50)}]
                 [0.90424355 0.89487179 0.88450873]
```

in the final step, we selected the best neural network for prediction using the pseudo code given:

```
best_nn=gridsearch.best_estimator_
y_pred=best_nn.predict(Test_set)
               print(accuracy_score(test_y, Predicting_y))
```

The mathematical calculation done to find out the accuracy is given (1). Since we have used the multi-layer perceptron model, the feed-forward neural network is used. The procedure to calculate the discontinuous function is also given:

$$w = w + lr * (E - P) * y \tag{1}$$

Here, in the (1), w represents weight, LR represents the rate of learning, E represents the expected value, P represents the predicted value:

$$accuracy = (myconfusionMatrix[0][0] + myconfusionMatrix[1][1]) / np.sum(myconfusionMatrix) \tag{2}$$

$$precision = myconfusionMatrix[0][0] / (myconfusionMatrix[0][0] + myconfusionMatrix[0][1]) \tag{3}$$

$$recall = myconfusionMatrix[0][0] / (myconfusionMatrix[0][0] + myconfusionMatrix[1][0]) \tag{4}$$

$$specificity = myconfusionMatrix[1][1] \,/\, (myconfusionMatrix[1][1] + myconfusionMatrix[0][1]) \quad (5)$$

$$f1score = 2 * myconfusionMatrix[0][0] \,/\, (2 * myconfusionMatrix[0][0] + myconfusionMatrix[1][0] + myconfusionMatrix[0][1]) \quad (6)$$

accuracy for the proposed weather prediction model is calculated using the formula:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (7)$$

Where:
- True positives (TP) are when actual and anticipated values are both positive.
- True negatives (TN) occur when both actual and anticipated values are negative.
- False positives (FP) are situations where the actual value is negative while the anticipated value is positive.
- False negatives (FN) are situations in which the actual value is positive while the anticipated value is negative.

We have used the Sigmoid function in the MLP model. A sigmoid function is a mathematical function that has a characteristic S-shaped curve. Sigmoid functions include the logistic sigmoid function, the hyperbolic tangent, and the arctangent. The mathematical equation for the sigmoid function is given:

$$y = \frac{1}{1+ e^{-x}} \quad (8)$$

it has a derivative, which is calculated as:

$$\frac{dy}{dx} = \frac{e^{x}}{(1+ e^{x})^2} \quad (9)$$

the algorithm used for the implementation is:

```
class NN:
  def __init__(self, inputs):
    self.inputs = inputs
    self.l = len(self.inputs)
    self.li = len(self.inputs[0])
    self.wi = np.random.random((self.li, self.l))
    self.wh = np.random.random((self.l, 1))
  def think(self, inp):
    s1 = sigmoid(np.dot(inp, self.wi))
    s2 = sigmoid(np.dot(s1, self.wh))
    return s2
  def train(self, inputs, outputs, it):
    for i in range(it):
      l0 = inputs
      l1 = sigmoid(np.dot(l0, self.wi))
      l2 = sigmoid(np.dot(l1, self.wh))
      l2_err = outputs - l2
      l2_delta = np.multiply(l2_err, sigmoid_der(l2))
      l1_err = np.dot(l2_delta, self.wh.T)
      l1_delta = np.multiply(l1_err, sigmoid_der(l1))
      self.wh += np.dot(l1.T, l2_delta)
      self.wi += np.dot(l0.T, l1_delta)
inputs = np.array([[0,0], [0,1], [1,0], [1,1] ])
                    outputs = np.array([ [0], [1], [1], [0] ])
```

The accuracy score was calculated using the sklearn library and compared known correct RainTomorrow values with the ones that were predicted. Therefore, this neural network predicted whether it rains Tomorrow with 90.42% accuracy. To assess the effectiveness of the model, comparison research is conducted with the other existing models in the same area. The result demonstrates that our approach is better in comparison to other similar products. The comparison analysis is done utilizing the one available technique, and the comparative analysis is shown in Table 2.

Table 2. Comparative analysis with similar approaches

Method name	Accuracy (%)
McGovern *et al.* [25]	84.9
Akbarian *et al.* [26]	70
Proposed method	90.4

The results of this study show that using a MLP neural network can lead to potential improvements in weather prediction. Our model uses the sklearn toolkit to instantiate a neural network with two hidden layers of 50 nodes each, and it has shown remarkable success in predicting weather patterns. Ensuring the reliability and efficiency of our technique required restricting the maximum number of epochs to 500 and setting the random state during model training. These variables support the model's stability and dependability, enabling constant and precise predictions. A comparative study using current models in the area confirms our suggested approach's advantages. After thorough evaluation with several methods, such as comparative analysis and comparison research, our model regularly performs better than other comparable items. This demonstrates our methodology's resilience and effectiveness in weather prediction challenges.

Accuracy metrics offer a thorough understanding of our model's performance, encompassing both positive and negative situations for actual and predicted values. With a final accuracy of 90.42% for the next day's prediction, our model has a high degree of precision and dependability. Not only is this degree of accuracy remarkable, but it also outperforms other research models, underscoring the importance of our contributions to the area. Our work has a significant impact on several industries, including transportation, agriculture, and disaster relief, that depend on precise weather forecasting. The effectiveness and dependability of weather prediction systems have improved significantly because of our use of cutting-edge machine learning techniques. The study's findings show how well a multi-layer perceptron neural network performs when used for weather prediction applications. By means of thorough testing and analysis, we have demonstrated the superiority of our approach over current alternatives. Future work on this topic and methodological improvements has the potential to boost weather forecasting even more, which would eventually benefit society as a whole.

5. CONCLUSION

The model started with two hidden layers of 50 nodes each, the authors instantiated a neural network from the sklearn library. The model uses a multi-layer perceptron for weather prediction. While training the model, it is always a good idea to set the random state, the maximum number of epochs that the proposed model can take is 500. To assess the effectiveness of the model, comparison research is conducted with the other existing models in the same area. The result demonstrates that our approach is better in comparison to other similar products. The comparison analysis is done utilizing the one available technique, and the comparative analysis. The accuracy is obtained for actual and anticipated values as positive, both actual and anticipated values are negative, situations where the actual value is negative while the anticipated value is positive as well as situations in which the actual value is positive while the anticipated value is negative. The final accuracy obtained for next-day prediction is 90.42%, which is efficient and better than other research models.

REFERENCES

[1] M. Shafiq, Z. Gu, O. Cheikhrouhou, W. Alhakami, and H. Hamam, "The rise of 'internet of things': Review and open research issues related to detection and prevention of IoT-based security attacks," *Wireless Communications and Mobile Computing*, vol. 2022, pp. 1–12, Aug. 2022, doi: 10.1155/2022/8669348.

[2] J. MacWilliams, "Convergence of smart computing device markets," *Connector Supplier*, 2015. https://connectorsupplier.com/convergence-of-smart-computing-device-markets/ (accessed Apr. 06, 2023).

[3] "Forecast end-user spending on IoT solutions worldwide from 2017 to 2025," *Statista*. https://www.statista.com/statistics/976313/global-iot-market-size/ (accessed Sep. 20, 2021).

[4] "A.I. model shows promise to generate faster, more accurate weather forecasts," *UW News*. https://www.washington.edu/news/2020/12/15/a-i-model-shows-promise-to-generate-faster-more-accurate-weather-forecasts/ (accessed Apr. 06, 2023).

[5] R. Ali *et al.*, "Financial hazard prediction due to power outages associated with severe weather-related natural disaster categories," *Energies*, vol. 15, no. 24, Dec. 2022, doi: 10.3390/en15249292.

[6] M. Hussain *et al.*, "Machine learning-driven approach for a COVID-19 warning system," *Electronics*, vol. 11, no. 23, Nov. 2022, doi: 10.3390/electronics11233875.

[7] U. Ishfaq *et al.*, "Empirical analysis of machine learning algorithms for multiclass prediction," *Wireless Communications and Mobile Computing*, pp. 1–12, Mar. 2022, doi: 10.1155/2022/7451152.

[8] A. B. Tufail *et al.*, "Early-stage Alzheimer's disease categorization using PET neuroimaging modality and convolutional neural networks in the 2D and 3D domains," *Sensors*, vol. 22, no. 12, Jun. 2022, doi: 10.3390/s22124609.

[9] N. Singh, S. Chaturvedi, and S. Akhter, "Weather forecasting using machine learning algorithm," in *2019 International Conference on Signal Processing and Communication (ICSC)*, IEEE, Mar. 2019, pp. 171–174, doi: 10.1109/ICSC45622.2019.8938211.

[10] B. Vasantha, R. Tamilkodi, and L. V. Kiran, "Rainfall pattern prediction using real time global climate parameters through machine learning," in *2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN)*, IEEE, Mar. 2019, pp. 1–4, doi: 10.1109/ViTECoN.2019.8899463.

[11] A. Durrani, M. Khurram, and H. R. Khan, "Smart weather alert system for dwellers of different areas," in *2019 16th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, IEEE, Jan. 2019, pp. 333–339, doi: 10.1109/IBCAST.2019.8667190.

[12] F. Jafari, J. Mousavi, K. Ponnambalam, F. Karray, and L. Nassar, "Machine learning tools for the prediction of fresh produce procurement price," in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, IEEE, Oct. 2020, pp. 3091–3095, doi: 10.1109/SMC42975.2020.9282887.

[13] M. Jadhav, N. Kolambe, S. Jain, and S. Chaudhari, "Farming made easy using machine learning," in *2021 2nd International Conference for Emerging Technology (INCET)*, IEEE, May 2021, pp. 1–5, doi: 10.1109/INCET51464.2021.9456351.

[14] M. Hatim, F. Siddiqui, and R. Kumar, "Addressing challenges and demands of intelligent seasonal rainfall forecasting using artificial intelligence approach," in *2020 International Conference on Computation, Automation and Knowledge Management (ICCAKM)*, IEEE, Jan. 2020, pp. 263–267, doi: 10.1109/ICCAKM46823.2020.9051516.

[15] S. Scher and J. Molinder, "Machine learning-based prediction of icing-related wind power production loss," *IEEE Access*, vol. 7, pp. 129421–129429, 2019, doi: 10.1109/ACCESS.2019.2939657.

[16] B. L. Pavuluri, R. S. Vejendla, P. Jithendra, T. Deepika, and S. Bano, "Forecasting meteorological analysis using machine learning algorithms," in *2020 International Conference on Smart Electronics and Communication (ICOSEC)*, IEEE, Sep. 2020, pp. 456–461, doi: 10.1109/ICOSEC49089.2020.9215440.

[17] N. Suresh *et al.*, "Crop yield prediction using random forest algorithm," in *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 2021, pp. 279–282. doi: 10.1109/ICACCS51430.2021.9441871.

[18] J. Arshad *et al.*, "Implementation of a LoRaWAN based smart agriculture decision support system for optimum crop yield," *Sustainability*, vol. 14, no. 2, Jan. 2022, doi: 10.3390/su14020827.

[19] S. V. Joshua *et al.*, "Crop yield prediction using machine learning approaches on a wide spectrum," *Computers, Materials and Continua*, vol. 72, no. 3, pp. 5663–5679, 2022, doi: 10.32604/cmc.2022.027178.

[20] S. Vulova, F. Meier, D. Fenner, H. Nouri, and B. Kleinschmit, "Summer nights in Berlin, Germany: modeling air temperature spatially with remote sensing, crowdsourced weather data, and machine learning," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 5074–5087, 2020, doi: 10.1109/JSTARS.2020.3019696.

[21] G. Ben Brahim, "Weather conditions impact on electricity consumption in smart homes: machine learning based prediction model," in *2021 8th International Conference on Electrical and Electronics Engineering (ICEEE)*, IEEE, Apr. 2021, pp. 93–98, doi: 10.1109/ICEEE52452.2021.9415917.

[22] R. Sharma, S. Rani, and I. Memon, "A smart approach for fire prediction under uncertain conditions using machine learning," *Multimedia Tools and Applications*, vol. 79, no. 37–38, Oct. 2020, doi: 10.1007/s11042-020-09347-x.

[23] S. Ogale and S. Srivastava, "Modelling and short term forecasting of flash floods in an urban environment," in *2019 National Conference on Communications (NCC)*, IEEE, Feb. 2019, pp. 1–6, doi: 10.1109/NCC.2019.8732193.

[24] S. Rani, M. Kaur, M. Kumar, V. Ravi, U. Ghosh, and J. R. Mohanty, "Detection of shilling attack in recommender system for YouTube video statistics using machine learning techniques," *Soft Computing*, vol. 27, no. 1, pp. 377–389, Jan. 2023, doi: 10.1007/s00500-021-05586-8.

[25] A. McGovern *et al.*, "Using artificial intelligence to improve real-time decision-making for high-impact weather," *Bulletin of the American Meteorological Society*, vol. 98, no. 10, pp. 2073–2090, Oct. 2017, doi: 10.1175/BAMS-D-16-0123.1.

[26] M. Akbarian, B. Saghafian, and S. Golian, "Monthly streamflow forecasting by machine learning methods using dynamic weather prediction model outputs over Iran," *Journal of Hydrology*, vol. 620, May 2023, doi: 10.1016/j.jhydrol.2023.129480.

BIOGRAPHIES OF AUTHORS

Keshav Kaushik is an accomplished academician, cybersecurity, and digital forensics expert currently serving as an Assistant Professor at the Amity School of Engineering and Technology, Amity University Punjab, India. As a key member of the Cybersecurity Centre of Excellence, he has been instrumental in advancing the field of cybersecurity through his dedicated teaching and innovative research. In addition to his academic role, he holds the prestigious position of Vice-Chairperson for the Meerut ACM Professional Chapter, highlighting his leadership and commitment to the professional community. His academic journey includes a notable stint as a Faculty Intern during the Summer Faculty Research Fellow Programme 2016 at the Indian Institute of Technology (IIT) Ropar, reflecting his continuous pursuit of knowledge and professional development. His scholarly contributions are extensive and impactful, with over 135 publications to his credit. This includes 25 peer-reviewed articles in SCI/SCIE/Scopus-indexed journals and 50+ publications in Scopus-indexed conferences. He is also an inventor, holding one granted patent and six published patents, alongside five granted copyrights. His editorial expertise is showcased by publishing 30 books and 25 book chapters, further cementing his reputation as a thought leader in the field. His professional certifications are a testament to his expertise and commitment to excellence. He is a Certified Ethical Hacker (CEH v11) by EC-Council, a CQI and IRCA Certified ISO/IEC 27001:2013 Lead Auditor, a Quick Heal Academy Certified Cyber Security Professional (QCSP), and an IBM Cybersecurity Analyst. He can be contacted at email: officialkeshavkaushik@gmail.com.

Gunjan Chhabra is a Ph.D. in computer science and engineering, working as Associate professor in the Department of Computer science and engineering, Graphic Era Hill University, Dehradun. He has published ten patents, with four Granted patents from IPR. Also, he has published several research papers in various renowned journals. Additionally, he has authored three text book on the domain IoT, smart technologies and machine learning. He has also mentored many projects in different domains of computer science to solve real-world problems. Under his guidance, his students have successfully incubated their ideas. He can be contacted at email: chhgunjan@gmail.com.

Salil Bharany is a proactive Assistant Professor and researcher with over 7 years of experience in teaching undergraduate and postgraduate courses. He earned his B.Tech from Punjab Technical University, and his M.Tech and Ph.D. (CSE) from Guru Nanak Dev University, India, respectively. His research focuses on Energy, WSN, IoT, social networks, artificial intelligence, cybersecurity, big data analytics, biomedical signal processing, and renewable energy technologies. He regularly serves as a TPC member for several conferences and performs reviews for leading journals such as Wiley, Taylor and Francis, IEEE, MDPI, Elsevier, and Springer. He can be contacted at email: salil.bharany@gmail.com.

Ateeq Ur Rehman received his BS Degree in Electrical (Telecommunication) Engineering from COMSATS Institute of Information Technology, Lahore, Pakistan, in 2009 and MS degree in Electrical Engineering with a specialization in Telecommunications from the Blekinge Institute of Technology (BTH), Karlskrona, Sweden in 2011. He completed his Ph.D. degree in the College of Internet of Things (IoT) Engineering, Hohai University (HHU), Changzhou Campus, China, in 2022. Currently, he is an assistant professor with the Faculty of Computing, Gachon University, South Korea. He has contributed to various international IEEE conferences and journals of repute. His research interests include but are not limited to biomedical signal processing, IoTs, social internet of things (SIoTs), big data. He can be contacted at email: 202411144@gachon.ac.kr.

Habib Hamam obtained the B.Eng. and M.Sc. degrees in information processing from the Technical University of Munich, Germany 1988 and 1992, and the Ph.D. degree in Physics and applications in telecommunications from Université de Rennes I conjointly with France Telecom Graduate School, France 1995. He also obtained a postdoctoral diploma, "Accreditation to Supervise Research in Signal Processing and Telecommunications", from Université de Rennes I in 2004. He was a Canada Research Chair holder in "Optics in Information and Communication Technologies", the most prestigious research position in Canada-which he held for a decade (2006-2016). The title is awarded by the Head of the Government of Canada after a selection by an international scientific jury in the related field. He is currently a full Professor in the Department of Electrical Engineering at Université de Moncton. He is OSA senior member, IEEE senior member and a registered professional engineer in New-Brunswick. He obtained several pedagogical and scientific awards. He is among others editor in chief and founder of CIT-Review, academic editor in Applied Sciences and associate editor of the IEEE Canadian Review. He also served as Guest editor in several journals. His research interests are in optical telecommunications, wireless communications, diffraction, fiber components, signal and image processing, IoT, data protection, AI and big data. He can be contacted at email: habib.hamam@umoncton.ca.

.