An Optimal K-Nearest Neighbor for Weather Prediction Using Whale Optimization Algorithm

Rajalakshmi Shenbaga Moorthy, St. Joseph’s Institute of Technology, Chennai, India*
Pabitha Parameshwaran, Madras Institute of Technology, India

ABSTRACT
The weather has a serious impact on the environment as its effects change day-to-day life. In recent days, many algorithms have been proposed to predict the weather. Although various machine learning algorithms predict the weather, the optimal prediction of weather is not addressed. Optimal prediction of weather is required as it has a serious impact on human life. Thus, this domain invites an optimal system that can forecast weather thereby saving human life. To optimally predict the changes in weather, a metaheuristic algorithm called whale optimization algorithm (WOA) is integrated with machine learning algorithm k-nearest neighbor (K-NN). Whale optimization is an algorithm inspired by the social behavior of whales. The proposed WOAK-NN is compared with K-NN. The integration of WOA with K-NN aims to maximize accuracy, F-measure, and minimize mean absolute error. Also, the time complexity of WOAK-NN is compared with K-NN, and it is observed that when the dataset is large, WOAK-NN requires minimum time for an optimal prediction.

KEYWORDS
Accuracy, F-Measure, K-Nearest Neighbor, Machine Learning Algorithms, Mean Absolute Error, Metaheuristic Algorithm, Time Complexity, Weather, Whale Optimization Algorithm

1. INTRODUCTION
Weather forecasting is a mechanism to predict weather i.e. changes in the atmospheric conditions for a given location. It is essential to predict weather as it has a significant impact on both living and non-living things. Forecasting weather is a crucial one as the changes in weather has a severe impact on the life of people, agricultural field (Kaur, S., & Cheema, S. S. 2017). Changes in weather are due to the molecules such as oxygen, carbon dioxide, Nitrogen dioxide and sulfur dioxide (Kaur, S., & Cheema, S. S. 2017). Weather forecasting can protect the people from disasters like Tsunami, heavy rainfall and also protects the crops thereby helping the farmers. The forecasting of weather is important in many fields such as climate monitoring, detecting drought, production of agriculture crops, pollution control etc. Forecasting weather is still a major issue in the department of metrology. Though, the technology grows in advance, the accurate prediction of weather is still a crucial question. Several researchers are concentrating on forecasting weather by building a model using machine learning algorithms. Now a day’s lot of IoT devices is used to sense the changes in parameters such as temperature, wind, humidity, level of oxygen etc. Thus using these measurements, a model can
be built using machine learning algorithms for accurate prediction of weather. Numerical Weather prediction is also widely used to predict the weather.

There are two methods used to predict weather viz. Empirical method and Dynamic method. Empirical method is the one which relies on past historical data to build the model to predict the future. The machine learning algorithms such as decision tree, Linear regression etc. fall into empirical methods. Dynamic methods are the one where the expectations from the physical model are used to predict the future (Bhatkande, S. S., & Hubballi, R. G. 2016). The dynamic approach is otherwise termed to be computer modeling and it is suitable for predicting large scale weather condition and does not forecast short term weather condition (Devi, C. J., Reddy, B. S. P., Kumar, K. V., Reddy, B. M., & Nayak, N. R. 2012).

Though there exists several machine learning algorithms for forecasting weather by predicting, the optimal prediction is still a concern. Thus the authors, integrated metaheuristic algorithm with machine learning algorithm for optimal prediction of weather. Some of the metaheuristic algorithms used for various applications are penguins search optimization algorithm for community detection (Guendouz, M., Amine, A., & Hamou, R. M. 2018), Genetic algorithm for vehicle routing problem (Yahyaoui, H., Krichen, S., & Dekdouk, A. 2018), Particle swarm optimization for task scheduling (Yin, P. Y., Yu, S. S., Wang, P. P., & Wang, Y. T. 2006), diabetes prediction using metaheuristic algorithm (Parameswaran, P., & Moorthy, R. S. 2019), provisioning of analytics as a service (Moorthy, R. S., & Pabitha, P. 2019) etc.. Also various metaheuristic algorithms studied in (Moorthy, R. S., & Pabitha, P. 2018, August) are analyzed and Whale Optimization algorithm is chosen to integrate with K-Nearest Neighbor (K-NN) for optimal prediction of weather. K-Nearest neighbor assigns an instance to a class to which the most of its neighbors belong. It is based on the concept that, if a new instance is similar to the instances, then the class of the instances is also same. The similarity of the instances is computed based on distance. If the distance between k-neighbors is minimum, then the most common class of K-Neighbors is assigned to the new instance. Whale Optimization Algorithm (WOA) which was proposed by (Mirjalili, S., & Lewis, A. 2016), is a metaheuristic algorithm inspired by the behavior of whales. WOA has an ability to switch between exploration and exploitation prominently using a single control parameter. WOA algorithm is easy to implement because of the limited control parameters, and also exhibit great level of flexibility. During exploration phase, the position of the solutions is updated using the random search agent rather than the best agent. While in the exploitation, the position of the solutions is updated using the position of the best agent. Also WOA has the tendency to balance between exploration and exploitation (Mohammed, H. M., Umar, S. U., & Rashid, T. A. 2019). Due to this great set of features, WOA is used and integrated with K-NN for optimal prediction of weather. Whether the new algorithm is accurately predicting weather may arise. Every problem can be solved by number of ways by more than one algorithm. There is a always a space for improvement of the solution for particular problem. There is always better methods exist to accurately predict the weather. Motivating by this, the authors propose hybridization of whale optimization algorithm and K-NN for weather prediction.

2. STATE OF THE ART

Artificial neural network and decision tree was designed to frame classification rules for predicting weather using the data gathered from Ibadan synoptic airport station. C5 decision tree algorithm was used to analyze weather data. Time lagged feed forward network and recurrent networks implemented using neuro solutions 6 were used for forecasting windspeed, evaporation, radiation, minimum temperature, maximum temperature and rainfall, given the inputs such as month and year. C5 algorithm were implemented in see5 software and it works based on information theory and generated both decision tree and rules. 10 fold cross validation was used and the algorithm was run for 10 times and the tree that produced minimum error was chosen. Still the accuracy of the tree is a concern in predicting weather as it paves for huge disaster (Olaiya, F., & Adeyemo, A. B.2012). Decision tree
was used for predicting weather using metrological data collected for 10 years. The attributes taken into account are year, month, average pressure, relative humidity, clouds quantity, precipitations and average temperature. CART algorithm in WEKA (Waikato Environment for Knowledge Analysis) was used to build decision tree to predict the future. The average temperature is forecasted with some level of accuracy (Petre, E. G. 2009). Support vector machine (SVM) was used for predicting atmospheric temperature. Daily maximum temperature is taken as input to create a model for forecasting the next day temperature. SVM based model for analyzing time series data was compared with multi layered perceptron (MLP) and the results of SVM was better. The reason behind the better performance of SVM is that it uses the principle of structured risk minimization (SRM) which thereby minimizes the upper bound of generalization error and also avoids the problem of over fitting. SVM had minimum mean square error than MLP (Radhika, Y., & Shashi, M. 2009).

Data mining techniques such as outlier analysis, clustering, prediction, classification, association rules mining was used to analyze meterological data at Gaze strip city collected for nine years. From the data gathered, outliers are removed based on the distance metric. Having removed the outliers K-Means algorithm is used to cluster the meterological data to group the regions which receives highest and least rainfall. Artificial neural network (ANN) and least median square linear regression was used to predict the daily average temperature. ANN achieves better correlation coefficient and lower root mean square error (Kohail, S. N., & El-Halees, A. M. 2011). Forest fire was predicted using data mining techniques such as logistic regression, decision tree, random forests, bagging and boosting of decision trees. The experimentation is carried over three datasets such as Kras, Primorska and continental Slovenia. Bagging of decision trees achieves better performance in terms of metrics such as accuracy, precision and Kappa statistics (Stojanova, D., Panov, P., Kobler, A., Dzeroski, S., & Taskova, K. 2006, October). Forecasting weather using machine learning techniques such as linear regression and variation of functional regression mode was sonel. The attributes taken into account are maximum temperature, minimum temperature, mean humidity, mean atmospheric pressure and weather classifications observed for four years from 2011 – 2015. Linear regression was used to predict the high and low temperature. The cost function for linear regression is minimizing the error between observed and predicted values. The variation of functional regression predicts the future based on past history. Both linear and function regression had better results than the conventional one. Linear regression is a low bias and high variance model, thus the occurrence of outliers is more, which can be avoided by collecting more training samples. On the otherhand functional regression, which is high bias and low variance model predominately shows that prediction accuracy could not be increased on further collection of data (Holmstrom, M., Liu, D., & Vo, C. 2016).

An integrated approach using data mining and forecasting analysis was designed for predicting weather. Weather data is considered to be time series data and thus can be analyzed using Auto Regressive Integrated Moving Average (ARIMA) or Auto Regressive Moving Average (ARMA) model. The data for 97 days gathered from Visakhapatnam city for various attributes such as wind pressure, humidity, minimum and maximum temperature. The ARIMA model was used to predict the weather for next 15 days (Krishna, G. V. 2015). Back propagation algorithm was used for forecasting weather using parameters such as rainfall, wind, temperature and humidity gathered using rain sensor, wind sensor and thermo-hygro sensor (Sawaitul, S. D., Wagh, K. P., & Chatur, P. N. 2012). FP growth algorithm is used for forecasting the weather given the past history (Dhore, A., Byakude, A., Sonar, B., & Waste, M. 2017).

Various data mining techniques were reviewed and found that K-Means and decision tree algorithm are better for weather prediction (Chauhan, D., & Thakur, J. 2014). Artificial neural network was used to forecast weather. The nn tool in Matlab 7.6.0 was used to implement feed forward neural network with back propagation. The dataset has 365 instances and 10 features. 60% of the instances are treated as training, 20% as validation and remaining 20% as test set. The performance of the neural network is evaluated using the metric called mean squared error (MSE). Some of the observations were: When the number of samples or neurons or hidden layers increases, the MSE decreases. Tan
sigmoid function is used as an activation function in hidden layer for better accuracy (Abhishek, K., & Singh, M. P. 2012). Artificial neural network was designed for predicting rainfall. The three layer perceptron (3LP) uses the sigmoid function for predicting the target output. The weights of the neural network are optimized using back propagation and random optimization methods (Kashiwao, T., Nakayama, K., Ando, S., Ikeda, K., Lee, M., & Bahadori, A. 2017)). Neural network and fuzzy inference system was used for forecasting weather. The neural network is trained using parameters such as humidity, temperature, pressure, wind speed, dew and visibility. The predicted values are then passed to fuzzy inference system for prediction of weather (Khajure, S., & Mohod, S. W. 2016).

From the survey, it’s been observed that, there is a need of optimal prediction of weather as it plays a serious role in saving people from disaster too. Also, the wide applicability of machine learning algorithms in various fields including health care for example Parkinson’s disease prediction (Patra, A. K., Ray, R., Abdullah, A. A., & Dash, S. R. 2019, November), prediction of heart disease (Panda, D., Ray, R., Abdullah, A. A., & Dash, S. R. 2019) and predicting quality of semen (Dash, S. R., & Ray, R. 2020) motivates the authors to apply machine learning algorithms for weather prediction.

2.1 Whale Optimization Algorithm (WOA)

Whale Optimization Algorithm is a metaheuristic algorithm that is inspired by the behavior of whales. Whales hunt their prey by circling around them and that behavior is modeled for searching the best solution. Any metaheuristic algorithm has two prime features viz. exploitation and exploration. The exploitation phase includes encircling a prey and doing spiral bubble net attacking method. In the exploration phase, the whales search for the prey in spiral path. Mathematical representation for WOA is represented as follows:

i) Encircling Prey: Each whale is represented by a position $W_i$ and they can identify the position of the target prey and try to encircle it. At each iteration, each whale compute the position based on the position of the best whale which is very nearer to the prey. The distance between the position of the best whale which is nearer to the prey and the $i^{th}$ whale is computed using equation (1).

$$\text{Dist} \leftarrow |C \ast W'_i - W'_i| \quad (1)$$

Where $t$ represents the current iteration number. The position of the $i^{th}$ whale at the iteration $t+1$ is computed using equation (2).

$$W'^{t+1}_i \leftarrow W'_i - A \ast \text{Dist} \quad (2)$$

Where $W'_i$ represents the position of the best whale at iteration $t$, $A$ and $C$ are coefficient vectors, calculated using equation (3) and (4).

$$A \leftarrow 2 \ast a \ast r - a \quad (3)$$

$$C \leftarrow 2 \ast r \quad (4)$$
\( \vec{r} \) is a random vector uniformly distributed in the interval \([0,1]\). \( \vec{a} \) is a variable that is initialized to 2 and gradually decreased to 0. The coefficient vectors \( \vec{A} \) and \( \vec{C} \) play a prominent role in adjusting the position of whale towards the position of the best whale. The encircling behavior can be achieved by using the variable \( \vec{a} \) which is computed using equation (5).

\[
a = 2 - t. \frac{2}{\text{MAX_ITER}}
\]  

(5)

\( \text{MAX_ITER} \) represents the maximum number of iteration. Equation (2) represents the mathematical model for encircling the prey.

ii) Bubble-net attacking method: Whales either tend to move towards the prey based on the best search agent or they can update the position spirally. The choice between updating the path is done by using the probability. The position of the whale when travelling in spiral path is computed using equation (6).

\[
\vec{W}_{i}^{t+1} \leftarrow \text{Dist}' e^{b} \cdot \cos(2\pi l) + \vec{W}_{i}^{t}
\]  

(6)

Where \( b \) is a constant, \( l \) is a random number in interval \([-1,1]\). \( \text{Dist}' \) is the distance between \( i^{th} \) whale and best whale represented in equation (7).

\[
\text{Dist}' \leftarrow |\vec{W}_{i}^{t+1} - \vec{W}_{i}^{t}|
\]  

(7)

These two mechanisms should be given equal probability which is represented in equation (8).

\[
\vec{W}_{i}^{t+1} \leftarrow \begin{cases} 
\text{Equation (2)} & \text{if } p < 0.5 \\
\text{Equation (6)} & \text{else}
\end{cases}
\]  

(8)

iii) Search for Prey: In the exploration phase, the position of the whale is computed based on any random whale rather than the best whale represented in equation (10). The corresponding \( \text{Dist} \) is also computed based on the position of the random whale represented in equation (9).

\[
\text{Dist} \leftarrow |C \cdot \vec{W}_{\text{rand}}^{t} - \vec{W}_{i}^{t}|
\]  

(9)

\[
\vec{W}_{i}^{t+1} \leftarrow \vec{W}_{\text{rand}}^{t} - \vec{A}\text{Dist}
\]  

(10)
The value of the coefficient vector is taken into account to switch between exploration and exploitation. If the value of $\vec{A}$ is less than 1, exploitation takes place i.e. whale’s position will be updated based on the position of the best whale. When the value $\vec{A}$ is greater than 1, exploration happens i.e. updating the position of the whale based on the position of the random whale. The convergence is guaranteed, since in the exploitation phase, the best position of the whale is used to update any other whale’s position. If the position of the whales are computed only by the best position, then it results in striking local optima. To get rid of it, the parameter $\vec{a}$ is used which balances between convergence and avoiding local optima.

2.2 K-Nearest Neighbor (K-NN)

The K-Nearest Neighbor is a non parametric classifier which does not have to know information about underlying data that made the algorithm more interesting as for real world problems the distribution of the data is not known in advance. KNN is otherwise termed to be instance based learning or lazy learning where the computation is postponed until classification and primary application of KNN includes pattern recognition, feature selection, statistical estimation etc. KNN computes the neighbor of test instance using Euclidean distance. For a new instance, the distance is calculated for all the training instances in the dataset and the distance is sorted to find the K-neighbors. The class will be assigned to the new instance based on the majority of the neighbor’s class. The problem with KNN is how to choose K, so that the classification error is minimum. If the value of K is chosen as 1, then KNN assigns a class label of the nearest 1 neighbor which may not be optimal. On the other hand, if the value of K is large, then voting is done by K neighbors and the class label of majority of the neighbors is assigned. The problem with the approach is that neighbors may be influenced by the outliers of other class as shown in Figure 1.

3. WOAK-NN

The block diagram for the proposed is shown in the figure 2. The weather dataset is present in weather data store, which is used by the preprocessor to fill the missing values and to normalize the values. The preprocessed dataset is given as input to WOAK-NN. WOAK-NN is an integration of whale optimization algorithm and K-Nearest Neighbor. The K-NN which is a lazy learning algorithm assigns a class label to the new instance based on the vote given by the nearest neighbors. The biased nature of the voting causes a problem of optimal assignment of class label. To overcome the biasing nature of voting, K-NN is integrated with WOAK-NN. The flowchart of the proposed WOAK-NN is shown in figure 3.

WOA algorithm uses K-NN as a fitness function for optimal prediction of weather. Algorithm 1 and 2 represents the prediction of weather using WOAK-NN. The instances in the dataset are represented as a whale. For each whale, the fitness value is computed using K-NN classifier. The whale which is having minimum fitness value is the best whale. The exploration and exploitation is balanced with probability value set as 0.5 and also using coefficient vector $\vec{A}$. If the probability value is less than 0.5 and the coefficient vector is less than 1, then the whales does encircling prey, which is the process of exploitation i.e. the next position of the whale is computed based on the position of the best whale. If the probability value is less than 0.5 and the coefficient vector is greater than or equal to 1, then the whales does exploration, i.e. the position of the whale is computed based on the position of any random whale. If the probability value is greater than or equal to 0.5, then the whales move in a spiral path where the next position is based on the position of best whale.
4. RESULTS AND DISCUSSION

The experimentation is carried out in a system with processor of 1.70 GHz, 4 GB RAM and windows 8.1 operating system to evaluate the proposed WOAK-NN. The experimentation is carried out with 50 whales. The experimentation details are shown in Table 1. Each whale represents the instances in the dataset. The proposed WOAK-NN is compared with K-NN, which is a lazy learning algorithm and the assignment of the class label is based on the voting of the neighbors which may reduce the accuracy of the classifier. Thus, to increase the accuracy of the classifier, Whale optimization algorithm is integrated with K-NN and the resultant WOAK-NN is compared with K-NN for various performance metrics and the results prove that the WOAK-NN achieves better performance than K-NN. Section 4.1 describes the dataset used for evaluation and section 4.2 describes the evaluation of metrics for measuring the performance of WOAK-NN.

4.1 Datasets

The Proposed WOAK-NN is simulated in python with the datasets taken from kaggle such as Delhi weather and Austin weather data to predict the weather. The first dataset, Delhi Weather (Dataset: https://www.kaggle.com/mahirkukreja/delhi-weather-data) contains 100990 instances, 19 independent attributes and 1 dependent attribute called condition. The dataset contains weather information collected from 1996 to 2017. Attribute details of Delhi Weather data is shown in Table 2. The second dataset, Austin Weather (Dataset: https://www.kaggle.com/grubenm/austin-weather) contains data gathered from 2013 to 2017. It contains 1319 instances, 20 independent attributes and 1 dependent attribute. Attribute details of Austin Weather data is shown in Table 3.

---

4. RESULTS AND DISCUSSION

The experimentation is carried out in a system with processor of 1.70 GHz, 4 GB RAM and windows 8.1 operating system to evaluate the proposed WOAK-NN. The experimentation is carried out with 50 whales. The experimentation details are shown in Table 1. Each whale represents the instances in the dataset. The proposed WOAK-NN is compared with K-NN, which is a lazy learning algorithm and the assignment of the class label is based on the voting of the neighbors which may reduce the accuracy of the classifier. Thus, to increase the accuracy of the classifier, Whale optimization algorithm is integrated with K-NN and the resultant WOAK-NN is compared with K-NN for various performance metrics and the results prove that the WOAK-NN achieves better performance than K-NN. Section 4.1 describes the dataset used for evaluation and section 4.2 describes the evaluation of metrics for measuring the performance of WOAK-NN.

4.1 Datasets

The Proposed WOAK-NN is simulated in python with the datasets taken from kaggle such as Delhi weather and Austin weather data to predict the weather. The first dataset, Delhi Weather (Dataset: https://www.kaggle.com/mahirkukreja/delhi-weather-data) contains 100990 instances, 19 independent attributes and 1 dependent attribute called condition. The dataset contains weather information collected from 1996 to 2017. Attribute details of Delhi Weather data is shown in Table 2. The second dataset, Austin Weather (Dataset: https://www.kaggle.com/grubenm/austin-weather) contains data gathered from 2013 to 2017. It contains 1319 instances, 20 independent attributes and 1 dependent attribute. Attribute details of Austin Weather data is shown in Table 3.
4.2 Performance of Proposed WOAK-NN

The performance of proposed WOAK-NN is compared with K-NN for various performance metrics such as accuracy, F-Measure, Mean Absolute error, Time complexity. Figure 4 and 5 shows the comparison of accuracy, F-measure and mean absolute error for the Delhi weather data set and Austin dataset respectively. Accuracy of a classifier represents number of correct classifications made by the classifier. A good classifier should have maximum accuracy. Accuracy is computed using equation (11).

\[
\text{Accuracy} = \frac{\text{NI}}{\text{Total}}
\]  
(11)
Algorithm 1:

Input: A dataset $WOAK - NN(D, NI)$ which consists of number of instances of weather data. $D$ represents new instance data for which weather has to be predicted, maximum number of iterations $NI$

Output: Class assigned for the new instance $MAX\_ITER$

Initialize population of whales $NI$ where $W ← \{W_1, W_2, ..., W_N\}$ represents number of whales.

For each $N$

Compute Fitness $W_i ∈ W$

End For

$F_{W_i} ← K - NN(W_i, NI)$

While $W_i ← \{W_i | \text{Min} (F_{W_i}, i = 1, N)\}$

For each ($t < MAX\_ITER$)

Calculate $W_i^t$ using Equation (5)

Calculate $a$ using Equation (3)

Calculate $A$ using Equation (4)

Generate $C$

Generate $l ← \text{rand}(-1,1)$

If $p ← \text{rand}(0,1)$

If $p < 0.5$

Calculate position of $W_i^t$ using Equation (2)

Else

Select random whale $W_i^t$

Calculate position of $W_{rand}^t$ using Equation (10)

End If

Else

continued on next page
Where $K$ represents true positives i.e. positive instances that are identified as positive. $I_{k} \in W_{i}$ represents true negatives i.e. negative instances that are identified as negative. $\text{dist}(I_{k}, NI)$ represents false positives i.e. positive instances that are misclassified as negative. $\text{dist}^{*} \leftarrow \sum_{i=1}^{K} \text{dist}(I_{i}, NI)$ represents false negative where negative instances that are misclassified as positive. WOAK-NN
achieves high accuracy than K-NN for both the datasets as shown in figure 3 and 4. The reason behind is that at each iteration WOAK-NN chooses the best instance which is very similar with the new instance. When the fitness value gets converged, the class label of the best instance is assigned to the new test instance.

F-measure is harmonic mean taken using precision and recall. It lies in the interval \([0,1]\). It provides the information of how precise the classifier is. It is computed using equation (12). From the figure 4 and 5, it is evident that WOAK-NN achieves high precision which proofs that the classifier is having high accuracy. WOAK-NN does iterating through the instances and find the instance which is similar to the new instance thereby assigning a class label. Thus, the classifier precisely does the classification thereby maximizing accuracy too.

\[
dist * \tag{12}
\]

Precision is defined as the ratio of number of correct classifications to the total number of positive predictions made by the classifier and is computed using equation (13)

\[
\text{Accuracy} \leftarrow \frac{tp + tn}{tp + fp + tn + fn} \tag{13}
\]
Figure 3. Working of WOAK-NN

Table 3. Attributes of Austin weather data set

| S.No | Attribute                                      |
|------|-----------------------------------------------|
| 1    | Date                                          |
| 2    | High Temperature in Fahrenheit                |
| 3    | Average Temperature in Fahrenheit             |
| 4    | Low Temperature in Fahrenheit                 |
| 5    | High Dew point in degrees Fahrenheit          |
| 6    | Average Dew point in degrees Fahrenheit       |
| 7    | Low Dew point in degrees Fahrenheit           |
| 8    | High Humidity as a percentage                 |
| 9    | Average humidity as a percentage              |
| 10   | Low humidity as a percentage                  |
| 11   | High Sea level pressure in inches of mercury  |
| 12   | Average sea level pressure in inches of mercury|
| 13   | Low sea level pressure in inches of mercury   |
| 14   | High visibility in miles                      |
| 15   | Average visibility in miles                   |
| 16   | Low visibility in miles                       |
| 17   | High wind speed in miles per hour             |
| 18   | Average wind speed in miles per hour          |
| 19   | Highest wind speed gust in miles per hour     |
| 20   | Total precipitation in inches                 |
| 21   | Weather Event                                 |
Recall is defined as the ratio of number of positive instances that are classified as positive to the sum of positive instances that are identified as positive and negative instances that are classified as positive which is shown in equation (14).

\[ tp \]

Mean absolute error  \( tn \) is the average error made by the classifier. Error is measured as difference between predicted value and actual value. Equation (14) shows the calculation of mean absolute error. A good classifier should have minimum mean absolute error.

\[ fp \]

Where \( fn \) represents the actual value and \( F - measure \leftarrow 2 * \frac{Precision \cdot Recall}{Precision + Recall} \) represents the predicted value. From figure 4 and 5, it is evident that WOAK-NN achieves minimum error than K-NN, as the proposed method uses heuristic approach to find the instance which is similar to itself.

**Table 4. Delhi weather dataset**

| Metrics           | WOAK-NN | K-NN     |
|-------------------|---------|----------|
| Accuracy          | 0.9843  | 0.829670 |
| F-Measure         | 0.9604  | 0.867    |
| Mean Absolute Error | 0.0323 | 0.083    |

**Table 5. Austin weather dataset**

| Metrics           | WOAK-NN | K-NN     |
|-------------------|---------|----------|
| Accuracy          | 0.9541  | 0.8241   |
| F-Measure         | 0.932   | 0.81     |
| Mean Absolute Error | 0.0421 | 0.0932   |

The K-NN is compared with other machine learning algorithms such as Naïve Bayes and Decision tree and the results are represented in Table 6. From table 6, it is evident that K-NN achieves maximum accuracy, F measure and minimum mean absolute error than other algorithms for both the datasets. Thus K-NN is chosen for predicting the weather. Also the performance of WOAK-NN is compared with other metaheuristic algorithm called Sine Cosine Algorithm based K-NN (SCAK-NN) (Moorthy RS, Pabitha P 2020) and the results are represented in Table 7. From table 7, it is evident that WOAK-NN achieves the best results than SCAK-NN.

**4.2.1 Comparison of Time Complexity**

Time complexity is defined as the amount of time taken to execute an algorithm. The time complexity of K-NN depends on number of instances and features whereas the time complexity of WOAK-NN
Table 6. Comparison of performance of K-NN with Naïve Bayes and J-48

| Dataset          | Metrics      | K-NN       | Naïve Bayes | J-48       |
|------------------|--------------|------------|-------------|------------|
| Delhi Weather Dataset | Accuracy     | 0.829670   | 0.7843      | 0.7612     |
|                  | F-Measure    | 0.867      | 0.814       | 0.875      |
|                  | Mean Absolute Error | 0.083      | 0.143       | 0.168      |
| Austin Weather Dataset | Accuracy     | 0.8241     | 0.8056      | 0.7954     |
|                  | F-Measure    | 0.81       | 0.791       | 0.773      |
|                  | Mean Absolute Error | 0.0932     | 0.189       | 0.173      |

Table 7. Comparison of WOAK-NN with SCAK-NN

| Dataset          | Metrics      | WOAK-NN   | SCAK-NN     |
|------------------|--------------|-----------|-------------|
| Delhi Weather Dataset | Accuracy     | 0.9843    | 0.9731      |
|                  | F-Measure    | 0.9604    | 0.9504      |
|                  | Mean Absolute Error | 0.0323    | 0.0512      |
| Austin Weather Dataset | Accuracy     | 0.9541    | 0.9437      |
|                  | F-Measure    | 0.932     | 0.901       |
|                  | Mean Absolute Error | 0.0421    | 0.0632      |

Figure 4. Comparison of performance metrics for Delhi weather dataset

```
for t=1:NP
    for p=1:Dim
        if rand <= k_r (knowledge ratio)
            if f(x_t) > f(x_r)
                x_{tp}^{new} = (x_t + k_f * ((x_{t-1} - x_{t+1}) + (x_r - x_t)))
            else
                x_{tp}^{new} = (x_t + k_f * ((x_{t-1} - x_{t+1}) + (x_t - x_r)))
            end
        else
            x_{tp}^{new} = x_{tp}^{old}
        end
    end
end
```
depends on the number of whales, number of iterations required for convergence and number of features. Time complexity of various datasets for WOAK-NN and K-NN is represented in table 8. Equation 16 and 17 represents the time complexity of K-NN and WOAK-NN respectively. The time complexity of WOAK-NN for Delhi Weather dataset is lower than K-NN, as the K-NN computes Euclidean distance for all the instances in the dataset with the new instance and finds the K-neighbors. From the voting of the neighbors, the class label is assigned to the new instance. Whereas WOAK-NN finds the neighbors based on the fitness value and using the best instance, at each iteration the position of the instance gets updated till the convergence of fitness. When the fitness converges, the class label of the best instance is assigned to the new instance. Also from the figure 6, it is evident that when the number of instances is large, time complexity of K-NN is larger than WOAK-NN. Thus the time complexity of Delhi weather dataset is larger for K-NN than WOAK-NN. For Austin weather data, the number of instances is lesser and thus time complexity of K-NN is smaller than WOAK-NN.

\[
\text{Precision} \leftarrow \frac{tp}{tp + fp} \quad (16)
\]

\[
\text{Recall} \leftarrow \frac{tp}{tp + fn} \quad (17)
\]

Where \( MAE \) represents number of instances in the dataset, \( MAE \leftarrow \frac{\sum_{i=1}^{n} \text{abs}(y_i - y_i')}{n} \) represents number of features for each instance in the dataset.
4.2.2 Rate of Convergence

The rate of convergence of WOAK-NN for Delhi weather data and Austin weather data is represented in Figure 7. For Delhi weather data, the convergence happens at iteration 71 and the best fitness value is 71.15588. WOAK-NN starts with the highest fitness value of 123.7893 and at iteration 18 the algorithm does a level of exploration and gradually switching between exploration and exploitation which converges at iteration 71. Similarly for Austin weather data, the convergence happens at iteration 60 and the best fitness value is 55.14923. The highest fitness of 99.752 is gradually decreased by switching between exploration and exploitation. From iteration 1 to iteration 7, the algorithm does exploitation i.e. searching solution space locally and at iteration 8, WOAK-NN does exploration i.e. the algorithm makes search for global optimal solution.

Table 8. Comparison of time complexity

| Datasets         | WOAK-NN (ms) | K-NN (ms) |
|------------------|--------------|-----------|
| Delhi Weather    | 95000        | 1918810   |
| Austin Weather   | 100000       | 26380     |

Figure 6. Comparison of time complexity

4.2.2 Rate of Convergence

The rate of convergence of WOAK-NN for Delhi weather data and Austin weather data is represented in Figure 7. For Delhi weather data, the convergence happens at iteration 71 and the best fitness value is 71.15588. WOAK-NN starts with the highest fitness value of 123.7893 and at iteration 18 the algorithm does a level of exploration and gradually switching between exploration and exploitation which converges at iteration 71. Similarly for Austin weather data, the convergence happens at iteration 60 and the best fitness value is 55.14923. The highest fitness of 99.752 is gradually decreased by switching between exploration and exploitation. From iteration 1 to iteration 7, the algorithm does exploitation i.e. searching solution space locally and at iteration 8, WOAK-NN does exploration i.e. the algorithm makes search for global optimal solution.
5. CONCLUSION AND FUTURE WORK

In this work, we have addressed the problem of weather prediction and measured the efficiency using various parameters such as accuracy, F-measure and Mean absolute error. The Weather data is taken as input by WOAK-NN. When compared to K-NN, WOAK-NN gives optimal solution as it integrates heuristic approach with lazy learning. The proposed Weather prediction mechanism achieves maximum accuracy, F- measure and minimum mean absolute error.

In Future, the proposed WOAK-NN will be used to test weather data gathered from IOT Sensors and also aims to select optimal subset of features from large set of features.
REFERENCES

Abhishek, K., & Singh, M. P. (2012). 2nd International Conference on Computer, Communication, Control and Information Technology (C3IT-2012). Procedia Technology, 4, 311-318.

Bhatkande, S. S., & Hubballi, R. G. (2016). Weather Prediction Based on Decision Tree Algorithm Using Data Mining Techniques. International Journal of Advanced Research in Computer and Communication Engineering, 5(5), 483–487.

Chauhan, D., & Thakur, J. (2014). Data mining techniques for weather prediction: A review. International Journal on Recent and Innovation Trends in Computing and Communication, 2(8), 2184–2189.

Dataset. (n.d.a). https://www.kaggle.com/grubenm/austin-weather

Dataset. (n.d.b). https://www.kaggle.com/mahirkukreja/delhi-weather-data

Devi, C. J., Reddy, B. S. P., Kumar, K. V., Reddy, B. M., & Nayak, N. R. (2012). ANN approach for weather prediction using back propagation. International Journal of Engineering Trends and Technology, 3(1), 19–23.

Dhore, A., Byakude, A., Sonar, B., & Waste, M. (2017). Weather prediction using the data mining techniques. Int. Res. J. Eng. Technol., 4(5), 2562–2565.

Guendouz, M., Amine, A., & Hamou, R. M. (2018). Penguins Search Optimization Algorithm for Community Detection in Complex Networks. International Journal of Applied Metaheuristic Computing, 9(1), 1–14. doi:10.4018/IJAMC.2018010101

Holmstrom, M., Liu, D., & Vo, C. (2016). Machine learning applied to weather forecasting. Academic Press.

Kashiwao, T., Nakayama, K., Ando, S., Ikeda, K., Lee, M., & Bahadori, A. (2017). A neural network-based local rainfall prediction system using meteorological data on the Internet: A case study using data from the Japan Meteorological Agency. Applied Soft Computing, 56, 317–330. doi:10.1016/j.asoc.2017.03.015

Kaur, S., & Cheema, S. S. (2017). Big data and analysis of weather forecasting system. International Journal of Advanced Research in Computer Science, 8(7), 176–179. doi:10.26483/ijarcs.v8i7.4149

Khajure, S., & Mohod, S. W. (2016). Future weather forecasting using soft computing techniques. Procedia Computer Science, 78, 402–407. doi:10.1016/j.procs.2016.02.081

Kohail, S. N., & El-Halees, A. M. (2011). Implementation of data mining techniques for meteorological data analysis. Implementation of Data Mining Techniques for Meteorological Data Analysis, 1(3).

Krishna, G. V. (2015). An integrated approach for weather forecasting based on data mining and forecasting analysis. International Journal of Computer Applications, 120(11).

Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. Advances in Engineering Software, 95, 51–67. doi:10.1016/j.advengsoft.2016.01.008

Moorthy, R. S., & Pabitha, P. (2018, August). A Study on Meta Heuristic Algorithms for Feature Selection. In International Conference on Intelligent Data Communication Technologies and Internet of Things (pp. 1291-1298). Springer.

Moorthy, R. S., & Pabitha, P. (2019). Optimal provisioning and scheduling of analytics as a service in cloud computing. Transactions on Emerging Telecommunications Technologies, 30(9).

Olaya, F., & Adeyemo, A. B. (2012). Application of data mining techniques in weather prediction and climate change studies. International Journal of Information Engineering and Electronic Business, 4(1), 51–59. doi:10.5815/ijieeb.2012.01.07

Parameswaran, P., & Moorthy, R. S. (2019). Secure Pervasive Healthcare System and Diabetes Prediction Using Heuristic Algorithm. Intelligent Pervasive Computing Systems for Smarter Healthcare. doi:10.1002/9781119439004.ch8

Petre, E. G. (2009). A decision tree for weather prediction. Academic Press.
Rajalakshmi Shenbaga Moorthy received her Bachelor of Technology under the stream of Information Technology from Mookambigai College of Engineering in 2010. She completed her Master of Engineering in the stream of Computer Science and Engineering from Madras Institute of Technology, Anna University in 2013. She is the Gold Medalist in her B.Tech and M.E programme. She is currently doing Ph.D in Information and Communication Engineering, and associated with St. Joseph’s Institute of Technology, Chennai as an Assistant Professor. Her research area includes Analysis of Algorithms, Machine learning, Resource Management Techniques. She has published various research articles indexed in IEEE and Elsevier.

P. Pabitha received her Bachelor of Engineering from University of Madras and Masters in Engineering from Anna University in Computer Science and Engineering. She received her Doctor of Philosophy from Anna University, Chennai. She is presently working as an Assistant professor in the Department of Computer Technology at Madras Institute of Technology Campus, Anna University, Chennai. She has nearly twenty years of Academic and Industry experience. She is an active researcher in the field of Computer Science especially Web technologies, Machine Intelligence, and Software Engineering. She has several research publications in various International Journals and Conferences which are indexed in IEEE.