A Machine Learning Approach to Weather Prediction in Wireless Sensor Networks

Mrs Suvarna S Patil[1]

Assistant Professor
Department of E&CE
RYMEC, Ballari

Dr B.M.Vidyavathi[2]

Professor and Head
Department of Artificial Intelligence and Machine Learning
BITM, Ballari, India

Abstract—**Weather prediction is the key requirement to save many lives from environmental disasters like landslides, earthquake, flood, forest fire, tsunami etc. Disaster monitoring and issuing forewarning to people, living in disaster-prone places, can help protect lives. In this paper, the Multiple Linear Regression (MLR) model is proposed for humidity prediction. After exploratory data analysis and outlier treatment, Multiple Linear Regression technique was applied to predict humidity. Intel lab dataset, collected by deploying 54 sensors, to form a wireless sensor network, an advanced networking technology that existed in the frontier of computer networks, is used for solution build. Inputs to the model are various meteorological variables, for predicting weather precisely. The model is evaluated using metrics - Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE). From experimentation, the applied method generated results with a minimum error of 11%, hence the model is statistically significant and predictions more reliable than other methods.**

Keywords—Data mining; wireless sensor network; multiple linear regression; outliers treatment; r-square; adjusted r-square

I. INTRODUCTION

Earlier information processing was done using general purpose devices like Mainframes, laptops, palmtops etc. In many applications these computational devices are used to process human centered information. But in some applications, controlling and monitoring action is required by focusing on physical environment. For example, in a chemical factory, processes can be controlled for exact temperature. Here controlling operation is embedded with computation without human intervention. Due to the technological advancement, another important aspect needed along with computation and control is communication. This processed information needs to be transferred to the place where it is necessary, a user or an actuator. Wired communication is expensive compared to wireless communication; even wires restrict devices from moving and prevent sensors and actuators being close to the event under observation. Hence, an implementation of a new network called wireless sensor network appeared. Sensor networks are built with a number of sensors, having sensing, processing and communicating capabilities, used in real time data analysis and monitoring applications like habitat monitoring, healthcare applications, environmental monitoring and tracking of objects to mention a few. Nodes are not costly but have memory, processing and energy limitations. An example sensor network is as shown in Fig. 1.

Recently Sensor Networks (WSN) have transformed largely due to the advancement in wireless communications, Micro Electro Mechanical Systems (MEMS), distributed processing and embedded systems. These networks are widely used in various areas such as agriculture monitoring, monitoring of habitat and surveillance [1]. The most crucial system of real time monitoring and controlling is environment. Nodes in WSN are small in size and consist of microcontroller, transceiver and memory, capable of short range communication. These battery operated nodes measure temperature, humidity, light and voltage of natural event from the place of deployment and send to the sink node. Sink nodes with enough processing capability compared to end nodes perform required pre-processing on the received raw data of sensors and forward to the base station. Base stations with embedded controlling and monitoring functionalities further process the data collected from sink node for knowledge extraction and decision making to ultimate user.

Weather forecasting is a major challenge in the meteorology department due to recurrent climatic changes [2]. There are very few solutions in generating weather reports with several limitations [3]. Many outdoor activities are affected due to wind chill, rainfall and snow, the results of frequent changes in weather [4]. Inaccurate weather reports will put someone into a dangerous state [2] if the climatic conditions are not safe. There are many existing data mining techniques for processing and evaluating huge amount of weather dataset. To predict weather, data mining process has three stages–data pre-processing, Model training and then prediction.
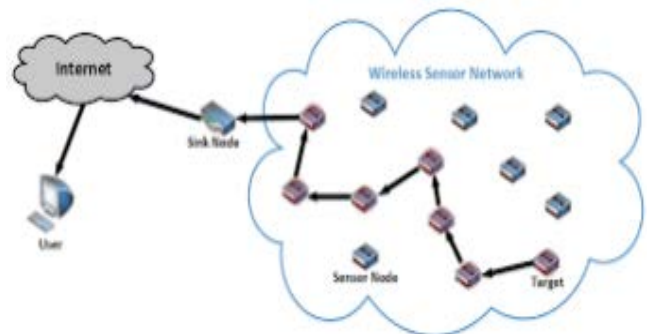

Fig. 1. An Example Wireless Sensor Network.

II. Related Work

An important section of research work is literature survey, facilitating researcher to gain knowledge in the relevant field and identifying challenges. This section presents the identified best works of considered domain.

N. Krishnaveni and A. Padma [5] introduced a decision tree based algorithm called SPRINT, which builds and constructs a decision tree with relevant data. Here an enhancement in observation is made using an open source historical dataset collected from weka tool (https://storm.cis.fordham.edu/~gweiss/datamining/datasets.html), a tool which permits direct mining of SQL databases. Using weka on the weather parameters of considered dataset, like temperature, outlook, and humidity and windy, weather is predicted as sunny, rainy or overcast. Results proved that SPRINT is more efficient and precise compared to the existing Navie Bayes algorithm proving the performance of the work.

Munmun et al. [2] proposed an integrated method for predicting weather in order to analyze and measure environmental data. Classification is done using Naive Bayes and Chi-square algorithms. A web application states weather information taking inputs as temperature, current outlook, wind and humidity condition. Accordingly implemented system is capable of predicting weather.

Taksande et al. [6] presented forecasting of weather by Frequent Pattern Growth Algorithm. Predicting rainfall is the major goal of implementation. Dataset was collected by Nagpur station from Jan 2010–Jan 2014 and computed using Frequent Pattern Growth Algorithm. Defined variables used for predicting rain are temperature, humidity and wind speed. The implemented model worked on these parameters and provided 90% of attainment. Wang et al. [7] implemented data mining method using cloud computing in order to predict weather. Decision tree and Artificial Neural Network Algorithms are applied on meteorological data gathered at appropriate time and place. This method worked effectively on averaged weather parameters and resulted in better classification.

Sushmitha Kothapalli et al. [3] presented Auto-Regressive Integrated Moving Average (ARIMA) model for analysing and forecasting real time data. The dataset contained humidity, temperature, wind and rainfall as variables. This gathered data was stored as CSV, JSON, and XML formats in the cloud. In this work, the author was able to achieve efficient results with correlation analysis on values followed by ARIMA model.

Salman et al [8] developed a deep learning method for predicting weather. The idea was to explore internal hierarchical pattern of the dataset. For experimenting, BMKG (Indonesian Agency for Meteorology, Climatology, and Geophysics) data was considered. Recurrence Neural Network (RNN), Conditional Restricted Boltzmann Machine (CRBM) and Convolutional Network (CN) models were used on the dataset. Prediction results of these models helped the agriculture and tourism sectors.

Maqsood et al [9] introduced a group of neural networks to predict weather. Here, the groups of artificial neural networks (ANNs) were compared considering relative humidity, wind speed and temperature as the key parameters. The predictive models used for experimenting were radial basis function network (RBFN), Elman recurrent neural network (ERNN), Hopfeld model (HFM), multi-layered perceptron network (MLPN) and regression techniques. Comparative analysis showed RBFN model is better while HFM gave lesser accuracy.

Almgren et al [4] utilized Hadoop distributed system for climatic prediction. This work showed that, prior prediction diminishes event planning disasters. Here data is stored in HDFS and then processed by MapReduce programming. Outdoor events can then be planned, by obtaining processed results about weather, location and time. Oury and Singh et al. [10] created Hadoop technique for weather data analysis. Climatic conditions were investigated using precipitation, snowfall and temperature as evaluation parameters. Utilizing Apache PIG and Hadoop map reduction executed dataset. Python language was used to present output in visual form.

Manogaran and Lopez et al. [11] introduced spatial cumulative sum algorithm to detect climatic changes. MapReduce technique was applied on weather data stored in Hadoop Distributed File System (HDFS). Climatic changes were detected by applying spatial autocorrelation. Mahmood et al. [12] produced a data mining technique for predicting weather. This paper presents a data mining technique called Naïve Bayes algorithm.

III. Proposed Work

The newly-created model considers meteorological data to predict values for humidity by technically analysing the data and then applying multiple linear regression algorithms. In the previous work of data cleansing and pre-processing step, it was found that variables - humidity, light and voltage had a few missing values, but since the percentage of missing values is very negligible (<2%), can omit them. In the initial phase of technical analysis, data pre-processing is carried-out to gain insights into underlying source data. Exploratory analysis was carried out on pre-processed data to understand underlying relationships between dependent and independent variables. In the next phase multiple linear regression algorithms is applied on the processed data, considering key attributes as voltage, light, temperature and humidity. The model is built, trained and validated by dividing the data into Train and Test sets. For experimenting, there are many ways to partition data. But the most approved one is partitioning data into Train/Test sets or cross validation. The first set called Train data is used for model fitting and the second set called Test data to test trained model. This was followed by Model build, and, later test data was used to score the Model and obtain predicted value, which is then validated.

The master dataset needs to be divided into training and testing data - 70 percent is trained and 30 percent is testing data. Previous works presented exploratory analysis on pre-processed data to understand underlying relationships between dependent and independent values. This was followed by Model build, and, later testing data was used to score the Model and obtain predicted value, which is then validated. In the initial phase of data [13] analysis having 0.2 million samples, outliers treatment is carried out to handle extreme values.
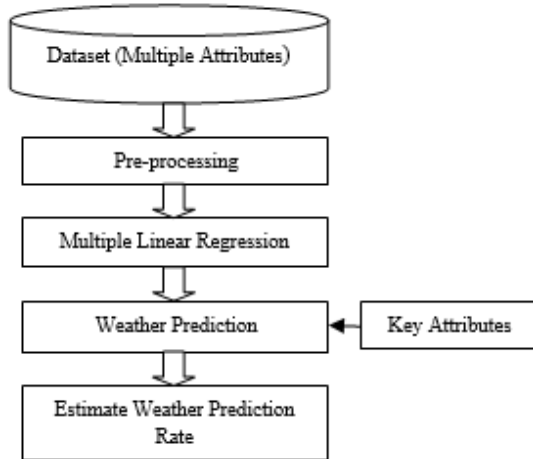
The proposed method is shown in Fig. 2.

Fig. 2.   Proposed Method Block Diagram.

A. Outliers Treatment

A measurement of variability called interquartile range (IQR) can be obtained by dividing data into quartiles. Depending on division the values are named as first, second, and third quartiles; denoted with Q1, Q2, and Q3, respectively. In the initial set of data, Q1 is the "middle" value given by equation 1.

$$Q1 = \left(\frac{n+1}{4}\right)^{th} term \qquad (1)$$

Median is Q2. Middle value is Q3 given by equation 2.

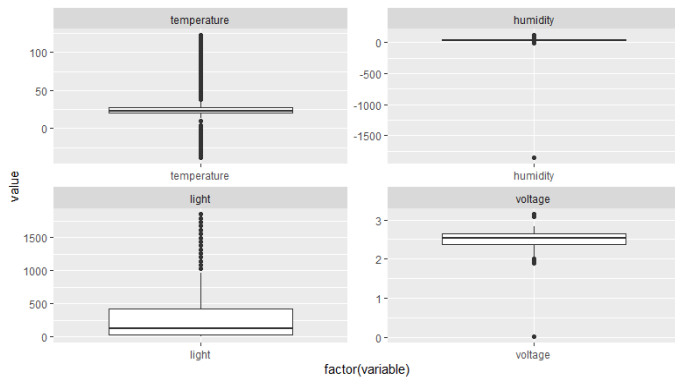$$Q3 = \left(\frac{3(n+1)}{4}\right)^{th} term \qquad (2)$$

Fig. 3.   Outliers.

The outliers in the master data are depicted in Fig. 3.

$$IQR = Q3 - Q1 \qquad (3)$$

$$\text{Low outliers: } Q1 - 1.5IQR \qquad (4)$$

$$\text{High outliers: } Q3 + 1.5IQR \qquad (5)$$

Outliers in the data-set are replaced by their corresponding nearest quantile values. Data values outside the upper-boundary (high outliers) are replaced with corresponding third quantile values, similarly, data values outside lower-boundary (low outliers) are replaced with corresponding first quantile values. Outlier treatment is carried-out for all key variables. Outlier detection is an essential step in data analysis since the un-treated outliers can affect the model results and predictions. Generally, outliers can be treated, suppressed or amplified. Our approach is to treat outliers as detailed above. Fig. 3 shows the outlined values for the four attributes which are in black colour.

The next step is weather prediction using multiple linear regression technique.

Multiple linear regression formula is:

$$y = \beta_0 + \beta_1 X_1 + + \beta_n X_n + \varepsilon \qquad (6)$$

In equation 6, relying parameter predicted value is y, β_0 defines y-intersection (y-value with other variables made 0). A predicted 'y' value change with an increase in independent variable is given by $\beta 1\ X_1$ or the first independent variable (X_1) with regression coefficient (β_1). This step of predicting y is repeated for all remaining independent variables to be tested. Finally the regression coefficient of the last independent variable is $\beta_n X_n$. Error present in the model is denoted by ε.

Important parameters required for identifying a best-fit line to each independent value in multiple linear regressions are:

- Coefficients resulting least error.

- t-statistic of the model. Z.

- The p-value.

t-statistic and p-value for each regression coefficient in the model is calculated and compared to determine statistical significance of the variable on outcome plus the magnitude of effect on outcome variable(y).

Regression analysis involves identification of residual data characteristics by means of assumption tests before Model build. Assumption tests are explained in the following subsection followed by model build. Regression analysis is essentially a parametric method and hence, validating assumptions is important. If underlying assumptions are violated Model results will not be accurate and predictions will be more prone to errors.

B. Regression Analysis Assumption Test

This section depicts assumption tests, which should be validated before regression analysis.

1) Linearity: Ideally, no fitted patterns are shown in the residual plot. It means, the red straight line shown in Fig. 4, should be approximately horizontal at zero. The model is found linear with the existence of a pattern, or a possible non-linear relationship. Here, there is no definitive pattern, and a linear relationship between the independent and dependent values seen, hence linear model suits.

2) Normality of residuals: For normality check, residuals can be visualized using QQ plot. As per normality assumption the residuals plot should be a straight line. In the given data, all the observations follow the defined reference straight line as shown in Fig. 5; hence normality assumption can be made.
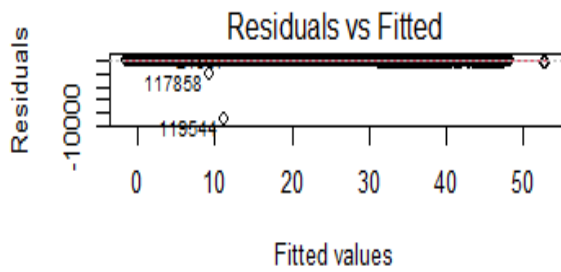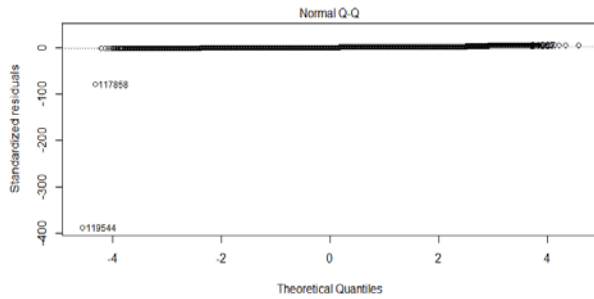
Fig. 4.   Linearity.

Fig. 5.   QQ Plots.

3) Homoscedasticity: Fig. 6 shows how residuals are evenly spread along the range of predictors. The red line in the plot is nearly horizontal with similar dense distribution of points on either side. We can assume homogeneity of variance.
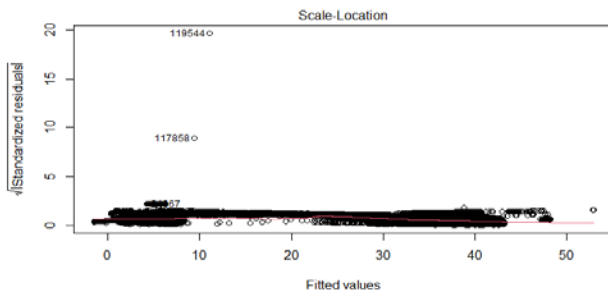

Fig. 6.   Homoscedasticity.

4) Durbin-Watson (DW) test: DW test examines whether the error terms are autocorrelated. Null hypothesis states that no autocorrelation exists. The statistical DW test was performed and based on the p-value, we conclude that no autocorrelation exists. Statistical DW test yields the test result as ~1.9, which means, no autocorrelation exists. Hence, this assumption is validated.

5) VIF for multicollinearity: On examining variable inflation factors for predictor variables, it was found that they do not exceed 5, hence, no multi-collinearity exists in the data set, which means, and the assumption of no multicollinearity is validated.

6) Residual v/s Leverage: The most influential 3 values are shown on the plot in Fig. 7; however, they can be exceptions, or, outliers. In this case, data does not present any high influence points.
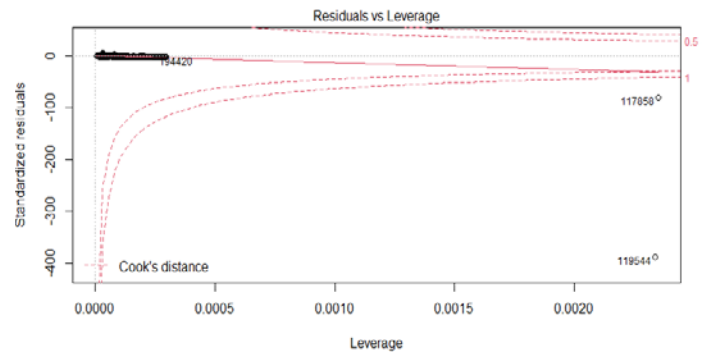

Fig. 7.   Residuals vs Leverage.

7) Cook's distance: This score considers the combined values of leverage and residual parameters to determine an influential value. Regression analysis results will change with the inclusion or exclusion of influential value. An influential value has a larger residual. In linear regression analysis all outliers (or extreme data points) are not significant. Residuals show nearly even spread along the range of predictors. The red line in the plot is nearly horizontal with similar dense distribution of points on either side. We can assume homogeneity of variance.
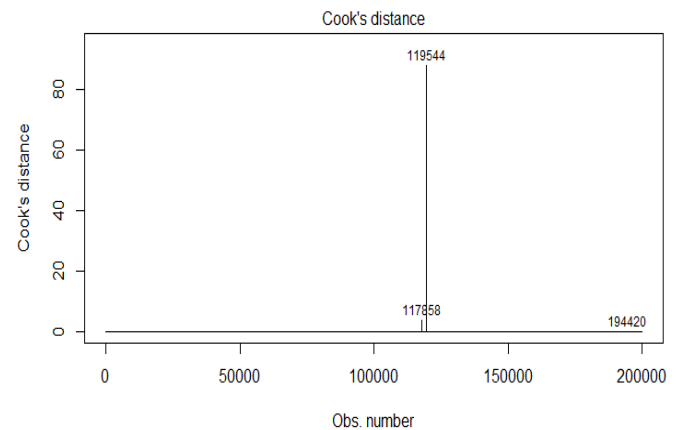

Fig. 8.   Cooks Distance.

Cook's distance aids in determining the influential value. Here the thumb rule is, observations will have larger influence, if Cook's distance is more than $4/(n - p - 1)$ [14]. In the expression n represents the count of observations and p- the number of predictor variables.

In the given data, Cook's distance is too small as depicted in Fig. 8 and does not have significant influence on regression analysis. However, outliers, if any, must be detected and suitably handled.

C. Multiple Linear Regression Model

MLR is a statistical approach for predicting the output of a dependent variable by considering multiple independent variables [15]. MLR is capable of building a linear relationship between predictor variables and response values. Data is collected from the Intel Research Labs. Initially to eliminate noisy values, data pre-processing is done, avoiding reduction in prediction accuracy. Now, pre-processed data must be divided

into training and test data. The proposed algorithm needs to be trained utilizing training data for establishing relationship with several parameters. The final model will predict outcome of any new given data set containing data for same independent variables.

1) Evaluation parameters: The Model built is evaluated using various statistical metrics as listed below.

a) R-square: In MLR models r-square is used to measure a goodness-of-fit. Purpose of using this statistic is to judge how independent variables can mutually explain the dependent variable variance in percentage.

$$R - Square = \frac{\text{Variance explained by full model}}{\text{Total Variance}} \quad (7)$$

R-square increases every time a new independent variable is added to the Model. While a higher R-square is desirable, one has to vary about over-inflating the results and overfitting the Model.

b) Adjusted R-square: In regression Models, this statistic is used for comparing the goodness-of-fit with independent variables. The number of terms in the Model is adjusted with adjusted r-square. Importantly this parameter is mainly used to check an improvement in the Model fit with a new term. The adjusted R-squared value will automatically decrease whenever the new term fails to enhance the model fit by an adequate amount.

In our situation, r-square and adjusted-r-square are above-average values, and acceptable.

c) Mean Absolute Error (MAE): While predicting a set of values MAE is used to measure errors average magnitude, independent of direction.

With all equal weighted individual differences, MEA is computed as the test samples average parameter, considering the absolute differences between actual and prediction observations.

$$MAE = \frac{1}{n} \sum_{j=1}^{n} |y_j - \hat{y}_j| \quad (8)$$

Where, n – Number of samples, y_j-Expected value and $\hat{y}_j$-Predicted value.

d) Root Mean Squared Error (RMSE): RMSE is a squared output used for measuring average score of error.

It's the square root of the average of squared differences between prediction and actual observation.

$$RMSE = \sqrt{\frac{1}{n} \sum_{j=1}^{n} (y_j - \hat{y}_j)^2} \quad (9)$$

Where, n – Number of samples, y_j-Expected value and $\hat{y}_j$-Predicted value.

MAE and RMSE values lie between 0 and ∞, and are independent of the direction of errors. Both the metrics can be used for error prediction, expressing the variables in the interest of values units. For efficient modeling lower values are preferred, since both MAE and RMSE yield negative results.

Usually squared values of errors are taken before averaging, since RMSE gives relatively more weight to larger errors. Hence, RMSE is most suitable for large number of undesirable errors. Though RMSE penalizes larger errors, MAE has better interpretability, hence is considered.

The size of the dataset was decided by extracting random samples from the master dataset which keeps the distribution at a defined significance level. Basically, in order to achieve higher statistical significance, a machine learning model must be trained for larger dataset. But to save time sub-samples are selected maintaining other statistics same. Data distribution is fairly normal.

IV. RESULTS

All data pre-processing, exploratory analysis, Model build, tune and validation were performed using R language. In order to predict humidity, data pre- processing followed by multiple linear regression method was used. Existing data mining methods worked on homogeneous data, but the presented model is capable of handling heterogeneous data. The model performance was evaluated by using statistical metrics like R^2, MAE, RMSE, etc.

1) Intel dataset: For experimentation, freely available Intel Lab dataset [16] was used. This dataset has nearly 2.3 million records collected by deploying 54 sensors in the Intel Berkeley Research lab. Sensors used were Mica2Dot, capable of collecting time-stamped weather information. The values are recorded by in-network query processing TinyDB system and these are recorded once every 31 seconds with humidity, temperature, light and voltage as key variables. The dataset format is given by: date, time, epoch, mote ID, temperature, humidity, light, and voltage. All the sensors are numbered with ids ranging from 1-54. Some sensor's values are missing or approximated. These measured variables are represented as, temperature in degrees Celsius, humidity ranging from 0-100% and its temperature corrected relative humidity. Light is recorded in Lux (1 Lux is equivalent to moonlight, 400 Lux corresponds to a bright office, and 100,000 Lux is equivalent to full sunlight). Voltage is in the range 2-3, measured in volts. Lithium ion cell batteries were used for providing constant voltage to sensors for their lifetime. It is observed that voltage variations are highly correlated with temperature.

2) Model Results and Evaluation Metrics

a) R-square: 0.692: Higher the R-square value, better it is. This statistical value shows variation between the dependent and independent variables. R-square is 0.692, which is considered a good-fit. Independent variables are able to explain a large amount of variance in dependent variable. p-value :<2.2e-16 => Model is statistically significant.

b) Metrics: As observed in Table I, average error is ~11%, hence we say Model is accurate upto 89%. MAE and RMSE are very small, hence our Model is very good; predictions generated from this Model will be very accurate.

c) Residuals: Table II gives residuals statistics for outliers treatment.

TABLE I.   STATISTICS OF RESIDUALS

MAE	MSE	RMSE	MAPE
0.113799	0.03070434	0.17522655	Inf.

TABLE II.   STATISTICS OF RESIDUALS FOR TREATING OUTLIERS

Min	1Q	Median	3Q	Max
-0.93682	-0.08049	0.00572	0..08174	0.97949

d) Coefficients: Table III lists the model coefficients for variables temperature, voltage, light and their correlations.

TABLE III.   MODEL COEFFICIENTS

Variables	Estimate	Std. Error	t Value	Pr(>|t|)
Intercept	0.936820	0.002421	386.889	< 2e-16 ***
Temperature	-0.919190	0.003173	-289.665	< 2e-16 ***
Voltage	-0.134744	0.003269	-41.224	< 2e-16 ***
Light	-0.080437	0.007696	-41.224	< 2e-16 ***
temperature:voltage	1.493132	0.013469	110.853	<2e-16 ***
temperature:light	0.140049	0.010163	13.780	<2e-16 ***
voltage:light	-0.008874	0.010172	-0.872	0.383
temperature:voltage:light	-1.228874	0.038373	-32.024	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1732 on 139994 degrees of freedom.

Multiple R-squared: 0.6919, Adjusted R-squared: 0.6918.

F-statistic: 4.49e+04 on 7 and 139994 DF, p-value:< 2.2e-16.

3) MLR Model with Interaction effects: From the Model statistics, it can be seen that most co-efficient are statistically significant when seen individually, as well as with interaction effects. The interaction term co-efficient are statistically significant, suggesting an implicit interaction relationship between predictor variables (other than voltage and light). Model equation can be represented as below:

humidity = 0.936 - 0.92*temperature - 0.13*Voltage - 0.08*light + 1.49*(temperature*voltage) + 0.14*(temperature*light)-1.22*(temperature*voltage*light)

V. CONCLUSION

Changes in weather affect lives of various living beings. Main idea of this paper is to analyze the data collected from WSNs of Intel Lab and make appropriate decision in order to convey right information at right time to help save lives.

Machine Learning techniques are used for weather prediction, and MLR algorithm is built using temperature, humidity, light and voltage as the key variables. We have evaluated the Model and model results are documented above. The resultant model can predict with high degree of accuracy and can be expanded for further work. Values of statistical parameters indicate that the proposed model is statistically more significant compared to other existing techniques.

REFERENCES

[1] Mohd Fauzi Othmana , Khairunnisa Shazalib, 2012,' Wireless Sensor Network Applications: A Study in Environment Monitoring System', International Symposium on Robotics and Intelligent Sensors 2012 (IR IS 2012) , Procedia Engineering 41 (2012) 1204 – 1210.

[2] Munmun B, Tanni D, Sayantanu B (2018), 'Weather forecast prediction: an integrated approach for analyzing and measuring weather data'. Int J Computer Appl. https://doi.org/10.5120/ijca2018918265.

[3] Kothapalli S, Totad SG (2017), 'A real-time weather forecasting and analysis'. In: IEEE international conference on power, control, signals and instrumentation engineering (ICPCSI-2017), pp 1567–1570.

[4] Almgren K, Alshahrani S, Lee J (2019),'Weather data analysis using hadoop to mitigate event planning disasters'. https://scholarworks.bridgeport.edu/xmlui/handle/123456789/1105 Accessed 1 Feb 2019.

[5] N. Krishnaveni, A. Padma, 2020, 'Weather forecast prediction and analysis using sprint algorithm', © Springer-Verlag GmbH Germany, part of Springer Nature 2020, Journal of Ambient Intelligence and Humanized Computing https://doi.org/10.1007/s12652-020-01928-w.

[6] Taksande AA, Mohod PS (2015), 'Applications of data mining in weather forecasting using frequent pattern growth algorithm'. Int J Sci Res (IJSR) 4(6):3048–3051.

[7] Wang Z, Mujib ABM (2017) , 'The weather forecast using data mining research based on cloud computing'. J Phys 1:1–6.

[8] Salman AG, Kanigoro B, Heryadi Y (2015), 'Weather forecasting using deep learning techniques'. In: 2015 International conference on advanced computer science and information systems (ICACSIS), pp 281-285.

[9] Maqsood I, Khan MR, Abraham A (2004), 'An ensemble of neural networks for weather forecasting'. Neural Comput Appl 13(2):112–122.

[10] Oury DTM, Singh A (2018), 'Data analysis of weather data using hadoop technology'. In: Smart computing and informatics, Springer, Singapore, pp 723–730.

[11] Manogaran G, Lopez D (2018), 'Spatial cumulative sum algorithm with big data analytics for climate change detection'. ComputElectrEng 65:207–221.

[12] Mahmood MR, Patra RK, Raja R, Sinha GR (2019), 'A novel approach for weather prediction using forecasting analysis and data mining techniques'. In: Saini H, Singh R, Kumar G, Rather G, Santhi K (eds) Innovations in electronics and communication engineering. Lecture notes in networks and systems, vol 65. Springer, Singapore.

[13] Madden, S. (2004). Wireless sensor data [Intel lab data]. Retrieved from http://db.lcs.mit.edu/labdata/labdata.html.

[14] Peter Bruce and Andrew Bruce, Practical Statistics for DataScientists, Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, Copyright © 2017.

[15] C. G. N. De Carvalho, D. G. Gomes, J. N. De Souza, and N. Agoulmine, "Multiple linear regression to improve prediction accuracy in WSN data reduction," in Network Operations and Management Symposium (LANOMS), 2011.

[16] Intel Lab Data. http://berkeley.intel-research.net/labdata/, 2004.