Internet of Plants Application for Smart Agriculture

Khurshid Aliev
Department of Management and Production Engineering
Politecnico Di Torino
Torino, Italy.

Eros Pasero
Department of Electronics and Telecommunication
Politecnico Di Torino
Torino, Italy.

Mohammad Moazzam Jawaid, Sanam Narejo
Department of Computer Systems Engineering
Mehran University
Jamshoro, Pakistan

Alim Pulatov
Head of EcoGIS centre
Tashkent Institute of Irrigation and Agricultural
Mechanization Engineers, Tashkent, Uzbekistan

***Abstract*—Nowadays, Internet of Things (IoT) is receiving a great attention due to its potential strength and ability to be integrated into any complex system. The IoT provides the acquired data from the environment to the Internet through the service providers. This further helps users to view the numerical or plotted data. In addition, it also allows objects which are located in long distances to be sensed and controlled remotely through embedded devices which are important in agriculture domain. Developing such a system for the IoT is a very complex task due to the diverse variety of devices, link layer technologies, and services. This paper proposes a practical approach to acquiring data of temperature, humidity and soil moisture of plants. In order to accomplish this, we developed a prototype device and an android application which acquires physical data and sends it to cloud. Moreover, in the subsequent part of current research work, we have focused towards a temperature forecasting application. Forecasting metrological parameters have a profound influence on crop growth, development and yields of agriculture. In response to this fact, an application is developed for 10 days ahead maximum and minimum temperatures forecasting using a type of recurrent neural network.**

Keywords*—Internet of Things; wireless sensor networks; smart agriculture; smartphone applications; artificial neural network; nonlinear autoregressive model; temperature forecasting

I. Introduction

Agriculture is a cultivation of products to feed the population. For centuries, it has remained as a key development factor for human civilization. Moreover, today the demand for efficient agriculture products is increasing [1]. In order to improve agriculture processes, we can acquire field data with sensors, make data analytics, perform analysis and take appropriate decisions and actions. Collecting big data from the field gives us a clearer understanding of product variability and quality of products [2].

In agriculture, physical parameters such as temperature, relative humidity and soil moisture are important [3]. There are several applications and well established measuring instruments to collect these data [4]. In addition, sensors to measure soil properties [5], detect and monitor foliar disease [1] or fertilizer management [6] already exist.

Data acquisition systems in agriculture need to cover large areas, collect representative samples and exchange measured information and control commands. In precision agriculture, one of the vital problems is how to distribute sensors and to establish reliable data communications. A robust and reliable data acquisition system enables synchronizing, exchange and storing of measured data. This kind of system is required to efficiently evaluate sensor signals and allow real-time or a-posteriori analysis of the behaviour of single parameters and their mutual impact [7].

Internet of Things (IoT) is receiving a great attention due to its potential strength and ability to be integrated into any complex system. It is becoming a great tool to acquire data from particular environment to the cloud. One of the use case fields of IoT is smart agriculture. However, there are some issues on developing low cost and power efficient WSN using advanced radio technologies for short and long-range applications. To satisfy the need for population, farmers and agriculture companies are turning to the Internet of Things (IoT). The IoT is pushing the future of farming to the next level. Smart agriculture is becoming more commonplace among farmers, and high tech farming is quickly becoming the standard thanks to the agricultural sensors.

WSN are playing important role in IoT technologies since it has been discovered. WSN systems are a strong and effective tool to distribute data among sensor nodes. The use of IoT applications and WSN (Wireless Sensor Networks) in the agriculture domain as proposed by other authors are discussed as follows. In [8], the authors have identified the issues related to reliability, autonomy, cost and accessibility to the application domain. The authors of [9] showed farm management system and its architecture is based on Internet of Things features. This architecture gives easy access to acquired data and advice. The authors of [10] have offered the use cases of Cloud Computing in the agriculture area. Moreover, they further describe it in the context of service providers and supply chain for cost-effective services for farmers. In [11], authors have illustrated the controlled architecture of smart agriculture based on IoT and Cloud Computing. Another issue arose with Wireless Sensor Networks (WSN) which must be connected to all sensor nodes in a smart way. The authors of [12] proposed a WSN for precision agriculture where a real-time data of the climatologically and other environmental properties are sensed and relayed to a central repository. Moreover, they proposed Wireless Mesh Network architecture which is convenient for agriculture applications [13]. The authors in [14] proposed an architecture Bayesian event prediction model which uses historical event data generated by the IoT cloud to predict future events. In [15], researchers reviewed the application of the data mining techniques for solving different challenges in event prediction system on time series. Despite the fact that there are many types of research, proposed architectures and WSN applications in agriculture domain, still, main problems rely on allocation sensors and thereby establishing reliable data communications in long-range areas. A robust and reliable data acquisition system enables synchronizing, exchange and storing of measured data. This kind of system is required to efficiently evaluate sensor signals and allow real-time or a posterior analysis of the behaviour of single parameters and their mutual impact [17].

This paper proposes a novel WSN approach and predictive model for smart agriculture sector which is different from above-mentioned technologies. Section 2 presents the detailed specifications followed and implemented to develop an IoT system and its Android application software. Our proposed IoT is capable to monitor ambient temperature, humidity and soil moisture. In Section 3, we introduce the use of an Artificial Neural Network (ANN) as time series forecaster to predict the temperature. Section 4 demonstrates the obtained results and evaluations for the chosen model. It also presents the discussion and summarized results. Section 5 concludes our work.

II. Research Methodology

In this section, we focused on the requirements and specifications of the system and described the developed hardware and software configurations of the Internet of Plants (IoP) system. In the foremost subsections, we specify the general architecture and components of our IoP based system. We further explain the basic flow of the designed system. Last two subsections are related to the server for storing data and developing PCB layout with Allegro ORCAD. The proposed system is based on monitoring temperature, humidity [16] and soil moisture of plants. We achieve this through developed IoP device and Android application, which provides the data, acquired from agricultural environment to the Internet through the Thingspeak platform. ThingSpeak [18] is an open source IoT platform. It is user-friendly and easy to extract data using API and HTTP protocol over the Internet or via a Local Area Network (LAN). The ThingSpeak IoT platform provides applications that let you analyze and visualize your data in MATLAB [19], and then act on the data. Sensor node data can be sent to ThingSpeak from Arduino [20], Raspberry Pi [21], BeagleBone Black [22], and other hardware.

A. General Architecture and Components of IoT System

The IoT is defined as a network of physical objects and devices embedded with Internet connectivity. It allows objects to be sensed and controlled remotely through embedded devices. The proposed prototype device gets information from plants and makes it available to the user. The design architecture of the IoP system is shown in Fig. 1. Next subsections will describe each stage of IoT based plant monitoring system.

B. Requirements and Specifications of the System

The IoP temperature, relative humidity and soil moisture acquisition system of plants consists of a device including a sensor and a microcontroller which wirelessly transmits plants data to the server using Wi-Fi protocol. DHT11 humidity and temperature sensors are selected for the system. 8-bit resolution digital output was used for reading the data. This sensor is ranked IP65, which means complete protection from dust and protection from water. It requires 2.5mA maximum current. The temperature ranges from 0 to 50 °C and response time from 6sec to 30sec, humidity ranges from 20% to 90% at 25 °C, response time 6 sec to 15sec.

To measure soil moisture, we have selected six Groovec soil moisture sensor. It is based on soil resistivity measurement, so with one device, it's possible to read the ground humidity in six different locations. Current consumption of the sensor is 35mA.

For data elaboration and communication management, the STM32L476RG microcontroller unit (MCU) of ST Microelectronics has been chosen. This component supports a Universal Synchronous/Asynchronous Receiver/Transmitter (USART) communication protocol required to communicate with the ESP-12E Wi-Fi module. It can drive various digital pins, that are required to communicate with the DHT11, drives the light-emitting diode (LED)s and performs other required functions. Maximum current usage of the MCU is about 150 mA. The main feature of this MCU is very low power consumption even in fully operative mode. This microcontroller enables users to serially upload the firmware through an embedded Joint Test Action Group (JTAG) serial wire debug port, which employs only two pins: PA13(the 46th pin) and PA14(the 49th pin).

For uploading the code it's possible to use an ST-Link device that can be connected to provide serial communication between the MCU and a personal computer (PC). In order to implement the connection between the device and Internet, we used ESP-12E Wi-Fi module. The WiFi module runs Lua language based scripts that are provided through USART communication by the MCU. WiFi module creates a client socket connection to the ThingSpeak internet protocol (IP) address through the predefined port and provides to a personal channel, which is created in the website database. In order to send data, we send an access request to the server. The thingspeak platform provides application programming interface (API) key which sends the package so that the data will be correctly processed. The maximum current consumption of the module is 250 mA.

We calibrated all sensors before prototyping. The calibration has been performed under standard working conditions. This means, for the DHT11, with a temperature range between (10 ÷ 35) °C and with a humidity range between (35 ÷ 65) %RH.

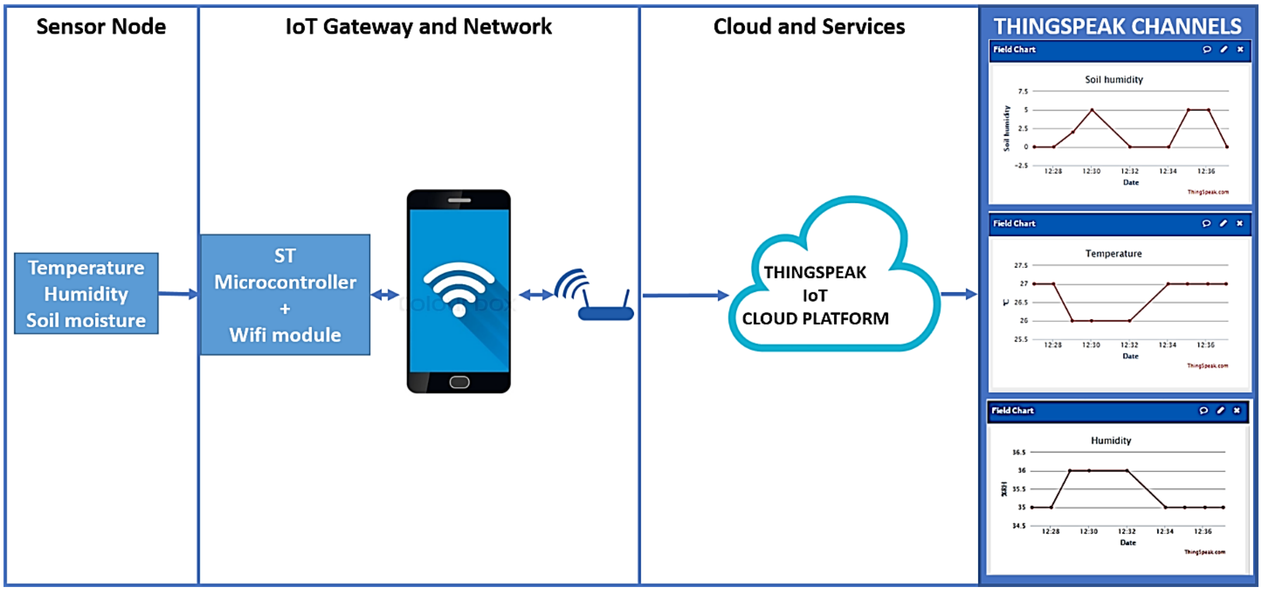

Fig. 1. IoT based plant monitoring system on the Thingspeak platform.

For the soil moisture sensor, another approach has been considered: First of all, the maximum reading value has been considered by measuring the value in water, and afterwards, the values in a dry soil that is gradually damping have been sensed. Considering the readability of the moisture data for the user, the result values have been divided into five ranges, that are represented by integer values as presented in Table I.

TABLE I. SOIL MOISTURE SENSOR CALIBRATION VALUES

Value	Data
0	Very dry soil
1	Dry soil
2	Humid soil
3	Very humid
4	Wet soil

C. The Design flow of the IOP

In order to have a working product, the design has been divided into three steps:

- First step: to create a working prototype sensor node using the NUCLEOL476RG developing board of ST Microelectronics and the NodeMCU WiFi board which is depicted in Fig. 2.
- Second step: to draw the schematics of design using the OrCAD Capture tool.
- Third step: to create the PCB layout using the OrCAD Allegro tool.

For developing prototype board we have been used STM MCU of the NUCLEOL476RG and ESP-12E component of the NodeMCU thus having an easier access to the offered features. Fig. 3 shows the final prototype of the IOP system.

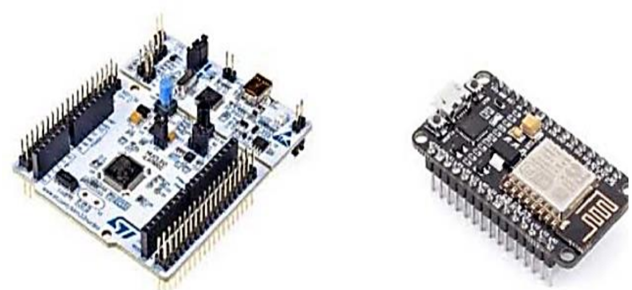

Fig. 2. The used boards NUCLEO-L476RG and NodeMCU.

D. WiFi Configurations

Whenever the device is booted, the WiFi connection needs configuration. Therefore, the first part of the firmware consists in a loop to correctly set the connection. For this phase, an Android application has been developed, so that the required information can be sent. The information required is:

- Service set-identifier (SSID) of the WiFi network in which the connection needs to be established.
- Password to connect to the network.
- API code for the ThingSpeak channel.
- Number of moisture sensors connected to the device.

Once the WiFi connection has been correctly set, the firmware enters into an infinite loop, which restarts every minute. Initially, data was collected from the sensors.

At this point, the values are correctly converted so that they can be transferred to the ESP module. Afterwards, the MCU through Lua scripts makes the ESP module open the socket connection. Data is thus sent to the ThingSpeak channel and plotted into each proper graph.

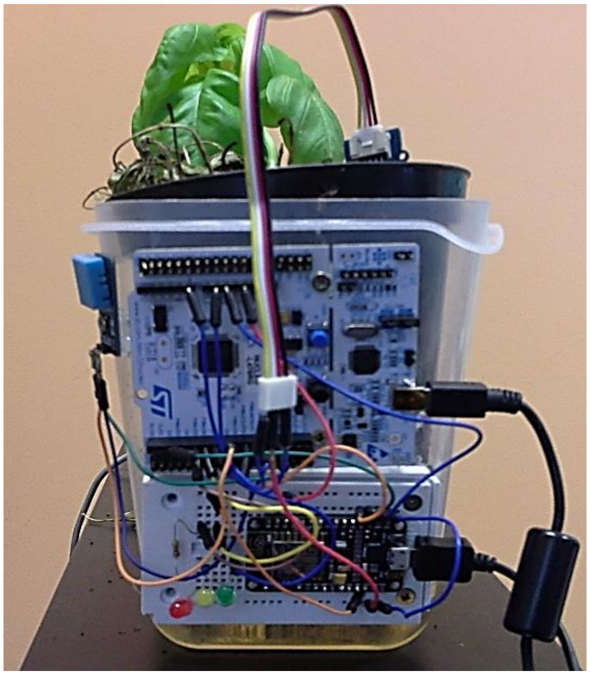

Fig. 3. Final development prototype of the IoP system.

E. IoP Android Application

Whenever the device is turned on or it's rebooted it is necessary to configure the WiFi connection to the access point. Solution before developing the application was to generate each time a telnet connection, with a computer or a general mobile application, by specifying the connection parameters and tokens. In order to make this procedure easier, we developed an application using Android Studio.

The application gives the possibility of configuring the connection by just entering the four required fields SSID, password, API code for the ThingSpeak channel, number of moisture sensors.

The Application is developed with a single activity that gives the possibility of entering the fields and by pressing the send button a Transmission Control Protocol (TCP). TCP/IP connection is established with the device and with a predefined timing the required data is sent. The user interface of the android application is shown in Fig. 4.

F. Web Service Settings

As mentioned earlier, Thingspeak is a free web service that gives the possibility to collect and store sensor data in the cloud for developing Internet of Things applications. The following steps are required to properly use this service.

Account registration

Channel settings:

Name: The name of the monitoring object

Field 1: Temperature

Field 2: Humidity

Field 3: Soil humidity

API generation: Each channel has an API key used to protect the user from undesired connections. When connecting to a channel, the API key specified to the App and the one specified on the portal must coincide. In Fig. 5, 6, 7 we have shown that how ThingSpeak plots data sent by the device and further collected by the sensors. Air temperature (Fig. 5) and humidity (Fig. 6) have been compared to the readings of a commercial thermo-hydrometer. Soil moisture sensor which is in Fig. 7 has been tested first in outside the soil with the absence of water, then, by inserting it into an almost dry soil; thus, by adding some water to the soil and finally into the pure water.

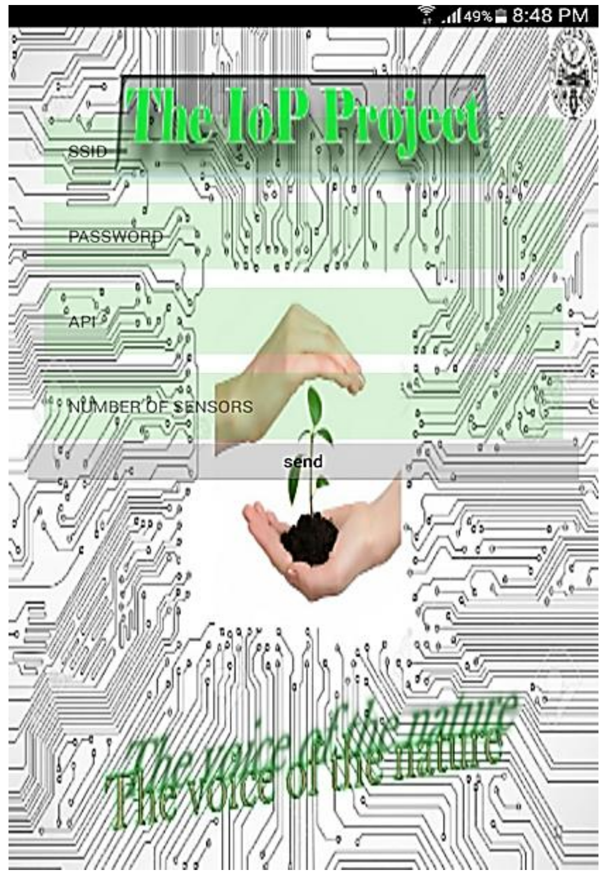

Fig. 4. The user interface of the IoP android application.

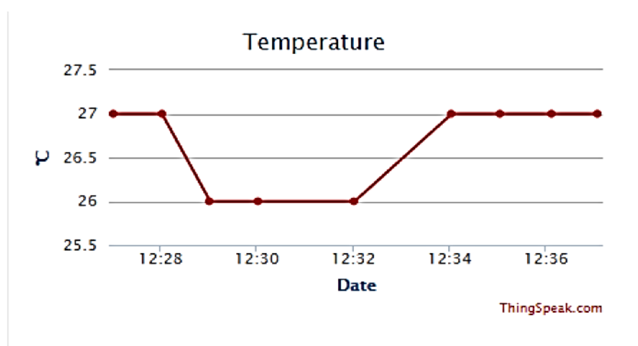

Fig. 5. Air temperature.

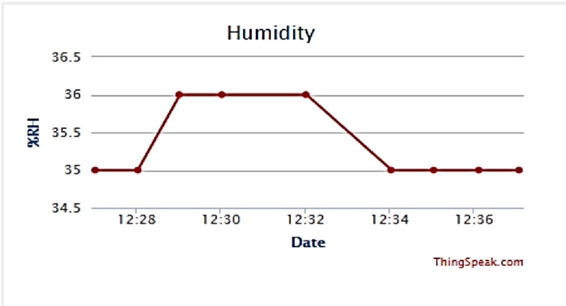

Fig. 6. Relative humidity.

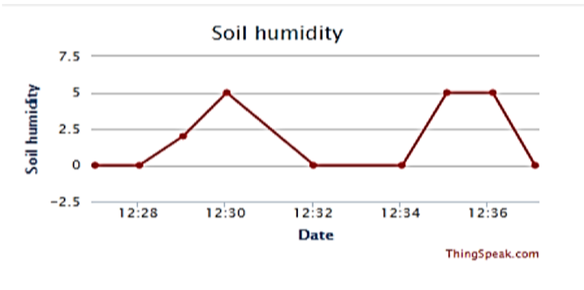

Fig. 7. Soil moisture.

G. PCB Layout with OrCAD Allegro

After creating a footprint of the PCB, the netlist of the circuit has been generated from the Capture tool and a file with the extension .brd is generated. Some constrain on the nets have been defined, in our case the power supply, +3V3 and GND nets need to change the configuration from the default one. After setting the correct environment, the components are placed. The board consists of a double layer PCB. The top layer holds components and provides a ground plane, the bottom one holds coupling capacitors and another ground plane. These two-ground plane are connected through series of vias to avoid big ground loops. In order to design correctly the PCB, several rules were employed for noise rejection and power dissipation. The methodology employed for placing and routing components is elaborated as follows.

PCB traces width: Maximum current consumption through the board occurs when the ESP module communicates over a WiFi connection. The total current, considering also the current required for the MCU module and the other circuitry is approximately 150 + 215 + (35*6) + 2.5 =577.5 mA. Thus, for 577.5 mA maximum current owing to power supply traces, 0.66 mm has been set for power traces.

PCB traces geometry and position: To avoid reactions 90 angles traces have been avoided. Since the board doesn't employ high-frequency signals, cross-talk effects have been neglected.

Decoupling capacitors: Near each power supply pins of each device decoupling capacitors have been used, to isolate direct current (DC) power supply pins from alternating current (AC) radiated noise.

Ground: Two ground planes, one for each side of the PCB have been used. These improve electromagnetic noise immunity but must be properly designed, to avoid undesired ground loops, by employing series of via between the two planes.

After defining the restrictions, it was possible to route the components. For this task, the automatic router of Allegro has been used and some more critical connections have been re-positioned.

The final PCB with highlighted measurements can be viewed in Fig. 8. It's also possible to see a 3D view of the PCB in Fig. 9.

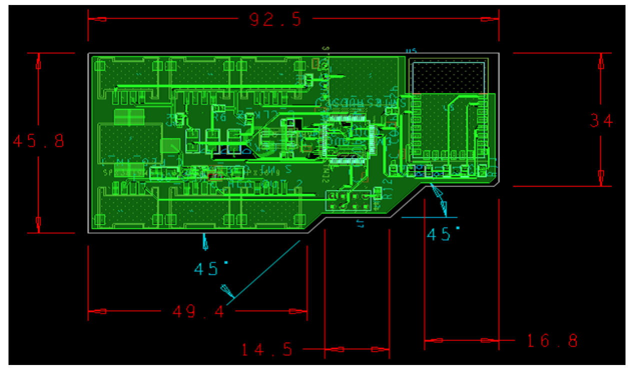

Fig. 8. PCB design with OrCAD Capture.

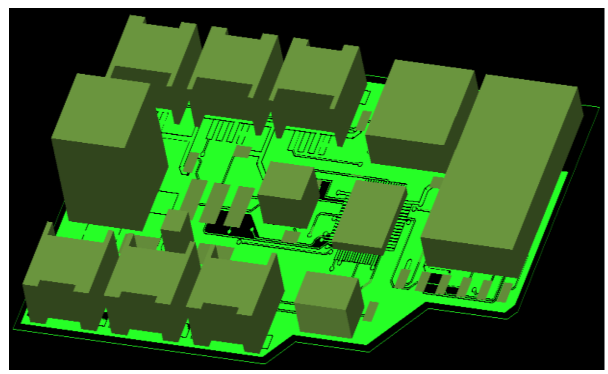

Fig. 9. 3D view of the PCB design.

The final cost of the product is around 25 Euro, where the main costs of production are provided from the DHT11 sensor, the MCU and the WiFi module, that cover about the 72% of the costs

III. Artificial Neural Network as Forecaster

Artificial Neural networks are becoming a significant method for prediction and forecasting in almost every field. Utilizing the time series approach to forecasting problem, forecasters actually collect and analyze the historical observations to create a model to capture the underlying behaviour of the data.

In Scientific research of meteorology, the weather forecasting is actually a typical unbiased time series forecasting problem. The real-life process is mostly nonlinear which brings deficiency in the traditional mathematical model. The metrological data is very irregular and it follows nonlinear Trend. Application of neural network in time series forecasting is based on the ability of the neural network to approximate nonlinear functions.

ANN is a mathematical model based on the structure and functions of biological neural networks. Information that flows through the network affects the structure of the ANN and it is generally based on three layers: an input layer, hidden and an output layer. Each layer contains nodes or neurons. Input layer consists of input nodes and output layer has as many nodes as the output. The middle layer or hidden layer contains an arbitrary number of nodes which is chosen after some trial and error initially. Fig. 10 shows a schematic view of feed forward Multilayer Perceptron (MLP). The mathematical representation is demonstrated in equation 1 where θ is the vector parameter which contains all the adjustable parameters of the network such as weights and biases$\{\boldsymbol{w_{j,l}}, \boldsymbol{W_{i,j}}\}$ and they are determined from a set of examples through the process called training. The examples, or the training data as they are usually called, are a set of inputs, u(t), and corresponding desired outputs, y(t).

$$\hat{y}(t) = g_i[z,\theta] = F_i\left[\sum_{j=1}^{n_h} W_{i,j} f_j\left(\sum_{l=1}^{n_z} w_{j,lzl} + w_{j,0}\right) + W_{i,0}\right]$$

In general, the aim of the experiment is to acquire a set of data that describes how the system behaves over its entire range of operation. Different input(s), u, is given on input layer and the impact on the output(s), y is observed. If the system to be identified is unstable or contains lightly damped dynamics it may be necessary to conduct the experiment again.

Afterwards, the network must be trained and when it starts to give reasonably accurate results, we can use the same network to predict unseen data. The flow chart for the selection and training of a predictive model is presented in Fig. 11.

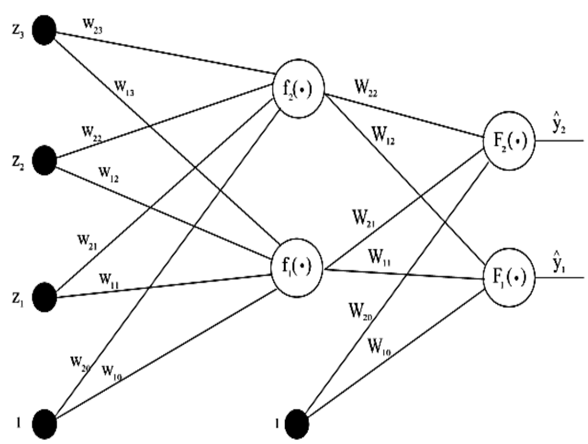

Fig. 10. MLP network with three input nodes and one bias, 2 hidden layers and an output layer with 2 nodes.

IV. Experimental Results

The primary goal of our research was to develop an IoT system which can acquire the remotely sensed weather data through an android application. Additionally, it should be capable to foresee the future data by implementing the neural networks for forecasting. Consequently, this aims to present an application for smart agriculture. In this part, we present the time series prediction of meteorological parameters using ANN.

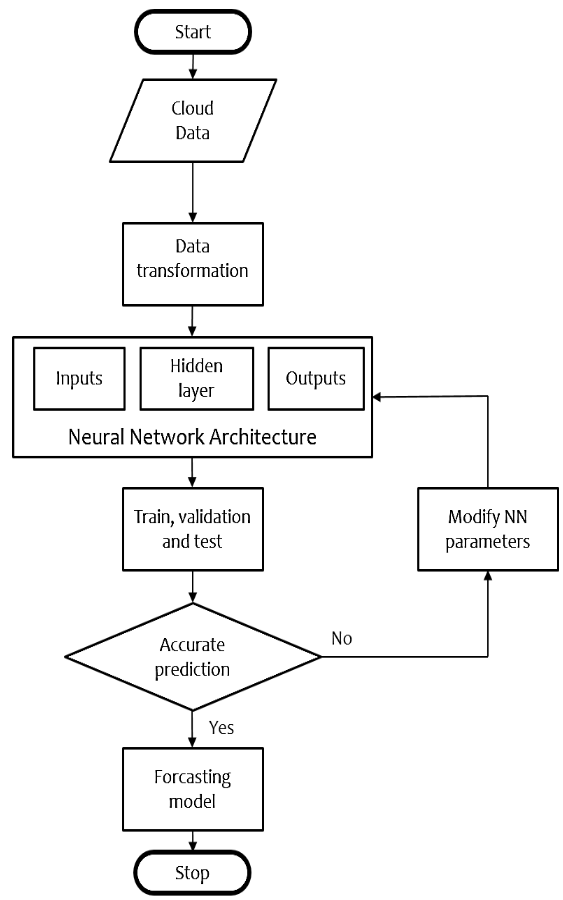

Fig. 11. The flow chart for the predictive model.

In order to select the appropriate predictive model, we took three different types of neural architectures into consideration which are further discussed in next section. Depending on the choice of input or regression vector, different nonlinear model structures emerge.

A. Time Series Temperature Prediction Model

Forecasting of temperature is important for many agriculture applications. Forecasts of the temperature of the soil, water, crop canopies or specific plant organs are also important in some specific cases. Crop species exhibit the phenomenon of thermo-periodicity, which is the differential response of crop species to daytime, nocturnal and mean air temperatures. It is possible to derive mean day and night-time temperatures from maximum and minimum temperatures data.

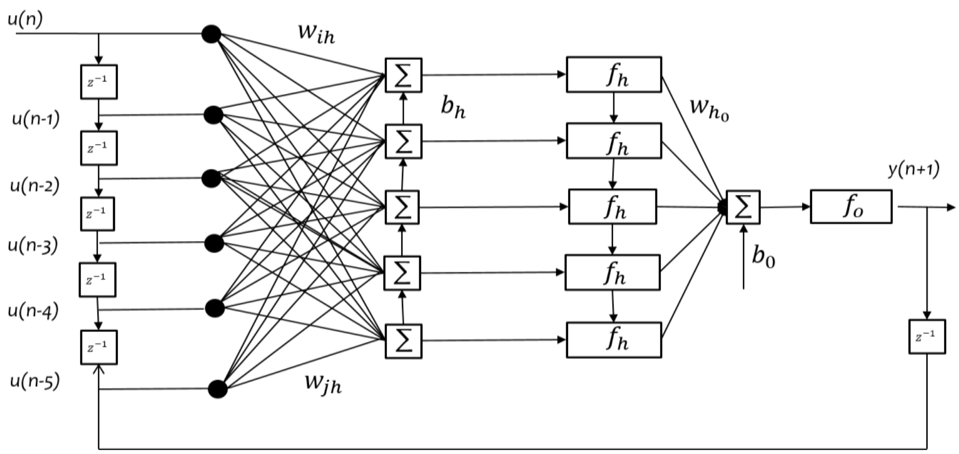

Fig. 12. NARX architecture model of 10-day forecasting maximum and minimum temperatures.

In order to predict maximum and minimum temperatures, we developed a nonlinear autoregressive exogenous model (NARX) ANN time series model. The followed approach is not intended for proposing a new method, but to use the dynamic and more robust method from the family of neural networks for forecasting the atmospheric temperature related to our aim. NARX has shown promising results and an accurate prediction in earlier studies [23]-[25]. NARX belongs to the family of Recurrent Neural Networks. The RNN have been widely used as one of the most popular data-driven dynamic models. In next sections, we present the data preprocessing and the followed steps for experimental setup.

B. Data Preprocessing and Feature Extraction

Since data collection takes a long time in agriculture applications, we took data from Oak Park weather station. In order to compute accurate predictions, we must have some meaningful attributes that provide content contribution and possibly reduced error rates. However, the valuable information can be lost as the choice of filtering and signal processing techniques require the expertise of data.

Feature selection, also known as variable selection or attribute selection. It is the process of selecting a subset of relevant features (variables, predictors) for use in model construction. As the input attributes, we used date (day, month, year), rainfall and season. The season attribute is related to each of the four divisions of the year (winter, spring, summer and autumn). As an input feature, this season attribute is represented by a numerical value consecutively. In total, there are five nodes in the input layer. Our target (output layer) in the model was to predict maximum and minimum temperatures.

In order to remove the noisy fluctuations from the time series data, we filtered the target data with a Savitzky–Golay filter. Savitzky-Golay filtering can be thought of as a generalized moving average. In fact, the moving average filter smoothes the data by replacing each data point with the average of the neighbouring data points defined within the span.

For more accurate predictions, it is necessary to choose statistically the lagged terms, which are highly correlated with the predicted value Autocorrelation and Cross-correlation functions were used to determine the correct input/output patterns for nonlinear time series forecasting [26]. The correct combination and selection of lag terms also place strong impact on proper forecasts. These terms are also known as delay terms.

Moreover, the time series data usually have an explicit dependency on time variable. It can be observed that, given an input x(t) at time t, the model computes y(t), albeit similar input at later in time can be related to different prediction. Due to this reason, the predictive model must be equipped with more data input from past or must have memory for past inputs.

C. Forecasting Model

Prior to network modelling, the next step is the selection of meaning full attributes and proper lag terms. Consequently, the subsequent step is to develop and train ANN model. Since our aim was to predict both, the maximum and minimum temperatures, we created two separate models. The above significant lag terms for both models were determined by the correlation function.

The number of hidden neurons was selected by optimizing the model several times depending on validation set error. The approach followed here is in line with [25]. Finally, two hidden layers were selected with [10 50] number of neurons which fitted well to both time series data better than others. These trained models were further transformed into closed loop models so that multi-step ahead predictions could be computed.

The general NARX architecture of the multi-step forecasting model for maximum and minimum temperatures is shown in Fig. 12.

Where W_{ih}, W_{jh} and who represents weights of network and they are multiplied by inputs and added together in bias $b_h$and $b_o$and subjected to an activation function which decides the output of the network. The training was performed with Levenberg-Marquardt training algorithm, which is good at curve fitting for nonlinear regression problems.

D. Discussion

This paper proposed Internet of Plants (IoP) system based on monitoring temperature, humidity and soil moisture of plants. Further, collected data could be used to train a predictive ANN model. In order to train ANN model, we used Oak Park weather station dataset. We selected NARX as a forecasting model due to its dynamic nature and long-term dependency. For both maximum and minimum temperatures data, we trained separate neural network models. Mean Square Error (MSE) is chosen as performance measuring criteria. Prior to training, the data was divided into a training set, validation set and test set.

Trained ANN model was further extrapolated in a closed loop. Afterwards, this closed loop model was able to predict 10-days forecast for maximum and minimum temperatures. As we increased the steps ahead for forecasting, it is obvious that the propagation of error in each sample will be raised.

In the following section, we present results of performance of ANN model. The performance of trained models is shown in Fig. 13 and 14. The model for maximum temperature prediction provided an error of 0.8826 on unseen data for the month of September. Similarly, the performance of the model predicting minimum temperature was tested. It resulted in an error value of 0.944.

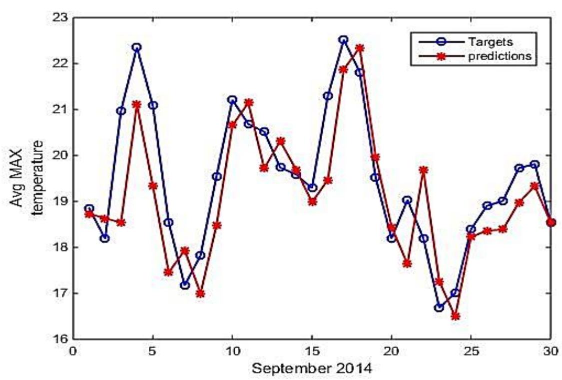

Fig. 13. Performance of the maximum temperature.

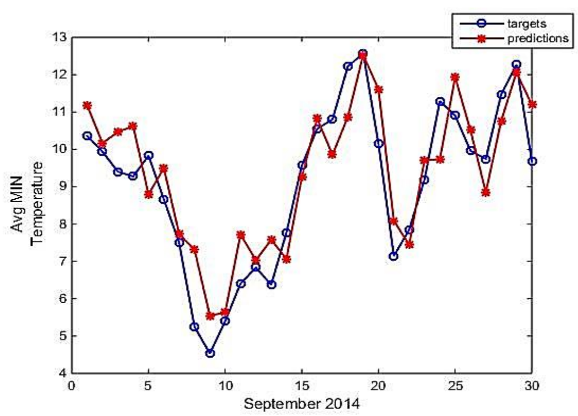

Fig. 14. Performance of the minimum temperature.

As mentioned earlier, the trained models were extrapolated in closed loop form to perform multi-step ahead predictions. The multistep forecasting computed for the next 10 days. The estimation for 10-days forecast maximum and minimum temperatures were obtained by a forecaster for the month of October. The performance of both predictive models for maximum and minimum temperatures for 10-days forecast is shown in Fig. 15 and 16 respectively. The figure shows, that the estimated forecasts for longer horizon can be achieved but with reduced accuracy. Since multistep forecasting is much more challenging than a one-step-ahead prediction. Thus, the accumulation of error increases, which degrades the prediction performance. This application can be considered general: it can be further applied to predict other climate attributes which are necessary for the deployment of smart agriculture.

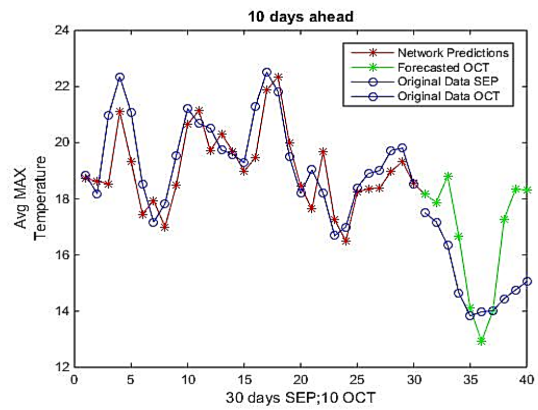

Fig. 15. Ten-days forecast of maximum temperature.

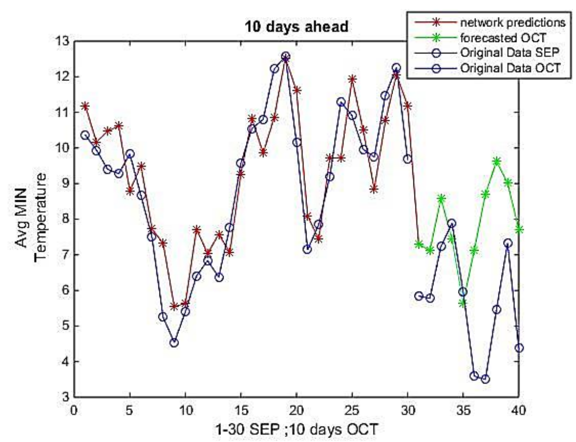

Fig. 16. Ten-days forecast of minimum temperature.

V. Conclusion

The main objective of our research was to develop an IoP based system which contains prototype device, the Android application as the hotspot to acquire physical data of the objects which are in long ranges and further send it to the cloud. We studied the network architecture, standards and protocols of IoT and selected the best communication protocols for our IoT designs. Considering the fact that in agriculture domain we need long distance communication and power efficient technologies, we proposed the WSN technology which is useful for field smart agriculture applications.

Additionally, we did 10 days ahead temperature forecasting using ANNs. Temperature prediction is important for many agriculture applications. However, the weather time series data is non-stationary and complex with unique characteristics that make them more challenging to get analyzed. In order to forecast the maximum and minimum temperatures for ten days ahead our preferred neural model is NARX. The deployed ANN application can be further applied to predict other climatic parameters of a smart agriculture.

ACKNOWLEDGEMENTS

Authors are thankful to Prof. Giansalvo Cirrincione for his support on supervising on the neural networks for modelling dynamic systems.

The work has been done during PhD period supported by TIMUR project, Erasmus Mundus Action 2 program and Higher Education Commission (Pakistan).

REFERENCES

[1] Adamchuk, V. I., Hummel, J. W., Morgan, M. T., & Upadhyaya, S. K. (2004). On-the-go soil sensors for precision agriculture. Computers and Electronics in Agriculture.

[2] Camilli, A., Cugnasca, C. E., Saraiva, A. M., Hirakawa, A. R., & Corrêa, P. L. P. (2007). From wireless sensors to field mapping: Anatomy of an application for precision agriculture. Computers and Electronics in Agriculture, 58(1).

[3] Zhang, N., Wang, M., & Wang, N. (2002). Precision agriculture—a worldwide overview. Computers and Electronics in Agriculture, 36(2–3), 113–132. https://doi.org/10.1016/S0168-1699(02)00096-0

[4] Schellberg, J., Hill, M. J., Gerhards, R., Rothmund, M., & Braun, M. (2008). Precision agriculture on grassland: Applications, perspectives and constraints. European Journal of Agronomy. https://doi.org/10.1016/j.eja.2008.05.005

[5] Cox, S. (2002). Information technology: The global key to precision agriculture and sustainability. In Computers and Electronics in Agriculture (Vol. 36, pp. 93–111).

[6] Corwin, D. L., & Lesch, S. M. (2005). Apparent soil electrical conductivity measurements in agriculture. Computers and Electronics in Agriculture. https://doi.org/10.1016/j.compag.2004.10.005

[7] Lee, W. S., Alchanatis, V., Yang, C., Hirafuji, M., Moshou, D., & Li, C. (2010). Sensing technologies for precision specialty crop production. Computers and Electronics in Agriculture.

[8] Mihai T. Lazarescu. (2017). Wireless Sensor Networks for the Internet of Things: Barriers and Synergies. In Components and Services for IoT Platforms (pp. 155–186). Springer International Publishing.

[9] Xiang YAN, Ji-yun JIN, Ping HE, M. L. (2008). Recent Advances on the Technologies to Increase Fertilizer Use Efficiency. Agricultural Sciences in China, 7(4), 469–479.

[10] Kaloxylos, A., Eigenmann, R., Teye, F., Politopoulou, Z., Wolfert, S., Shrank, C., … Kormentzas, G. (2012). Farm management systems and the Future Internet era. Computers and Electronics in Agriculture, 89, 130–144

[11] F. TongKe. (2013). Smart agriculture based on cloud computing and iot. Journal of Convergence Information Technology, 8(2).

[12] Siuli Roy, A. D., & Bandyopadhyay, S. (2008). Agro-sense: Precision agriculture using sensor-based wireless mesh networks. In International Telecommunication Union - Proceedings of the 1st ITU-T Kaleidoscope Academic Conference, Innovations in NGN, K-INGN.

[13] Yinbiao, Shu, P. Lanctot, and F. J. (2015). Internet of things: wireless sensor networks. White Paper, International Electrotechnical Commission.

[14] Karakostas, B. (2016). Event Prediction in an IoT Environment Using Naïve Bayesian Models. In Procedia Computer Science (Vol. 83, pp. 11–17).

[15] Molaei, S. M., & Keyvanpour, M. R. (2015). An analytical review for event prediction system on time series. In 2015 2nd International Conference on Pattern Recognition and Image Analysis (IPRIA) (pp. 1–6).

[16] Aliev Khurshid, Rugiano Francesco, Pasero Eros. (2016b). The Use of Bluetooth Low Energy Smart Sensor for Mobile Devices Yields an Efficient Level of Power Consumption. In The First International Conference on Advances in Sensors, Actuators, Metering and Sensing (pp. 5–9). Venice.

[17] Aliev Khurshid, Rugiano Francesco, Pasero Eros. (2016a). Smartphone and Bluetooth Smart Sensor Usage in IoT Applications. Sensors & Transducers, 201(6), 27–34.

[18] https://thingspeak.com/

[19] www.mathworks.com

[20] https://www.arduino.cc/

[21] https://www.raspberrypi.org

[22] http://beagleboard.org.

[23] Bektas Oguz, Jeffrey Jones. (2016). NARX Time Series Model for Remaining Useful Life Estimation of Gas Turbine Engines. In Third European Conference of the Prognostics and Health Management Society (pp. 1–10).

[24] Aliev Khurshid, Narejo Sanam, Pasero Eros, I. J. (2016). A Predictive Model of Artificial Neural Network for Fuel Consumption in Engine Control System. In WIRN 2016 26th Italian Workshop on Neural Networks

[25] Sanam Narejo and Eros Pasero. (2016b). Time series forecasting for outdoor temperature using nonlinear autoregressive neural network models. Journal of Theoretical and Applied Information Technology, 94(2), 451–463

[26] Sanam Narejo and Eros Pasero. (2016a). A Hybrid Approach for Time series Forecasting using Deep Learning and Nonlinear Autoregressive Neural Network. In INTELLI 2016 : The Fifth International Conference on Intelligent Systems and Applications (includes InManEnt 2016) (pp. 69–75).