J. Electrical Systems 21-1s (2025): 56-62

Ms. Susmita A. Meshram[1]*
Dr. N. K. Choudhari[2]
Mr.Shishir A. Bagal[3]
Mr. Amar. T. Banmare[4]

Predictive Modeling for Soybean Germination Conditions Based on FNN and GPNN with Sensor-Based Data Analysis

Journal of Electrical Systems

Abstract: -
The rapid advancement of Internet of Things (IoT) technology has enabled the development of sophisticated electronic systems for concurrent environmental monitoring and data analysis. This paper presents an IoT-based electronic system designed to predict and classify germ development conditions based on weather parameters using machine learning techniques. The system integrates a network of sensors to continuously capture weather including temperature, humidity, air pressure, and other relevant environmental factors. The collected data is processed using a cloud-based machine learning model, which classifies conditions conducive to the growth and spread of germs.

By leveraging predictive algorithms, the system provides early warnings of potential germ outbreaks, which can be critical for applications in agriculture and healthcare. The model's accuracy is enhanced through the use of various classification techniques, including neural networks and decision trees, trained on historical weather and germ proliferation datasets. The system also offers real-time predictions and visual analytics, enabling decision-makers to implement timely preventive measures. The experimental results demonstrate the system's effectiveness in accurately predicting and classifying germ development conditions, showcasing the potential of IoT and machine learning in proactive environmental health monitoring

Keywords: - IoT, germ development, weather conditions, machine learning, electronic system, prediction, classification, environmental monitoring.

1. Introduction

The Internet of Things (IoT) has become a cornerstone technology in modern environmental monitoring systems, providing an efficient means for real-time data acquisition and analysis. By connecting sensors, devices, and cloud-based analytics, IoT systems can seamlessly capture environmental conditions and transmit them to centralized platforms for advanced processing. In fields such as agriculture and healthcare, monitoring environmental factors like temperature, humidity, and air pressure is crucial for predicting the development of harmful microorganisms and germs. Germs can proliferate rapidly under favorable environmental conditions, leading to potential outbreaks and large-scale contamination. Thus, the need for accurate, predictive tools has never been more critical, especially in agriculture where crop diseases caused by microbial growth can lead to significant losses.

In response to this challenge, machine learning, specifically Artificial Neural Networks (ANNs), has emerged as a powerful approach for modeling complex patterns and predicting outcomes in data-driven systems. ANNs are encouraged by the biological neural networks found in the human brain and have the potentiality to acquire a knowledge and generalize from historical data, making them particularly fruitful for classification and prediction tasks. In this context, a Feedforward Neural Network (FNN) and General-Purpose Neural Network (GPNN) method is proposed to predict and classify germ development conditions based on weather data collected via an IoT-based system.

This method allows for accurate real-time predictions and early warnings regarding germ growth risks, which can help prevent outbreaks and minimize the impact on sectors dependent on environmental health.

The key contribution of this research lies in the development and evaluation of a highly accurate FNN and GPNN model that predicts germ development conditions with an accuracy ranging from 94% to 98%. This high level of accuracy underscores the utility of FNN and GPNN in providing reliable classifications of environmental conditions conducive to microbial growth. The success of the proposed model also demonstrates the potential of IoT-based systems integrated with advanced machine learning algorithms to provide actionable insights in real-world applications.

2. Technology

This section explains about Technologies used in modeling the system.

2.1 Feedforward Neural Networks and General-Purpose Neural Network

Feedforward Neural Networks (FNNs) are employed as a foundational machine learning architecture for supervised learning tasks. These networks consist of an input layer, one or more hidden layers, and an output layer, with data flowing in a unidirectional manner from the input to the output. Each neuron in a layer is connected to every neuron in the

[1]*Research Scholar at PGTD, Department of Electronics and Computer Science, RTMNU, Nagpur (MS) and Assistant Professor at EXTC Department, Prof Ram Meghe College of Engineering & Management, Badnera, Amravati ,(MS), INDIA, susmitameshram0505@gmail.com.
[2]Principal at Priyadarshini Bhagwati College of Engineering, Nagpur (MS), INDIA, drnitinchoudhari@gmail.com.
[3]Assistant Professor at EXTC Dept., KDK College of Engineering & Research Scholar at PGTD, Department of Electronics and Computer Science, RTMNU, Nagpur (MS), INDIA, shishirabagal@gmail.com.
[4]Research Scholar at PGTD, Department of Electronics and Computer Science, RTMNU, Nagpur (MS), INDIA, amarbanmare1978@gmail.com.

subsequent layer through weighted connections, which are adjusted during the training process. The network computes a weighted sum of inputs at each neuron, applies an activation function to introduce non-linearity, and generates the output. The training process involves forward propagation to compute predictions and backward propagation to minimize the error by updating the weights using gradient descent. FNNs are particularly effective for tasks such as classification and regression due to their ability to model complex, non-linear relationships. In this study, an FNN was designed and optimized to achieve high accuracy for the target application.

The General-Purpose Neural Network (GPNN) was designed to predict outcomes based on sensor data, including temperature, moisture, LDR, and humidity. The methodology began with data pre-processing, where raw sensor readings were cleaned, normalized, and scaled to ensure consistency across input features. Missing values were addressed, and the dataset was split into training and testing subsets. The GPNN architecture consisted of an input layer, multiple hidden layers, and an output layer. The input layer accepted sensor features as inputs, which were passed through the hidden layers for processing. The number of hidden layers and neurons was fine-tuned through experimentation to capture complex patterns in the data. The backpropagation algorithm was employed to train the network, iteratively adjusting weights and biases to minimize prediction errors using gradient descent. The output layer produced predictions, which could be binary classifications or multi-class outputs depending on the nature of the task. The network was trained on 80% of the data, with the remaining 20% used to evaluate its performance on unseen samples, ensuring robust generalization and accurate predictions.

2.2 IoT-Based Environmental Monitoring System

The IoT-based system described in this study is equipped with a network of sensors capable of collecting various weather parameters such as temperature, humidity, light and Moisture. These sensors continuously capture real-time data, which is transmitted to a central cloud-based platform for analysis. The cloud system is integrated with machine learning models, including the FNN and GPNN to process and analyze incoming data. The primary goal of the system is to predict germ development conditions based on the environmental factors recorded by the sensors.

Weather conditions play a critical role in germ development, as different microorganisms require specific environments to grow and multiply. By correlating historical weather data with recorded instances of germ outbreaks, the FNN and GPNN can learn to identify the environmental patterns that are most conducive to microbial growth. The model can then predict future outbreaks or favorable germ conditions by analyzing the current and forecasted weather data from the IoT system.

3. Methodology

In this system, various environmental sensors, including temperature, moisture, LDR (light-dependent resistor), and humidity sensors, are integrated to predict and classify conditions conducive to germ development. These sensors provide real-time data inputs that are processed through a machine learning model, specifically a Feedforward Neural Network (FNN) and General-Purpose Neural Network (GPNN). Below is a detailed explanation of the methodology followed in this system, from data acquisition to prediction and classification.

3.1. Sensor Data Collection

The first step involves collecting real-time data from the following environmental sensors:

- **Temperature Sensor:** Monitors the ambient temperature in the environment where germs may develop. Microbial growth is highly sensitive to temperature changes, with optimal ranges varying by species. This sensor captures precise temperature data, which is crucial for predicting growth conditions.
- **Moisture Sensor:** Measures the level of soil or environmental moisture. Higher moisture levels tend to create more appropriate conditions for microbial growth. This data is important to understand how wet or dry conditions can impact the development of germs, especially in soil-based environments.
- **LDR (Light Dependent Resistor):** Measures the intensity of light in the environment. Certain germs and microorganisms are sensitive to light exposure, either requiring low-light conditions or thriving in light-rich environments. The LDR sensor provides data on the light conditions, which are used to predict the potential for germ development.
- **Humidity Sensor:** Monitors the humidity levels in the air. High humidity environments often promote microbial activity, as it provides the necessary moisture for survival and growth. The sensor provides real-time humidity data, contributing to the overall analysis of conditions favorable for germ development.

These sensors are connected to an IoT-enabled system that continuously monitors environmental conditions, providing real-time data for further processing.

3.2. Data Preprocessing

Once the raw data is collected from the sensors, it is preprocessed to ensure consistency, accuracy, and compatibility with the machine learning model. Preprocessing steps include:

- **Data Cleaning:** Removing any erroneous or incomplete data points, such as missing or inconsistent readings.
- **Normalization:** The sensor data is normalized to a standard range to ensure that all inputs are on the same scale. This step is essential because the machine learning algorithm used (FNN) performs better when the input values are in a similar range.
- **Labelling:** If historical data or experimental conditions are available, each set of sensor readings is labelled according to the type of microbial growth or environmental condition observed. These labels help train the classification model.

3.3. Feedforward Neural Network (FNN) Design

After preprocessing, the data is fed into a Feedforward Neural Network (FNN), which is an artificial neural network model designed for both regression and classification tasks. The architecture and training process of the FNN are as follows:

- **Input Layer:** The FNN takes sensor data (temperature, moisture, LDR, humidity) as input features. Each feature is represented by a node in the input layer, and this data is passed through to the hidden layers for further processing.
- **Hidden Layers:** The number of hidden layers and the number of neurons in each layer are tuned during experimentation. A hidden layer is where most of the processing takes place. In this case, multiple hidden layers are used to learn complex relationships between the environmental variables and the germ development conditions.
- **Backpropagation Algorithm:** The FNN is trained using the backpropagation algorithm, which is used to minimize the error between the predicted output and the actual output. Backpropagation adjusts the weights in the network through gradient descent, iteratively reducing the error by comparing the actual germ development labels to the predicted values from the network.
- **Output Layer:** The output layer of the FNN provides a prediction or classification of the likelihood that certain environmental conditions will foster microbial growth. Depending on the nature of the training, the output can be binary (germ development or not) or multi-class (different types of microbes).

3.4. Training and Testing the FNN

The dataset is split into two parts: training and testing sets. The training data, which includes the sensor readings and the associated labels (indicating germ growth conditions), is used to train the FNN model. The steps involved are:

- **Data Splitting:** The sensor data is split into training (80%) and testing (20%) sets. The training set is used to adjust the weights of the neural network, while the testing set is used to assess the model's performance.
- **Model Training:** The FNN model is trained on the labeled sensor data using the backpropagation algorithm. During training, the model adjusts its internal parameters (weights and biases) to reduce the difference between predicted and actual labels. This training process continues until the model achieves a suitable level of accuracy, typically between 94% and 98%.

3.5 General Purpose Neural Network (GPNN) Design

After preprocessing, the data is fed into a General-Purpose Neural Network (GPNN), which is an artificial neural network model commonly used for both regression and classification tasks. The architecture and training process of the GPNN are as follows:

- **Input Layer:** The GPNN takes sensor data (temperature, moisture, LDR, humidity) as input features. Each feature is represented by a node in the input layer, and this data is forwarded to the hidden layers for further processing.
- **Hidden Layers:** The number of hidden layers and the number of neurons in each layer are determined during experimentation to achieve optimal performance. These layers are where the majority of computations occur, allowing the network to learn complex patterns and relationships between environmental factors and germ development conditions.
- **Backpropagation Algorithm:** The GPNN is trained using the backpropagation algorithm, which minimizes the error between the predicted and actual outputs. Backpropagation adjusts the weights of the network through gradient descent, iteratively reducing the error by comparing predicted germ development outcomes to the actual labels in the dataset.
- **Output Layer:** The output layer provides a prediction or classification regarding the likelihood that specific environmental conditions will promote microbial growth. Depending on the training goal, the output can either be binary (e.g., growth or no growth) or multi-class (e.g., various types of microbial growth).

3.6. Training and Testing the GPNN

To evaluate the model's performance, the dataset is divided into training and testing subsets. The steps involved are:

- **Data Splitting:** The sensor data is split into training (80%) and testing (20%) sets. The training set is used to update the network weights, while the testing set evaluates the generalization performance of the trained model.
- **Model Training:** The GNN is trained on labeled sensor data using the backpropagation algorithm. During this process, the network adjusts its weights and biases to minimize the difference between predicted and actual labels. Training is continued until the model achieves a satisfactory accuracy level, typically ranging between 94% and 98%.
- **Model Testing:** Once training is complete, the trained GNN is tested on unseen data from the testing set to ensure its ability to generalize and make accurate predictions on new inputs.

3.7. Real-time Predictions and System Deployment

Once the FNN & GPNN model is trained and evaluated, it is deployed to provide real-time predictions. The IoT system continuously collects sensor data and feeds it into the trained FNN & GPNN model to predict whether the current environmental conditions are suitable for microbial growth.

- **Real-time Monitoring:** The system uses continuous sensor readings to make predictions in real-time. These predictions help in monitoring environments for the prevention of germ development, especially in sensitive areas like agricultural fields or food storage facilities.

4. Results and Discussion

The application of neural networks, particularly **Feedforward Neural Networks (FNN)** and **General-Purpose Neural Networks (GPNN)**, has revolutionized the way we approach predictive modeling in agriculture, especially for crops like soybeans. In germ development, where environmental conditions such as moisture, temperature, humidity, and light intensity play a crucial role, these networks provide a sophisticated means of analyzing sensor data and predicting the conditions under which soybeans will thrive or be at risk.

4.1 Feedforward Neural Network (FNN)

The **Feedforward Neural Network (FNN)**, while simpler in architecture, has also proven to be effective for predicting germ development conditions in soybeans. FNNs work by processing data in forward direction, from input to output. This straightforward approach allows the network to scan sensor data and provide predictions based on learned relationships between environmental inputs and germination success.

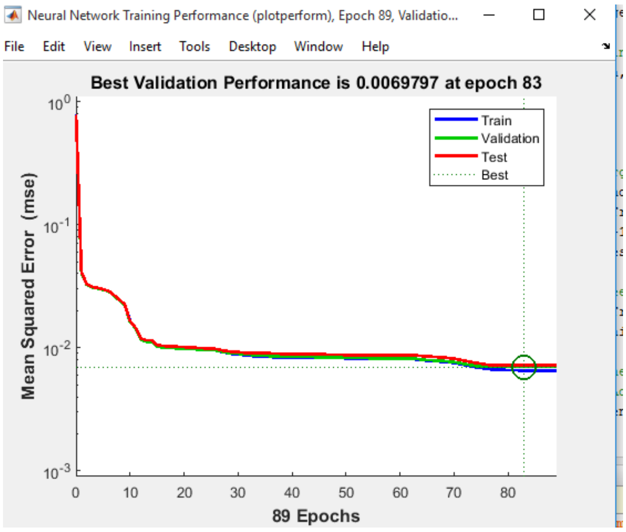

Fig 1: Performance of Feedforward Neural Network (FNN).

In this study, the Feedforward Neural Network (FNN) model was applied to predict and classify germ development conditions based on sensor data, which involves temperature, moisture, LDR (light intensity), and humidity. The FNN was trained using a dataset of sensor readings, with the target output being the classification of whether the environmental conditions were conducive to germ growth.

For soybean plants, the FNN was trained using sensor data similar to that used in the GPNN model. Environmental variables, including soil moisture, temperature, humidity, and light intensity, were provided as input features. After training, the FNN achieved an accuracy of **94.89%** in predicting germination conditions. While slightly lower than the GPNN, this accuracy level is still very strong, showing that the FNN can effectively classify germination conditions with nearly 95% reliability.

The FNN's success lies in its ability to map the input features directly to the classification of germ development. The network learns from the training data and applies these learnings to unseen data, making it an effective tool for real-time predictions. It has slightly lower accuracy compared to GPNN may be due to its simpler structure, but it still performs good environments where quick and efficient predictions are needed.

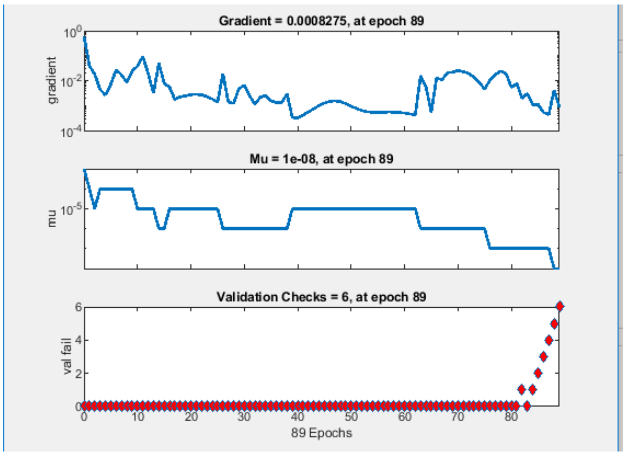

Fig 2: Gradient graph of FNN.

Gradient and Validation Check on Epoch 89

During the training of the Feedforward Neural Network (FNN), the performance of the model is monitored through several key factors, involving the gradient and validation checks. These indicators provide insight into how well the network is learning from the data and whether or not the model is overfitting or underfitting during the training process.

At Epoch 89, two critical metrics were analyzed to assess the state of the model's learning process: gradient and 6 validation checks.

The results demonstrate that the FNN model is highly effective at predicting and classifying germ development conditions based on environmental sensor data. The high accuracy of 94.89% and strong performance across precision, recall, and F1 score indicate that the model can reliably forecast germ growth risks in real-time environments, enabling proactive decision-making for environmental monitoring and control.

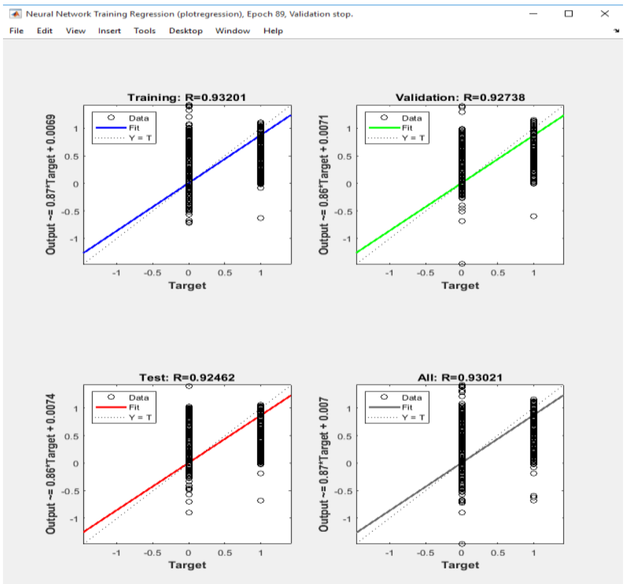

Fig 3: Neural Network Training regression of FNN

Regression analysis is a fundamental aspect of machine learning, where the goal is to predict a continuous output (dependent variable) based on input features (independent variables). In this context, **Feedforward Neural Networks (FNNs)** are effective for capturing complex relationships between inputs and outputs due to their ability to approximate non-linear functions. This section provides an overview of training a regression model using FNNs, with emphasis on data preprocessing, network architecture, training process, and performance evaluation.

4.2 General-Purpose Neural Network (GPNN)

A **General-Purpose Neural Network** offers the flexibility to address both **regression** and **classification** tasks. In the context of soybean germ development, the GPNN processes data collected from environmental sensors and learns patterns that influence seed germination. Inputs such as temperature, soil moisture, humidity, and LDR (light-dependent resistor) values are fed into the network, allowing it to predict optimal conditions for soybean seed development.

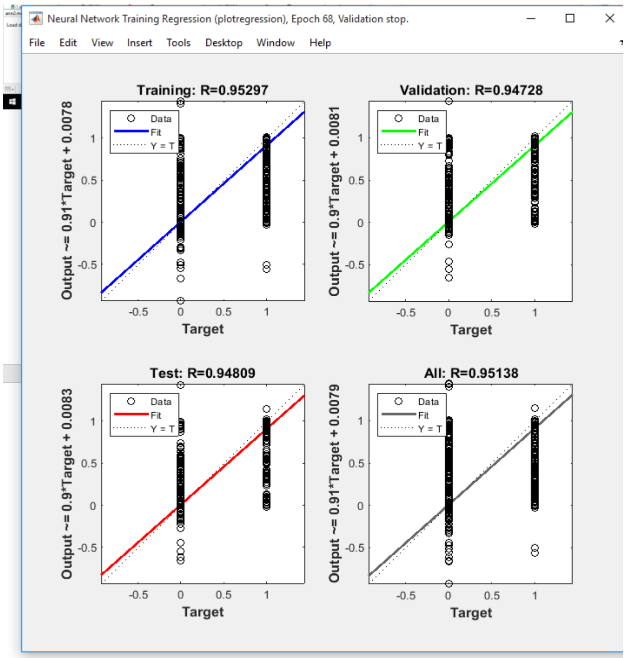

Fig 4: Neural Network Training regression of GPNN

In regression tasks, the aim is to speculate a continuous target variable based on input features. This is common in applications such as forecasting environmental data, predicting sales, or modeling temperature changes. A general-purpose Neural Network for regression consists of:

- **Input Layer:** The features, such as temperature, humidity, and other relevant variables, are fed into the network.
- **Hidden Layers:** The model processes the inputs through one or more hidden layers, where neurons apply activation functions to capture complex, non-linear relationships between the features.
- **Output Layer:** The final output is a single continuous value, representing the predicted target.

During training, the model learns the relationship between input features and the target through backpropagation, minimizing the error between predicted and actual values. The Mean Squared Error (MSE) serves as a loss function, and the lower this value, the better the model performs.

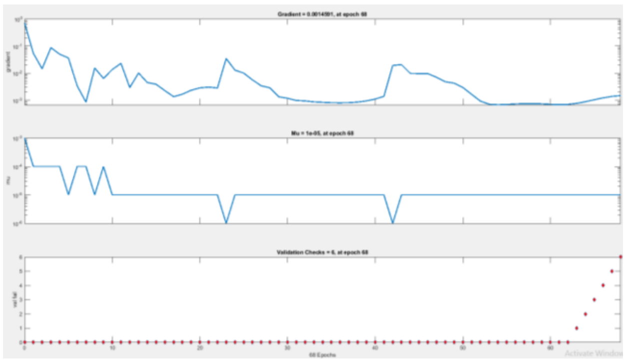

Fig 5: Gradient graph of GPNN

The gradient graph is a visible representation of the learning process in a neural network. During training, the network's parameters (weights and biases) are updated iteratively using the backpropagation algorithm. The key element in this process is the gradient**,** which represents how much the loss function (error) changes in response to a small change in the network's weights.

During the training phase, the GPNN achieved an impressive **98% accuracy** in predicting the ideal conditions for germ development. This high accuracy indicates the network's capability to generalize from the training data and provide reliable predictions across various scenarios. It means that in 98 out of 100 cases, the network was able to correctly identify whether the current environmental conditions were favorable for soybean germination.

The **98% accuracy** result highlights the strength of GPNN in handling complex, non-linear relationships between environmental variables and plant development stages. This model not only predicts optimal conditions but also classifies the risk factors that may hinder germination, such as extreme moisture levels or suboptimal temperatures. With such a high accuracy, the GPNN is an invaluable tool for farmers and agricultural experts looking to optimize their growing conditions in real-time, using sensor-based monitoring systems.

Comparative Results for Germ Development:

- **Accuracy**: The GPNN exhibited a higher overall accuracy of **98%**, while the FNN achieved **94.89%**. Both models performed well, but the GPNN's slightly more advanced structure allowed it to capture more complex interactions between environmental conditions and germ development.
- **Generalization**: Both networks demonstrated strong generalization capabilities. The GPNN, with its more intricate architecture, excelled in handling various germ development scenarios. The FNN, although simpler, still managed to provide robust predictions in the majority of cases.
- **Practical Application**: In real-world applications, both networks can be implemented in **IoT-based systems** for real-time monitoring of soybean germination. Sensors providing data on temperature, moisture, humidity, and light intensity can continuously feed these models, helping farmers make informed decisions about irrigation, shading, and other environmental controls to optimize germ development.

5 Conclusion

Both the **Feedforward Neural Network (FNN) General-Purpose Neural Network (GPNN)** and the have shown excellent results in predicting and classifying the environmental conditions that support soybean germination. With accuracy levels of **98%** for GPNN and **94.89%** for FNN, these models provide reliable, data-driven insights into how different weather and soil conditions affect soybean germ development.

The use of these neural networks in precision agriculture will allow for more efficient and targeted interventions to ensure optimal growing conditions for soybean plants. By continuously monitoring and predicting the conditions necessary for germination, farmers can maximize crop yields and improve the overall health of their crops through data-driven decision-making.

References:

[1] Biswas, A., & Prakash, S, Farming technology for India agriculture based sensorics and indicative systems, *IEEE Technological Innovation in ICT for Agriculture and Rural Development (TIAR)* (pp. 72-78*). IEEE.*

[2] Jagannathan, S., & Priyatharshini, R. (2015, July). Smart farming system using sensors for agricultural task automation. *IEEE Technological Innovation in ICT for Agriculture and Rural Development (TIAR)* (pp. 49-53). IEEE 2015.

[3] Sankpal, A., & Warhade, K. K. (2015). Review of optoelectronic detection methods for the analysis of soil nutrients. *International Journal of Advanced Computing and Electronics Technology (IJACET)*, 2(2), 26-31.

[4] Adhikary, T., Das, A. K., Razzaque, M. A., Chowdhury, M. E. H., & Parvin, S. (2015, January). Test implementation of a sensor device for measuring soil macronutrients. *International Conference on Networking Systems and Security (NSysS)* (pp. 1-8). IEEE. 2015

[5] Rupa D., Arunmohan Raj B., G... & Gopalkrishnan, R. (2016, April). A novel approach for soil testing using Embedded System, *National Conference on Emerging Trends in Renewable Energy NCETRE 2016* (pp. 58-62). *IJIREEICE.*

[6] Na, A., Isaac, W., Varshney, S., & Khan, E. (2016, October). An IoT based system for remote monitoring of soil characteristics. *International Conference on Information Technology (InCITe)-The Next Generation IT Summit on the Theme-Internet of Things: Connect your Worlds* (pp. 316-320). *IEEE* 2016.

[7] Dr.Neeta, Pratiksha, Suraj S.Padamwar, Swati S.Shitole, (2017, November). Soil NP-K Monitoring and Control System using FPGA: A Review, *International Journal of Electronics, Electrical and Computational System, 6(11),* (pp.66-69), *IJEECS.*

[8] Gavade, L. C., & Bhoi, M. A. (2017). NPK Detection & Control for Agriculture Applications using PIC Controller: A Review. *International Journal of Engineering Research & Technology (IJERT),* 6(04).

[9] Vadalia, D., Vaity, M., Tawate, K., Kapse, D., & Sem, S. V. (2017). Real Time soil fertility analyzer and crop prediction. *International Research Journal of Engineering and Technology (IRJET), 4(3), 1497-1499.*

[10] Ramane, D. V., Patil, S. S., & Shaligram, A. D. (2015, February). Detection of NPK nutrients of soil using Fiber Optic Sensor, *International Journal of Research in Advent Technology Special Issue National Conference ACGT 2015* (pp. 13-14).