

## Article

# A Deep Learning-Based Sensor Modeling for Smart Irrigation System

Maira Sami <sup>1,2</sup>, Saad Qasim Khan <sup>2</sup>, Muhammad Khurram <sup>3</sup>, Muhammad Umar Farooq <sup>4</sup>, Rukhshanda Anjum <sup>5</sup>, Saddam Aziz <sup>6</sup>, Rizwan Qureshi <sup>7</sup> and Ferhat Sadak <sup>8,\*</sup> 

- <sup>1</sup> Department of Computer Science, Shaheed Zulfiqar Ali Bhutto Institute of Science and Technology, Karachi 75600, Pakistan; mairasami03@gmail.com
  - <sup>2</sup> Department of Computer and Information Systems, NED University of Engineering and Technology, Karachi 75270, Pakistan; saadqasimkhan@neduet.edu.pk
  - <sup>3</sup> National Center of Artificial Intelligence, NED University of Engineering and Technology, Karachi 75270, Pakistan; mkhurram@neduet.edu.pk
  - <sup>4</sup> Department of Business Studies, Namal Institute Mianwali, Mianwali 42200, Pakistan; m.umar@namal.edu.pk
  - <sup>5</sup> Department of Mathematics and Statistics, University of Lahore, Lahore 54590, Pakistan; rukhshanda.anjum@math.uol.edu.pk
  - <sup>6</sup> College of Mechatronics Engineering, Shenzhen University, Shenzhen 518061, China; saddamaziz@yahoo.com
  - <sup>7</sup> College of Science and Engineering, Hammad Bin Khalifa University, Doha 34410, Qatar; engr.rizwanqureshi786@gmail.com
  - <sup>8</sup> Department of Mechanical Engineering, Bartin University, Bartin 74100, Turkey
- \* Correspondence: fsadak@bartin.edu.tr



**Citation:** Sami, M.; Khan, S.Q.; Khurram, M.; Farooq, M.U.; Ajum, R.; Aziz, S.; Qureshi, R.; Sadak, F. A Deep Learning-Based Sensor Modeling for Smart Irrigation System. *Agronomy* **2022**, *12*, 212. <https://doi.org/10.3390/agronomy12010212>

Academic Editors: Saeid Homayouni, Yacine Bouroubi and Karem Chokmani

Received: 6 November 2021

Accepted: 13 December 2021

Published: 16 January 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Abstract:** The use of Internet of things (IoT)-based physical sensors to perceive the environment is a prevalent and global approach. However, one major problem is the reliability of physical sensors' nodes, which creates difficulty in a real-time system to identify whether the physical sensor is transmitting correct values or malfunctioning due to external disturbances affecting the system, such as noise. In this paper, the use of Long Short-Term Memory (LSTM)-based neural networks is proposed as an alternate approach to address this problem. The proposed solution is tested for a smart irrigation system, where a physical sensor is replaced by a neural sensor. The Smart Irrigation System (SIS) contains several physical sensors, which transmit temperature, humidity, and soil moisture data to calculate the transpiration in a particular field. The real-world values are taken from an agriculture field, located in a field of lemons near the Ghadap Sindh province of Pakistan. The LM35 sensor is used for temperature, DHT-22 for humidity, and we designed a customized sensor in our lab for the acquisition of moisture values. The results of the experiment show that the proposed deep learning-based neural sensor predicts the real-time values with high accuracy, especially the temperature values. The humidity and moisture values are also in an acceptable range. Our results highlight the possibility of using a neural network, referred to as a neural sensor here, to complement the functioning of a physical sensor deployed in an agriculture field in order to make smart irrigation systems more reliable.

**Keywords:** neural networks; artificial intelligence; sensor reliability; agritech; precision agriculture; Recurrent Neural Networks; sensor modeling

## 1. Introduction

The agriculture industry plays a significant role in the economies of many countries. For this reason, the technological advancements in the agricultural industry are considered to be a vital contributor to the economy, as well as to sustainable development. The implementation of smart monitoring and automatic condition sensing systems in agricultural fields enable farmers to effectively monitor their land and take necessary actions to maintain productivity. Several new technologies and approaches [1,2] have been studied

and implemented to promote continuous improvement in the agriculture sector. Computer vision-based systems [2] for precision agriculture, hyper-spectral imaging [3], and disease/productivity prediction [4] are some examples. AirSurf, an automated and open-source analytic platform that combines computer vision techniques with deep learning to measure yield-related phenotypes from ultra-large aerial imagery, is presented in [5]. Artificial Intelligence (AI)-based systems are used for the analysis and prediction of the five most produced grains in the world: maize, rice, wheat, soybean, and barley. A systematic review for AI applications in precision agriculture is presented in [6].

Pakistan is an agricultural country, with the majority of the population associated with the agricultural industry directly or indirectly [7]. Despite advancements in the agricultural sector and evolution of the concept of precision agriculture [4], Pakistani farmers still lack access to modern technological advancements in the field and are forced to use conventional and traditional mechanisms for farming. In this paper, we have designed and implemented a Smart Irrigation System (SIS) [8] and deployed it in different agricultural fields in Sindh and other parts of the country. The SIS is a Wireless Sensor Network (WSN) [9], which collects information on temperature, soil moisture and humidity from a particular field and passes this information back to a central server station, which calculates the water required in a particular field and passes this information back to the farmer's smart phone. The farmer then irrigates the crops as communicated by the application. This allows the farmer to save water resources while increasing the crop yield.

One of the major flaws in the current SIS system and most of the WSNs deployed around the world is the sudden failure of physical nodes deployed along with the WSNs. It is sometimes impossible to assess whether the physical nodes are passing on the correct sensor values or are malfunctioning over the course of time. The only way to identify malfunctioning physical nodes is through manual inspection, which is a tedious and cumbersome job. The malfunctioning sensor node also has drastic effects on any real-time system. Providing incorrect values to the farmer from the SIS system will have a very negative impact on the overall reliability and public perception of the system.

A study related to the current work is presented in [10], which focuses on three vital parameters, i.e., the speedy and precise prediction of the network lifetime, internode distance and power level. Since these three constraints are closely associated with each other, by fixing the two parameters, one can predict the optimal value of the third parameter with high accuracy. Moreover, the more tedious and complicated task is to design a WSN (i.e., determining network parameters) whilst employing a heuristic algorithm for the neural network.

Another research study [11] proposed a novel architecture based on a Long Short-Term Memory (LSTM) network. It uses an appropriate training function and introduces a target generation to find suitable ways to explain and deal with the complications related to RUL (Remaining Useful Life) [12] estimations for physical systems. The different sensors, which uninterruptedly record diverse indicators about a working asset such as vibration intensity or applied pressure, gather data. The suggested network uses a bidirectional sequence processing methodology in a sequential fashion, bidirectional handshaking LSTM [11], to continuously monitor the sequential data. Instead of partially processing units or sequences, it passes the final states from forward processing LSTM units to backward processing units up to the time of predictions, to ensure the immediate availability of predictions. The aforementioned research study also proposed that RUL training does not require the hypothesis related to the deterioration of function, as it is based on the calculation of sensor reading to determine the health of the current system.

Neural networks are used in image processing [3], saliency detection [13], biomedical image analysis [14], drug design [15] and weather forecasting applications [16]. The use of neural networks (NN) in environment monitoring for the creation of datasets of atmospheric parameters as well as for weather forecasting is also increasing day by day. Researchers have trained deep neural networks for weather prediction accompanied with the complexity of climate model. With the advancement in computational speed and power,

the availability of massive datasets, and complex deep architectures, consisting of millions of parameters, deep learning is becoming the state of the art in many fields, including agriculture [17].

This study tries to solve this problem by designing a neural network, referred as a neural sensor, complementing a physical sensor. This research also endeavors to provide, with initial feasibility, a study that utilizes a neural network mimicking a physical sensor node. Using such a model may also increase the reliability and efficiency of the current water resource management system, SIS, while reducing the dependency on physical sensors to effectively facilitate farmers in managing the productivity of their agricultural fields.

The main contributions of this research study are:

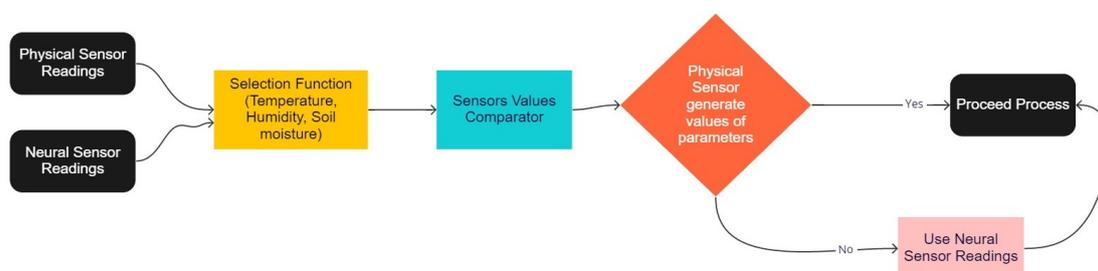
- The development of an LSTM-based smart system that provides readings for irrigation based on the predictive analysis of temperature, soil moisture and humidity.
- The continuous validation of the developed smart system using both physical and neural sensor readings to avoid malfunctions.
- A comprehensive comparative forecasting of temperature, humidity and moisture.

The rest of this paper is organized as follows: In Section 2, the design methodology and data collection are presented, and in Section 3, the results are discussed. Finally, Section 4 concludes the paper with future prospects.

## 2. Materials and Methods

### 2.1. Design Methodology

A sequential model is a linear stack of layers which enables building of the model layer by layer, where each layer poses one input and one output tensor. A linear sequential model [18,19] is used as the development methodology for the neural sensor system. The proposed model is sequential in nature and consists of blocks to process input data and feedback. The complete SIS system data flow is shown in Figure 1. It illustrates diverse stages and events of this model, which depicts the implementation of a neural network to mimic the behavior of the physical sensor node, deployed in the agricultural field. Neural networks are universally employed as predictive models [20]. In the proposed model, the neural network is trained in the supervised mode using actual data values acquired in the previous year by the SIS system deployed in the field. In supervised learning, the actual labels are also fed to the network during training. The data values mainly include temperature and humidity readings. The trained neural network will then be used to predict sensor data in the test mode. Since, there is a very little support available in the existing literature regarding the role of neural networks mimicking the behavior of a physical sensor node, the proposed module is referred to as a neural sensor. The flowchart of the complete system is shown in Figure 1, which also illustrates how the proposed system operates.

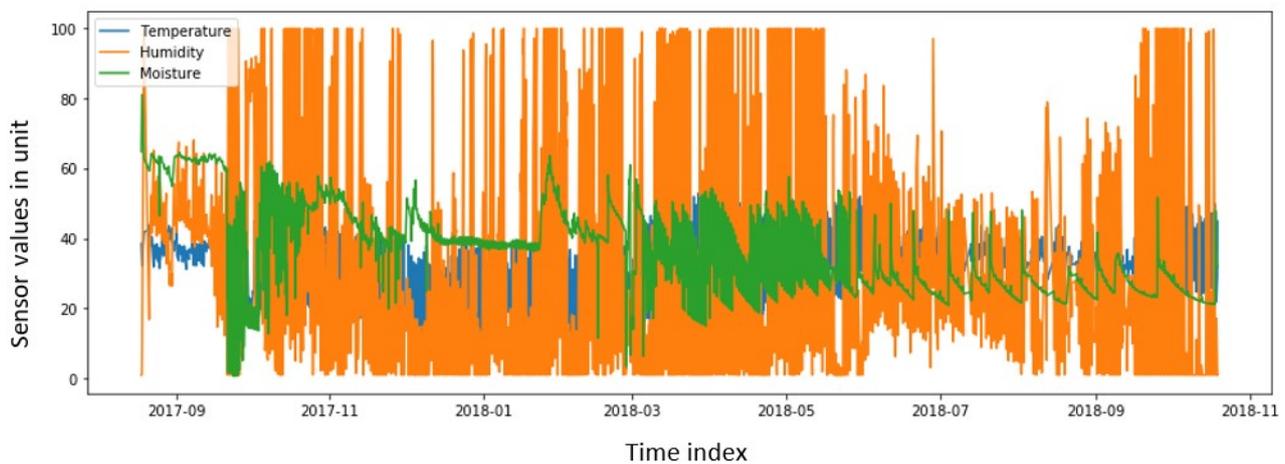


**Figure 1.** The flowchart of the proposed model for the flow of the SIS system in the presence of the proposed neural sensor node. The system is fed with the values from the physical sensor and neural sensor values simultaneously. The comparator module compares both the values. If the physical sensor value deviates from a neural sensor node by a huge margin, then a red flag is raised regarding the physical sensor and the neural sensor value is passed to the SIS system to save it from experiencing a drastic degradation in performance.

## 2.2. Data Collection

This section emphasizes the data gathering and how the raw data were preprocessed by using different methods and techniques to obtain clean and observable data. The resulting data is used in the training of the proposed neural sensor model. Data processing is solely the alteration of raw data to extract meaningful information through a process. For the neural sensor, we used the data of physical sensors embedded in the agricultural field of lemons near Ghadap, Sindh, Pakistan.

Data preprocessing consists of: importing data values, concatenating and sorting them, resampling the data, handling missing values and finally downsampling these values. The targeted variables in the dataset are temperature, humidity and soil moisture. The targeted variables plot is represented in Figure 2, where the orange-colored graph shows the value of humidity against the date-time variable and blue-colored plot represents the temperature, while green represents the soil moisture readings of the physical sensor against a particular date-time variable. The plot shown in Figure 2 consists of the values transmitted from the physical sensor of SIS fitted in the lemon crop field. Since index will not be a part of the prediction variable, the date components were extracted from it to highlight seasonality and time components.



**Figure 2.** Plot of targeted variables (Temperature in °C, humidity (H) and soil moisture).

The data values acquired from the field include temperature, humidity and the soil moisture values. The temperature value is relayed from a temperature sensor LM35 which is a precision integrated circuit (IC) temperature device. LM35 can operate accurately between temperature ranges of  $-50$  degrees centigrade to 150 degrees centigrade, which is a sufficient range for the local temperatures in Pakistan. The humidity values are taken from the humidity sensor DHT-22. A customized in-house sensor has been designed to calculate the values of soil moisture in the agricultural field. Soil moisture works by passing a current between the two probes inserted closely in the soil and then measuring the strength of the output signal at the controller side. A comparator integrated circuit, LM 393, has been used to adjust the sensitivity of the soil moisture sensor. The data from the temperature, humidity and soil moisture sensors are passed to the controller, which transfers these values instantaneously to the cloud. The controller is tuned to transfer sensor values on an hourly basis to the main decision support system via the cloud.

## 2.3. Neural Sensor Modeling

A popular machine learning model used for data processing is a neural network (NN). A neural network is a set of algorithms that attempts to recognize the underlying relationships in a set of data using a method that is similar to how the human brain works. The architecture of an NN is inspired by the functional and structural aspects of a biological

brain. A neuron receives inputs with adjustable weights along the incoming edges. In the most predictive model, the sigmoid or Relu activation function is frequently used as the activation function to model the non-linearity in the data [21].

Recurrent Neural Networks (RNNs) [22] are a robust type of neural network for processing sequence-dependent data as they handle variable length sequences and avoid the need for fixed-sized time windows. These are densely connected NNs with feedback loops to connect the hidden neurons across time. A state vector in the hidden units maintains the memory of all the previous elements of the sequence. Due to this internal memory, RNNs can learn the long-term dependencies across many time steps and generalize across input sequences rather than individual patterns. It is well suited for time-series analysis, forecasting, and many other areas of natural language processing [23]. It enables precise future predictions based on the patterns of time series data.

A precise Recurrent Neural Network (RNN) architecture is demonstrated in Figure 3, which is extended into a deep neural network to illustrate how a state is constructed with time. It takes an existing sequence element at time step  $t$  and the hidden state from the preceding time step at  $-1$ . Now, the hidden state (memory) updated to and the network output is determined. “ $U$ ” is the weight matrix that maps the input to the hidden layer and “ $W$ ” is the weight matrix for the recurrent transition among one hidden state to the next. The hidden layer is propagated to the output layer through the weight matrix “ $V$ ”. Hence, the recent output is reliant on all the prior inputs (for  $t' \leq t$ ). Mathematically, it can be expressed as:

$$h(t) = f(h_{t-1}, x_t) \quad (1)$$

where  $h(t)$  is the current state,  $x_t$  is the input, and  $h_{t-1}$  is the previous state. After that, the tanh activation function is applied.

$$h_t = \tanh(W_{hh} * h_{t-1} + W_{xh} * x_t) \quad (2)$$

where  $W_{hh}$  is the weight matrix multiplied by the hidden state and  $W_{xh}$  is the hidden matrix multiplied to the input sequence. Since, our data is sequential in nature, such a model is well suited to our problem.

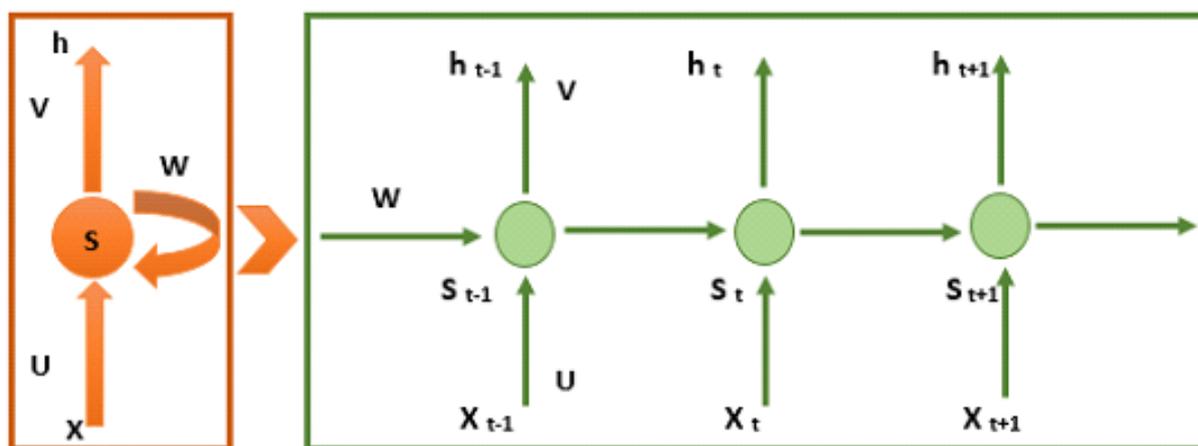


Figure 3. RNN architecture.

LSTM networks are a specific RNN architecture that can maintain long-term dependencies. The core idea behind this architecture is a memory cell, which maintains the temporal state of the network over several time steps (over 1000), and non-linear analog gating units, which controls the information flow by selectively storing and deleting in-

formation in this cell. Initially, the basic LSTM components are the input gate, forget gate, output gate and hidden layer, i.e., the Constant Error Carousel (CEC).

$$output(t) = \sigma(w_o[h_{t-1}, x_t] + b_o) \quad (3)$$

The output gate controls what information needs to be released from the CEC and thus ensures that other units are not disturbed by the stored information. This architecture, however, led to instability in the network in the presence of a continuous input stream without any marked start and end points, as it did not reset the cell contents after finishing the processing of a sequence or before it started a new one. To deal with this issue, forget gates are used, which enable the LSTM cell to reset or forget the cell's memory at appropriate times.

A typical LSTM architecture with forget gates and peephole connections, with LSTM RNNs, is shown in Figure 3. The training of the network takes a shorter time and is of high accuracy compared to conventional RNNs. It prevents the error from back-propagating through time and makes it feasible for tasks with high complexity and temporal extent to learn over multiple time steps, thereby allowing the model to predict future data points based on long-term context dependencies without any vanishing gradient problem.

#### 2.4. Training of the Neural Network

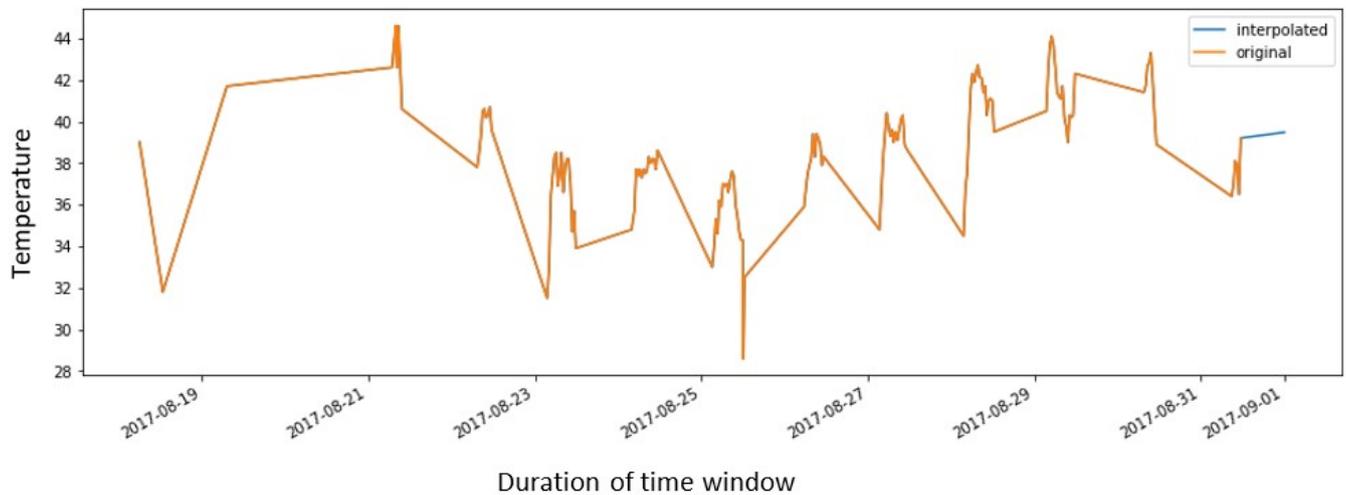
This section deals with the training of a neural sensor with the help of a recurrent neural network (LSTM). Since our data is sequential in nature with long-term dependencies, we decided to choose the Long Short-Term Memory (LSTM) architecture. Before applying the modeling technique, shift steps should be identified. Shift steps define how many steps into the future one wants to predict with each input to the network. It shifts the data backwards by the number of steps, and as one wants to predict into the future, the input and output dimensions remain the same. In the neural sensor, the shift steps number twenty-four for the input of each date, and it will generate twenty-four records of the predicted values of temperature, humidity and soil moisture on the hour against each date. These shifts have created an empty space in the tail of the target dataset equal to the number of shift steps. A sample dataset is shown in Table 1.

**Table 1.** Data captured from multiple fields at time steps. Physical sensors were used to obtain the values.

	Created at	Field1 Temperature	Field2 Heat Index	Field3 Humidity	Field4 Average Flow Rate	Field5 Soil Moisture	Field6 Flow Reading (Quantity)
0	2017-08-18 @6:07:22	39.0	34.08	1.0	0	63.62	0.0
1	2017-08-18 @12:48:44	31.8	28.59	1.0	0	81.70	0.0
2	2017-08-19 @07:15:31	41.7	121.97	99.9	0	62.88	0.0
3	2017-08-21 @06:51:22	42.6	42.83	19.0	0	59.46	0.0
4	2017-08-21 @07:22:22	43.5	44.32	19.2	0	59.71	0.0

The bottom half from both the original and target datasets will be truncated, which correspond to the empty space. The number of rows remains the same, and we save them in x and y data where x corresponds to the input port of the neural net and y corresponds to the output port. Note that the output y data is ahead of the input x data in terms of

number of shift steps. Interpolation was used to handle the missing values, as shown in Figure 4.

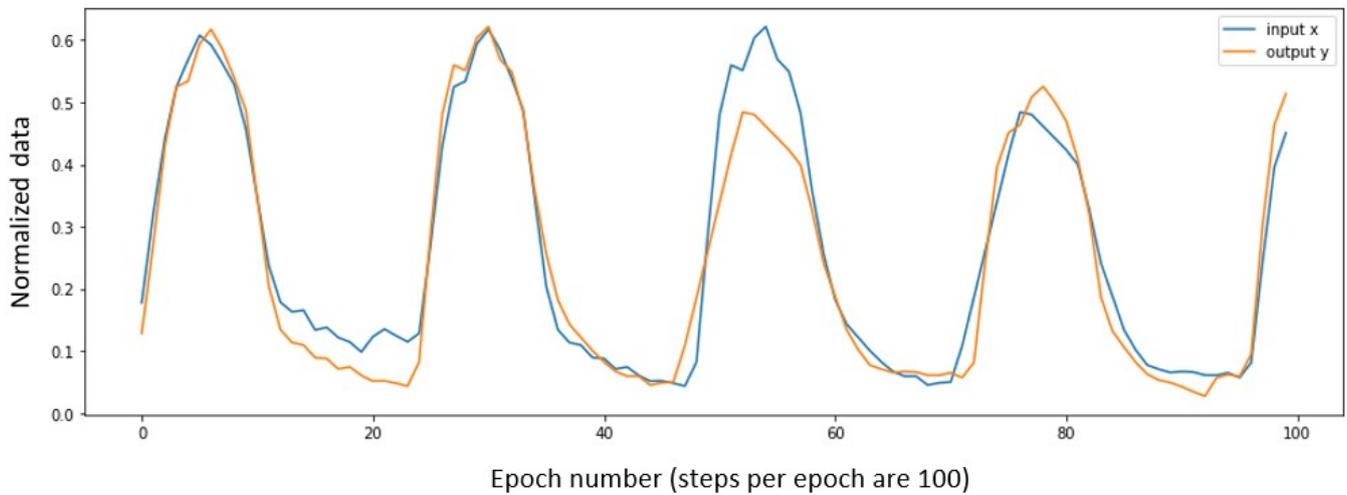


**Figure 4.** Plot of data after interpolation.

To train the neural network, the dataset was splitted into a training dataset and testing dataset. The physical sensor readings for temperature, humidity and soil moisture were used as an input, and the neural network output was the future predicted values for temperature, humidity and soil moisture. We chose the common criteria of 80–20% split. In total, 80% of the data were used for training and 20% of the data were used for testing purposes. This allowed us to obtain a reasonable number of samples to train our models, and out of 10,223 records, 8178 records were assigned for training and 2045 were allocated for testing. We did not prepare the validation sets. Validations sets are used for hyperparameter tuning [24]. We tuned our parameters during the training phase and they also performed well during the testing phase. Root mean square propagation was used as an optimizer with a learning rate of 0.01. We divided the data into twenty batches.

After the collection of the dataset, the first step was normalization. The normalization of data is the removal of repetition and redundancy in data. The data were normalized using a min-max normalization method. The training data is made up of almost eight thousand observations. Instead of training the RNN on the complete sequences of observations, we used a batch of shorter sub-sequences picked at random from the training data. Batch size is the number of training examples utilized in one epoch or iteration. Sequence length represents the number of records in one batch. Lowering the batch size will reduce the size of the data loaded into memory. Lowering the sequence size too much will make the signal history too small to make predictions. For training the model, we chose 20 batch size sets, while the sequence length was set to  $24 \times 7 \times 8$ , i.e., two-month window used for training.

In Figure 5, the input (blue curve) and output (orange curve) readings in the graph are shown after setting the step size for model training. The shift steps being set to 24 represents that, as a result of training of model, 24 predicted values will be obtained against each input date.



**Figure 5.** Plot of dataset after shift stepping.

The neural network trains quickly by running multiple training epochs. However, there is a chance of overfitting the model to the training set, leaving it incapable of oversimplifying the unknown data. Therefore, it is necessary to screen how well the model works on the test data after every epoch and only include the model's weights if the functioning is enhanced on the test data. The batch generator arbitrarily chooses a batch of small sequences from the training data values and employs it during training. For the validation data value, it is desired to instead route through the complete structure from the test data and measure the prediction correctness on that complete structure. The architecture and number of Training parameters is shown in Table 2.

**Table 2.** Neural network architecture and number of parameters.

Layer (Type)	Output Shape	Number of Parameters
LSTM	(None, None, 100)	42,800
DENSE	(None, None, 202)	202

To train the neural network, note that a single “epoch” does not correspond to a single processing of the training set because of how the batch-generator randomly selects sub-sequences from the training set. Instead, different researchers have selected steps per epoch in order to process each epoch in a small amount of time. The training of an RNN consists of learning new weight values per epoch. For the modeling of the neural sensor, the defined epochs are 20 and steps per epochs are 100.

### 3. Results and Discussion

Some conventional testing procedures can be used to evaluate the model's performance on the test set. This involves examining the model's anticipated values at various time intervals on the same day. This is a reliable method of demonstrating the neural sensor's performance. From 1900 to 2200, the prediction is made on an hourly basis.

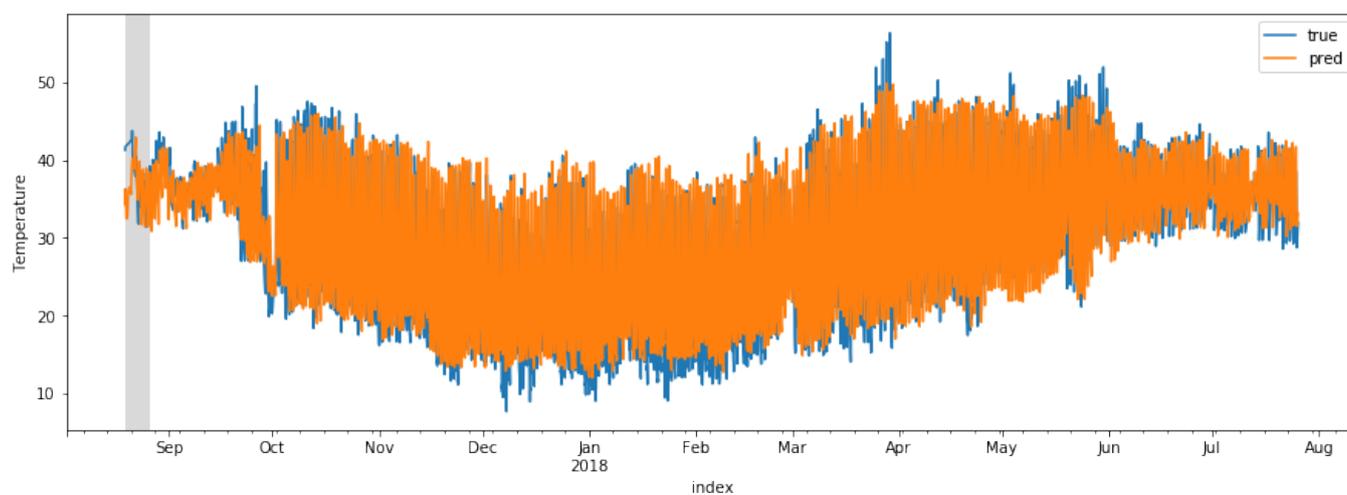
The results of prediction on the training dataset are shown in Table 3.

The plot of the predicted output signals is important to understand the workings of the neural sensor in predicting different sensor values. These plots only show the output signals. The time shift between the input and output is fixed. The model foresees the output signals, for example, twenty-four hours into the future (as defined in the shift-steps variable above). Therefore, the x-axis of the plot displays how many time steps of the input-signals have been grasped by the extrapolative model so far.

**Table 3.** Predicted values by neural sensor during training.

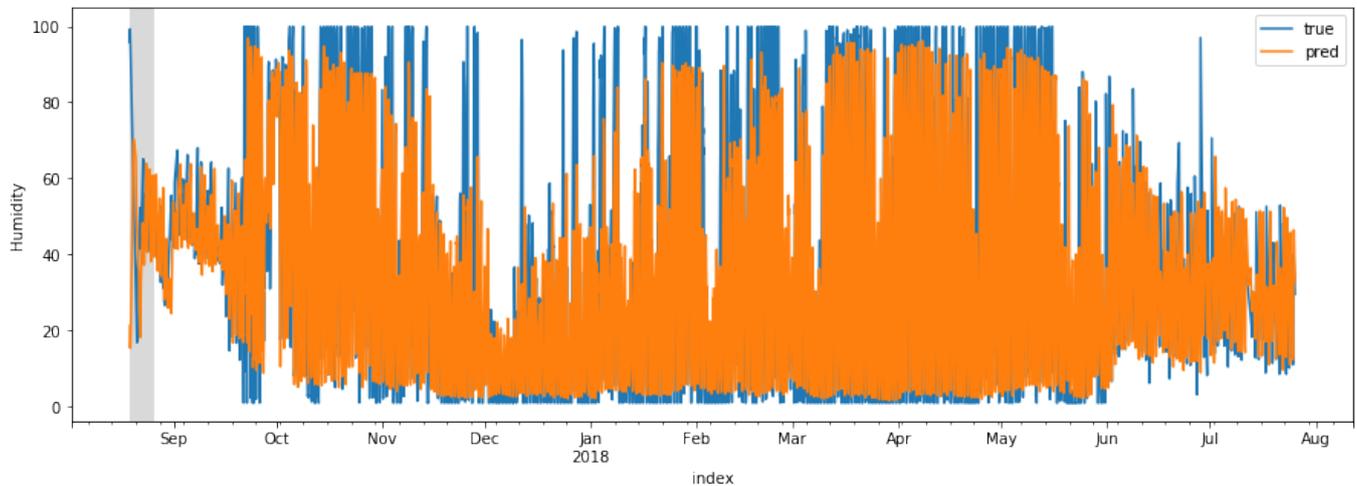
Index	Predicted Humidity	Predicted Moisture	Predicted Temperature
2018-07-25, 19:00:00	32.45	38.24	27.45
2018-07-25, 20:00:00	32.64	36.67	27.45
2018-07-25, 21:00:00	32.79	35.12	27.36
2018-07-25, 22:00:00	32.90	33.69	27.25

The prediction is not very accurate for the first 30 to 50 time steps because initially the model has fewer input data at this point. The model generates a single time step of output data for each time step of the input data. The model requires an adequate window of time by processing perhaps 30–50 time steps before it starts generating usable output signals. The initial 50 time steps were ignored, while the mean-squared error in the loss function was calculated, referred to here as the warm-up period. The warm-up period is shown in a grey box in Figure 6. The blue-colored plot represents the physical sensor values of temperature while the orange-colored plot represents the predicted values of temperature by the neural sensor in Figure 6. The predicted values are within the acceptable range, except a few spikes between the months of April and January. The created neural sensor, in particular, has shown a high level of overlap with physical sensors utilized from October to April and May to August. This means that the neural sensor learns the patterns of the physical sensor and is able to generate precise predictions.

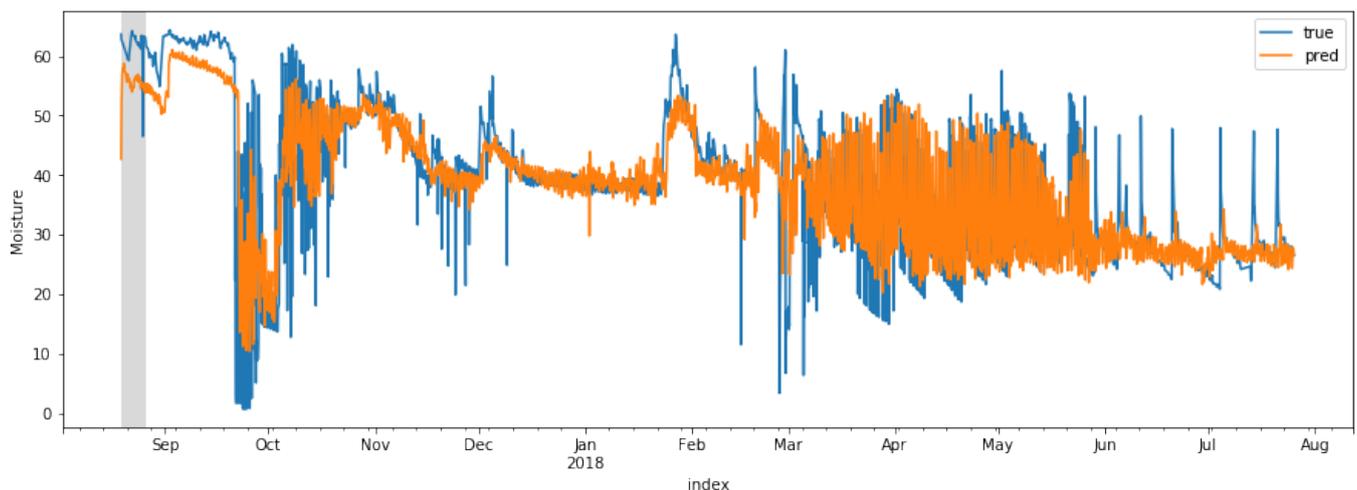
**Figure 6.** Plot of true (blue curve) and predicted (orange curve) readings of temperature.

Figures 7 and 8 show the true and predicted values for humidity and soil moisture, respectively. The performance of the neural sensor well mimics the physical sensor values for humidity compared to soil moisture values throughout the year, despite the fact that the number of spikes in moisture values is slightly greater than that which follows the pattern in the temperature and humidity readings in the test dataset. It was observed that the temperature and humidity predictions successfully overlap with the ground truth dataset, while the moisture value predictions are within an acceptable range for use in agriculture. It should also be noted that all predictions for temperature, humidity and moisture are sensitive to the small changes in values compared to the true values. For instance, the moisture values changed from December to February, as shown in Figure 8, and are quite small compared to the rest of the ground truth dataset. The observed value differences are minor, as illustrated in Figure 8, and the suggested neural sensor is capable of measuring the proper moisture values nearly and successfully mimicking the actual values. This is further confirmed in Figure 6, which shows the temperature readings from September to

October, and Figure 7, which shows humidity readings from September to October and July to August. This demonstrates the neural sensor's robustness and sensitivity presented in this study.



**Figure 7.** Plot of true (blue curve) and predicted (orange curve) readings of humidity.



**Figure 8.** Plot of true (blue curve) and predicted (orange curve) readings of moisture.

For modeling a neural sensor, a forecasting function is designed which take date as an object to forecast temperature, humidity and soil moisture against that particular date and time, the essential parameter for object (date) is time delta. Time deltas are differences in times. After training of LSTM network, the forecast values of temperature, humidity and soil moisture by neural sensor are shown in Figures 9–11, respectively. The blue-colored graph shows the value of the physical sensor embedded in the field and orange curve in the graph is the plot of predicted value of neural sensor that is obtained after the training of the LSTM model. Figures 9–11 further confirms the generalization of the proposed neural sensor model and illustrates its potential to be used to complement the physical sensors.

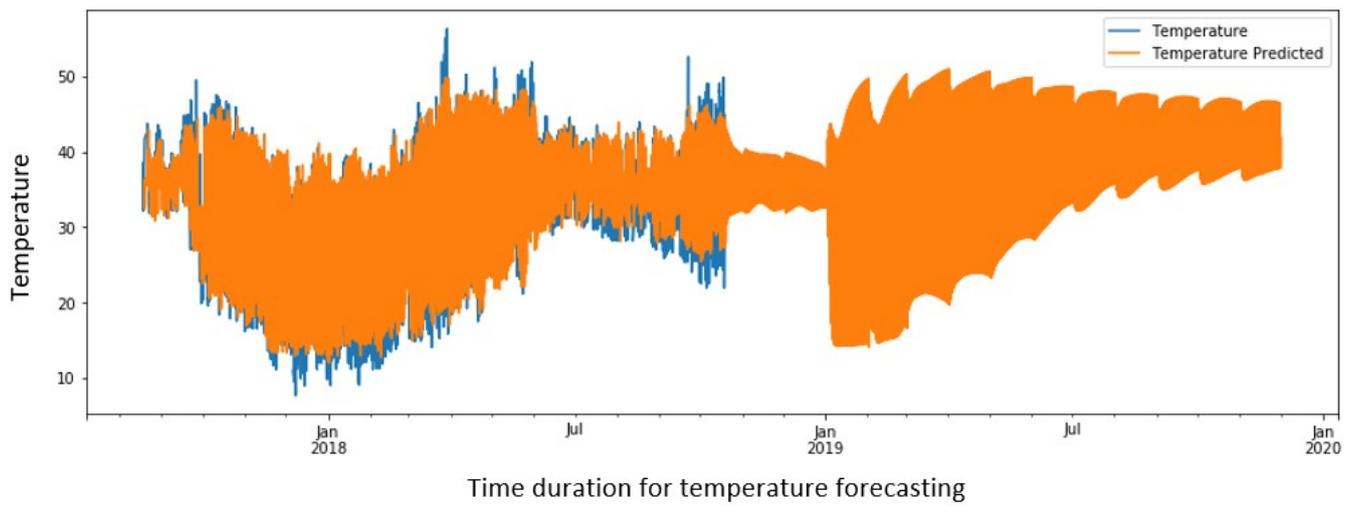


Figure 9. Forecasted readings of temperature.

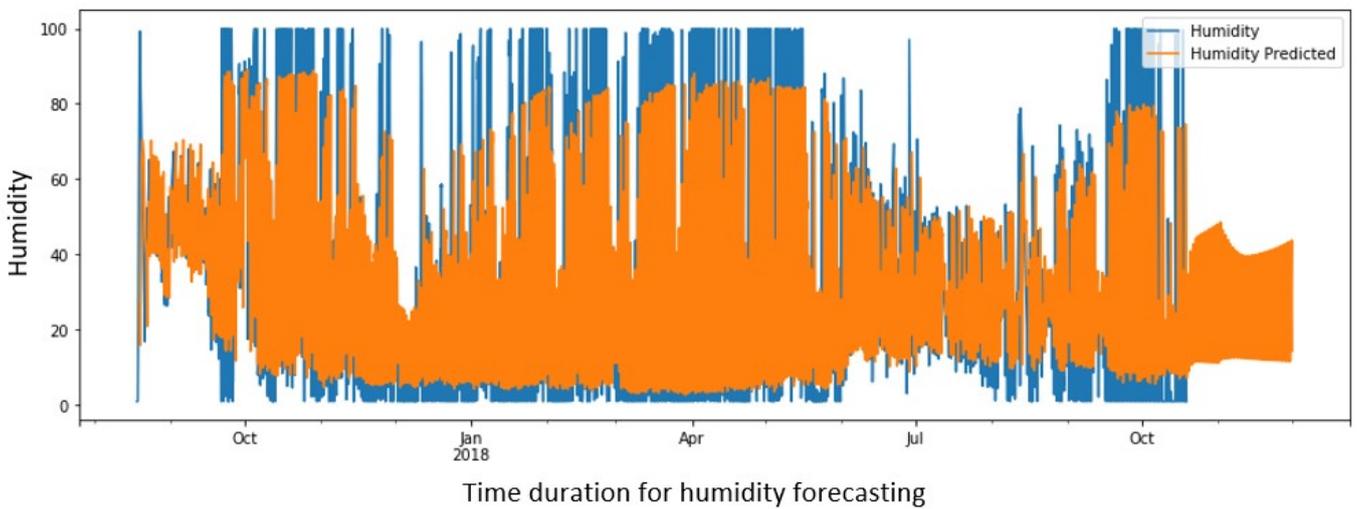


Figure 10. Forecasted readings of humidity.

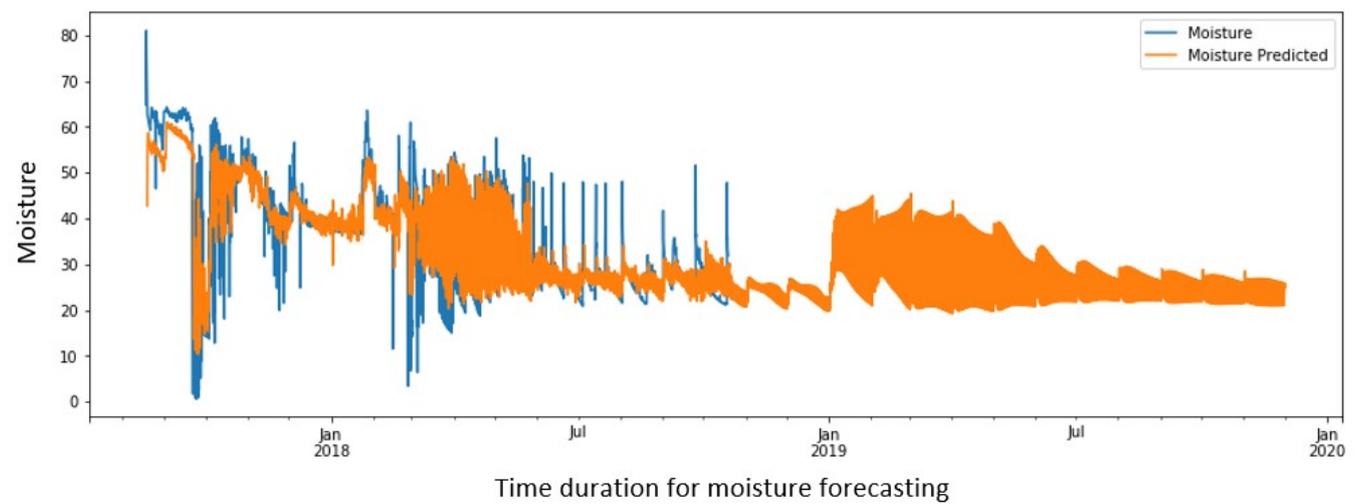


Figure 11. Forecasted readings of moisture.

Table 4 presents a statistical analysis where the mean and standard deviation of physical and neural sensors are compared. The mean and standard deviation of physical and neural sensors are very close and show good agreement. Regarding humidity, there is a large standard deviation due to sudden spikes and fluctuation in physical sensor data. The neural sensor does not capture this well. The mean value for physical sensor data is 35.920 and for neural sensor is 30.510. This is due to the sudden rise and fall in physical sensor data values. However, the neural sensor learns after some time and generates reasonable predictions for humidity. For the case of moisture, the behavior of the neural sensor is similar to humidity. We can conclude that the predictions are precise and accurate for temperature and less precise for humidity and moisture.

**Table 4.** Mean and Standard deviation of physical sensor and neural sensor values.

Physical Sensor	Mean Value	Standard Deviation	Neural Sensor Prediction	Mean Value	Standard Deviation
Temperature	30.386	8.9510	Temperature	31.761	8.736
Humidity	35.920	30.385	Humidity	30.510	22.865
Soil moisture	36.746	12.576	Soil moisture	32.415	8.907

Table 5 presents a comparison with the related studies. Despite the fact that our work is novel and has not been developed previously in the field of agriculture, we compared the accuracy of our results to other types of neural sensors that mimic the physical sensor behavior for various applications. For instance, for effective sensor array reduction in a gas mixtures analysis system, a neural network sensitivity analysis method was used in [25], by improving a basic engine parameter, Artificial Neural Network (ANN)-based virtual sensors were employed to manage exhaust emissions and physical sensors can be eliminated during vehicle tuning in [26], and a convolutional bi-directional LSTM-based approach was proposed in [27] for a tool wear prediction task as a part of machine health monitoring. Although the applications of neural network models varies, a common goal is to develop an ANN-based sensor to replace physical sensors in a variety of industrial applications. This comparison is conducted based on the accuracy of the models developed in the literature. The accuracy is measured using root mean square values (RMSE). We averaged the RMSE values of temperature, humidity and moisture. The proposed method achieves the best performance within the agriculture sector, compared to other models that are developed for various applications, in terms of accuracy. This research can also be used as a guide for other disciplines looking for alternative solutions, such as medical microrobotics, where physical sensors are difficult to integrate into the system for various measurements such as 3D vibration measurements [28] and force sensing [29] for biological cell injection.

**Table 5.** Comparison with other state-of-the-art methods.

Reference	Algorithm	Application	Accuracy
[25]	Feed Forward Neural Network	Sensitivity analysis for sensor array optimization	Not reported
[26]	Neural Network Virtual Sensors	Hydrogen Vehicle Tuning	RMSE 4.1%
[27]	Convolutional Bi-Directional LSTM	Tool wear prediction tasks	RMSE 7.1%
Proposed	LSTM-based Neural sensor	Smart Irrigation	RMSE 2.35%

The requirement for large datasets to be used as input during the training procedure is a significant limitation of deep learning-based neural sensors that is developed in our study. Furthermore, there are few publicly available datasets for agricultural researchers to work with, so most must create their own datasets, which takes time.

#### 4. Conclusions

The focus of this study is to present a model that addresses the problem of physical sensor nodes malfunctioning by informing the system regarding that particular node. The presented model achieves this by utilizing an alternate mechanism that provides the predictive values of temperature, humidity and soil moisture to avoid instances of complete system failure. The proposed model increased the overall reliability and efficiency of the existing system by providing additional information regarding the working of the physical sensor in the agricultural context. The proposed neural sensor in this paper is successfully able to predict key values related to the SIS system such as temperature, humidity and soil moisture, and is thus able to inform us if the physical sensors embedded in the agricultural field in a real-time environment have malfunctioned.

It also provided a set of predicted values of desired variables for a particular instance on an hourly basis. The current model further provides a mechanism to save the system from complete failure. It helps to improve the mean time to failure of the system and the system performance gradually becomes worse in the case of a malfunctioning physical sensor.

The future possibilities of this research are in the integration of this model with the decision support system of SIS, which decides the supply of water in the field by calculating the amount of water already in the soil with the help of transpiration parameters such as temperature, humidity and soil moisture. This model can be utilized in a number of real-time systems, relying heavily on physical sensor values to avoid any catastrophic event due to the failure/malfunctioning of physical sensor nodes, and to ensure the reliable and safe operation of the system.

**Author Contributions:** Implementation of the model, methodology, writing, review, and editing, M.S.; conceptualization and editing, S.Q.K.; conceptualization and resources, M.K.; review and editing, R.A., M.U.F. and S.A.; conceptualization, writing and review, R.Q. and F.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** The datasets and codes are available from the first author upon request.

**Acknowledgments:** The authors would like to thank Smart City Lab, National Center for Artificial Intelligence, Computer and Information Systems Department, NED University of Engineering and Technology, Karachi, Pakistan.

**Conflicts of Interest:** The authors declare no conflict of interest.

#### References

1. Byerlee, D.; De Janvry, A.; Sadoulet, E. Agriculture for development: Toward a new paradigm. *Annu. Rev. Resour. Econ.* **2009**, *1*, 15–31. [[CrossRef](#)]
2. Seng, K.P.; Ang, L.M.; Schmidtke, L.M.; Rogiers, S.Y. Computer vision and machine learning for viticulture technology. *IEEE Access* **2018**, *6*, 67494–67510. [[CrossRef](#)]
3. Qureshi, R.; Uzair, M.; Khurshid, K.; Yan, H. Hyperspectral document image processing: Applications, challenges and future prospects. *Pattern Recognit.* **2019**, *90*, 12–22. [[CrossRef](#)]
4. Gebbers, R.; Adamchuk, V.I. Precision agriculture and food security. *Science* **2010**, *327*, 828–831. [[CrossRef](#)]
5. Bauer, A.; Bostrom, A.G.; Ball, J.; Applegate, C.; Cheng, T.; Laycock, S.; Rojas, S.M.; Kirwan, J.; Zhou, J. Combining computer vision and deep learning to enable ultra-scale aerial phenotyping and precision agriculture: A case study of lettuce production. *Hortic. Res.* **2019**, *6*, 70. [[CrossRef](#)]
6. Patrício, D.I.; Rieder, R. Computer vision and artificial intelligence in precision agriculture for grain crops: A systematic review. *Comput. Electron. Agric.* **2018**, *153*, 69–81. [[CrossRef](#)]
7. Azam, A.; Shafique, M. Agriculture in Pakistan and its Impact on Economy. *Rev. Inter. J. Adv. Sci. Technol.* **2017**, *103*, 47–60. [[CrossRef](#)]
8. Darshna, S.; Sangavi, T.; Mohan, S.; Soundharya, A.; Desikan, S. Smart irrigation system. *IOSR J. Electron. Commun. Eng. (IOSR-JECE)* **2015**, *10*, 32–36.
9. Ullah, U.; Shahid, A.R.; Irfan, M.; Qadir, J.; Nawaz, M.; Qureshi, R. A stable and reliable short-path routing scheme for efficient acoustic wireless sensor networks (AWSNs). *IEEE Access* **2019**, *8*, 1458–1474. [[CrossRef](#)]

10. Akbas, A.; Yildiz, H.U.; Ozbayoglu, A.M.; Tavli, B. Neural network based instant parameter prediction for wireless sensor network optimization models. *Wirel. Netw.* **2019**, *25*, 3405–3418. [[CrossRef](#)]
11. Elsheikh, A.; Yacout, S.; Ouali, M.S. Bidirectional handshaking LSTM for remaining useful life prediction. *Neurocomputing* **2019**, *323*, 148–156. [[CrossRef](#)]
12. Lei, Y.; Li, N.; Guo, L.; Li, N.; Yan, T.; Lin, J. Machinery health prognostics: A systematic review from data acquisition to RUL prediction. *Mech. Syst. Signal Process.* **2018**, *104*, 799–834. [[CrossRef](#)]
13. Nawaz, M.; Khan, S.; Cao, J.; Qureshi, R.; Yan, H. Saliency detection by using blended membership maps of fast fuzzy-C-mean clustering. In Proceedings of the Eleventh International Conference on Machine Vision (ICMV 2018), Munich, Germany, 1–3 November 2018; International Society for Optics and Photonics: Bellingham, WA, USA, 2019; Volume 11041, p. 1104123.
14. Zhang, P.; Zhong, Y.; Deng, Y.; Tang, X.; Li, X. A survey on deep learning of small sample in biomedical image analysis. *arXiv* **2019**, arXiv:1908.00473.
15. Jing, Y.; Bian, Y.; Hu, Z.; Wang, L.; Xie, X.Q.S. Deep learning for drug design: An artificial intelligence paradigm for drug discovery in the big data era. *AAPS J.* **2018**, *20*, 58. [[CrossRef](#)]
16. Scher, S.; Messori, G. Weather and climate forecasting with neural networks: Using general circulation models (GCMs) with different complexity as a study ground. *Geosci. Model Dev.* **2019**, *12*, 2797–2809. [[CrossRef](#)]
17. Kamilaris, A.; Prenafeta-Boldú, F.X. Deep learning in agriculture: A survey. *Comput. Electron. Agric.* **2018**, *147*, 70–90. [[CrossRef](#)]
18. Warner, B.; Misra, M. Understanding neural networks as statistical tools. *Am. Stat.* **1996**, *50*, 284–293.
19. Hutter, F.; Hoos, H.H.; Leyton-Brown, K. Sequential model-based optimization for general algorithm configuration. In *Learning and Intelligent Optimization*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 507–523.
20. Åkesson, B.M.; Toivonen, H.T. A neural network model predictive controller. *J. Process Control* **2006**, *16*, 937–946. [[CrossRef](#)]
21. Zadeh, M.R.; Amin, S.; Khalili, D.; Singh, V.P. Daily outflow prediction by multi layer perceptron with logistic sigmoid and tangent sigmoid activation functions. *Water Resour. Manag.* **2010**, *24*, 2673–2688. [[CrossRef](#)]
22. Zaremba, W.; Sutskever, I.; Vinyals, O. Recurrent neural network regularization. *arXiv* **2014**, arXiv:1409.2329.
23. Palangi, H.; Deng, L.; Shen, Y.; Gao, J.; He, X.; Chen, J.; Song, X.; Ward, R. Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2016**, *24*, 694–707. [[CrossRef](#)]
24. Bergmeir, C.; Benítez, J.M. On the use of cross-validation for time series predictor evaluation. *Inf. Sci.* **2012**, *191*, 192–213. [[CrossRef](#)]
25. Szczówka, P.; Szczurek, A.; Licznarski, B. On reliability of neural network sensitivity analysis applied for sensor array optimization. *Sens. Actuators B Chem.* **2011**, *157*, 298–303. [[CrossRef](#)]
26. Yap, W.K.; Ho, T.; Karri, V. Exhaust emissions control and engine parameters optimization using artificial neural network virtual sensors for a hydrogen-powered vehicle. *Int. J. Hydrog. Energy* **2012**, *37*, 8704–8715. [[CrossRef](#)]
27. Zhao, R.; Yan, R.; Wang, J.; Mao, K. Learning to monitor machine health with convolutional bi-directional LSTM networks. *Sensors* **2017**, *17*, 273. [[CrossRef](#)] [[PubMed](#)]
28. Sadak, F.; Saadat, M.; Hajiyavand, A.M. Vision-based sensor for three-dimensional vibrational motion detection in biological cell injection. *Sensors* **2019**, *19*, 5074. [[CrossRef](#)] [[PubMed](#)]
29. Karimirad, F.; Chauhan, S.; Shirinzadeh, B. Vision-based force measurement using neural networks for biological cell microinjection. *J. Biomech.* **2014**, *47*, 1157–1163. [[CrossRef](#)]