Received: October 2024, Accepted: December 2024, Published: December 2024
Digital Object Identifier: *https://doi.org/10.34302/CJEE/XLCI7386*

DESIGN OF A VIRTUAL TEMPERATURE SENSOR WITH DATA DISPLAY IN A WEB INTERFACE

Paul **NISTOR**, Ioan **ORHA**
Electric, Electronic and Computer Engineering Department, Technical University of Cluj-Napoca, Romania
nistorpaul97@gmail.com, ioan.orha@ieec.utcluj.ro

Keywords: sensors, machine learning, web interfaces

Abstract: *The project presents the design and implementation of a virtual temperature sensor with the display of the data provided by it in a web interface. The objective of the whole project is based on the development of a virtual sensor that estimates the temperature with the help of historical but also current data provided by other sensors. The goal of this project is to reduce the cost of temperature monitoring systems by reducing the physical sensors required. It is also aimed at the possibility of implementing these sensors in inaccessible places. Following the use of a machine learning type system, I want to implement a predictive model, which, based on the data provided by the other physical sensors, generates the response of the virtual sensor. Both the data provided by the virtual sensor and the data provided by the physical sensors will be displayed within the web interface. The project aims to develop a sensor network for temperature monitoring in various applications. Also, by implementing this system, I want to obtain a web interface that allows viewing and managing the measured temperature data.*

1. INTRODUCTION

In our century, data monitoring and management has become a matter of major importance in multiple fields. Among them we can list industry, agriculture and the research environment itself. These fields and many others need this data in order to function optimally but also for continuous development [1].

But in many cases the collection of this data would not be possible without the help of sensors. Among them the temperature sensor stands out, it plays a crucial role in the collection of data aimed at ensuring optimal equipment performance, product protection, process optimization and user comfort. Therefore, temperature is a parameter that must be constantly monitored in multiple fields [2].

As a result of this need to measure temperature, several types of sensors with various properties and characteristics have been developed. We can divide these sensors into two categories. The first category being contact sensors, which measure the temperature by direct contact with the object. The second category is non-contact sensors, which measure the temperature through the heat radiation emitted by the object, without making direct contact with it [3].

Although sensors have constantly evolved and developed, installing an adequate number of sensors to obtain a quality measurement can cause various problems. Therefore adding additional sensors is limited by factors such as budget, available space and accessibility. For example, in an industrial facility with an extensive area, adding additional temperature sensors to expand the monitoring network can involve high costs and increased complexity. In research laboratories, where space is often constrained, adding additional sensors may not be the most efficient solution [4].

In order to solve these problems, we found that replacing a set of physical sensors with virtual sensors within a sensor network is an efficient and pragmatic solution. Thus, by means of virtual sensors costs can be reduced, they facilitate the monitoring of places inaccessible to physical sensors, they save space and solving problems in case of failure is easier to solve than in the case of physical sensors.

This project aims to develop a virtual temperature sensor that, with the help of machine learning algorithms, will estimate the temperature based on data received from nearby physical sensors. The project also aims to integrate this virtual sensor within a web interface to facilitate access to the data provided by the sensors in real time. Therefore, the implementation of this system based on virtual sensors, in addition to solving the previously described problems, also improves the quality of measurements by integrating and analyzing data from multiple sources.

2. SYSTEM DESCRIPTION

2.1. Description of Functional Blocks

The presented system consists of two parts, a hardware part and a software part. The hardware part consists of the temperature sensors and the ESP-32 microcontroller. The software part includes storing the data in a database, the machine learning algorithm that predicts the response for the virtual sensor and displaying the data in a web interface. The block diagram of the whole system is shown in *fig. 1.*

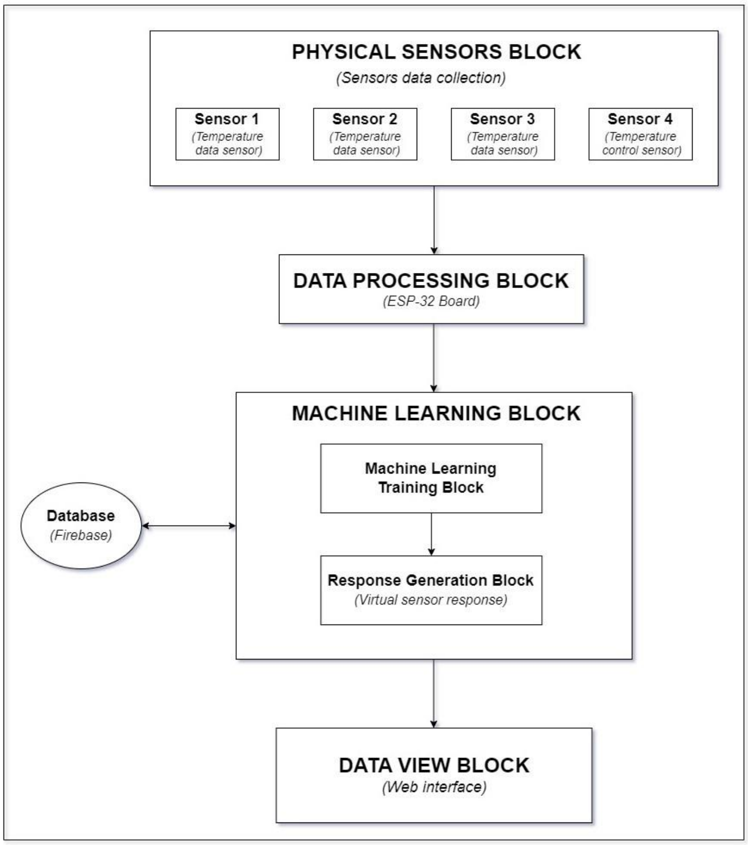

Fig. 1. System structure

The first block encountered in the presented scheme comprises the physical sensors, this block contains four temperature sensors. Although four sensors are shown, only the data from the first three will be used in the machine learning model. The fourth sensor, as the name suggests is a control sensor, it is located in the same location as the virtual sensor and aims to measure the actual temperature in that area. The value provided by this sensor will be used to compare the response generated by machine learning with the actual temperature value. The number of sensors present within this block may vary depending on the needs of the application. For example, in a real application, the control sensor can be removed, and other physical sensors can be added as needed to increase the accuracy of the virtual sensor response.

The second block is represented by the ESP-32 board, it plays a crucial role in the system, because as the name of the block suggests, it deals with data processing. Specifically, this block collects the data from the temperature sensors, filters the noise, normalizes the values, and prepares this data for transmission to the next block.

The next stage within the presented scheme is carried out by the machine learning block, which represents the most important component of the entire system. It contains two essential blocks.

The machine learning training block, in this the data is used to train the machine learning model. In this stage of the system, historical data stored in the database and new data received from sensors via the ESP-32 board are analyzed. The model is built and updated based on this data with the aim of improving itself over time in such a way as to generate the most accurate estimate for the virtual sensor. Essentially, constantly adding new data to the existing data set and retraining the model leads to improved response accuracy.

The response generation block, once the training of the machine learning model is finished, predictions can be made based on the current data, and the obtained value representing the response of the virtual sensor. In this block the virtual sensor value prediction is actually performed.

Within the scheme presented, the database block can also be observed, it represents an important component for the best and most accurate functioning of machine learning. The Firebase database stores all the data generated by the physical sensors but also all the predictions of the virtual sensor. With the help of this block, the system constantly stores new data for continuous training of the machine learning model.

The last block in the scheme is the data visualization block, in this block the data is displayed in a web interface. Through this interface data can be monitored in real time and easily. The processed data from the sensors and the responses generated by the virtual sensor are visualized in the form of a table in this interface.

In conclusion, the presented block diagram describes a well-structured system that combines the hardware part with the software part in order to obtain and visualize the most accurate data, both from the physical sensors and from the virtual sensor.

2.2. Basis of Machine Learning - Polynomial Regression

Polynomial regression is an extension of linear regression, it is used to model more complex relationships between variables, especially when the dependent variable and the independent variables have a non-linear relationship [5].

Within the sensor system, we considered the use of polynomial regression an optimal solution for obtaining the values of the virtual sensor based on the data collected from the other physical sensors. The model underlying machine learning is capable of processing non-linear variations from real data, thus making the chosen model ideal for modeling sensor data that does not have a strictly linear relationship.

The working principle of polynomial regression is to try to minimize the difference between the actual values and the prediction values in the data set. This modeling uses a curve that can have several changes in direction in order to fit the data as well as possible to obtain the most accurate temperature prediction. In *fig. 2* you can see the polynomial regression graph for predicting the temperature.

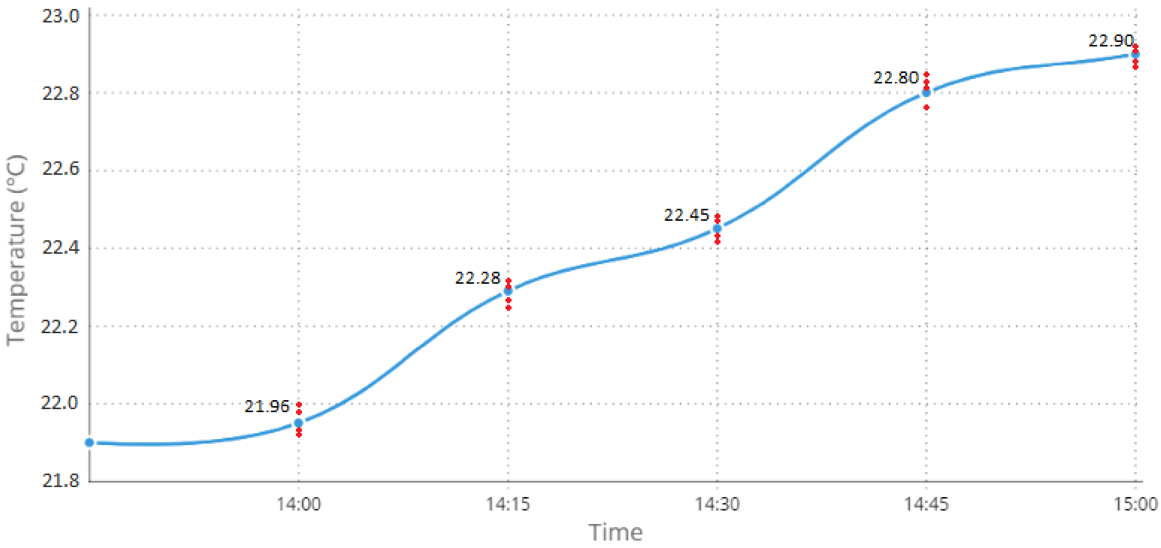

Fig. 2. Polynomial regression graph for temperature prediction

In the graph presented, it can be seen how the polynomial regression curve representing the response of the virtual sensor is generated based on the four values from the physical sensors.

In the previous paragraphs, the operating principle of the model that is the basis of machine learning was presented in a theoretical way. In the following we will present how polynomial regression works in machine learning code.

Within the code, the historical data from the four physical sensors are combined into a two-dimensional matrix, where each row represents the sensor measurements at a particular time. And in order to provide the prediction, the output is calculated as an average of the four sensors, which gives us an approximate value for the temperature of the virtual sensor. The described process represents the training of the machine learning model with the historical data and the code sequence responsible for this step is shown in *fig. 3*.

```
function trainModel() {
    const input = sensor1Data.map((_, index) => [
        sensor1Data[index],
        sensor2Data[index],
        sensor3Data[index],
        sensor4Data[index]
    ]);

    console.log(input);
    const output = sensor1Data.map((_, index) =>
        (sensor1Data[index] + sensor2Data[index] + sensor3Data[index] + sensor4Data[index]) / 4
    );
```

Fig. 3. Syntax code responsible for training of the machine learning model with historical data

After training the polynomial regression with this historical data, a model is generated that is used for real-time predictions.

Although there are other possibilities to implement machine learning, we considered that in this sensor system the use of this solution is optimal from the point of view of complexity and performance. Polynomial regression is simpler than other models used for machine learning, but in terms of performance it contributes significantly to improving virtual sensor temperature prediction by adapting to fluctuations in physical sensor data. This fact makes machine learning much more effective in real environments.

3. EXPERIMENTAL MEASUREMENTS

In this chapter we evaluated the performance of the system in different configurations. Before presenting these measurements, it is important to describe how the system operates. In a simplified way, the system works like this: the data collected from the physical sensors is sent to the machine learning model, which uses the new data together with the historical data retrieved from the database to generate the response of the virtual sensor. Following this operation, the physical sensor and virtual sensor data will be saved in the database and transmitted to the web interface for display. The described process is repeated at a predetermined time interval.

In this chapter we tested the performance and reliability of the virtual sensor through two experiments. During these experiments, we exposed the system to various configurations where we varied the number of physical sensors used and the sensor placement arrangement. In order to obtain the most conclusive results, the two experiments were carried out in similar environments, more precisely in a laboratory room.

We mention that in both experiments to measure the accuracy of the virtual sensor, we placed a physical control sensor, its physical location is exactly where the virtual sensor makes the temperature prediction.

In the following we will present the two configurations in which the system was tested, the various measurements performed and the results obtained.

3.1. Evaluation of Triangle and Square Configurations

In the first experiment we used the triangle configuration, this involves the use of three physical temperature sensors that were placed in the shape of a triangle and a control sensor placed in the middle of the triangle. Based on the data collected from the three sensors, machine learning generated the response of the virtual sensor. And finally, to compare the accuracy of the virtual sensor, its responses were compared with the values measured by the fourth physical sensor, the control sensor, which is located in the same location where the virtual sensor makes the temperature prediction. In Table 1 you can see the data obtained with the triangle configuration.

Table 1. Data table for the triangle configuration

Time	Sensor 1 (°C)	Sensor 2 (°C)	Sensor 3 (°C)	Control Sensor (°C)	Virtual Sensor (°C)	Difference (%)
12:15	20.94	20.71	20.87	20.77	20.92	0.72
12:30	21.15	20.82	20.89	20.81	20.90	0.43
12:45	21.36	20.93	21.05	20.87	21.01	0.67
13:00	21.58	21.05	21.20	21.05	21.15	0.47
13:15	21.79	21.18	21.36	21.45	21.33	0.56
13:30	21.92	21.45	21.54	21.75	21.65	0.47
13:45	22.03	21.68	21.70	21.89	21.97	0.37
14:00	22.27	21.87	21.85	22.11	22.18	0.32
14:15	22.48	22.10	22.01	22.53	22.42	0.48
14:30	22.74	22.40	22.25	22.68	22.53	0.67
14:45	22.85	22.75	22.50	22.75	22.88	0.57
15:00	23.14	23.01	22.75	23.07	22.97	0.44

The presented table contains the data collected from the physical and virtual sensors during the experiment. In the last column we made a comparison where we can see the difference between the generated values and the actual measured value.

In the second experiment, we used the square configuration, using four temperature sensors that we placed in a square shape in the laboratory room. As in the first experiment, the data collected from the sensors was used for the machine learning model that generated the response of the virtual sensor. The value obtained is again compared with the value of the control sensor located in the center of the square. Table 2 shows the data obtained with the square configuration.

Table 2. Data table for the square configuration

Time	Sensor 1 (°C)	Sensor 2 (°C)	Sensor 3 (°C)	Sensor 4 (°C)	Control Sensor (°C)	Virtual Sensor (°C)	Difference (%)
12:45	20.39	20.10	19.90	20.15	20.12	20.19	0.35
13:00	20.51	20.33	20.15	20.43	20.41	20.35	0.29
13:15	20.82	20.60	20.45	20.75	20.67	20.77	0.48
13:30	21.14	20.95	20.80	21.19	21.02	21.07	0.23
13:45	21.25	21.20	21.12	21.37	21.32	21.25	0.33
14:00	21.60	21.55	21.33	21.45	21.56	21.50	0.28
14:15	21.90	21.70	21.62	21.85	21.75	21.85	0.46
14:30	22.27	22.05	21.91	22.15	22.10	22.19	0.36
14:45	22.58	22.40	22.01	22.25	22.35	22.42	0.31
15:00	22.79	22.65	22.31	22.55	22.67	22.58	0.40
15:15	22.84	22.75	22.52	22.83	22.74	22.69	0.22
15:30	23.04	22.85	22.77	23.12	22.88	22.94	0.26

We chose this configuration to see what impact adding a temperature sensor in addition to the previous configuration has on temperature prediction accuracy. We also looked at whether the square geometric position improves the accuracy and reliability of the system.

As can be seen in the last column of each table presented above, the machine learning model was able to generate accurate temperature predictions in both scenarios, but as expected the accuracy was higher in the case of the four-sensor configuration. Therefore, we can conclude that a larger number of physical sensors contribute to improving the performance and obtaining a better temperature prediction. Following the two experiments, it can also be seen that both configurations produced satisfactory results, specifically the triangle configuration had a margin of error of the response between 0.32-0.72%, while the square configuration had a smaller margin of error, between 0.22-0.48%.

Analyzing the two experiments strictly from a performance point of view, it is clear that the four-sensor system is the favorite, however using this configuration in an extended network would generate additional costs due to the fourth sensor. Therefore, if we wish a more economical system, but with slightly reduced accuracy, the configuration with three sensors is ideal. On the other hand, if high efficiency is a priority, the four-sensor system is the optimal choice.

We mention that both configurations improve their accuracy over time, because in the experiments we used a relatively small amount of data to train the machine learning model. However, in a real application, running over an extended period of time, the system can accumulate a very large set of data, which will train the machine learning model and greatly improve its accuracy.

3.2. The Motivation Behind the Chosen Configurations

We would like to point out that the presented system can work in different configurations than those presented. For example, the virtual sensor can work with only two physical temperature sensors, but it can also work with five or more sensors. But based on the experiments we concluded that the use of only two sensors considerably reduces the performance, and the use of several sensors considerably increases the costs. Thus, within the system we opted for the use of the two scenarios, with three and four sensors, because in this way the system has a balance in terms of performance and economy.

The choice of the two geometric configurations, triangle and square respectively, is not random. We considered these arrangements to be optimal for future sensor network developments, as these simple geometries provide us with an efficient solution in terms of how to place and interconnect sensors. For example, the implementation of the system on a larger scale requires a model of placement and organization of sensors because a random placement would complicate the expansion of the network later. Therefore, the geometric shapes presented give us a way of connecting and distributing the sensors very well organized.

Another reason we chose these configurations is that they allow for more accurate spatial monitoring. Triangle and square configurations ensure an even distribution of sensors, effectively covering the entire monitored area. If the physical temperature sensors were randomly placed, some areas might not be monitored correctly, creating an inconsistency in the data collected, which would later negatively influence the virtual sensor. Therefore, the chosen geometric configurations contribute considerably to improving the performance of the machine learning model.

In conclusion, the chosen geometric configurations allowed the effective monitoring of the temperature in the space where the measurements were made, this can also be seen in *fig. 4,* it illustrates the web interface of the system. It displays the data measured by the sensors in real time.

Sensors Table

Date & Hour	Sensor 1 (°C)	Sensor 2 (°C)	Sensor 3 (°C)	Sensor 4 (°C)	Control Sensor (°C)	Virtual Sensor (°C)
2024-09-10 / 12:45	20.39	20.10	19.90	20.15	20.12	20.19
2024-09-10 / 13:00	20.51	20.30	20.15	20.40	20.35	20.41
2024-09-10 / 13:15	20.82	20.60	20.45	20.75	20.67	20.77
2024-09-10 / 13:30	21.14	20.95	20.80	21.00	21.02	21.07
2024-09-10 / 13:45	21.25	21.20	21.10	21.30	21.25	21.32
2024-09-10 / 14:00	21.60	21.55	21.30	21.45	21.50	21.56
2024-09-10 / 14:15	21.90	21.70	21.60	21.85	21.75	21.85
2024-09-10 / 14:30	22.27	22.05	21.90	22.15	22.10	22.19
2024-09-10 / 14:45	22.58	22.40	22.00	22.25	22.35	22.42
2024-09-10 / 15:00	22.79	22.65	22.30	22.55	22.58	22.67
2024-09-10 / 15:15	22.84	22.75	22.50	22.80	22.69	22.74
2024-09-10 / 15:30	23.04	22.85	22.70	23.00	22.88	22.94

Fig. 4. The web interface for monitoring sensors

In the web interface you can see in the first column the time at which the sensor data was collected, the following columns are those dedicated to the physical data sensors. After that, the control sensor column follows and finally in the last column you can see the response of the virtual sensor.

4. CONCLUSION

The presented project demonstrates the efficiency and utility of a virtual temperature sensor that uses data collected from physical sensors to train a machine learning algorithm. This approach represents an innovative solution within existing temperature monitoring systems.

One of the most important aspects of the project is the considerable cost reduction, especially for large temperature monitoring systems. This fact is due to the possibility of replacing some physical sensors with virtual sensors, thus reducing the number of physical sensors needed. Therefore, the use of virtual sensors leads to a significant decrease in the expenses related to the purchase, installation and maintenance of such a system. This aspect being very important in the industry where budget and space are limited.

The presented solution not only significantly reduces the costs associated with such a system, but also gives it increased flexibility. Temperature monitoring systems using virtual sensors are more flexible because they can measure temperature in locations inaccessible to physical sensors.

The use of polynomial regression within machine learning algorithms has proven optimal for temperature prediction, adapting well to non-linear data variations and providing accurate estimates based on data collected from physical sensors.

The developed web interface allows users to access and analyze data collected from sensors in real time. This aspect representing an improvement in the process of monitoring and managing data.

The presented system has high versatility, it can be used in various configurations depending on the need, for applications that require high precision the system can be adapted to a configuration with several sensors. On the other hand, if precision is not a crucial factor and the goal is to obtain an economical system, then a configuration with fewer sensors can be chosen. It should also be noted that continuous training of the machine learning model with historical and new data contributes significantly to increasing the accuracy of the virtual sensor predictions.

In conclusion, the use of a virtual temperature sensor based on machine learning algorithms represents an effective and innovative solution to improve the efficiency of temperature monitoring systems, while optimizing the costs and space required for these types of systems.

In further developments we aimed at using more advanced learning algorithms, changing sensors and improving the web interface in order to expand the scope of the system.

REFERENCES

[1] K. Vyas, S. Shukla, H. Dsouza, J. Chacko, D. D'Souza, V. Narnaware, *A Survey on Environment Monitoring using Sensor Networks*, Asian Journal of Convergence in Technology, Vol. VI, No. III, pp. 95-99, .2020

[2] M. Mansoor, I. Haneef, S. Akhtar, A. D. Luca, F. Udrea, *Silicon diode temperature sensors – A review of applications*, Sensors and Actuators A Physical, Vol. 232, pp. 63-74, May 2015.

[3] R. S. Diarah, C. Osueke, A. Adekunle, S. Adebayo, A. B. Aaron, O. O. Joshua, Types of Temperature Sensors, IntechOpen, October 2023.

[4] A. Brunello, A. Urgolo, F. Pittino, A. Montvay, A. Montanari, *Virtual Sensing and Sensors Selection for Efficient Temperature Monitoring in Indoor Environments*, Sensors, Vol. 21, 2728, April 2021.

[5] E. Ostertagová, *Modelling using polynomial regression*, Procedia Engineering, Vol. 48, pp. 500-506, December 2012.

[6] M. A. Alsheikh, S. Lin, D. Niyato, H. P. Tan, *Machine Learning in Wireless Sensor Networks: Algorithms, Strategies, and Applications*, IEEE Communications Surveys & Tutorials, Vol. 16, No. 4, pp. 1996-2018, May 2014.

[7] P. Ghule, M. Kambli, *Web Based Environment Monitoring System Using IOT*, 3rd International Conference on Trends in Electronics and Informatics, April 2019.

[8] S. Corbellini, E. D Francia, S. Grassini, L. Iannucci, L. Lombardo, M. Parvis, *Cloud based sensor network for environmental monitoring*, Measurement, October 2017.

[9] X. Cai, *The Development and Diverse Applications of Temperature Sensors in Different Temperature Environments*, Applied and Computational Engineering, Vol. 37, No. 1, pp. 141-148, February 2024.

[10] D. Fabiocchi, N. Giulietti, M. Carnevale, H. Giberti, *AI-Driven Virtual Sensors for Real-Time Dynamic Analysis of Mechanisms: A Feasibility Study*, Machines, Vol. 12, No. 4, pp. 257, April 2024.