JOURNAL OF ENGINEERING SCIENCES

Volume 7, Issue 1 (2020)

Kanwal, T., Altaf, S., Javed, M. K. (2020). Environmental monitoring smart system with self-sustaining wireless sensor network using data validation algorithms. Journal of Engineering Sciences, Vol. 7(1), pp. E10–E19, doi: 10.21272/jes.2020.7(1).e3

Environmental Monitoring Smart System with Self-Sustaining Wireless Sensor Network Using Data Validation Algorithms

Kanwal T.[1], Altaf S.[2*][0000-0002-3455-6854], Javed M. K.[1]

[1] Pir Mehr Ali Shah Arid Agriculture University Rawalpindi, Muree Rd., 46000 Punjab, Pakistan;
[2] Auckland University of Technology, 55 Wellesley St., 1010 Auckland, New Zealand

Article info:
Paper received: February 14, 2020
The final version of the paper received: May 5, 2020
Paper accepted online: May 19, 2020

***Corresponding email:**
saltaf@aut.ac.nz

Abstract. Study in Wireless Sensor Network (WSN) has been becoming an emerging and promising research topic aiming for the advancement in the Internet of Things (IoT) for a reliable connection. The capability of the wireless sensor to be used in a complex environment can become hard to reach areas and also be able to communicate in an ad-hoc manner, attracted researchers in recent times. Development in wireless sensor network producing a lot of new applications to sense environment remotely are facing challenges restricting it to perform up to its potential. Data validation and data reliability are such existing problems in this domain that needed to be addressed. Because sensed data cannot be blindly trusted upon, as it may have faults and errors occurred with-in the sensing environment. Besides, to guarantee the active state of the sensing system in a remote area is also essential in terms of power usage and management. The focus of the paper is data validation acquired from sensors deployed in remote areas. Although, lots of data validation algorithms have been proposed by researchers to identify single data fault. However, our research identifies multiple faults, namely spike fault, out of range fault, outliers, and stuck at fault using a hybrid form of an algorithm. A comparison with the existing algorithm shows that the proposed algorithm improved data validation by 97 % in detecting multiple data faults using Artificial Intelligence techniques.

Keywords: wireless sensor network, data validation, feature extraction, feature identification, algorithm.

1 Introduction

The sensor is the ubiquitous technology that is used to extend the human senses by sensing the environment digitally and converting them to human-readable format. Sensors are becoming part of our lives by integrating them in our daily used item; they are becoming part of our lifestyle and becoming part of our life. Although it was the circumstance previously that sensors had been used as a primary program between the physical world and human notion, sensor data is more regularly combined and prepared today, launching additional indifference measures to the chain. One common everyday-example is the mobile phone, where the typical consumer is not considering the original data of microphone, CCD camera, MEMS accelerometer, the GSM-modem on its own, and many more, however, in the functions these sensors provide, producing the utilization of these devices better, secure or interesting. Several computer technology visions further go actually, introducing ideas such as pervasive or ubiquitous processing [1] clever dirt or the Internet of issues. Most of these possess a common basis that involves the implementation of an electronic user interface to the real world, creating an immersive experience. This interface is made up of small and low-priced CPUs usually, with the capacity of contacting one another and which can feel their environment. These situations have had a significant impact on the development of a new marketing idea, comprising distributed stuck systems, which can be referred to as Wireless Sensor Network (WSN). Predicated on advantages the WSN- idea brings to a huge amount of different applications, curiosity inside the equivalent technology can be excessive. Preferably, a WSN permits the application of a sizable quantity of sensor nodes, which configure themselves, with respect to the network community and topology situation. After sensing their physical atmosphere and locally digesting the obtained data, nodes communicate all their data (or an extract) towards a network sink, where data is additionally made and processed designed for readout.

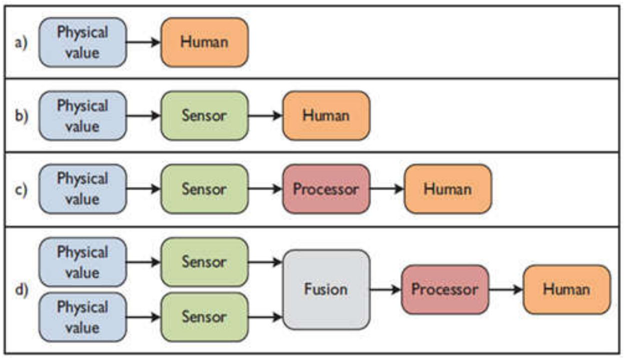

Figure 1 – Advancement of sensing

While transmitted info should find a very decent path towards its vacation spot automatically, the network can be managed and therefore be used as one large measurement instrument remotely. Typical app areas just for WSNs consist of business control systems, movement detecting, as well as the monitoring of environments or structures. Depending on the application constraints and issues, WSNs can undertake different varieties, use several technologies, and connect through different network topologies, making the look of WSNs application-specific extremely [2].

2 Literature Review

Although EM can mean the monitoring of any type or kind of environment, it is mostly thought of as the monitoring and research of natural environments often. The building blocks of EM may be the collection of data, which allows a better understanding of our all-natural environment being obtained using observation. Scientifically, Environmental Monitoring includes the areas of Physics, Biology, and chemistry. However, because more systems for data acquisition, especially, get involved, so do the real number of technical areas of research. The motivation predicated on the ever-increasing globe inhabitants implies that EM is not just about getting the knowledge of environment it is a complete science that has the primary role in our lives and affects our life significantly as the environment affects our behavior and lifestyle and monitoring of the environment is very important to be collected and recorded .main application of EM is to safeguard drinking water supplies, monitoring of waste material especially radioactive waste, treatment of the populated environment, it is used mainly in weather monitoring and weather prediction is the most essential to protect our crops and stays alert of floods and droughts [3]. EM uses a sensor to gather data from the connected environment using its senor nodes than the sensed data is sent to base node or sink node which checks and validate this sensed data, and it is compiled and then sent to the server where it is converted to understandable human form and then shown to the user, and also used for other purposes like prediction and forecasting. For sensing the environment, the different sensor is connected with the processing node which is usually a microcontroller which controls and gather the data from the help of sensor and send them to their respective sink node and to achieve all this the senor node need power which is the main issue in remote areas where the power generation is main issue and storage of power is an issue which is needed to be addressed, so the sensor node stays working for longer and give us up to date data. Grab sampling is the manual removal of an example from the surroundings for additional analysis. While this is simply not performed as was the case previously because of technological progress frequently, in some full cases, get sampling continues to be used to permit unique sampling or even more complex examination than can be done in the field. One particular immense problem with getting the sample may be the involvement of humans, which leads to cost and time issues, furthermore to high invasiveness. Sampling stations refer to sensor devices deployed in the environment of interest, monitoring the surrounding or in defined intervals continuously.

In recent years, different problems made investing more valuable in the monitoring of environment like a drought in the US in 2012. To overcome this, a novel approach is made to monitor the environment and predict the next drought. This system used infrared rays that use remote image sensing to gather information and also it uses geographic information system to search for air irregularities and water quality, with the improvement in sensing technology and advancement in low-cost microcontroller it is becoming more and more favorable in recent years.

A network program was recommended for environmental monitoring of indoor circumstances of building. The operational system architecture contains low-power. Wireless detectors on a prerelapse SOC machine, linked to the web applying Wi-Fi requirements. Before brought up conversation lists, just several solutions that have recently been proposed meant environment monitoring remotely, using wireless sensing strategies and GIS-based technologies correctly. However, many of these solutions only solve the data monitoring and gathering areas of the nagging problem. To tackle the nagging issue in its entirety, the acquired data must become analyzed, and relevant findings end up being presented through prompt notifications to monitoring personnel [4] propose a remedy where automotive Radio- regularity identification tags, built with temperature, light, and speed detector, are used. The ability is had by the device to collect RF energy, and its procedure has been validated testbeds and experiments, in which the obtained data happen to be gathered by simply a bunch personal computer by using a reader antenna. The look achieved a transmission selection of up to 10 and 20 m in fully battery-powered, correspondingly. The proposed system can be used for monitoring the outside or ambient weather parameters, and, if the web host PC gets a Web connection, it can be composed of an IoT-based solution. Benchmark [5] symbolizes an open-source Wi-Fi fine mesh networking component for environmental monitoring, to advertise this sort of course-plotting in such applications. That is based on a low-cost RF transceiver, with an increase of small and less complicated code compared to the among a ZigBee style, and performs regarding a sensor node. The operational system was compared to an off the shelf product, X-Bee, with the conclusion that it can provide comparable or better performance than commercial items even. Various other systems designed for monitoring applications owned by numerous fields, each one predicated on ZigBee, are reported in [6]. The main drawback of the consists of the necessity for that gateway in the event data should be sent on the Internet, a fundamental requirement of IoT cases.

Different monitoring solutions based on different technology possess are and appeared attaining surface primarily in-home automation, following its benefits this year 2018 [7] The task in [8] is made up of the advancement of an innovative energy management approach just for smart homes predicated on enabled wi-fi networks. By simply providing low power, low cost, and lowered device measurements, the freelance writers think that this kind of technology includes a high potential to become essential both IoT and for wise homes. This trend shall also be sustained by the availability of native support provided by current mobile devices in comparison to IEEE 802. 15. 4, which will reduce the expense of devices also. The simulation results show that this approach plays a part in the reduced amount of peak load electrical power and demand consumption fees, resulting in monetary savings ultimately. Furthermore, it is been shown the fact that the overall performance of this proposed BLE network is preferable to the one acquired regarding IEEE 802. 12-15. 4 with regards to packet delivery ratio, hesitate, and jitter. With the constant improvements taken to the process, like the support for nylon uppers networking, plus the expansion of the number presented, it is believed that technology shall be considered for implementing environmental monitoring applications.

The authors reported the development of Wi fi sensors mailing heat and relative moisture measurements into a bottom section using UDP. A battery pack lifetime of 24 months with a twenty min way of measuring routine was achieved. This motivated the advancement of any gadget employing HTTP, for investigating the charged power efficiency of the more reliable solution, from the connection viewpoint. To be cost-effective, the sensor nodes are powered by extremely restricted-energy reserves often. Premature strength depletion can easily severely limit the network program [9] and must be addressed considering the IoT software requirements for price, deployment, repair, and provider availability. These become more very important to monitoring applications in severe climatic environments even, such as for example, snow, volcanoes [5]. The knowledge of this kind of environment can considerably benefit from continuous long-term monitoring, but their circumstances emphasize the problems of client energy administration, mechanical and conversation solidifying, size, excess weight, and deployment procedures.

Open character deployments [10] and interaction protocol advancements and trials [11] show that WSN optimization for the reliable procedure is costly and time-consuming. It satisfies the IoT program requirements for long-term barely, reliable and low-cost service unless of course, reusable equipment and software platforms [12] can be found, including flexible Internet- allowed servers [13] to gather and procedure the field info for IoT applications.

This kind of paper additions of curiosity for experts in the WSN field could be summarized when comprehensive specifications to get a challenging WSN app for long-term environmental monitoring that can be utilized to investigate the optimality of innovative WSN alternatives, specifications, design factors, and trial and error outcomes with respect to platform elements that match the typical IoT application requirements of low priced, high reliability, and lengthy service period, specifications and design and style considerations with regards to system reusability for an array of the distributed event- centered environmental monitoring applications, and an easy and configuration-free field application procedure ideal for large scale IoT software deployments.

3 Research Methodology

3.1 Data validation faults

Wireless sensor network that is deployed at remote areas ranges from a few hundred to thousand to gather environment data. After data is sensed, it is sent to the base station, which is then transmitted to the server for decision making and prediction. Data could get faulty due to internal or external factors [14]. These effects include environment effect, limited resources hardware or software problem, etc.

Out of range fault is mostly refer to the data values that are out of range of reasonable limit of data values this range depends on different parameters like sensor and feature selected the limit always remain within domain limit [15].

Stuck at fault refers to data fault when the data values remain on a single value, or the difference between the previous and selected data value is zero [16]. These types of fault are hard to discover because the normal data value can remain the same for the different periods.

Outliers are those data value that is different from another sensor they usually are within the range and sometimes can go out of range of normal data range [17]. When within range these data values cannot be detected without the reference of other sensor data values.

Spike fault occurs when the data values change too much over a short value of time this change can be on the higher side or on the lower side [18]. these types of fault usually make peaks in data value.

3.2 Data validation fault detection algorithms

In this section, three different types of algorithms are discussed for data validation and its fault detection to improve accuracy.

The heuristic rule is used to check data for faults if data is within some threshold limit; it is marked good. Otherwise, it is marked faulty that the algorithm is defined below [19]. The limit is defined on the knowledge of the domain:

```
Input: Queue of sensed (x)
Output: Position for sensed data (x)
  Count Faults = 0;
  for i ← 0 to number_data do i ← i + 1
     if  (x[i]  ≥  min_limt  and  x[i]  ≤  max_limit)
then
        status [i] ← likely good
     else
        status [i]← likely fault
        location [count Faults] ← i
        Count Faults ← Count Faults + 1
     end if
   end
```

Temporal correlation is used to check data for faults, it usually checks for stuck at fault by comparing the result of the same sensor with the previous value, and if the data is not changed then it is stuck at fault [14]. The algorithm is defined below:

```
Input: array of sensed data (x)
Output: status for sensed data (x)
Count Faults = 0; Stuck fault = 0;
        for i ← 0 to number_data do i ← i + 1
            if (|x [i] − x [i + 1] | = 0) then
                fault location [Count Faults] ← i
        + 1
                Count Faults   ← Count Faults +1
            Else
                 status [i] ←. likely good
                 status [i +1] ← likely good
        end  if
        end
        for j  ← 0 to Count Faults do  k ← k + 1
            i ← fault location  [k]
            i + 1 ← fault location  [k + 1]
                if (|x [i] − x [i + 1] | = 0)
                    then status  [i + 1] ← Stuck
            fault
                    Stuck fault location [Stuck
            fault]  ← i + 1
                    Stuck fault  ← Stuck fault +
            1
            else
                status [i] ← likely  good
                status [i+1] ← likely good
 end
 if end
```

Spatial correlation is used to check the faults by comparing it with other sensor data, and if the data value is among the average of the other value, they are considered good otherwise false [16]. The algorithm is defined below:

```
  Input: An array of sensed data (x) Output:
status for sensed data (x) count likely faults =
0;
  for i ← 0 to n do i ← i + 1
    if(|x[i]-s_median|<threshold)
  then status [i]← likely good else status [i] ←
likely fault location [count likely faults] ← i
  count likely faults ← count likely faults + 1
  end if end
```

3.3 The improved data validation algorithm

In our approach, we combined different algorithm and it is used to detect multiple data faults at once it combines all the faults identification capabilities into one single algorithm [19]. It is based on 3 steps. First, the data is checked for spike faults, then it is checked for out of range faults, and in the end, it is used to check for outliers, which is most difficult to check. The algorithm is shown below:

```
  Input: array of sensed data (x) Output: status
for sensed data (x) count_Faults  = 0;
  for i ← 0 to number_data do i ← i + 1
  if (x[i] ≥ min_limt and x[i] ≤ max_limit) then
   status [i] ← likely good else status [i]←
likely fault location [count_Faults  ] ← i
  count_Faults  ← count_Faults  + 1
  end if
  end
  //step 2
  count_Faults  = 0; Stuck at fault = 0; for i ←
0 to number_data do i ← i + 1
  if (|x [i] − x [i + 1] | = 0) then fault
location [count_Faults  ] ← i + 1
  count_Faults  ← count_Faults  +1
  else status [i] ← likely good status [i +1] ←
likely good
  end if
  end
  for j ← 0 to count_Faults  do j ← j + 1
  i ← fault location [j]
  i + 1 ← fault location [j + 1]
  if (|x [i] − x [i + 1] | = 0) then status [i +
1] ← struck-at fault
  fault location [Stuck at fault] ← i + 1 Stuck
at fault ← Stuck at fault + 1
  else status [i] ← likely good status [i+1] ←
likely good
  end if end
  //step 3
   Countlikely_faults = 0; countspike_faults =
0; for i ← 0 to n do i ← i + 1
  M = 0.6745 × (|x [i] − x̃) /MAD
  if (|Mi| > 3.5)
  then status [i] ← likely fault fault location
[countlikely_faults] ← i
  countlikely_faults ← countlikely_faults + 1
else status [i] ← likely good end if
  end
  for j ← 0 to countlikely_faults do j ← j + 1
  i ← fault location [j]
  i + 1 ← fault location [j + 1] if (|fault
location [i] − fault location [i + 1]| = 1)
  then
  status [i + 1] ← spike fault fault
locationspike faults [countspike faults] ← i + 1
countspike faults ← countspike faults + 1
  else status [i + 1] ← likely good
  end
  if end
```

4 Results and Discussion

4.1 Simulation results

For verifying our algorithms, we built testbed and simulate in Matlab the models to gather different data samples and results are as follow (Figure 2).

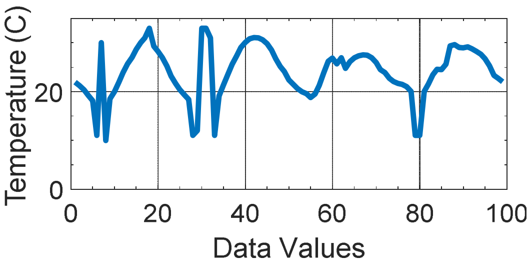

Figure 2 – Sensor data values

For gathering temperature data, we placed a different sensor and collected the following data sets. Firstly, we collected the original data from different sensors, and data values are shown below in Figure 3.

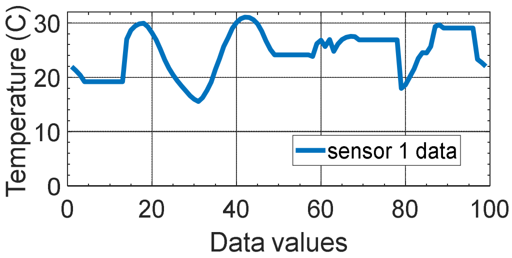

Figure 3 – Stuck at fault

Stuck at fault is shown below when a fault is detected by the algorithm it is highlighted in red (Figure 4).

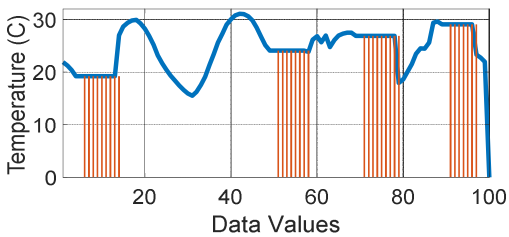

Figure 4 – Stuck at fault detected

Out of range fault is shown in below graphs where it is compared with original data, and any data that is out of bound is marked with red and original data is shown in yellow (Figure 5).

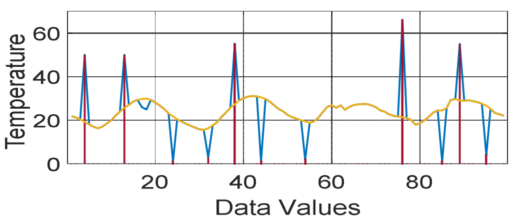

Figure 5 –Spike fault

In the below graph, the outlier is shown when it is compared with different data values; it is identified in red and other sensor data values are in yellow and purple (Figure 6).

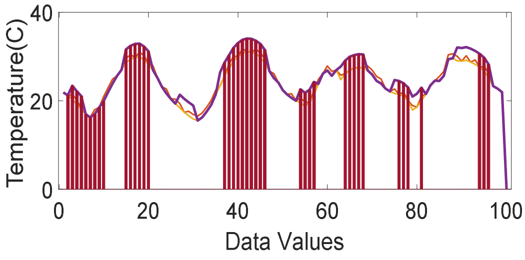

Figure 6 – Outlier fault detection

When data is fed through our state-of-the-art algorithm, it detects almost all of the faults and correctly identified all of them (Figure 7).

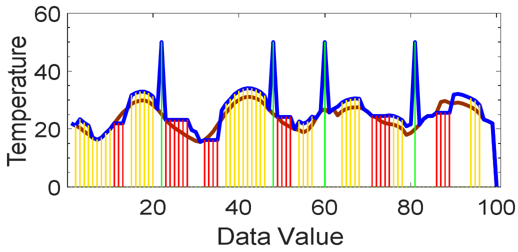

Figure 7 – State of the art result

When multiple fault data are feed to algorithm it only detects stuck-at faults and ignore out of range and spike faults (Figure 8).

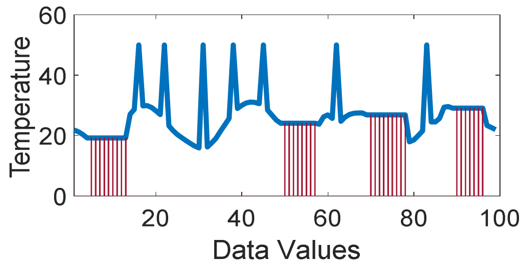

Figure 8 – Heuristic rule

When it is feed to the heuristic rule algorithm. It only detects out of range. Spike faults ignors stuck at faults. Result is shown in Figure 9.

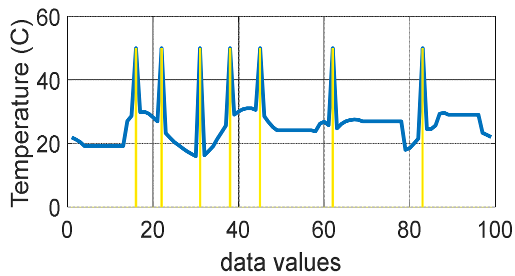

Figure 9 – Spike faults

When the sensor data is feed to modified z score it is compared with the other data and only outliers are identified and it ignored stuck at faults.

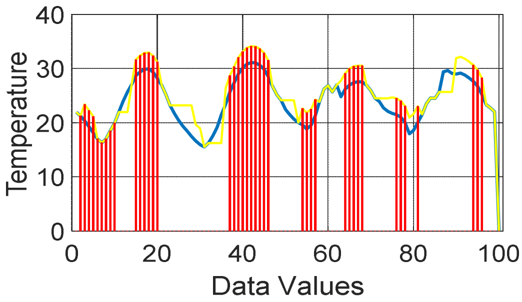

Figure 10 – Modified Z Score

To test our algorithm, we also consider humidity and all the test are also run on the algorithms (Figure 11).

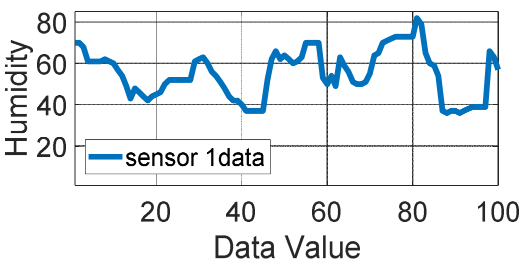

Figure 11 – Sensor data 1

Humidity values are collected using DHT 11 sensor attached with the testbed.

Sensor 1 data it introduced with the stuck-at fault, and the resulted value are shown in Figure 12.

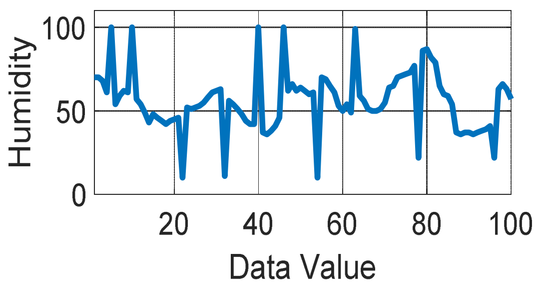

Figure 12 – Sensor data with faults added

While the sensor 2 value is getting spike faults and is shown below. Stuck at fault is shown below in Figure 13. when a fault is detected by the algorithm it is highlighted in red.

Out of range fault is shown in the below. Where it is compared with original data, and any data that is out of bound is marked with red and original data is shown in blue.

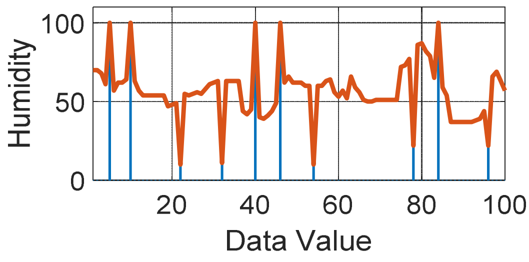

Figure 13 – Spike fault

In below graph outlier is shown in Figure 14. when it is compared with different data values, it is identified in red.

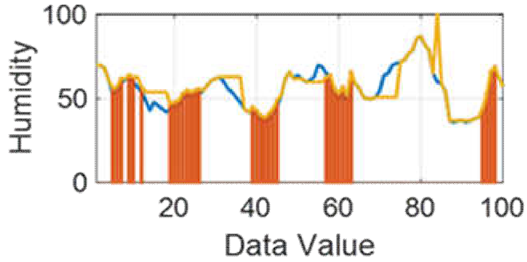

Figure 14 – Outliers faults

When it is feed to the heuristic rule algorithm. It only detects out of range. Spike faults. Ignoring stuck at faults. The result is shown in Figure 15.

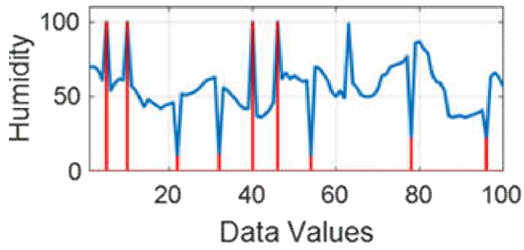

Figure 15 – Spikes faults

When multiple fault data are feed to algorithm it only detects stuck-at faults and ignores out of range and spike faults (Figure 16).

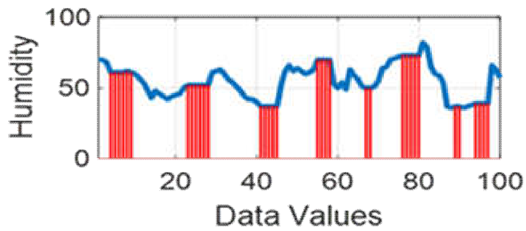

Figure 16 – Stuck at fault

When the sensor data is feed to a modified z-score it is compared with the other data, and only outliers are identified, and it ignored stuck at faults (Figure 17).

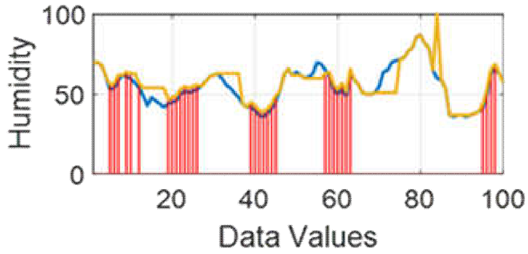

Figure 17 – Modified Z score algorithms results

When data is fed through our state-of-the-art algorithm, it detects almost all of the faults and correctly identified all of them (Figure 18).

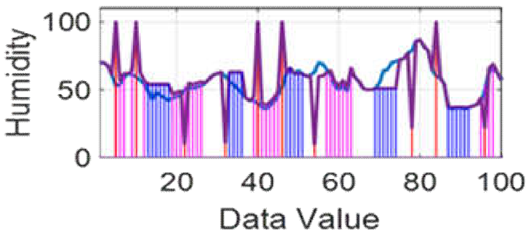

Figure 18 – The modified algorithm

As seen from the results that no single algorithm is capable to detect different type of fault. When used they only detect the fault. They are designed to detect despite having multiple faults in data, which can cause the faulty data to get inside our prediction systems which in return lower the reliability of a system and forecasting. These algorithms are also responsible for false-positive reports when data is in a faulty state.

4.2 Comparison od different case studies results

To compare different algorithms, we consider the following cases for evaluation.

Case 1. Data set with 10 % out-of-range faults.

Case 2. Data set with 10 % stuck-at faults.

Case 3. Data set with 10 % spike faults.

Case 4. Data set with 10 % outliers.

Case 5. Data set with 10 % out of-range faults and 10 % stuck-at faults.

Case 6. Data set with 20 % out-of-range faults and 20 % stuck-at faults, 20 % outliers.

Case 7. Data set with 20 % out-of-range faults and 20 %struck-at faults, 20 %outliers, 20 % stuck-at faults.

This Fault detection rate of Algorithm 1 is 100 % in Case 1, 0 % in Case 2, 0 % in Case 3, 33 % in Case 4, 50 % in Case 5, 40 % in Case 6, and 33 % in Case 7, respectively. Result in Case 1 shows that Algorithm 1 is effective in detecting out-of-range faults, and the results in other cases show; the presence of struck-at faults, outliers and spike faults are not detected by Algorithm.

In case 1 (Figure 19), we introduce the out of range faults to data and compare its performance with other algorithms .as seen in Figure the heuristic rule identifies 100 % of the faults, temporal correlation identifies 0 %, Spatial correlation also identifies 0 % of the faults, Modified z-score identified 0 % while our improved algorithms manage to identify 100 % of the introduced faults.

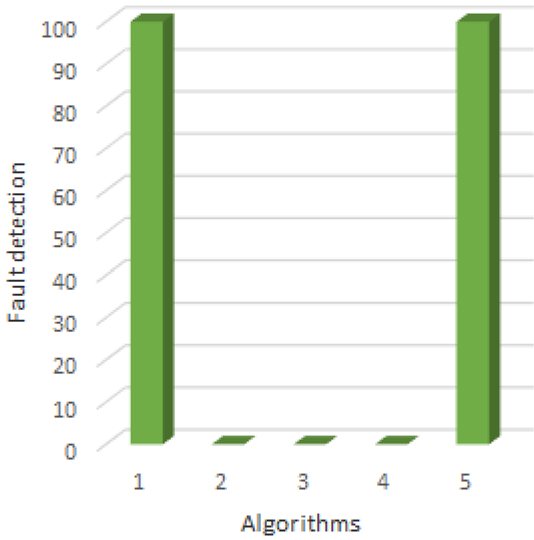

Figure 19 – Case 1 result

In case 2 (Figure 20), we introduce the Stuck-at faults to data and compare its performance with other algorithms. As seen in the figure, the heuristic rule identifies 0 % of the faults, temporal correlation identifies 100 % of the faults, Spatial correlation identifies 0 % of the faults, Modified z-score identified 0 % while our improved algorithms manage to identify 100 % of the introduced faults.

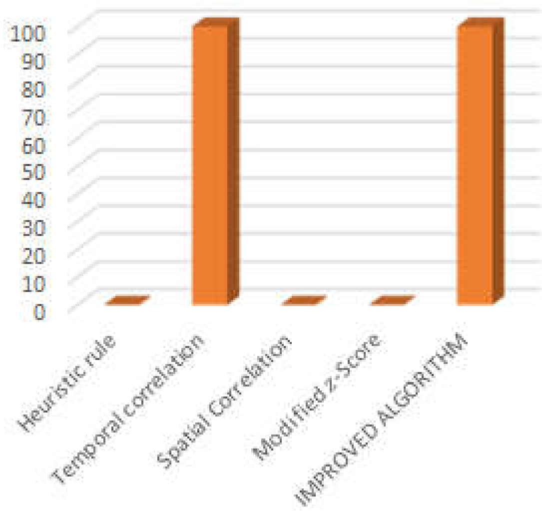

Figure 20 – Case 2 result

In case 3 (Figure 21), we introduce the Spike faults to data and compare its performance with other algorithms. As seen in the figure, the heuristic rule identifies 0 % of the faults, temporal correlation identifies 0 % of the faults, Spatial correlation identifies 100 % of the faults, Modified z-score identified 0 % while our improved algorithms manage to identify 100 % of the introduced faults.

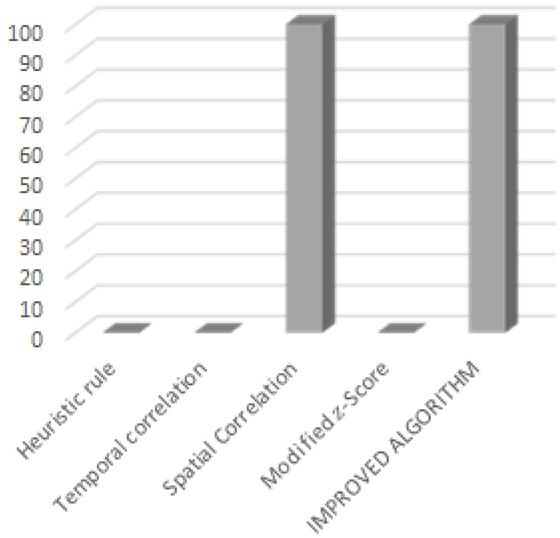

Figure 21 – Case 3 result

In case 4 (Figure 22), we introduce the Outliers faults to data and compare its performance with other algorithms. As seen in the figure the heuristic rule identifies 0 % of the faults, temporal correlation identifies 0 % of the faults, Spatial correlation identifies 0 % of the faults, Modified z-score identified 100 % while our improved algorithms manage to identify 100 % of the introduced faults.

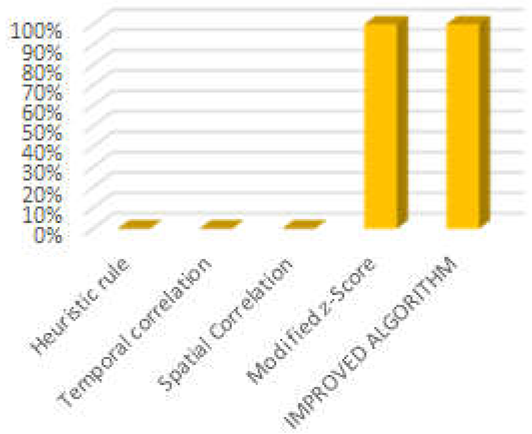

Figure 22 – Case 4 result

In case 5 (Figure 23), we introduce the combination of Out of-Range Faults and Stuck-at Faults to data and compare its performance with other algorithms. As seen in the figure the heuristic rule identifies 50 % of the faults, temporal correlation identifies 50 % of the faults, Spatial correlation identifies 0 % of the faults, Modified z-score identified 0 % while our improved algorithms manage to identify 100 % of the introduced faults.

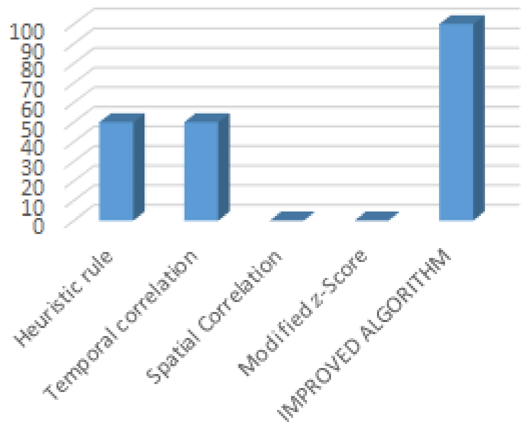

Figure 23 – Case 5 result

In case 6 (Figure 24), we introduce the combination of Out-of-Range Faults, Struck-at Faults, Outliers and Spike Faults to data and compare its performance with other algorithms. As seen in the figure the heuristic rule identifies 33 % of the faults, temporal correlation identifies 0 % of the faults, Spatial correlation identifies 33 % of the faults, Modified z-score identified 33 % while our improved algorithms manage to identify 100 % of the introduced faults.

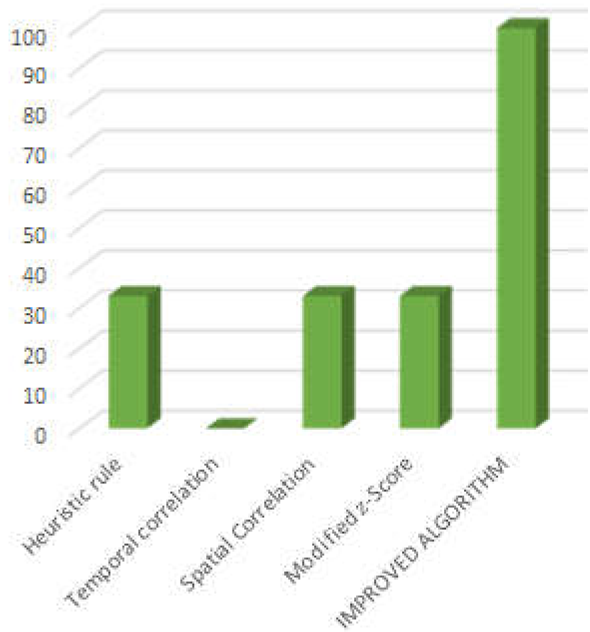

Figure 24 – Case 6 result

In case 7 (Figure 25), we introduce the combination of Out-of-Range Faults, Struck-at Faults, Outliers and Spike Faults to data and compare its performance with other algorithms. As seen in the figure the heuristic rule identifies 25 % of the faults, temporal correlation identifies 25 % of the faults, Spatial correlation identifies 25 % of the faults, Modified z-score identified 25 % while our improved algorithms manage to identify 100 % of the introduced faults.

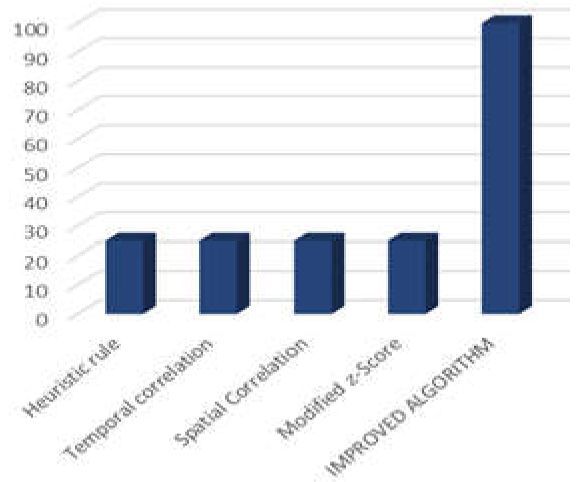

Figure 25 – Case 7 result

As seen from the results that no single algorithm is capable to detect different type of fault. when used they only detect the fault. They are designed to detect despite having multiple faults in data, which can cause the faulty data to get inside our prediction systems, which in return lower the reliability of a system and forecasting. These algorithms are also responsible for false-positive reports when data is in a faulty state.

To accurately detect all the faults, our algorithm uses different methods combined, which in turn detect multiple faults in the same data set. by using multiple data sets, it is more reliable than other single detection-based algorithms. The result compared our algorithm with the other algorithms, which shows that by using our algorithm we detected multiple data faults in single sensor value, which was previously ignored from another algorithm.

5 Conclusions

Data validation is the main key point of this research. WSN is used to sense and collect different parameters of the environment and then used to predict or forecast the environment parameters. When deployed in remote areas, it is essential that the data, we are gathering correct and fault free. There is a different type of data faults that can occur in the data gathering process these faults include stuck at specific value fault out of range fault which can occur when the data value goes out of bounds of normal working range and most difficult of all outliers which remain in the normal working range of the data but are entirely different from the other data values that are being sensed. There is a different algorithm that can detect these faults, but when multiple data faults occur in a data set single algorithm is not enough to identify all the values completely, so we purposed an algorithm that can detect different data faults in a single data source. And in a remote area, a big problem is the lifetime of the sensor, which is for how long it is available for data transfer to self-sustain this setup we used a solar panel to satisfy their energy needfully.

References

1. Zhou, Z., Qin, W., Du, W., Zhu, P., Liu, Q. (2019). Improving energy harvesting from random excitation by nonlinear flexible bi-stable energy harvester with a variable potential energy function. *Mech. Syst. Signal Process.*, Vol. 115, pp. 162–172.
2. Adame, T., Bel, A., Carreras, A., Melia-Segui, J., Oliver, M., Pous, R. (2018). CUIDATS: An RFID–WSN hybrid monitoring system for smart health care environments. *Futur. Gener. Comput. Syst.*, Vol. 78, pp. 602–615.
3. Khalili, H. H., Green, P. R., George, D., Watson, G., Schiffers, W. (2017). Wireless sensor networks for monitoring gas turbine engines during development. *IEEE Symp. Comput. Commun.*, pp. 1325–1331.
4. Cabaccan, C. N., Cruz, F. R. G., Agulto, I. C. (2018). Wireless sensor network for the agricultural environment using raspberry PI based sensor nodes. H*NICEM 2017 - 9th Int. Conf. Humanoid, Nanotechnology, Inf. Technol. Commun. Control. Environ. Manag.*, Vol. 2018, pp. 1–5.
5. Rao Jaladi, A., Khithani, K., Pawar, P., Malvi, K., Sahoo, G. (2017). Environmental monitoring using wireless sensor networks (WSN) based on IoT. *Int. Res. J. Eng. Technol.*, pp. 1371–1378.
6. Galmes, S., Escolar, S. (2018). Analytical model for the duty cycle in solar-based EH-WSN for environmental monitoring. *Sensors (Switzerland)*, Vol. 18(8), pp. 1–32.
7. Jaisankar, N., Subramanian, M. (2018). An optimal path selection in a clustered wireless sensor network environment with swarm intelligence-based data aggregation for air pollution monitoring system. *Int. J. Comput. Aided Eng. Technol.*, Vol. 10(4), 378.
8. Mohemed, R. E., Saleh, A. I., Abdelrazzak, M., Samra, A. S. (2017). Energy-efficient routing protocols for solving energy hole problem in wireless sensor networks. *Comput. Networks*, Vol. 114, pp. 51–66.
9. Zhang, J., Tian, G. Y., Marindra, A. M. J., Sunny, A. I., Zhao, A. B. (2017). A review of passive RFID tag antenna-based sensors and systems for structural health monitoring applications. *Sensors (Bazel)*, Vol. 17(2), E265, doi: 10.3390/s17020265.
10. Khalifa, B., Al Aghbari, Z., Khedr, A. M., Abawajy, J. H. (2017). Coverage hole repair in WSNs using cascaded neighbor intervention. *IEEE Sens. J.*, Vol. 17(21), pp. 7209–7216.

11. Muravyov, S. V., Khudonogova, L. I. (2016). Multisensor accuracy enhancement on the base of interval voting in the form of preference aggregation in WSN for ecological monitoring. *Int. Congr. Ultra Mod. Telecommun. Control Syst. Work.*, Vol. 2016, pp. 293–297.
12. Chaudhary, R., Member, S., Aujla, G. S. (2018). SDN-enabled multi-attribute-based secure communication for smart grid in IIoT environment. *IEEE Transactions on Industrial Informatics*, Vol. 14(6), pp. 2629–2640, doi: 10.1109/TII.2018.2789442.
13. Jin, Y., Qian, Z., Xing, X., Shen, L. (2017). Opportunistic cooperative sensing for indoor complex environment monitoring. *International Journal of Online and Biomedical Engineering*, Vol. 13(8), pp. 4–17.
14. Ni, K., et al. (2009). Sensor network data fault types. *ACM Trans. Sens. Networks*, Vol. 5(3), pp. 1–29.
15. Szewczyk, R., Mainwaring, A., Polastre, J., Anderson, J., Culler, D. (2005). An analysis of a large scale habitat monitoring application. *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, pp. 214–226, doi: 10.1145/1031495.1031521.
16. Tolk, A., Jain, L. C. (2011). Intelligence-based systems engineering. *Intell. Syst. Ref. Libr.*, Vol. 10, pp. 309–325.
17. Barrenetxea, G., Ingelrest, F., Schaefer, G., Vetterli, M., Couach, O., Parlange, M. (2008). SensorScope: Out-of-the-box environmental monitoring. *2008 Int. Conf. Inf. Process. Sens. Networks (IPSN 2008)*, pp. 332–343.
18. Ramanathan, N., et al. (2006). *Rapid Deployment with Confidence: Calibration and Fault Detection in Environmental Sensor*. Center for Embedded Network Sensing, University of California, USA.
19. Jurdak, R., et al. (2009). SensorScope: Out-of-the-box environmental monitoring. *Intell. Syst. Ref. Libr.*, Vol. 10(3), pp. 309–325.