# A Data Cleansing Approach In Smart Home Environments Using Artificial Neural Networks

Chafiq Titouna*, Farid Naït-Abdesselam†* and Asim Darwaish*
*University of Paris, France
email:{chafiq.titouna,asim.darwaish}@parisdescartes.fr
†University of Missouri Kansas City, USA
email: naf@umkc.edu

***Abstract*—A smart home is generally equipped with a set of sensors/devices able to provide intelligent and personalized services to end users. These sensors/devices can sense multiple information related to the physical environment and the residents. This information is then transmitted to a central station for further processing through wireless communication. However, the wireless medium is considered vulnerable and the sensors can fail in providing correct measurements. Moreover, a smart home system should also be able to implement a cleaning system of its sensed data and discard those instances that are erroneous or incoherent. To achieve the data quality improvements, this paper proposes a new approach that uses an Artificial Neural Network (ANN) to detect faulty measurements. The proposed scheme can prematurely and efficiently detect outlier data before forwarding it to a central station. The performance of the solution is validated through simulations, using realistic datasets, and compared with other well-known models. Our findings demonstrate that the proposed approach outperforms the compared models in terms of accuracy, f-score, recall and precision metrics.**

***Index Terms*—Smart homes, Internet of Things, Artificial Neural Networks, outlier detection**

## I. INTRODUCTION

The Internet of Things (IoT) has attracted much attention on the last decade due to the expansion of smart devices. The IoT devices have the ability to sense, aggregate, transmit and make a decision. The potential goals of IoT are often providing to the consumers safety, comfort, assistance, and more exploitation of resources. One of the uses cases is smart homes or home automation. In this use case, we can cite the control of lighting, heating and air conditioning, media and security systems. The monitoring of the resident's behavior or old persons who live alone is also a beneficial usage of smart homes and IoT systems. Thus, energy savings can be ensured by automatically turning off air conditioning, electronic devices, lights, etc. In fact, a smart home allows a set of devices to control the residence and serve home users. The home devices, such as smartphones, smart electric appliances, TV box, etc, provide a huge amount of data every minute which transmitted to a control station in order for making decisions. These devices constitute a complex network that is characterized by its distributed topology, dynamism and frequent changes. In addition, these devices can provide several types of data as temperature, humidity, wind, light, etc. The success of IoT comes from the variety of application's field as smart home [1], smart cities [2], environment monitoring [3], military surveillance [4], etc. However, the data gathered from these devices are endangered and their quality is uncertain, especially when it comes from a remote home. The sensors within devices can become faulty or sent intermittently erroneous measurements. In reality, IoT systems often suffer from some weaknesses such as device energy depletion, poor connectivity, and malicious attacks, which may result in incorrect measurements and data lost. Hence, if the sensed data transmitted by sensors are inaccurate, decisions are likely to be unsound. In IoT system, outliers are considered among the sources that influence data quality. An outlier can be defined as the patterns in data that do not conform to normal behaviour [5]. Recently, several outlier detection techniques have been proposed in both IoT and WSN [6], [7], [8]. Many researches are about an automatic detection of outliers as statistical-based approaches where they require using previously a dataset to estimate a model that can be used to identify outliers. Other techniques train a model in order to classify the sensed data in normal and abnormal data. Despite their interesting features, these solutions cannot give a high quality of data and they require high computational capabilities and exchange of control packets. The main challenge faced when modeling a solution is to check the state of the sensed data and to detect/clean all outliers in while reducing communication costs and energy consumption.

In this paper, we propose an outlier detection technique to improve the quality of the gathered data in the smart home systems. In fact, after the clustering phase, the elected Cluster-Head (CH) classifies and cleans all the abnormal data. To achieve this aim, we use the Artificial Neural Network (ANN) model in order to make the outlier detection process more efficient and accurate. In order to demonstrate the performance of our approach, we have conducted various simulation scenarios using a real dataset. We have also compared our proposal with a set of models proposed by [6]. The obtained results confirm that our approach outperforms the compared techniques.

The rest of the paper is organized as follows. In Section 2, we overview some related work that are close to our approach. Then we describe the preliminaries as well as the network model, graph representation and artificial neural network concepts in Section 3. After, we present in Section 4 our proposed solution. In Section 5, we give the result of the simulation. Finally, we conclude the paper in Section 6.

## II. Related Work

Outlier detection techniques have been proposed during the past few years [9], [10] in aim to detect the outliers data and improve the quality of the gathered data. Several surveys also have been published by [11], [12], in which the approaches are classified in statistical, nearest neighbor, spectral decomposition, clustering and classification. In fact, outlier detection is very useful in detecting strange behaviour and can be applied in several domains as home automation system [13], precision agriculture [14], medicine and health care [15], etc.

The authors in [16] proposed a novel model called NRDD-DBSCAN. The proposed model is based DBSCAN (on density-based spatial clustering of applications with noise) and resilient distributed datasets (RDDs). The purpose is detecting outliers that influence the quality of data of IoT systems. The authors applied their proposal on three datasets with different dimensions (2-D, 3-D, and 25-D) and they obtained good results comparing to DBSCAN and RDD-DBSCAN techniques.

Nesa et al. [6] proposed an IoT architecture in order to detect outliers using four statistical models called: Linear Discriminant Analysis (LDA), Random Forest (RF), Gradient Boosting Machine (GBM) and Classification and Regression Trees (CART). The proposal explored the temporal and spatial dependencies of the sensed data.

Furthermore, Deng et al. [17] presented two techniques in order to identify outliers and detect anomalies in IoT systems. The authors called the techniques OCSTuM (One-class Support Tucker Machine) and OCSTuM based on a genetic algorithm named GA-OCSTuM. These techniques are an extension of the one-class support vector machine to tensor space. They are considered as an unsupervised method for detecting anomalies in IoT sensor data. The result of the experimental evaluations presents an acceptable accuracy.

Recently, Zhu et al. [18] presented a framework called GAAOD for Grid-based Approximate Average Outlier Detection. The authors studied the issue of applying the KNN algorithm on streaming data of IoT applications for detecting outliers. The proposed framework first learns the distance among nodes and their k-th nearest neighbors and constructs a grid-based index to maintain streaming data in the window. Second, they used the k-skybands protocol in order to find the nodes which may be considered outliers.

In this paper, we propose a distributed outlier detection approach for Smart Home Environments. The proposal is based on artificial neural network to identify the outliers in sensed data values gathered by the sensors. The algorithm is performed in a distributed manner and aims to take into account the spatial correlation between the measured data.

## III. Preliminaries

In this section, we describe the preliminaries of the proposed approach by presenting some assumptions of the network model, the graph representation and the artificial neural network concepts.

### A. Network model

In the design of our approach, some assumptions are considered. We suppose that a smart home is composed of sensors in a 2-dimensional field. Sensors are recognized using identifiers and they can change their positions. The architecture of the smart home system adopted in our approach is a hierarchical one, in which a set of clusters is scattered in a home, ie, we consider the sensors which sense the same physical property belong to the same cluster. Finally, the sensed data are transmitted to end users via a central station, where the data are processed.

### B. Graph representation

Smart home system can be represented by a graph $G(V, E)$ where $V$ is the set of vertices representing each smart home sensor (SHS) and $E \subseteq V^2$ is the set of edges (communication link between sensors). $E$ can be defined as: $E = \{(k, m) \in V^2 | k \neq m \wedge Ed(k, m) \leq R\}$, where $Ed(k, m)$ is the euclidean distance between k-th and m-th of an $SHS_i$ (cluster $i$) and $R$ is the communication range. We use the euclidean distance in order to compute the distance between two sensors $k$ and $m$ $(k, m \in V^2)$ belong to the same cluster, ie, $k$ and $m$ sense the same physical property (temperature, humidity, light, etc). The euclidean distance $(Ed)$ is defined as follows:

$$Ed(k, m) = \sqrt{(x_{s_k} - x_{s_m})^2 + (y_{s_k} - y_{s_m})^2} \tag{1}$$

where $x_{s_k}, x_{s_m}, y_{s_k}$ and $y_{s_m}$ represent the coordinates of the sensor $s_k$ and $s_m$ respectively.

In fact, it exists a communication link between two sensors $s_k$ and $s_m$, noted as $cl(s_k, s_m)$, if and, only if the euclidean distance between them $(Ed(s_k, s_m))$ is less than or equal to the shortest transmission range of them.

$$\exists\, cl(s_k, s_m) \leftrightarrow Ed(s_k, s_m) \leq min(tr_{s_k}, tr_{s_m}) \tag{2}$$

Here, $tr_{s_k}, tr_{s_m}$ are the transmission ranges of the sensors $s_k, s_m$ respectively.

### C. Artificial Neural Network

Artificial Neural Networks (ANN) [19] can be defined as an interconnected group of nodes, inspired by a simplification of neurons in the human brain in order to be a computational model to find a solution of particular problems. The basic unit of computation in an ANN is the neuron, often called a node or unit. This unit takes input from one or many units and calculates an output. The output result of a neuron is given by:

$$\begin{aligned} output &= \sigma(\textstyle\sum(weight * input) + bias) \\ &= \sigma(\textstyle\sum_i w_{ji}^l x_i^{l-1} + b_j^l) \\ &= \sigma(W^T X + b) \end{aligned} \tag{3}$$

where

- $W$: vector of weights;

- $X$: vector of inputs;
- $\sigma$: the activation function;
- $x_i^{l-1}$: the inputs of the $i^{th}$ neuron in the $(l-1)^{th}$ layer;
- $w_{ji}^l$: the weight for connection from $i^{th}$ neuron in $(l-1)^{th}$ layer and the $j^{th}$ neuron in the $l^{th}$ layer;
- $b_j^l$: the bias of the $j^{th}$ neuron in the $l^{th}$ layer.

The fundamental structure of ANN is composed of three neuron layers: input layer, hidden layer and output layer. In this case, the outputs of one layer become the inputs of the next layer [19]. Figures 1 and 2 show a simple neuron and a standard ANN. The artificial neural networks have some advantages that make them most suitable for classification. ANNs can learn and model non-linear and complex relationships, which is important in IoT applications. In addition, after learning, the ANN can generalize and infer unseen data for classification and prediction tasks.

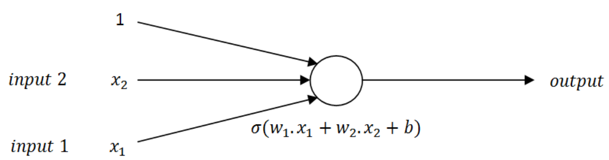


Fig. 1: Structure of an artificial neuron.

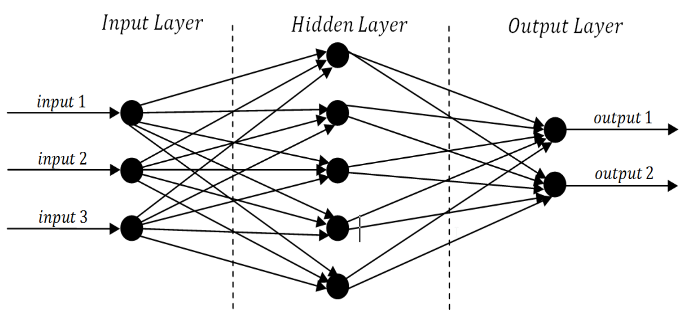


Fig. 2: Structure of an ANN.

## IV. Proposed approach

The approach proposed in this paper is described in detail in this section. The approach includes two phases: the cluster formation phase and the ANN model application phase. In fact, our approach is based on an artificial neural network for detecting outliers within the sensor measurements. To achieve this end, the ANN algorithm should be performed at the control station using the sensed data sent by home sensors. In fact, we need to create a set of categories for the sensed data. These categories will group the measurements of the same physical property such as temperature, humidity, etc. To do that, (a) first, we form clusters that contain smart home sensors ($SHS_i$) which sense the same physical property. (b) second, we perform our ANN algorithm for each cluster measurements.

TABLE I: Table of Sensors (TS)

| ID | Type | PO |
|---|---|---|
| identifier | type of sensor | position coordinates (x,y) |

**Algorithm 1** Control Station ($CS$)

Let $s_k$ be the sensor's ID;
Let $PO_{s_k}$ be the position of the sensor $s_k$;
**Step 1**
Each elapsed time $dt_{s_k}$: broadcast $init\_packet$;
**The packet $init\_packet$ contains $(ID, PO_{s_k})$**
**Step 2-3**
**while** receiving the packet ($init\_packet$) **do**
  - Insert the $init\_packet$ in the table $TS\_table$;
**end while**
**creation of clusters**
**while not** end of table $TS\_table$ **do**
  - Group all $ID$ with the same $Type$ in the same cluster $cl$;
**end while**
- Call *Algorithm 2*

### A. Cluster formation phase

The goal of this phase is to create clusters using the transmitted data. It is performed online within the control station. We describe the steps of this phase as follows:

1) Each sensor sends an initialisation packet called $init\_packet$ which contains two information: $ID$ (the identifier of the sensor) and $PO$ (the position of the sensor);
2) when the control station $CS$ receives an $init\_packet$ from sensors, it saves all information in a specific table called *Table of Sensors* $TS\_table$ (Table I);
3) the $CS$ waits $dt$ time in order to receive all $init\_packet$, then, basing on the received $ID$ and the $Type$ of the sensor, the $CS$ creates clusters $Cl_i$ which contain a set of smart home sensors ($SHS_i$), ie, $SHS_i$ will be composed from sensors which sense the same physical property.
   The algorithm 1 describes the previous steps (steps 1, 2, and 3), in order to initiate the cluster formation and choosing the CH.
4) the $CS$ elects a cluster head
   - creation of an undirected graph $G = (V, E)$ (see the subsection III-B)
   - creation a matrix of adjacency using $G = (V, E)$
   - calculation of the degree of connectivity $deg\_c$; we choose the sensor with higher degree. If two or more sensors have the same higher degree, we select the sensor with higher residual energy.

   The algorithm 2 gives more details on performing the election of the cluster head.

**Algorithm 2** Selection of the cluster head

Let $cl_j$ be a cluster
Let $deg\_c$ be the degree of connectivity;
Let $M[n, n]$ be the matrix of adjacency of the cluster (we use the *Algorithm 3*) $cl_j$
**Step 4**
$i \longleftarrow 1$;
**while** $i \leq n$ **do**
  $j \longleftarrow 1$;
  **while** $j < n$ **do**
    $sum\_array[i] \longleftarrow sum\_array[i] + M[i, j]$;
  **end while**
**end while**
- extract the ID of the sensor with higher $deg\_c$ from $max(sum\_array[])$;
**if** $\exists$ more than 1 with higher $deg\_c$ **then**
  - we choose the sensor with higher energy;
**end if**

**Algorithm 3** Creation of adjacency matrix

Let $s_1, s_2, \ldots, s_n$ be the sensors within the transmission range of sensor $s_k$;
Let $PO_{s_1}, PO_{s_2}, \ldots, PO_{s_n}$ be the position of of the sensors $s_1, s_2, \ldots, s_n$ respectively;
Let $M$ be a matrix with dimensions $(n \times n)$;
**Step 5**
$i \longleftarrow 1$;
**while** $i < n$ **do**
  $j \longleftarrow 1$;
  **while** $j < n$ **do**
    $Ed(s_i, s_j) = \sqrt{(x_{s_i} - x_{s_j})^2 + (y_{s_i} - y_{s_j})^2}$
    **if** $Ed(s_i, s_j) \leq min(PO_{s_i}, PO_{s_J})$ **then**
      - $M[i, j] \longleftarrow 1$;
    **else**
      - $M[i, j] \longleftarrow 0$;
    **end if**
  **end while**
**end while**

### *B. ANN application phase*

After the cluster formation and the election of the cluster head. The control station proceed to the collection of data from the cluster members, then, execute the artificial neural algorithm in order to detect and clean the data from outliers. The proposed ANN algorithm has three layers: one input layer, one hidden layer and one output layer. In addition, the number of nodes in the input layer is equal to the number of cluster members ($cl_j$). In addition, the number of nodes used in the output layer is equal to the number of classes (intervals of values) defined in the training phase. Thus, the trained ANN is constituted by a set of units and a set of

**Algorithm 4** ANN algorithm for a cluster $cl_j$

Let $n$ be the number of the cluster members belong to the cluster $cl_j$;
Let $s_1, s_2, \ldots, s_n$ be the cluster members;
Let $Vec\_D = vec\_d_{s_1}, vec\_d_{s_2}, \ldots, vec\_d_{s_n}$ be the vectors of sensed data of the the cluster members $s_1, s_2, \ldots, s_n$ respectively;
Let $D = d_{s_1}, d_{s_2}, \ldots, d_{s_n}$ be the sensed data of the the cluster members $s_1, s_2, \ldots, s_n$ respectively;
Let $N\_class$ be the output class;
- call $ANN\_training\_procedure(Vec\_D)$;
- $O\_class \longleftarrow$ call $ANN\_testing\_procedure(D)$;
- $N\_class \longleftarrow extract(d_{s_i})$;
**if** $O\_class \cap N\_class \neq \emptyset$ then
  - the sensed data $d_{s_i}$ is normal data;
  - $forward(d_{s_i})$;
**else**
  - the sensed data $d_{s_i}$ is outlier data;
  - $discard(d_{s_i})$;
**end if**

output units. The input units are the data measured by the cluster members $D = d_{s_1}, d_{s_2}, \ldots, d_{s_n}$, where $s_1, s_2, \ldots, s_n$ represent the sensors of the same cluster($cl_j$). While the output represents two class (two intervals of values). In our approach, we use several ANN algorithm for each type of data, ie, we model, train and test each type of data separately, and then we specify for each cluster the corresponding ANN algorithm. The trained ANN is executed offline using a dataset and tested online locally in control station to eliminate outliers instead of passing erroneous data over the network.

After the training phase, the control station can classify the sensed data sent by cluster members. First, it executes the ANN classifier to obtain a class called ($O\_class$), then, it will compare the result (the obtained class $O\_class$) with the normal class ($N\_class$) in which the sensed data is normally belong. In other words, if the $O\_class = [v_q, v_p]$ ($v_q$ and $v_p$ represent the bounds of the interval or the class) and the data $d_{s_i}$ in which the CH would test and classify it, belongs to the same class. So, the data $d_{s_i}$ is considered as a normal (correct) one and the control station will forward it to another processing, otherwise, it is considered as outlier data and the control station will discard it. The algorithm 4 illustrates our ANN-based approach.

## V. Performance Evaluation

In this section, we present the results of applying our proposed approach on real test data in order to demonstrate its performance in the detection of outliers. To do that, we use the open source dataset collected from the Montesinho natural park in the northeast region of Portugal [20]. The data correspond to 517 entries and contain thirteen attributes. Table II shows all the details of the used dataset. In our experiments, we focused only on the temperature attribute. We added 10%, 20%, 30%, 40% and 50% of outliers manually (extreme values) to the previous dataset in order to test our approach on detecting outliers. In addition, we split the dataset into training and testing part for 60% and 40% respectively. We compare our experiment results to the models proposed by Nesa et al. [6]. The authors presented four classification models LDA (Linear Discriminant), RF (Random Forest), CART (Classification And Regression Trees) and GBM (Gradient Boosting Machine).

The present simulations were implemented using MATLAB Neural Networks Toolbox [21]. We employ a multi-layer perceptron (MLP) with one hidden layer which seems a good architecture for classification tasks. In addition, in order to train the ANN, the backpropagation technique is used [22] and the $Sigmo\ddot{\imath}de$ function is considered as the function of activation in our approach. To evaluate the performance of our proposed approach, we use a confusion matrix. This matrix is often used to demonstrate the performance of a classifier basing on a set of test data. The confusion matrix gives the values of True Negative (TN), True Positive (TP), False Negative (FN) and False Positive (FP). These values allow a careful evaluation the approach where the following metrics can be computed:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$F - score = \frac{2 * TP}{2 * TP + FP + FN} \tag{7}$$

The accuracy computed for our proposed approach as well as for the statistical models proposed by Nesa et al. [6] are presented in Fig. 3. As known, an optimal outlier detection algorithm has the ability to identify and detect all existing outliers in data. As Fig. 3 reveals that our proposal is visibly better. Actually, it outperforms the compared statistical models (CART, RF, GBM, LDA) and its accuracy rate is above 98%. The LDA model shows poor results compared to the other proposed approaches. Even in the case of the percentage of outliers in the dataset is increased, our approach works better.

TABLE II: Dataset description

| Attribute | Description | Values |
|---|---|---|
| X | x-axis coordinate | from 1 to 9 |
| Y | y-axis coordinate | from 1 to 9 |
| month | Month of the year | January to December |
| day | Day of the week | Monday to Sunday |
| ISI | Initial Spread Index | 0.0 to 56.10 |
| temp | Outside temperature (in C) | 2.2 to 33.30 |
| RH | Outside relative humidity (in %) | 15 to 100 |
| wind | Outside wind speed (in km/h) | 0.40 to 9.40 |
| rain | Outside rain (in mm/m2) | 0.0 to 6.4 |
| area | Fire detected/ not detected | 1/0 |

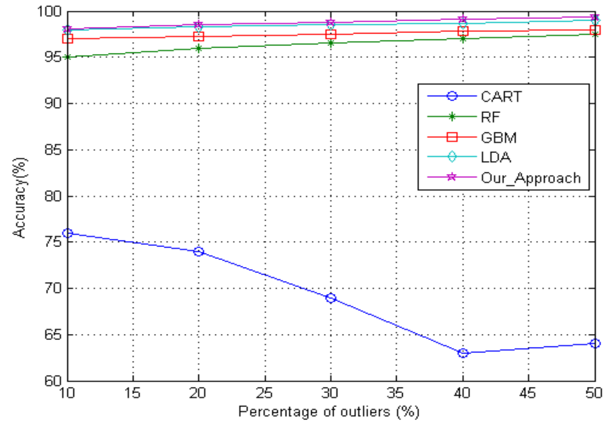


Fig. 3: Accuracy vs. Percentage of outliers.

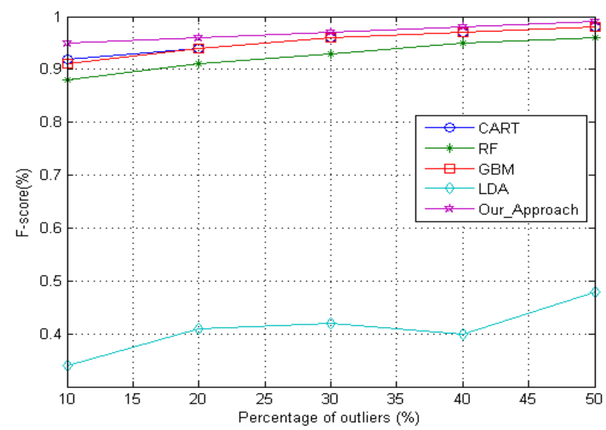


Fig. 4: F-score vs. Percentage of outliers.

Besides the evaluation of the accuracy metric, we compare our proposal using the f-score metric. Fig. 4 demonstrates the results of simulations. The curves show that there is a clear trend that our approach presents better values comparing to the other algorithms. These good results come from the idea of delegating the execution of the ANN classifier to the control station by considering the clusters' creation phase developed in algorithm 2. In fact, the control station detects outliers by categories in order to exploit the spatial correlation that exists between the sensed data of the same region (room, floor, etc) and the same type (temperature, humidity, etc).

The two figures (Fig 5 and Fig 6) depict histograms of recall and precision respectively of our proposal and the compared approaches. The ideal performance of a technique is where the recall and the precision are maximized. We notice that by far most recall and precision values of our approach are around 1 (over 0.9). This shows that our proposal outperforms the other models (CART, RF, GBM, LDA). Note that our approach first creates clusters in order to exploit the spatial correlation that exists in measurements and then, the control

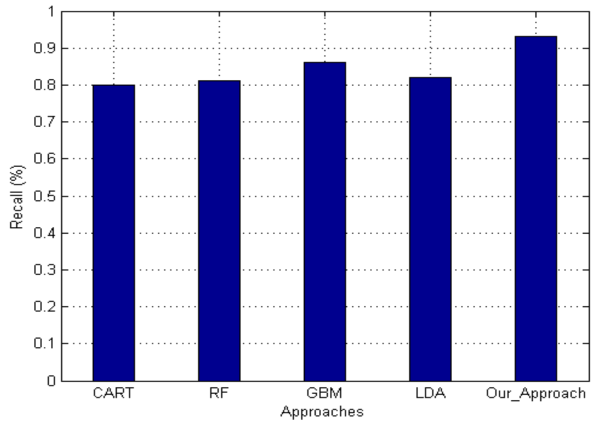


Fig. 5: Recall results for the different approaches.

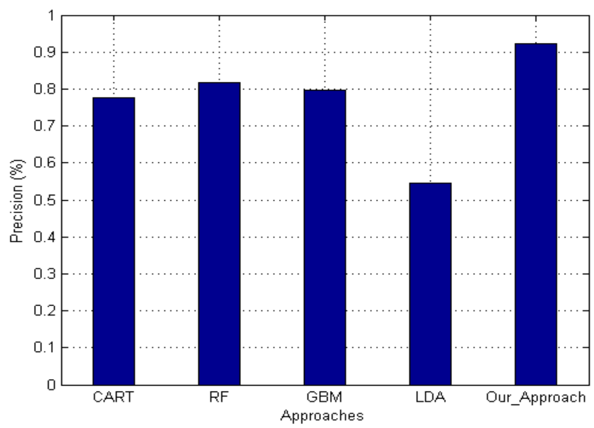


Fig. 6: Precision results for the different approaches.

station uses an ANN model specific for each type of data to identify and detect outliers in the sensed data received from the sensors. This indicates that using ANN models in a distributed manner for the outlier detection avoids the large deviations in the measured values of different data types in an IoT system which influences the outlier detection accuracy.

## VI. Conclusion

IoT-based systems bring great benefits to end-users to improve the quality of service of various applications, such as in smart homes. In order to achieve this, an IoT-based system should provide reliable and accurate data to the control station. Therefore, the sensed data must be cleaned from some erroneous items, called outliers. In this paper, we proposed an ANN-based approach to detect these outliers in the context of the Internet of Things. The ANN model is trained and tested using real datasets and the results revealed its higher efficiency under different metrics to comparable models. As future work, investigations of other variants of artificial neural networks, in order to achieve a high detection accuracy.

## References

[1] T. Malche and P. Maheshwary, "Internet of things (iot) for building smart home system," in *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, pp. 65–70, Feb 2017.

[2] K. Su, J. Li, and H. Fu, "Smart city and the applications," in *2011 International Conference on Electronics, Communications and Control (ICECC)*, pp. 1028–1031, Sep. 2011.

[3] S. Abraham, J. Beard, and R. Manijacob, "Remote environmental monitoring using internet of things (iot)," in *2017 IEEE Global Humanitarian Technology Conference (GHTC)*, pp. 1–6, Oct 2017.

[4] P. Raja and S. Bagwari, "Iot based military assistance and surveillance," in *2018 International Conference on Intelligent Circuits and Systems (ICICS)*, pp. 340–344, April 2018.

[5] D.Hawkins, *Identification of Outliers*. Chapman and Hall, 1980.

[6] N. Nesa, T. Ghosh, and I. Banerjee, "Outlier detection in sensed data using statistical learning models for iot," in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, April 2018.

[7] C. Titouna, F. Naït-Abdesselam, and A. Khokhar, "A novel data cleansing approach for sensitive applications of wireless sensor networks," in *2019 International Conference on Smart Applications, Communications and Networking (SmartNets)*, pp. 1–6, 2019.

[8] C. Titouna, F. Nait-Abdesselam, and A. Khokhar, "A multivariate outlier detection algorithm for wireless sensor networks," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, pp. 1–6, May 2019.

[9] M. Ernst and G. Haesbroeck, "Comparison of local outlier detection techniques in spatial multivariate data," *Data Mining and Knowledge Discovery*, vol. 31, pp. 371–399, Mar 2017.

[10] H. Liu, X. Li, J. Li, and S. Zhang, "Efficient outlier detection for high-dimensional data," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, pp. 2451–2461, Dec 2018.

[11] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han, "Outlier detection for temporal data: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, pp. 2250–2267, Sep. 2014.

[12] D. R. F. M. M. Pietro and Z. Jihane, "A comparative evaluation of outlier detection algorithms: Experiments and analyses," *Pattern Recognition*, vol. 74, pp. 406 – 421, 2018.

[13] A. Kanev, A. Nasteka, C. Bessonova, D. Nevmerzhitsky, A. Silaev, A. Efremov, and K. Nikiforova, "Anomaly detection in wireless sensor network of the "smart home" system," in *2017 20th Conference of Open Innovations Association (FRUCT)*, pp. 118–124, April 2017.

[14] G. Stamatescu, C. Drăgana, I. Stamatescu, L. Ichim, and D. Popescu, "Iot-enabled distributed data processing for precision agriculture," in *2019 27th Mediterranean Conference on Control and Automation (MED)*, pp. 286–291, July 2019.

[15] H. D. P. d. Santos, A. H. D. P. S. Ulbrich, V. Woloszyn, and R. Vieira, "Ddc-outlier: Preventing medication errors using unsupervised learning," *IEEE Journal of Biomedical and Health Informatics*, vol. 23, pp. 874–881, March 2019.

[16] H. Ghallab, H. Fahmy, and M. Nasr, "Detection outliers on internet of things using big data technology," *Egyptian Informatics Journal*, 2019.

[17] X. Deng, P. Jiang, X. Peng, and C. Mi, "An intelligent outlier detection method with one class support tucker machine and genetic algorithm toward big sensor data in internet of things," *IEEE Transactions on Industrial Electronics*, vol. 66, pp. 4672–4683, June 2019.

[18] R. Zhu, X. Ji, D. Yu, Z. Tan, L. Zhao, J. Li, and X. Xia, "Knn-based approximate outlier detection algorithm over iot streaming data," *IEEE Access*, vol. 8, pp. 42749–42759, 2020.

[19] A. Abraham, *Artificial Neural Networks*, ch. 129. American Cancer Society, 2005.

[20] A. Asuncion and D. Newman, "Uci machine learning repository," *Irvine, CA: University of California, Department of Information and Computer Science*, 2007.

[21] "Mathworks, matlab neural network toolbox," https://fr.mathworks.com/products/deep-learning.html, 2016.

[22] M. Riedmiller, "Advanced supervised learning in multi-layer perceptrons from backpropagation to adaptive learning algorithms," *Computer Standards & Interfaces*, vol. 16, no. 3, pp. 265–278, 1994.

[23] A. Abid, A. Masmoudi, A. Kachouri, and A. Mahfoudhi, "Outlier detection in wireless sensor networks based on optics method for events and errors identification," *Wireless Personal Communications*, vol. 97, pp. 1503–1515, Nov 2017.