ISSN(Online) : 2456-8910
International Journal of Innovative Research in Applied Sciences and Engineering (IJIRASE)
Volume 4, Issue 3, DOI: 10.29027/IJIRASE.v4.i3.2020.675-680, September 2020

Machine Learning Based Trust Routing for Clustered IoT devices

Aishwarya Kurle[1], Dr K.R.Radhika[2]
[1]*Student, Computer Networks Engineering, Department of ISE*
[2]*Professor, Department of ISE*
B.M.S College of Engineering, Bangalore, India
Email : aishwarya.scn18@bmsce.ac.in

Abstract - With the increase in the number, types of sensors and their applications, the number of devices connected wirelessly to the internet are huge, bringing IoT technologies to the forefront. In a world where new wireless devices are added into the network every second, security and trust are of crucial importance. In this paper, a method to compute trust of the sensor node is proposed and the best route, among multiple routes, is selected to send a packet from source node to destination node via the most trusted route. The sensor data that is considered to decide the trust level of the nodes and best path are response time or end-to-end delay, many hops, energy consumed, residual energy, etc. The sensor network is a non-hierarchical network where the nodes are distributed over an area, the nodes are then clustered using the DBSCAN algorithm. The source node sends packets to all its neighbouring nodes and waits for the reply, based on the parameters stated above it selects the next-hop and keeps adding the trust values for that selected route. The process continues until it reaches the destination. The list of multiple routes based on cumulative trust value is found and the one with the highest trust is selected. This route can then be chosen for packet transfer.

Keywords - Internet of things (IoT), sensor nodes, trust, Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm.

I. Introduction

Internet of Thing (IoT) and wireless communication has brought about tremendous connectivity among devices over the internet. Sensors form a large part of this network and can deliver real-time data. With the increase in the number of sensor devices, the types of sensors available and their various applications, sensors are now part of our daily life and all of us reap the benefits. IoT taps into many areas that include remote patient monitoring, home automation, smart cities, wearable technology, motion sensor gaming, connected vehicles, security, agriculture, infrastructure applications, etc. With IoT gaining so much momentum and so many devices connected to the internet, trust and security are of vital importance and play a crucial role in data reliability. An efficient trust management model must be incorporated into every IoT system to protect the system against malicious attacks and thereby ensuring reliable and secure data transmission. It helps to overcome perceptions of uncertainty and risk and engages in user acceptance and consumption on IoT services and applications [1]. Trust among IoT devices in this paper is given by the reliable transmission of packets from source to destination via nodes which have high trust levels. Sensor data that is considered in the computation of trust levels are as follows - response time or end-to-end delay, several hops, energy consumed and residual energy. Response time is the time taken by the node to respond to the communication from its neighbour, End-to-end delay is the cumulative response time of all intermediate nodes on the path from source to destination, several hops is the number of intermediate nodes the packet from source to destination is routed through, the energy consumed is the amount of energy that is taken up by the node for packet generation and transmission and residual energy is the remaining battery power of that node.

In this paper, we use a non-hierarchical network, on which we run a machine learning clustering algorithm called Density-Based Spatial Clustering of Applications with Noise (DBSCAN). This is an unsupervised learning algorithm which groups data into clusters based on the density of clusters. DBSCAN takes into consideration the min_samples - minimum number of points needed to form a cluster and the epsilon value - the radius around a point for which it will check for other points to be part of the cluster. The cluster contains core points and border points, core points are the points with min_samples number of points in its epsilon radius and border point are those that are part of a core points epsilon radius. The unclustered points are considered as noise [8].

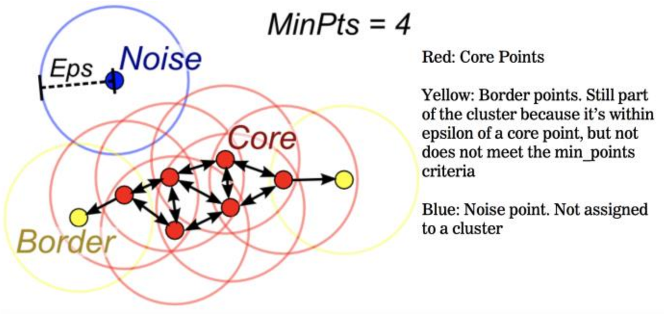
MinPts = 4
Eps
Noise
Core
Border
Red: Core Points
Yellow: Border points. Still part of the cluster because it's within epsilon of a core point, but not does not meet the min_points criteria
Blue: Noise point. Not assigned to a cluster

Fig 1: classification of points as noise or as part of a cluster (core and border points), using DBSCAN algorithm

Rest of this paper explains how clustering can be applied to a field of non-hierarchical IoT nodes and the difference it could make to the selected path. In section II we see the related work of authors on trust issues and mechanisms in IoT devices and networks. Section III provides an overview of a proposed system using clustering and the trust model and its design in the paper. Section IV contains the simulation carried out and results obtained. Finally, Section V presents applications, conclusions and future work.

II. Related Works

Numerous numbers of interconnected devices over the internet raise the issues of security and trust among them. These technologies have been widely studied and investigated. As presented in 'A Classification of Trust Computation Models for Service-Oriented Internet of Things Systems' [2], the authors classify existing trust computation models based on five design dimensions: trust composition, trust propagation, trust aggregation, trust update, and trust formation. They also draw attention to the advantages, drawbacks and the gaps in IoT trust computation research. In [3], the authors propose a novel adaptive filtering technique to determine the best way to combine direct trust and indirect trust dynamically to minimize convergence time and trust estimation bias in the presence of malicious nodes performing opportunistic service and collusion attacks. The difference between trust and security, even though the two terms are used interchangeably, and trust and reputation are explained by the authors in [4], The factors affecting trust updating are summarized and some examples of the systems in which these factors have been implemented are also given. Since sensors are mainly deployed to monitor events and report data new trust models to combine the data trust and the communication trust to infer the total trust

In [5], we see that the author mentions the motivation of providing a trust management system for IoT systems: There are misbehaving owners and consequently misbehaving devices that may perform discriminatory attacks based on their social relationships with others for their gain at the expense of other IoT devices which provide similar services. On the other hand, social interactions among objects disclose the valuable information of trust in analogy to the sociology concept of human interactions based on trust relationships. In this regard, authors in [6] have developed a social model of cyber objects corresponding to their owner's social behaviour. In such models' objects interact with each other based on their trust relationships and reveal any information in terms of the trust. They discuss trust assessment of a social network based on concepts like a community of interest, friendship, followers as well as frequency, duration and behaviour of the objects. There are many things that we need to take into consideration before we build a trust model and implement it into our network. All the above author's papers show us a step forward into what parameters must be considered and how they need to be applied.

III. System Design and Implementation

In this paper, we consider a network of a hundred randomly scattered IOT nodes over a field of hundred square units. We then run the Dbscan algorithm on these nodes to group them into clusters. The objective of this system is to send data from one node to another node through trusted nodes. The strategy of selecting a trusted node is that the intermediary nodes should pass the data in its original form without tampering.

A. Assigning trust level to the nodes:

Initially, we use Kmeans, to classify the nodes into high, medium and low trust based on the sensor data. Kmeans is an iterative algorithm that attempts to divide the dataset into K pre-defined distinct subgroups. Here K value will be 3 since we need 3 subgroups that are high, medium and low.

B. Route discovery process:

When the source node initiates packet transmission, the packet is sent to all the nodes in its transmission range and the node with fast response, high trust and high residual energy is chosen as the next node in the path. This selected next node or the intermediate node repeats the same process of neighbour discovery and selection based on trust. This way we get multiple routes from source to destination via intermediate nodes. The path with the highest trust level is considered as the best route and is selected to route packets, as shown in Fig 2. The trust values of the nodes are incremented with successful delivery and decremented with failure to deliver [7].

The DBSCAN clustering comes into play while selecting the next intermediary or the next hop. We apply a condition wherein there cannot be 4 hops within the same cluster. This can change depending on the number of nodes in the network and the area over which the network is spread. Since we have selected a network with 100 nodes, spread over an area of hundred square units and each node having a transmission range of 30 units, the DBSCAN parameters of min_samples=2 and epsilon=10 hold feasible. In a different topology, it should be ensured that the transmission range is greater than twice than that of the epsilon value.

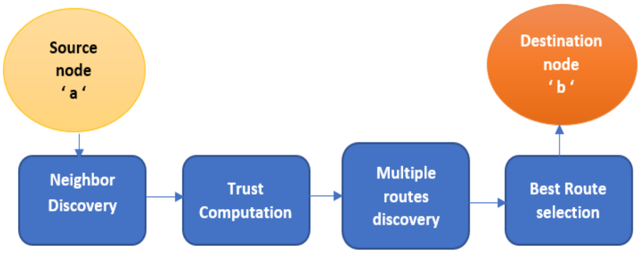

Figure 2: Trust-based route selection from source to destination node

C. Simulation:

The simulation is done in MATLAB since it integrates computation, programming and visualization in a user-friendly environment. It is a collaborative system whose basic data component is an array that does not require dimensioning. This allows users to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time that it would take in a scalar noninteractive language such as C or Fortran.

IV. Results and Discussions

The results obtained from MATLAB have been discussed below with figures. For comparative analysis, we make use of another simple trust-based computation model- Eigen trust model, given in Fig 3.

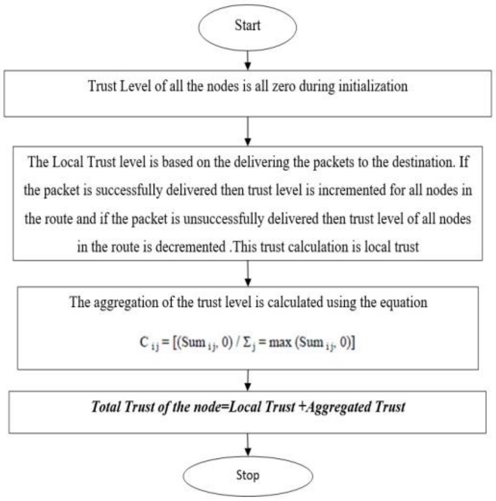
Start
Trust Level of all the nodes is all zero during initialization
The Local Trust level is based on the delivering the packets to the destination. If the packet is successfully delivered then trust level is incremented for all nodes in the route and if the packet is unsuccessfully delivered then trust level of all nodes in the route is decremented .This trust calculation is local trust
The aggregation of the trust level is calculated using the equation
C ij = [(Sum ij, 0) / Σ j = max (Sum ij, 0)]
Total Trust of the node=Local Trust +Aggregated Trust
Stop

Figure 3: Eigen Method of Trust Level Computation

A. Input Parameters for Simulation and node deployment:

The following two tables give the input parameters to both the models. Table 1 gives the input parameters to the Eigen trust model and Table 2 gives the input parameters to the Cluster-based trust model. Fig 4 shows the network of nodes simulated with the given parameters and Fig 5 shows the nodes after clustering. We also get the number of clusters and the cluster IDs that play a role is the route selection process as stated in the previous section.

TABLE I. Eigen Trust Model Input

Sl No	Parameter name	Value
1	Number of nodes in the network	100
2	Source node	44
3	Destination node	35
4	Transmission range	30
5	Energy required for generation in mJ	10
6	Energy required for transmission in mJ	20
7	Attenuation factor (0.1 to 1)	0.6
8	Initial battery power in mJ	2000

TABLE II. Cluster-based Trust Model

Sl No	Parameter name	Value
1	Number of nodes in the network	100
2	Minimum points	2
3	Epsilon value	10
4	Source node	44
5	Destination node	35
6	Transmission range	30
7	Energy required for generation in mJ	10
8	Energy required for transmission in mJ	20
9	Attenuation factor (0.1 to 1)	0.6
10	Initial battery power in mJ	2000

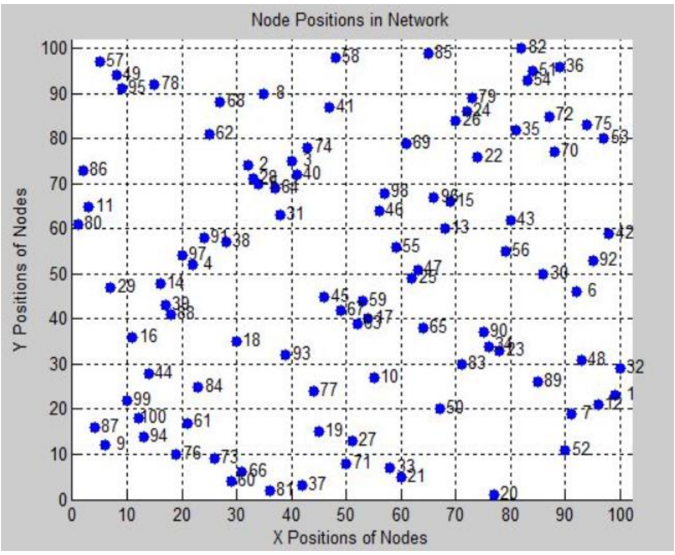

Figure 4: Network of nodes simulated

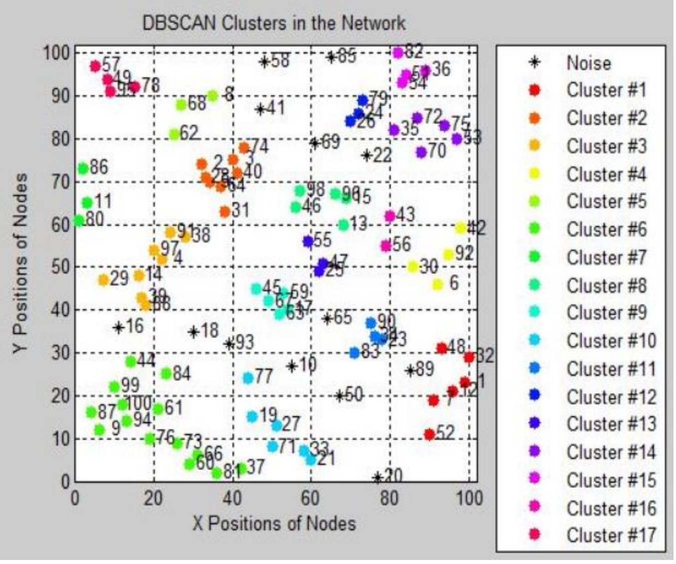

Figure 5: Clustering of nodes using the DBSCAN algorithm

B. Multiple Paths obtained showing trust levels:

List of all paths found from source to destination with the cumulative trust values for each path is shown below for both the models. Here it can be noticed that the trust values obtained by the paths in the cluster-based trust model are higher than the ones obtained by the Eigen trust model.

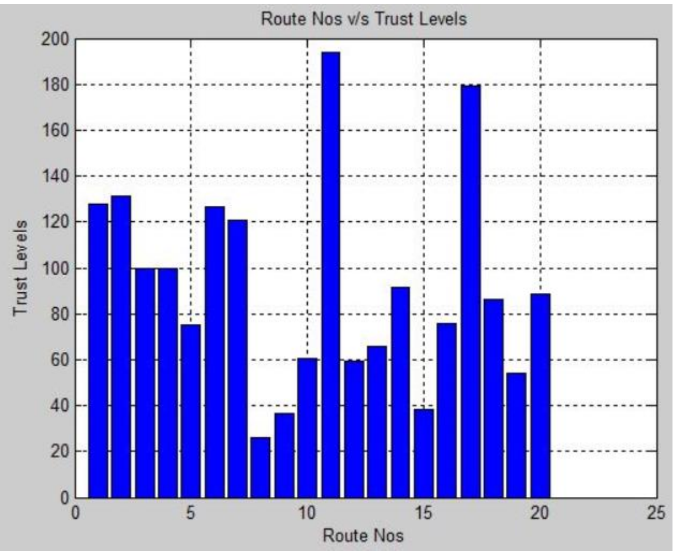

Figure 6: Eigen Trust Model: paths found and trust values of each path

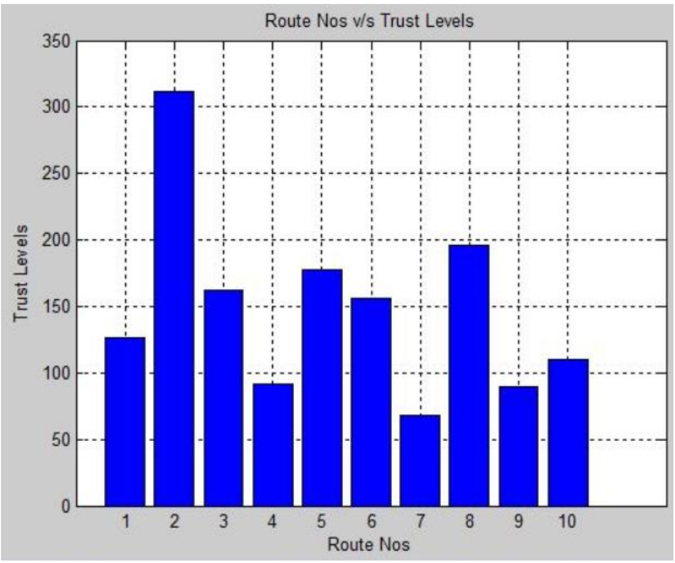

Figure 7: Cluster-Based Trust Model: paths found and trust values of each path

C. Selected route for packet transfer:

As shown above, in Fig 6, the best route with the highest trust value for Eigen trust model is route number 11. Fig 8 shows the hops in the route taken. Similarly, Fig 7 shows the best route with the highest trust value for cluster-based trust model is route 2, Fig 9 shows the hops in the route taken. It is noticeable that the number of hops in the cluster-based trust model is much lesser than the number of hops in the Eigen trust model.

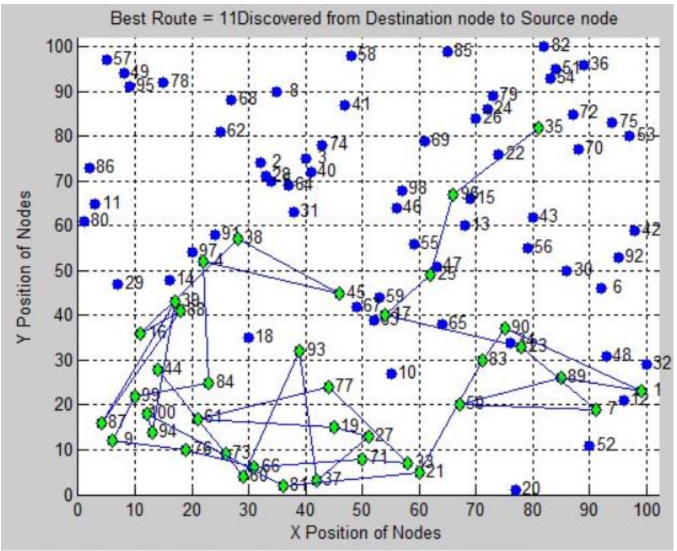

Figure 8: Eigen Trust Model: Path from source to destination showing the number of hops

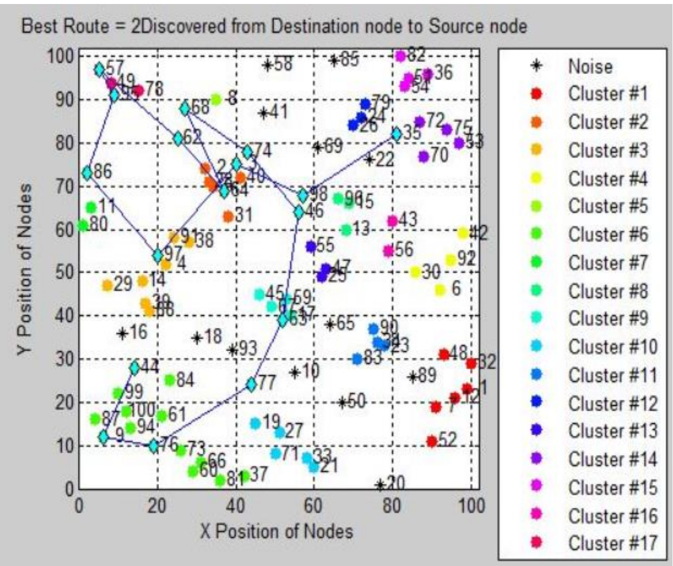

Figure 9: Cluster-Based Trust Model: Path from source to destination showing the number of hops

D. Residual Energy:

Residual energy is the remaining battery power of the node. Results for Cluster-Based model show that there is a reduction in the network power consumption and an increase in the lifetime of all nodes. With IoT nodes, one of the main issues lies in the limited lifetime due to dependence on batteries. The following figures show the difference in the residual energy of the nodes in the network after the packet transfer has taken place. The performance is much better with the cluster-based model as compared to eigen trust model.

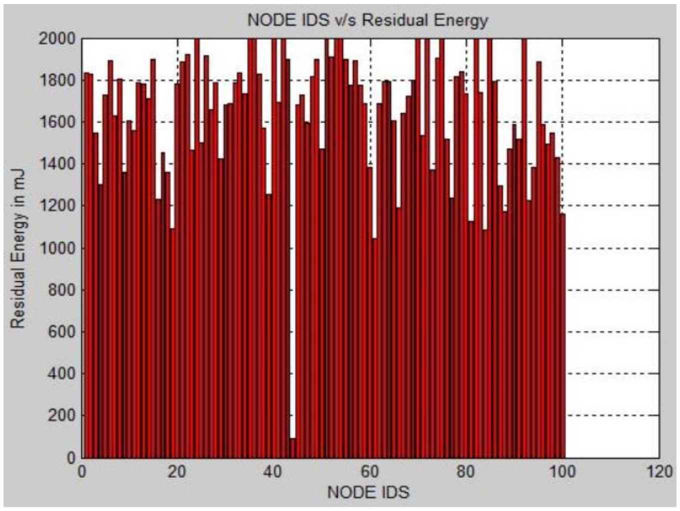

Figure 10: Eigen Trust Model: Residual Energy

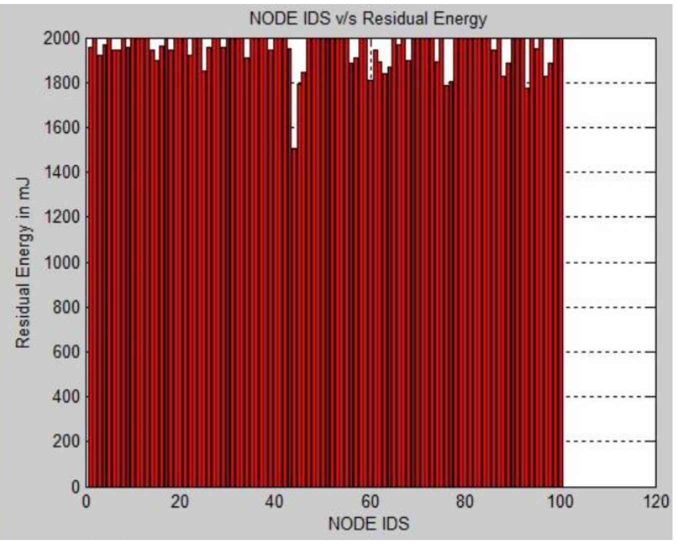

Figure 11: Cluster-Based Trust Model: Residual Energy

V. Conclusions and Future Scope

Conclusion: Since the major issues faced by sensor nodes is the short lifetime due to dependence on batteries and the trust between devices, we need to find methods to increase both the lifetime and the trust levels of the nodes without compromising anyone to increase the other. Experimenting with clustering of the nodes and applying conditions in the way the route is selected and trust is gained seems to be the way forward to find a balance. While doing this we need to keep in mind that different topologies may need different parameter settings and this depends on the number of nodes in the network, area over which the nodes are deployed, the density of the nodes, environmental conditions on which sensor readings depend, etc.

Future scope: This work can be extended to hierarchical networks in which one can make use of the Cluster Head node to manage the trust and routing between clusters accurately. We can also identify the core nodes with the help of the DBSCAN algorithm and enable routing only between the core nodes or also vary the number of hops between clusters and within clusters to compare performance.

References

[1] Yan, Z., and P. Zhang. "&amp; Vasilakos, AV (2014). A survey on trust management for internet of things." *Journal of Network and Computer Applications* 42.

[2] Guo, Jia, and Ray Chen. "A classification of trust computation models for service-oriented internet of things systems." In *2015 IEEE International Conference on Services Computing*, pp. 324-331. IEEE, 2015.

[3] Chen, Ray, Jia Guo, and Fenye Bao. "Trust management for SOA-based IoT and its application to service composition." *IEEE Transactions on Services Computing* 9, no. 3 (2014): 482-495.

[4] Momani, Mohammad, and Subhash Challa. "Survey of trust models in different network domains." *arXiv preprint arXiv:1010.0168* (2010).

[5] Yan, Zheng, Peng Zhang, and Athanasios V. Vasilakos. "A survey on trust management for Internet of Things." *Journal of network and computer applications* 42 (2014): 120-134.

[6] Holmquist, Lars Erik, Friedemann Mattern, Bernt Schiele, Petteri Alahuhta, Michael Beigl, and Hans-W. Gellersen. "Smart-its friends: A technique for users to easily establish connections between smart artefacts." In *international conference on Ubiquitous Computing*, pp. 116-122. Springer, Berlin, Heidelberg, 2001.

[7] Jayasinghe, Upul, Gyu Myoung Lee, Tai-Won Um, and Qi Shi. "Machine learning based trust computational model for IoT services." *IEEE Transactions on Sustainable Computing* 4, no. 1 (2018): 39-52.