Scalable Computing: Practice and Experience

Scientific International Journal for Parallel and Distributed Computing

ISSN: 1895-1767

Universitatea de Vest
din Timişoara

Volume 20(4)

December 2019

SUBSCRIPTION INFORMATION: please visit http://www.scpe.org

Scalable Computing: Practice and Experience

Volume 20, Number 4, December 2019

TABLE OF CONTENTS

Special Issue on Smart Sensors for Scalable Computing:

Regular Papers:

DOI 10.12694/scpe.v20i4.1472
Scalable Computing: Practice and Experience ISSN 1895-1767
Volume 20, Number 4, pp. iii–iv. http://www.scpe.org © 2019 SCPE

INTRODUCTION TO THE SPECIAL ISSUE ON SMART SENSORS FOR SCALABLE COMPUTING

Various ICT applications, whether for city infrastructures, factories, or wearable devices, use large arrays of sensors collecting data for transmission over the Internet to a central, cloud-based computing resource. Analytics software running on the cloud computers reduces the huge volumes of generated data into actionable information for users, and commands to actuators back out in the field. So, smart sensors are one key factor in various ICT Application successes but these are not conventional types, which simply convert physical variables into electrical signals. They have needed to evolve into something more sophisticated to perform a technical, scalable and economical viable role within the ICT environment.

This special issue is devoted to the latest research in smart sensors for scalable computing in various application area such agriculture and Industries 4.0.

Ashish Kr. Luhach, The PNG Univeristy of Technology, Papua New Guinea
Dharm Singh Jat, Namibia's University of Science and Technology, Namibia
Kamarul Hawari bin Ghazali, Universiti Malaysia Pahang, Pekan, Malaysia

DOI 10.12694/scpe.v20i4.1553
Scalable Computing: Practice and Experience ISSN 1895-1767
Volume 20, Number 4, pp. 591–597. http://www.scpe.org © 2019 SCPE

HONEY BEE OPTIMIZATION BASED SINK MOBILITY AWARE HETEROGENEOUS PROTOCOL FOR WIRELESS SENSOR NETWORK

ASHISH KR. LUHACH[*] ADITYA KHAMPARIA[†] RAVINDRA SIHAG[‡] AND RAJ KUMAR [§]

Abstract. Wireless Sensor Network (WSN) has emerged as one of the most important technologies serving an array of solutions for critical applications such as defense, industrial monitoring and decision purposes. Data routing in WSN is effective or non-effective depending upon the energy saving for nodes while transferring data packets to the sink. Mainly WSN divided into two modes; heterogeneous and homogeneous. Heterogeneous network in WSN mainly focused on the cluster head selection. Sink mobility in the heterogeneous network has still many open research issues, it is observed that it makes the network more energy efficient. The optimization in the network leads to the stability of the network at a much higher level. In this paper, the sink mobility is optimized for WSN using Honey Bee Optimization (HBO) technique by considering the parameters such as energy and distance. The proposed protocol shows significant improvement in the stability period by 33 % by covering 2928 rounds and enhanced network lifetime by 1500 rounds in compared with 2033 and 14084 rounds for iMBEENISH protocol respectively.

Key words: Honey Bee Optimization, Wireless Sensor Network, Swarm intelligence, LEACH protocol.

AMS subject classifications. 68M10, 92B20

1. Introduction.

Wireless sensor networks (WSN) sometimes also called as wireless sensor and actor networks (WSAN) are spatially distributed autonomous sensors network to monitor physical or environmental conditions such as temperature, sound, pressure, etc. for critical application such as defense and e-healthcare [2, 17, 18]. The modern sensor networks are bi-directional in the data flow, which means the data can be transmitted from nodes to sink and vice versa enabling control of sensor activity [19, 14, 7].

The properties of WSN are explained below.

- Infrastructure less: WSN has not any fixed infrastructure, which results in low communication overhead and the network has not forced to follow any specific topology for communication.
- Multi-Hoped: WSN comprises of multiple nodes and the different nodes communicate with each other through different routes to reach each other. In this topology, the data packet travels from one node to another node to reach its final destination. In this way, it helps in energy conservation which is one of the most prominent concerns in WSN [10].
- Requirement of Location information in case of Location Base Algorithm: In case of WSN, the location information is required according to the application for which it is being deployed. Generally, the location information is required in case of geographical based routing protocols. For this purpose, Global Positioning System (GPS) is installed in the sensor nodes. From the installed GPS, they find the distance from the other node and data is sent accordingly.
- Network size: WSN randomly deployed in the remote and large area to sense the information where human intervention is not possible. Generally to cover a large area the number of the node is also more and also a number of the nodes depend on the application for it is used.
- Homogeneous Network and Heterogeneous Networks: WSN may be homogeneous or heterogeneous depending upon the application for which it is used. Heterogeneous networks perform energy efficient routing.

This research work focused on a heterogeneous network of WSN as various clustering topology assign different roles for data collection to the different nodes. The cluster head will consume more energy than the rest of the nodes as it performs more function and operation than others. The research conducted for the same recommends, electing the node having more energy as a cluster head [4] and making it feasible to preserving more energy for the network.

[*] The PNG University of Technology, Papua New Guinea (ashishluhach@acm.org).
[†] Lovely Professional University, Punjab, India
[‡] Banasthali University, Rajasthan, India
[§] The PNG University of Technology, Papua New Guinea

2. Related Work. Due to advancement in the field, a lot of research conducted to improve the network lifetime for wireless sensors and energy conversion is one of the most important challenges faced by the researcher. The network stability gets achieved with help of embedding homogeneity in existing networks. To enhance the network lifetime and introduce a variety of energy level nodes in network Stable Election Protocol (SEP) [16] framework was introduced. In this protocol, two types of nodes exist i.e. normal and advance one where an advance node has (1+a) times effective energy than normal one. It helps in enhancing the stability of node i.e. increase the gap time interval before the previous node gets expired. The probability of nodes to be cluster heads is shown in equation 2.1:

$$P = \begin{cases} \dfrac{P_{opt}}{1+am} & \text{for normal nodes;} \\ \dfrac{(p_{opt}(1+a))}{(1+am)} & \text{for advance nodes;} \end{cases} \tag{2.1}$$

If two level distributions considered then SEP not retrieved results and get failed. Distributed Energy Efficient Clustering (DEEC) [12] proposed ration of nodes average energy obtained with the residual energy left. DEEC also worked on two levels of energy like SEP. Advanced nodes are having $(1+a)$ times energy than normal one. The probabilistic formula for selection of CH is given in equation 2.2:

$$P = \begin{cases} \dfrac{((p_{opt}E_i(r))}{((1+am)E(r))} & \text{for normal nodes;} \\ \dfrac{(p_{opt}(1+a)E_i(r))}{((1+am)E(r))} & \text{for advance nodes;} \end{cases} \tag{2.2}$$

where $E_i(r)$ is the residual energy of the node i at round $r, E(r)$ is the average energy at round r of the network which is determined a priori before the nodes are deployed in the network. The formation of cluster heads using advance nodes suffered from penalized effect without consideration of energy reservoir. Developed Distributed Energy-Efficient Clustering (DDEEC) [5] decides the selection of available nodes from cluster heads which are working equally with existing nodes with help of threshold limit. Two levels of heterogeneity protocols proposed where, Energy Efficient Heterogeneous Clustering (EEHC) [9] was the first one who introduced three levels of heterogeneity, containing normal nodes, advanced nodes, and super nodes. The EEHC has 10% performance improvement over protocol Low Energy Adaptive Clustering Hierarchy (LEACH) [6]. Enhanced Developed Distributed Energy-Efficient Clustering (EDDEEC) [15] worked on threshold values which avoid the energy-rich nodes penalizing by existing three levels of nodes formation. Balanced Energy Efficient Network Integrated Super Heterogeneous Protocol (BEENISH) [13] introduced ultra level nodes heterogeneity which comprised of four levels with the involvement of normal, super and advanced nodes. There have been various techniques based on HBO for the mobility of sink in homogeneous WSN [3, 11]. However, for heterogeneous WSN the mobility of sink is still not touched much.

3. Honey Bee Algorithm for sink mobility in Heterogeneous WSN. Honey Bees algorithm performs random search along with the neighborhood search for both functional and combinatorial optimizations. As shown in Figure 3.1, the main aim of this algorithm is to find an optimal solution by the honey bees natural foraging behavior. Here, various parameters are required in general i.e. scout bees (n), selected sites in visited sites (m), stopping criteria, best sites in selected sites (e), initial patch size that includes the size of the network and its neighborhood, bees for selected sites, bees for ($m-e$) sites [8]. Bees are randomly placed in a space and then the evaluation of bee's fitness is done. Now, the bees with the highest fitnesses are the selected bees and the bees that visit the sites are selected for the neighborhood search. Now for the selected sites, recruit bees and evaluate fitness. Fittest bees from each patch are selected. Remaining bees are randomly assigned in search space and then their fitness is evaluated. The steps are further repeated until the stopping criterion is met. The bees algorithm is used in various applications such as data clustering, pattern recognition in neural networks, engineering.

In a sensor network, the nodes lying near the sink have to forward the data of their own along with the data of the nodes which are far away from the sink, as a consequence of which the nodes nearby the sink got depleted in terms of their energies. This depletion of the energy of the nearby nodes results in the network isolation or in

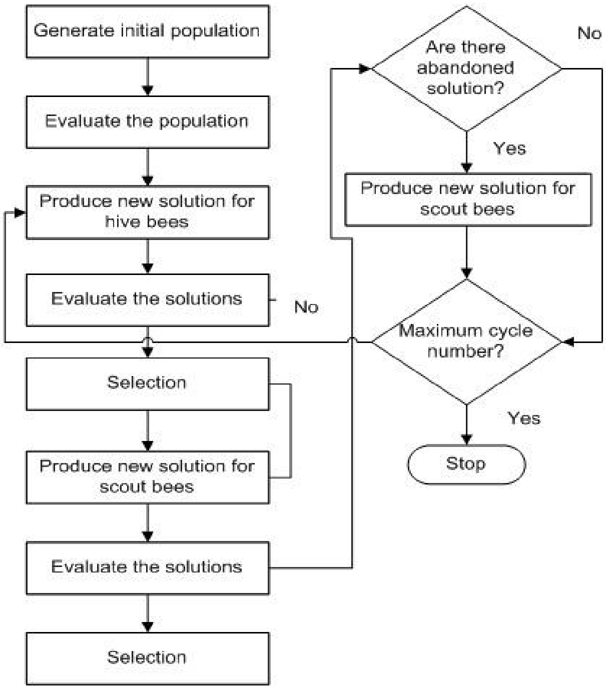

Generate initial population
Evaluate the population
Produce new solution for hive bees
Evaluate the solutions
No
Selection
Produce new solution for scout bees
Evaluate the solutions
Selection
Are there abandoned solution?
No
Yes
Produce new solution for scout bees
Maximum cycle number?
Yes
Stop

FIG. 3.1. *Honey Bees algorithm*

other words the HOT SPOT problem. Using sink mobility this problem will be mitigated to a significant level i.e. the energy consumption of the nearby nodes is balanced. There are also some biological protocols which are used to enhance the performance of the sensor network in terms of network lifetime, throughput and quality of service.

4. Proposed Protocol: HBO-iMBEENISH. In this protocol, the research aims to improve mobility aware BEENISH protocol [1]. In this four level of heterogeneity like in BEENISH consisting of normal nodes, advanced nodes, and ultra-nodes. This is the first time any protocol used four levels of energy heterogeneity in the network. $P(i)$ is the probability of the node to become the cluster head and is expressed differently by the different protocols like DEEC, EDEEC, and DDEEC etc. More efficient the probabilistic selection of cluster head more chances of the network having a much better life. As mentioned in the section of problem definition there is penalization of more energy enriched nodes to stay as cluster head which tends to drain their energy much faster when their energy becomes equal to the normal nodes. The concept of a threshold is being applied in the EDDEEC for the three levels of energy heterogeneity but never been done the same in the four levels.

So the new proposed probabilistic equation for the selection of cluster head is given below:

$$P(i) = \begin{cases} \dfrac{Popt * Ei}{(1+m(a+m0(-a+b+m1(-b+u)))*Ea)} & \text{for normnodes if } E(i) > Tab \\ \dfrac{Popt * Ei * (1+a))}{(1+m(a+m0(-a+b+m1(-b+u)))*Ea)} & \text{for advnodes if } E(i) > Tab \\ \dfrac{(Popt * Ei * (1+b))}{(1+m(a+m0(-a+b+m1(-b+u)))*Ea)} & \text{for supnodes if } E(i) > Tab \\ \dfrac{(Popt * Ei * (1+u))}{(1+m(a+m0(-a+b+m1(-b+u)))*Ea)} & \text{for ultnodes if } E(i) > Tab \\ \dfrac{(c * Popt * Ei * (1+u))}{(1+m(a+m0(-a+b+m1(-b+u)))*Ea)} & \text{for allnodes if } E(i) > Tab \end{cases} \tag{4.1}$$

In above equation the absolute value of Threshold is written as $Tab = z * E0$ and here $z \in (0, 1)$. In our

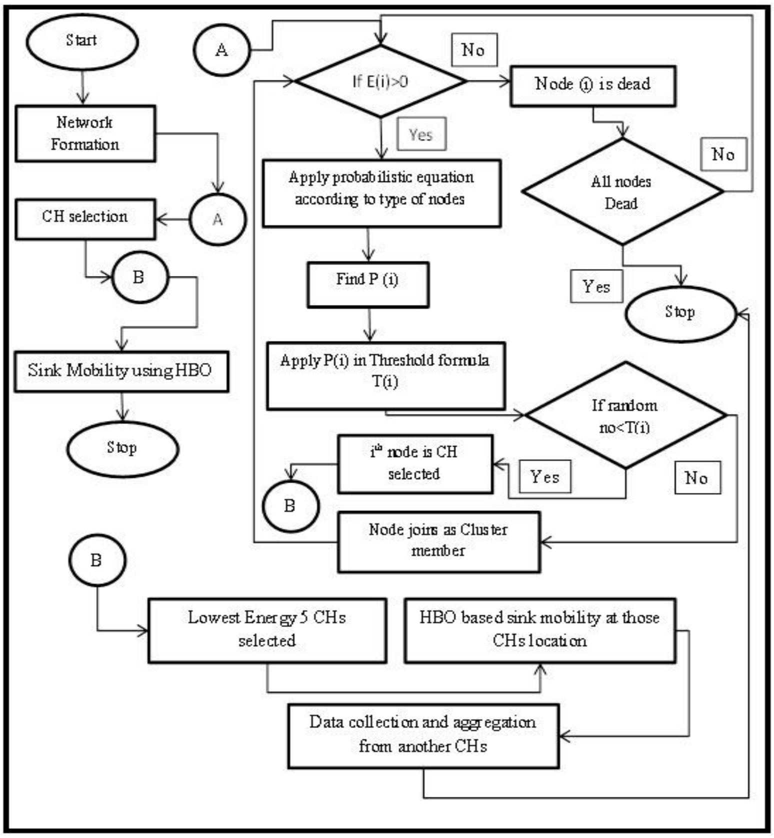

Start
Network Formation
CH selection
A
B
Sink Mobility using HBO
Stop
A
If E(i)>0
No
Node (i) is dead
Yes
Apply probabilistic equation according to type of nodes
Find P (i)
Apply P(i) in Threshold formula T(i)
All nodes Dead
No
Yes
Stop
If random no<T(i)
ith node is CH selected
Yes
No
B
Node joins as Cluster member
B
Lowest Energy 5 CHs selected
HBO based sink mobility at those CHs location
Data collection and aggregation from another CHs

Fig. 4.1. *Methodology*

proposed scheme we have used $z = 0.9$ and Ea being the average energy of the network which is given by the expression below:

$$Ea = \frac{Et(1 - \frac{r}{R})}{n} \tag{4.2}$$

Here r is the current round and R being the maximum number of rounds used in the network. The selection of cluster head is done by the threshold formula which is being used in the BEENISH but with some modification by multiplying it by the terms of residual energy and average energy of network.

$$T(Si) = \begin{cases} \dfrac{((P(i) * E(i))}{((1 - P(i)(rmod(1/P(i))) * Ea)} & \text{if } S(i) \in G \\ 0 & \text{otherwise} \end{cases} \tag{4.3}$$

Nodes which belong to set G are eligible for cluster formation or else other nodes get out of cluster if they previously selected. By utilizing value of $P(i)$ using equation 4.3 different normal, ultra and advance nodes selects under cluster head.

The methodology for the proposed work is stated as follows.

i The process for the data collection starts with the network formation as shown in Figure 4.1. In network formation, the network area is defined with a defined number of nodes. The energy values to those nodes are assigned. These energy values are radio values which are fundamentally the same in all wireless routing protocols.

ii Network formation is connected to the connector step A. It checks for the energy of every node and applies probabilistic equation according to the types of nodes. If the energy of a node is 0, a node is

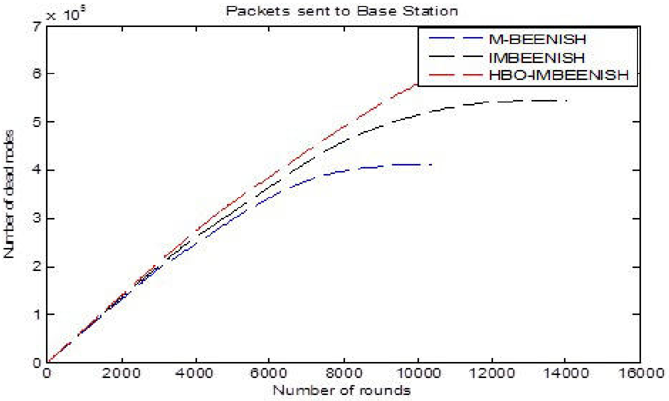


Fig. 5.1. *Packets sent to Base Station*

said to be dead. A counter is checked for the total number of nodes. If all the nodes are dead, then network stops functioning. If the energy of the node is non zero, then it determines the $P(i)$ value for each node. Then it determines the threshold formula while using the $P(i)$ value.

iii A random number is generated for each node and it is compared with the Threshold value, if the random number is less than the threshold value, it is selected as Cluster Head node. Otherwise, the node is selected as a Cluster Member node.

iv Thereafter, the connection to the B terminal is done. The selection of the lowest energy CHs is done. Honey Bee Optimization is applied to select the CH and to move the sink towards the selected CH. Then sink collects the data and aggregates from the CHs thereafter the network operation is halted.

5. Results and Discussions. The simulation is performed in MATLAB software. The simulation analysis is being shown between Figure 5.1 - 5.4. It is observed that the HBO-iMBEENISH has outperformed the protocols MBEENISH and iMBEENISH.

Throughput in case of HBO-IMBEENISH has improved significantly as compared to iMBEENISH and MBEENISH as shown in Figure 5.1. It is due to the optimized sink mobility which reduces the number of packet that is dropped while data transmission. The optimized mobility helps in efficient collection of data from the network. The network longevity is observed in Figure 5.1 and Figure 5.3. The stability period is observed to be enhanced by 33% as compared to iMBEENISH and MBEENISH. It is due to the reduced energy consumption by the proposed approach. The remaining energy of HBO-IMBEENISH covers more number of rounds as compared to the IMBEENISH and MBEENISH as shown in Figure 5.4. It is due to balance in energy consumption in the network by the proposed approach.

6. Conclusion. The growing demand for energy efficient WSN has led to the exponential rise in a communication system by triggering the global academia and industry in the same direction. The mobility in WSN has pushed the horizon of its applications to much further level. It is observed that clustering in WSN has not only brought load balancing in the network but also has distributed the energy consumption evenly in the network. In this paper, applying HBO technique for the sink mobility has not only enhanced stability period but also has improved network lifetime with a much larger amount. The proposed model efficiency gets enhanced to 33% which covers 2928 rounds and enhance accuracy whereas iMBEENISHpropotcol involves only 2033 rounds. Improvement in network lifetime also moved to 15000 rounds in comparison with the 14084 rounds completed by the iMBEENISH protocol. Total improvement of 6% gets covered by the proposed protocol. The enhancement in the proposed protocol is accounted to the optimized sink movement in the network and cluster head selection is incorporated with the provision of avoiding the penalization of high energy nodes. The proposed protocol has outperformed the MBEENISH protocol in terms of different performance metrics which are shown in the Figures in the result section. In future work, it is proposed to investigate the performance of the protocol by varying the energy values of nodes.

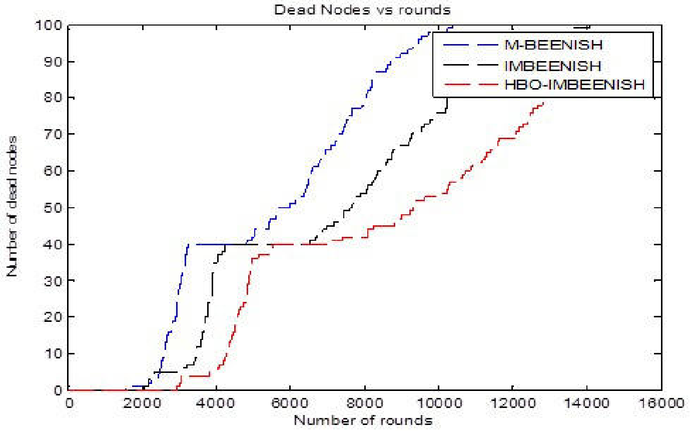


FIG. 5.2. *Dead Nodes vs. Rounds*

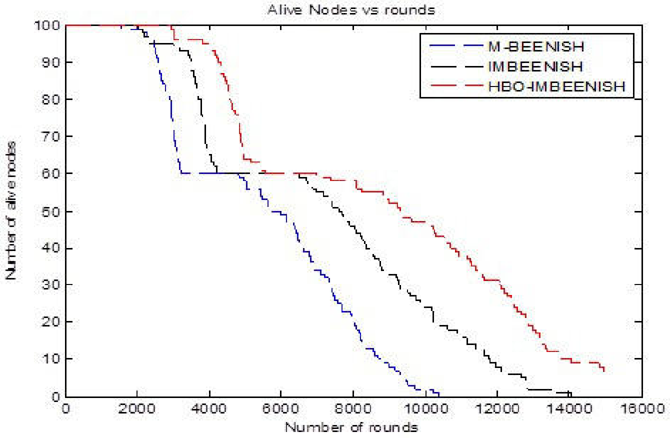


FIG. 5.3. *Alive Nodes vs. Rounds*

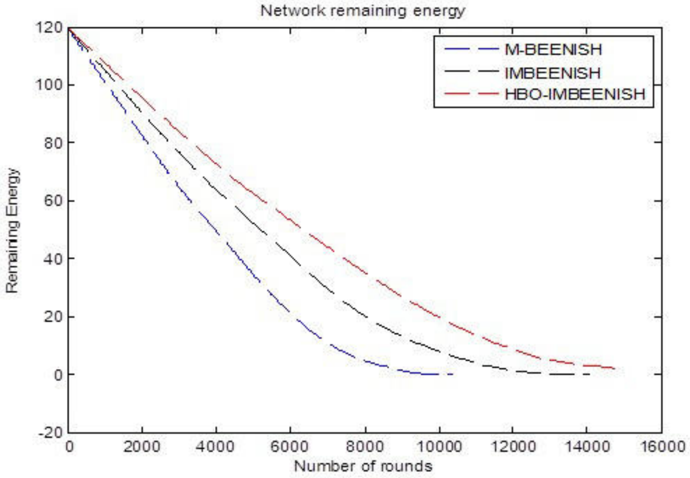


FIG. 5.4. *Network Remaining Energy*

Acknowledgments. The authors thanks the anonymous authors whose work largely constitutes the sample file. They also thanks the INFO-TeX mailing list for the valuable indirect assistance they received.

REFERENCES

[1] M. Akbar, N. Javaid, M. Imran, N. Amjad, M. I. Khan, and M. Guizani, *Sink mobility aware energy-efficient network integrated super heterogeneous protocol for wsns*, EURASIP Journal on Wireless Communications and Networking, 2016 (2016), p. 66.

[2] I. F. Akyildiz, Weilian Su, Y. Sankarasubramaniam, and E. Cayirci, *A survey on sensor networks*, IEEE Communications Magazine, 40 (2002), pp. 102–114.

[3] I. Batra, A. K. Luhach, and N. Pathak, *Research and analysis of lightweight cryptographic solutions for internet of things*, in Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies, ICTCS '16, New York, NY, USA, 2016, ACM, pp. 23:1–23:5.

[4] J. Corchado Rodrguez and A. Abraham, *Using heterogeneous wireless sensor networks in a telemonitoring system for healthcare*, IEEE Trans. Inform. Technol. Biomed., 14 (2010), pp. 234–240.

[5] B. Elbhiri, R. Saadane, S. El fldhi, and D. Aboutajdine, *Developed distributed energy-efficient clustering (ddeec) for heterogeneous wireless sensor networks*, in 2010 5th International Symposium On I/V Communications and Mobile Network, Sep. 2010, pp. 1–4.

[6] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, *Energy-efficient communication protocol for wireless microsensor networks*, in Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, Jan 2000, pp. 10 pp. vol.2–.

[7] M. Johnson, M. Healy, P. van de Ven, M. Hayes, J. Nelson, T. Newe, and E. Lewis, *A comparative review of wireless sensor network mote technologies*, 11 2009, pp. 1439 – 1442.

[8] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, *A comprehensive survey: artificial bee colony (abc) algorithm and applications*, Artificial Intelligence Review, 42 (2014), pp. 21–57.

[9] D. Kumar, T. C. Aseri, and R. Patel, *Eehc: Energy efficient heterogeneous clustered scheme for wireless sensor networks*, Computer Communications, 32 (2009), pp. 662 – 667.

[10] S. Meguerdichian, F. Koushanfar, G. Qu, and M. Potkonjak, *Exposure in wireless ad-hoc sensor networks*, in Proceedings of the 7th Annual International Conference on Mobile Computing and Networking, MobiCom '01, New York, NY, USA, 2001, ACM, pp. 139–150.

[11] A. Pramanik, A. K. Luhach, I. Batra, and U. Singh, *A systematic survey on congestion mechanisms of coap based internet of things*, in Advanced Informatics for Computing Research, D. Singh, B. Raman, A. K. Luhach, and P. Lingras, eds., Singapore, 2017, Springer Singapore, pp. 306–317.

[12] L. Qing, Q. Zhu, and M. Wang, *Design of a distributed energy-efficient clustering algorithm for heterogeneous wireless sensor networks*, Computer Communications, 29 (2006), pp. 2230 – 2237.

[13] T. Qureshi, N. Javaid, A. Khan, A. Iqbal, E. Akhtar, and M. Ishfaq, *Beenish: Balanced energy efficient network integrated super heterogeneous protocol for wireless sensor networks*, Procedia Computer Science, 19 (2013), pp. 920 – 925. The 4th International Conference on Ambient Systems, Networks and Technologies (ANT 2013), the 3rd International Conference on Sustainable Energy Information Technology (SEIT-2013).

[14] K. Rmer and F. Mattern, *The design space of wireless sensor networks*, Wireless Communications, IEEE, 11 (2005), pp. 54 – 61.

[15] P. Saini and A. K. Sharma, *E-deec- enhanced distributed energy efficient clustering scheme for heterogeneous wsn*, in 2010 First International Conference On Parallel, Distributed and Grid Computing (PDGC 2010), Oct 2010, pp. 205–210.

[16] G. Smaragdakis, I. Matta, and A. Bestavros, *Sep: A stable election protocol for clustered heterogeneous wireless sensor networks*, 2004.

[17] S. Sudevalayam and P. Kulkarni, *Energy harvesting sensor nodes: Survey and implications*, IEEE Communications Surveys and Tutorials, 13 (2011), pp. 443–461.

[18] M. A. M. Vieira, C. N. Coelho, D. C. da Silva, and J. M. da Mata, *Survey on wireless sensor network devices*, in EFTA 2003. 2003 IEEE Conference on Emerging Technologies and Factory Automation. Proceedings (Cat. No.03TH8696), vol. 1, Sep. 2003, pp. 537–544 vol.1.

[19] Y. Wang, *Topology Control for Wireless Sensor Networks*, Springer US, Boston, MA, 2008, pp. 113–147.

Edited by: Dharm Singh Jat
Received: Apr 17, 2019
Accepted: Jun 14, 2019

DOI 10.12694/scpe.v20i4.1561
Scalable Computing: Practice and Experience ISSN 1895-1767
Volume 20, Number 4, pp. 599–606. http://www.scpe.org © 2019 SCPE

IOT BASED AIR QUALITY MONITORING SYSTEM USING MQ135 AND MQ7 WITH MACHINE LEARNING ANALYSIS

KINNERA BHARATH KUMAR SAI[*], SOMULA RAMASUBBAREDDY[†] AND ASHISH KR. LUHACH[‡]

Abstract. This paper deals with measuring the Air Quality using MQ135 sensor along with Carbon Monoxide CO using MQ7 sensor. Measuring Air Quality is an important element for bringing awareness to take care of the future generations and for a healthier life. Based on this, Government of India has already taken certain measures to ban Single Stroke and Two Stroke Engine based motorcycles which are emitting high pollution. We are trying to implement a system using IoT platforms like Thingspeak or Cayenne in order to bring awareness to every individual about the harm we are doing to our environment. Already, New Delhi is remarked as the most pollution city in the world recording Air Quality above 300 PPM. We have used easiest platform like Thingspeak and set the dashboard to public such that everyone can come to know the Air Quality at the location where the system is installed. Machine Learning analysis brings us a lot of depth in understanding the information that we obtained from the data. Moreover, we are proviing a reducement of the cost of components versus the state of the art.

Key words: IoT, MQ135, MQ7, Thingspeak, Machine Learning

AMS subject classifications. 68M10, 92B20

1. Introduction.

Air is getting polluted because of release of toxic gases by industries, vehicle emissions and increased concentration of harmful gases and particulate matter in the atmosphere. The level of pollution is increasing rapidly due to factors like industries, urbanization, increasing in population, vehicle use which can affect human health. Particulate matter has the most significant contribution to the increase in air pollution [2]. This creates a need for measurement and analysis of real-time air quality monitoring so that appropriate decisions can be taken in a timely period. This paper presents a real-time standalone air quality monitoring.

Internet of Things (IoT) is nowadays finding profound use in each and every sector and plays a key role in our air quality monitoring system too. Our setup will show the air quality in PPM (Parts per Million) in a web page so that we can monitor it very easily. In this IoT project, one can monitor the pollution level from anywhere using a computer or a mobile [1].

Air Pollution is increasing heavily these days due to the many important factors like Vehicle Emissions, Deforestation, Industrialization [21]. Old vehicles could produce more smoke and hence those vehicles could be banned from using. But we need a proper metric that tells us whats the intensity level of air pollution, whether it is low, medium or high. So, we can take help of various sensors that could detect the air quality and from the corresponding values, we do mathematical calculations and from that results, we could fix the threshold value from which we can say whether the desired air quality is below the threshold value or not.

Every sensor that helps us in this experiment will come with a manufacturer sheet which helps us for the mathematical calculations. Now, Internet of Things helps us to connect these sensors to the Cloud where we are able to access the system from anywhere using login credentials [22]-[27]. There are many online free sources available targeting Internet of things (IoT) that could give a ready dashboard and a means of connecting the account to the physical system. We can use the above said free sources or we could build our own website maintaining a background database and access to the physical system and this could be a tedious job. If one's requirement is something other than available options being provided in the open source websites, then we could develop our own website. Moreover, as the sensors are analog in nature, it would continuously project the values from time to time. We need an enough amount of space that could handle this real time data being generated. Also, making an Android application that could be actively projecting these changing values accurately and fastly is challenging. Open source websites like mydevices.cayyene.com provides both web interface as well as mobile application for both Android and iOS.

Finally, we could apply machine learning for detailed analysis like which are the areas having more air

[*] School of Computer Science and Engineering (SCOPE), VIT University, Vellore, India (yourfriend9014@gmail.com).

[†] Information Technology, VNRVJIET, Hyderabad,India.(svramasubbareddy1219@gmail.com)

[‡] Department of Electrical and Communication Engineering,The PNG University of Technology, Papua New Guinea.(ashishluhach@acm.org)

Table 2.1
Air Quality index

Range(PPM)	Status
0-50	Good
51-100	Moderate
100-150	Unhealthy for sensitive groups
151-200	Unhealthy
201-300	Very Unhealthy
301-500	Hazardous

quality than the desired threshold and also at what time we are getting more values from the sensors. If we could integrate these values at respective places in the maps, more awareness can be targeted such that people will take necessary steps for improvising the air quality like afforestation, growing pot plants inside, upgrading engine quality for reducing bad emissions from vehicles, encouraging electric vehicles etc. Calibrating the sensors before installing at a location matters a lot. Also, almost all the electronic components suffer from aging effect.

2. Related Work. Table 1 [11] explains the Air Quality Index ranges. Firstly, 0-50 PPM can be considered completely safe. 51-100 PPM can be considered as Moderate – this could be usually observed at traffic areas [4]. 100-150 PPM can be considered as Unhealthy, but only for sensitive groups. Above 151 PPM [11] is completely unsafe or unhealthy – India's capital New Delhi falls in this range. Its very rare to record 300 PPM and above, which can be considered as Hazardous, possibly due to Coal gas in mines [10].

The paper [11] implemented the idea of [1] with less cost i.e., while pushing the data to the Cloud, ther is no need to see the output on LCD which adds more cost to the project [14, 18, 28, 29]. Targeting IoT as a platform, our intension should be to present the data on Internet using the platforms like thinger.io or thingspeak or Cayenne website which are beautifully designed to present the output and even able to download the dataset. When doing an experiment for air quality monitoring, there is no need to use LPG (Liquified Petroleum Gas) or methane detecting sensors as it is used for home/office safety. This paper used WiFi to push the data onto the Cloud rather using GSM or GPRS module as in [2]. Note that in [3, 30] hasn't been calibrated the sensor and not converted the sensor output value into PPM. As per the guidelines by UN Data, 0-50 PPM is SAFE value, 51-100 is moderate as shown in Table 1. New Delhi, the capital of India is the most polluted city in the world recorded around 250PPM [15]. As this paper uses two sensors, both of them have internal heater element, it draws more power (P=V*I), so though the both sensors are turned ON, its output voltage levels varies and shows unpredictable values due to insufficient power drive. So we used a 9 Volts battery and a 7805 family LM7805 Regulator for the CO sensor MQ7 as the power from Arduino alone is not sufficient to drive two sensors.

In the paper [4] is not clear with the components used and the cloud used. This paper also aims to implement the machine learning on the real time dataset collected from the Thingspeak website and try to convey the information about the adverse affects on one's health to the people and government if the same pollution continues [31, 32]. This paper also aims to extend by adding three more sensors related to air quality, i.e, ozone sensor, PM 2.5 laser dust sensor, MG811 (CO_2) sensor that acts as an all-in-one setup giving in depth monitoring of the air quality [4]. The paper [5] had completely taken wrong assumption where they have showed the output 997 PPM as the fresh air, while Delhi which is the most polluted city recording 250 PPM. Its clear understanding that they have not calibrated the sensor and did not even convert the raw sensor data into PPM [16] using derivations we did. They have used a Local Host which is limited where they are able to see the output only on the laptop within the area where experimental setup is connected. But this paper targets using standard IoT platform which is highly secured and open source [6].

3. Numerical Evaluation. We have used Arduino Uno Development kit that comes with ATMega 328P microcontroller. In order to provide WiFi Support for it, we have used cost effective ESP-01 WiFi module which helps us to connect to the ThingSpeak Platform. Figure 3.1 represents the connections between the components used like Arduino Uno, MQ135, MQ7, ESP-01 WiFi Module, 9 Volts Battery, LM7805 Regulator. As shown in

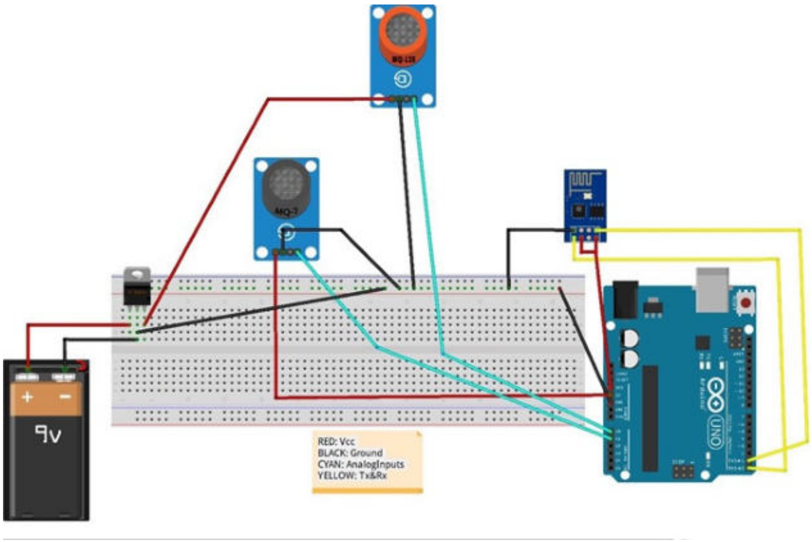


Fig. 3.1. *Connections diagram*

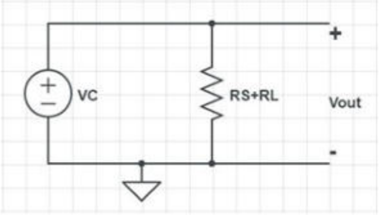


Fig. 3.2. *Internal circuit diagram of MQ135*

Figure 3.1, ESP-01 is connected to 3.3 Volts pin of Arduino Uno. MQ135 is connected to 5 Volts pin of Arduino Uno. As power wont be sufficient to drive one more sensor, MQ7 is connected to 9 Volts Battery via 5 Volts LM7805 Regulator. ESP-01 is connected to the Local Hotspot by giving corresponding SSID and Password. The reason for using LM7805 Regulator is that 9 Volts supply should not be directly given to MQ7 sensor [17] where it needs only 5 Volts input at maximum, so the regulator does the job of stepping 9 Volts to 5 Volts [7, 9].

The most important step is to calibrate the sensor in fresh air and then draws an equation that converts the sensor output voltage value into our convenient units PPM. Here are the mathematical calculations derived from [1].

From Ohm's Law, at constant temperature, we can derive I as follows:

$$I = \frac{V}{R} \tag{3.1}$$

Equation 3.1 is equivalent according to [12] with

$$I = \frac{V_c}{R_s + R_l} \tag{3.2}$$

From Figure 3.2, we can obtain the output voltage at the load resistor using the value obtained for I and Ohm's Law at constant temperature ($V = I \cdot R$):

$$V_{R_l} = \left[\frac{V_c}{R_s + R_l}\right] \cdot R_L \tag{3.3}$$

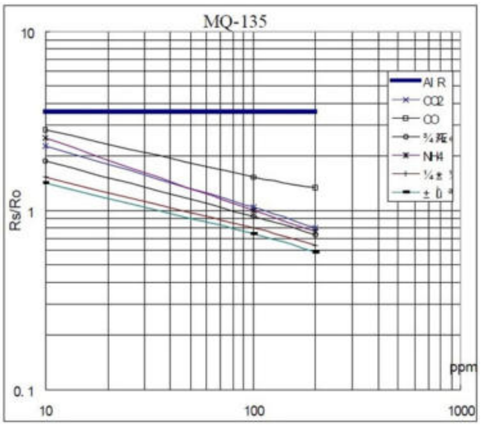


Fig. 3.3. *MQ135 Datasheet-Change in Resistance vs change in PPM*

$$V_{R_l} = \left[\frac{V_c * R_l}{R_s + R_l}\right] \tag{3.4}$$

So now we solve for R_s:

$$V_{R_l} * (R_s + R_l) = V_c * R_L \tag{3.5}$$

$$(V_{R_l} * R_s) + (V_{R_l} * R_l) = V_c * R_L \tag{3.6}$$

$$V_{R_l} * R_s = (V_c * R_l) - (V_{R_l} * R_l) \tag{3.7}$$

$$R_s = \frac{(V_c * R_l) - (V_{R_l} * R_l)}{V_{R_l}} \tag{3.8}$$

$$R_s = \frac{(V_c * R_l)}{V_{R_l}} - R_l \tag{3.9}$$

Equation 3.9 help us to find the internal sensor resistance for fresh air [14].

From the graph shown in Figure 3.3, we can see that the resistance ratio in fresh air is a constant. Figure 3.3 is taken from the MQ135 datasheet where X axis represents the PPM and Y axis represents R_s/R_0 ratio. Now, the sensor is subjected to the respective gases alone and on increasing the PPMs of the gas say methane or Carbon Dioxide, then the corresponding resitance ratio is plotted. If we are interested to calculate the Carbon Monoxide, we have to consider the respective line in the graph and calculate the resistance ratio:

$$\frac{R_s}{R_0} = 3.6 \tag{3.10}$$

Value 3.6 which is mentioned in Equation 3.10 is derived from the datasheet mentioned in Figure 3.3. To calculate R_0, we will need to find the value of the R_s in fresh air. This will be done by taking the analog average readings from the sensor and converting it to voltage [12]. Then we will use the R_s formula to find R_0. First of all, we will treat the lines as if they were linear. This way we can use one formula that linearly relates the ratio and the concentration [19, 20]. By doing so, we can find the concentration of a gas at any ratio value even outside of the graphs boundaries. The formula we will be using is the equation for a line, but for a log-log scale. The formula for a line is [21]:

$$y = mx + b \tag{3.11}$$

From Figure 3.3, we try to derive the following calculations. For a log-log scale, the formula looks like this:

$$\log y = m \log x + b \tag{3.12}$$

Lets find the slope. To do so, we need to choose 2 points from the graph. In our case, we chose the points (200,2.6) and (10000,0.75) from the LPG (Liquified Petroleum Gas) line and we can choose any line shown in Figure 3.3. The formula to calculate m is the following:

$$m = \frac{\log y - \log y_0}{\log x - \log x_0} \tag{3.13}$$

If we apply the logarithmic quotient rule we get the following:

$$m = \frac{\log(y/y_0)}{\log(x/(x_0)} \tag{3.14}$$

Now we substitute the values for x, x_0, y, and y_0:

$$m = \frac{\log(0.75/2.6)}{\log(10000/200)} \tag{3.15}$$

$$m = -0.318 \tag{3.16}$$

Now that we have m, we can calculate the y intercept. To do so, we need to choose one point from the graph (once again from the CO2 line). In our case, we chose (5000,0.9),

$$\log(y) = m \log(x) + b \tag{3.17}$$

$$b = \log(0.9) - (-0.318) * \log(5000) \tag{3.18}$$

$$b = 1.13 \tag{3.19}$$

Now that we have m and b, we can find the gas concentration for any ratio with the following formula:

$$\log(x) = \frac{\log(y) - b}{m} \tag{3.20}$$

However, in order to get the real value of the gas concentration according to the log-log plot we need to find the inverse log of x:

$$x = 10^{\frac{\log(y)-b}{m}} \tag{3.21}$$

Using equations 3.9 and 3.21, we will be able to convert the sensor output values into PPM [12].

We developed the code and flashed it into the Arduino Uno giving proper connections as mentioned in Figure 3.1.

4. Implementation. After connecting the ESP-01 successfully to the hotspot, it gets established with Thingspeak wesbite and the account API Key is written in Arduino Code which helps to save the data only to our account bearing the given API key. Thingspeak needs 15 seconds of refresh interval to push to the data. Figure 4.1 shows the field charts of MQ135 and MQ7 sensor values for the location where the experiment is conducted in PPM [7, 8]. Also Figure 4.1 shows the visualization charts for corresponding sensors.

Figure 4.2 shows the graphical analysis of the values collected with timestamping on X axis and Air Quality PPM on Y axis. It represents Linear Regression applied on the dataset collected from Thingspeak account in CSV format (Comma separated values). Using Python and Anaconda, linear regression is applied for the dataset and the graph as shown in Figure 4.2.

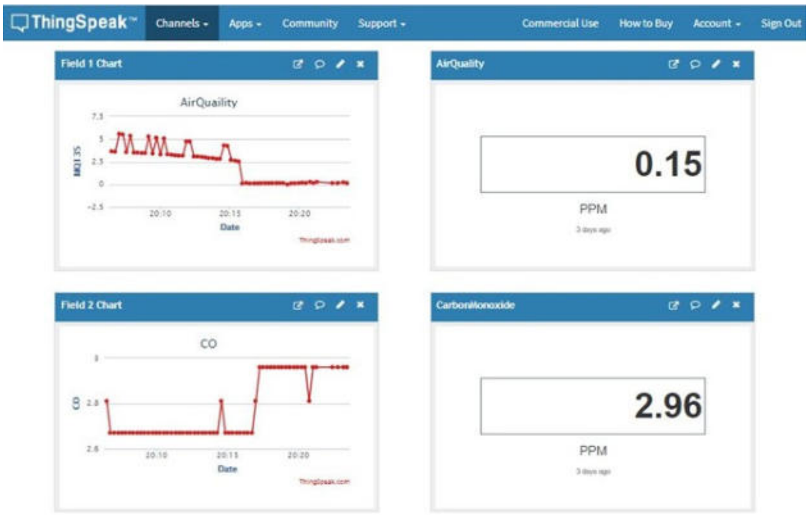


Fig. 4.1. *Output on Thingspeak*

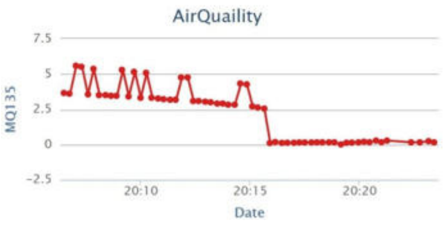


Fig. 4.2. *Graph showing AirQuality*

- Mean Absolute Error: 1.238542891142161
- Mean Squared Error: 4.252972617980345
- Root Mean Squared Error: 2.0622736525447696

In Figure 4.2, X axis is Air Quality in PPM, Y axis is Time stamping in 24 Hour clock format and if the slope of the green color line is very high, AirQuality is very good, if the slope is 45 degrees, Air Quality is moderate, if the slope is very low towards X axis, then the Air Quality is very bad and needs to be concerned.

5. Conclusion. From all the above derivations, figures mentioned, connections diagram, we are able to calculate AirQuality in PPM. The problem with MQ135 sensor is that specifically it cannot tell the Carbon Monoxide or Carbon Dioxide level in the atmosphere, but the pros of MQ135 is that it is able to detect smoke, CO, CO_2, NH_4 as mentioned in Figure 3.3. So, just to tell the individual gases level particularly, we have used CO (Carbon Monoxide) MQ7 sensor. This paper also corrects the PPM calculations mentioned at Literature Survey. This project can be used both for indoor as well as outdoor. For indoor, we can make this kit as a compact device such that if every home started using the device, we can monitor the indoor air quality of a particular targeted area. Due to increasing air pollution, there is necessity to keep an eye on Indoor air quality too. But for outdoor purpose, certainly one sensor is not sufficient because one sensor has a sensitivity range of around 1 meter, so a network of sensors has to be deployed to monitor the outdoor air quality. Enough care is taken while calibrating the sensors. This paper also targets the Machine Learning analysis on the dataset collected.

6. Future Work. We can use one more sensor that tells the ozone layer status, but it costs very high. Also we can use PM2.5 laser dust sensor helpful for exclusively for vehicle and factory emissions sensing. Thingspeak

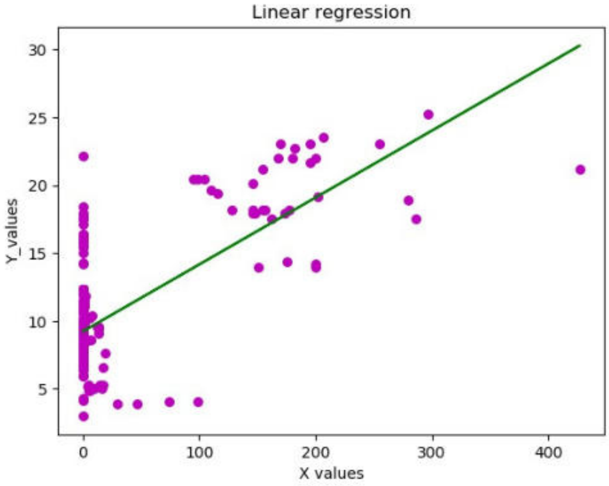


Fig. 4.3. *Linear Regression*

has a limitation that it requires 15-20 seconds for every push of the values which is not reliable. We plan to use another IoT platform mydevices cayenne which is very fast in showing the values from the Arduino that helps us to collect more values in the dataset. Cayenne also comes with a ready Android/iOS application. But it doesnt work with Arduino Uno rather works with only NodeMCU or Raspberry Pi. If we use NodeMCU, even the cost becomes less than the current setup. But the limitation in NodeMCU is, it has only one analog input pin, so we will use ADS1115 I2C 16Bit ADC as an analog extender for NodeMCU or a simple CD4051 8 to 1 Analog Multiplexer could easily overcome the problem of having only one analog input pin of NodeMCU (ESP8266). CD4051 Multiplexer is highly recommended as it is very cheap to purchase and easy to handle many sensors for NodeMCU. NodeMCU (ES8266) has inbuilt Wi-fi support (ESP-12E) and microcontroller. We can link this to the Facebook API using IFTTT, Webhooks and adafruit platform collectively, such that user can request the airquality via facebook messenger chat application and get the output on the screen using chatbot. Machine learning can also be implemented on the dataset such that we can predict the harmfulness of the airquality on people if the same bad airquality continues.

REFERENCES

[1] Tragos, E. Z., Angelakis, V., Fragkiadakis, A., Gundlegard, D., Nechifor, C. S., Oikonomou, G., Gavras, A. (2014). Enabling reliable and secure IoT-based smart city applications. In 2014 IEEE International Conference on Pervasive Computing and Communication Workshops (PERCOM WORKSHOPS) (pp. 111-116). IEEE.

[2] Shah, J., Mishra, B. (2016). IoT enabled environmental monitoring system for smart cities. In 2016 International Conference on Internet of Things and Applications (IOTA) (pp. 383-388). IEEE.

[3] Pasha, S. (2016). ThingSpeak based sensing and monitoring system for IoT with Matlab Analysis. International Journal of New Technology and Research, 2(6).

[4] Khan, R., Khan, S. U., Zaheer, R., Khan, S. (2012). Future internet: the internet of things architecture, possible applications and key challenges. In 2012 10th international conference on frontiers of information technology (pp. 257-260). IEEE.

[5] Ranjan M., Rai Kumar, R. (2009), Understanding Parts per million in real time air quality index, Journal of Mathematics and advanced sciences, pp. 23-29.

[6] Kumar, N. S., Vuayalakshmi, B., Prarthana, R. J., Shankar, A. (2016, November). IOT based smart garbage alert system using Arduino UNO. In 2016 IEEE Region 10 Conference (TENCON) (pp. 1028-1034). IEEE.

[7] Kumar, S., Jasuja, A. (2017, May). Air quality monitoring system based on IoT using Raspberry Pi. In 2017 International Conference on Computing, Communication and Automation (ICCCA) (pp. 1341-1346). IEEE.

[8] Talari, S., Shafie-Khah, M., Siano, P., Loia, V., Tommasetti, A., Catalao, J. (2017). A review of smart cities based on the internet of things concept. Energies, 10(4), 421.

[9] Ma, Y., Yang, S., Huang, Z., Hou, Y., Cui, L., Yang, D. (2014, December). Hierarchical air quality monitoring system design. In 2014 International Symposium on Integrated Circuits (ISIC) (pp. 284-287). IEEE.

[10] Ahlgren, B., Hidell, M., Ngai, E. C. H. (2016). Internet of things for smart cities: Interoperability and open data. IEEE

Internet Computing, 20(6), 52-56.

[11] Cho, R. (2018, June 26). What you should know about air quality alerts. Retrieved from https://phys.org/news/2018-06-air-quality.html

[12] Systems, J. (2016, May 09). Understanding a Gas Sensor. Retrieved from https://jayconsystems.com/blog/understanding-a-gas-sensor

[13] Xiaojun, C., Xianpeng, L., Peng, X. (2015, January). IOT-based air pollution monitoring and forecasting system. In 2015 International Conference on Computer and Computational Sciences (ICCCS) (pp. 257-260). IEEE.

[14] Fang, S., Da Xu, L., Zhu, Y., Ahati, J., Pei, H., Yan, J., Liu, Z. (2014). An integrated system for regional environmental monitoring and management based on internet of things. IEEE Transactions on Industrial Informatics, 10(2), 1596-1605.

[15] Zheng, K., Zhao, S., Yang, Z., Xiong, X., Xiang, W. (2016). Design and implementation of LPWA-based air quality monitoring system. IEEE Access, 4, 3238-3245.

[16] Rushikesh, R., Sivappagari, C. M. R. (2015). Development of IoT based vehicular pollution monitoring system. In 2015 International Conference on Green Computing and Internet of Things (ICGCIoT) (pp. 779-783). IEEE.

[17] Marques, G., Pitarma, R. (2016). An indoor monitoring system for ambient assisted living based on internet of things architecture. International journal of environmental research and public health, 13(11), 1152.

[18] Manna, S., Bhunia, S. S., Mukherjee, N. (2014). Vehicular pollution monitoring using IoT. In International Conference on Recent Advances and Innovations in Engineering (ICRAIE-2014) (pp. 1-5). IEEE.

[19] Kang, B., Park, S., Lee, T., Park, S. (2015). IoT-based monitoring system using tri-level context making model for smart home services. In 2015 IEEE International Conference on Consumer Electronics (ICCE) (pp. 198-199). IEEE.

[20] Ibrahim, M., Elgamri, A., Babiker, S., Mohamed, A. (2015). Internet of things based smart environmental monitoring using the Raspberry-Pi computer. In 2015 Fifth International Conference on Digital Information Processing and Communications (ICDIPC) (pp. 159-164). IEEE.

[21] Xiaojun, C., Xianpeng, L., Peng, X. (2015). IOT-based air pollution monitoring and forecasting system. In 2015 International Conference on Computer and Computational Sciences (ICCCS) (pp. 257-260). IEEE.

[22] Atzori, L., Iera, A., Morabito, G. (2010). The internet of things: A survey. Computer networks, 54(15), 2787-2805.

[23] Somula, R., Sasikala, R. (2018). Round robin with load degree: An algorithm for optimal cloudlet discovery in mobile cloud computing. Scalable Computing: Practice and Experience, 19(1), 39-52.

[24] Somula, R. S., Sasikala, R. (2018). A survey on mobile cloud computing: mobile computing+ cloud computing (MCC= MC+ CC). Scalable Computing: Practice and Experience, 19(4), 309-337.

[25] Somula, R., Sasikala, R. (2019). A load and distance aware cloudlet selection strategy in multi-cloudlet environment. International Journal of Grid and High Performance Computing (IJGHPC), 11(2), 85-102.

[26] Somula, R., Sasikala, R. (2019). A honey bee inspired cloudlet selection for resource allocation. In Smart Intelligent Computing and Applications (pp. 335-343). Springer, Singapore.

[27] Somula, R., Sasikala, R. (2019). A research review on energy consumption of different frameworks in mobile cloud computing. In Innovations in Computer Science and Engineering (pp. 129-142). Springer, Singapore.

[28] Ramasubbareddy, S., Sasikala, R. (2019). RTTSMCE: a response time aware task scheduling in multi-cloudlet environment. International Journal of Computers and Applications, 1-6.

[29] Ramasubbareddy, S., Vedavasu, G., Krishna, G., KB, N., Savithri, A. (2019). PIOCM: Properly Identifying Optimized Cloudlet in Mobile Cloud Computing. Journal of Computational and Theoretical Nanoscience, 16(5-6), 1967-1971.

[30] Basu, S., Kannayaram, G., Ramasubbareddy, S., Venkatasubbaiah, C. (2019). Improved Genetic Algorithm for Monitoring of Virtual Machines in Cloud Environment. In Smart Intelligent Computing and Applications (pp. 319-326). Springer, Singapore.

[31] Ellaji, C. H., Jayasri, P., Ramasubbareddy, S., Kannayaram, G. (2019). Cypher Query Processing for Secure Data Provenance in Cloud. Journal of Computational and Theoretical Nanoscience, 16(5-6), 2517-2522.

[32] Nalluri, S., Ramasubbareddy, S., Kannayaram, G. (2019). Weather Prediction Using Clustering Strategies in Machine Learning. Journal of Computational and Theoretical Nanoscience, 16(5-6), 1977-1981.

Edited by: Dharm Singh Jat
Received: May 11, 2019
Accepted: Aug 20, 2019

DOI 10.12694/scpe.v20i4.1562
Scalable Computing: Practice and Experience ISSN 1895-1767
Volume 20, Number 4, pp. 607–629. http://www.scpe.org © 2019 SCPE

DATA ANALYTICS IN THE INTERNET OF THINGS: A SURVEY

TAUSIFA JAN SALEEM*AND MOHAMMAD AHSAN CHISHTI†

Abstract. The plethora of sensors deployed in Internet of Things (IoT) environments generate unprecedented volumes of data, thereby creating a data deluge. Data collected from these sensors can be used to comprehend, examine and control intricate environments around us, facilitating greater intelligence, smarter decision-making, and better performance. The key challenge here is how to mine out proficient information from such immense data. Copious solutions have been put forth to obtain valuable inferences and insights, however, these solutions are still in their developing stages. Moreover, conventional procedures do not address the surging analytical demands of IoT systems. Motivated to resolve this concern, this work investigates the key enablers for performing desired data analytics in IoT applications. A comprehensive survey on the identified key enablers including their role in IoT data analytics, use-cases in which they have been applied and the performance results of the use-cases is presented. Furthermore, open research challenges and future research opportunities are also discussed. This article can be used as a basis to foster advanced research in the arena of IoT data analytics.

Key words: Internet of things, Big Data, Data Analytics, Data Mining, Machine Learning, Time Series Forecasting.

AMS subject classifications. 68M11, 97R40

1. Introduction.

IoT is regarded as the vital research paradigm in the current epoch. It has dramatically revolutionized every facet of our lives. This technology is characterized by enormous amounts of smart devices that cooperate seamlessly with each other by means of a global network infrastructure, thereby facilitating a wide number of pervasive and ubiquitous applications spanning diverse fields [1]. It is an ecosystem of smart devices, i.e., devices that possess sensing and processing efficacies and can comprehend and respond to their surroundings via sensors and actuators. The blend of diverse technological advancements like near field communication, Radio Frequency Identification (RFID), real-time localization, embedded systems, and the networking expedite the conversion of day to day entities into smart entities [2]. These entities are incorporated impeccably into a web like framework so that they can communicate with one another and with other cyber agents so as to accomplish goal-oriented tasks [3]. IoT enables sensors and objects to interact coherently within smart environs and facilitates information transfer in a suitable manner. The continuum of devices in IoT are connected via several diverse access networks and communication solutions equipped with technologies such as RFID, Wireless Sensor Network (WSN), Bluetooth, Wi-Fi, ZigBee, GSM etc. [4]. Over 30 billion [5] devices ranging from smart phones, to vehicles are prophesied to be linked to the Internet by 2020. The large number of sensors deployed in IoT environments continuously generate unprecedented volumes of structured, unstructured and semi-structured data (Big Data), that cant be handled by conventional processing, storage and analytical systems [6]. Data generated from IoT is different from traditional data in following ways [7]: (i) data is generated continuously at high speed, (ii) Apart from structured data, data may be of semi-structured or unstructured nature as well, (iii) data sources are diverse and fully distributed, and (iv) integration of multi-modal data becomes complex.

The fundamental objective of IoT is to augment the standard of living. Nevertheless, this vision is based on being able to efficiently process, analyze, and comprehend the data generated by IoT devices [8]. Hence, analyzing IoT data in order to divulge trends, concealed patterns, hidden correlations, inferences, and actionable insights is crucial for dispensing elite services to IoT users. In this regard, investigating the technological advancements that can assist in analyzing unstructured and semi-structured data apart from structured data, integrating the data from heterogeneous data sources, performing real-time analytics in delay critical applications as well as in optimizing the process of data analytics becomes indispensable.

1.1. Related Surveys.

To the best of our knowledge, this work is the first of this kind that investigates the key enablers for IoT data analytics and surveys their role in the IoT use-cases. There are few research works in the literature that have surveyed data analytics in IoT applications, however, the focus of those research works is different from the prime concern of this article. The main focus of this article is on the key enablers for

*National Institute of Technology Srinagar, India (tausifasaleem19@gmail.com)
†National Institute of Technology Srinagar, India

IoT data analytics, the use-cases in which they have been applied, their purpose in those use-cases, the datasets that have been used, and the performance results of the use-cases.

In [9], Siow et al. have reviewed the applications of IoT data analytics across different domains. Moreover, the article proposes taxonomy for IoT data analytics in order to guide future research in the field. The focus of the paper is the enabling infrastructure for IoT data analytics that includes data generation, data collection, data aggregation and integration, storage technologies etc. However, the article lacks the detailed survey on the enablers like data mining techniques, machine learning techniques, dimensionality reduction techniques, time series forecasting, etc.

The work in [10] by Ahmed et al. investigated the latest developments in big data analytics for IoT applications. Moreover, the opportunities created from the amalgam of big data analytics and IoT were also identified. Several research challenges in IoT data analytics were also discussed. However, the article lacks the comprehensive survey of the IoT use cases and the analytics techniques used.

1.2. Contribution and Structure of Paper. This work provides a delineation of the present state-of-the-art in the realm of data analytics in IoT. More precisely, it:

- investigates the key enablers for IoT data analytics
- surveys the role of identified key enablers in IoT data analytics
- identifies the challenges that the research community still has to face in this arena.

The rest of the paper is structured as follows. In Section 2 an introduction of IoT data analytics is provided. Section 3 presents the purpose of data analytics in IoT applications. Section 4 discusses the key enablers for IoT data analytics in detail. Furthermore, the role of the key enablers in IoT data analytics is surveyed in this section. Section 5 highlights the challenges faced by IoT data analytics. Moreover, future research opportunities in the arena of IoT data analytics are also provided in this section. Section 6 presents the concluding remarks.

2. IoT Data Analytics. With the brisk advancements in sensing, communicating, analytic and actuating technologies, the vision of intelligent recognition, real-time observation, monitoring, and management is turning into reality [11,12]. The surfeit of sensors deployed in IoT environments generate masses of structured, unstructured and semi-structured data, including health care data, logistic data, astronomical data, environmental data, etc. [13]. Data collected from these sensors can be utilized to comprehend, examine and control intricate environments around us, facilitating greater intelligence, smarter decision making, and better performance. The enormous amounts of heterogeneous and diverse data generated by millions of IoT devices (monitoring certain phenomenon) make traditional information processing solutions obsolete. This is because traditional information processing systems cannot handle such volume of data [14]. IoT data analytics refers to the analysis of every fragment of data generated from IoT devices at right time in order to extract intelligent insights [15]. It is receiving a wide range of attention from researchers and practitioners, as extracting intelligent insights from IoT data is a tricky task and demands a great deal of attention. The question of how to collect, aggregate, and analyze the data generated from IoT environs has become an important impediment that requires urgent solution [16, 17]. Data Mining and Machine learning may help in creating smarter IoT by extracting unseen patterns, hidden correlations, trends, inferences, and actionable insights, facilitating greater intelligence, smarter decision making, enhancing performance, automation, productivity, and accuracy. However, the unprecedented rise in the magnitude and intricacy of data pose novel challenges to these domains [18, 19]. Moreover, it is crucial to formulate appropriate techniques for dealing with noisy, inaccurate, uncertain and real-time data. Furthermore, in several settings, it is indispensable to merge historical data with the current sensor data so as to draw out effective insights [20].

Most of the IoT devices continuously emanate massive volumes of time series data and such data are ephemeral in nature, thereby demanding real-time action. Consequently, apart from Big IoT data analytics, IoT demands one more category of data analytics, i.e., real-time analytics, to support time stringent applications. Examples include self-driving cars, elder posture recognition, surveillance systems etc. Hence, the aforementioned applications demand fast data analytics with minimal delay. In such applications, transferring data to cloud for analysis is not feasible. The finest remedy for such time stringent IoT applications is to bring analytics closer to IoT data source in order to remove needless delays. However, bringing analytics closer to IoT data source puts forth a new set of challenges, including limitation of power, storage and computing resources [21].

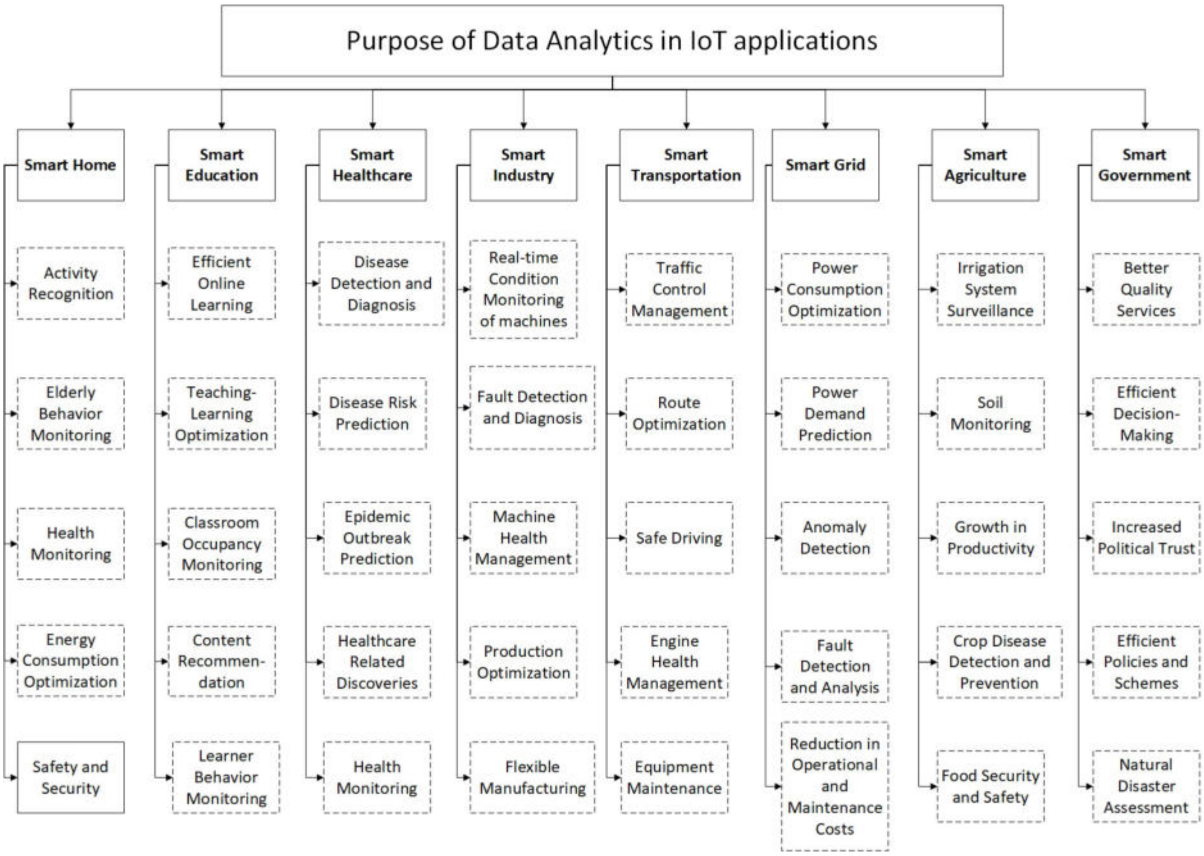

Purpose of Data Analytics in IoT applications
Smart Home
Activity Recognition
Elderly Behavior Monitoring
Health Monitoring
Energy Consumption Optimization
Safety and Security
Smart Education
Efficient Online Learning
Teaching-Learning Optimization
Classroom Occupancy Monitoring
Content Recommen-dation
Learner Behavior Monitoring
Smart Healthcare
Disease Detection and Diagnosis
Disease Risk Prediction
Epidemic Outbreak Prediction
Healthcare Related Discoveries
Health Monitoring
Smart Industry
Real-time Condition Monitoring of machines
Fault Detection and Diagnosis
Machine Health Management
Production Optimization
Flexible Manufacturing
Smart Transportation
Traffic Control Management
Route Optimization
Safe Driving
Engine Health Management
Equipment Maintenance
Smart Grid
Power Consumption Optimization
Power Demand Prediction
Anomaly Detection
Fault Detection and Analysis
Reduction in Operational and Maintenance Costs
Smart Agriculture
Irrigation System Surveillance
Soil Monitoring
Growth in Productivity
Crop Disease Detection and Prevention
Food Security and Safety
Smart Government
Better Quality Services
Efficient Decision-Making
Increased Political Trust
Efficient Policies and Schemes
Natural Disaster Assessment

FIG. 3.1. *Purpose of Data Analytics in IoT applications*

IoT data analytics can be categorized into three groups, descriptive analytics, predictive analytics and prescriptive analytics [10]. Descriptive analytics delineates what has occurred and what is going on. It assists in perceiving novel business challenges and opportunities by utilizing data aggregation and data mining techniques. Descriptive analytics use-cases include energy consumption [22], urban designing [23], etc. Predictive analytics describes what will happen and why. It envisages future conditions and states precisely with the aid of statistical models and prediction techniques. Predictive analytics use-cases include disease prediction [24], predicting energy usage [25], machine failure prediction [26], anomaly prediction [27], etc. Prescriptive analytics characterizes what to do and why it needs to be done. It employs decision support systems to explore diverse possibilities and provides recommendations for decision-making using optimization and simulation algorithms. Prescriptive analytics use-cases include failure risk management in industrial IoT [28], clinical process design and optimization in healthcare [29], etc.

3. Purpose of Data Analytics in IoT Applications. IoT has brought colossal value to our lives by facilitating the growth of a myriad of business-specific and user-oriented applications in different sectors. These applications have triumphed in providing massive benefits to the users. Data analytics has a remarkable part in the development and success of IoT applications. It is used to extract meaningful inferences from IoT data and these inferences are generally in the form of intelligent control decisions, patterns, and statistics that assist IoT applications in powerful decision-making. Hence, utilization of data analytics in IoT applications bring immense benefits including better services, improved productivity, automation, and smarter decision-making. Fig 3.1 presents the purpose of data analytics in IoT applications. Following presents a brief discussion on the IoT applications and the role played by data analytics in these applications.

3.1. Smart Home. A smart home is an important development of IoT in which the dwellings are embodied with intelligence to provide smart services like user comfort, healthcare, security, remote monitoring and control of devices, energy conservation, etc. [30]. Smart homes provide a better standard of living by incorporating automation in the device access, control, and monitoring. The purpose of data analytics in smart home is to render intelligence in order to produce an interactive environment by utilizing foundational services like physiological and psychological state detection, image recognition, voice recognition etc. Analytics of smart home data help in tracking daily activities of the inhabitants, monitoring elderly behavior, optimization of energy consumption, ensuring security, health monitoring, etc.

3.2. Smart Healthcare. Increase in the number of long-term illness cases and regularly aging population is putting a consequential burden on nowadays healthcare organizations. Consequently, there is a dire need to alleviate the stress on healthcare organizations whilst continuing to dispense exorbitant healthcare services to patients. IoT has been recognized as a prospective panacea to reduce the stress on healthcare organizations, thereby transforming healthcare into smart healthcare [31]. The purpose of data analytics in smart healthcare is to provide intelligence in order to ensure remote health monitoring, assist in early disease diagnostics, make novel findings in disease trends, etc. by utilizing foundational services like physiological and psychological state detection, image recognition, voice recognition, etc.

3.3. Smart Industry. with the brisk advancement in communication, computing and manufacturing technologies, production in industrial organizations is being shifted from digital to intelligent [32]. Smart Industry is a smart manufacturing system which integrates production and services together to meet the industrial requirements. The data generated from smart industry typically consist of data pertinent to machine logs and manufacturing processes. Analytics of such data results in services like condition monitoring of machines, fault detection and analysis, machine health management, production optimization, flexible manufacturing, etc. [32,33].

3.4. Smart Transportation. This system aims to exploit sturdy and leading sensing, computational and communication technologies in order to facilitate smart recognition, tracking, and monitoring of vehicles [4]. These technologies will capacitate vehicle-to-vehicle communication in a meticulous way without human arbitration. Moreover, incorporating IoT in transportation systems will provide smart services like traffic congestion management, route optimization, safe driving, etc. Furthermore, real-time information about the availability of parking slots, weather condition, road condition, engine health, equipment maintenance will also be provided [34].

3.5. Smart Grid. Smart Grid is another consequential advancement of IoT for administering and disseminating electricity between suppliers and consumers in order to ameliorate efficiency, safety, reliability with real-time tracking and control [4]. Integrating IoT with electrical systems will facilitate services like optimization of power system performance, fault detection and analysis, security, reduction in operational and maintenance costs [35]. Sensors deployed in smart Grid continuously emanate data pertinent to control loops and security and the data generated demands fast analytics in order to optimize power consumption, predict future power supply needs, detect anomalies, etc.

3.6. Smart Agriculture. Factors like increasing population and dwindling of cultivable land as a result of urbanization, demand extraction of the most out of available resources. Smart agriculture is a novel approach of accomplishing farming tasks by mitigating human endeavor and by efficient utilization of farming resources. Smart agriculture employs advanced sensing, communication, computing and actuating technologies in order to facilitate services like climate control based on harvesting requisites, growth in productivity, automatic irrigation system monitoring, crop disease detection and prevention, soil monitoring, livestock monitoring, etc. [36]. Sensors deployed in Smart agriculture generate data pertinent to moisture content of soil, diameter of the trunk of plants, climatic conditions, humidity conditions, etc. and the generated data demand real-time analytics in order to facilitate aforementioned services.

3.7. Smart Government. Governments can attain a number of benefits from the amalgam of IoT and data analytics. Almost, all the tasks pertaining to government administration demand accurate analysis and prediction. Incorporating IoT and data analytics in the government functionalities will lead to better quality

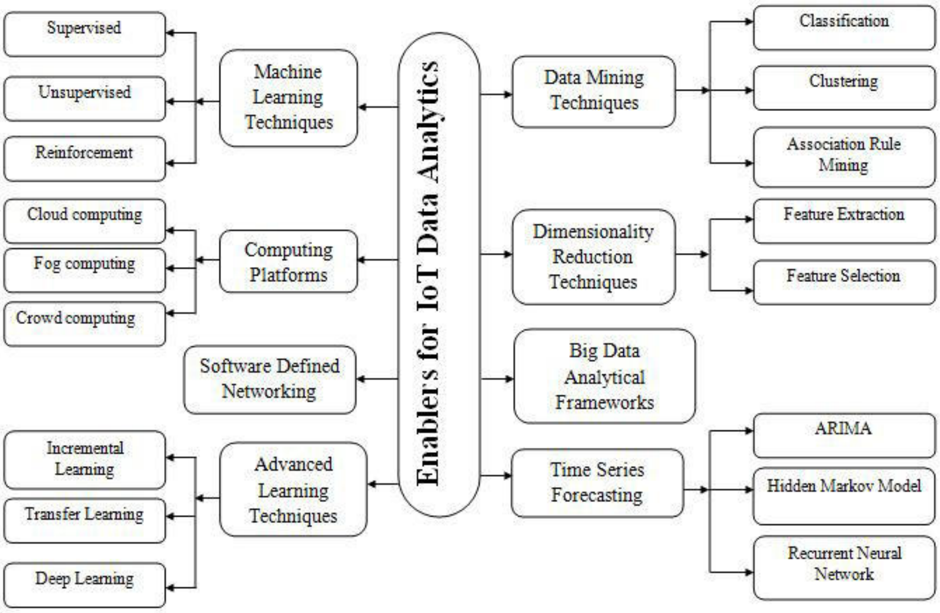


Fig. 4.1. *Key Enablers for IoT data analytics*

services, efficient decision-making, cost optimization, efficient policies and schemes, increase political trust, environmental monitoring, prediction and assessment of natural disasters, assessment of public demands, etc. [37].

3.8. Smart Education. IoT and data analytics contribute to the competence of education systems to a greater extent by enabling services like efficient online learning, teaching-learning optimization, classroom occupancy monitoring, content recommendation, learner behavior monitoring etc [38,39]. Moreover, Integrating IoT with education systems helps in motivating students, identifying weak and struggling students, learners progress assessment, and hence makes learning process efficient.

4. Key Enablers for IoT Data Analytics. From the discussion on IoT data analytics presented in section 2, it is apparent that recognizing and extracting the hidden information from IoT data is a pressing chore that surpasses the potential of conventional information processing and analyzing strategies. However, recent advancements in computational intelligence, data mining, and machine learning approaches are paving way for requisite data analytics in IoT. Fig 4.1 presents the key enablers for IoT data analytics. In the following subsections, we present an elucidation of these techniques in the realm of data analytics.

4.1. Data Mining Techniques. Data mining is utilized to discover concealed patterns and information from the data generated by IoT devices. The main objective of the data mining procedure is to reveal implicit knowledge from the data and mutate it into a valuable shape. Data mining techniques are of three types: classification, clustering and association rule mining.

Classification is a supervised learning procedure that uses a set of labeled data for training purposes to categorize data items into pre-defined classes [4]. The prime goal of utilizing classification in IoT is to predict a class for every instance of input data (unlabelled data). The set of labeled data is utilized for training to build the classification model while as unlabelled data is classified by the classification model. The objective of classification is to develop a classifier that learns the distribution of patterns in the set of labeled data. Classification has been used in numerous IoT use-cases including real-time ECG monitoring [40], twitter sentiment analysis [41], ebola virus outbreak control [42], real-time monitoring of breast cancer patients [43], automatic people counter in stores [44], real-time fall detection system for elderly people [45], defect detection in machines [46], cardiac arrest prediction [47], video surveillance [48], rice disease monitor and control [49], real-time condition monitoring of electric machines [50], etc.

Clustering is an unsupervised learning procedure that groups data items with similar characteristics together into the same cluster [18]. In other words, data items in the same cluster have identical traits and data items in different clusters have highly disparate traits. Examples of IoT use-cases that utilized clustering include activity recognition [51], heart disease survival prediction [52], electricity load prediction [53], behaviour visualization of Sybil attacker [54], Type 2 diabetes monitoring [55], wormhole attack detection [56], weather data analysis [57], safe driving [58], gesture recognition [59], etc.

Association rule mining includes recognition of frequently occurring attribute-value relationships. It assists in the creation of more qualitative information for effective decision-making [18]. Association rule mining focuses on discovering all the frequently occurring associations from a set of data items. It has been used in diverse IoT use-cases including data mining in medical applications [60], human activity recognition [61], extraction of usage patterns of devices [62], etc.

Table 4.1 presents the purpose of data mining in the IoT use-cases mentioned in this sub-section.

4.2. Machine Learning Techniques. Machine learning offers the ability to systems to automatically learn and improve from experience without demanding the obligation of adhering to static program directions. Machines learning approaches craft an effective correlation among input data instances and the output actions and are competent of accomplishing forecasting and decision-making tasks in IoT applications [20]. These approaches are generally divided into three categories: supervised, unsupervised and reinforcement learning.

Supervised learning techniques model dependencies and associations between the target prediction outcome and the input attributes so that outputs for upcoming data instances are forecasted depending on the associations it learned from the dataset [63,64]. Techniques in this category include Linear Regression, Decision tree, Random Forests, Naive Bayes, K-Nearest Neighbour (KNN), Support Vector Machine (SVM) and Artificial Neural Network (ANN).

Regression is a supervised learning algorithm that is used to forecast a real-valued output from the correlations learned from the training data. Linear regression presumes a linear correlation between the input predictors and the target output. Example of IoT use-case that utilized linear regression is energy consumption prediction in digital manufacturing systems [65].

Decision tree follows a greedy strategy to classify data items by arranging them based on attribute values. Example of IoT use case that utilized decision tree is activity and movement recognition [66].

Random Forest is a supervised learning technique in which a myriad of decision trees are trained on different subsets of training set chosen randomly. Example of IoT use-case that utilized random forest is diagnosis and prediction of diseases [67].

Naive Bayes is a supervised learning technique for performing multi-class classification. It uses Bayes theorem for determining the probability of a class given a data item. Example of IoT use-case that utilized naive bayes is device problem detection [68].

KNN is a supervised learning technique in which outputs for new data instances are predicted by exploring K identical data instances in the dataset and taking the mode of their output values as the predicted output for the new data instance. Example of IoT use-case that utilized KNN is appliance recognition in power management systems [69].

SVM, a supervised learning technique is based on the concept of augmenting the margin, i.e., each of the two sides of a hyperplane that splits the linearly separable input variable space into two classes. Example of IoT use-case that implemented SVM is indoor acoustic surveillance [70].

Artificial Neuron is the elementary computational unit in an ANN. It accepts one or more inputs and performs their weighted sum, which is then passed as an input to a non-linear function called as activation function. Example of IoT use-case that implemented ANN is intelligent intrusion detection [71].

Unsupervised learning applies techniques on the input data instances to mine useful information, detect patterns and group the data instances so that valuable insights are obtained [63,72]. These techniques include K-Means Clustering, Apriori and FP Growth.

K-Means clustering is an unsupervised learning technique that is utilized in scenarios with unlabeled data. The objective of this algorithm is to group data items into a K number of clusters. Example of IoT use-case

Table 4.1
Purpose of Data Mining in IoT use-cases.

Work	IoT Use-Case	Data Mining Method	Purpose of Data Mining	Dataset	Performance Results
[40]	Real-time ECG monitoring	Classification	To classify ECG data into different cardiovascular conditions	Data obtained from ECG sensor	-
[41]	Twitter sentiment analysis	Classification	To categorize tweets into two classes, positive and negative	Gold standard dataset from SemEval 2017	Accuracy: 99.2 percent
[42]	Ebola virus outbreak control	Classification	To assess the intensity of infection in a user based on the symptoms	EVD database	Accuracy: 94 percent
[43]	Real-time monitoring of breast cancer patients	Classification	To categorize breast cancer into two classes, benign and malignant	Breast cancer dataset from UCI repository	Accuracy: 95.6 percent
[44]	Automatic people counter in stores	Classification	To categorize people into adults and children based on their height	-	Accuracy: 91 percent
[45]	Real-time fall detection system for elderly people	Classification	To classify images into two types; standing state and falling state	Dataset consisting of fall/non fall events	Accuracy: 95.5 percent
[46]	Defect detection in machines	Classification	To categorize products into defected and non-defected classes	-	-
[47]	Cardiac arrest prediction	Classification	To classify ECG signal patterns into two types; normal and abnormal	Data collected from subjects with different age groups and heights	-
[48]	Video surveillance	Classification	To categorize traffic into five classes: non-critical traffic, little critical traffic, rather critical traffic, critical traffic, very critical traffic	Network traffic	Accuracy: 77 percent
[49]	Rice disease monitor and control	Classification	To classify rice diseases into four categories; rice bacterial blight, rice blast, rice brown spot and rice sheath rot	Images of infected rice leaves	Accuracy: 89.23 percent
[50]	Real-time condition monitoring of electric machines	Classification	To formulate condition monitoring decisions for electric machines based on the vibration patterns of the shaft	Data is gathered from the vibration analysis of the shaft	-
[51]	Activity recognition	Clustering	To categorize the activity patterns of the user into different clusters	Data from Washington State University (WSU) CASAS smart home project	Accuracy: 88 percent
[52]	Heart disease survival prediction	Clustering	To group data items into two clusters based on the attribute value similarity	Heart Disease Dataset	-
[53]	Electricity load prediction	Clustering	To categorize the massive dataset into small clusters	Electric load data from power industry	MAPE: 3.0554
[54]	Behavior visualization of Sybil attacker in IoT	Clustering	To group compromised identities and deploy the sybil node for corresponding identities without violating the set of adjacent nodes	Network Traffic	Coverage: 48.7 percent
[55]	Type-2 diabetes monitoring	Clustering	To categorize data into different clusters	Data of individuals with Type-2 Diabetes	-
[56]	Wormhole attack detection in IoT	Clustering	To divide the nodes into various clusters based on their location from the root node	Data from RPL network in IoT	Accuracy: 93 percent
[57]	Analysis of weather data and sensor fault detection	Clustering	To categorize the regions with different weather data characteristics	Linked Sensor Data and Linked Observation Data	-
[58]	Safe driving	Clustering	To identify accident-prone areas	Data collected using accelerometer, and GPS sensor	-
[59]	Gesture recognition	Clustering	To detect the presence of an event	-	Accuracy: 100 percent
[60]	Data Mining in medical applications	Association Rule Mining	To find similar items in the dataset	Medical Data	Number of scans: 122
[61]	Human activity recognition	Association Rule Mining	To mine frequent patterns	Data collected using wearable sensors	Accuracy: 95.16 percent
[62]	Extraction of usage patterns of IoT devices	Association Rule Mining	To extract device co-usage patterns	Data gathered from 201 residential broadband subscribers of a large European ISP	Confidence: 0.78

Table 4.2
Purpose of Machine Learning in IoT use-cases.

Work	IoT Use-Case	Machine Learning Technique	Purpose of Machine Learning techniques	Dataset	Performance Results
[65]	Predicting energy consumption of digital manufacturing systems	Linear Regression	To Predict the power consumption	Data obtained from SLS manufacturing system (EOS P700)	Accuracy: 96.1 percent
[66]	Activity and movement recognition	Decision Tree	To recognize the activities and movements of the patient	Data obtained from smart phone	Accuracy: 76.83 percent
[67]	Diagnosis and prediction of diseases	Random Forests	To predict the risk of chronic heart disease for the stroke affected patients	Data obtained from patients body	Accuracy: 93 percent
[68]	Device problem detection	Naive Bayes	To predict the problem in a device	-	-
[69]	Appliance recognition in Power Management systems	K Nearest Neighbour	To recognize an appliance	Appliance signature database	Accuracy: 92.73 percent
[70]	Indoor acoustic surveillance	Support Vector Machine	To identify high stress speech signals	Surveillance of Waterloo International Airport	Accuracy: 89.67 percent
[71]	Intelligent intrusion detection	Artificial Neural Network	To identify benign and malicious network traffic	Malicious shellcode data	Accuracy: 98 percent
[73]	Optimization of real-time traffic network assignment	K-Means Cluster	To cluster the similar data points	GIS data	-
[74]	Human sequential Movement Prediction	Apriori	To predict the human movement sequence patterns	Data collected using GPS device	F-measure: 0.687
[75]	Early detection of liver cancer	FP Growth	To discover patterns from liver cancer dataset for early detection	Data obtained from the British Columbia Cancer (BC) Agency	-
[77]	Predictive analytics	Q-learning	To forward a query to a proper query processor	-	-

that implemented K-Means clustering is optimization of real-time traffic network assignment [73].

Apriori is an extensively used algorithm for association rule mining. It is used for recognizing frequently occurring attribute-value relationships in the dataset. Example of IoT use-case that utilized apriori is human sequential movement prediction [74].

Another procedure for association rule mining is FP Growth. Apriori utilizes a breadth-first search approach to determine the set of frequently occurring data items and hence is quite expensive in terms of memory usage. While as FP Growth algorithm utilizes a depth-first search approach. Example of IoT use-case that utilized FP Growth is early diagnosis of liver cancer [75], etc.

Reinforcement learning algorithms learn incessantly from the experience of the environment in an iterative manner until they inspect the full range of feasible states [63, 76] e.g., Q-Learning.

Q-learning is a reinforcement learning algorithm that is based on value instead of policy. It is an easy method for agents to comprehend how to proceed efficiently in controlled environments. It operates by continuously advancing its evaluation measures of the quality of specific actions at specific states. Example of IoT use-case that utilized reinforcement learning is predictive analytics in smart cities [77].

Table 4.2 presents the purpose of machine learning in the IoT use-cases mentioned in this sub-section.

4.3. Advanced Machine Learning Techniques. Apart from traditional machine learning approaches, various advanced learning approaches like deep learning, incremental learning, and transfer learning are also used to dig out valuable knowledge from IoT data. Deep learning is appropriate for modeling complex behaviours of diverse data sets and transfer learning is mostly useful for scenarios with limited data sets while as incremental learning means real-time learning. It is appropriate for the scenarios where data arrive over time in a sequential fashion.

Deep learning is a representation learning approach that utilizes a hierarchical learning process to mine representations from data by making use of several hidden layers with non-linear transformations [78]. It offers an exemplary solution for various classification and recognition tasks as it encapsulates various levels of abstraction. It is appropriate for modeling complex behaviours of diverse datasets. Deep learning consists of diverse architectures including Restricted Boltzmann Machine (RBM), Deep Belief Network (DBN), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Long Short Term Memory (LSTM), Auto-Encoder (AE) etc. RBM and DBN capture high-level representations of input data in an unsupervised manner. CNN works exceptionally well with image data. RNN and LSTM are utilized for time series forecasting. AEs are utilized for dimensionality reduction of high dimensional data. Deep learning models have been utilized in numerous IoT use-cases including transportation analysis [79], localization [80], air quality prediction [81], human activity detection [82], malware detection [83], traffic sign detection [84], crop recognition [85], fault diagnosis [86], plant classification [87], pose detection [88], etc.

Incremental learning means real-time learning. It is appropriate for the scenarios where data arrive over time in a sequential fashion [20, 89]. By means of their sequential treatment, these learning settings offer an elegant inferencing scheme for processing big data. To make upcoming learning and data analytics effective and beneficial, data-rigorous use cases demand that the learning algorithms should have the ability of performing incremental learning so that knowledge base is built over time [90]. Examples of IoT use-cases that utilized incremental learning include fire detection [91], self learning [92], outlier detection [93], etc.

Transfer learning is mostly useful for scenarios with limited datasets. It is a machine learning approach in which the learning parameters of a modeled predictive task are exploited to improve generalization in a different but related problem with limited data [94, 95, 96]. Transfer learning ensures better performance by saving time while modeling a predictive problem. Given the massive resource requirements of deep learning models on large and challenging datasets, transfer learning is admired in deep learning. Transfer learning involves the following steps:

1. Select a related source task: A related predictive modeling problem with ample amount of data is chosen.
2. Develop a model for the chosen source task.
3. The model developed for the source task is then used as a starting point for developing a model on the actual task.
4. Tune model.

Examples of IoT use-cases that harnessed transfer learning include human activity recognition [97], microscopic image classification [98], acceleration of neural network model execution [99], etc.

Table 4.3 presents the purpose of advanced machine learning techniques in the IoT use-cases mentioned in this sub-section.

4.4. Dimensionality Reduction Techniques. Data pre-processing is a vital step for effectual machine learning and data mining. Most machine learning, time series forecasting, and data mining techniques may not be effective for high dimensional data. Dimensionality means the number of attributes in the input data instances of a dataset. When the number of attributes in the input data instances is very huge as opposed to the number of instances in the dataset, certain algorithms struggle to train effective and efficient models. This anomaly is known as the Curse of Dimensionality [100]. To combat this curse of Dimensionality phenomenon, data downsizing techniques have been designed. These techniques are broadly classified into two types: Feature Selection and Feature Extraction.

Feature extraction techniques create a new, smaller set of features that are able to capture most of the useful information [101]. These techniques consist of Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), and AEs.

PCA generates linear combination of the original attributes. The new attributes formed are arranged according to their explained variance. Examples of IoT use-cases that utilized PCA include include soil moisture retrieval [102], face recognition [103], intrusion detection [104], structural health monitoring [105], network

Table 4.3
Purpose of Advanced Machine Learning in IoT use-cases.

Work	IoT Use-Case	Advanced Learning Technique	Purpose of Advanced Machine Learning Techniques	Dataset	Performance Results
[79]	Transportation analysis	Deep Learning/ RBM and RNN	To forecast congestion of traffic	GPS data	Accuracy: 88 percent
[80]	Localization	Deep Learning	To predict indoor positioning based on indoor fingerprinting	CSI values gathered from three antennas	Mean Error: 0.9425
[81]	Air quality prediction	Deep Learning/ LSTM	To predict air quality	Pollution dataset from City Pulse EU FP7 Project	Precision: 98 percent
[82]	Human activity detection	Deep Learning/ CNN and LSTM	To predict activities based on data from multimodal wearable sensors	Opportunity dataset	F1 score: 95.8 percent
[83]	Malware Detection	Deep Learning/ DBN	To detect android malwares in smart phone	Android applications	Accuracy: 96 percent
[84]	Traffic sign detection	Deep Learning/ CNN	To detect traffic signs	-	-
[85]	Crop recognition	Deep Learning/ CNN	To distinguish summer crop types	Data collected from Landsat-8 and Sentinel-1A RS satellites in Ukraine	Accuracy: 85 percent
[86]	Fault diagnosis	Deep Learning/ AE	To learn the useful fault features and carry out fault diagnosis	CWRU bearing data	Accuracy: 94.11 percent
[87]	Plant Classification	Deep Learning/ CNN	To classify images of plants based on their types	Data collected from TARBIL project in Turkey	Accuracy: 97.47 percent
[88]	Pose detection	Deep Learning	To detect human poses	Image Parse dataset	Percentage of correct parts: 69 percent
[91]	Fire detection	Incremental Learning	To become accustomed to the timely changes in the data	Dataset from the Metropolitan Fire Brigade from a state in Australia	-
[92]	Self-learning	Incremental Learning	To enable self learning in IoT environments	-	-
[93]	Outlier detection	Incremental Learning	To detect outliers	-	-
[97]	Human Activity recognition	Transfer Learning	To utilize a pre-trained Autoencoder based activity model for unseen human activity recognition with unlabeled data	Data collected from accelerometer sensor	Accuracy: 98 percent

[98]	Microscopic image classification	Transfer Learning	To utilize features extracted from pre-trained Convolutional Neural Network models	2D-Hela and PAP-smear datasets	Accuracy (2D-Hela):
					92.57 percent
					Accuracy (PAP-smear):
					92.63 percent

[99]	Acceleration of neural network model execution	Transfer Learning	To make the deployment of Deep learning architectures possible on edge devices	-	-

anomaly detection [106], machine health management [107], etc.

LDA also generates linear combinations of original attributes. However, contrary to PCA, LDA doesn't maximize the explained variance. Rather, it augments the separability between classes. Examples of IoT use-cases that implemented LDA include ECG classification [108], event prediction [109], irrigation system surveillance [110], online activity recognition [111], intrusion detection [112], etc.

AEs are neural networks that are trained to regenerate their original inputs. The idea is to structure the hidden layer to have lesser neurons than the input/output layers. With the result, the hidden layer learns to build a smaller representation of the input. Examples of IoT use-cases that utilized AEs include human activity recognition [113], privacy preservation in sensor data analytics [114], prediction performance improvement in sensor and wearable systems [115], botnet traffic detection [116], fault diagnosis [117], etc.

Feature selection techniques filter irrelevant or redundant features from the dataset. These techniques include Genetic Algorithms. Genetic Algorithm accomplishes supervised feature selection. It efficiently selects features from high dimensional data sets where exhaustive search is not feasible. Examples of IoT use-cases that have utilized genetic algorithm include intrusion detection [118], medical image feature extraction and selection [119], pattern recognition [120], building energy optimization [121], gait analysis [122], etc.

Table 4.4 presents the purpose of dimensionality reduction techniques in IoT use-cases mentioned in this sub-section.

4.5. Time Series Forecasting. Most of the data produced by IoT devices are time-indexed [123]. And analyzing such data to extract relevant features, predict future instances, and to explore the relationship between multiple data streams is the main aim of time series modeling [124,125]. Time series data exhibit the property of autocorrelation i.e., the current value in the time series is correlated with the past values. In linear models, the current value depends linearly on the past observations while as in nonlinear models, the current value is a nonlinear function of past values. If the properties of a stochastic process fluctuate with time, it is hard to forecast the future values from its observed time series, this phenomenon is known as non-stationarity. Time series modeling techniques include Auto-Regressive Integrated Moving Average (ARIMA), Hidden Markov Model (HMM) and Recurrent Neural Network (RNN).

ARIMA is an extension of Auto-Regressive Moving Average [126]. Both of these techniques are used to forecast future instances in the series. However, ARMA cannot be applied in scenarios where data exhibit non-stationarity. In order to combat this problem, ARIMA was proposed. Since ARIMA is inherently linear, it is not able to model complex data patterns as opposed to approaches like HMM and RNN. ARIMA has been applied in various IoT use-cases including failure prediction in machines [127], weather forecasting [128], occupancy prediction in smart buildings [129], load prediction [130], building energy consumption forecasting [131], etc.

HMM is eminent for its competence in modeling short-term dependencies between adjoining observations. However, it is not suitable for scenarios with long-term dependencies [132]. Examples of IoT use-cases that have employed HMM include anomaly detection [133], physical activity recognition [134], traffic control management [135], health monitoring [136], prediction of user mobility [137], detection of sitting posture activities [138], etc.

RNN and its variants are highly effective in modeling sequences with complex structures because of the following reasons:

- They can extract patterns in time series data with long time lags.
- They are Robust to noise and can perform prediction in the presence of missing values.
- They are inherently non-linear which makes them suitable for modeling complex data patterns.
- They provide support for multi-variate and multi-step forecasting.

RNN suffers from vanishing/exploding gradient problem [139] due to which its performance gets degraded significantly while modeling long input sequences. To overcome this problem Long Short-Term Memory (LSTM), a variant of RNN was designed, that works exceptionally well for long input sequences. Examples of IoT use-cases that have used RNN include activity recognition based on multi-sensor data [140], network traffic classification [141], real-time deterministic control [142], weather forecasting [143], etc.

Table 4.5 provides the purpose of time series forecasting techniques in the IoT use-cases mentioned in this sub-section.

Table 4.4
Purpose of Dimensionality Reduction in IoT use-cases.

Work	IoT Use-Case	Dimensionality Reduction Technique	Purpose of Dimensionality Reduction Techniques	Dataset	Performance Results
[102]	Soil moisture retrieval	Principal Component Analysis	To reduce the number of dimensions in the feature set	Data collected from UWB radar sensor (P410)	Accuracy: 95.96 percent
[103]	Face recognition	Principal Component Analysis	To diminish the dimension of face feature so as to improve the computational efficiency	Cohn-Kanade face database	Accuracy: 95 percent
[104]	Intrusion detection	Principal Component Analysis	To decrease the number of features so as to decrease the processing time	Mobile network traffic data	F-measure weighted average: 0.834
[105]	Structural health monitoring	Principal Component Analysis	To get rid of environment interferences from the sensor data	Data is gathered from sensors attached to architectural structure	Accuracy: 94 percent
[106]	Network anomaly detection	Principal Component Analysis	To alleviate the dimensionality of the dataset	-	-
[107]	Machine health management	Principal Component Analysis	To convert high-dimensional data to low-dimensional space	-	-
[108]	ECG classification	Linear Discriminant Analysis	To decrease the number of features in the ECG signal	-	-
[109]	Event prediction	Linear Discriminant Analysis	To reduce the number of features in the dataset for improving the event prediction performance of SVM	Data obtained from IoT devices	Precision: 87.17 percent
[110]	Irrigation system surveillance	Linear Discriminant Analysis	To retrieve the colour of plants and soil images	Data collected from sensors deployed in agricultural fields	-
[111]	Online Activity recognition	Linear Discriminant Analysis	Feature Extraction	WSU Cairo ADL dataset	Accuracy: 98.36 percent
[112]	Intrusion detection	Linear Discriminant Analysis	Classification for intrusion detection	Network traffic data	Accuracy: 99.44 percent
[113]	Human activity recognition	Autoencoders	Feature Extraction	-	-

[114]	Privacy preservation in sensor data analytics	Autoencoders	To convert sensitive discriminative features of data into non-sensitive features in order to guard privacy of the users	Opportunity, Skoda, and Hand-Gesture datasets	F1 Score (Opportunity dataset):
					97.36 percent
					F1 Score (Skoda dataset):
					94.94 percent
					F1 Score (Hand-Gesture dataset):
					75.43 percent

[115]	Prediction performance improvement in mobile and wearable systems	Autoencoders	To enquire about unfamiliar features in an efficient manner	HAPT dataset	Accuracy: 91 percent
[116]	Botnet traffic detection	Autoencoders	To extract new set of features in order to distinguish malicious and benign network traffic	Network traffics from ISCX	True Positive Rate: 91 percent
[117]	Fault Diagnosis	Autoencoders	Gearbox fault diagnosis	-	-
[118]	Intrusion detection	Genetic Algorithm	To carry out the attribute reduction of the feature sets	KDD-CUP99 dataset	Accuracy: 96.8 percent
[119]	Medical image feature extraction and selection	Genetic Algorithm	To choose the reduced set of features	-	-
[120]	Pattern recognition	Genetic Algorithm	Feature selection	Leaf shape image dataset	-
[121]	Building energy optimization	Genetic Algorithm	To curtail the expenditure of the energy consumption	Data obtained from a building in Cardiff, UK	Energy saving: 25 percent
[122]	Gait analysis	Genetic Algorithm	To select meaningful features	Data Collected using the 8 camera ELITE stereophotogrammetric system	Accuracy: 97 percent

Table 4.5
Purpose of Time Series Forecasting in IoT use-cases.

Work	IoT Use-Case	Time Series Forecasting Technique	Purpose of Time Series Forecasting Techniques	Dataset	Performance Results
[127]	Failure Prediction in machines	Auto Regressive Integrated Moving Average	To predict failure in machines	Data collected from the sensors attached to a slitting machine	Accuracy: 98.69 percent
[128]	Weather forecasting	Auto Regressive Integrated Moving Average	Time series based weather forecasting	Data collected from sensors that measure following parameters: Temperature, Humidity, Station Barometric pressure, Wind speed, and Wind direction	Root Mean Squared Error: 0.003867201
[129]	Occupancy prediction in smart buildings	Auto Regressive Integrated Moving Average	To predict the number of residents in a smart building at a given location and time	Wifi dataset collected from University of Houston main campus	-
[130]	Load prediction	Auto Regressive Integrated Moving Average	To predict load behaviour in IoT	Data collected from the IoT devices	Average Response Time: 49.79 milliseconds
[131]	Building energy consumption forecasting	Auto Regressive Integrated Moving Average	To predict building energy consumption	Data collected from sensors	MAPE: 1.05-2.59
[133]	Anomaly detection	Hidden Markov Model	To detect the device anomaly	Data collected from the IoT devices	Accuracy: 98 percent
[134]	Physical activity recognition	Hidden Markov Model	To recognize physical activities	Data collected from 10 subjects	Precision: 82.51 percent
[135]	Traffic control management	Hidden Markov Model	To learn the profile information of traffic in less time	Data collected from different traffic profiles	Accuracy: 95 percent
[136]	Health monitoring	Hidden Markov Model	Real time monitoring of cardio vascular patients	Data collected from patients body	-
[137]	Prediction of user mobility	Hidden Markov Model	To estimate the next location	27 day traffic data of mobile network	Prediction time: 1.39 seconds
[138]	Detection of sitting posture activities	Hidden Markov Model	To identify sitting posture activities	Kinect and Smartwatch based 42 dimensional data	Accuracy: 64.88 percent
[140]	Activity recognition based on multi-sensor data	Recurrent Neural Network	To predict future activities of a resident	MIT dataset for activity	Accuracy: 90.85 percent
[141]	Classification of network traffic	Recurrent Neural Network	To classify the traffic flowing in a network	Dataset from RedIRIS	Accuracy: 99.59 percent
[142]	Real time deterministic control	Recurrent Neural Network	Knowledge discovery	Nottingham and CMU datasets	Accuracy for Nottingham: 93.9 percent Accuracy for CMU: 82.3 percent
[143]	Weather forecasting	Recurrent Neural Network	To predict weather	Dataset obtained from Valley weather station in Anglesey (North Wales, UK)	Mean Absolute Error: 0.0476

4.6. Computing Platforms. Cloud computing [144,145,146] and Fog computing [20] are two important models for managing the enormous volume of data produced from IoT environs. With the brisk growth of the IoT, the traditional cloud computing is facing stern issues, like undesirable network latency, and spectral inefficiency which does not make it suitable for scenarios requiring minimal latency, real-time treatment, and mobility support. Determined to resolve these issues, new paradigm transfers the functioning of cloud computing

Table 4.6
Distinction between Cloud Computing and Fog Computing

Features	Cloud computing	Fog computing	References
Size	Server extremely large in size	Servers small in size	[151]
Computational Capacity	Huge	Limited	[18]
Applications	Appropriate for delay-tolerant and computationally exhaustive implementations	Appropriate for delay-critical applications requiring minimal latency, real-time treatment	[18,151]
Communication Overhead	High, as devices are connected to the internet during the whole period	Low, because devices can acquire cached contents straight from edge gateway	[18]
Deployment	Demands composite installation planning	Require ad-hoc installation with no or slight drafting	[18,151]
Operation	Operate in environments fully guarded by cloud operators	Usually, operate in scenarios that are mainly determined by requirements of customers	[151]
Location	Centralized	Distributed over the large geographical area	[151]

closer to the data source. This technology is referred to as Fog computing [147]. With the result network congestion is reduced and decision-making becomes fast. However, these fog devices generally do not have adequate storage and computational resources. Table 4.6 provides the comparison of these computing platforms.

Examples of IoT use-cases that utilized cloud computing as the computing platform include industrial IoT big learning [148], hybrid systems for smart agriculture [149], disease diagnosis [150], disease prevention in precision agriculture [151], temperature control systems [152] etc.

Examples of IoT use-cases that harnessed fog computing as the computing platform include preventive healthcare and assisted living in smart ambient [153], video surveillance [154], asset provisioning for crowd sensing applications in IoT [155], crime assistance [156], data analytics in smart cities using big data [157], etc. [155] proposed a fog based computing scheme known as Mist computing that provides cost-effective resource provisioning for IoT crowd sensing applications.

Apart from cloud computing and fog computing, another emerging computing paradigm known as crowd computing can be utilized for managing the data produced by IoT systems. In crowd computing, IoT devices deployed close to each other and with related interests can share computing and power resources so as to optimize the performance of the IoT systems [158]. Crowd computing has got a massive potential in IoT applications.

4.7. Big Data Analytical Frameworks. This section explores the big data analytical frameworks that can be utilized for analyzing humongous volumes of data produced from IoT environments. Applying the right data analytical framework is fundamental for the successful development of an IoT application. Depending on the analytical requirements of IoT application, data analytics can be performed either in Cloud or near the IoT data source (Fog nodes). Table 4.7 summarizes big data analytical frameworks.

4.8. Software Defined Networking. Software Defined Networking (SDN) is a novel technology that simplifies network administration by segregating the control plane from the data plane, thereby centralizing the network intelligence [174]. It enables virtualization within the network and enhances networking capability. SDN fulfils following fundamental requisites of IoT applications:

4.8.1. Network Management. Network management is a vital consideration in IoT for supervising the tremendous number of devices and the massive volume of data produced by them [175]. SDN enables programmatically efficient control mechanism and hides the complexities of network management from end users. It optimizes network management functionalities such as efficient utilization of bandwidth, minimization of latency and load balancing.

Table 4.7
Big Data Analytical Frameworks

Framework	Description	Benefits	Limitations	Used in IoT application
Hadoop [159]	Apache Hadoop is a software platform that carries out batch processing of massive datasets in a distributed manner using clusters of computing devices	Less susceptible to failure Scalable Provides reliable storage	Slow Lacks security functionalities Lacks support for stream processing	Smart City development and Urban planning [164] Defining human behaviours in social IoT [165], etc.
Spark [160]	Apache Spark is a platform for conducting analytics of huge datasets using distributed computing. It endorses in-memory data processing so as to augment the proficiency of data analytical applications	Supports stream processing Relatively faster because of In-memory computation Fault-tolerant	Expensive Requires manual optimization	Smart Building system [166] Cloud based data analytics for smart cities [167], etc.
Storm [161]	Apache Storm is a platform that carries out data processing in real-time in a distributed manner and generates the result promptly with minimal delay	Scalable Fault-tolerant Less latency Supports stream processing	No flow control	Real time monitoring system in Automotive Manufacturing [168] Intelligent data processing on edge devices [169], etc.
Flink [162]	Apache Flink offers immense potential to perform real-time data processing in a fault tolerant manner at a rate of millions of events per second	Faster Better memory management High throughput Requires less configuration	Not common	Analytics in Industrial Environments [170] Real-time analysis of social networks [171], etc.
Azure Stream Analytics [163]	An event processing engine that analyzes massive volumes of streaming data in order to extract inferences, recognize patterns etc.	Real-time processing capabilities Scalable Data aggregation capabilities	Lacks Job management	Multimedia analytics [172] Monitoring and performance analysis of power plants in real-time [173], etc.

4.8.2. Efficient resource utilization. Efficient resource utilization is fundamental for improving the performance of the network [175]. SDN relieves the simpler edge devices from accomplishing the multifaceted networking tasks and utilizes the available resources efficiently [174].

4.8.3. Energy Management. Massive number of data centers are deployed to process the humongous magnitude of data sensed by IoT devices. Consequently, large quantity of energy is utilized to power these data centers. SDN plays an important role in optimizing the energy usage as it maps the traffic efficiently to the suitable servers and switches off the other unnecessary devices in the data center [175].

4.8.4. Security and Privacy. The utilization of flow-rule-based traffic forwarding concept in SDN facilitates secure control of flows between the various devices in the network, which in turn improves the security and privacy of the data generated by IoT devices [175].

SDN has been applied in numerous IoT use-cases including efficient traffic management for emergency situations [176], intrusion detection [177], service delivery [178], traffic congestion avoidance [179], dynamic distribution of IoT analytics and effective utilization of network resources [180], low latency anomaly detection in smart city [181], etc.

5. Vision and Open Challenges.

Data analytics has brought considerable benefits to IoT applications. However, in order to leverage the full potential of data analytics in IoT applications, following major challenges need to be addressed:

5.1. Data Pre-processing. In data pre-processing, noisy data are smoothened, ambiguities in the data are removed, and missing values are filled, thus making it suitable for further processing. Because of the constrained nature of IoT sensors and intermittent loss of connectivity, the massive scale of IoT data contains more irregularities and uncertainties, thereby complicating data pre-processing. Moreover, IoT data may contain missing and incomplete values that lead to poor data quality. Ensuring completeness in IoT data is vital for its data quality. Efficient pre-processing techniques that can remove irregularities and uncertainties in the data and make it suitable for further processing should be researched.

5.2. Data compression and redundancy reduction. Not all the data generated from IoT environments are useful. Also, there exists a high level of redundancy in IoT data. The closely deployed sensor nodes in IoT tend to capture similar information that leads to redundancy in IoT data. Redundant data not only lead to energy wastage but also dissipate the storage space. Moreover, it also affects the feature extraction process. Hence, removing redundancy in IoT data and ensuring its uniqueness is crucial for its data quality. Effective data compression and redundancy reduction techniques need to be employed so as to alleviate the burden of storage and analytics in such systems.

5.3. Data Integration. How to integrate and analyze heterogeneous data arriving from distributed and diverse sources so that a unified view of these diverse data formats is created is an impediment to IoT data analytics. Deep learning is quite effective in analyzing heterogeneous data. However, the severe resource requirements of deep learning algorithm limit its use in IoT applications. Hence, investigating the ways that will reduce the computational requirements of deep learning models becomes crucial. However, this should be done while preserving the accuracy of deep learning models.

5.4. Visualization. Data visualization aims to make data more meaningful for further analysis and interpretation. However, inappropriate data visualization will diminish the significance of the original data and may even thwart efficient data analysis. Orchestrating visualization in IoT data is complex because of its massive scale. Moreover, visualization in case of highly heterogeneous and diverse IoT environments is a challenging task. Given the importance of appropriate data visualization, devising visualization techniques that are well-suited for the representation of highly complex IoT data becomes crucial.

5.5. Expandability and scalability. The sharply growing IoT data bring in the challenges of expandability and scalability for the IoT analytical systems. Analytical paradigms that are proficient enough to deal with progressively growing complex datasets are highly required. Running analytical techniques on distributed systems with parallel processing is the potential solution for this problem.

5.6. Energy management. With the exponential growth of IoT data, transmission, storage, processing and analysis of such enormous data will certainly dissipate more energy. Energy consumption control and management solutions should be designed for such systems.

5.7. Security and Privacy. Data generated from IoT environs are susceptible to external intervention. Hence, Authentication, authorization, and encryption techniques should be utilized to ensure security of IoT systems. However, conventional data protection solutions are not applicable to the IoT data because of its massive scale and highly diverse nature. To this purpose, design of novel Security and Privacy solutions for such systems becomes inevitable.

6. Conclusion. Ample volumes of data have been generated since the previous decade with the escalation in the number of smart devices. Analyzing this voluminous magnitude of data so as to explore novel knowledge, forecast potential insights and to formulate management decisions is a vital procedure that makes IoT a laudable technology for enhancing the standard of our lives. The prime requisite for most of the IoT applications is an intelligent analytical mechanism that can carry out tasks like classification, clustering, association rule mining, or time series analysis. However, conventional analytical procedures do not tackle the surging analytical needs of IoT systems. To this purpose, this paper identifies the key enablers for IoT data analytics and surveys their role in data analytics. Furthermore, several challenges faced by IoT data analytics were identified so as to stimulate research directions in this arena. At last, it is worthwhile to state that data analytics has brought immense benefits to IoT applications, however, a number of challenges still remain unaddressed, the list of which has been discussed. To the best of our knowledge, this work is the first of this kind and we hope that this survey will be beneficial for the researchers in the field of Data Analytics to understand the key enablers and lead them to the direction of possible future research in this field.

REFERENCES

[1] M. Ge, H. Bangui, and B. Buhnova, *Big data for internet of things: A survey*, Future Generation Computer Systems, Elsevier, 2018.

[2] Gerd Kortuem, Daniel Fitton, and Vasughi Sundramoorth, *Smart Objects as Building Blocks for the Internet of Things*, IEEE Internet Computing, January/February, pp.44-51, 2010.

[3] Yunchuan Sun, Houbing Song, Antonio J. Jara, and Rongfang Bie , *Internet of Things and Big Data Analytics for Smart and Connected Communities*, IEEE Access, Vol. 14, No. 8, 2015.

[4] Mohsen Marjani, Fariza Nasaruddin, Abdullah Gani, Ahmad Karim, Ibrahim Abaker Targio Hashem, and Aisha Siddiqa,Ibrar Yaqoob, *Big IoT Data Analytics: Architecture, Opportunities, and Open Research Challenges*, IEEE Access, Vol. 5, pp. 5247-5261, 2016.

[5] Amy Nordrum, *Popular Internet of Things Forecast of 50 Billion Devices by 2020 Is Outdated*, https://spectrum.ieee.org/tech-talk/telecom/internet/popular-internet-of-things-forecast-of-50-billion-devices-by-2020-is-outdated, 2016.

[6] S. Yin, and O. Kaynak, *Big data for modern industry: Challenges and trends*, Proceedings of IEEE, Vol. 103, No. 2, pp. 143-146, 2015.

[7] Han Hu, Yonggang Wen, Tat-Seng Chua, and Xuelong Li, *Toward scalable systems for big data analytics: A technology tutorial*, IEEE Access, Vol. 2, pp. 652687, 2014.

[8] Simmhan Y., and Perera S., *Big Data Analytics Platforms for Real-Time Applications in IoT*, Big Data Analytics, Springer, New Delhi, pp. 115-135, 2016.

[9] Eugene Siow, Thanassis Tiropanis, and Wendy Hall., *Analytics for the Internet of Things: A survey*, ACM Computing Surveys (CSUR) Vol. 1, No. 1, 2018.

[10] Ejaz Ahmed, Ibrar Yaqoob, Ibrahim Abaker Targio Hashem, Imran Khan, Abdelmuttlib Ibrahim Abdalla Ahmed, Muhammad Imran, and Athanasios V. Vasilakos, *The role of big data analytics in Internet of Things*, Computer Networks, Elsevier, Vol. 129, pp. 459-471, 2017.

[11] Meng Ma, Ping Wang, and Chao-Hsien Chu, *Data Management for Internet of Things: Challenges, Approaches and Opportunities*, International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, IEEE, pp. 1144-1151, 2013.

[12] Tingli Li, Yang Liu, Ye Tian, Shuo Shen, and Wei Mao, *A Storage Solution for Massive IoT Data Based on NoSQL*, International Conference on Green Computing and Communications, Conference on Internet of Things, and Conference on Cyber, Physical and Social Computing, IEEE, Besancon, France , pp. 50-57, 2012.

[13] M. Chen, S. Mao, and Y. Liu, *Big Data: A Survey*, Mobile Networks and applications, Springer, Vol. 19, pp. 171-209, 2014.

[14] Mervat Abu-Elkheir, Mohammad Hayajneh, and Najah Abu Ali, *Data Management for the Internet of Things: Design Primitives and Solution*, Sensors, Vol. 13, pp. 15582-15612, 2013.

[15] Shikhar Verma, Yuichi Kawamoto, Zubair Md. Fadlullah, Hiroki Nishiyama, and Nei Kato, *A Survey on Network Methodologies for Real-Time Analytics of Massive IoT Data and Open Research Issues*, IEEE Communications Surveys and Tutorials, Vol. 19, No. 3, pp. 1457-1477, 2016.

[16] Sai Xie, and Zhe Chen, *Anomaly Detection and Redundancy Elimination of Big Sensor Data in Internet of Things*, https://arxiv.org/abs/1703.03225, 2017.

[17] Hongming Cai, Boyi Xu, Lihong Jiang, and Athanasios V. Vasilakos, *IoT-based Big Data Storage Systems in Cloud Computing: Perspectives and Challenges*, IEEE Internet of Things Journal, Vol. 4, No. 1, pp. 75-87, 2016.

[18] Chun-Wei Tsai, Chin-Feng Lai, Ming-Chao Chiang, and Laurence T. Yang, *Data Mining for Internet of Things: A Survey*, IEEE Communications Surveys and Tutorials, Vol. 16, No. 1, pp. 77-97, 2014.

[19] Furqan Alam, Rashid Mehmood, Iyad Katib, and Aiiad Albeshri, *Analysis of Eight Data Mining Algorithms for Smarter Internet of Things (IoT)*, International Workshop on Data Mining in IoT Systems, Procedia Computer Science, Elsevier, 2016.

[20] Shree Krishna Sharma, and Xianbin Wang, *Live Data Analytics With Collaborative Edge and Cloud Processing in Wireless IoT Networks*, IEEE Access, Vol. 5, pp. 4621-4635, 2017.

[21] Mehdi Mohammadi, Ala Al-Fuqaha, Sameh Sorour, and Mohsen Guizani, *Deep Learning for IoT Big Data and Streaming Analytics: A Survey*, IEEE Communications Surveys and Tutorials, 2018.

[22] Albert Malama, Lilias Makashini, Henry Abanda, Austine Ngombe, and Priscilla Mudenda, *A Comparative Analysis of Energy Usage and Energy Efficiency Behavior in Low- and High-Income Households: The Case of Kitwe, Zambia*, Resources, Vol. 4, pp. 871-902, 2015.

[23] Lszl Lengyel, Pter Ekler, Tams Ujj, Tams Balogh, and Hassan Charaf, *SensorHUB: An IoT Driver Framework for Supporting Sensor Networks and Data Analysis*, International Journal of Distributed Sensor Networks, 2015.

[24] Min Chen, Yixue Hao, Kai Hwang, Lu Wang, and Lin Wang, *Disease Prediction by Machine Learning Over Big Data From Healthcare Communities*, IEEE Access, Vol. 5, pp. 8869-8879, 2017.

[25] Antorweep Chakravorty, Chunming Rong, Pal Evensen, and Tomasz Wiktor Wlodarczyk, *A Distributed Gaussian-Means Clustering Algorithm for Forecasting Domestic Energy Usage*, proceedings of SMARTCOMP, IEEE, China, 2014.

[26] J.C. O'Brien, J.R. Leech, C.C. Wright, C.R. Reeves, N.C. Steele, and C.Y. Choi, *Neural Networks for early prediction of machine failure*, Colloquium on Advanced Vibration Measurements, Techniques and Instrumentation for the early prediction of failure, IEEE, UK, 1992.

[27] Aaron Waibel, Abdullah Ali Alshehri, Soundararajan Ezekiel, *Anomaly Prediction in Seismic Signals Using Neural Networks*, Proceedings of Applied Imagery Pattern Recognition Workshop, IEEE, USA, 2013.

[28] Davorin Kuchan, *Prescriptive Analytics for Industrial IoT Failure Risk Management*, https://www.sparklinglogic.com/prescriptive-analytics-industrial-iot/, 2016.

[29] River Logic, *10 Use Cases for Prescriptive Analytics in Healthcare*, https://blog.riverlogic.com/infographic-10-use-cases-prescriptive-analytics-in-healthcare, 2016.

[30] Muhammad Raisul Alam, Mamun Bin Ibne Reaz, and Mohd Alauddin Mohd Ali, *A Review of Smart HomesPast, Present, and Future*, IEEE Transactions On Systems, Man, And CyberneticsPart C: Applications And Reviews, Vol. 42, No. 6, pp. 1190-1203, 2012.

[31] Stephanie B. Baker, Wei Xiang, and Ian Atkinson, *Internet of Things for Smart Healthcare: Technologies, Challenges, and Opportunities*, IEEE Access, Vol. 5, pp. 26521-26544, 2017.

[32] Baotong Chen, Jiafu Wan, Lei Shu, Peng Li, Mithun Mukherjee, and Boxing Yin, *Smart Factory of Industry 4.0: Key Technologies, Application Case, and Challenges*, IEEE Access, Vol. 6, pp. 6505-6519, 2017.

[33] Hyoung Seok Kang, Ju Yeon Lee, SangSu Choi, Hyun Kim, Jun Hee Park, Ji Yeon Son, Bo Hyun Kim, and Sang Do Noh, *Smart Manufacturing: Past Research, Present Findings, and Future Directions*, International Journal Of Precision Engineering And Manufacturing-Green Technology Vol. 3, No. 1, pp. 111-128, 2016.

[34] Luo Qi, *Research on Intelligent Transportation System Technologies and Applications*, Workshop on Power Electronics and Intelligent Transportation System, IEEE, Guangzhou, China, 2018.

[35] Rabab Hassan, and Ghadir Radman, *Survey on Smart Grid*, International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS), IEEE, Pudukkottai, India, 2010.

[36] Aqeel-ur-Rehman, and Zubair A. Shaikh, *Smart Agriculture*, Applications of Modern High Performance Networks, Bentham Science Publishers, pp. 120-129, 2009.

[37] Feng Chen, Pan Deng, Jiafu Wan, Daqiang Zhang, Athanasios V.Vasilakos, and Xiaohui Rong , *Data Mining for the Internet of Things: Literature Review and Challenges*, International Journal of Distributed Sensor Networks, 2015.

[38] M.B.Ibanez, A.Di Serio, D.Villaran, and C.D.Kloos, *Experimenting with electromagnetism using augmented reality Impact on flow student experience and educational effectiveness* , Computers and Education, Vol.71, pp. 1-13, 2014.

[39] L.-f. Kwok, *A vision for the development of i-campus*, Smart Learning Environments, vol. 2, pp. 112, 2015.

[40] Granados J., Westerlund T., Zheng L., and Zou Z., *IoT Platform for Real-Time Multichannel ECG Monitoring and Classification with Neural Networks*, Research and Practical Issues of Enterprise Information Systems, CONFENIS 2017, Lecture Notes in Business Information Processing, Springer, Cham, Vol. 310, 2018.

[41] Salha M. Alzahrani, *Development of IoT Mining Machine for Twitter Sentiment Analysis: Mining in the Cloud and Results on the Mirror*, 15th Learning and Technology Conference, Jeddah, Saudi Arabia, IEEE, 2018.

[42] Sanjay Sareen, Sandeep K. Sood, and Sunil Kumar Gupta, *IoT-based cloud framework to control Ebola virus outbreak*, J Ambient Intell Human Comput., Springer, Vol. 9, pp. 459-476, 2016.

[43] A. Suresh, R. Udendhran, M. Balamurgan, and R. Varatharajan, *A Novel Internet of Things Framework Integrated with Real Time Monitoring for Intelligent Healthcare Environment*, Journal of Medical Systems, Springer, Vol. 43, 2019.

[44] Viriyavisuthisakul S., Sanguansat P., Toriumi S., Hayashi M., and Yamasaki T., *An Automatic People Counter in Stores Using a Low-Cost IoT Sensing Platform*, Advances in Intelligent Information Hiding and Multimedia Signal Processing, IIH-MSP 2017, Smart Innovation, Systems and Technologies, Springer, Cham, Vol. 81, 2018.

[45] Xiangbo Kong, Zelin Meng, Naoto Nojiri, Yuji Iwahori, Lin Meng, and Hiroyuki Tomiyama, *A HOG-SVM based fall detection IoT system for elderly persons using deep sensor*, Procedia Computer Science, Vol. 147, pp. 276282, 2019.

[46] Der-Chen Huang, Chun-Fu Lin, Chih-Yen Chen, and Jyh-Rou Sze, *The Internet Technology for Defect Detection System with Deep Learning Method in Smart Factory*, 4th International Conference on Information Management, IEEE, Oxford, UK, 2018.

[47] AKM Jahangir Alam Majumder, Yosuf Amr ElSaadany, Roger Young Jr., and Donald R.Ucci, *An Energy Efficient Wearable Smart IoT System to Predict Cardiac Arrest*, Advances in Human-Computer Interaction, Hindawi, 2019.

[48] Albert Rego, Alejandro Canovas, Jose M. Jimnez, and Jaime Lloret, *An Intelligent System for Video Surveillance in IoT Environments*, IEEE Access, Vol. 6, pp. 31580-31598, 2018.

[49] Amrita A. Joshi, and B.D.Jadhav, *Monitoring and Controlling Rice Diseases Using Image Processing Techniques*, International Conference on Computing, Analytics and Security Trends (CAST), IEEE, 2016.

[50] D.Ganga, and V. Ramachandran, *IoT based Vibration Analytics of Electrical Machines*, IEEE Internet of Things Journal, 2017.

[51] Serge Thomas, Mickala Bourobou, and Younghwan Yoo, *User Activity Recognition in Smart Homes Using Pattern Clustering Applied to Temporal ANN Algorithm*, Sensors, Vol. 15, pp. 11953-11971, 2015.

[52] Manish Joshi, Bramah Hazela, and Vineet Singh, *An application of IoT on Hungarian database using Data mining Techniques: A collaborative approach*, 3rd International Conference on Advances in Computing, Communication and Automation, IEEE, Dehradun, India, 2017.

[53] Xishuang Dong, Lijun Qian, Lei Huang, *Short-Term Load Forecasting in Smart Grid: A Combined CNN and K-Means Clustering Approach*, International Conference on Big Data and Smart Computing (BigComp), IEEE, 2017.

[54] Alekha Kumar Mishra, Asis Kumar Tripathy, Deepak Puthal, and Laurence T. Yang, *Analytical Model for Sybil Attack Phases in Internet of Things*, IEEE, Internet of Things Journal, 2018.

[55] John M Dennis, Beverley M Shields, William E Henley, Angus G Jones, and Andrew T Hattersley, *Disease progression and treatment response in data-driven subgroups of type 2 diabetes compared with models based on simple clinical features: an analysis using clinical trial data*, Lancet Diabetes Endocrinol, 2019.

[56] Prachi Shukla, *ML-IDS: A Machine Learning Approach to Detect Wormhole Attacks in Internet of Thing*, Intelligent Systems Conference, IEEE, London, UK, 2017.

[57] Aras Can Onal, Omer Berat Sezer, Murat Ozbayoglu, and Erdogan Dogdu,*Weather Data Analysis and Sensor Fault Detection Using An Extended IoT Framework with Semantics, Big Data, and Machine Learning*, International Conference on Big Data (BIGDATA), IEEE, Boston, MA, USA., 2017.

[58] Mohd Abdullah Al Mamun, Jinat Afroj Puspo, and Amit Kumar Das, *An Intelligent Smartphone Based Approach Using IoT for Ensuring Safe Driving*, International Conference on Electrical Engineering and Computer Science (ICECOS), IEEE, Palembang, Indonesia, 2017.

[59] Raul parada, Kamruddin Nur, Joan Melia-Segui, and Rafael Pous, *Smart Surface: RFID-Based Gesture Recognition Using k-Means Algorithm*, 12th International Conference on Intelligent Environments, IEEE, London, UK, 2016.

[60] Mohiuddin Ali Khan, Sateesh Kumar Pradhan, and Huda Fatima, *An Efficient Technique for Apriori Algorithm in Medical Data Mining*, Innovations in Computer Science and Engineering, Springer, 2019.

[61] Martin Atzmueller, Naveed Hayat, Matthias Trojahn, and Dennis Kroll, *Explicative Human Activity Recognition using Adaptive Association Rule-Based Classification*, International Conference on future IoT technologies (Future IoT), IEEE, Eger, Hungary, 2018.

[62] Gevorg Poghosyan, Ioannis Pefkianakis, Pascal Le Guyadec, and Vassilis Christophides, *Extracting Usage Patterns of Home loT Devices*, Symposium on Computers and Communications (ISCC), IEEE, 2017.

[63] David Fumo, *Types of Machine Learning Algorithms You Should Know*, https://towardsdatascience.com/types-of-machine-learning-algorithms-you-should-know-953a08248861, 2017.

[64] Amanpreet Singh, Narina Thakur, and Aakanksha Sharma, *A Review of Supervised Machine Learning Algorithms*, 3rd International Conference on Computing for Sustainable Global Development, IEEE, New Delhi, India, 2016.

[65] Jian Qin, Ying Liu, and Roger Grosvenor, *Data analytics for energy consumption of digital manufacturing systems using Internet of Things method*, 13th Conference on Automation Science and Engineering (CASE), IEEE, Xi'an, China, 2017.

[66] Igor Bisio, Alessandro Delfino, Fabio Lavagetto, and Andrea Sciarrone, *Enabling IoT for In Home rehabilitation Accelerometer signals classification methods for activity and movement recognition*, IEEE Internet of Things Journal, 2016.

[67] Ani R, Krishna S, Anju N, Sona Aslam M, and O.S Deepa, *IoT Based Patient Monitoring and Diagnostic Prediction Tool using Ensemble Classifier*, International Conference on Advances in Computing, Communications, and Informatics (ICACCI), IEEE, Udupi, India, 2017.

[68] Pooja A. Dhobi, and Niraj Tevar, *IoT Based Home Appliances Control*, Proceedings of the International Conference on Computing Methodologies and Communication (ICCMC), IEEE, Erode, India, 2017.

[69] Lester James V. Miranda, Marian Joice S. Gutierrez, Samuel Matthew G. Dumlao, and Rosula SJ Reyes, *Appliance Recognition using Hall Effect Sensors and K-Nearest Neighbors for Power Management Systems*, Region 10 Conference (TENCON): IEEE, Singapore, 2016.

[70] Allaa R.Hilal, Aya Sayedelahl, Arash Tabibiazar, Mohamed S.Kamel, and Otman A.Basir, *A distributed sensor management for large-scale IoT indoor acoustic surveillance*, Future Generation Computer Systems, Elsevier, 2018.

[71] Alex Shenfield, David Day, and Aladdin Ayesh, *Intelligent intrusion detection systems using artificial neural networks*, ICT express, Elsevier, Vol.4, pp. 95-99, 2018.

[72] Greene D., Cunningham P., and Mayer R., *Unsupervised Learning and Clustering*, Machine Learning Techniques for Multimedia. Cognitive Technologies. Springer, Berlin, Heidelberg, 2018.

[73] Jiachen Yang, Yurong Han, Yafang Wang, Bin Jiang, Zhihan Lv, and Houbing Song, *Optimization of Real Time Traffic Network Assignment Based on IoT Data Using DBN and Clustering Model in Smart City*, Future Generation Computer Systems, Elsevier, 2017.

[74] S. Sridhar Raj, and M. Nandhini, *Ensemble human movement sequence prediction model with apriori based probability tree classifier (APTC) and Bagged J48 on machine learning*, Journal of King Saud University-Computer and Information Sciences, Elsevier, 2018.

[75] Pinheiro, Fabiola, Kuo, Mu-Hsing, Thomo, Alex, Barnett, and Jeff, *Extracting association rules from liver cancer data using the FP-growth algorithm*, 3rd International Conference on Computational Advances in Bio and Medical Sciences (ICCABS), IEEE, New Orleans, LA, USA, 2013.

[76] Leslie Pack Kaelbling, Michael L. Littman, and Andrew W. Moore, *Reinforcement Learning: A survey*, Journal of Artificial Intelligence Research, Vol. 4, pp. 237-285, 1996.

[77] Kostas Kolomvatsos, and Christos Anagnostopoulos, *Reinforcement Learning for Predictive Analytics in Smart Cities*, Informatics, MDPI, Vol. 3, No. 16, 2017.

[78] Alexandra Lheureux, Katarina Grolinger, Hany F. Elyamany, and Miriam A. M. Capretz, *Machine Learning With Big Data: Challenges and Approaches*, IEEE Access, Vol. 5, pp. 7776-7797, 2017.

[79] X. Ma, H. Yu, Y. Wang, and Y. Wang, *Large-scale transportation network congestion evolution prediction using deep learning theory*, PloS one, Vol. 10, No. 3, 2015.

[80] X. Wang, L. Gao, S. Mao, and S. Pandey, *Deepfi: Deep learning for indoor fingerprinting using channel state information*, Wireless Communications and Networking Conference (WCNC), IEEE, New Orleans, LA, USA, 2015.

[81] brahim KK, Mehmet Ulvi MEK, and Suat ZDEMR, *A deep learning model for air quality prediction in smart cities*, International Conference on Big Data (BIGDATA), IEEE, Boston, MA, USA, 2017.

[82] F. J. Ordonez, and D. Roggen, *Deep convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition*, Sensors, Vol. 16, 2016.

[83] Z. Yuan, Y. Lu, Z. Wang, and Y. Xue, *Droid-sec: deep learning in android malware detection*, Computer Communication Review, SIGCOMM, ACM, Vol. 44, No. 4, pp. 371372, 2014.

[84] K. Lim, Y. Hong, Y. Choi, and H. Byun, *Real-time traffic sign recognition based on a general purpose gpu and deep-learning*, PLoS one, Vol. 12, No. 3, 2017.

[85] N. Kussul, M. Lavreniuk, S. Skakun, and A. Shelestov, *Deep learning classification of land cover and crop types using remote sensing data*, Geoscience and Remote Sensing Letters, IEEE, 2017.

[86] H. Shao, H. Jiang, F. Wang, and H. Zhao, *An enhancement deep feature fusion method for rotating machinery fault diagnosis*, Knowledge Based Systems, Elsevier, Vol. 119, pp. 200220, 2017.

[87] Salar Razavi, and Hulya Yalcin, *Plant classification using group of features*, 24th Signal Processing and Communication Application Conference (SIU), IEEE, 2016.

[88] A. Toshev, and C. Szegedy, *Deeppose: Human pose estimation via deep neural networks*, Conference on Computer Vision and Pattern Recognition, IEEE, Columbus, OH, USA, 2014.

[89] Alexander Gepperth, and Barbara Hammer, *Incremental learning algorithms and applications*, proceedings of European Symposium on Artificial Neural Networks, Computational Intelligence, and Machine Learning, Belgium, 2016.

[90] Haibo He, Sheng Chen, Kang Li, and Xin Xu, *Incremental Learning from Stream Data*, IEEE Transactions on Neural Networks, Vol. 22, No. 12, pp. 1901-1914, 2011.

[91] Rashmika Nawaratne, Damminda Alahakoon, Daswin De Silva, Prem Chhetri, and Naveen Chilamkurti, *Self-evolving intelligent algorithms for facilitating data interoperability in IoT environments*, Future Generation Computer Systems, Elsevier, 2018.

[92] Arun Kishore Ramakrishnan, Davy Preuveneers, and Yolande Berbers, *Enabling self learning in Dynamic and open IoT environments*, Procedia Computer Science, Elsevier, Vol. No. 32, pp. 207-214.

[93] Alexander Gepperth, and Barbara Hammer, *Incremental learning algorithms and applications*, Proceedings of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), Bruges, Belgium, 2016.

[94] Lisa Torrey, Jude Shavlik, *Transfer Learning Handbook of Research on Machine Learning Applications*, IGI Global, 2009.

[95] Jason Brownlee, *A Gentle Introduction to Transfer Learning for Deep Learning*, https://machinelearningmastery.com/transfer-learning-for-deep-learning/, 2017.

[96] Pranoy Radhakrishnan, *What is transfer Learning?*, https://towardsdatascience.com/what-is-transfer-learning-8b1a0fa42b4, 2017.

[97] Md Abdullah Al Hafiz Khan, and Nirmalaya Roy, *UnTran Recognizing Unseen Activities with Unlabeled data using Transfer Learning*, Third International Conference on Internet of Things Design and Implementation, IEEE/ACM, Orlando, FL, USA, 2018.

[98] Long D. Nguyen, Dongyun Lin, Zhiping Lin, and Jiuwen Cao, *Deep CNNs for microscopic image classification by exploiting transfer learning and feature concatenation*, International Symposium on Circuits and Systems (ISCAS), IEEE, Florence, Italy, 2018.

[99] Chang-Jiun Chen, Kai-Chun Chen, and May-chen Martin-Kuo, *Acceleration of Neural Network Model Execution on Embedded Systems*, International Symposium on VLSI Design, Automation and Test (VLSI-DAT), IEEE, Hsinchu, Taiwan, 2018.

[100] Lei Yu, Jieping Ye, and Huan Liu, *Dimensionality Reduction for Data Mining - Techniques, Applications and Trends*, http://www.cs.binghamton.edu/ lyu/SDM07/DR-SDM07.pdf.

[101] *Dimensionality Reduction algorithms: Strengths and weaknesses*, https://elitedatascience.com/dimensionality-reduction-algorithms.

[102] Jing Liang, Xiaoxu Liu, and Kuo Liao, *Soil Moisture Retrieval using UWB Echoes via Fuzzy Logic and Machine Learning*, IEEE Internet of Things Journal, Vol. 14, No. 8, 2015.

[103] Yong-Ping Chen, Qi-Hui Chen, Kuan-Yu Chou, and Ren-Hau Wu, *Low-Cost Face Recognition System Based on Extended Local Binary Pattern*, International Automatic Control Conference (CACS), IEEE, Taiwan, 2016.

[104] Mohammed Faisal Elrawy, Ali Ismail Awad, and Hesham F. A. Hamed, *Flow based Features for a Robust Intrusion Detection System Targeting Mobile Traffic*, 23rd International Conference on Telecommunications (ICT), IEEE, Thessaloniki, Greece, 2016.

[105] Hongyang Zhang, Junqi Guo, Xiaobo Xie, Rongfang Bie, and Yunchuan Sun, *Environmental Effect Removal Based Structural Health Monitoring in the Internet of Things*, Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, IEEE, Taichung, Taiwan, 2013.

[106] Gilberto Fernandes Jr., Joel J. P. C. Rodrigues, Luiz Fernando Carvalho, Jalal F. Al-Muhtadi, and MarioLemes Proena Jr., *A comprehensive survey on network anomaly detection*, Telecommunication Systems, Springer, 2018.

[107] Gil-Yong Lee et al., *Machine health management in smart factory: A review*, Journal of Mechanical Science and Technology, Springer, Vol. 32, No. 3, pp. 987-1009, 2018.

[108] R. Varatharajan, Gunasekaran Manogaran, and M. K. Priyan, *A big data classification approach using LDA with an enhanced SVM method for ECG signals in cloud computing*, Multimed Tools Appl, Springer.

[109] Sina Dami, and Mahtab Yahaghizadeh, *Efficient Event Prediction in an IOT Environment Based On LDA Model and Support Vector Machine*, 6th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS), IEEE, Kerman, Iran, 2018.

[110] Kabilan N, and Dr. M. Senthamil Selvi, *Surveillance and Steering of Irrigation System in Cloud using Wireless Sensor Network and Wi-Fi Module*, Fifth International Conference on Recent Trends in Information Technology, IEEE, Chennai, India, 2016.

[111] Z. Huang, K.-J. Lin, B.-L. Tsai, S. Yan, and C.-S. Shih, *Building edge intelligence for online activity recognition in service-oriented IoT systems*, Future Generation Computer Systems, Elsevier, 2018.

[112] Eduardo Viegas, Altair Santin, Luiz Oliveira, Andre Franca, Ricardo Jasinski, and Volnei Pedroni, *A Reliable and Energy Efficient Classifier Combination Scheme for Intrusion Detection in Embedded Systems*, Computers and Security, Elsevier, Vol. 78, pp. 16-32, 2018.

[113] Henry Friday Nweke, Ying Wah Teh, Mohammed Ali Algaradi, and Uzoma Rita Alo, *Deep Learning Algorithms for Human Activity Recognition using Mobile and Wearable Sensor Networks State of the Art and Research Challenges*, Expert Systems With Applications, Elsevier, Vol. 105, pp. 233-261, 2018.

[114] Mohammad Malekzadeh, Richard G. Clegg, and Hamed Haddadi, *Replacement AutoEncoder: A Privacy-Preserving Algorithm for Sensory Data Analysis*, Third International Conference on Internet-of-Things Design and Implementation,

IEEE, IEEE/ACM, Orlando, FL, USA, 2018.

[115] Mohammad Kachuee, Anahita Hosseini, Babak Moatamed, Sajad Darabi, Majid Sarrafzadeh, *Context-Aware Feature Query To Improve The Prediction Performance*, Global Conference on Signal and Information Processing (GlobalSIP), IEEE, Montreal, QC, Canada, 2017.

[116] Homayoun S., Ahmadzadeh M., Hashemi S., Dehghantanha A., and Khayami R., *BoTShark: A Deep Learning Approach for Botnet Traffic Detection*, Cyber Threat Intelligence, Advances in Information Security, Vol 70. Springer, Cham.

[117] Guifang Liu, Huaiqian Bao, and Baokun Han, *A Stacked Autoencoder-Based Deep Neural Network for Achieving Gearbox Fault Diagnosis*, Mathematical Problems in Engineering, Hindawi.

[118] Lianbing Deng, Daming Li, Xiang Yao, David Cox, and Haoxiang Wang, *Mobile network intrusion detection for IoT system based on transfer learning algorithm*, Cluster Computing, Springer, 2018.

[119] G.Nagarajan, R.I.Minu, B Muthukumar, V.Vedanarayanan, and S.D.Sundarsingh, *Hybrid Genetic Algorithm for Medical Image Feature Extraction and selection*, International Conference on Computational Modeling and Security (CMS), Procedia Computer Science, Elsevier, Vol. 85, pp. 455-462, 2016.

[120] Sourabh S. Patil, and Dr. Mrs. A. S. Bhalchandra, *Pattern Recognition Using Genetic Algorithm*, International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), IEEE, Palladam, India.

[121] Jonathan Reynolds, Yacine Rezgui, Alan Kwan, Solne Piriou, and A ZoneLevel, *Building Energy Optimisation Combining an Artificial Neural Network, a Genetic Algorithm, and Model Predictive Control*, Energy, Elsevier, Vol. 151, pp. 729-739, 2018.

[122] Rosa Altilio, Luca Liparulo, Andrea Proietti, Marco Paoloni, and Massimo Panella, *A Genetic Algorithm for Feature Selection in Gait Analysis*, IEEE Congress on Evolutionary Computation (CEC), IEEE, Vancouver, BC, Canada, 2016.

[123] Sameh Ben Fredj, *Analyzing IoT Data: Introduction to Time Series Forecasting with Python*, http://iot-ee.com/en/2017/08/07/analysing-iot-data-introduction-time-series-forecasting-python/, 2017.

[124] Ratnadip Adhikari, and R. K. Agrawal, *An Introductory Study on Time Series Modeling and forecasting*, Available: https://arxiv.org/ftp/arxiv/papers/1302/1302.6613.pdf.

[125] Siddhartha Bhandari, Neil Bergmann, Raja Jurdak, and Branislav Kusy, *Time Series Data Analysis of Wireless Sensor Network Measurements of Temperature*, Sensors, Vol. 17, 2017.

[126] Jeff Morrison, *Autoregressive Integrated Moving Average Models (ARIMA)*, http://www.forecastingsolutions.com/arima.html

[127] Ameeth Kanawaday, and Aditya Sane, *Machine Learning for Predictive Maintenance of Industrial Machines using IoT Sensor Data*, 8th International Conference on Software Engineering and Service Science, IEEE, Beijing, China, 2017.

[128] Gaurav Chavan, and Dr. Bashirahamad Momin, *An Integrated approach for Weather Forecasting over Internet of Things: A Brief Review*, International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), IEEE, Palladam, India, 2017.

[129] Basheer Qolomany, Ala Al-Fuqaha, Driss Benhaddou, and Ajay Gupta, *Role of Deep LSTM Neural Networks And Wi-Fi Networks in Support of Occupancy Prediction in Smart Buildings*, 19th International Conference on High Performance Computing and Communications; 15th International Conference on Smart City; 3rd International Conference on Data Science and Systems, IEEE, Bangkok, Thailand, 2017.

[130] Rodrigoda Rosa Righi, Everton Correa, Mrcio Migu el Gomes, and Cristiano Andrda Costa, *Enhancing performance of IoT applications with load prediction and cloud elasticity*, Future Generation Computer Systems, Elsevier, 2018.

[131] Chirag Deb, Fan Zhang, Junjing Yang, Siew Eang Lee, and Kwok Wei Shah, *A review on time series forecasting techniques for building energy consumption*, Renewable and Sustainable Energy Reviews, Elsevier, Vol. 74, pp. 902-924, 2017.

[132] Bilal Esmael, Arghad Arnaout, RudolfK. Fruhwirth, and Gerhard Thonhauser, *Improving Time Series Classification Using Hidden Markov Models*, 12th International Conference on Hybrid Intelligent Systems, IEEE, Pune, India, 2012.

[133] Palani kumar, and Meenakshi DSouza, *Design a Power Aware Methodology in IoT based on Hidden Markov Model*, 9th International Conference on Communication Systems and Networks (COMSNETS), IEEE, Banglore, India.

[134] Jun Qi, Po Yang, Martin Hanneghan, Stephen Tang, Bo Zhou, *A Hybrid Hierarchical Framework for Gym Physical Activity Recognition and Measurement Using Wearable Sensors*, IEEE Internet of Things Journal, 2018.

[135] Ramkumar Eswaraprasad,and Linesh Raja, *Improved Intelligent Transport System for Reliable Traffic Control Management by Adapting Internet of Things*, International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions), IEEE, Dubai, United Arab Emirates, 2017.

[136] Maryem Neyja, Shahid Mumtaz, Kazi Mohammed Saidul Huq, Sherif Adeshina Busari, Jonathan Rodriguez, and Zhenyu Zhou, *An IoT-Based E-Health Monitoring System Using ECG Signal*, Global Communications Conference, IEEE, Singapore, 2017.

[137] Yihang Cheng, Yuanyuan Qiao, and Jie Yang, *An Improved Markov Method for Prediction of User Mobility*,IFIP, 2016.

[138] M. Tariq, H. Majeed, M.O. Beg, F.A. Khan, and A. Derhab, *Accurate detection of sitting posture activities in a secure IoT based assisted living environment*,Future Generation Computer Systems, Elsevier, 2018.

[139] Qingchen Zhang, Laurence T. Yang, Zhikui Chen , and Peng Li, *A survey on deep learning for big data*, Information Fusion, Elsevier, Vol. 42, pp. 146-157, 2018.

[140] Jiho Park, Kiyoung Jang, and Sung-Bong Yang, *Deep Neural Networks for Activity Recognition with Multi-Sensor Data in a Smart Home*, 4th World Forum on Internet of Things (WF-IoT), Singapore, 2018.

[141] Manuel Lopez-Martin, Belen Carro, Antonio Sanchez-Esguevillas, and Jaime Lloret, *Network Traffic Classifier*

With Convolutional and Recurrent Neural Networks for Internet of Things, IEEE Access, Vol. 5, pp. 18042-18050, 2017.

[142] Shin Kamada, and Takumi Ichimura, *Knowledge Extracted from Recurrent Deep Belief Network for Real Time Deterministic Control*, International Conference on Systems, Man, and Cybernetics (SMC), IEEE, Banff, Canada, 2017.

[143] A.J. Hussain et al., *A dynamic neural network architecture with immunology inspired optimization for weather data forecasting*, Big Data Research, Elsevier, 2018.

[144] Singh, Parminder, Pooja Gupta, and Kiran Jyoti, *Tasm: technocrat arima and svr model for workload prediction of web applications in cloud*, Cluster Computing Vol. 22, No. 2, pp. 619-633, 2019.

[145] Singh, Parminder, Pooja Gupta, Kiran Jyoti, and Anand Nayyar, *Research on Auto-Scaling of Web Applications in Cloud: Survey, Trends and Future Directions*, Scalable Computing: Practice and Experience Vol. 20, No. 2, pp. 399-432, 2019.

[146] Singh, Parminder, Pooja Gupta, and Kiran Jyoti, *Triangulation Resource Provisioning for Web Applications in Cloud Computing: A Profit-Aware Approach*,Scalable Computing: Practice and Experience Vol. 20, No. 2, pp. 207-222, 2019.

[147] Mung Chiang, and Tao Zhang, *Fog and IoT: An Overview of Research Opportunities*, IEEE Internet of Things Journal, Vol. 3, No. 6, pp. 854-864, 2016.

[148] Qingchen Zhang, Laurence T. Yang, Zhikui Chen, Peng Li, and Fanyu Bu, *An Adaptive Dropout Deep Computation Model for Industrial IoT Big Data Learning with Crowdsourcing to Cloud Computing*, IEEE Transactions On Industrial Informatics, 2017.

[149] Sahitya Roy, Dr Rajarshi Ray, Aishwarya Roy, Subhajit Sinha, Gourab Mukherjee, Supratik Pyne, Sayantan Mitra, Sounak Basu, and Subhadip Hazra, *IoT, Big Data Science and Analytics, Cloud Computing and Mobile App based Hybrid System for Smart Agriculture*, 8th Industrial Automation and Electromechanical Conference (IEMECON), IEEE, Bangkok, Thailand, 2017.

[150] P. Verma, and S.K. Sood, *Cloud-centric IoT based disease diagnosis healthcare framework*, Journal of Parallel and Distributed Computing, Elsevier, Vol. 116, pp. 27-38, 2017.

[151] Karim Foughali, Karim Fathallah, and Ali Frihida, *Using Cloud IOT for disease prevention in precision agriculture*, Procedia Computer Science, Elsevier, Vol. 130, pp. 575-582, 2018.

[152] Pablo Palacios, and Andres Cordova, *Approximation and Temperature Control System via an Actuator and a Cloud An Application Based on the IoT for Smart Houses*, International Conference on eDemocracy and eGovernment (ICEDEG), IEEE, Ambato, Ecuador, 2018.

[153] Luca Cerina, Sara Notargiacomo, Matteo Greco, Luca Paccani, and Marco Domenico Santambrogio, *A Fog-Computing architecture for Preventive Healthcare and Assisted Living in Smart Ambients*, 3rd International Forum on Research and Technologies for Society and Industry (RTSI), IEEE, Modena, Italy, 2017.

[154] Ning Chen, Yu Chen, Yang You, Haibin Ling, Pengpeng Liang, and Roger Zimmermann, *Dynamic Urban Surveillance Video Stream Processing Using Fog Computing*, Second International Conference on Multimedia Big Data (BigMM), IEEE, Taipei, Taiwan, 2016.

[155] Hamid Reza Arkian, Abolfazl Diyanat, and Atefe Pourkhalili, *MIST: Fog-based Data Analytics Scheme with Cost-Efficient Resource Provisioning for IoT Crowdsensing Applications*, Journal of Network and Computer Applications, Elsevier, Vol. 82, pp. 152-165, 2017.

[156] Augusto J. V. Neto, Zhongliang Zhao, Joel J. P. C. Rodrigues, Hugo B. Camboim, and Torsten Braun, *Fog-based Crime-Assistance in Smart IoT Transportation System*, IEEE Access, 2018.

[157] Bo Tang, Zhen Chen, Gerald Hefferman, Shuyi Pei, Tao Wei, Haibo He, and Qing Yang, *Incorporating Intelligence in Fog Computing for Big Data Analysis in Smart Cities*, IEEE Transactions on Industrial Informatics, 2016.

[158] S.Bi, R. Zhang, Z. Ding, and S. Cui, *Wireless communications in the era of big data*, IEEE Commun. Mag., Vol.53, No.10, pp.190199, Oct.2015.

[159] *Apache Hadoop*, https://hadoop.apache.org/

[160] *Apache Spark*, http://spark.apache.org/

[161] *Apache Storm*, http://storm.apache.org

[162] *Apache Flink*, https://flink.apache.org/

[163] *Azure Stream Analytics*, https://azure.microsoft.com/en-in/services/stream-analytics/

[164] Muhammad Mazhar Ullah Rathore, Awais Ahmad, Anand Paul, and Dr. Seungmin Rho, *Urban planning and building smart cities based on the Internet of Things*, Computer Networks, Vol. 101, pp. 63-80, 2016.

[165] Anand Paul, Awais Ahmad, M. Mazhar Rathore, and Sohail Jabbar, *Smartbuddy: Defining Human Behaviors Using Big Data Analytics In Social Internet Of Things*, Wireless Communications, IEEE, pp. 68-74, 2016.

[166] Muhammad Rizwan Bashir, and Asif Qumer Gill, *Towards an IoT Big Data Analytics Framework: Smart Buildings Systems*, 18th International Conference on High Performance Computing and Communications, IEEE, Sydney, NSW, Australia, 2016.

[167] Zaheer Khan, Ashiq Anjum, Kamran Soomro, and Muhammad Atif Tahir, *Towards cloud based big data analytics for smart future cities*, Journal of Cloud Computing Advances, Systems and Applications, Vol. 4, No. 2, 2015.

[168] Muhammad Syafrudin, Ganjar Alfian, Norma Latif Fitriyani, and Jongtae Rhee, *Performance Analysis of IoT Based Sensor, Big Data Processing, and Machine Learning Model for Real Time Monitoring System in Automotive Manufacturing*, Sensors, MDPI, 2018.

[169] Roger Young, Sheila Fallon, and Paul Jacob, *An Architecture for Intelligent Data Processing on IoT Edge Devices*, 19th International Conference on Modelling and Simulation, IEEE, Cambridge, UK, 2017.

[170] Michael Zimmermann, Felix W. Baumann, Michael Falkenthal, Frank Leymann, and Ulrich Odefey, *Automating the Provisioning and Integration of Analytics Tools with Data Resources in Industrial Environments using OpenTOSCA*, 21st International Enterprise Distributed Object Computing Conference Workshops, IEEE, Quebec City, QC, Canada,

2017.

[171] Giacomo Marciani, Marco Piu, Michele Porretta, and Matteo Nardelli, *Grand Challenge: Real-time Analysis of Social Networks Leveraging the Flink Framework*, DEBS, ACM, Irvine, CA, USA, 2016.

[172] Aleksandr Farseev, Ivan Samborskii, and Tat-Seng Chua, *bBridge: A Big Data Platform for Social Multimedia Analytics*, ACM, Amsterdam, The Netherlands, 2016.

[173] Patric Boscolo, *Real time monitoring and performance analysis of power plants at BaxEnergy*, https://microsoft.github.io/techcasestudies/iot/2017/06/30/baxenergy.html, 2017.

[174] Ahmet Cihat Baktir, Atay Ozgovde, and Cem Ersoy, *How Can Edge Computing Benefit from Software-Defined Networking: A Survey, Use Cases and Future Directions*, IEEE Communications Surveys and Tutorials, Vol. 19, pp. 2359-2391, 2017.

[175] Samaresh Bera, Sudip Misra, and Athanasios V. Vasilakos, *Software-Defined Networking for Internet of Things: A Survey*, IEEE Internet of Things Journal, Vol. 4, No. 6, pp. 1994-2008, 2017.

[176] A.Rego, L.Garcia, S.Sendra, and J.Lloret, *Software Defined Network-based control system for an efficient traffic management for emergency situations in smart cities*, Future Generation Computer Systems, Elsevier, Vol. 88, pp. 243-253, 2018.

[177] Wani A., and Revathi S, *Analyzing Threats of IoT Networks Using SDN Based Intrusion Detection System (SDIoT-IDS)*, Smart and Innovative Trends in Next Generation Computing Technologies, NGCT, Communications in Computer and Information Science, Vol. 828, Springer, Singapore, 2018.

[178] Arbiza L.M.R., Tarouco L.M.R., Bertholdo L.M., and Granville L.Z.S, *SDN-Based Service Delivery in Smart Environments*, Intelligent Distributed Computing IX, Studies in Computational Intelligence, Vol. 616, Springer, Cham, 2016.

[179] Raul Muoz, Ricard Vilalta, Noboru Yoshikane, Ramon Casellas, Ricardo Martnez, Takehiro Tsuritani, and Itsuro Morita, *IoT-aware Multi-layer Transport SDN and Cloud Architecture for Traffic Congestion Avoidance Through Dynamic Distribution of IoT Analytics*, European Conference on Optical Communication (ECOC), IEEE, Gothenburg, Sweden, 2017.

[180] Raul Muoz, Ricard Vilalta, Noboru Yoshikane, Ramon Casellas, Ricardo Martnez, Takehiro Tsuritani, and Itsuro Morita, *Integration of IoT, Transport SDN and Edge/Cloud computing for Dynamic Distribution of IoT Analytics and Efficient Use of Network Resource*, Journal of Lightwave Technology, Vol. 36, No. 7, pp. 1420-1428, 2018.

[181] Jose Santos, Philip Leroux, Tim Wauters, Bruno Volckaert, F.D Turck, *Anomaly detection for Smart City applications over 5G Low Power Wide Area Network*, Network Operations and Management Symposium, Taiwan, 2018.

Edited by: Ashish Kumar Luhach
Received: May 21, 2019
Accepted: Oct 25, 2019

DOI 10.12694/scpe.v20i4.1563
Scalable Computing: Practice and Experience ISSN 1895-1767
Volume 20, Number 4, pp. 631–639. http://www.scpe.org © 2019 SCPE

PERFORMANCE OF ENERGY CONSERVATION MODELS, GENERIC, MICAZ AND MICAMOTES, USING AODV ROUTING PROTOCOL ON A WIRELESS SENSOR NETWORK

SURESH KUMAR*[1], KIRAN DHULL*, PAYAL ARORA*AND ASHISH KR. LUHACH†

Abstract. Wireless Sensor Networks (WSN's) have gained a considerable importance and are used for a variety of applications. In WSN, an arrangement of sensor nodes is done to sense and collect information from its nearby environment and to send it back to the base station using routing protocol. The biggest challenges are how to handle the routing problems and to optimize the energy consumption in WSN. In this paper, performance evaluation of three energy models, Generic, Micaz and Micamotes, is presented using Ad-Hoc On demand Distance Vector (AODV) routing protocol. The performance evaluation is done using several parameters: Throughput, Jitter, Average End-to-End Delay (AEED), Total Packets Received (TPR) and Energy consumption in three modes (transmit, receive and idle). Based on the evaluation, it has been found that Micamotes energy model using AODV routing protocol consumes less energy by 80.46% and 428.57% in transmit mode , 102.94% and 335.6% in receive mode from Micaz and Generic energy models , respectively.

Key words: WSN, AODV, Mobile Ad-hoc Networks (MANETS), Route Reply with Error (RRER), Wireless Personal Area Networks (WPAN), Probabilistic Energy Efficient Routing (PEER), Dynamic MANET On-demand (DYMO), AEED, TPR

AMS subject classifications. 68M10, 68M20, 60K05, 60K25

1. Introduction.

WSN is a group of multi functional sensor nodes used for recording and monitoring the environmental conditions such as sound, pollution level, vibrations, temperature, wind, humidity, seismic events etc. positioned over a geographical area. The architecture of WSN is shown in Figure 1.1. Some applications of WSN are area monitoring, industrial Monitoring, health care monitoring, environment sensing etc. [1].

In WSN, energy can be consumed usefully and wastefully. Hence, to improve the energy efficiency, it is essential to reduce the wastage of energy so that the network can work with better efficiency and more battery lifetime. The main focus of energy efficient protocols is reducing the energy consumption during network activities [2].The protocol design for lifetime-maximization and improvement in performance of lifetime for a WSN is expressed by the average network lifetime. It is defined as the average amount of time until the network expiry [3].

Organization of this paper is as follows. Performance analysis of three energy models Generic, Micaz and Micamotes is done using AODV routing protocol based on several performance metrics. Section 2 gives a brief description of AODV protocol used for simulation and explains several energy models.The related current research work is given in Section 3. Section 4 relates with the simulation model and parameters used for simulation. The results of the simulative model are discussed in Section 5 followed by Conclusion in Section 6.

2. Routing Protocols and Energy Models.

2.1. AODV.

AODV is a new level of the destination sequence distance vector routing protocol for MANETS with a different mechanism for routing information. It is purely an on demand protocol. The important feature of AODV is that it is time based working protocol. AODV gives demand and destination sequence number on the basis of latest information for the route to destination. AODV takes less time to set up connection. Due to all these advantages, AODV has become popular nowadays. It works in two steps: path discovery and path maintenance. In the path discovery phase, a connection is established between source and destination nodes using Route Request (RREQ) and Route Reply (RREP) packets as shown in Figure 2.1.

The source node sends a RREQ to the nearest neighbouring nodes. If a destination node is found, it replies back with a RREP otherwise the RREQ propagates to other neighbours. While in the second phase, the nodes which receive RREP provide information to the source node about their routing table for changes in topology. In case of link breakage, RRER is sent to the source node to again initiate the routing process.

*Department of Electronics and Communication Engineering, University Institute of Engineering and Technology, Maharshi Dayanand University, Rohtak, Haryana, India (sureshvashist.uiet.ece@mdurohtak.ac.in, kirandhull19@gmail.com, payalarora325@gmail.com).

†The PNG University of Technology, Papua New Guinea (ashishluhach@gmail.com).

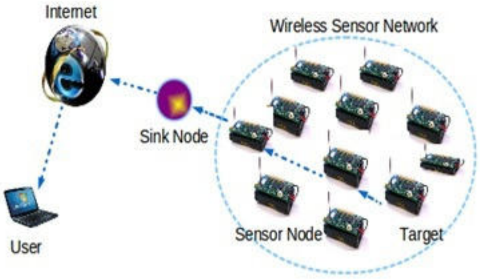


Fig. 1.1. *Architecture of WSN*

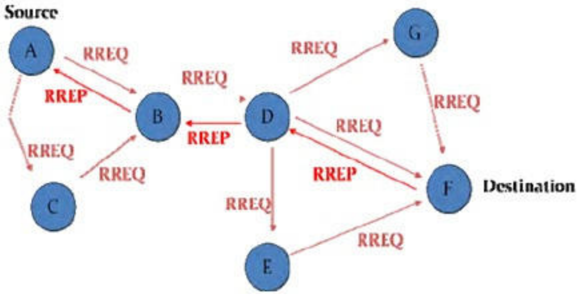


Fig. 2.1. *AODV Routing Protocol*

2.2. ZigBee. It is a high-level communication protocol based on IEEE 802.15.4 specification used in WPAN. Figure 2.2 shows the ZigBee network. It is simple, inexpensive and consumes less power and data rate than other WPANs. It has several applications such as home energy monitors, wireless light switches and traffic management systems. It limits the transmission distance for low power consumption. It is employed in low data rate applications that involve longer battery life, secure networking and low-latency communication. The topologies used in ZigBee layer are Star, Tree and Generic Mesh. In star network topology, there is one coordinator device which is the central node. The communication network can be extended by the use of ZigBee routers in tree and mesh topology.

2.3. Generic. The Generic model is used for reducing the energy consumption by decreasing the task that have to be done by the sensors and their associated networks. Some methods are used to refill the energy capacity of sensor by building components and mechanisms for additional energy harvesting while operation with the environmental conditions [4].

2.4. Micaz. Micaz is a third generation device used for low power WSNs development. It is an IEEE 802.15.4 standard based RF transceiver which uses2.4 GHz Mote module. It works on 2.4 to 2.48 GHz for deeply embedded sensor network. It is always characterised with larger range and high data rate due to 3G mote [5].

2.5. Micamotes. Micamotes is a second generation mote broadly used for enabling low power WSNs by developers and researchers. It always runs at 4MHz and use processor named Atrnel ATmega 128 L that is 8 bit microcontroller and has 128 Kb flash memory for storing the mote program. It consumes 8 A and 15 A in running mode and sleep mode respectively.Micamotes use two AA batteries that can run for more than a year which generate 1000 mA-hour and has small energy consumption. ATmegacan be operated for about 120 hours at 8 mA. Tiny OS is operating system where Micamotes is built [5].

3. Related and Current Research Work. In [6], a comparative analysis of two energy models (Micamotes and Micaz) and modulation formats has been presented to determine the energy efficiency and Quality

Fig. 2.2. *ZigBee Network*

of Service (QoS) of WSN. The authors have revealed that the Micamotes energy model performs efficiently with ASK modulation format than BPSK and O-QPSK. In terms of energy efficiency, Micaz energy model consume less energy than Micamotes in all the WSN modes. The ASK modulation is superior in terms of QoS with higher throughput, lower jitter and delay. The BPSK format has small fraction of error of all the three formats. In [7], authors have compared and analysed routing protocols -AODV and DYMO on two mobile CBR network based on parameters like time required for information transmission from sender to receiver, throughput, jitter and AEED. On simulation, it is found that AODV protocol is better than DYMO at same number of nodes and parameters.

In [8], the authors have proposed a hybrid combination of Genetic Algorithm (GA) with Artificial Neural Network (ANN)for energy minimization in a Wireless Mesh Network. Proposed model is compared with AODV and GA based on hop count, energy consumption and lifetime of nodes. It has been observed that the proposed ANN with GA gives better result with energy saving of 51%. In [9], authors have proposed Modified AODV (M-AODV) protocol to enhance system performance and lifetime of a WSN. The M-AODV protocol has two additional important parameters in HELLO Packets- trust and eagerness of the node which is based on the remaining battery. The simulation results based on Packet Delivery Ratio (PDR), throughput,packets dropped, routing and control overhead indicate that M-AODV is superior compared to AODV protocol.

In [10], the authors have presented a Probabilistic protocol (PEER) to overcome the issue of energy efficiency in a WSN. The overall network's lifetime is improved by balancing the energy consumption and probabilistic forwarding of route requests. On comparison with AODV and Energy Efficient AODV (EEAODV), this protocol shows improvement in battery capacity by 5%. In [11], authors have performed a comparison of DSR and DYMO protocol based on CBR and multi-clustering technique using parameters- throughput, delay, TPR at the PAN coordinator and battery capacity. It was concluded that DYMO performs better in terms of all performance metrics. In [12], a comprehensive review of WSN's has been presented for conservation of energy regarding several design issues such as collection and aggregation of data, clustering and routing based on structure free and structured modelling. From the study of routing protocols, it has been found that a routing protocol is best saving if it minimizes number of data transmissions, the total energy spent and maximizes the number of alive nodes in WSN.

In [13], the authors have presented a review of routing protocols used in Adhoc WSN. It also revealed that the use of ZigBee application with WPAN are highly reliable in mesh networks with many advantages such as lesser cost, increased range, longer battery life and low power consumption.

In [14], the authors have done a comparative analysis of AODV and DSR based on several QoS metrics in AWSN using Zigbee. The AODV routing protocol is found better than DSR for all metrics which therby enhances the QoS.

In [15], the performance of various Ad-Hoc routing protocols (Table Driven- Intrazone Routing Protocol (IARP), On Demand- Interzone Routing Protocol (IERP), Dynamic Source Routing (DSR) and Hybrid- Zone Routing Protocol (ZRP))has been analysed in terms of different performance metrics. It was revealed from the results that IERP and DSR have better efficiency for 3D terrestrial data communication. Also, the authors have suggested that in future, field testing can be used for detecting errors in data. In [16], Mobility Aware and Dual Phase AODV with Adaptive Hello Messages routing protocol is proposed which is an extension of the

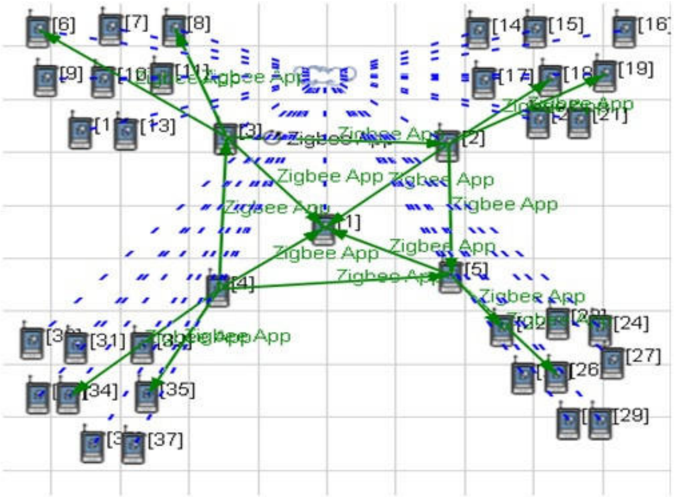


FIG. 4.1. *The scenario architecture*

AODV protocol. This proposed protocol is found efficient for route discovery procedure which thereby helps in finding more stable routes compared to conventional AODV protocol.

In [17], authors have proposed a fuzzy controlled rate and Hello interval based congestion control which on the basis of energy consumed and node mobility changes the interval of HELLO packets. An improvement in performance parameters is found when the HELLO messages are made adaptive. It has been concluded that the frequency of HELLO messages can be modified to control congestion. In [18], authors have evaluated the performance of protocols- AODV, ZRP and DSR to control congestion in MANET's in terms of error packets, collided packets, throughput and number of transmitted packets for different number of nodes. The simulative results indicate that the AODV routing protocol is effective in controlling congestion.

In [19], an algorithm for cluster head selection has been proposed with the LEACH clustering protocol based on distance, residual energy and reliability. From the observation, it has been analysed that proposed algorithm is more efficient, stable and enhances the lifetime of the network by balancing of nodes and efficient information delivery. So far, the researchers have not evaluated the sensor nodes for all the parameters i.e.Throughput, Jitter, AEED, TPR and Energy consumption in all three modes (transmit, receive and idle) together which has motivated us to do this.

4. Simulation Model. In this work, simulation and analysis is done using QualNet 7.3.1 network simulator.The designed network consists of 37 nodes that are placed randomly in the 1000m X 1000m terrain where node 1 act as a sink node which is Full Function Device (FFD) and rest are Reduced Function Devices (RFD). The scenario architecture is shown in Figure 4.1. There are 100 packets of data transmitted with each packet containing about 70 bytes of data. Node 1 is connected to nodes 2, 3, 4 and 5 via wireless link in a star topology that receives the data from central PAN co-ordinator. Some nodes are not involved in the process of communication. ZigBee applications are only applied between nodes 1, 2, 3, 5, 8, 18, 19, 22 and 35. The performance of various energy models such as Generic, Micaz and Micamotes are analysed on the basis of network metrics such as Throughput, Jitter, AEED, TPR and Energy Consumption in all the nodes using AODV routing protocol. The parameters used for simulation are given in Table 4.1.

5. Results and Discussion. In this proposed work, we have simulated the designed network using AODV routing protocol in order to analyze the performance of three energy models i.e. Generic, Micaz and Micamotes. Five performance parameters are taken into consideration which is explained one by one in the following section:

5.1. Throughput. It is defined as the total data arrived at the receiver through network in a given time and always taken in bits/ seconds. The analysis of average throughput obtained in case of three energy models of the proposed WSN is depicted by the Figure 5.1. On analysis of above plotted graph, it can be concluded that

Table 4.1
Scenario Parameters

Parameter	**Value**
Simulator	Qualnet 7.3.1
Terrain Size	1000m * 1000 m
No. of Nodes	37
MAC Protocol	IEEE 802.15.4
Packet Reception Model	PHY 802.15.4
Radio Type	IEEE 802.15.4
Energy Model	Generic, Micaz, Micamotes
Routing Protocol	AODV
Antenna Model	Omnidirectional
Network Protocol	IPV4
Device Type	Sensor(RFD, FFD)
Traffic Type	ZIGBEE
Items to Send	100
Item Size	70 Bytes
Simulation time	500 secs
Link	Wireless
Channel Frequency	868MHz
Modulation	O-QPSK

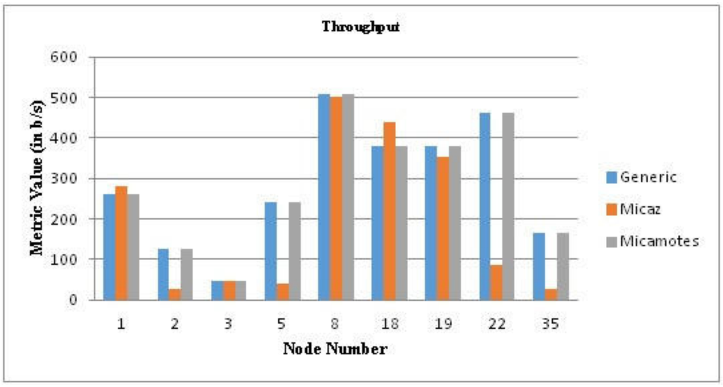


Fig. 5.1. *Throughput for different Energy Models*

maximum throughput is obtained for Generic and Micamotes energy models because of their reactive nature and least in case of Micazas it is hybrid in nature depending upon number of participating nodes. Therefore, on the basis of data packets received, Generic and Micaz gives better performance at the destination per unit time.

5.2. AEED. It is defined as average time required to send a data packet to destination. It is also known as data latency and is measured in seconds. Analysis between three models on the basis of data latency for the proposed WSN is shown in the Figure 5.2 given below. On analyzing the above plotted graph, it is found that the Micaz has smaller AEED because of its reactive nature compared to others which are proactive in nature. Therefore, the instant at which the first packet will receive is lesser in case of Micaz as compared to Generic and Micamotes.

5.3. Jitter. It is defined as the variation in the latency of packets at the receiver due to congestion, topology, route change etc. It is measured in seconds.It arises when the packets takes different amount of time,

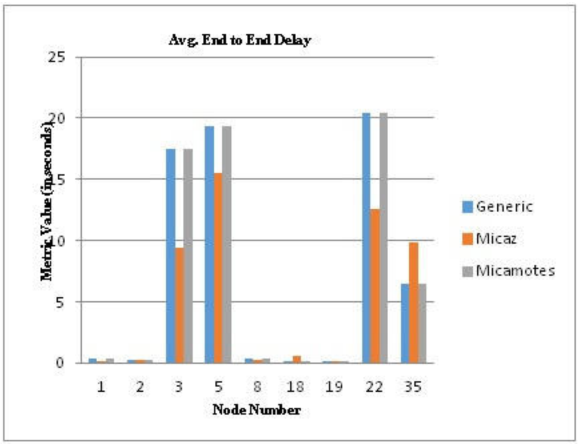


Fig. 5.2. *AEED for different Energy Models*

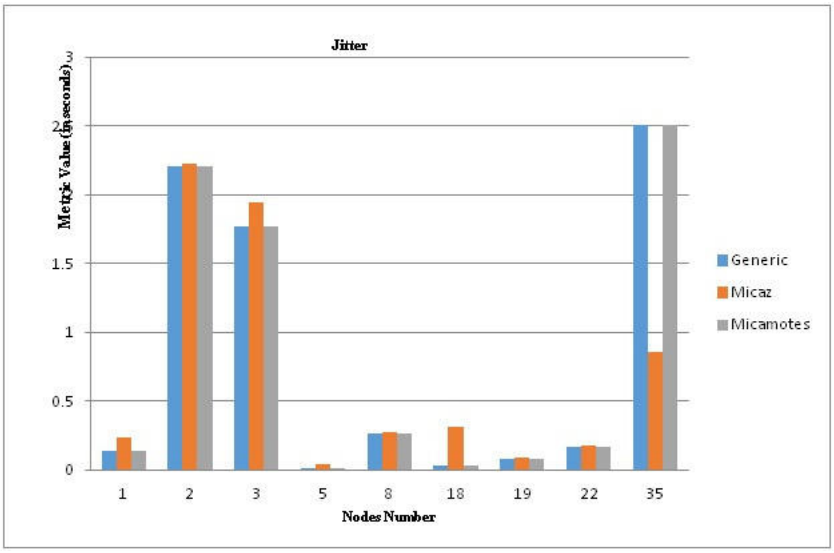


Fig. 5.3. *Jitter for different Energy Models*

reaching from source to destination in a transmission scenario. For the better performance, its value should be as low as possible. The analysis of jitter obtained in case of three energy models of the proposed WSN is depicted by the Figure 5.3. On observing the graph, it is noted that jitter is maximum for Micaz which in proactive in nature while smaller for the rest which are reactive in nature. Proactive models shows maximum jitter and reactive shows smaller jitter.

5.4. TPR. It determines the overall message received successfully at destination by any server. Efficiency of the network will be high, if the number of message received per unit time is more. Figure 5.4 shows the graph of total message received. It has been analysed from the figure that there are least number of packets received in case of Micaz which is proactive in nature while packets received are higher forrest of two which are reactive in nature. To enhance reliability of energy model, the network must receive higher number of packets. Hence, the reactive protocol provides good results.

5.5. Energy Consumption. The WSN parameters including lifetime, response time, scalability and effective sampling frequency etc. depends upon the power. To preserve the health of each node, energy is required

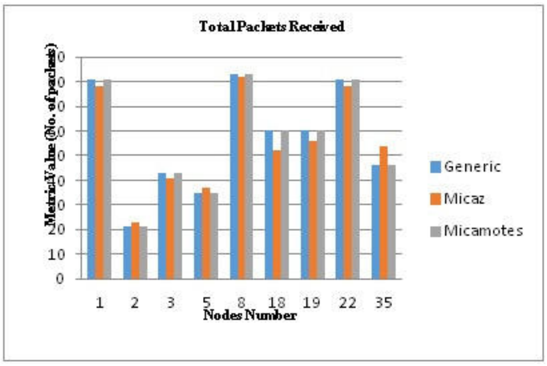


Fig. 5.4. *TPR for different Energy Models*

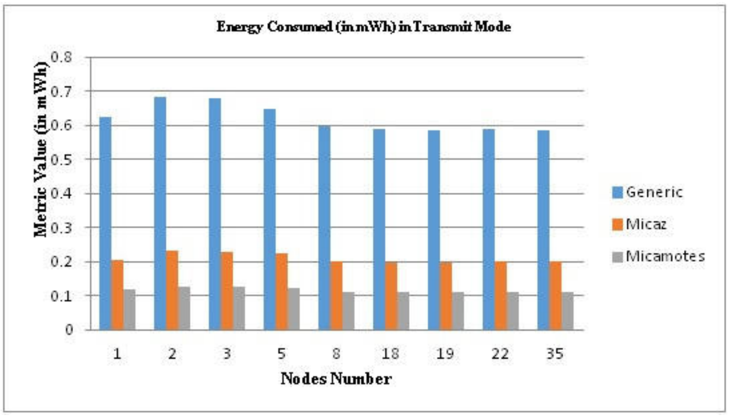


Fig. 5.5. *Energy Consumed for different Energy Models in Transmit Mode*

during transmission and reception of the data packets. The Figure 5.5 represents the energy consumption in transmit mode for various energy models i.e. Generic, Micaz and Micamotes. It has been observed from the graph that Generic model which is proactive in nature consumes more energy while the Micaz which is hybrid in nature operates in moderate energy and Micamotes which is reactive in nature requires lesser energy. Hence, Micamotes model is more energy efficient in comparison with Generic model. The Figure 5.6 shows energy consumed in Receive mode that is similar results as obtained for the energy consumed in case of transmit mode by routing protocol named AODV.

From analysis, it has been found that Generic Model which is proactive in nature consumes highest energy while Micamotes which is reactive in nature requires least energy. Therefore, Micamotes model prove to be more energy saving than Generic and Micaz models. The energy consumed by different models in idle mode is depicted in Figure 5.7. It has been observed that idle mode shows different results in comparison to transmit and receive modes.

In case of idle mode, the energy consumption is comparatively higher in Generic than other two modes. Therefore, Micamotes seems to be best energy saving model than other two models. It is also noted that for all the energy models, most of the energy consumption occurs in the idle mode.

After simulating, it has been observed that the energy consumption is highest in case of Generic model and lowest in case of Micamotes energy model. Therefore, Micamotes has been proved to be the best energy efficient model as compared to Generic and Micaz energy models in transmit, received and idle modes. The sensor networks can also be analyzed using other routing protocols for different design layouts and simulation parameters.

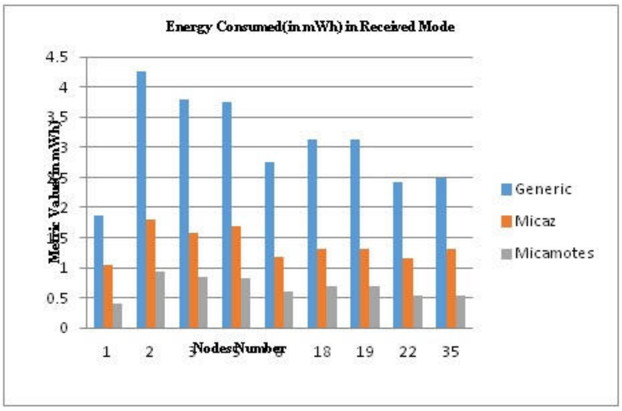


FIG. 5.6. *Energy Consumed for different Energy Models in received mode*

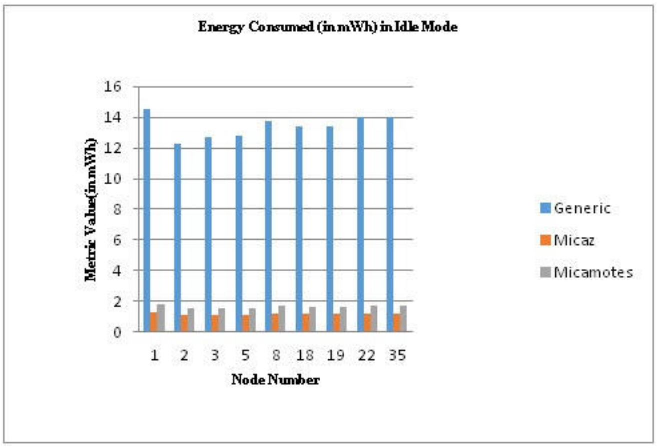


FIG. 5.7. *Energy Consumed for different Energy Models in idle mode*

6. Conclusion. This paper gives performance analysis of three energy models i.e. Generic, Micaz and Micamotes using AODV routing protocol on the QualNet simulator and their performance analysis is done on the basis of network metric such as Throughput, Jitter, AEED, TPR, and Energy Consumption in all the nodes. From simulation results it is concluded that Micaz has least throughput and maximum average jitter compare to Generic and Micamotes. AEED and TPR are least for Micaz and similar in Generic and Micamotes. For energy consumption, Micamotes consumes less energy by 80.46% and 428.57% in transmit mode, 102.94% and 335.6% in receive mode from Micaz and Generic energy models respectively. However, in idle mode Micaz energy model is found to be consuming less energy by 54.36% and 23.45% than Micamotes and Generic energy models respectively. Therefore, in over all analysis Micamotes seems to be the best energy saving model in all modes using AODV routing protocol.

REFERENCES

[1] C. Karthik, S. Kumar, R.S. Kumar, M. Nageswari, *Sensors Lifetime Enhancement Techniques in Wireless Sensor Networks - A Critical Review*International Journal of Computer Science and Information Technology Security, Vol. 3, April 2013.

[2] Y. Yu, B. Krishnamachari, V.K Prasanna, *Issues in Designing Middleware for Wireless Sensor Network*, IEEE Network, Vol.18, 2014, pp 15-21.

[3] Y. Chen, Q. Zhao , *On the Lifetime of Wireless Sensor Networks*, IEEE Communications Letters, Vol. 9, November 2005.

[4] A. Goyal, S. Vijay, D. Kumar, *Simulation and Performance Analysis of Routing Protocols in Wireless Sensor Network*

using QualNet, International Journal of Computer Applications, (ISSN 0975-8887) Vol. 52, August 2012, pp 47-50.

[5] N. Kamyabpou, D.B. Hoang, LU-*Modeling Overall Energy Consumption in Wireless Sensor Networks*,2007.

[6] S.K. Singh, R. Paulus, S.V. Rajput, T. Kaur, *Analysis of energy model and QoS in wireless sensor network under different modulation schemes*, International Conference for Convergence of Technology, (ISSN 4799-3759) Vol. 1, 2014.

[7] N. Radhey, V. Nandal, *Simulation and Analysis of AODV and DYMO Protocols under CBR in Wireless Sensor Network using QualNet*,International Journal of Computer Sciences and Engineering, (ISSN 2347-2693) Vol. 6, May 2018, pp 100-104.

[8] B. Prakash,, S. Jayashri, T.S. Karthik, *A hybrid genetic artificial neural network (G-ANN) algorithm for optimization of energy component in a wireless mesh network toward green computing*, Soft Computing, (ISSN01234567).January 2019. doi: 10.1007/s00500-019-03789-8.

[9] H. K. Sampada, K. R. Shobha, *Performance Analysis of Energy-Efficient MANETs-Using Modified AODV (M-AODV)*, ICCNCT 2019, doi: 10.1007/978-981-10-8681-6_9, pp 75-86

[10] M, Rajgor, P. Shete, R.N. Awale, *Probabilistic Energy Efficient Routing ProtocoL*, International Conference on Communication", Information and Computing Technology. Vol. 1, 2018.

[11] Muhammad A.K. Pandey, P.N. Gupta, H. Vardhan, *Performance Evaluation of Various Routing Protocols and quality of service for Wireless Sensor Network*, Journal of Telecommunication Study, vol .4

[12] S. Yadav, R.S Yadav, *A review on energy efficient protocols in wireless sensor networks*, Wireless Network, 2018, DOI :10.1007/s11276-015-1025-x.

[13] K. Mor, S. Kumar, D. Sharma, *Ad-Hoc Wireless Sensor Network Based on IEEE 802.15.4: Theoretical Review*, International Journal of Computer Sciences and Engineering" (ISSN:2347-2693) Vol. 6, 2018, DOI: https://doi.org/10.26438/ijcse/v6i3.220225 pp.220-225

[14] K. Mor, S. Kumar, *Evaluation of QoS Metrics in Ad-Hoc Wireless Sensor Networks using Zigbee*, International Journal of Computer Sciences and Engineering", (ISSN 2347-2693) Vol. 6, March 2018, DOI: 10.26438/ijcse/v6i3.9296 pp.92-96.

[15] R. Monir, R. Thalore, P. P. Bhattacharyana, *Performance Comparison and Analysis of On-Demand, Table-driven and Hybrid Routing Protocols in 3D*, Mody University International Journal of Computing and Engineering Research, (ISSN 2456-9607) Vol.2, 2018, pp 20-25.

[16] K.A. Darabkh, M.S.E Judeh, *An Improved Reactive Routing Protocol over Mobile Adhoc Networks*,2018, pp 707-711.

[17] N. Kaur, R. Singhai, *Minimizing Congestion in Mobile Ad hoc Network Using Adaptive Control Packet Frequency and Data Rate*, Proceedings of ICCASP, 2018, DOI:.org/10.1007/978-981-13-1513-8_30 pp 285-294.

[18] S. Khurana, S. Kumar, D. Sharma, *Performance Evaluation of Congestion Control in MANETs using AODV, DSR and ZRP Protocols*, International Journals of Advanced Research in Computer Science and Software Engineering, (ISSN: 2277-128X) Vol. 7, June 2017, pp 398-403.

[19] R. Munjal, S. Kumar, *Maximizing Lifetime of Cluster-based WSN through Energy-Efficient Clustering Method*, International Journal of Mobile and Ad-hoc Network, Vol. 2, August 2011, pp 239-243.

Edited by: Dharm Singh Jat
Received: May 22, 2019
Accepted: Nov 14, 2019

DOI 10.12694/scpe.v20i4.1579
Scalable Computing: Practice and Experience ISSN 1895-1767
Volume 20, Number 4, pp. 641–651. http://www.scpe.org © 2019 SCPE

ENERGY EFFICIENT ROUTING PROTOCOL FOR MOBILE SOCIAL SENSING NETWORKS

SALEM SATI, AHMED SOHOUB, AND TAREG ABULIFA[*]

Abstract. Mobile Social Sensing Network (MSSN) is a subclass of Wireless Sensor Networks (WSN). This MSSN is consists of mobile sensing transducers carried by people. Sensing information gathered by mobile sensors will transmit to the data sink. This data sink may it is fixed or mobile. But in optimal cases, it should have efficient energy and position compared with other mobile sensors. On the other hand, mobile sensors may have a social tie because they carried by people. Traditional MANET routing protocols such as AODV and DSR are inapplicable or perform poorly for mobile social data sensing. Especially for distributed mobile social sensing. Insufficient performance due to the nature of the mobile sensors which suffering from a limited energy source. In recent days, there are many routing protocols proposed by researchers. These protocols improve the total delivered messages in mobile social sensing networks, but most of them do not take into account the link bandwidth and node storage limitation, thus routing may lead to more energy consumption among mobile sensing nodes. In this paper, we design an Energy-Efficient Routing Protocol (EERP) for mobile social sensing networks. We consider the node energy as a balance function between the delay of collected data and transmission of sensor nodes to the data sink. Furthermore, we also develop an enhanced version of the suggested EERP which named EERP+S, EERP+S combines the energy percentage and social metric of node degree. EERP and its updated version EERP+S are dynamically adjusting the control function based on data delay and transmission in addition to node activity. Simulation results demonstrate the efficiency of EERP and EERP+S compared with the flooding behavior of an Epidemic. Epidemic and its social version Ep-Soc are compared with suggested protocols in distributed mobile social sensing paradigms.

Key words: Energy Efficiency, Mobile Sensing, Social Sensing, Node Energy, Routing Performance, Forwarding Decision

AMS subject classifications. 68M12

1. Introduction.

Mobile Social Sensor Network (MSSN) is a future network with a higher increase in social sensing application scenarios. Due to recent advanced technology in smartphones carried by people, it is easily feasible for the mobile and fixed sensor not only to collect industrial environment sensed data, but also to gather data in social and environment of real-time applications.

There are a lot of mobile sensing applications with different sensing functions. Where the nodes maybe humans, cars or animals which may be connected to the internet. Different sensing mobile nodes collecting data for applications that form a new sensing topology paradigm. Socially scenarios based on smartphones and other sensing scenarios based on car or animal tracking are applications of mobile sensing [1, 2, 3]. Those scenarios are collecting and processing sensing data from different mobile sensors nodes. The data collecting is one of the challenges on the performance of mobile sensing. This challenge depends on how to efficiently gathering the sensing data with the limitation of bandwidth and storage. One possible solution is to deploy mobile devices to transmit sensing data rather than deploy a traditional fixed data sink.

Mobile sensors nodes are energy-constrained devices. Therefore, MSSN researchers from different parts of the WSN group are trying to minimize energy consumption. This energy prevention is for increasing the nodes' lifetime on the network. In real applications of MSSN, nodes lifetime should be longer as it is possible. This mobile sensor activity of sensing and transmission must be without communication interruption. Specially interruption between mobile sensors and head cluster (data sink).

There are MSSN applications that should be reported to the data sink on time. This is because the application sensing real-time which sent within a shorter time. Besides, the system monitoring, it may need a control response by an increase or decrease some parameters such as temperature or pressure. On the other hand, from energy concepts the delay and node degree are related to sensing and transmission to the data sink. There are few pieces of research which are focusing on real-time communication in MSSN.

Some of the researchers analyzed communication technologies. This is because wireless communication states consume high energy of sensor node when compared with stored sensing data. Additionally the receiver and transmit states of communications consume most of the energy compared with CPU processing.

[*] Misurata University, Libya (salem.sati@it.misuratau.edu.ly)

To minimize or prevent energy consumption, many energy metrics were considered in the literature researches. However, most of these researches ignore the delay and social aspect of the sensor networks. The simplest routing algorithm for MSSN is Epidemic [4], this protocol based on whenever a sensor node storing a sensing message encounters with another sensor node, it transmits a replica of the sensed message. It forwards the message to the encountered node. This occurs when both nodes in coverage range. Moreover, there is a buffer space on the nodes which has no copy. However, such flooding-based and blind replica also causes relatively high energy consumption. To overcome the problems of Epidemic routing, many routing protocols limit the number of replicas, such as [5, 8]. Generally, the delivery ratio of flooding-based strategies is relevant high, but the heavy load of nodes may cause serious congestions or energy issues. To investigate how to control the forwarding list and how to order it there is some publications analysis this issue [7].

In our research paper, while the contribution of the paper presents a new opportunistic routing protocol based on Epidemic behavior. This new routing protocol can optimize energy consumption and delay in MSSN. This paper finds out a controlled threshold that can consider the requirements of social sensing, i.e. delay, energy and node degree as a control function for message replication. The following sections of this paper are organized as follows. Section 2, demonstrates the previous researches as related work. While section 3 provides the proposed routing protocol EERP and its model analysis. Section 4, the paper presents the updated version of EERP which named EERP+S. Simulation sets and performance metrics have been given in section 5. Simulations results and analysis are presented in section 6. Finally, section 7 gives the conclusion and future work.

2. Related Work. There are many opportunistic routing that can be deployed in WSNs. One example of these routing is (OWR) suggested by [9]. This routing considered a practical scheme for WSNs. The idea of this protocol is inspired by the original ExOR [10] routing. ExOR designed for mesh networks. OWR protocol is implemented for a duty-cycled configuration. Moreover, data units are marked to sets of specific receivers. These receivers are forwarded by the next-hop that wakes up and successfully accepts the message. This behavior reduces the delay due to the source spent time for a specific node to wake up. Also, it reduces energy consumption for path transmission in the network. The energy reduction due to deploying all neighbors as critical forwarders.

Several routing protocols for WSN have been proposed. For example the paper [11] suggests an energy cost. This cost is optimally limited and increase the nodes lifetime. Also, the paper [12] has proposed a couple of energy concentrate data-forwarding rules for single path and multiple paths. The paper suggests a mechanism to minimize energy consumption via this cost. Moreover, this paper finds a trade-off between node energy and message delivery ratio. The paper [13] which takes into account the nodes residual energy as forwarding criteria. However, in all of the mentioned papers, the consideration of latency is missing and there are more wastage of message dissemination in wireless sensor networks. The main aim of opportunistic routing costs is the concept of minimizing the number of message copies to keep energy. Furthermore, applying the advantages of flooding is the nature of wireless networks. One advantage of flooding to send a message through multi-path in the network as possible. ETX was a first metric suggested for opportunistic routing especially in wireless networks. Researchers have implemented new routing costs and weights such as OEC (Opportunistic End-to-end Cost)[14], Also there is another research concerning Opportunistic Expected One hope Throughput (EOT)[15]. These two papers [14, 15] show the trade-off between the advancement of messages and the message forwarding time. Furthermore, the authors of the paper [16]suggest a Distance-based Energy Aware Routing (DEAR). DEAR ensures energy saving and balancing based on different energy and traffic analyzing models. Moreover, the paper [17] is address the power efficiency issue by suggesting a Real-Time routing with Controlled Dissemination (RTCD) of the message to mobile sink. RTCD consists of two stages which are flooding and routing stages. These two stages considered as data collection done by the mobile sink. This mobile sink reduces the energy consumption of the nodes. Controlling message dissemination is accomplished by limiting the topology diameter. In addition to setting a triggered value for the remaining energy of the sensors. Simulation results show that the delivery ratio for RTCD routing is better than other protocols. Finally, in this paper [18], authors suggest opportunistic routing protocol for sensor networks, This protocol based on the selection of the best forwarder node. This selection of neighbor node is the main factor that enhances the power prevention and nodes lifetime. To improve routing performance more efficiently, authors present a sleep algorithm named, PSS algorithm for the sensor nodes combined with opportunistic routing. Simulation results show that proposed routing has optimal energy consumption with less overhead.

3. Energy Efficient Routing Protocol (EERP). As the MSSN runs continuously such as real-time monitoring scenarios, the sensors will eventually die due to unnecessary message replication which leads to energy wastage. The problems raised by the energy limitation of mobile social sensors. There are mainly two aspects: (1) The message transmission which is the main factor of energy consumption; (2) The sensing data stored in the nodes which may be delayed as real-time application. The first point will impact immediately on sensors' energy consumption of the network, it directly degrades the routing performance from the reliability concept. The second point will lead to poorly end-to-end delay of the real-time application in MSSN, this will additionally effect on the energy of the sensors where stored message also will consume a piece of power. From the perspective of energy saving and real-time decision, this paper presents an Energy-Efficient Routing Protocol(EERP) for MSSNs. The basic idea as follow: To reduce the complete energy loss of sensor node we propose a new scheduling mechanism. This scheduling strategy derives the optimal number of message replication according to variables. These variables are the transmission and the number of stored sensing data. Moreover, both variables impact on total energy consumption.

The number of messages generated by traditional mobile social routing tends to be large, resulting in an insufficient energy consumption (i.e., the energy consumption of flooding routing in social sensing topology is often much larger than that in fixed WSN.). To overcome this challenge, based on the opportunistic scheduling, the message life is considered as the main factor in the energy. The messages are ordered in the scheduling queue of the node's buffer. Then in the forwarding decision phase, we implement the energy ratio of the message balance. Moreover, the energy ratio depends on the number of transmitted and stored copies. This forwarding strategy of EERP improves the ROUTING performance of distributed and cluster topologies. In terms of optimal message replication, we take the life of the message into account as a part of the residual energy. This is basically different from traditional social sensing protocols which consider only the social tie. The main advantages points of proposed routing protocol include the following:

- The optimal threshold of replication is derived to minimize the sensors' energy consumption of MSSN.
- Message transmission and sensing are both considered as the energy balance parameters.
- The message life energy ratio is necessary to compute the energy consumption instead of a social tie only in MSSN.
- A new forwarding priority function can ensure the message with lower energy loss.
- Sensor node residual energy can be saved when deploying the proposed forwarding function.
- The new routing achieves better routing performance by reduction of storage, energy, and delay.

The aim of EERP is to minimize the energy consumption of the network by limiting the number of message. Recall that in traditional social networks the message will not be replicated to the met node always, but message replicated when the node which holds the message has lower social weight. Here, Social measuring could be any existing social indicators, such as node degree or centrality.

EERP is a routing protocol to reduce the traffic load of each node. It can be deployed to any social-based distributed or clustered topology. EERP makes its decision using energy and sensing factors. Traditional social routing uses social metrics per node for its ordering and forwarding decisions. Where to compute the social weight of a node, a social graph is required to express the social ties among all nodes. Commonly such a social graph is built from recorded contacts of the node with other nodes in the network. Assume that N is the set of nodes in the social network. Each sensor node can transmit and receive messages when it connected with another node. The message replicated when other node has higher weight and has no copy.

The EERP uses a threshold on queue order to satisfy whether there is a chance between two messages to be replicated. If the number of transmission and stored times between the two messages is equal. Then there is an energy balance between these two messages based on their life TTL.

In traditional social routing protocols, the messages are replicated to the contacted nodes which have a higher social weight. This may grantee to achieve best delivery ratios, but nodes that have large social wight may die soon due to their huge traffic load. Therefore, we consider other parameters in proposed EERP compared with traditional social-based sensing routing in MSSN.

EERP decision is based on the energy metric of message and node. These energy metrics are transmission and storing times of the message copy. Here, energy ratio $0 \leq E_{\%} \leq 1$. Thus, it becomes more difficult for the current message to transfer. This is because the other message needs to have energy ratio less than that of the message to be forwarded. By applying this rule, the number of message redundancy in the network will be minimized. Naturally, the delivery of the new replication decision decreases, thus we dynamically modified the energy function based on the Time To Live (TTL). The dynamic behavior of the decision is to minimize the energy consumption. TTL of the message shows whether the message is expired and when the message should be evicted. At the message originator, the TTL of a message is configured to an initial value TTL_0. After each transmission or forwarding decision, the value of TTL will be copied to the new copy. When TTL is reached to zero, the message will be expired and evicted from the node's buffer.

EERP forwarding decision is made only when the energy function of the message is less than that of the message to be forwarded. The basic concept of the dynamical energy function ratio is as follows. In the beginning, when the message created TTL is set to maximum, EERP puts minimizing the energy ratio of the message as its first priority, thus the value of energy ratio subtracted from 1. However, after several transmission and storage times, when TTL goes to a small value, which leads to that message will be evicted soon, EERP modifies the energy ratio by considering the transmission and sensing times as its first priority. Therefore, we set the energy ratio as:

$$F_d = MS_{(S_e,T_r)} \quad . \quad E_{\%} \tag{3.1}$$

where f_d is a forwarding decision function which consists of two parts used to determine the threshold of message replication, The first part $MS_{(S_e,T_r)}$ is related to mobile sensing where data sink and sensors considered as mobile devices. The second part $E_{\%}$ related to the energy ratio of the message.The following equation shows the factors which impact on the forwarding decision as follow:

$$F_d = (1 + T_m + S_m) \quad . \quad (1 - \frac{TTL_r}{TTL_0}) \tag{3.2}$$

The mobile sensing function considers the two main function of the sensing. These functions are transmission T_m and sensing S_m. On the other hand, the message energy ratio based on TTL_0 and TTL_r are the initial message TTL value and the current TTL value of the message, respectively. Note that EERP considered as mobile sensing energy aware routing protocol when deploying the forwarding function. Algorithm 1 shows the detailed description of the proposed EERP.

Algorithm 1 *EERP* Message Forwarding Decision

```
procedure READ ([$T_m, S_m, TTL_0, TTL_r$] ← $Message$)
    Sorting                                          ▷ EERP sorting based on $F_d$
    while Encountered node has no copies of $m_1$ and $m_2$ do
        calculate $F_d$ for $m_1$ and $m_2$
        if $(m_1, F_d) > (m_2, F_d)$ then
            $Message \leftarrow (m_2, F_d)$
        else
            $Message \leftarrow (m_1, F_d)$
    return $Message$
    $Forward \leftarrow Message$
```

4. Enhanced Energy Efficient Routing Protocol (EERP+S). We improve the proposed EERP by considering the social metric. This social metric related to the energy of the node instead of the message only as in EERP. Where the message replicated to a specific group of connected neighbors. The traditional replication is based on replication to any number of neighbors. The new strategy reduces the energy consumption of the node. Energy-saving achieved by replicating the message to multiple nodes with single message energy T_m. We adopted the forwarding decision of EERP+S by taking into account the social metric. EERP+S combines the forwarding function of EERP with the energy consumption of the node. This node energy saving is based on a

Table 5.1
Simulation Settings

No	Settings	Value(s)
1	Simulation Region	4500 in 3500 m^2
2	Simulation Duration	720 min
3	Number of nodes	125
4	Sensors Type with speed	80 Smart-phones Sensors (0.5-1.3 km/h) 40 Vehicular Sensors (10-75 km/h) 5 Mobile Data Sink (15-60 km/h)
5	Routing protocols	*EERP*,*EERP+S*, Epidemic and Ep-Soc
6	NIC	One-to-All
7	Radio Range	0.25 km
8	Link Speed	0.25 MBps
9	Social Metric	Node Degree
10	Sensing Data Generated	500-1024 KB
11	Sensing interval	25-35 s
12	Sensing Data (TTL)	120, 240, 360, 480, 600 min
13	Sensor Memory Space	Smart-phones: 5 MB Mobile Data Sink: 50 MB

social metric which is node degree N_d. Where it might be better to consider message and node energy. As a social metric, it is more efficient to find a node that its activity is larger than other nodes. This phenomenon could have the best impact on energy consumption. To improve the forwarding decision of EEPR+S, we include an additional node energy metric (N_d). This metric is used to further dynamically adjust energy consumption based on EERP. When a node encounters more than a specific number N_d of nodes that have no copy of the message, it forwards the message to all N_d or greater number of nodes. When the number of nodes is less than N_d, it does not forward the message (since the activity of the met nodes is not better than the current node). Our proposed enhanced protocol EERP+S slowly relaxes the forwarding decision by combining the message and node energy consumption indicators. In this improved version EERP+S, the forwarding function is dynamically adjusted by both F_d of the message and the number of met nodes (i.e., node degree). These nodes buffers have no copy of the message. Here, N_d is threshold calculated depending on the node degree. This node degree is related to the radio range of the sensor node interface as shown in the following equation:

$$(EERP + S) = \begin{cases} EERP & (F_d)_{message} < (F_d)_{others} \\ N_d & N_d \geq \sqrt{r} \end{cases} \tag{4.1}$$

5. Simulation Scenarios. This section conducts simulation experiments over social scenarios. The comparison is evaluated two proposed EERP and EERP+S with Epidemic and its social version protocols. The evaluation with other protocols conducted using ONE simulator[6]. The evaluation uses the EERP based on F_d as a routing protocol for MSSN. Additionally, it deploys the enhanced protocol EERP+S by using the node degree as the social metric. Obviously, in EERP+S routing, the message has more chance of forwarding to the met node. The forwarding decision satisfied when two conditions are available. First. the node has a high social metric. Second, the message has low energy consumption.

The EERP is an energy-weighted protocol. We compare EERP and EERP+S with the traditional following existing routing protocols.

1. Epidemic: When the node connected with any encountered neighbor nodes, the node which holds the message copy it to any number of neighbors nodes.
2. Ep-Soc: The message is only forwarded from the node which holds the message to the connected nodes. The forwarding decision is true when the number of neighbors is equal to or greater than N_d. The value of N_d calculated based on $N_d = \sqrt{Radio\ Range}$.

Our simulation scenario is a set of parameters for sensing environment such as pollution. This use case simulated by the ONE simulator. These settings are used for different routing protocols. This paper evaluates only the sensing and transmission of data.

The message TTL and sensing interval are changed for simulate MSSN scenario. The evaluation is to address various kinds of social sensing use cases. Table 5.1 shows the basic settings for MSSN scenarios, they are derived from the social settings from the ONE simulator. The simulations of different routing protocols run for 720 minutes. It simulates 125 sensor nodes.

Nodes generally divided into groups. The nodes have a buffer space limited by the number of messages. Sensor nodes can only move on different routes, which simulate streets and buildings in the smart city. There are two types of sensors, which are carried by humans as smart-phones or fixed on cars. In addition. We simulate 5 mobile data sinks, these data sinks have a large memory space compared with other sensors nodes. In all our simulation experiments, we evaluate the performance of different routing protocols using the following metrics.

1. Data Delivery Ratio: the average ratio of the successfully delivered sensed data from the sensors to the mobile data sinks.
2. Overhead Percentage: the number of relayed sensed data to the total successfully delivered data to mobile data sink.
3. Average Delay: the average time spent to successfully delivered sensed data from the sensors to the mobile data sink.
4. Relayed Messages: the average number of forwarding times needed to deliver a single successfully message.
5. Average Hop Counts: the average number of relayed nodes during each successful path.
6. Average Sensing Time: the average time spent in sensor memory during the delivery of the message.

6. Numerical Results. The evaluation of proposed EERP and EERP+S with traditional Epidemic and Ep-Soc routing as described in Table 5.1. Through our simulation social scenarios, we modify sensing data TTL from 120 to 600 minutes with a step of 120 minutes. This is for simulating different sensed data. Also, we pay attention to energy and social balancing for EERP forwarding decisions. In addition, we apply social metrics as additional criteria in the EERP+S routing protocol. We apply node degree N_d as a threshold which computed based on radio range (r) and node activity. Both conditions of EERP and EERP+S depends on Eq. (3.2). For the EERP protocol scenario, we select the forwarding function of the message based on the energy threshold of $E_{\%}$. Also, we select the number of neighbors as the second tie for a proposed energy-efficient social sensing EERP+S.

6.1. Data Delivery Ratio. The data delivery ratio of EERP is compared with EERP+S which use scheduling based on N_d in addition to F_d. We consider FIFO evict policy as default for all different routing protocols. Figure 6.1 shows the data delivery ratio of different protocols. The figure shows that EERP+S is better than Ep-Soc. This clear when deploying forwarding threshold based on social metric as node degree. The enhancement in data delivery ratio of EERP+S reaches 15% when compare with EERP by applying F_d decision. Also it is better more than 3% when compare EERP+S with Ep-Soc at data sensed of TTL= 600 min. The EERP+S enhancement due to combining of energy model of EERP with social function of node degree N_d. The social metric doesn't considered by EERP. EERP forwarding function has better performance when compared with traditional Epidemic. Obviously, the performance of EERP+S is the best because it minimizes the energy consumption of the data sensing. EERP when applies forwarding function F_d considered as better than traditional Epidemic ($\approx 1 - 18\%$).

6.2. Overhead Percentage. The overhead percentage metric is considered as main energy consumption indication. Where it shows how much energy consumed for delivering the sensed data in distributed MSSN. Also the overhead percentage can be calculating by N_d and F_d, where EERP+S eliminates more of overhead of EERP as shown in Figure 6.2. Clearly, traditional Epidemic and Ep-Soc are suffering from consuming of the total energy by applying unlimited and social forwarding decision respectively. Figure 6.2 shows that EERP+S has lower overhead when combining F_d and N_d by deploying Eq. (4.1). EERP+S protocol minimizes the number of relayed messages percentage to ($\leq 10\%$). This is because the replication of social sensed data implemented based on energy percentage and node social metric. Furthermore, from figure 6.2, we observe that Ep-Soc

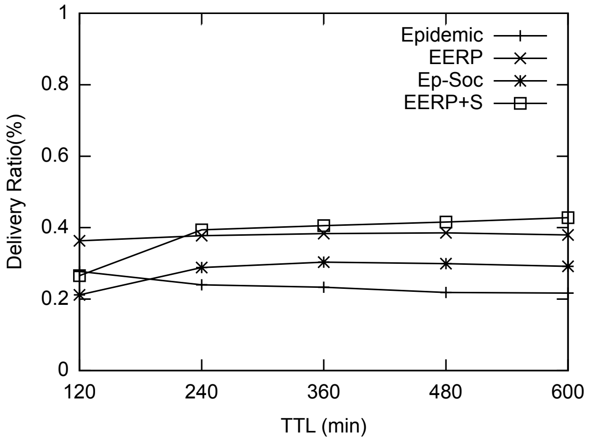


Fig. 6.1. *Data Delivery Ratio*

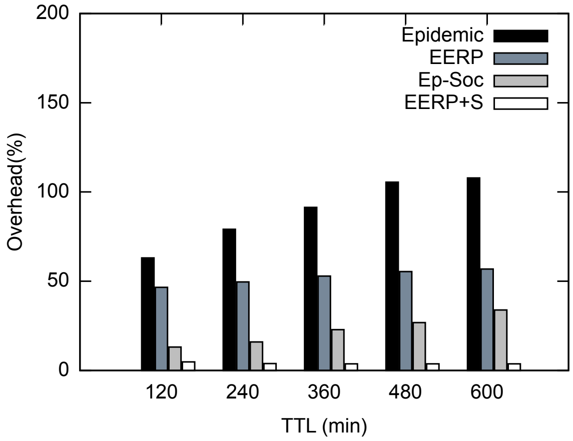


Fig. 6.2. *Overhead Percentage*

has a regular overhead percentage $\leq 55\%$ and EERP has $\leq 60\%$ compared with traditional flooding Epidemic (60-110%). The optimal threshold of the applied forwarding decision for EERP+S and EERP depends on node and message energy consumption, respectively. Where the main concept of forwarding decision of both EERP and EERP+S based on balanced between energy and message delivery.

Obviously, as we varying the TTL of sensed data for varying traffic load, in figure 6.2 the overhead percentage increases clearly when applying traditional Epidemic and quite when deploying Ep-Soc. This is because the social metric of node degree increase the delivered data through a high activity mobile nodes. Furthermore, figure 6.2 shows that EERP+S which deploy energy and social formulation still has lower overhead percentage ($\leq 10\%$). Furthermore, EERP+S has minimum overhead when compared with Ep-Soc and EERP.

6.3. Delay. The value of delay considered as application quality indication and main factor for quality of services (QoS). Commonly, delay is used as performance indication specially for real-time applications. Figure 6.3 shows that the lowest delay for different four protocols can be achieved when applying EERP, where EERP consider the optimal forwarding decision by F_d as function of message energy. Basically, we found that the number of buffered times of sensed data is impact directly on total end to end delay. As one contribution, we get from delay metric results that the social metric is directly impact on delay of sensed data specially in mobile social sensing networks. The forwarding decision formulation which used by EERP will impact on the average sensed data delay. As second advantage, it also minimize the energy when considering radio range as degree variable of EERP+S. Therefore, EERP+S considers the social and energy as criteria of forwarding

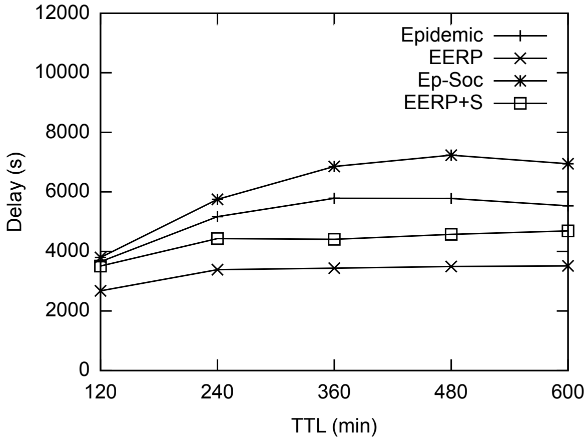


Fig. 6.3. *Delay*

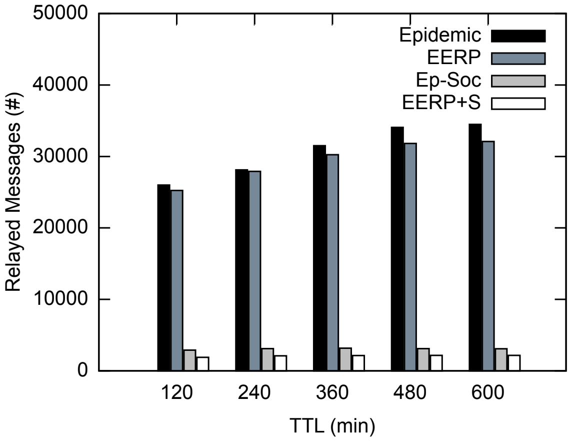


Fig. 6.4. *Relayed Messages*

functions. Moreover, we observe that when we apply different forwarding based on each routing deployed as shown in Figure 6.3, EERP has minimum delay ($\approx 3000s$ at TTL= 480 min) when it compared with EERP+S and Epidemic ($\approx 4500s$ and $\approx 5800s$,respectively). Additionally. social epidemic Ep-Soc has a higher delay ($\approx 7000s$). Furthermore, Figure 6.3 demonstrates that EERP has minimum data sensed delay when it deploy its forwarding decision by applying Eq. (3.2). This can be observed specially when compare EERP with EERP+S by deploying Eq. (4.1) as forwarding decision which has social metric as second tie.

6.4. Relayed Message. The energy consumption in MSSN is very criteria. Mobile sensors have limited energy. Therefore, it is better to design routing protocol which decrease the number of relayed sensed data. As the number of relayed messages is decreased as the energy efficacy is increased, but at the same time, routing protocol should keep the desired data delivery percentage or it is better to increase it. The main concept of EERP and EERP+S are optimize the number of sensed data with keeping of desired delivery percentage. Relayed sensed data as energy indication is demonstrated by figure 6.4. This figure shows relayed messages of EERP and EERP+S in comparison with Ep-Soc and traditional Epidemic.

In addition, the number of relayed messages sensed is calculated for every routing protocol. Proposed protocols are apply different forwarding decision of EERP and EERP+S. Basically, EERP aims to reduce energy consuming of sensors nodes by calculating energy percentage as TTL function.

Figure 6.4 shows the results of energy and social consideration of EERP and EERP+S, respectively. Figure 6.4 shows that Epidemic relays ≈ 33000 more data than EERP which exchanges ≈ 31000 messages at TTL

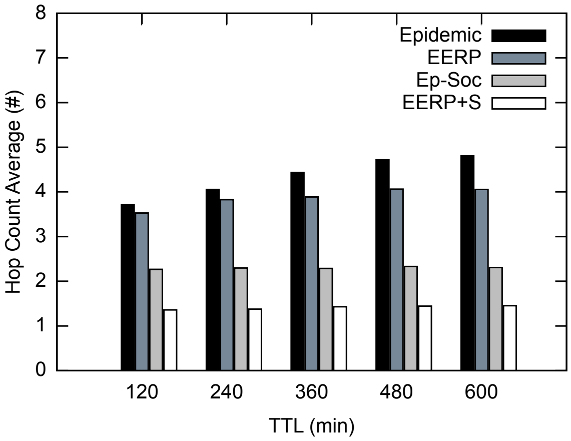


FIG. 6.5. *Hop Count Average*

= 480 min. Also Ep-Soc relays ≈ 2200 and EERP+S relays only ≈ 1800 sensed data at TTL = 480 min. We observe from figure 6.4 that the two social protocols EERP+S and Ep-Soc have minimum relayed messages. Furthermore, they are lead to more delay compared with other protocols as figure 6.3 demonstrated. EERP when deploys the function of F_d it relays ≈ 23000 sensed data at TTL=120 min and ≈ 31000 sensed data at TTL= 600 min. This due to that EERP reduces the number of sensed data to ≈ 20%. EERP when compared with traditional Epidemic at TTL= 600 Epidemic relays ≈ 33000 sensed data. Epidemic is the highest values for relayed messages in figure 6.4. This is due to it doesn't consider any energy or social metrics. Figure 6.4 demonstrates that EERP+S eliminates ≈ 5% of sensed data when compared with Ep-Soc at different TTL values. This due to the fact that EERP+S is also consider the EERP forwarding decision instead of social metric only. This advantage of energy percentage is to enhance the EERP+S and EERP compared with Ep-Soc and Epidemic respectively.

6.5. Hop Count Average. Hop count is one of energy and social metric. Concerning energy it indicates that the number of times the energy consumed during transmission and receiving, in addition it shows the sensing time of sensed data. On the other hand hop count as social metric. It shows the social tie of the nodes in the path with message destination. As the number of hops increases as the social metric is decreases between the source and destination. Basically, when the hop count decreases then the energy consumption decreases and social tie increases. For our evaluation experiments, we consider the hop count average to compare EERP and EERP+S with other competitive protocols. It is important to take into account the hop count as metric for social and energy costs.This helps on adapting between sensing time and energy consumption of the data during the same path. When the hop count is averaged for all successfully received sensed data. It shows the average energy consumption between sensor node and data sink.

Figure 6.5 shows the results of different applied routing protocols for comparison based on average hop count. Figure 6.5 demonstrates that EERP+S has lowest average hop count < 2. Ep-Soc has average hop count > 2 which means that EERP+S perform better than Ep-Soc. When apply the forwarding decision of EERP, it has average hop count ≤ 4 when compared with traditional Epidemic which has average of (5-6). As the hop count increases as the message consumes more energy, In addition, the node consumes more resources such as bandwidth and memory. On the other hand, as the hop count average decreases it ensures that both congestion and energy will decrease. We observe from figure 6.5 that when deploying EERP+S the hop count for all paths will reduced to ≈ 40%. This percentage of EERP+S hop count reduction is when compared with traditional Epidemic.

6.6. Sensing Time. The sensing time is the time spent on the sensor nodes buffers. This time is spent by sensed data till reach the mobile data sink. For detailed analyze of sensed data, simulation scenarios consider the sensing time as indication of real-time delay. We apply F_d in both proposed protocols for minimize the energy consumption of message and the node. We analyze EERP and EERP+S which both consider the energy

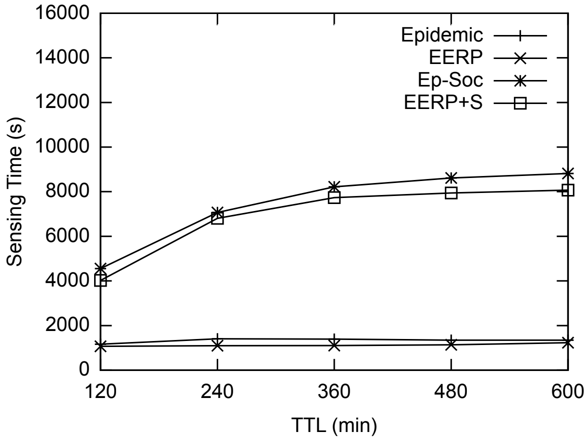


Fig. 6.6. *Sensing Time*

percentage by deploying message TTL. Figure 6.6 shows the that traditional Epidemic reaches sensing time higher than EERP, where EERP+S reaches minimum sensed data time when compared with Ep-Soc. EERP+S sensing time reaches higher values when compared with EERP. This insufficient sensing time of EERP+S caused by social tie of EERP+S.

EERP+S has a routing metric which based on the energy percentage and social tie of node degree. Figure 6.6 shows that as TTL increases as the sensing time increases. Sensing time is increased due to as TTL increases the congestion will normally increases.

The sensed time, i.e. buffering delay, is one of the component of average delay in addition to the transmission delay. This time has great impact on the energy consumption of the node.

SSRP+S increase sensing time because it minimize the number of paths for the sensed data. But EERP increases the number of sensed data paths which minimize this time. On the other hand, multi path has a disadvantage from the view of the energy, where multi path increases the overhead percentage.

Essentially, the routing protocol's forwarding decision may considered one of factors which leads to increase the energy consumption. On the other hand, limiting or reducing the number of sensed data copies minimizes the energy consumption. In addition it increase congestion which leads to increasing the average sensed time. Therefore, we compare EERP with traditional Epidemic by taking into account the forwarding decision based on energy percentage. As shown in figure 6.6, we apply different forwarding function depending on every routing protocol. EERP has a minimum sensed time ($\approx 1100s$ at TTL= 360 min) when it compared with traditional Epidemic which has sensing time of ($\approx 1700s$). In addition to social routers EERP+S and Ep-Soc which have sensing time of ($\approx 7800s$ and $\approx 8100s$, respectively) at TTL= 360 min.

7. Conclusion. Mobile social sensing routing is very important for sensor networks. The nodes in MSSN are suffering from limited resources such as energy. This paper considers forwarding straggly for MSSN routing. The paper investigates the flooding or blind forwarding strategies, these forwarding strategies are commonly used by an Epidemic with or without social metric (Ep-Soc).These protocols are target to achieve a high performance without considering the delay and energy in MSSN. We inspired by the two Epidemic routing versions (Ep-Soc & Epidemic) two suggested routing protocols (EERP+S & EERP). The two suggested protocols are with and without social metric of node degree. In addition, we consider the criteria of energy as main factor of our design for both protocols. Our simulation evaluation was in terms of data delivery ratio, delay and overhead percentage. Based on energy design, we implement social version EERP+S by deploying the social metric of node degree. We formulate the forwarding decision of EERP based on energy percentage, where enhanced version EERP+S considered as energy-based and social-based routing protocol. Simulation results using ONE simulator are analyzed to evaluate proposed protocols. Our proposed protocols improves the data delivery ratio. Moreover, both protocols are decrease the overhead and relayed messages. Additionally, the proposed routing protocols decrease average delay and sensing time. EERP and EERP+S are reduce the average hop count. We observe this when compare EERP & EERP+S with Epidemic & Ep-Soc protocols. Therefore, combining social features with energy aware forwarding techniques are significant for improve MSSN routing. These proposed protocols are an efficient routing protocols for MSSN. As future work, we have planning to consider the number of nodes as density parameter. We will investigate node density impact on the proposed routing protocols.

REFERENCES

[1] B. Hull, V. Bychkovsky, K. Chen, M. Goraczko, A. Miu, E. Shih, Y. Zhang, H. Balakrishnan, and S. Madden, *CarTel: A Distributed Mobile Sensor Computing System*, ACM SenSys, (2006), pp. 125–138.

[2] B. Kranstauber, A. Cameron, R. Weinzerl, T. Fountain, S. Tilak, M. Wikelski, and R. Kays, *The Movebank data model for animal tracking*, Journal of Environmental Modelling and Software, Vol.26. No.6 (2011), pp. 834–835.

[3] L. Ferrari, and M. Mamei, *Discovering daily routines from Google Latitude with topic models*,International Conference on Pervasive Computing and Communications, PerCom 2011, 21-25 March 2011, Seattle, WA, USA, Workshop Proceedings, pp. 432–437.

[4] A. Vahdat and D. Becker, *Epidemic Routing for Partially-Connected Ad Hoc Networks*, *Technique Report, Department of Computer Science, Duke University, USA*, vol.20, no. 6, 2000.

[5] S. Sati, A. Ippisch, and K. Graffi, *Dynamic replication control strategy for opportunistic networks*,in *2017 International Conference on Computing, Networking and Communications, ICNC 2017, Silicon Valley, CA, USA, January 26-29, 2017*, 2017, pp. 1017–1023.

[6] A. Keränen, J. Ott, and T. Kärkkäinen, *The ONE simulator for DTN protocol evaluation*, Proceedings of the 2nd International Conference on Simulation Tools and Techniques for Communications, Networks and Systems, SimuTools 2009, Rome, Italy, March 2-6, 2009, pp. 55.

[7] S. Sati, C. Probst, and K. Graffi, *Analysis of Buffer Management Policies for Opportunistic Networks*,in 25th International Conference on Computer Communication and Networks, ICCCN 2016, Waikoloa, HI, USA, August 1-4, 2016, pp. 1–8.

[8] S. Sati, A. Sohoub, A. Eltahar,K. Bin Ahmad, K. Ahmad,and A. Bakeer, *Degree Contact DC-Epidemic Routing Protocol For Opportunistic Networks*, Advances in Wireless and Optical Communications (RTUWO) 2018, Riga, Latvia, November 15 - 16,, 2018, pp. 198–203.

[9] O. Landsiedel , E. Ghadimi , S. Duquennoy , and M. Johansson, *Low power, low delay: opportunistic routing meets duty cycling* , The 11th International Conference on Information Processing in Sensor Networks (co-located with CPS Week 2012), IPSN 2012, Beijing, China, April 16-19, pp. 185–196.

[10] S. Biswas , and R. Morris, *Exor: opportunistic multi-hop routing for wireless networks* ,SIGCOMM Computer Communication Review, vol. 35, no. 4,pp. 133–144.

[11] Q. Cao, T. He, L. Fang, T. Abdelzaher,J. Stankovic ,and S. Son, *Efficiency Centric Communication Model for Wireless Sensor Networks*, Proceedings of 25th IEEE INFOCOM, Barcelona, Spain. 2006, pp. 1–12.

[12] M. Busse, T. Haenselmann, and W. Effelsberg, *An Energy- Efficient Forwarding Scheme for Wireless Sensor Networks*, Proceedings of WOWMOM06, IEEE Computer Society, Washington, DC, USA. 2005, pp. 125–133.

[13] M. Busse, T. Haenselmann, and W. Effelsberg, *Poster-Abstract: A Lifetime-Efficient Forwarding Strategy for Wireless Sensor Networks*, Wireless Sensor Network. [Poster Abstract], 2006; 20.

[14] M. Hung, K. Lin, C. Hsu, C. Chou,and J. Tu, *On enhancing network-lifetime using opportunistic routing in wireless sensor networks*, Proceedings of 19th International Conference on Computer Communications and Networks (ICCCN), Zurich. 2010, pp. 1–6.

[15] C. Hsu, H. Liu, and W. Seah, *Opportunistic routing - A review and the challenges ahead* ,Computer Networks. 2011; 55(15), pp. 3592–3603.

[16] J. Wang,J. Kim, L. Shu,Y. Niu , and S. Lee, *A distance-based energy aware routing algorithm for wireless sensor networks*, Sensors. 2010; 10(10):,pp. 9493–9511.

[17] R. Mikkili , and J. Thyagarajan, *A real-time routing protocol with controlled dissemination of data queries by mobile sink in wireless sensor networks*, Indian Journal of Science and Technology. 2015 Aug; 8(19),pp. 1–10.

[18] S. Baba, and K. Rao, *Improving the Network Life Time of a Wireless Sensor Network using the Integration of Progressive Sleep Scheduling Algorithm with Opportunistic Routing Protocol*, Indian Journal of Science and Technology. 2016 May; 9(17),pp. 1–6.

Edited by: Ashish Kr. Luhach
Received: Jun 26, 2019
Accepted: Sep 26, 2019

DOI 10.12694/scpe.v20i4.1584
Scalable Computing: Practice and Experience ISSN 1895-1767
Volume 20, Number 4, pp. 653–661. http://www.scpe.org © 2019 SCPE

EVALUATION OF AODV AND DYMO ROUTING PROTOCOL USING GENERIC, MICAZ AND MICAMOTES ENERGY CONSERVATION MODELS IN AWSN WITH STATIC AND MOBILE SCENARIO

SURESH KUMAR*[1], KIRAN DHULL*, DEEPAK SHARMA*, ARORA PAYAL*AND SANDEEP DAHIYA[†]

Abstract. Adhoc Wireless Sensor Networks (AWSN's) have found an increasing utility in various applications. In AWSN, sensor nodes are arranged in a topology which is suitable to the requirement of the nature of task. As the sensors are not connected in a fixed manner, their networking, location and functioning keeps on changing based on the situation due to which these are called Adhoc networks. The biggest challenge is to keep the sensors working for longer time by conserving their energy. Therefore, a suitable routing protocol needs to be selected to meet the energy conservation requirement at different nodes. In the present paper, we have evaluated the three energy conservation models i.e. Generic, Micaz and Micamotes for Ad-Hoc On demand Distance Vector (AODV) and Dynamic MANET On Demand (DYMO) routing protocol. The evaluation is carried out using the parameter metrices: Average End-to-End Delay (AEED), Throughput, Energy consumed in Transmit mode and Receive mode. Based on the simulation results, it has been observed that Micamotes energy model using AODV routing protocol performs better in terms of energy consumption upto 42.99% and 29.90% in transmit and up to 59.24% and 33.96% in receive mode respectively as compared to Generic and Micaz energy model.

Key words: WSN, Ad-hoc Wireless Sensor Network (AWSN), AODV, Mobile Ad-hoc Networks (MANETS), RREQ, RREP, RERR, DYMO, AEED, Total Packet Received (TPR), Packet Delivery Ratio (PDR), Optimized Link State Routing(OLSR)

AMS subject classifications. 68M10, 68M20, 60K05, 60K25

1. Introduction.

AWSN consists of independent nodes that communicate with each other by preserving connectivity in a distributed manner and creating a multi hop radio system. Wireless communication ability is present in every sensor node. They also have intelligence for processing of signal and data networking. They draw energy from batteries for communication between the sensors and the network. While communicating with other sensors, the energy consumed is higher [1].

Deployment of AWSN's is not on a large scale as the research in this field is by and large simulation based. Among other simulation parameters, the mobility model in AWSN plays a vital role to evaluate the performance of the protocol. AWSN's are joint arrangement of mobile nodes without the support of any central node that can communicate with one another. It gives the realistic and efficient use of radio communication channel and multi hop radio relaying. End user controls this network with the enhancement technology rather than a single authority and can be used for particularly sensitive applications. In AWSN's, node mobility is a major problem because of the ad-hoc features like restricted bandwidth, vibrant network topology, shared medium, security and multi hop character etc. [2]. Seamless mobility is an efficient method of mobility management in ad-hoc networks which offers simple access and efficient communication between nodes present in the arrangement.

Simulation is one of the most significant method for testing the features of ad-hoc networking protocol which gives various benefits like parameters isolation, repeatable scenarios and discovery of a several metrics. Topology and movement of the nodes are the important issues in evaluating the performance of network protocol. The mobility model defines the status of the nodes which are moving in a position on being dispersed initially inside the network. Accurate performance cannot be obtained by simulation results that are achieved with idealistic movement models,since nodes mobility directly distracts the performance of the protocol and for AWSN, idealistic movement scenarios cannot be obtained by the majority of existing mobility models[3]. The architecture of AWSN is depicted in figure 1.1. WSN's are deployed in volatile and unfriendly environments where deployment using wired networks is not possible. In case of forests, it is difficult to go down and setup a wired network. Therefore, the sensor nodes are made to fall from the air. Scalability is another merit of WSN which allows its use in Structural Health Monitoring, as the fixed and dense deployment in wired mode is

*Department of Electronics and Communication Engineering, University Institute of Engineering and Technology, Maharshi Dayanand University, Rohtak, Haryana, India (sureshvashist.uiet.ece@mdurohtak.ac.in,, kirandhull19@gmail.com, d.29deepak@gmail.com payalarora325@gmail.com).

[†]BPS Mahila Vishwavidayala, Khanpur, Haryana, India (sandydahiya2001@yahoo.com).

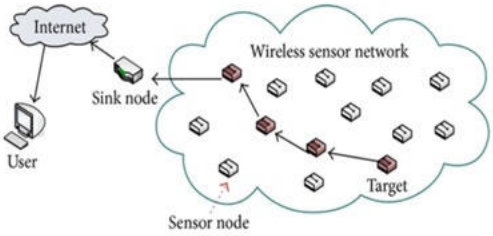


Fig. 1.1. *Architecture of AWSN*

required which is very expensive. However, WSNs can be deployed with least efforts and developed on a large scale according to the application. These can be networked easily and accessed with the remote area.

Three topologies (Mesh, Star and Tree) can be used for designing of the WSN's. These are economical,consist of little power with various nodes and at the same time can be maintained easily. Lesser energy computation capabilities and communication assets are the main disadvantage of WSN. In spite of these, it also has less storage space, low battery power and bandwidth and is prone to the attacks. Despite of these weaknesses, WSN's are broadly used and hence it may be rightly called as backbone of "Internet of Things". AWSN's are basically used in networks such as community and home network, vehicle and emergency response networks.

The other characteristics of AWSN's are unlicensed frequency spectrum, less cost of infrastructure instalment, sudden allocation of data around the sender, flexibility and mobility enhancement and capable of bringing up and tearing down in a very small time frame. Due to multi-hop support, communication beyond the Line of Sight (LOS) is possible at high frequencies, which diminishes the consumption of power of wireless devices. The Section 2 discusses the various routing protocols, Section 3 contains the recent related work reported in literature, the proposed work and simulation setup is described in Section 4. Section 5 covers the simulation results with discussion and Section 6 concludes the paper.

2. Routing Protocols and Energy Models.

2.1. AODV. AODV is one of the routing protocols, where every mobile host acts as a dedicated router. Concept of Bellman ford distance vector algorithm is mainly expanded by AODV in a virtual way. It was particularly considered for dynamic WSNs. It becomes the virtual dynamic WSN's, when unpredictable topology changes take place in WSN due to failure of nodes. To develop the ad-hoc network, AODV algorithm is fairly appropriate for dynamic self-starting network as required by users. It includes a series number and one entry per table which is exactly same as the conventional approach of routing to retain the routing data up to date. It is a reactive routing protocol where routes are created in case of requirements. It ensures loop-free routes in the altered condition and also in case of repairing broken links where any node ought to be treated as dead node if currently not in use.

The demand of bandwidth existing to the mobile nodes is significantly less because the protocol does not practice any type of universal cyclic advertisement. Symmetric links are used by AODV among the adjusting nodes and it does not track the path between nodes, if one of the nodes cannot attend the other one. Nodes neither keep any routing data nor contribute in any periodic exchange of routing table and also do not lie on dynamic path. It comprises the conventional concept of routing table that includes the factors like: address of next hop, routing information, a series number and node handling, since the node retain the specific time spam after that its entrance must be discarded. It is notified by the adjusting node if the link is not working. In AODV, two series are used for obtaining the routing mainly (i) Query (ii) Reply with the control requests i.e. RREQ, RREP, RERR and HELLO message. While execution, RREQ message broadcasted by a node to the other node, thereafter the RREP message comes as unicast. The neighbouring nodes observe the RERR message only if the link is disconnected. For estimation and detection of the links, the HELLO message is used among the nodes. Figure 2.1 depicts the AODV Routing protocol.

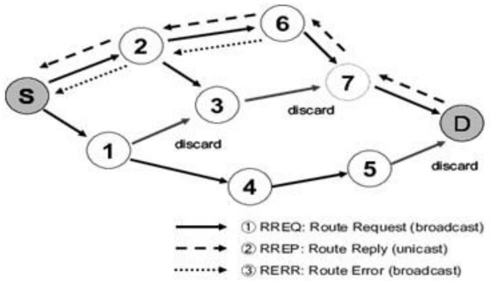

S
1
2
3
4
5
6
7
D
discard
discard
discard
discard
① RREQ: Route Request (broadcast)
② RREP: Route Reply (unicast)
③ RERR: Route Error (broadcast)

Fig. 2.1. *AODV Routing Protocol*

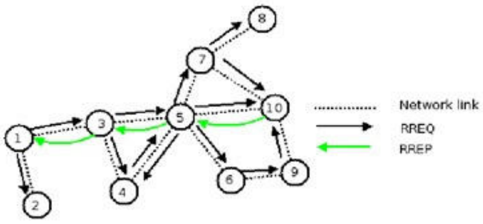


Fig. 2.2. *DYMO routing protocol*

2.2. DYMO. For multi-hop networks, DYMO is one of the effortless and high-speed routing protocols. It determines the routes based on the demand request and gives improved coverage for dynamic topologies inside the networks. To find out the route, data packet was transmitted by the source with RREQ message same as the AODV protocol. After receiving the RREQ message, DYMO waits and look for assignment of a route and it may issue another RREQ, if route is not assigned within waiting period. For minimising the obstruction, exponential back off method is used in the network. Figure 2.2 depicts the DYMO routing protocol network. When data packets have not been delivered then RERR message is issued to the destination due to absence of route. Small state information such as the active source and destination is retained in the DYMO router because the applicable devices like WSN have less memory [3].

Various Energy Models used for energy conservation are Generic, Micaz and Micamotes. To minimise the consumption of energy, Generic energy model is used to reduce the task which are completed by the sensors and their linked networks. Various approaches have to be used to replenish the energy capacity of sensor by constructing apparatus and mechanism for extra energy harvesting. The low power WSNs, Micamotes energy model used by researchers are second generation motes. It works at frequency 4MHz and processor used by it is Atrnel ATmega 128 L which is an 8 bit microcontroller and has 128 Kb flash memories to store the mote program. It consumes 8 A and 15 A current in running mode and sleep mode respectively. Two AA batteries are used by Micamotes which can run for more than a year. Micaz is a third generation tool used for improvement of WSNs which are made up of low power. It works on 2.4 to 2.48 GHz for extremely embedded sensor network. Because of 3G mote characteristics, it always uses higher range and larger data rate.

3. Related Research Work. In [3], the authors have presented an evaluation of performance using AODV and DYMO protocols based on CBR network using AEED, Jitter, TPR and throughput. When compared with DYMO, AODV has better throughput and less jitter and received more number of packets. In [4], authors have proposed a vigorous protocol i.e. Distributed X-layer Faster Path (DXFP) that depends on the cross layer metric which has several parameters like topology, mobility, congestion of nodes, fading, channel quality and power constraints. These are integrated with less cost which reduces the network crossing time. The DXFP protocol is further compared with AODV and DSR and is found better which improves performance and robustness of WSN.

In [5], the authors have presented a comparison of three protocols i.e. AODV, Destination Sequence

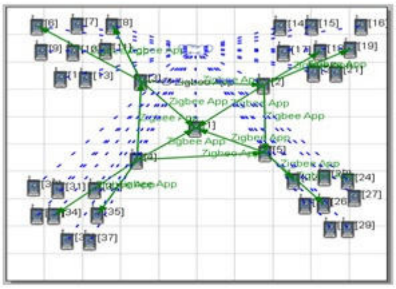


Fig. 4.1. *Network Scenario I(37 static nodes) Before Simulation*

Distance Vector (DSDV) and Ad-hoc on-Demand Multipath Distance vector (AOMDV) using IEEE802.11 and IEEE802.15.4 MAC standards. The simulation is done using MATLAB and NS-2 and the evaluation is based on MinMax and MinTotal performance metrics. The results determined using IEEE802.15.4 standard is greatly beneficial for engineers and designers in the energy sensitive applications and for efficient deployment of WSNs. In [6], authors have presented an approach for reducing the problem related to dropping of packets by malicious nodes by using IPSec in OLSR and STAR routing protocols in the network. From simulation outcomes, it has been evaluated that proposed approach has better TPR and throughput and has less packet dropped as compared to existing without IPSec protocol. In presence of malicious nodes, the proposed approach gives better performance of MANET.

The authors in [7] presented two reactive protocols named AODV and DYMO and then particle swarm optimization is used for optimization. For resolving the problems regarding consumption of energy and network overhead, adaptive hello messaging scheme is used and a dependable connection between the source and destination node has been created. It has been proved from the simulation results that this scheme minimised the consumption of battery and restrained needless hello messaging. In [8], the authors discussed and analyzed the Angle based routing protocols named AAODV, AOLSR and AZRP.These have been developed for Mobile Ad-Hoc networks or mobile WSNs and their comparison is done with the exiting protocols named AODV, DYMO, OLSR, ZRP. In the case if data request received, result is not greatly influenced except AAODV and in the case of transmitted signal; all the protocols are affected uniformly except AOLSR.

In [9], the authors evaluated the effects of wormhole attack using protocols DSR and AODV which has been have been analyzed and compared. After simulation, it has been observed from graphs that AODV gives better performance in case of small number of nodes but if nodes increased, overhead are also increased in huge amount. If the length of colluding link enhances, the performance for DSR performance is degraded than AODV. It is proved that, as comparison of AODV protocol, the effect of wormhole attack is much strict for DSR.

In [10], the authors have compared the three types of protocols namely: Reactive(AODV and DYMO), Proactive (OSLR) and Hybrid(IERP) protocol based upon Throughput, AEED, Jitter, PDR and TPR as performance metric using Random Way point Mobility Model. It has been concluded from simulative results that reactive protocols have improved performance incase of throughput, PDR and TPR.

In [11], the authors have evaluated OLSR and DYMO routing protocol's performance on the basis of two network metrics i.e. normalized routing load (NRL) and PDR. It has been examine that OSLR outperforms DYMO in terms of NRL and DYMO outperforms OLSR in terms of PDR. It was further observed that NRL increases and PDR decreases with increase in speed, Increasing topological area and number of nodes whereas NRL and PDR are better in case of increasing pause time and mobility speed of both protocols.

4. Simulation Setup. In this present work, we have evaluated the performance of routing protocols AODV and DYMO by using different number of nodes for a designed mobile Ad-hoc wireless sensor network with two scenarios of 37 and 50 nodes using Qualnet 7.3.1 simulator. The designed network scenarios consists of 37 nodes uniformly placed and 50 nodes randomly placed nodes in a terrain 1500 x 1500 square meters.The figures 4.1-4.4 show the designed network before and during simulation by using 37 nodes and 50 nodes separately for static and mobile scenarios.

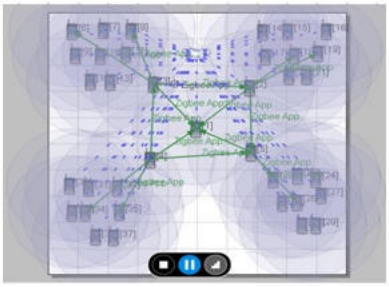


Fig. 4.2. *Network Scenario I(37 static nodes) After Simulation*

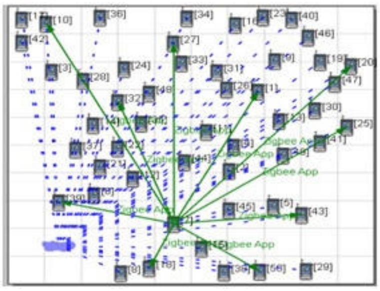


Fig. 4.3. *Network Scenario II(50 static nodes) Before Simulation*

For simulation, we have transmitted 100 data packets containing about 70 bytes of data over a wireless link. ZigBee application is used between the selected nodes and simulation time is kept at 500 seconds with a start and end time of 1 and 300 second respectively using ”Random Way point Mobility” mobility model . The performance of three different energy conservation models Generic, Micaz and Micamotes is evaluated using AODV and DYMO routing protocol. Energy consumed in transmit and receive modes by varying number of nodes are taken as performance metric. The various parameters are considered for simulation is given in Table 4.1.

5. Results and Discussion. In this proposed work, we have simulated the designed network scenario using AODV and DYMO routing protocol in order to analyze the performance of three energy models i.e. Generic, Micaz and Micamotes.

Energy consumed in Transmit and received mode:. This is equal to the total energy consumed in transmission plus reception mode. To achieve greatest sensor life, this performance metric ought to be least. Figures 5.1 and 5.2 show the variation of energy consumed in transmit and received mode for three different energy models using AODV and DYMO routing protocol for static scenario of 37 nodes.

From Figures 5.1 and 5.2, it has been observed that AODV offers enhanced performance when compared with DYMO routing protocol. Further from figure 5.1 and 5.2, the Micamotes energy model consumed lesser energy and an improved performance of 28.31% and 4.89% as compared to Generic and Micaz energy models respectively in transmit mode and an improvement of 9% and 3.5% for Generic and Micaz energy models respectively in random mobile scenario.

The Table 5.1 shows the numerical values of the graphical results of figure 5.1 and 5.2. Hence, it can be concluded that Micamotes energy model seems to be better in energy saving and the Generic model consumes more energy.

Figures 5.3 and 5.4 show the variation of energy consumed in transmit and received mode for three different energy models using AODV and DYMO routing protocol for mobile scenario of 50 mobile nodes.

From the results, it can be figured out that, behaviour of the energy consumption in transmit and received mode for scenario II is somewhat different and consumes less energy as compared to static scenario. Further from figure 8 and 9, the Micamotes energy model consumed lesser energy and an improved performance of

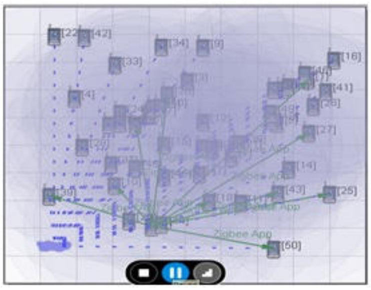


Fig. 4.4. *Network Scenario II(50 static nodes) After Simulation*

Table 4.1
Simulation Parameters

Parameter	**Value**
Simulator	Qualnet
Terrain Size	1500m * 1500 m
No. of Nodes	37,50
MAC Protocol	IEEE 802.15.4
Packet Reception Model	PHY 802.15.4
Radio Type	IEEE 802.15.4
Energy Model	Generic, Micaz, Micamotes
Routing Protocol	AODV, DYMO
Routing Protocol	AODV, DYMO
Traffic Type	ZIGBEE
Link	Wireless
Channel Frequency	868 MHz
Modulation	O-QPSK

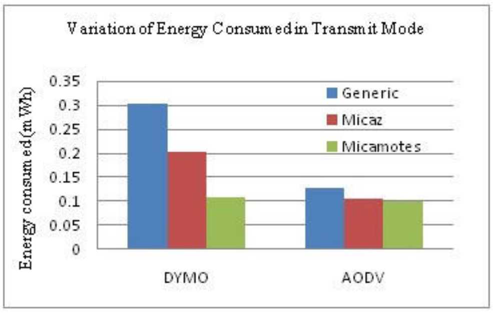


Fig. 5.1. *Energy Consumed in Transmit Mode in static scenario*

Table 5.1
Average Energy consumed in Transmit and Receive mode for static scenario

Model	**AODV Tansmit**	**DYMO Tansmit**	**AODV Receive**	**DYMO Receive**
Generic	0.3018	0.1287	1.755	0.549
Micaz	0.2033	0.1053	1.249	0.520
Micamotes	0.1073	0.1003	0.617	0.502

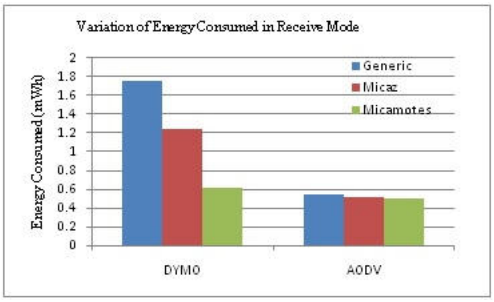


FIG. 5.2. *Energy Consumed in Received Mode in static scenario*

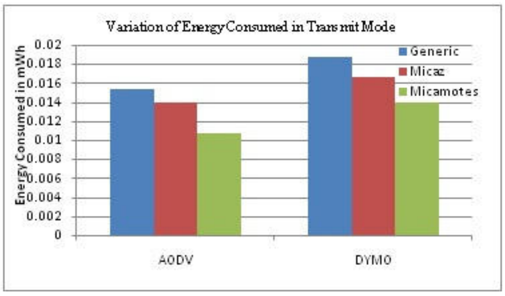


FIG. 5.3. *Energy Consumed in Transmit Mode in mobile scenario*

42.99% and 29.90% as compared to Generic and Micaz energy models respectively in transmit mode and an improvement of 59.4% and 33.96% for Generic and Micaz energy models respectively in random mobile scenario. The Table 3 shows the numerical values of the graphical results of Figures 5.3 and 5.4.

Hence from Figures 5.1, 5.2, 5.3 and 5.4 and Table 5.1 and 5.2, it can be concluded that Micamotes energy model seems to be better energy saving and the Generic model consumes more energy. Further using Micamotes as energy conservation model we have carried out performance evaluation of AODV and DYMO routing protocols for the parameters (i) AEED (ii) Throughput.

AEED. It is average time taken for messages to travel through the network from sender to destination. However, time interval is because of propagation, buffering, queuing and retransmissions of all existing data packets which are being sent.

From Figure 5.5, it concluded that in terms of AEED, AODV performs better and provides less AEED as compared to DYMO for both scenario I (static nodes) and scenario II (mobile nodes).

Throughput. It is the average date rate successfully transmitted over a communication channel and is measured in (i) Bits/sec (ii) Packets/sec. For a better performance, higher throughput is always desirable. The throughput variation with and without number of nodes mobility is shown in Figure 5.6.

From Figure 5.6, it can be concluded that for static and dynamic scenarios, AODV provides higher throughput as compared to DYMO protocol. Hence, from the results AODV outperforms DYMO protocol in selected performance metrics.

6. Conclusion. The performance of various energy models such as Generic, Micaz and Micamotes are analysed on the basis of network metrics such as Energy consumed in transmit and received mode over two scenarios of static nodes and random nodes using AODV and DYMO routing protocols. Form simulation results, it has been observed that Micamotes energy model using AODV routing protocol gives superior performance in terms of energy consumption upto 42.99% and 29.90% in transmit and upto 59.24% and 33.96% in receive mode respectively as compared to Generic and Micaz energy model. Upon further performance evaluation, AODV outperforms DYMO in terms of Throughput and AEED.

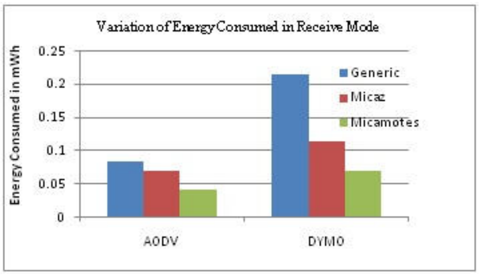


Fig. 5.4. *Energy Consumed in Received Mode in mobile scenario*

Table 5.2
Average Energy consumed in Transmit and Receive mode in mobile scenario

Model	AODV Tansmit	DYMO Tansmit	AODV Receive	DYMO Receive
Generic	0.0153	0.0187	0.0844	0.21499
Micaz	0.0139	0.01655	0.0710	0.11409
Micamotes	0.0107	0.0140	0.0530	0.07102

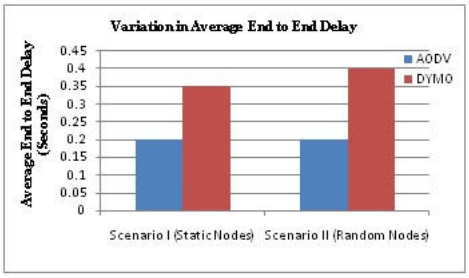


Fig. 5.5. *Variation of AEED for AODV and DYMO*

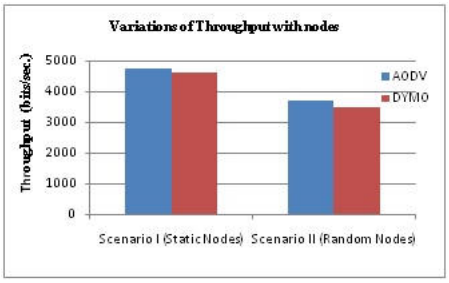


Fig. 5.6. *Evaluation of Throughput(i) AODV (ii) DYMO*

REFERENCES

[1] Kirti Mor, Suresh Kumar, Deepak Sharma, *Ad-Hoc Wireless Sensor Network Based on IEEE 802.15.4: Theoretical Review*International Journal of Computer Sciences and Engineering. Vol.6. Issue 3, pp (219-224) March 2018. ISSN 2347-2693.doi: 10.26438/ijcse/v6i3.219224.

[2] Deepak Sharma, *An overview of Wireless Sensor Networks*, International Journal of Enhanced Research in Management and Computer Applications, Vol. 4 Issue 4,pp(47-51), April-2015, ISSN: 2319-7471.

[3] Payal, Deepak Sharma, Suresh Kumar, , *Performance Evaluation of Reactive Routing Protocols Using IEEE 802.15.4 Application in Designed Wireless Sensor Network*, International Journal of Computer Sciences and Engineering. Vol.6. Issue 4, pp (90-96), March 2018, ISSN 2347-2693.doi: 10.26438/ijcse/v6i4.9096.

[4] L. Mucchi, L. Chisci, G. Giovannettiand L. Fabbrini,, *Robust Cross-Layer Routing Protocol for Mobile Ad Hoc Networks*, IEEE, pp (278-284), 2012, ISSN: 4673-2569.

[5] B.Brahma Reddy, K.Kishan Rao, LU-*Energy Optimization using Cross-Layer Protocols in Wireless Sensor Networks*, International Journal of Computer Applications, Vol. 65- No.4, pp (18-22), March 2013, ISSN:0975 - 888.

[6] Harish Shakywar, Sanjeev Sharma and Santoh Sahu, *Securing OLSR and STAR Routing Protocols against Packet Dropping by Malicious Nodes*, International Journal of Computer Applications, Vol. 35- No.3, pp (7-12) , December 2011, ISSN: 0975 - 8887.

[7] Sukhvinder Kaur, Sheenam Malhotra, *Neighbor Discovery based on Adaptive Hello Messaging Scheme On-Demand MANET Routing Protocols using PSO*, International Journal of Computer Applications,Vol. 138 - No.6, pp (36-39), March 2016, ISSN: 0975 - 8887.

[8] Saurabh mishra, Sandip Vijay, *Colparative Analysis of Angle Based and Traditional Routing Protocols for MANETs / WSNs*, Intelligent and Computing in Engineering , Vol. 10, pp. (251-254), 2017,ISSN 2300-5963, DOI: 10.15439/2017R90.

[9] Richa Agrawal, Rajeev Tripathi, Sudarshan Tiwari, *Performance Evaluation and Comparison of AODV and DSR Under Adversarial Environment*, International Conference on Computational Intelligence and Communication Systems, pp 596-600, 2011, ISSN: 7695-4587, DOI 10.1109/CICN.2011.129.

[10] Ashish Allen Roberts, Rajeev Paulus, A.K. Jaiswal, *WSN Performance Parameters of AODV, DYMO, OLSR and IERP in RWP Mobility Model through Qualnet*, International Journal of Computer Applications,Vol. 65- No.22, pp (28-32),March 2013, ISSN: 0975 - 8887.

[11] Muhammad Amin, Abdusalam, Abdus Salam Awan, Arshad Iqbal, *Analyses of OLSR and DYMO Routing Protocols Using Normalize Routing Load and Packet Delivery Ratio in Mobile Ad-Hoc Networks*, WULFENIA Journal, Vol. 19, No. 9, pp (416-437), Sep 2012, ISSN: 1561-882X.

Edited by: Ashish Kumar Luhach
Received: Jun 19, 2019
Accepted: Aug 13, 2019

DOI 10.12694/scpe.v20i4.1585
Scalable Computing: Practice and Experience ISSN 1895-1767
Volume 20, Number 4, pp. 663–667. http://www.scpe.org © 2019 SCPE

PREDICTIVE MAINTENANCE AND MONITORING OF INDUSTRIAL MACHINE USING MACHINE LEARNING

KAUSHA I. MASANI, PARITA OZA AND SMITA AGRAWAL*

Abstract. Machine learning is one of the break-through technologies of the modern digital world. It's applications are found in various research domain such as medicine, image processing, production and manufacturing, aviation and autonomics and many more. To efficiently run a machine, it's maintenance and its monitoring automation system play key role.The major problem we are targetting is to overcome the lack of an automation system which can give accuracy rate of the production machine at a given instance of time. Also the important energy meter parameters required to make power report in automation system for addressing the production issues, at given interval of time, were also not recorded. Thus in this paper, we describe how machine learning techniques is used for prediction of accuracy of running production machine. To address this issues, we have used supervised machine learning technique of Binary decision tree using CART method and for power report, while the data is fetched using RS232 to RS485 convertor via Modbus communication protocol. Using CART we have predicted the machine accuracy at a given time with specific energy meter readings as its input features. This paper discusses the problem definition identified, data analysis of energy meter data and it's fetching and at the end ML techniques applied to predict accuracy of running production machine. In the end we prepare various power reports of the different machines from the fetched parameters as well as produce a graphical warning of deteriorate performance of the machine at a given instance of the time.

Key words: Machine Learning, Decision tree, CART, Binary Classification, Supervised Machine Learning technique, Energy Meter, ModBus Communication Protocol, Power.

AMS subject classifications. 97R40

1. Introduction. The world is moving towards fast automation, an era of the digital and advanced technology-driven world of production. The production of any product depends upon the performance of the machine used for its manufacturing process. The maintenance and monitoring of this machine hence is a vital aspect to increase the sale of the product as well as to maintain the quality of the product.

We addressed the area of machine maintenance and monitoring via a machine learning model, for a single machine of a cement production plant. In this paper, we discuss the problem area, use of a machine learning model to address the problem, selected features for the model and their criticality in detail and output of the model. The supporting idea for this research work was found in [1-16] where fault detections were made in the automatic meter reading systems, to improve scalability and reliability.

2. Problem. During measure the machine accuracy for the given set of data, specific problems may arise. There is no such system which can fetch the data from energy meters of the machine and prepare data for the ML model. Also, there is no such ML model which can predict the accuracy of the machine for the given set of the data.

Here we proposed an ML model which can predict the accuracy in which machine is running for the given set of energy meter data and predict the accuracy of the machine for the future process before it comes to breakdown point. This paper also discussed the method used to fetch the energy meter data from a remote location to the local system using RS232 to RS485 converter.

3. Introduction to ML and It's Techniques. Machine learning is fast taking over the manufacturing industry to provide prognostics for the industrial machine to increase the efficiency of the machine and to prevent machine failure. The use of machine learning and its techniques depends on the type of application and data available. Machine learning has mainly four types, i.e. supervised learning, Semi-supervised learning, unsupervised learning and reinforcement learning.

Supervised learning is the method which we are adapting to predict the performance of the pre-decided machine. Supervised learning provides a variety of techniques for classification regression analysis of the data. These techniques are Bayesian Networks, Support vector machines, Naive Bayes, Decision trees, k-NN, Neural

*Department of Computer Science and Engineering, Nirma University, Ahmedabad, India ({kaushamasani@gmail.com, parita.prajapati@nirmauni.ac.in, smita.agrawal@nirmauni.ac.in})

networks and so forth [2]. All these serve the purpose according to the data and nature off the applications [14,15].

Some brief explanation about these is mentioned below [2]:

- Baysian Network [2]: It has interpretation ability to define problem into the structural relationship among predictors. Due to this training time takes less computational period. The additional advantage is that there are no free parameters to be set.
  However, disadvantages for Supervised ML method is that it is unable to perform when dataset grows as well as it is unable to handle high dimensional data.
- Logistic Regression [2]: This method has been used when the dependent variable or target variable is divided. In this method, the probabilistic interpretation is excellent, and the model can be updated to take new data easily. The main disadvantage of this method is that it needs a large sample size to achieve stable results. This model has its application in the area which deals with crash types, injury severity, voters types, and so forth.
- K Nearest Neighbor [2]: k-NN is non-parametric classification algorithm. It assigns to a non-labelled point, which is the nearest class of previously labelled point. It is widely used for multi-label applications and multi-modal classes. Though it has lower efficiency, it is also addressed as a simple lazy learning method. The performance of this model all depends on selection value of 'k'. The major drawback of this method is that its sensitive to irrelevant features, affected adversely by noise, its performance also varies with varied data.
- SVM [2]: Support Vector Machines named SVM, have complex algorithms which provide high accuracy. With appropriate kernel, they can work well even if the data is not linearly separable. It avoids overfitting, has the flexibility for selection of kernels for nonlinearity. It has good generalisation ability, but it has some disadvantages also. It is very complex, along with slower training speed. Also, its performance mostly depends on the selection of parameters this method majorly used in text-classification applications.
- Decision Tree [2]: Decision tree is easy to interpret and explain. Additionally, they can easily handle the interaction between the features. Though non-parametric, outliers do not affect the model. Some famous algorithm is ID3, CART, C4.5, and C5.0 according to the various splitting data such as Info Gain, Gini Index, Gain Ratio and Gini Co-efficient. A decision tree can handle a variety of the data, missing values, redundant attributes; have good generalisation abilities; are robust to noise; provides high-performance data with very small computational data. These use divide and conquer approach, which performs accurately with highly relevant features.
  Due to these reasons, the relevancy of dataset and the availability of highly relevant features, a Decision tree is finalized as the model for the present system in this paper.

4. Proposed ML model. The proposed model adapted for this application is the Decision tree with binary tree classification. Cross-validation and entropy are the methods used for splitting the sampled data and calculating the highest variation parameter, respectively.

Dependent Variable. The dependent variable as explained in [3] determines the goal of the study; the user chooses it. In our example, power is selected as the dependent variable. Below the root node, we find the next level of the tree. Here, the tree selects variable current-phase as a predictor for the dependent variable and separates all households according to the predictor's values. The separation of data is called a split.

Selection of Particular Split. A split in a decision tree relates to the predictor with the maximum separating power. In other words, the best split does the best job in the creation of nodes where a single class dominates. In our example, power best splits the data based on its entropy, as power has highest entropy amongst all parameters. There are several methods of calculating the predictor's power to separate data. One of the best-known methods is based on the Gini coefficient of inequality [3].

Gini Coefficient. The Gini coefficient is [13],essentially, a measure of how well the predictor separates the classes contained in the parent node. Computation of the Gini coefficient is illustrated in Fig. 4.1[3]. Explaining with an example to explain Gini- coefficient more in simple form. Suppose if we compute the graph between richest to poorest members of the society, in terms of wealth, we get diagonal corresponds to an equal

FIG. 4.1. *Example to Explain Gini-Coefficient*

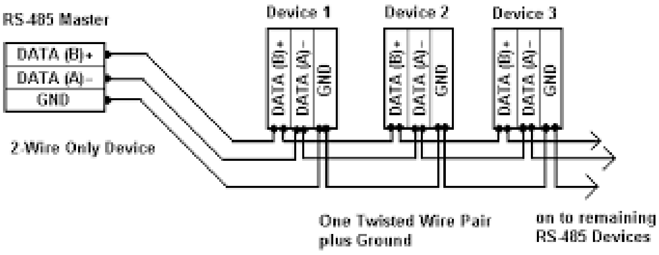


FIG. 5.1. *Master-Slave Serially Connection Used In ModBus Protocol*

distribution of wealth, and the curve above it represents a real economy, where there is always some inequality in the income distribution [3].

The Gini coefficient is calculated as the area between the curve and the diagonal divided by the area below the diagonal. In Fig. 4.1 the computation of the Gini coefficient of inequality Classification and Regression Trees (CART) use the Gini's measure of inequality for selecting splits. With the help of the Gini index, it is decided which attribute is to be split and what is the splitting point of the splitting of the attribute. The Gini index is minimized at each split so that the tree becomes less diverse as we progress. The class histograms are built for each successive pairs of values of attributes. At any particular node, after obtaining all the histograms for all attributes, the Gini index for each histogram is computed. The histogram that gives the least Gini index gives us the splitting point P for the node under consideration.

The most widely used measure of impurity (at least by CART) is according [4]

$$\text{Gini index: } i(p) = \sum_{i \neq j} p_i p_j = 1 - \sum_j p_j^2$$

We define the impurity of a tree to be the sum over all terminal nodes of the impurity of a node multiplied by the proportion $p_i p_j$ of cases that reach that node of the tree [4]. The stopping criterion is when the Gini index at a node becomes zero, as this implies that all data records at that node have been classified completely.

5. Fetching The Data For The ML Model. The real-time data or input features used for the ML model are first being fetched from energy meters, which are located at the various remote location of the cement plant. These fetched data is then dumped into the database created for the same. The fetched data is then used for the preparation of the input file for the ML model.

The energy meter of the cement mill machine is Modbus enabled. Thus it shall pass the values over LAN using Modbus communication protocol. This protocol is a serial communication protocol, widely used in industrial machine communications. This idea for fetching the data through a convertor was sourced from inspiration from [5]. The connection diagram of a master device with a serially connected slave device is shown in Fig. 5.1.

The system build using the convertor is shown in the Fig. 5.2.

As shown in Fig. 5.2 the data from energy meter is fetched via RS232-Rs485 convertor into the developed SCADA-like system in a remote computer and then it is dumped into the database. 1440 data records per energy meter are fetched in 1 day. So the pool of data for data processing and analysis becomes large when total records are multiplied by total no of energy meters.

6. Data Pre-processing and Analysis. The features for the proposed system are selected from amongst 32 parameters, based on their criticality concerning the system. These parameters are average voltageCurrent phase-1, current phase-2, current phase-3, kWh. Below mentioned is the standard power equation for the

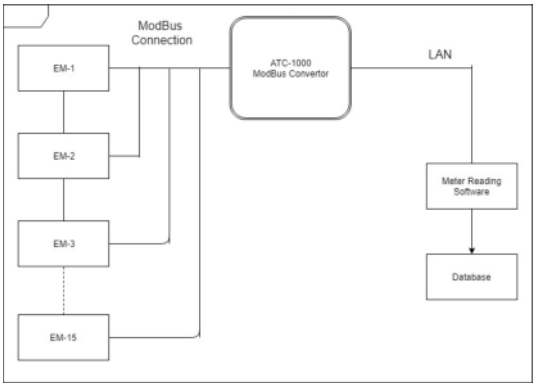


Fig. 5.2. *Block Diagram of Data Fetching Mechanism*

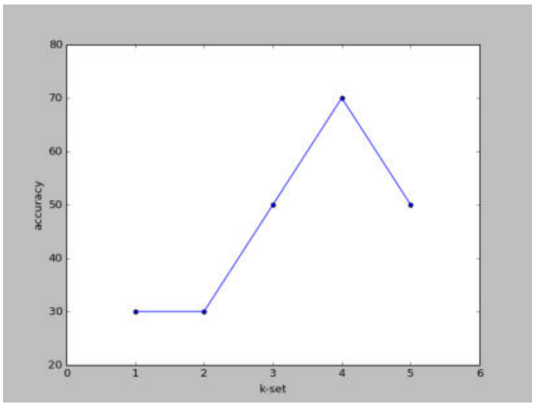


Fig. 7.1. *Reultant Graph Representation*

purpose to understand why power is critical and shall be having more entropy, which can be explained like:

$$\text{Power Equation: } P = \sqrt{3} \cdot V_L \cdot I_L \cdot Cos\phi \tag{6.1}$$

where:

V_L = Voltage Line
I_L = Current Line
$Cos\phi$ = Power factor (0.8)

As observed from the above equation, power depends upon its supplied current, voltage and power factor. The power factor we are taking as a constant value. The I_L in the equation varies depending upon the load of the material machine is currently running for. Thus current phases 1, 2 and 3 are also considered equally critical along with power.

The data received is converted into a decimal form from the exponential form for analysis purpose. Once the CSV file is ready, we are ready to deploy it as an input file to the proposed model.

7. Test Results. When the program is deployed with input file in the form of CSV, initially the whole records are made in the set of 5 sets. These sets are then processed with the ML model giving the graphical representation results, as shown in Fig. 7.1.

As we observed that, the whole dataset is sampled into 5 sets where each of them giving efficiency rate individually. We can deduce that the dataset of set one has accuracy the lowest along with dataset two. However, as the workload of the machine is increased, the power and efficiency of the machine also rise as depicted in the above graph. This shows the required result of this whole system, i.e. the maximum efficiency of the running machine before it reaches its breakdown point.

This system helps the engineers to avoid untimely breakdown of the system during peak production hours. The implementation of this system not only helps with knowing the efficiency at a given point of time for the data but also helps in making the production report of the designated department in any manufacturing unit. This system not only helps in improving the working of an industrial machine but also helps in monitoring its performance during it is running time.

8. Conclusion. Machine learning is currently taking over not only the robotics section of the technology but its also take place faster in the industrial and production sector of the manufacturing industry. The adaptation and implementation of ML techniques are now catching up the trend and is fast moving towards automation with quality production.

The main aim here is to implement one such method to predict the accuracy of the production machine, to improve the production quality as well as quantity and remaining useful lifetime of the machine. The paper also tries to provide one such novel method in the subject area of automation in the industrial machine with the help of machine learning, and it is available techniques. This outcome of the system not only shall help to prepare various power reports of the different machines but also shall be able to generate a graphical warning of deteriorating performance of the machine at a given instance of the time.

REFERENCES

[1] Y. Kou, G. Cui, J. Fan, X. Chen, W. Li, Machine learning based models for fault detection in automatic meter reading systems, 2017 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC), pp. 684-689, 10.1109/SPAC.2017.8304362

[2] A. Singh, N. Thakur, A. Sharma, A Review of Supervised Machine Learning Algorithms, 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), pp. 1310–1315, 2016

[3] S. Sivagama Sundhari, A knowledge discovery using decision tree by Gini coefficient, 2011 International Conference on Business, Engineering and Industrial Applications, pp. 232-235, 2011

[4] Z. Jian, W. Zhaowei, Q. Changsong, Study on the key quality parameter decision for multi-process by CART method, 2010 International Conference on Computer Application and System Modeling (ICCASM 2010), vol. 13, pp. V13-612-V13-616, 2010

[5] A. Kanawaday, A. Sane, Machine learning for predictive maintenance of industrial machines using IoT sensor data, 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS), pp. 87-90, 2017

[6] X. Zhao, M. Li, J. Xu, G. Song, Multi-Class Semi-Supervised Learning in Machine Condition Monitoring, 2009 International Conference on Information Engineering and Computer Science, pp. 1-4, 2009

[7] N. Amruthnath, T. Gupta, A research study on unsupervised machine learning algorithms for early fault detection in predictive maintenance, 2018 5th International Conference on Industrial Engineering and Applications (ICIEA), pp.355-361, 2018

[8] L. Kovács, G. Z. Terstyánszky, Diagnosising Faults by Supervised and Unsupervised Learning, 1999 European Control Conference (ECC), pp. 4238–4242, 1999

[9] V. Mathew, T. Toby, V. Singh, B.M. Rao, M. G. Kumar, Prediction of Remaining Useful Lifetime (RUL) of turbofan engine using machine learning, 2017 IEEE International Conference on Circuits and Systems (ICCS), pp.306-311,2017

[10] S. Dhankhad, E. Mohammed, B. Far,Supervised Machine Learning Algorithms for Credit Card Fraudulent Transaction Detection: A Comparative Study, 2018 IEEE International Conference on Information Reuse and Integration (IRI), pp. 122-125, 2018

[11] M. Balamurugan, S. Kannan, Performance analysis of cart and C5.0 using sampling techniques, 2016 IEEE International Conference on Advances in Computer Applications (ICACA), pp. 72-75, 2016

[12] L. Yifei, W. He, P. Shuai, S. Weiqiong, D. Ning, W. Fang, Application of supervised machine learning algorithms in diagnosis of abnormal voltage, pp. 1-5, China International Conference on Electricity Distribution (CICED), 10.1109/CICED.2016.7576347,2016

[13] Y. Kou, G. Cui, J. Fan, X. Chen, W. Li, Towards Statistical Modeling and Machine Learning Based Energy Usage Forecasting in Smart Grid, SIGAPP Appl. Comput. Rev., vol 15, pp. 6–16, 2015

[14] S.S. Agrawal, A. Patel, CSG cluster: A collaborative similarity based graph clustering for community detection in complex networks, International Journal of Engineering and Advanced Technology, 8 (5), pp. 1682-1687, 2019

[15] S. Agrawal, A. Patel, Clustering Algorithm for Community Detection in Complex Network: a Comprehensive Review, Recent Patents on Computer Science, 2019, 12: 1.

[16] N. Patel, P. Oza P., S. Agrawal, Homomorphic Cryptography and Its Applications in Various Domains. In: Bhattacharyya S., Hassanien A., Gupta D., Khanna A., Pan I. (eds) International Conference on Innovative Computing and Communications. Lecture Notes in Networks and Systems, vol 55, Springer, 2019

Edited by: Ashish Kumar Luhach
Received: Jun 9, 2019
Accepted: Aug 20, 2019

DOI 10.12694/scpe.v20i4.1569
Scalable Computing: Practice and Experience ISSN 1895-1767
Volume 20, Number 4, pp. 669–685. http://www.scpe.org © 2019 SCPE

THE SLOW HTTP DDOS ATTACKS: DETECTION, MITIGATION AND PREVENTION IN THE CLOUD ENVIRONMENT

A. DHANAPAL*AND P. NITHYANANDAM†

Abstract. Cloud computing is the latest buzzword and cutting-edge technology. The cost-efficiency, easy to operate, on-demand services, availability, makes the cloud so popular. The online web applications based on the internet such as E-Healthcare, E-Commerce are moving to the cloud to reduce the operating investment cost. These applications are vulnerable to slow HTTP Distributed Denial of Service (DDoS) attack in the cloud. This kind of attacks aims to consume the resources of the application as well as the hosting system so that to bring down the services. The various forms of the slow HTTP DDoS are HTTP header attack, HTTP body attack and HTTP read attack. Due to the nature of mimicking the slow network behaviour, this attack is very challenging to detect. This is even more difficult to identify in the cloud environment as it has multiple attack paths. The web applications running in the cloud should have been safeguarded from the slow HTTP DDoS attacks. This paper proposed a novel multi-stage zone-based classification model to identify, mitigate and prevent the slow HTTP DDoS attacks in the cloud environment. The solution is implemented using the OpenStack cloud environment. The open-source slowHTTPTest tool is used to generate different types of slow HTTP DDoS attacks,

Key words: DDoS, Application Layer Attacks, slowloris, RUDY attacks, slow HTTP attacks, OpenStack, Cloud Security, Layer 7 Attacks

AMS subject classifications. 68M14

1. Introduction.

Cloud computing helps the medium and small companies to reduce their initial investments [1] to build their data centres and the infrastructures. National Institute of Standards and Technology (NIST) defines cloud computing [2] is a model for enabling convenient on-demand network access to a shared pool of configurable computing resources such as networks, servers, storage, applications and services that can be rapidly provisioned and released with minimal management effort or service provider interaction. The cloud computing promotes the availability, and it exhibits the following five characteristics: on-demand self-service, broad network access, resource pooling, rapid elasticity and measured service.

1.1. Cloud computing classifications and threats.

The general broad classifications of cloud computing are:

- Service Delivery based model:
  This is based on the services provided by the cloud. This is further sub-classified as Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) [3].
- Deployment based model:
  This explains how the cloud is deployed. The finer sub-classification of this category are Private cloud, Public cloud, Hybrid cloud and Community Cloud [4].

Cloud computing is still evolving. The typical issues and threats faced by cloud computing are DDoS, access-control, data integrity, data security [5]. The cloud runs business-critical applications for E-Commerce, E-Healthcare and finance. The availability of these applications is very crucial in their business. The Distributed Denial of Service attack (DDoS) is one of the major threats to cloud applications. The details on the DDoS attacks explained in next sub-section.

1.2. DDoS attacks and types.

The Denial of Services (DoS) attack is a method in which attackers try to make the services or resources unavailable to legitimate users. In Distributed Denial of Service (DDoS) attacks, the intruder launches the attack to victim server or application from compromised machines from multiple locations. The attacker seed the malware into several computers over the internet with the help of the control machine. The compromised machines are acting as bot or army for the DDoS attack. Whenever the attacker decided to attack the victim server or application, he sends an attack command to the bot machines so that the bot machines start attacking the victim. The fundamental idea of the DDoS attack is shown in figure 1.1.

[*] Research Scholar, School of Computing Science and Engineering, Vellore Institute of Technology, Chennai, India. (Dhanapal.a2013@vit.ac.in, dhanapal.ang@gmail.com).

[†] Professor, School of Computing Science and Engineering, Vellore Institute of Technology, Chennai, India.

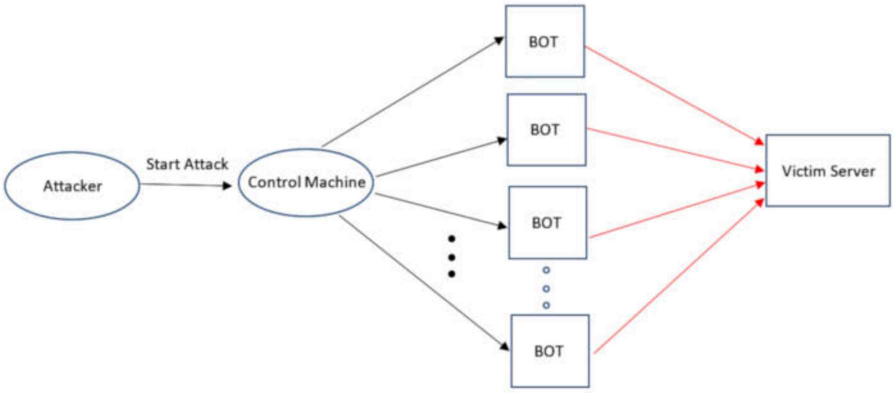


Fig. 1.1. *The basic model of DDoS attack*

The DDoS attacks are categories are [5, 12] :

- Volumetric Attacks :
  The network bandwidth is targeted in this type of attacks.
  Example: UDP flooding attack, ICMP flooding attack.
- Protocol Attacks :
  These attacks aim to exhaust the server resources.
  Example: Ping of death.
- Application Layer Attacks :
  These attacks focused to disturb the availability of the applications.
  Example: slow HTTP DDoS attack.

The DDoS attacks may be carried out due to various reasons like proving the ability, attacking the competition, etc. [6]. This work addressed one of the application layer attacks known as slow HTTP DDoS attacks in the cloud.

2. The HTTP Protocol overview and the slow HTTP DDoS attacks. This section explains the basic design of the HTTP protocol overview, the slow HTTP DDoS attacks and its various types.

2.1. The HTTP Protocol Overview. The Hypertext Transfer Protocol (HTTP) is the very well-known application layer protocol being used over the Internet to access web-based applications or services. The HTTP protocol runs on top of the TCP/IP suite of protocols. The World Wide Web (WWW) uses this HTTP protocol to access any web pages on the Internet. Whenever we type any Uniform Resource Locator (URL) on the browser to access the web pages, TCP 3-way handshaking has been done between web client and web server to establish the valid TCP connection, after successful connection establishment, HTTP protocol sends out one or more HTTP request to the corresponding web server and one or more HTTP responses received from the web server. The HTTP Protocol modelling has been captured in figure 2.1.

2.2. Slow HTTP DDoS attacks. Sslow HTTP DDoS attack is an application layer attack, and it incorporates multiple HTTP incomplete requests to the server. This type of attacks tries to exploit the flaws in the design or implementation of the applications. It is very challenging to detect compared to other DDoS attacks due to the following reasons [7, 8]:

- The attack originates from the real host in a genuine way;
- This type of attack consumes a very low bandwidth as a mimicking low latency network;
- This attack focuses on exhausting the resources of the application in a legitimate manner.

2.3. Slow HTTP DDoS attack types. Slow HTTP DDoS attack further divided into three types:

- Slow HTTP Headers attacks (or) Slowloris attacks [9];
- Slow HTTP Body attacks (or) R-U-Dead-Yet attacsk [10];
- Slow HTTP Read attacks [11].

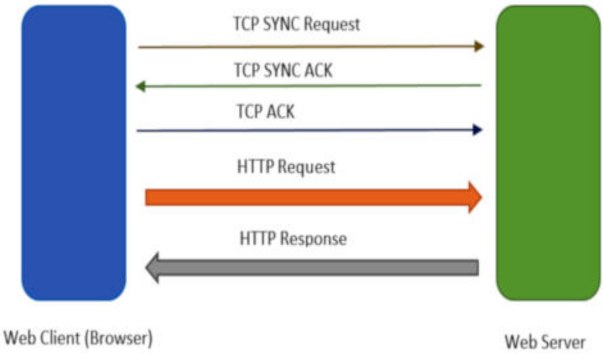
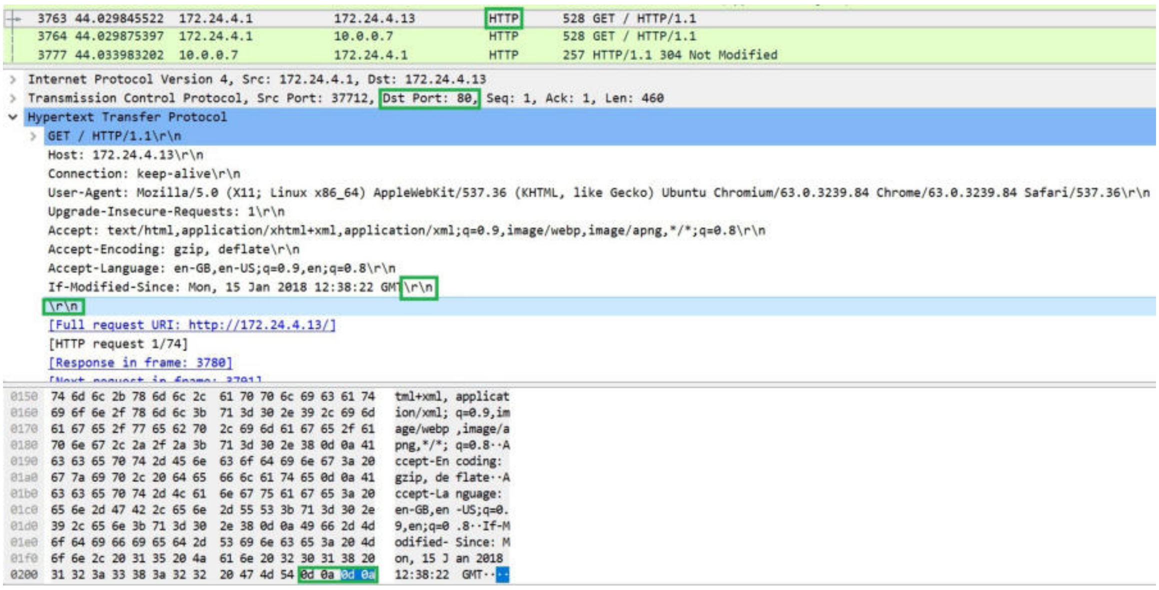


FIG. 2.1. *The HTTP Protocol model*



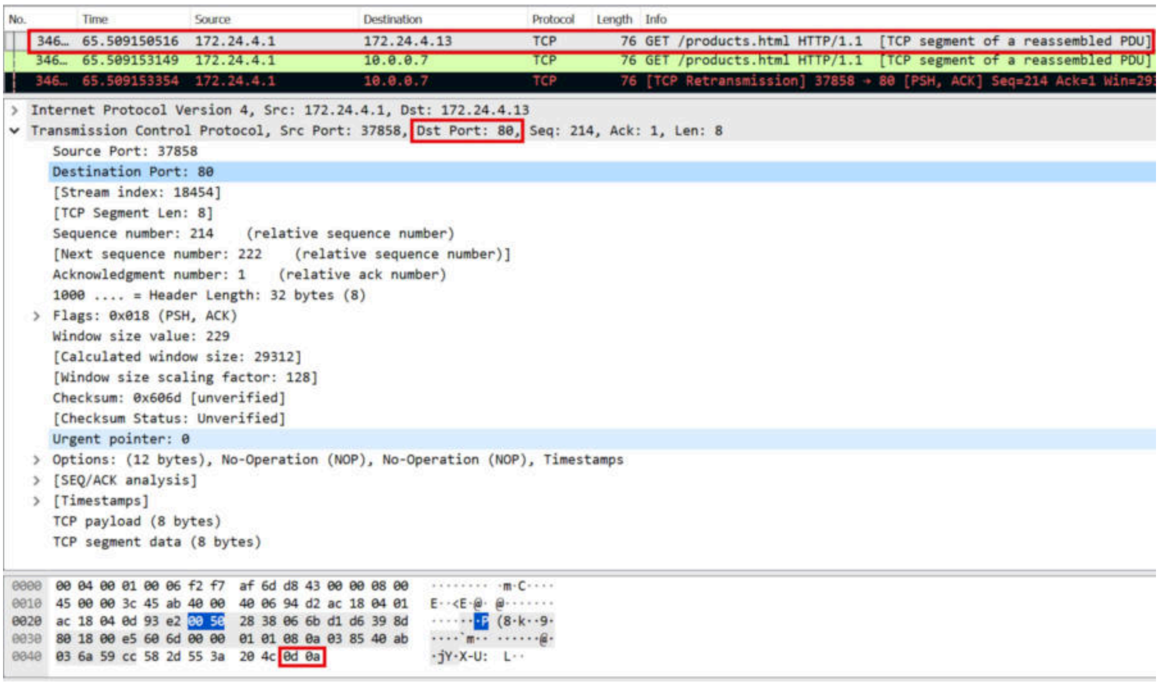
```
3763 44.029845522  172.24.4.1        172.24.4.13       HTTP      528 GET / HTTP/1.1
3764 44.029875397  172.24.4.1        10.0.0.7          HTTP      528 GET / HTTP/1.1
3777 44.033983202  10.0.0.7          172.24.4.1        HTTP      257 HTTP/1.1 304 Not Modified
> Internet Protocol Version 4, Src: 172.24.4.1, Dst: 172.24.4.13
> Transmission Control Protocol, Src Port: 37712, Dst Port: 80, Seq: 1, Ack: 1, Len: 460
v Hypertext Transfer Protocol
  > GET / HTTP/1.1\r\n
    Host: 172.24.4.13\r\n
    Connection: keep-alive\r\n
    User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Ubuntu Chromium/63.0.3239.84 Chrome/63.0.3239.84 Safari/537.36\r\n
    Upgrade-Insecure-Requests: 1\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: en-GB,en-US;q=0.9,en;q=0.8\r\n
    If-Modified-Since: Mon, 15 Jan 2018 12:38:22 GMT\r\n
    \r\n
    [Full request URI: http://172.24.4.13/]
    [HTTP request 1/74]
    [Response in frame: 3780]
    [Next request in frame: 3791]
0150  74 6d 6c 2b 78 6d 6c 2c  61 70 70 6c 69 63 61 74   tml+xml, applicat
0160  69 6f 6e 2f 78 6d 6c 3b  71 3d 30 2e 39 2c 69 6d   ion/xml; q=0.9,im
0170  61 67 65 2f 77 65 62 70  2c 69 6d 61 67 65 2f 61   age/webp ,image/a
0180  70 6e 67 2c 2a 2f 2a 3b  71 3d 30 2e 38 0d 0a 41   png,*/*; q=0.8··A
0190  63 63 65 70 74 2d 45 6e  63 6f 64 69 6e 67 3a 20   ccept-En coding:
01a0  67 7a 69 70 2c 20 64 65  66 6c 61 74 65 0d 0a 41   gzip, de flate··A
01b0  63 63 65 70 74 2d 4c 61  6e 67 75 61 67 65 3a 20   ccept-La nguage:
01c0  65 6e 2d 47 42 2c 65 6e  2d 55 53 3b 71 3d 30 2e   en-GB,en -US;q=0.
01d0  39 2c 65 6e 3b 71 3d 30  2e 38 0d 0a 49 66 2d 4d   9,en;q=0 .8··If-M
01e0  6f 64 69 66 69 65 64 2d  53 69 6e 63 65 3a 20 4d   odified- Since: M
01f0  6f 6e 2c 20 31 35 20 4a  61 6e 20 32 30 31 38 20   on, 15 J an 2018
0200  31 32 3a 33 38 3a 32 32  20 47 4d 54 0d 0a 0d 0a   12:38:22  GMT····
```

FIG. 2.2. *Normal HTTP request header*

2.3.1. Slow HTTP Headers attacks or Slowloris attacks. In general, the during HTTP GET, the web browser sends HTTP GET requests along with HTTP header. The HTTP protocol designed in such a way that the web server waits until to receive the complete HTTP header for processing the GET requests. Whenever the web server receives an incomplete header, it believes that the requests are being sent from the slow network and keep waiting for the complete header. The slow HTTP header attack exploits this behaviour and sends requests with incomplete HTTP headers. Since the attacker never sends complete header information, the server keeps reserving the allocated resource. This result in the starvation of resources and failure to serve any legitimate requests.

The conventional HTTP GET request header looks as shown in figure 2.2. The header always ends with "CR LF CR LF" (0d 0a 0d 0a). During the slow HTTP header attack scenario, the attacker sends only "CR LF" (0d 0a) to indicate the incomplete header. The same has been depicted in figure 2.3.

2.3.2. Slow HTTP Body attacks or R-U-Dead-Yet attacks. This attack comprises HTTP POST requests with the complete header, but the length field contains a large value so that the server reserves all the allocated resources until to receive the entire message.

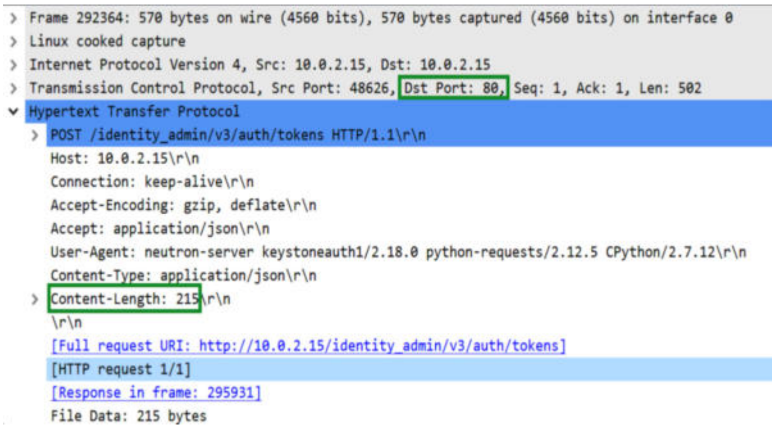


```
No.     Time            Source         Destination     Protocol  Length  Info
346…    65.509150516    172.24.4.1     172.24.4.13     TCP           76  GET /products.html HTTP/1.1  [TCP segment of a reassembled PDU]
346…    65.509153149    172.24.4.1     10.0.0.7        TCP           76  GET /products.html HTTP/1.1  [TCP segment of a reassembled PDU]
346…    65.509153354    172.24.4.1     10.0.0.7        TCP           76  [TCP Retransmission] 37858 → 80 [PSH, ACK] Seq=214 Ack=1 Win=29

> Internet Protocol Version 4, Src: 172.24.4.1, Dst: 172.24.4.13
v Transmission Control Protocol, Src Port: 37858, Dst Port: 80, Seq: 214, Ack: 1, Len: 8
    Source Port: 37858
    Destination Port: 80
    [Stream index: 18454]
    [TCP Segment Len: 8]
    Sequence number: 214    (relative sequence number)
    [Next sequence number: 222    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    1000 .... = Header Length: 32 bytes (8)
  > Flags: 0x018 (PSH, ACK)
    Window size value: 229
    [Calculated window size: 29312]
    [Window size scaling factor: 128]
    Checksum: 0x606d [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  > Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  > [SEQ/ACK analysis]
  > [Timestamps]
    TCP payload (8 bytes)
    TCP segment data (8 bytes)

0000  00 04 00 01 00 06 f2 f7  af 6d d8 43 00 00 08 00   ········ ·m·C····
0010  45 00 00 3c 45 ab 40 00  40 06 94 d2 ac 18 04 01   E··<E·@· @·······
0020  ac 18 04 0d 93 e2 00 50  28 38 06 6b d1 d6 39 8d   ·······P (8·k··9·
0030  80 18 00 e5 60 6d 00 00  01 01 08 0a 03 85 40 ab   ····`m·· ······@·
0040  03 6a 59 cc 58 2d 55 3a  20 4c 0d 0a               ·jY·X-U:  L··
```

FIG. 2.3. *Malicious HTTP header during the attack scenario*



```
> Frame 292364: 570 bytes on wire (4560 bits), 570 bytes captured (4560 bits) on interface 0
> Linux cooked capture
> Internet Protocol Version 4, Src: 10.0.2.15, Dst: 10.0.2.15
> Transmission Control Protocol, Src Port: 48626, Dst Port: 80, Seq: 1, Ack: 1, Len: 502
v Hypertext Transfer Protocol
  > POST /identity_admin/v3/auth/tokens HTTP/1.1\r\n
    Host: 10.0.2.15\r\n
    Connection: keep-alive\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept: application/json\r\n
    User-Agent: neutron-server keystoneauth1/2.18.0 python-requests/2.12.5 CPython/2.7.12\r\n
    Content-Type: application/json\r\n
  > Content-Length: 215\r\n
    \r\n
    [Full request URI: http://10.0.2.15/identity_admin/v3/auth/tokens]
    [HTTP request 1/1]
    [Response in frame: 295931]
    File Data: 215 bytes
```

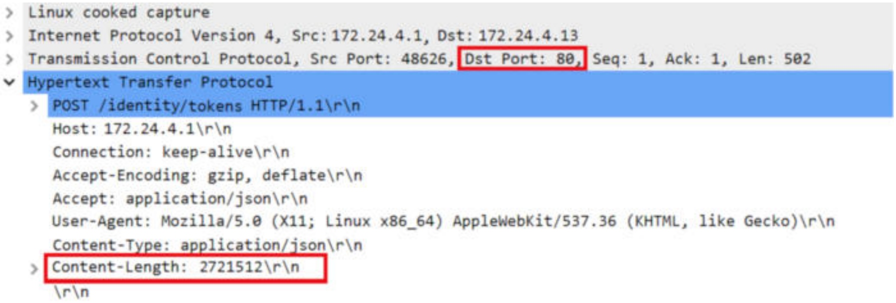
FIG. 2.4. *Normal HTTP Post request*

During this attack, the attacker sends the message with content size as small chunks like one or few bytes in very slow intervals. The attacker also opens multiple such connections to the web server. This way attacker consumes the resources and makes it unavailable to the real users. The regular HTTP POST request depicted in figure 2.4.

The HTTP body attack request captured in figure 2.5, where the length value is notably high.

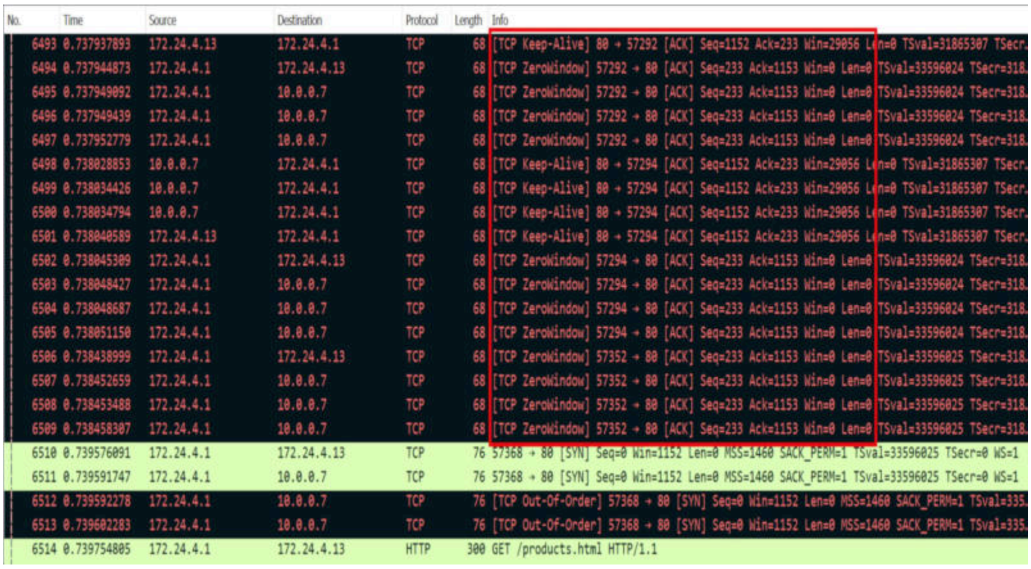
2.3.3. Slow HTTP Read attacks. Another method of slow HTTP attack where the attackers send the HTTP GET requests with the proper header as well as body to the web server. The attack trick is to read the response in a very slow fashion.

Any HTTP requests start with creating the TCP session between the browser and the server. The flow of data between the browser and the web server is controlled by TCP window size value. This value indicates that the number of bytes that the browser can receive on the TCP session. The attacker frequently informs to the



```
> Linux cooked capture
> Internet Protocol Version 4, Src: 172.24.4.1, Dst: 172.24.4.13
> Transmission Control Protocol, Src Port: 48626, Dst Port: 80, Seq: 1, Ack: 1, Len: 502
v Hypertext Transfer Protocol
  > POST /identity/tokens HTTP/1.1\r\n
    Host: 172.24.4.1\r\n
    Connection: keep-alive\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept: application/json\r\n
    User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)\r\n
    Content-Type: application/json\r\n
  > Content-Length: 2721512\r\n
    \r\n
```

FIG. 2.5. *Slow HTTP Body attacks POST request*



```
No.   Time         Source       Destination  Protocol Length Info
6493 0.737937893  172.24.4.13  172.24.4.1   TCP      68 [TCP Keep-Alive] 80 → 57292 [ACK] Seq=1152 Ack=233 Win=29056 Len=0 TSval=31865307 TSecr
6494 0.737944873  172.24.4.1   172.24.4.13  TCP      68 [TCP ZeroWindow] 57292 → 80 [ACK] Seq=233 Ack=1153 Win=0 Len=0 TSval=33596024 TSecr=318
6495 0.737949092  172.24.4.1   10.0.0.7     TCP      68 [TCP ZeroWindow] 57292 → 80 [ACK] Seq=233 Ack=1153 Win=0 Len=0 TSval=33596024 TSecr=318
6496 0.737949439  172.24.4.1   10.0.0.7     TCP      68 [TCP ZeroWindow] 57292 → 80 [ACK] Seq=233 Ack=1153 Win=0 Len=0 TSval=33596024 TSecr=318
6497 0.737952779  172.24.4.1   10.0.0.7     TCP      68 [TCP ZeroWindow] 57292 → 80 [ACK] Seq=233 Ack=1153 Win=0 Len=0 TSval=33596024 TSecr=318
6498 0.738028853  10.0.0.7     172.24.4.1   TCP      68 [TCP Keep-Alive] 80 → 57294 [ACK] Seq=1152 Ack=233 Win=29056 Len=0 TSval=31865307 TSecr
6499 0.738034426  10.0.0.7     172.24.4.1   TCP      68 [TCP Keep-Alive] 80 → 57294 [ACK] Seq=1152 Ack=233 Win=29056 Len=0 TSval=31865307 TSecr
6500 0.738034794  10.0.0.7     172.24.4.1   TCP      68 [TCP Keep-Alive] 80 → 57294 [ACK] Seq=1152 Ack=233 Win=29056 Len=0 TSval=31865307 TSecr
6501 0.738040589  172.24.4.13  172.24.4.1   TCP      68 [TCP Keep-Alive] 80 → 57294 [ACK] Seq=1152 Ack=233 Win=29056 Len=0 TSval=31865307 TSecr
6502 0.738045309  172.24.4.1   172.24.4.13  TCP      68 [TCP ZeroWindow] 57294 → 80 [ACK] Seq=233 Ack=1153 Win=0 Len=0 TSval=33596024 TSecr=318
6503 0.738048427  172.24.4.1   10.0.0.7     TCP      68 [TCP ZeroWindow] 57294 → 80 [ACK] Seq=233 Ack=1153 Win=0 Len=0 TSval=33596024 TSecr=318
6504 0.738048687  172.24.4.1   10.0.0.7     TCP      68 [TCP ZeroWindow] 57294 → 80 [ACK] Seq=233 Ack=1153 Win=0 Len=0 TSval=33596024 TSecr=318
6505 0.738051150  172.24.4.1   10.0.0.7     TCP      68 [TCP ZeroWindow] 57294 → 80 [ACK] Seq=233 Ack=1153 Win=0 Len=0 TSval=33596024 TSecr=318
6506 0.738438999  172.24.4.1   172.24.4.13  TCP      68 [TCP ZeroWindow] 57352 → 80 [ACK] Seq=233 Ack=1153 Win=0 Len=0 TSval=33596025 TSecr=318
6507 0.738452659  172.24.4.1   10.0.0.7     TCP      68 [TCP ZeroWindow] 57352 → 80 [ACK] Seq=233 Ack=1153 Win=0 Len=0 TSval=33596025 TSecr=318
6508 0.738453488  172.24.4.1   10.0.0.7     TCP      68 [TCP ZeroWindow] 57352 → 80 [ACK] Seq=233 Ack=1153 Win=0 Len=0 TSval=33596025 TSecr=318
6509 0.738458307  172.24.4.1   10.0.0.7     TCP      68 [TCP ZeroWindow] 57352 → 80 [ACK] Seq=233 Ack=1153 Win=0 Len=0 TSval=33596025 TSecr=318
6510 0.739576091  172.24.4.1   172.24.4.13  TCP      76 57368 → 80 [SYN] Seq=0 Win=1152 Len=0 MSS=1460 SACK_PERM=1 TSval=33596025 TSecr=0 WS=1
6511 0.739591747  172.24.4.1   10.0.0.7     TCP      76 57368 → 80 [SYN] Seq=0 Win=1152 Len=0 MSS=1460 SACK_PERM=1 TSval=33596025 TSecr=0 WS=1
6512 0.739592278  172.24.4.1   10.0.0.7     TCP      76 [TCP Out-Of-Order] 57368 → 80 [SYN] Seq=0 Win=1152 Len=0 MSS=1460 SACK_PERM=1 TSval=335
6513 0.739602283  172.24.4.1   10.0.0.7     TCP      76 [TCP Out-Of-Order] 57368 → 80 [SYN] Seq=0 Win=1152 Len=0 MSS=1460 SACK_PERM=1 TSval=335
6514 0.739754805  172.24.4.1   172.24.4.13  HTTP    300 GET /products.html HTTP/1.1
```

FIG. 2.6. *Slow HTTP Read Attacks*

web server that it can only receive few bytes, or it is not ready to receive any bytes. The attacker opens many such sessions with the server to consume all the resource so that the server will not be able to process any real requests from the legitimate user. The slow HTTP Read attack depicted in figure 2.6.

The rest of the paper organised as sec. 3 covers the literature review of the related works. The proposed architectural model explained in sec. 4. Section 5 captures cloud environmental details. This work is improved version of our earlier work done to detect slow HTTP DDoS attack detection in the cloud environment in [22].

The experimentations, results and discussions are part of sec. 6. The conclusion and future enhancements discussed in sec. 7.

3. Literature Review of the related works. Though the authors reviewed many works related to HTTP DDoS attacks, restricting to add only the most relevant work in this section.

Enrico Cambiaso et al. [13] reviewed the slow DDoS attacks in detail. The authors defined each of the attacks and categorized into different classes. This paper provides an insight into the various slow DDoS attacks.

Thomas Lukaseder et al. [14] used the SDN controller to realise the resolution against the slow HTTP attacks. The solution implemented in three phases. The first phase is the detection, the second phase is the identification, and the final one is the mitigation phase. During the detection phase, the application tries to reach the webserver. The moment the server is not reachable, then it assumes that the server is down and starts the identification phase. In the identification phase, they calculate the low packet rate, packet distance uniformity, etc., to find out the attacks and in the final phase, block those attacks using the SDN controller.

Aanshi Bhardwaj et al. [15] analyzed and evaluated the DDoS attack using the OpenStack cloud platform and captures the attack Wireshark graph. This article studied only the DDoS attack using various tools. This paper has gaps in apprehending their OpenStack details like instances, network topology and providing the solution for detection or mitigation in the cloud environment.

Tetsuya Hirakawa et al. [16] proposed a generic defence mechanism against the slow HTTP DDoS attacks. The attacks identified using the number of connections and their life span to the server. The solution needs to be extended for the cloud environment to detect and mitigate the same.

Kiwon Hong et al. [17] defined the slow HTTP DDoS attack solution using the SDN approach. This method implements the slow HTTP DDoS defence module in the SDN controller. The flow through the switches from the browser to the server and from the server to clients are input to the defence application. This assumes three types of a request such as attacks, slow browser and real clients. The attack flows are identified and blocked. This is a generic solution and has to be enhanced for the cloud environment.

Shunsuke Tayama et al. [18] investigated the slow Read DoS attack to the server, and it considers the bandwidth rate as the essential parameter to identify the attacks. The paper concluded that the bandwidth above 500Kbps and having the connection rate equal to the server processing capability is sufficient to conduct the more efficient slow read attacks.

Suroto [19] discussed the different slow HTTP DDoS attacks. This work explored the various configuration parameters fine-tuning for different web servers like Apache, NGINX to mitigate such slow HTTP attacks.

Junhan Park et al. [20] evaluated the effectiveness of the slow Read in a virtual environment. The conclusion of adding the security measures to the webserver is capable to defend the attacks from a single machine (DoS), but not sufficient for the distributed DoS attacks.

A. Dhanapal et al. [21] proposed an OpenStack cloud testbed model for evaluating the DDoS attacks in the cloud environment. This work explored various attack possibilities for the application running in the cloud environment.

The proposed work is enhanced over our previous works in [21] and [22] to detect, mitigate and prevent slow HTTP DDoS attacks in the cloud environment. The novelty contribution of this works to define the multi-stage zonal classification model integrated into the cloud to detect and prevent the slow HTTP DDoS attack. The multi-stage zonal classification architecture discussed in detail in the upcoming section.

4. The Proposed Architecture Model. The slow HTTP DDoS attacks can happen in multiple ways to the web server in the cloud environment. The attacks in the cloud classified as:

- Slow HTTP attacks within the cloud environment
- Slow HTTP attacks from the outside/internet

4.1. Slow HTTP attacks within the cloud environment. The internal HTTP DDoS attacks originate from the instances running within the cloud environment itself. The target victim web server runs within the same cloud environment. The attacks within the cloud are architecturally represented as in figure 4.1.

4.2. Slow HTTP attacks from outside of the cloud or internet. This type of attack is launched from the internet and targeting to the webserver running in the cloud. The architecture model of the attacks from the external environment represented in figure 4.2.

4.3. Proposed Solution. The offered solution addresses both the internal and external slow HTTP DDoS attacks in the cloud environment. This solution is integrated into cloud infrastructure to safeguard webserver from all possible attack situations. The request from any client to the webserver is categorized into any one of the zone states at any given point in time using the multi-stage zonal model. The multi-stage zonal classification architecture model is shown in figure 4.3.

Initially, all the incoming connection requests monitored for their activity in the monitoring state zone. The legitimate behaviour clients moved into the green state zone. In the green state zone, all the client requests forwarded to the webserver.

In case, the client activities are suspicious like opening higher number of connections and sending frequent GET requests with the incomplete header or POST requests with a smaller number of bytes with longer interval duration to its turn-around time are moved to orange state zone and continue to be monitored.

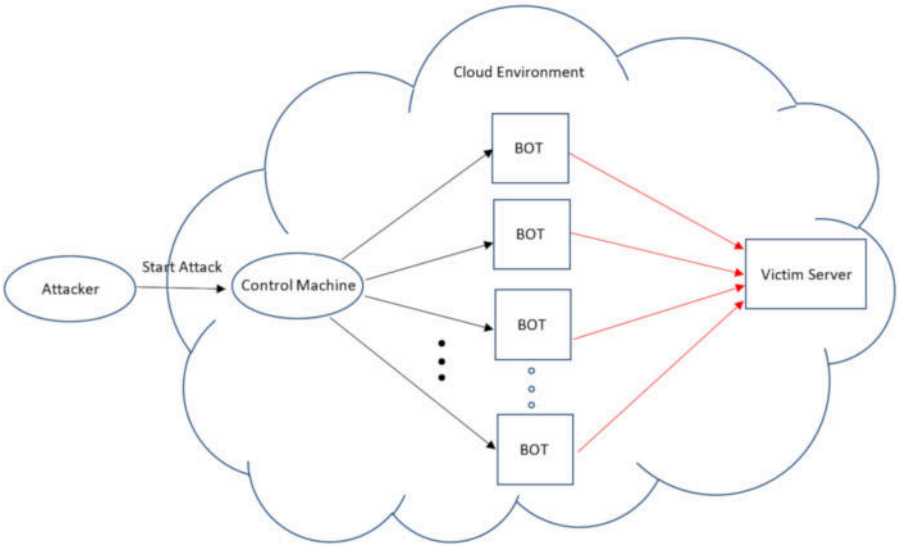


Fig. 4.1. *The DDoS attacks architecture within the cloud environment*

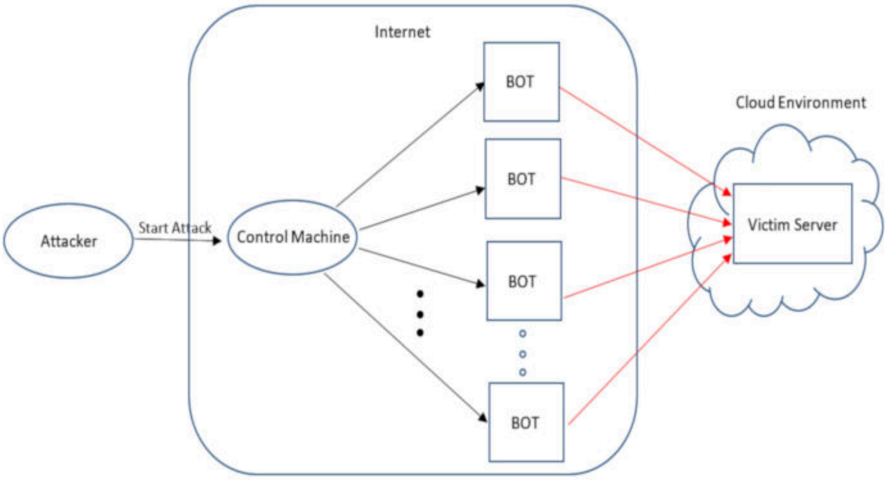


Fig. 4.2. *Architecture Model of external DDoS attacks to the cloud environment*

The clients in the monitor state zone or green state zone or orange state zone activity looks abnormal such as opening very high number of connection requests than the allowed maximum number of connections and every request received with partial header or post request with few bytes or request to server with very high interval like more than 80% of the defined keep alive interval when compared to its turn-around time are treated as DDoS attacks and those clients are put into the red state zone. Any request coming from the clients in the red state zone is blocked.

The clients in the green state zone or orange state zone or red state zone with no activity or no active connection request are moved back to the monitor state zone.

The green state zone client exhibits the suspicious activities persistently is moved into the orange state zone, and similarly, the client shows normal activities after entering orange state zone consistently for every request is placed into the green state zone.

The multi-state zonal model has a provision to limit the allowed number of connections per client and the number of ping requests to calculate the average real network latency value. The configuration file Zone.conf

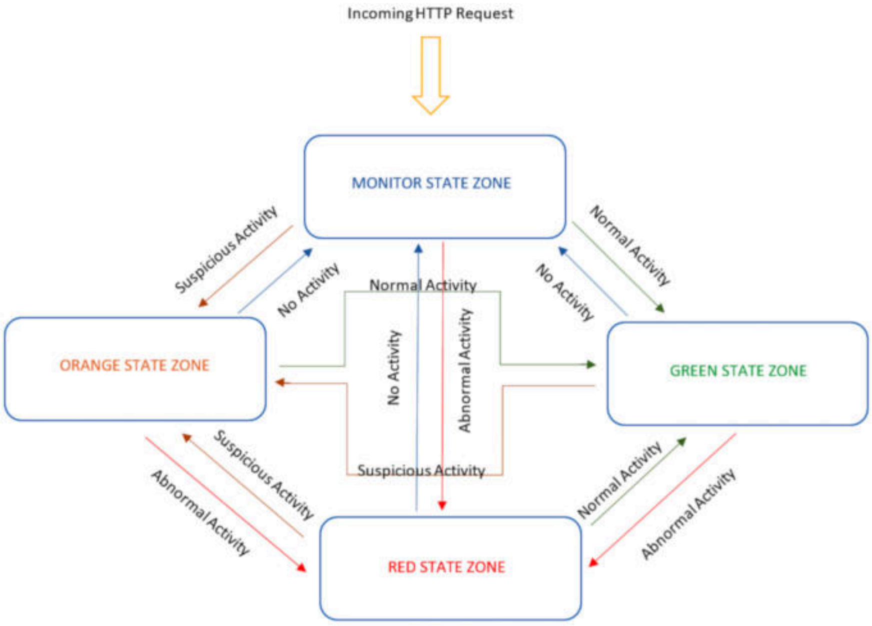
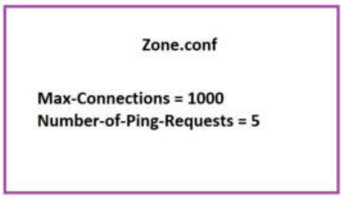


Fig. 4.3. *The multi-stage zonal classification architecture*



```
Zone.conf

Max-Connections = 1000
Number-of-Ping-Requests = 5
```

Fig. 4.4. *Configuration file for the multi-stage zonal classifier model*

used for the same. The default value of the web server configuration utilised if the configuration value or file is not present. The Zone.conf looks like as shown in figure 4.4.

The zonal model uses the maximum number of connections allowed per client, the average real network latency (ARNL) of the client reachability are parameters to decide the slow HTTP DDoS attack to the webserver.

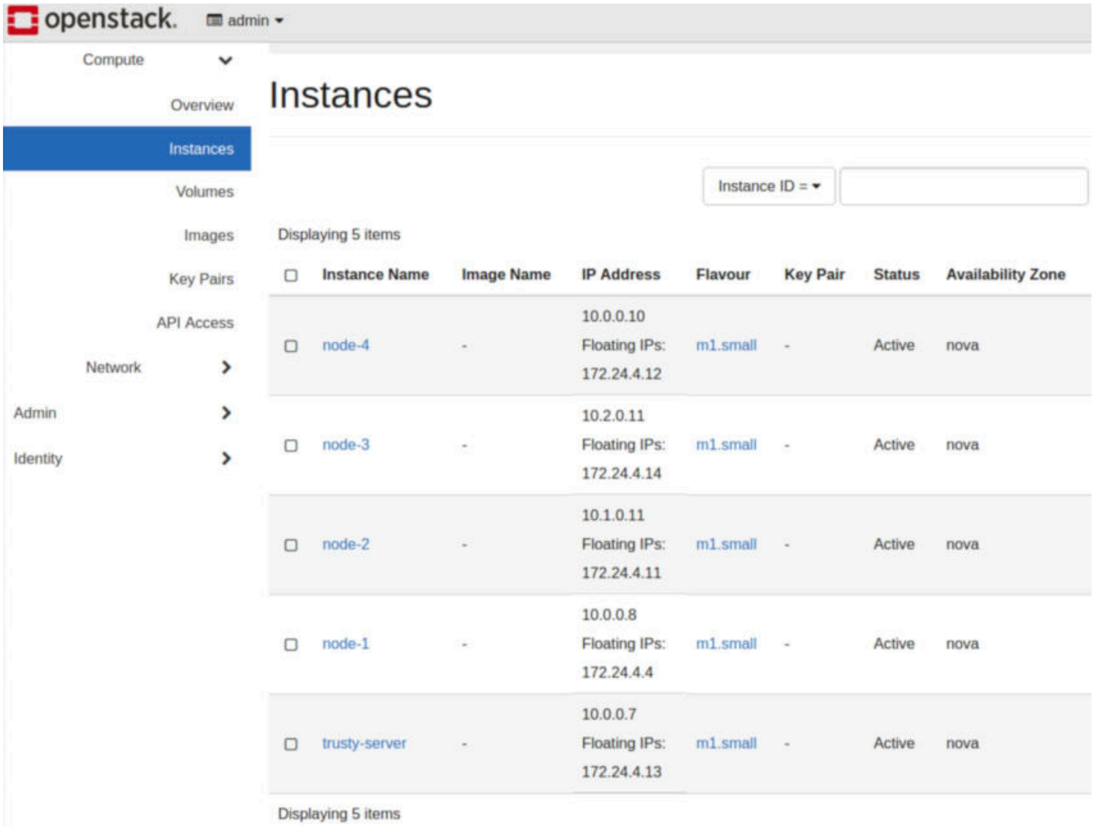
The clients try to open more than 70% of the allowed connections are closely monitored. The proposed model sends 5 ping requests to the client. The response to the ping used to calculate the average network latency.

The ARNL value is compared with the clients average interval request time. The attackers mimicking the slow network and the client with the real slow network are identified correctly using this technique. The clients with slow requests exceed ten times of its ARNL is considered as an attack and moved into the red state zone. The value ten is chosen by considering the application processing time.

The clients with 61 to 70% of the allowed connection requests and the average request time delay between 6 to 9 times of its ARNL moved into the orange state zone. The clients in this state remain monitored and classified accordingly based on analyzing behaviour continuously.

The clients with the connection rate between 40 to 60% and the average delay between requests are between 4 to 6 times of the average network latency are kept in the green state zone.

A newly incoming client is kept into the monitor state zone to identify the activity pattern and classified into any one of the states based on its behaviour defined above.



Instances

Displaying 5 items

□	Instance Name	Image Name	IP Address	Flavour	Key Pair	Status	Availability Zone
□	node-4	-	10.0.0.10 Floating IPs: 172.24.4.12	m1.small	-	Active	nova
□	node-3	-	10.2.0.11 Floating IPs: 172.24.4.14	m1.small	-	Active	nova
□	node-2	-	10.1.0.11 Floating IPs: 172.24.4.11	m1.small	-	Active	nova
□	node-1	-	10.0.0.8 Floating IPs: 172.24.4.4	m1.small	-	Active	nova
□	trusty-server	-	10.0.0.7 Floating IPs: 172.24.4.13	m1.small	-	Active	nova

Displaying 5 items

FIG. 4.5. *The OpenStack instance details*

The abnormal activity is defined as the client trying to open high number of connection than the maximum allowed number of connections and every request received with partial header or post request with few bytes or request to server with very high interval like more than 80% of the defined keep alive interval when compared to its real turn-around time.

The suspicious activity is defined as the clients trying to open 61 to 70% of the allowed connection requests and keep-alive as close to 71 to 80% of the predefined value or with the average request time delay between 6 to 9 times of its real the average network latency.

4.4. The Cloud Environmental Details . The proposed model implemented using the OpenStack cloud environment. The OpenStack is open- source software available freely over the internet and providing Infrastructure as a Service cloud service model. The cloud environment runs multiple virtual instances for different customers to implement multitenancy of the cloud. The experiment environment has four networks namely private, green-private, orange-private and public network. Each of the networks represents different customers. The OpenStack instance details depicted in figure 4.5.

The OpenStack network topology in the experiment showed in figure 4.6. Every OpenStack instance associated with a local/internal IP address for communication within the cloud environment and one public/floating IP address for connecting with the external world via the internet. The internal IP address is in the range of 10.xx.xx.xx network and public network IP starts with 172.24.xx.xx.

The customer in the green-private network runs the NGINX webserver in the instance trusty-server. The internal IP address of the webserver is 10.0.0.7, and the public IP address is 172.24.4.13.

4.5. The Experiments, Results and Discussion . The slowHTTPtest tool used for generating different type attacks like slow HTTP header, slow HTTP body and slow HTTP read DDoS attack.

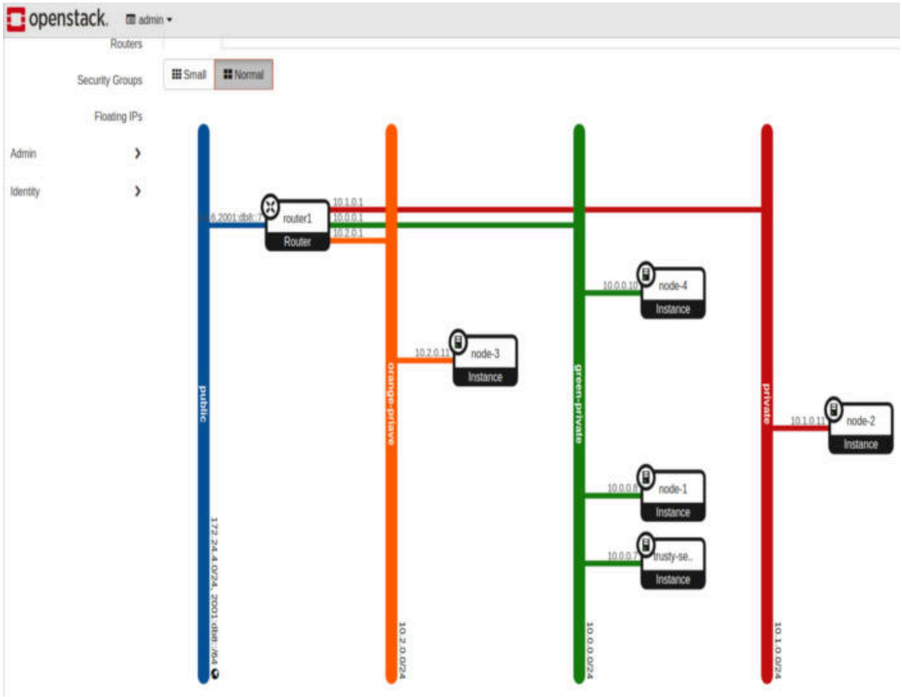


Fig. 4.6. *The OpenStack network topology*

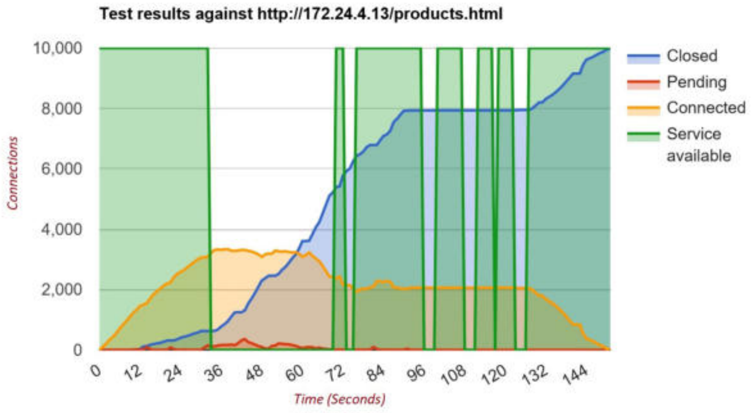


Fig. 4.7. *The result of slow HTTP header attack with NGINX default configuration*

Initially, the authors launched the slow HTTP attacks to the webserver without any changes in the webserver configuration. This experiment did not have the slow DDoS protection layer implementation as well.

The study of the results are as follows:

> The slow HTTP header attack launched against the webserver with
> URL: http://172.24.4.13/products.html.

The parameters used for these attacks and results captured in figure 4.7 and different values used for this testing are:

- The total number of connections: 10000;
- The interval between follow-up data : 10 seconds;
- The number of connection requests: 500 request/seconds.

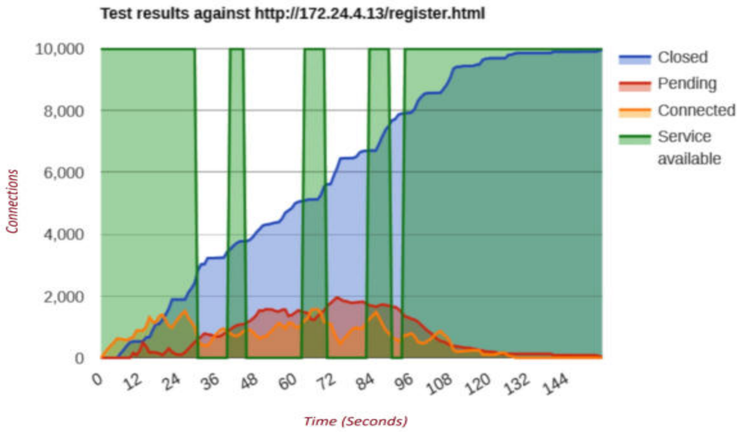


Fig. 4.8. *The result of slow HTTP body attack with NGINX default configuration*

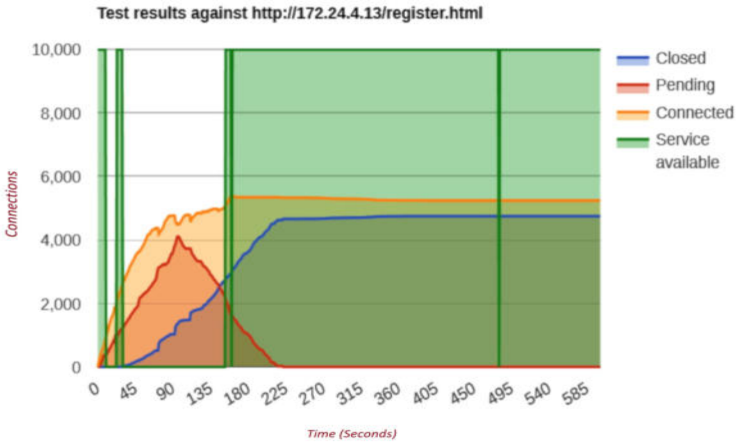


Fig. 4.9. *The result of slow HTTP read attack with NGINX default configuration*

The results of this experiment show that a slow header attack with the total number of connections above 3000 can bring down the service completely. The webserver is will not be available to process any of the legitimate requests.

The number of connections closed during this point of time is 500. The pending requests are 150. Once the service becomes unavailable, the connection requests are dropped by the server. This can be seen with the line indicating connected in figure 4.7.

The slow HTTP body attack test parameter values used are same as the slow HTTP Header attacks.

The slow HTTP body attack carried on the webserver against
URL: http://172.24.4.13/register.html.

The result along with parameters are showed in figure 4.8.

From figure 4.8, the service unavailability is frequently seen at multiple intervals of time.

The connection closed is increased in a continuous manner as shown by closed line in figure 4.8. The connected request is toggling and this is seen from connected line of figure 4.8.

The slow HTTP read attack launched against
URL: http://172.24.4.13/register.html.

The test results depicted in figure 4.9.

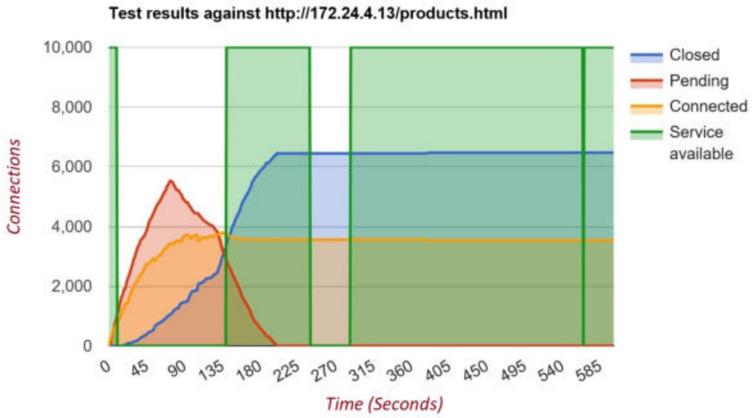


Fig. 4.10. *The slow HTTP header attack results with server fine-tuned configuration*

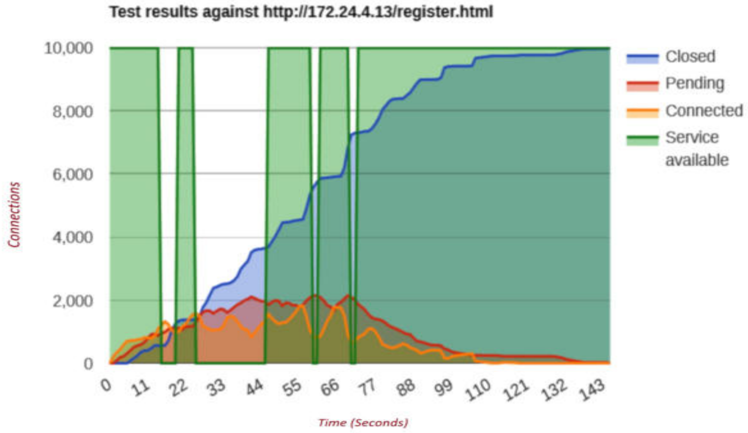


Fig. 4.11. *The slow HTTP body attack results with server fine-tuned configuration*

The testing parameters used are 10000 number of connections made with the rate of 500 connection request per second. The rate at which request processed are 5 bytes per second. From figure 4.9, the server can withstand till around 2000 connections, beyond which failed to serve any requests. The number of pending requests are increased very sharply as depicted in pending line of figure 4.9.

There is a solution discussed as part of [19] to tighten the web server configuration to mitigate the slow HTTP DDoS attacks. The suggested configuration has been done to the webserver to understand the results.

The same above-mentioned experiments are carried out with recommended web server configurations. The result shows that the configuration helps some extent to mitigate the attack and still there is an excellent scope for improvements.

The service failures detected when the connection number reached 1000 for the slow HTTP header attack shown in figure 4.10.

The slow HTTP body attack also shows the server failure in two intervals of the period when the connection reached above 1500. Figure 4.11 captures the observation of this attack. The different values used for this testing are:

- The total number of connections: 10000
- The interval between follow-up data : 10 seconds
- The number of connection requests: 500 request/seconds

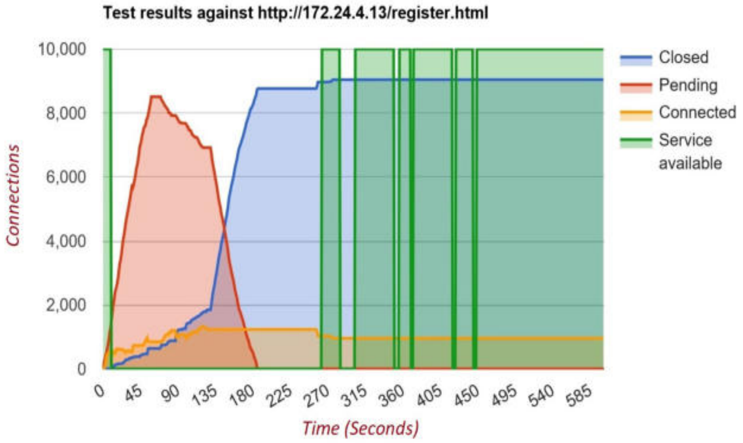


Fig. 4.12. *The slow HTTP read attack results with server fine-tuned configuration*

The slow HTTP read attack with the updated configuration depicted in figure 4.12. The web server brought down and the service unavailable for multiple intervals. The failure started when the connection reached 500.

The receive window size used in the range of 1 to 512 bytes and the read rate buffer size is 5 bytes per second.

All the above experiments conducted it is evident that once the service is back an available as shown by service available line, it is still seen by the attacker for further attacks. The attacker is not identified and blocked in by the cloud. This makes the server vulnerable once it is restored by the administrator.

The same experiment has been done with the implementation of the proposed solution. The results exhibit that with the implemented solution, the webserver can identify the slow HTTP DDoS attacks effectively and prevent such attacks. Whenever the suspicious clients recognised, the multi-stage zone model classifies them to the orange state zone. Upon further observation and understanding of the clients continuous malicious activity, classifying them as attacks and those clients moved to the red state zone. Any further request from the client in the red state zone is blocked in the system. Since none of the newer requests serviced, the attacker thinks that the webserver is down, and attack is successful. In reality, those attackers are classified by the proposed solution and blocked to reach the webserver to make sure that the server is available to the legitimate requests.

The experiment result of the various attacks with the proposed solution is displayed below:

The slow HTTP header attack against the webserver with
URL: http://172.24.4.13/products.html with the implemented solution as seen in figure 4.13.

The total number of connections request used in this experiment is 10000. The follow-up interval of data sent with 10 second interval and the number of requests per second is 500.

The service is seen as unavailable by the attacker when the connection reached 400. At this point of time the attacker is identified and blocked by the implemented solution. Since the attacker is blocked by the solution and none of the requests are further serviced. This makes the attacker to think that server is down. The same has been understood from service available line is plotted against to zero figure 4.13.

The slow HTTP body attack has been launched against the web page http://172.24.4.13/register.html. The total number of connection request made are 10000 and the number of connections per second is 500. The follow-up data send with the interval of 10 seconds to make the connections alive.

The result is shown in figure 4.14. The implemented protection layer is detected the body attack at the early stage when the number of connections reached 1000 by the attacker. The attacker has been successfully blocked by the system. This results in server unavailable to the attacker and continued to unavailable as the service available line is plotted as zero in figure 4.14. But, the attack is identified, and attacker is blocked by the system.

The slow HTTP read attack is launched for the webpage http://172.24.4.13/register.html. The number of

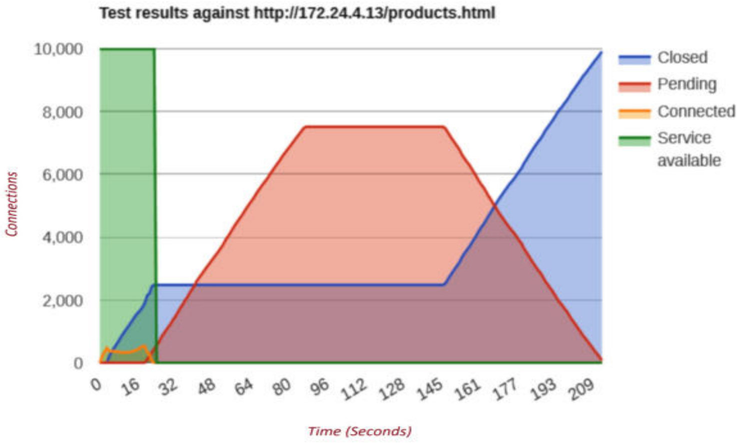


Fig. 4.13. *The slow HTTP header attack results with the proposed solution*

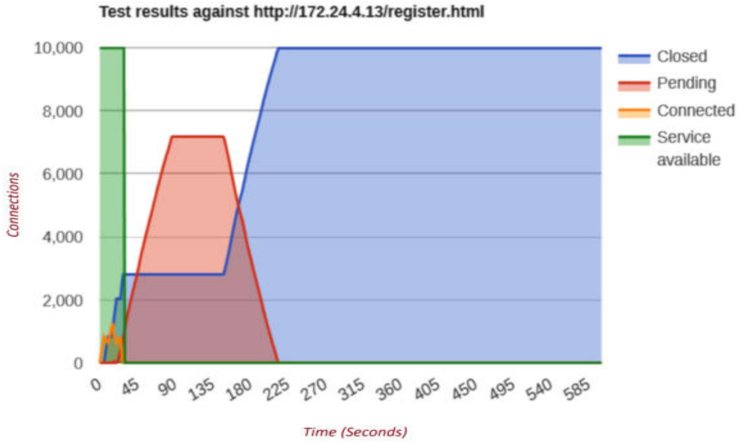


Fig. 4.14. *The slow HTTP body attack results with the proposed solution*

bytes processed per second is 5 bytes in the read window. Thus, makes the read attack to the webserver to consume the resources.

The service is unavailable as soon as system detects the attack during when the number of connections request reached 1200 as shown in figure 4.15. Upon detected, the attacker is moved into red zone and further request are not serviced. This is the reason for service available line is plotted against zero by the attacker. The webserver is available and continue to service the legitimate requests.

The several experiments carried with default web server configurations and the suggested fine-tuned configurations are not adequate to identify and mitigate the slow HTTP DDoS attacks. The proposed solution can identify and prevent those attacks efficiently.

4.6. Comparison of related works with the proposed work. Table 4.1 compares the various work done so far in the space of slow HTTP DDoS attacks detection, mitigation and prevention. This research work covers all the aspects and the gaps identified with the existing works.

5. Conclusion and Future Enhancements. In this paper, we have discussed slow HTTP DDoS attacks and several forms of such attacks in details. The slow HTTP attack scenarios in the cloud are discussed. The solution is proposed to detect and prevent such attacks in the cloud environment. The offered solution is implemented using the open-source OpenStack Infrastructure as a Service model. Each type of the slow HTTP

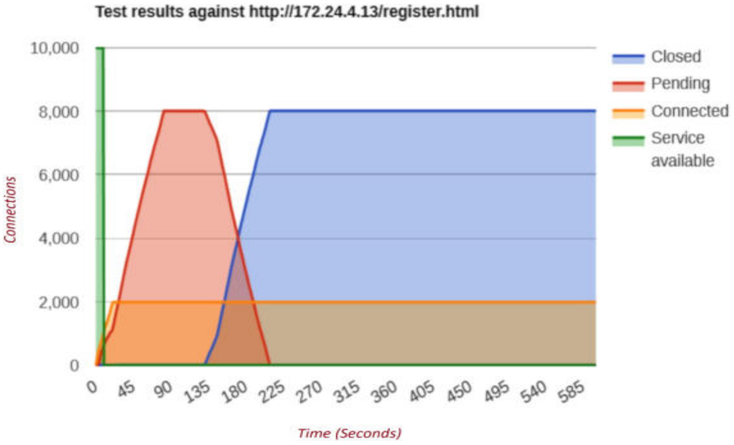


FIG. 4.15. *The slow HTTP read attack results with the proposed solution*

attacks carried out using slowHTTPTest tool against NGINX webserver running in the OpenStack cloud. The experiments with the default NGINX settings and the fine-tuned configurations are done to understand the results. The observation is that the web server fine-tuning is not enough to detect and mitigate attacks. The experiments performed with the recommended solution exhibits excellent results as it identifies the slow HTTP attack and prevents effectively. The corresponding experiments details and results are discussed.

The future enhancement is to generate slow HTTP attacks using various available tools in the cloud environment to study the result of the offered solution and improve the rate of success.

REFERENCES

[1] SALESFORCE, *https://www.salesforce.com/uk/blog/2015/11/why-move-to-the-cloud-10-benefits-of-cloud-computing.html*, 09-Jan-2019.

[2] NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY (NIST), *https://www.nist.gov/programs-projects/nist-cloud-computing-program-nccp*, 09-Jan-2019.

[3] WHATISCLOUD.COM, *http://whatiscloud.com/cloud_delivery_models/index*, 09-Jan-2019.

[4] WHATISCLOUD.COM, *http://whatiscloud.com/cloud_deployment_models/index*, 09-Jan-2019.

[5] IMPERVA INC., *https://www.incapsula.com/blog/top-10-cloud-security-concerns.html*, 09-Jan-2019.

[6] SIMPLICABLE, *http://arch.simplicable.com/arch/new/the-5-motives-for-DDoS-attack*, 09-Jan-2019.

[7] ACUNETIX, *https://www.acunetix.com/blog/articles/need-pay-attention-slow-http-attack/*, 09-Jan-2019.

[8] QUALYS,*https://blog.qualys.com/securitylabs/2011/07/07/identifying-slow-http-attack-vulnerabilities-on-web-applications*, 09-Jan-2019.

[9] CLOUDFARE, LU-*https://www.cloudflare.com/learning/ddos/ddos-attack-tools/slowloris/*, 09-Jan-2019.

[10] CLOUDFARE, *https://www.cloudflare.com/learning/ddos/ddos-low-and-slow-attack/* , 09-Jan-2019.

[11] CLOUDFARE, *https://www.cloudflare.com/learning/ddos/ddos-attack-tools/r-u-dead-yet-rudy/*, 09-Jan-2019.

[12] RADWARE, *https://security.radware.com/ddos-knowledge-center/ddospedia/dos-attack/*, 09-Jan-2019.

[13] ENRICO CAMBIASO, GIANLUCA PAPALEO, GIOVANNI CHIOLA, AND MAURIZIO AIELLO, *PSlow DoS attacks: definition and categorisation*, Int. J. Trust Management in Computing and Communications, Vol. 1, Nos. 3/4, pp. 300–319, (2011).

[14] THOMAS LUKASEDER, LISA MAILE, BENJAMIN ERB, AND FRANK KARGL, *SDN-Assisted Network- Based Mitigation of Slow DDoS Attacks*, SecureComm18, arXiv:1804.06750 [cs.CR], (2018).

[15] AANSHI BHARDWAJ, ATUL SHARMA, VEENU MANGAT, KRISHAN KUMAR AND RENU VIG , *Experimental Analysis of DDoS Attacks on OpenStack Cloud Platform* , Proceedings of 2nd International Conference on Communication, Computing and Networking, Lecture Notes in Networks and Systems 46, (2019).

[16] TETSUYA HIRAKAWA, KANAYO OGURA, BHED BAHADUR BISTA, TOYOO TAKATA, *A Defense Method against Distributed Slow HTTP DoS Attack*, 2016 19th International Conference on Network-Based Information Systems, DOI: 10.1109/NBiS.2016.58, (2016).

[17] KIWON HONG, YOUNJUN KIM, HYUNGOO CHOI, AND JINWOO PARK , *SDN-Assisted Slow HTTP DDoS Attack Defense Method* , IEEE Communications Letters, Vol. 22, Iss. 4 , DOI: 10.1109/LCOMM.2017.2766636, pp. 688–691, (2018).

[18] SHUNSUKE TAYAMA, AND HIDEMA TANAKA , *Analysis of Slow Read DoS Attack and Communication Environment*, Mobile and Wireless Technologies 2017, Lecture Notes in Electrical Engineering 425, DOI 10.1007/978-981-10-5281-1_38, pp. 1718–1724, (2018)

Table 4.1
Comparison of related works with proposed work

Related Works	**Cloud Characteristics Considered**	**Cloud Based Solution**	**Given Solution**	**Gaps Identified**
Enrico Cambiaso et al.	No	No	Definition and Categorization of the slow DDoS Attacks.	This paper gives insight into slow DDoS attacks, but the solution is not defined.
Thomas Lukaseder et al.	No	No	The SDN Based solution.	This is specific to SDN based environment and has to be enhanced for the cloud
Aanshi Bhardwaj et al.	Yes	Yes	This is OpenStack Based slow DDoS attack study using various tools.	The OpenStack cloud instance, network details are not discussed. The detection, mitigation and prevention are not covered.
Tetsuya Hirakawa et al.	No	No	The generic solution defined for slow HTTP DDoS attacks.	The solution needs to be enhanced for cloud environment.
Kiwon Hong et al.	No	No	The SDN based solution for slow HTTP DDoS attacks.	The cloud-specific characteristics and attack paths to be covered to fit into the cloud environment.
Shunsuke Tayama et al.	No	No	Investigated the slow Read DDoS attacks based on Bandwidth.	Only slow Read attacks considered. Slow Header and Body attacks are not covered.
Suroto	No	No	Configuration tuning-based detection and mitigation of slow HTTP DDoS attacks.	The configuration tuning is alone not enough to mitigate and prevent attacks. The same configuration tuning is explored to confirmed further opportunities available to betterment the results.
Junhan Park et al.	Yes	No	This work confirmed adding security measure to the webserver can defend DoS attacks, but not DDoS attacks.	This work needs to be enhanced for DDoS attacks in the cloud environment.
Proposed Work	**Yes**	**Yes**	**Alow HTTP DDoS attack detection, mitigation and prevention in the cloud.**	**This research work focused on all types of slow HTTP DDoS attacks in the cloud environment. Covered all possible attacks paths to safeguard webserver from slow HTTP DDoS attacks.**

[19] Suroto, *A Review of Defense Against Slow HTTP Attack*, , Int. J. on Informatics Visualization, Vol. 1, No. 4, pp. 127–134, (2017).

[20] Junhan Park, Keisuke Iwai, Hidema Tanaka, and Takakazu Kurokawa, *Analysis of Slow Read DoS Attack and Countermeasures on Web servers*, Int. J. of Cyber-Security and Digital Forensics, pp. 339–353. (2015).

[21] A.Dhanapal, and P. Nithyanandam , *An OpenStack based cloud testbed framework for evaluating HTTP flooding attacks*, Wireless Networks - Springer, DOI: 10.1007/s11276-019-01937-4, pp. 570–575, (2019).

[22] A.Dhanapal, and P. Nithyanandam , *The slow HTTP Distributed Denial of Service Attack Detection in Cloud*, Scalable Computing, DOI: https://doi.org/10.12694/scpe.v20i2.1501,Volume 20, Number 2, pp. 285–297, (2019).

[23] Clifford Kemp, Chad Calvert, Taghi M. Khoshgoftaar, *Utilizing Netflow Data to Detect Slow Read Attacks*, 2018 IEEE International Conference on Information Reuse and Integration (IRI), DOI: http://doi.ieeecomputersociety.org/10.1109/IRI.2018.00023, pp. 108–116, (2018).

[24] Tanishka Shorey, Deepthi Subbaiah, Ashwin Goyal, Anuraag Sakxena, Alekha Kumar Mishra, *Performance Comparison and Analysis of Slowloris, GoldenEye and Xerxes DDoS Attack Tools*, 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), DOI: 10.1109/ICACCI.2018.8554590, (2018).

[25] Ghafar A. Jaafar, Shahidan M. Abdullah, and Saifuladli Ismail, *Review of Recent Detection Methods for HTTP DDoS Attack*, Journal of Computer Networks and Communications - Hindawi, DOI: https://doi.org/10.1155/2019/1283472, Volume 2019, Article ID 1283472, 10 pages, (2019).

[26] Trung V. Phan, and Minho Park, *Efficient Distributed Denial-of-Service Attack Defense in SDN-Based Cloud*, IEEE Access, DOI: 10.1109/ACCESS.2019.2896783, (2019)

Edited by: Dana Petcu
Received: Jun 3, 2019
Accepted: Nov 30, 2019

DOI 10.12694/scpe.v20i4.1570
Scalable Computing: Practice and Experience ISSN 1895-1767
Volume 20, Number 4, pp. 687–698. http://www.scpe.org © 2019 SCPE

AN EFFECTIVE CONTEXT SENSITIVE OFFLOADING SYSTEM FOR MOBILE CLOUD ENVIRONMENTS USING SUPPORT VALUE-BASED CLASSIFICATION

MOSTAFA ABDULGHAFOOR MOHAMMED[*] AYA AHKAM KAMIL[†] RAED ABDULKAREEM HASAN[‡] AND NICOLAE ŢĂPUŞ[§]

Abstract. Mobile cloud computing (MCC) has drawn significant research attention recently due to the popularity and capability of portable devices. This paper presents an MCC offloading system based on internet offloading choices. This system guarantees the conservation of battery life and reduced execution time. The proposed effective context sensitive offloading approach using support value-based classification is processed in different steps. Initially, the context data of the input tasks is extracted through the energy consumption model, cost model, execution model, communication model and stored. Then, the support value-based classification approach classifies the tasks based on the context information. This classification creates the information about the tasks and finally, a decision is made at the right time to achieve better offloading. The result indicates the presented offloading framework can choose reasonable cloud assets depending on various contexts of the mobile devices and achieve significant performance enhancement.

Key words: Context data extraction, Support value measure, Classification, Generated knowledge, Offloading decision

AMS subject classifications. 68Q25, 68R10, 68U05

1. Introduction.

Mobile cloud computing is a recent model that integrated cloud computing [29] . It enables the portable clients to use cloud services of their interest. It is imagined that this worldview will help to defeat the restrictions of the cell phones equipment [14, 32]. Offloading is an essential procedure in MCC [5]. It offloads the assignment to a cloud benefit by means of remote systems. Because of the generally low battery limit of cell phones (just as delicate versatile systems), several explorations have been performed on offloading to the cloud to accomplish two principle goal: to extend battery lifetime [5, 16, 7, 27], and to shorten the execution time of substantial uses [28, 9, 24, 13]. Offloading is a significant method in MCC [21, 23] that has been extensively investigated. For instance, in Think Air [16] , versatile side offloading gathers the runtime information and feed them into the energy estimation display. The energy estimation is also actualized inside the Think Air energy profiler on the versatile side. It is used to evaluate the utilization of each running strategy [4]. Like Think Air, many offloading frameworks [1, 8, 2] have executed the profilers, cost forecast models, and choice motor segments on the versatile side.

Numerous works have been done in MCC [17, 10, 19, 20] and most of them focused on the offloading strategies, accepting a steady system and adequate transfer speed. This sort of offloading approaches keeps the offloading choice parts running amid application execution to settle on ideal offloading choices at runtime. Be that as it may, this procedure of settling on an offloading choice delivers a noteworthy overhead on the cell phone. This overhead is expensive on cell phones in terms of battery utilization and handling assets [30, 15].

To process the issues stated above and improve the service performance in mobile cloud computing, we propose an effective context sensitive offloading approach using support value-based classification. The proposed offloading approach is a promising method to enhance the execution process and reduce battery utilization in versatile applications. The target of the proposed framework is to determine an ideal offloading choice under the setting of the cell phone and cloud assets to give a better execution and less battery utilization. The foremost contributions of this work are as follows:

- Various contexts information is extracted via distinctive models, such as the energy consumption model, communication model, cost model, and execution model.
- Support value is estimated for the separate context information of each task and this measure is utilized for the viable context sensitive offloading in MCC.

[*] Faculty of Automatic Control and Computers, University Polytechnic of Bucharest 313 Splaiul Independentei, 060042, Romania. (alqaisy86@gmail.com).

[†] Middle Technical University Electrical Engineering Technical College, Baghdad, Iraq. (aa_eng_89@yahoo.com).

[‡] Faculty of Al-dour Technical institute, Northern Technical University, Mosel, 41002, Iraq (raed.isc.sa@gmail.com).

[§] Faculty of Automatic Control and Computers, University Polytechnic of Bucharest 313 Splaiul Independentei, 060042, Romania (nicolae.tapus@cs.pub.ro).

- Analyse the total processing energy of each task between two configurations based on Support value evaluation.

J48 is an extension of ID3. The additional features of J48 are accounting for missing values, decision trees pruning, continuous attribute value ranges, derivation of rules, etc. In the WEKA data mining tool, J48 is an open source Java implementation of the C4.5 algorithms. Where JRIP It implements a propositional rule learner called as Repeated Incremental Pruning to Produce Error Reduction (RIPPER) and uses sequential covering algorithms for creating ordered rule lists. The algorithm goes through 4 stages: Growing a rule, Pruning, Optimization and Selection. IBK or K-Nearest Neighbor classification classifies instances based on their similarity [3]. It is a type of Lazy learning where the function is only approximated locally and all computation is deferred until classification. An object is classified by a majority of its neighbors. K is always a positive integer. The layout of this paper is as follows: Section 2 surveys the related works regarding the proposed strategy. In sections 3, a short discussion about the proposed system is given; section 4 examines the exploratory outcomes, and section 5 finalized the study.

2. Related Work and System Overview. In this section, an overview of the proposed framework and the literature works was presented. Numerous related works in offloading for MCC were considered, including offloading decision amongst cell phones and open cloud administrations. Cloud offloading is delicate to the numerous constraints of the framework based on the context of the gadget. Subsequently, current works have proposed new frameworks to address these difficulties in MCC and to empower intermittent observing of numerous advanced measurements used to gather data. Warley Junior et al. [15] proposed a Context-Sensitive Offloading System (CSOS) that exploits the primary machine-picking up thinking methods and strong profiling framework to give offloading choices and increase the level of precision. The method failed to meet expectations when offloading choices overlooks logical data. CSOS executes all assignments identified with the choice on the portable device since the executing demand tasks identified with the preparation and testing of the categorization techniques are accomplished in the arranged period of the framework as a discontinuous procedure. In this manner, the expense of handling at running time is short; meanwhile, the portable devices possibly need to parse the preparation method to choose when to offload information. Tianhui Meng et al. [26] introduced and assessed a protected and effective offloading plan that was a mixture of irregular fillings. To proceed to a quantifiable dealing, a crossbreed Continuous time Markov chain (CTMC) was established; They proposed the security measurements which framework modelers need to settle on educated exchange off choices, including framework security.

Yongin Kwon et al. [18] displayed a specific execution offloading for applications with dynamic conduct in MCC. Execution offloading which moves a string between a cell phone and a server was regularly utilized. In such execution offloading procedures, it was run off the mill to progressively choose the code part to be offloaded through basic leadership calculations. To accomplish an ideal offloading execution, nonetheless, the gain and cost of offloading must be anticipated precisely for such methods. Fangxiaoqi Yu et al. [31] proposed an innovative dynamic mobility aware partial offloading (DMPO) technique to make sense of the measure of information progressively, composed with choice of correspondence way, limiting the energy utilization while fulfilling defer requirement. The suggested technique calculates the opportunity to subsequent transfer and allocates information size to all vacancy in that stage to accomplish the set objectives. Xing Chen et al. [6] proposed a structure that supports versatile applications through the context-aware computation offloading ability. It is a best method to enhance the execution process and reduce the battery utilization. Table 2.1 presents the qualitative examination of the existing context-aware offloading techniques.

Table 2.1 showed a subjective assessment of the prevailing context-aware offloading methods. It presented the strategies used in making the offloading choice and determine when offloading will enhance execution. Moreover, it demonstrates whether the investigation assesses the execution of the technique to identify the level of right choices made. Contrary to the related works, there is still a gap between rapid growth of power consumption and power capacities of current mobile devices. The proposed offloading approach is a promising method to enhance the execution process and reduce battery utilization in versatile applications. The target of the proposed framework is to determine an ideal offloading choice under the setting of the cell phone and cloud assets to give a better execution and less battery utilization. our framework considers the distinctive models for the extraction of context information, such as execution model, communication model, cost model, and

Table 2.1
Qualitative analysis of the existing context-aware offloading techniques

Framework	Decision support	Solutions	Main contribution
CSOS [33]	JRIP and J48 classifiers	High accuracy and context-sensitive.	Offloading decisions that achieved gains and energy efficacy
CTMC [22]	CTMC and queuing method	Secure and cost-efficient Offloading Policy	Cost-efficient offloading scheme
OMMC [11]	TOPSIS and Energy model	Managed the trade-off between time and energy consumption.	Context-aware
Context-aware computation offloading [25]	Advanced structure for context-aware computation offloading	Rightly choose the suitable cloud assets and offload mobile codes on request	Reduced execution time and power consumption
DMPxcO [12]	DMPO algorithm	Decreased energy utilization and achieved the delay limit	Better performance in meeting the delay constraint
f_Mantis [33]	Sparse Polynomial Regression	Better accuracy in offloading	Precise execution offloading
mCloud [22]	TOPSIS and Cost model	mCloud	Context-aware
CoSMOS [22]	Cost functions	CoSMOS	Context-aware
Energy effective offloading choice algorithm [11]	Lyapunov optimization algorithm	Minimized average energy consumption, response time, less computational complexity	Energy-efficient

energy consumption model. Our proposed extricates the context information of input tasks via these models. Afterward, the support consideration for the context information of each assignment is estimated and tasks were arranged based on the settings information. This support consideration achieved better offloading. The outcomes demonstrate that the proposed technique outperformed the existing techniques in terms of accuracy, sensitivity, specificity, FPR, FNR, energy utilization, throughput, and execution time.

3. Effective Context Sensitive Offloading Using Support Value Based Classification Approach.

This paper provides an effective context sensitive offloading framework for mobile cloud environment utilizing support value-based classification in several steps; at first, the input tasks context information is extracted through the distinctive models (energy consumption model, cost model, execution model, and communication models). At this point, the context information is stored as training data through the processing module. After that, the support value is estimated for the context information of each task and creates knowledge about the tasks before making the final decision in the right way to prompt achieving better offloading intensive tasks. The flow structure of the presented technique is shown in Figure 3.1.

3.1. Tasks collection.

Firstly, the users present their tasks in the mobile cloud framework through user requests. The number of tasks gathered depend on the number of clients at the remote site, determined as:

$$T_k = t_1, t_2, t_3, ...t_n \tag{3.1}$$

where T_k is the set of requests and every request contains distinctive assignments. The task demands are accumulated from the clients for each specific time and after that, the accompanying parameters are extracted from the tasks to gather the information of the assignments.

3.2. Context data extraction of tasks.

In this stage, the developer characterized the components that are necessary to be considered as important data for the offloading choice. Here, we distinguished the elements such as the execution time, application nature, execution cost, data size, communication models, energy utilization by CPU, I/O operation, memory, and cache. Every single measurement can vary individually of the others notwithstanding some that occasionally vary after some time whereas others are static. Consequently,

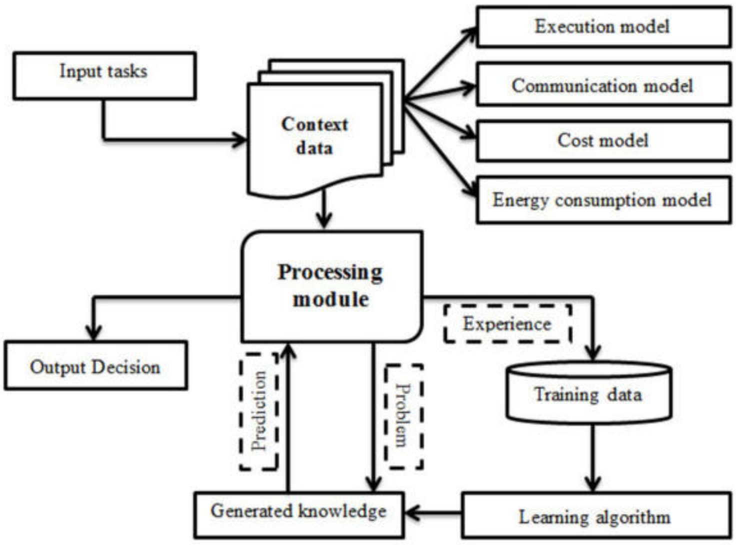


FIG. 3.1. *The flow structure of the presented decision module*

these different groupings can lead to the creation of various choices depending on the relevant data. Here, the context data of the input tasks separated by using the energy consumption model, cost model, execution model, and communication model are presented in Figure 3.1.

The decision are made according context data which is collected in terms of $\hat{E}_{con}$, Ct_{mod}, $\hat{C}_{mod}$ and E_{time} where $\hat{E}_{con}$ Energy Consumption Model, Ct_{mod} Cost model, $\hat{C}_{mod}$ Communication Model and E_{time} Executions Model and these terms are explained in equitions to implement the methodology in processing model by using data traing to generate the knowledge and out the decision.

3.2.1. Energy Consumption Model ($\hat{E}_{con}$).

▷ **Energy consumption by CPU**

The energy consumed by the CPU is designated as $\tilde{E}_{cpu}$ and is calculated using Equation 3.2.

$$\tilde{E}_{cpu} = P_{avg} \cdot T_{exe} \tag{3.2}$$

where P_{avg} signifies the average power utilized by the CPU, T_{exe} signifies the execution time.

▷ **Energy utilization by I/O operation**

The energy utilization through I/O operation by a movable device is calculated using Equation 3.3.

$$\hat{E}_{I/O} = (\hat{T}_{elaps} - \hat{T}_{cpu}) \cdot \hat{P}_{I/O} \tag{3.3}$$

where $\hat{T}_{elaps}$, and $\hat{T}_{cpu}$ represents the overall time and CPU time essential for I/O processes, $\hat{P}_{I/O}$ signifies the power utilization for I/O activity.

▷ **Energy consumption by the memory**

The mobile device consumes its energy to reading and writes memory access. This energy is calculated using Equation 3.4.

$$\hat{E}_{memory} = (E_{read} \cdot \overline{read}) + (E_{write} \cdot \overline{write}) \tag{3.4}$$

where E_{read} and E_{write} are the energy used to execute read and write activities, respectively. This activity signifies the number of reads and writes, correspondingly.

▷ **Energy consumption by the cache**

A mobile device consumes energy for cache read and write and this energy is calculated using Equation 3.5.

$$\tilde{E}_{cash} = E_{cash1} + E_{cashL1} + E_{cashL2} \tag{3.5}$$

where E_{cash1} , E_{cashL1} and E_{cashL2} represent the energy consumption due to the Instruction cache, level 1, and level 2 cache, respectively. The total energy utilization is computed using Equation 3.6.

$$\tilde{E} = \tilde{E}_{cpu} + \tilde{E}_{I\backslash O} + \tilde{E}_{memory} + \tilde{E}_{cash} \tag{3.6}$$

The residual energy of every mobile device can be determined using Equation 3.7.

$$\tilde{E}_{res} = \tilde{E}_T - \tilde{E}_{initial} \tag{3.7}$$

where $\tilde{E}_T$ and $\tilde{E}_{initial}$ represent the total and initial energy of a mobile device.

3.2.2. Communication model ($\hat{C}_{mod}$). The link between a mobile device and multi-sites is viewed as homogeneous; this suggests that the entire information exchange between the mobile device and any site is communicated at a similar rate. If is the power of a dynamic link, then, the energy utilization amid the transfer of the information between links is determined according to Equation 3.8:

$$\tilde{E}D_{active} = \hat{P}_{link-active} \cdot \sum_{i=1}^{t} DS(i, S) \tag{3.8}$$

where $DS(i, S)$ is the data size of module i of an application offloaded on to site (S).

Likewise, the energy expended during the information exchange amongst the sites is determined using Equation 3.9,

$$\tilde{E}S_{acive} = \tilde{P}_{site-active} \cdot \sum_{i=1}^{t} DS(i, S, k) \tag{3.9}$$

where $DS(i, S, k)$ is the task size of the module i of an application offloaded on site S from the site k . $\tilde{P}_{site-active}$ is the power among the dynamic sites. The overall energy utilization throughout the transfer of information is the summation of Equations 3.8 and 3.9 as displayed in Equation 3.10.

$$\tilde{E}_{active} = \tilde{E}D_{active} + \tilde{E}S_{active} \tag{3.10}$$

3.2.3. Execution model(E_{time}).

▷ **Execution time**

The execution time of the module i on location S is determined using Equation 3.11, i.e., the addition of the processing time and information transfer time.

$$E_{time}(i, S) = \frac{tz_i}{CP} + \frac{DS(i, S)}{B(i, S)} \tag{3.11}$$

where CP is the cyclic prefix, $E_{time}(i, S)$ is the complete execution time of the entirely offloaded z modules on the cloud, and is determined using Equation 3.12:

$$TE_{time}(i, S) = \sum_{i=1}^{z} E_{time}(i, S) \tag{3.12}$$

where $E_{time}(i, S)$ is the execution time and $TE_{time}(i, S)$ is the overall execution time of the model.

3.2.4. Cost model (Ct_{mod}).

▷ **Data size**

The data of an input task is signified in a sequence of bits. The size of the information is estimated in bits. Data is a set of values of subjects regarding qualitative or quantitative factors. Data and information are frequently used reciprocally; anyway, data move towards information when it is seen in context. Here, the size of the data is estimated as a context.

▷ **Execution cost**

The execution cost of the module $\hat{E}C(i, S)$ is the product of the unit charger rate and the execution time. This is expressed in Equation 3.13:

$$\hat{E}C(i, S) = \beta \times \hat{E}T(i, S) \tag{3.13}$$

where β is the charge unit of utilizing the site for every time period; the total execution cost is determined using Equation 3.14:

$$\hat{E}C_T = \sum_{i=1}^{t} \hat{E}C(i, S) \tag{3.14}$$

The execution of a module on the quickest site is determined using Equation 3.14. These models might be utilized to enhance the evaluation of decision outcomes.

3.3. Processing Module. The component processing module gets the context information and form of the task, formulates a problem by describing the input data with the context data, and determine the support value for the context information of every task. The information about the context and choice is generated by characterizing the offloading decision which is done by classifying the context data using the support value-based classification method. The processing module chooses the information to be stored in the training database. The processing component is responsible for activities like processing of input information, storing the context data as a training data, classifying the gained information using the support value-based classification method, and executing the offloading procedure.

3.3.1. Storing context data in the database. Storing context data in the database refers to the loading of a training dataset with the context data. Here, all the extracted parameters from the execution model, communication model, energy consumption model, and cost model are stored as training context information. Toward the end of this stage, the support value is estimated for each context data of each task and stored for classification.

3.3.2. Support value measure. In this section, the context data of each task is classified based on the support value of the contexts. Here, the support value calculation depends on the extraction models such as execution model, energy consumption model, communication model, and the cost model. It is calculated using Equation 3.15.

$$\tilde{S}_{value} = (\hat{E}_{con} + \hat{C}_{mod} + Ct_{mod}) \;/\; (\hat{E}_{con} \cdot \hat{C}_{mod} \cdot Ct_{mod}) \tag{3.15}$$

where $\hat{E}_{con}$ is the energy utilization models' data, $\hat{C}_{mod}$ is the communication models' data, and Ct_{mod} is the communication models' data.

3.3.3. Offloading decision using support value-based classification. In this section, the final offloading decision is made using the support value-based classification. Moreover, the information gained from the classified models should be saved in CSOS to consent the decision to load them at runtime. The pseudo code of the offloading decision model using support value-based classification is presented in Algorithm 1.

The pseudocode of the offloading decision-making given in Algorithm 1 deals with the context information and categorizes the utmost current task in the dataset using support value-based classification. Firstly, it gets the extricated context data and the support value-based classifier individually. The context information is extricated from the execution model, communication model, energy utilization model, and cost model. In any case, the estimation of this measurement is well-known at the time that the application's client chooses the

Algorithm 1 Proceduer for affloading decision with classifier

```
begin (Context data, support value based classifier Packet Reception);
for each Contexts $A \in$ to Database do
  Contexts[]contexts← get Contexts (A);
  for i = 1, tasks do
    if context.Get Name[i] = Data Size then
      Contexts[]contexts← get Contexts (A);
    else
      Instant.Set Value[i] ← Values[i];
    end if
  end for
end for
result ← Clasify Instant(support value based classfier, Instant);
if threshold($T_k$) $\leqslant$ result then
  response ← true;
  return response ;
else
  return response ;
end if
```

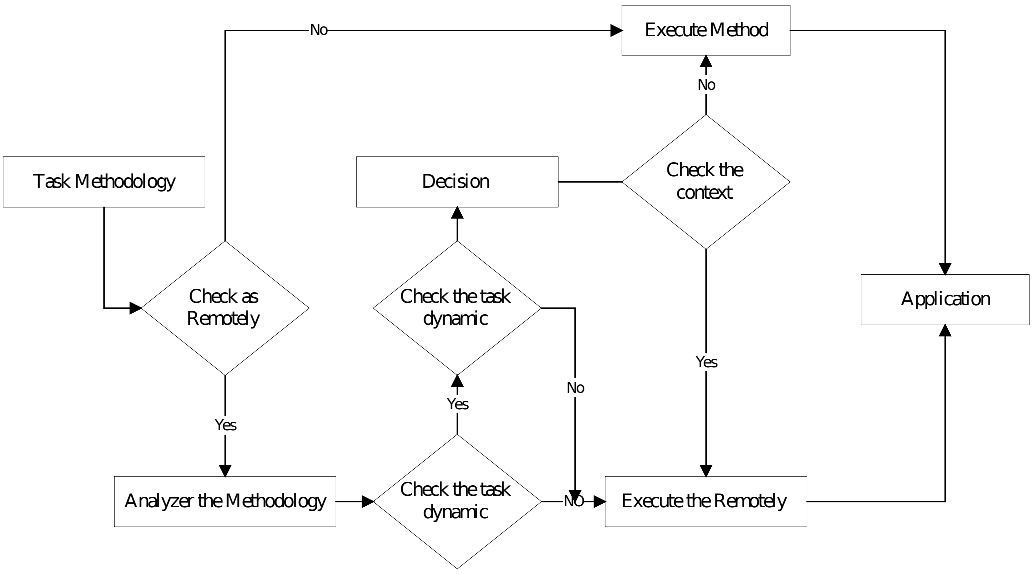


Fig. 3.2. *Execution flow diagram of the proposed offloading decision*

preferred context for processing. Subsequently, the profiling framework runs a search for a specific time which is unrealistic to precisely catch the estimation of this measurement and feed it to the decision engine. Next, every context of the utmost latest record in the database for data gathering is checked. At last, decision is made at the right time to accomplish more prominent advantages in offloading the computation-intensive tasks. The proposed offloading decision execution flow is illustrated in Figure 3.2.

Offloading decision explained in Figure 3.2 that show it gets the extricated context data and the support value-based classifier individually. In any case, the estimation of this measurement is well-known at the time that the application's client chooses the preferred context for processing. Subsequently, it excute the remolty with syncornization its excute the method the find the right decision for such appilaction

4. Performance Measures. The performance of the proposed work was examined using different parameters, such as accuracy, sensitivity, specificity, FPR, FNR, precision, energy consumption, CPU usage, throughput, and execution time as described in the following subsections.

4.1. Accuracy. Accuracy refers to the determination of the accuracy of an outcome in a populace (either positive or negative). It quantifies the level of data classification correctness. Accuracy is calculated using Equation 4.1:

$$\frac{(TN+TP)}{(TN+TP+FN+FP)} \tag{4.1}$$

where TN and TP are true negative and positive correspondingly, FN and FP are false negative and positive, correspondingly.

4.2. Sensitivity. Sensitivity is the measure of the effective differences between the in a classification task. It demonstrates the greatness of a system in data classification and can be calculated using Equation 4.2:

$$Sensitivity = \frac{(TP)}{(TP+FN)} \tag{4.2}$$

4.3. Specificity. Specificity refers to the measure of the effective differences between the by a classification method. It measures the ability of a method to distinguish between normal and abnormal information. Specificity can be calculated using Equation 4.3:

$$Specificity = \frac{(TN)}{(TN+FP)} \tag{4.3}$$

4.3.1. FPR. False positive rate (FPR) ascertains the extent negative events are incorrectly considered as positives, as well as the aggregate quantity of actual negative events. FPR is computed using Equation 4.4:

$$FPR = \frac{(FP)}{(FP+TN)} \tag{4.4}$$

4.3.2. FNR. False negative rate (FNR) is the number of positives which gives negative results. FNR is computed using Equation 4.5.

$$FPR = \frac{(TP)}{(TP+FP)} \tag{4.5}$$

4.3.3. Precision. Precision is the likelihood that information classification with a correct test result truly has precise information. Precision is commonly computed using Equation 4.6:

$$Precision = \frac{(TP)}{(TP+FP)} \tag{4.6}$$

4.3.4. Throughput. Throughput measures process performance per unit time. It is estimated by the fraction of the number of processes finished for a particular time. This is computed using Equation 4.7:

$$\widehat{T}_k = \overline{N}_f \ / \ t_p \tag{4.7}$$

where $\overline{N}_f$ is the amount of process finished, t_p is the chosen time period, and $\widehat{T}_k$ is the throughput.

4.3.5. Energy Consumption. Energy consumption involves all the energy consumed when performing a task. Energy is mainly sourced from electric power (E_{pi}); execution time (E_{exei}) is the time required to process the i^{th} execution up to a number of repetitions. Energy consumption is represented as:

$$\overline{E}_{Ci} = E_{pi} \cdot E_{exei} \tag{4.8}$$

The mean energy is computed using Equation 4.9:

$$\overline{E}_{mean} = \frac{\sum_i^n \overline{E}_{Ci}}{\widehat{n}} \tag{4.9}$$

where $\overline{E}_{mean}$ signifies the mean energy utilization and $\widehat{n}$ signifies the number of reiterations.

Table 4.1
Comparison of the proposed technique in terms of different performance measures

Method	Accuracy	Specificity	Sensitivity	Precision	False positive rate	False negative rate	F1
Proposed	97.05	91.99	95.89	95.12	6.07	1.09	97.32
J48	94.30	93.35	95.97	94.68	4.41	4.91	94.34
JRIP	94.35	88.93	94.77	91.01	5.79	4.54	93.59
IBK	93.12	90.09	94.15	92.34	4.70	4.79	94.75

4.3.6. Execution Time. This is the amount of time utilized to accomplish a given task. The time used to execute a process is calculated using Equation 4.10:

$$\widehat{E}_t = S_d \ / \ \widehat{B}_r \tag{4.10}$$

where $\widehat{E}_t$ signifies the execution time, S_d represents the size of data, $\widehat{B}_r$ represents the bit rate. The evaluation table of the proposed technique compared to the existing techniques in terms of different performance measures is presented in Table 4.1.

5. Results and Discussion. The presented effective context sensitive offloading approach using support value-based classification was implemented in a JAVA platform with Cloudsim. To examine the performance of the presented work, different measures were evaluated. The performance of the proposed method is presented in Table 4.1 and Figure 5.1.

The graphical representation of the analysis of the proposed work is represented in this section. Figure 5.1 compared the performances of the classification algorithms using different indicators like accuracy, sensitivity, specificity, FPR, FNR, precision, energy consumption, CPU usage, throughput, and execution time.

Sensitivity and precision are the two major performance metrics used in applications that placed a premium on the successful detection of a class of events over the other classes. Unfortunately, these two measures are opposed to each other due to the differences between them; for instance, if we want to extract more relevant records (aimed at increasing the rate of sensitivity), more irrelevant records will also be retrieved (decreases the precision rate). Figures 5.1.b and 5.1.e clearly showed that the proposed technique has F1 greater than JRIP, J48, and IBK with a rate of 97.32%. Figure 5.1.a depicted the comparative analysis in terms of accuracy. High accuracy is important in MCC due to the dynamic and adaptive nature of mobile systems. Such nature leads to inaccurate decisions that later results in high energy utilization and performance degradation. The results of this study showed that the accuracy of the rules generated by the support-based algorithm was slightly higher than those of J48, JRIP, and IBK based on the contextual dataset. The proposed algorithm achieved an accuracy of 97.05 while JRIP and J48 algorithms achieved 94.35% and 94.30%, respectively. IBK achieved the worst accuracy level of 93.12% of correctly classified records. From these results, the good classifiers are the proposed algorithm, JRIP, and J48 based on their performances over the context database.

Figure 5.1.f showed the throughput of our proposed effective context sensitive offloading approach using support value-based classification and its CPU usage. The comparison graph of the proposed approach with the existing J48 classifier, JRIP classifier, IBK classifiers in terms of energy consumption was presented in Figure 5.1.g which illustrated the proposed algorithm to give better classification results compared to the existing J48 classifier, JRIP classifier, and IBK classifier. The comparison graph of the proposed approach with the existing classifiers in terms of execution time was presented in Figure 5.1.h, where the proposed approach was shown to give better classification results compared to the existing classifiers.

6. Conclusion. In this paper, we presented an effective context sensitive offloading approach using support value-based classification for context-aware computation offloading. Our framework utilizes diverse models to extricate the context data to ensure a progressive selection of suitable cloud assets and offload mobile codes to them on request. Additionally, the support value-based classification approach effectively generates information about the tasks before making a decision on the right time to accomplish a better offloading. The experimental results showed that our proposed approach outperformed the existing J48 classifier, JRIP classifier, and IBK classifier in terms of accuracy, sensitivity, specificity, FPR, FNR, precision, throughput, energy utilization, and execution time.

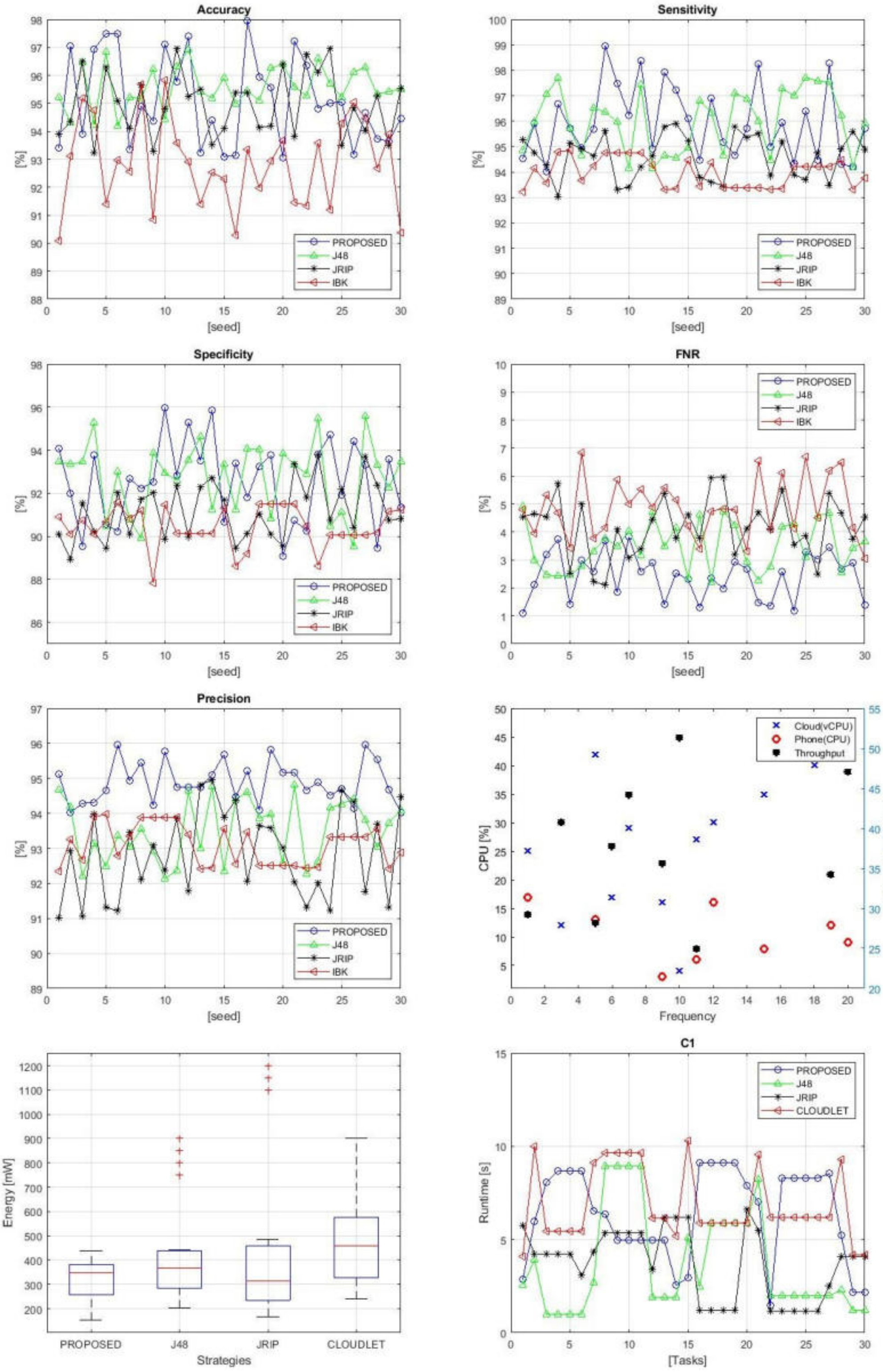


Fig. 5.1. *Comparison of the performances of the classification algorithms using different indicators – from left to right and from top to bottom: (a) Accuracy; (b) Sensitivity; (c) Specifity; (d) FNR; (e) Precision; (f) Throughput; (g) Energy consumption; (h) Execution time*

REFERENCES

[1] E. Cuervo, A. Balasubramanian, D.-k. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, *MAUI: Making Smartphones Last Longer with Code Offload*, in ACM MobiSys 2010, Association for Computing Machinery, Inc., June 2010, https://www.microsoft.com/en-us/research/publication/maui-making-smartphones-last-longer-with-code-offload/.

[2] K. Akherfi, M. Gerndt, and H. Harroud, *Mobile cloud computing for computation offloading: Issues and challenges*, Applied Computing and Informatics, 14 (2018), pp. 1 – 16, https://doi.org/https://doi.org/10.1016/j.aci.2016.11.002, http://www.sciencedirect.com/science/article/pii/S2210832716300400.

[3] A. Al-Badarneh, M. Ali, and S. M. Ghaleb, *An Improved Classifier for Arabic Text*, Journal of Convergence Information Technology (JCIT), 11 (2016), pp. 69–84, http://www.globalcis.org/dl/citation.html?id=JCIT-4357&Search=&op=Title.

[4] A. Alelaiwi, *An efficient method of computation offloading in an edge cloud platform*, Journal of Parallel and Distributed Computing, 127 (2019), pp. 58 – 64, https://doi.org/https://doi.org/10.1016/j.jpdc.2019.01.003, http://www.sciencedirect.com/science/article/pii/S0743731519300140.

[5] X. Chen, *Decentralized Computation Offloading Game for Mobile Cloud Computing*, IEEE Transactions on Parallel and Distributed Systems, 26 (2015), pp. 974–983, https://doi.org/10.1109/TPDS.2014.2316834.

[6] X. Chen, S. Chen, X. Zeng, X. Zheng, Y. Zhang, and C. Rong, *Framework for Context-aware Computation Offloading in Mobile Cloud Computing*, J. Cloud Comput., 6 (2017), pp. 71:1–71:17, https://doi.org/10.1186/s13677-016-0071-y, https://doi.org/10.1186/s13677-016-0071-y.

[7] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, *CloneCloud: Elastic Execution Between Mobile Device and Cloud*, in Proceedings of the Sixth Conference on Computer Systems, EuroSys '11, New York, NY, USA, 2011, ACM, pp. 301–314, https://doi.org/10.1145/1966445.1966473, http://doi.acm.org/10.1145/1966445.1966473.

[8] Fei Gu and Jianwei Niu and Zhiping Qi and Mohammed Atiquzzaman, *Partitioning and offloading in smart mobile devices for mobile cloud computing: State of the art and future directions*, Journal of Network and Computer Applications, 119 (2018), pp. 83 – 96, https://doi.org/{https://doi.org/10.1016/j.jnca.2018.06.009}, {http://www.sciencedirect.com/science/article/pii/S1084804518302170}.

[9] D. Fesehaye, Y. Gao, K. Nahrstedt, and G. Wang, in 2012 IEEE 16th International Enterprise Distributed Object Computing Conference.

[10] S. Ghasemi-Falavarjani, M. Nematbakhsh, and B. S. Ghahfarokhi, *Context-aware multi-objective resource allocation in mobile cloud*, Computers and Electrical Engineering, 44 (2015), pp. 218 – 240, https://doi.org/https://doi.org/10.1016/j.compeleceng.2015.02.006, http://www.sciencedirect.com/science/article/pii/S0045790615000312.

[11] R. A. Hasan, M. A. Mohammed, Z. H. Salih, M. A. B. Ameedeen, N. Ţăpuş, and M. N. Mohammed, *HSO: A Hybrid Swarm Optimization Algorithm for Reducing Energy Consumption in the Cloudlets*, TELKOMNIKA, 16 (2018), pp. 2144–2154, https://search.proquest.com/docview/2126486677?accountid=33993.

[12] R. A. Hasan, M. A. Mohammed, Z. H. Salih, M. A. B. Ameedeen, N. Ţăpuş, and M. N. Mohammed, *HSO: A Hybrid Swarm Optimization Algorithm for Reducing Energy Consumption in the Cloudlets*, TELKOMNIKA, 16 (2018), pp. 2144–2154, https://search.proquest.com/docview/2126486677?accountid=33993.

[13] R. A. HASAN and M. N. MOHAMMED, *A Krill Herd Behaviour Inspired Load Balancing of Tasks in Cloud Computing*, Studies in Informatics and Control, 26 (2017), https://doi.org/10.24846/v26i4y201705, https://app.dimensions.ai/details/publication/pub.1099870816andhttps://sic.ici.ro/wp-content/uploads/2017/12/SIC_2017-4-Art.5.pdf.

[14] H. Jadad, A. Touzene, K. Day, and N. Alzeidir, *A cloud-side decision offloading scheme for mobile cloud computing*, International Journal of Machine Learning and Computing, 8 (2018), pp. 367–371.

[15] W. Junior, E. Oliveira, A. Santos, and K. Dias, *A context-sensitive offloading system using machine-learning classification algorithms for mobile cloud environment*, Future Generation Computer Systems, 90 (2019), pp. 503 – 520, https://doi.org/https://doi.org/10.1016/j.future.2018.08.026, http://www.sciencedirect.com/science/article/pii/S0167739X17326729.

[16] S. Kosta, A. Aucinas, Pan Hui, R. Mortier, and Xinwen Zhang, *ThinkAir: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading*, in 2012 Proceedings IEEE INFOCOM, March 2012, pp. 945–953, https://doi.org/10.1109/INFCOM.2012.6195845.

[17] M. D. Kristensen and N. O. Bouvin, *Scheduling and development support in the Scavenger cyber foraging system*, Pervasive and Mobile Computing, 6 (2010), pp. 677 – 692, https://doi.org/https://doi.org/10.1016/j.pmcj.2010.07.004, http://www.sciencedirect.com/science/article/pii/S1574119210000581. Special Issue PerCom 2010.

[18] Y. Kwon, H. Yi, D. Kwon, S. Yang, Y. Cho, and Y. Paek, *Precise execution offloading for applications with dynamic behavior in mobile cloud computing*, Pervasive and Mobile Computing, 27 (2016), pp. 58 – 74, https://doi.org/https://doi.org/10.1016/j.pmcj.2015.10.001, http://www.sciencedirect.com/science/article/pii/S1574119215001856.

[19] K. Lin, S. Pankaj, and D. Wang, *Task offloading and resource allocation for edge-of-things computing on smart healthcare systems*, Computers and Electrical Engineering, 72 (2018), pp. 348 – 360, https://doi.org/https://doi.org/10.1016/j.compeleceng.2018.10.003, http://www.sciencedirect.com/science/article/pii/S0045790617339174.

[20] C. M. S. Magurawalage, K. Yang, L. Hu, and J. Zhang, *Energy-efficient and network-aware offloading algorithm for mobile cloud computing*, Computer Networks, 74 (2014), pp. 22 – 33, https://doi.org/https://doi.org/10.1016/j.comnet.2014.06.020, http://www.sciencedirect.com/science/article/pii/S1389128614003193. Special Issue on Mobile Computing for Content/Service-Oriented Networking Architecture.

[21] T. Meng, K. Wolter, H. Wu, and Q. Wang, *A secure and cost-efficient offloading policy for Mobile Cloud Computing against timing attacks*, Pervasive and Mobile Computing, 45 (2018), pp. 4 – 18, https://doi.org/https://doi.org/10.

1016/j.pmcj.2018.01.007, http://www.sciencedirect.com/science/article/pii/S1574119216304308.

[22] M. A. Mohammed, R. A. Hasan, M. A. Ahmed, N. Tapus, M. A. Shanan, M. K. Khaleel, and A. H. Ali, *A Focal load balancer based algorithm for task assignment in cloud environment*, in 2018 10th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), June 2018, pp. 1–4, https://doi.org/10.1109/ECAI.2018.8679043.

[23] M. A. MOHAMMED and N. ŢĂPUŞ, *A Novel Approach of Reducing Energy Consumption by Utilizing Enthalpy in Mobile Cloud Computing*, Studies in Informatics and Control, 26 (2017), https://doi.org/10.24846/v26i4y201706, https://app.dimensions.ai/details/publication/pub.1099870815andhttps://sic.ici.ro/wp-content/uploads/2017/12/SIC_2017-4-Art.6.pdf.

[24] N. Q. Mohammed, M. S. Ahmed, M. A. Mohammed, O. A. Hammood, H. A. N. Alshara, and A. A. Kamil, *Comparative Analysis between Solar and Wind Turbine Energy Sources in IoT Based on Economical and Efficiency Considerations*, in 2019 22nd International Conference on Control Systems and Computer Science (CSCS), May 2019, pp. 448–452, https://doi.org/10.1109/CSCS.2019.00082.

[25] F. A. Nakahara and D. M. Beder, *A context-aware and self-adaptive offloading decision support model for mobile cloud computing system*, Journal of Ambient Intelligence and Humanized Computing, 9 (2018), p. 1561, https://doi.org/https://doi.org/10.1007/s12652-018-0790-7.

[26] C. Z. Radulescu and M. Radulescu, *Group decision support approach for cloud quality of service criteria weighting*, Studies in Informatics and Control, 27 (2018), pp. 275–284.

[27] Z. H. Salih, G. T. Hasan, M. A. Mohammed, M. A. S. Klib, A. H. Ali, and R. A. Ibrahim, *Study the Effect of Integrating the Solar Energy Source on Stability of Electrical Distribution System*, in 2019 22nd International Conference on Control Systems and Computer Science (CSCS), May 2019, pp. 443–447, https://doi.org/10.1109/CSCS.2019.00081.

[28] Shuang Chen, Yanzhi Wang, and M. Pedram, *A semi-Markovian decision process based control method for offloading tasks from mobile devices to the cloud*, in 2013 IEEE Global Communications Conference (GLOBECOM), Dec 2013, pp. 2885–2890, https://doi.org/10.1109/GLOCOM.2013.6831512.

[29] G. Skourletopoulos, C. X. Mavromoustakis, G. Mastorakis, J. M. Batalla, C. Dobre, S. Panagiotakis, and E. Pallis, *Towards Mobile Cloud Computing in 5G Mobile Networks: Applications, Big Data Services and Future Opportunities*, Springer International Publishing, Cham, pp. 43–62, https://doi.org/10.1007/978-3-319-45145-9_3, https://doi.org/10.1007/978-3-319-45145-9_3.

[30] H. Tout, C. Talhi, N. Kara, and A. Mourad, *Smart mobile computation offloading: Centralized selective and multi-objective approach*, Expert Systems with Applications, 80 (2017), pp. 1 – 13, https://doi.org/https://doi.org/10.1016/j.eswa.2017.03.011, http://www.sciencedirect.com/science/article/pii/S0957417417301586.

[31] F. Yu, H. Chen, and J. Xu, *DMPO: Dynamic mobility-aware partial offloading in mobile edge computing*, Future Generation Computer Systems, 89 (2018), pp. 722 – 735, https://doi.org/https://doi.org/10.1016/j.future.2018.07.032, http://www.sciencedirect.com/science/article/pii/S0167739X17329813.

[32] B. Zhou, A. V. Dastjerdi, R. N. Calheiros, S. N. Srirama, and R. Buyya, *A Context Sensitive Offloading Scheme for Mobile Cloud Computing Service*, in 2015 IEEE 8th International Conference on Cloud Computing, June 2015, pp. 869–876, https://doi.org/10.1109/CLOUD.2015.119.

[33] B. Zhou, A. V. Dastjerdi, R. N. Calheiros, S. N. Srirama, and R. Buyya, *mCloud: A Context-Aware Offloading Framework for Heterogeneous Mobile Cloud*, IEEE Transactions on Services Computing, 10 (2017), pp. 797–810, https://doi.org/10.1109/TSC.2015.2511002.

Edited by: Dana Petcu
Received: Jun 6, 2019
Accepted: Aug 9, 2019

DOI 10.12694/scpe.v20i4.1586
Scalable Computing: Practice and Experience ISSN 1895-1767
Volume 20, Number 4, pp. 699–707. http://www.scpe.org © 2019 SCPE

APPLYING SEMANTIC WEB TECHNOLOGIES TO DISCOVER AN ONTOLOGY OF COMPUTER ATTACKS

ANDREI ZAMFIRA, RALUCA FAT, AND CĂLIN CENAN*

Abstract. The main scope of this paper is to present a methodology of engineering an ontology and to demonstrate how it is applied for designing and evaluating cyber-defense systems. The ontology is intended to be a vast model of the cybersecurity domain that captures a lot of information about attacks, source and target systems, methods, vulnerabilities exploited, consequences, controls for mitigation etc. For evaluating the quality of the proposed model we headed towards state-of-art methodologies comprised of a suite of metrics for assessing, among others: correctness, consistency, accuracy, completeness, soundness, task orientation. For the most important task, evaluation of efficacy in attacks detection, the proposed ontology was used as a knowledge model of a prototype web application firewall and we tested the system on a known evaluation dataset. The proposed system yielded a good detection rate and a low rate of false positives and negatives on the test data, and it was compared with other existing solutions in the field.

Key words: Ontology, data model, IDS, Semantic Web technology, knowledge representation and sharing

AMS subject classifications. 68Q55

1. Introduction.

Computer Security, also known in some places "Cyber-Security" or "IT security", is the science that deals with the protection of computer systems from theft or damage to the hardware, software or data from them, as well from disruption or unauthorized use of their services. It includes controlling the physical access of hardware, protection against harm that come from network access, the malpractice by operators, either intentional or accidental. The field is of growing importance due to the reliance on Internet and computer networks of the society (e.g. Wi-Fi, Bluetooth etc), and the growth of "smart devices", such as mobile phones, television, devices from the Internet of Things etc [21].

Cybersecurity is critical in almost every industry that relies on computing equipment. Today most electronic devices (PCs, laptops, cellphones) come with built in firewalls software, but these do not make them 100% accurately protected against threats [2]. There are many ways in which computer systems can be hacked: using the network, download files from unsafe sites, connect to untrusted Wi-Fi networks, resource consumption, electromagnetic radiation etc. They can be protected through good software and hardware. By having strong internal interactions of properties, software complexity can prevent security failures and software crash [3]. The most important areas of industry that need protection against cybernetic threats are finances, aviation, automotive, industrial equipment, Internet of Things, among others.

Because this is what our current paper is intended to do, build a system that detects computer attacks, next we will present the reader with some basic notions and references where he can find more knowledge in the literature.

A system that is used in a suspicious situation to defend another system (or group) from the occurrence of cybernetic attacks is called an Intrusion Detection System (IDS). According to the NIST guide [6], the following is the definition of an IDS: "intrusion detection is the process of monitoring the events occurring in a computer system or network and analyze them for signs of possible incidents, which are violations or imminent threats of violation of computer security policies, acceptable usage policies, or standard security practices. Intrusion prevention is the process of detection supplied with the capability to stop the possible incidents".
Four types of IDSs are known today [4]:

- *network-based:* monitor network traffic for particular network segments or devices and analyzes the network and application protocol activity to identify aware situations
- *host-based:* monitors the characteristics of a single host and the events occurring within it
- *wireless:* monitors wireless network traffic and analyze it to identify suspicious activity involving the wireless networking protocols
- *network behavior analysis:* examines network traffic to identify threats that generate unusual traffic flows, such as distributed denial of service (DDoS), certain forms of malware, and policy violations

[*] Politehnica University of Timişoara, Romania (andreizamfira@gmail.com)

The domain literature states that IDSs have gone through a few stages of evolution until present, that are:

- attacks signatures
- attack taxonomies
- attack ontologies

The first type of IDSs were signature-based, which means that they keep a syntactic representation of the attacks. This method is not very efficient due to many reasons, like attack signatures are not generic in nature, use specific languages related to particular domains and depend upon specific environments and systems, consequently they lack extensibility and dont suit for communication in heterogeneous environments. Attack signatures carry vague semantic information and lack solid ground of any formal logic, the smallest variation in business logic makes the signature invalid [5].

The second phase in the evolution of IDSs represented the use of taxonomies. Central components in the functionality of IDSs are taxonomies, which characterize and classify the attacks information, and a language to describe the instances of the taxonomy.

The current phase in IDSs evolution relies on the Semantic Web technologies, the principals are ontologies. Security systems built using an ontological approach are a promising new line of defense that can detect zero-day and sophisticated attacks because of the ability to capture the context of information and filter them by specific criteria [2]. Various generic security controls, like signature-based firewalls, intrusion detection and prevention systems or encryption devices have been developed, but their effectiveness against web-based threats is restricted due to their extreme rigidness. To obtain an efficient mitigation and stop the attack the system should understand the context of information to be processed and have the ability to filter the contents based on their effect on the target application. This is why security frameworks that rely ontologies are used in these situations [2].

An ontology is an explicit specification of the conceptualization of a domain which captures its context (interpretation of words in a specific domain). Ontological models are flexible in defining the concepts to desired level of detail, easily extensible and provide reasoning ability to reason over the instances of data of the domain. The fields of Artificial Intelligence and semantics use formal ontologies for knowledge sharing and reuse between software entities [3].

The main contributions of the current work are:

- ontological model of attacks: captures the context of important attacks, various of their technologies, sources and targets, consequences on the systems, vulnerabilities exploited and controls for mitigation of the attacks.
- a comprehensive best metrics suite for evaluation of the ontology in order to assess the quality of the proposed model; it includes: formal correctness, consistency, soundness, task orientation, completeness, conciseness, expandability, reusability and others.

2. Related Works. The use of semantic information in the development of security systems is an emerging research field that aims to create more effective defense systems that have a better performance in detecting the ever increasing in number and complexity of cybernetic attacks. The main part of technologies used in this area, such as ontologies, agents, neural networks, Bayesian filters, etc come mostly from the Semantic Web and Artificial Intelligence domains.

The current research represents a continuation of our work from [30]. This time, for the creation of our ontology we used a different development methodology, we took care to make a better description of each of its steps in order for the reader to better understand the process. For the evaluation phase, our ontology was tested for detection actual attacks using a commercial application firewall, Shadow Daemon, that has rules for detecting multiple types of attacks in computer networks, and we described the entire functionality of the system in order for the reader to understand how the process of detection is done, using figures and textual descriptions. This also hasn't been done in the other paper.

In [14] is presented a study made by a team from the MITRE corporation, as part of the JASON project, on the cybersecurity domain, to identify what is needed in creating a full-fledged science of Cybersecurity and recommend specific ways in which can be applied scientific methods. The study identified some fields of Computer Science that are most relevant and provided recommendations on further development of the science.

Razzaq [1] proposes two ontological models that store information from the cyber-security domain: one is about attacks that occur at the application level and the other captures information about HTTP communication protocol. He sustains that security frameworks built using ontological approaches are the next-gen line of defense because are able to capture the context of information.

Garcia-Teodoro et al. make in [11] a literature review of the techniques employed in building anomaly-based network NIDS systems, which he put into 3 categories: statistical-based, knowledge-based and machine-learning based. In the latter we find many techniques from the AI domain, such as Neural networks, Fuzzy logic, Markov models, Genetic algorithms, Clustering and outlier etc.

Papers [12] and [16] make a literary study about the use of an AI technology, namely Machine Learning in construction of detection systems. Compare the methods for intrusion detection based on the classifiers type: single, hybrid and ensemble.

In [6] also is sustained the idea of using ontologies in construction of intrusion detection systems. Propose an ontology to classify information about contexts of attacks that is used by a system to detect attacks at application level.

In [15] is presented an implementation of an IDS using Genetic Algorithms for detecting various types of intrusions in networks. This technology from AI uses evolution and natural selection that relies on a chromosome-like data structure and evolve the chromosomes using the operators of selection, recombination and mutation [18].

The most rightful guide in Computer Security and IDS systems is the recommendation of NIST (National Institute of Standards and Technology) in [6], which presents in large details (but not too broadly) what are the types of IDPS systems, capabilities and features of each one, how they are integrated and how is one selected for a special kind of activity.

In [17] is sustained the idea that a full-fledged science of cybersecurity has to be created in order to solve the problems related to vulnerabilities found in networks of computer systems. Its core principle is to cognize the cyberspace as a hybrid framework of interactions between humans and machines where security and privacy policies play a crucial role.

The contribution of our work compared to the ones presented above is that we do not only scratch the surface in building ontological models, but we tried to build a rigorous model by following state-of-art construction methodologies from the literature, and also to evaluate its capabilities by placing it in scenarios for which it was constructed to be used in the real applications. The experiments and results demonstrated that our ontological model behaves good for the scope for which it was created, that is improve the detection accuracy of IDS systems.

3. Building the Ontology. As it was stated in previous sections, an ontology is a specification in form of a data structure that captures the important concepts of an application domain and their relations. The process of ontology design is an iteration to determine its purpose, define concepts (classes), relations (properties), axioms, constraints and instances.

For the construction of our ontological model of cyber-attacks we chose the METHONTOLOGY development methodology. In [4] it is stated that this is the most mature methodology for ontologies engineering, as compared to others like Uschold and Kings, Gruniger and Foxs, or Bernaras methodologies.

In figure 3.1 can be seen the 6-steps process of constructing ontologies according to the METHONTOLOGY methodology [5].

Next we will try to explain this process from the perspective of developing our ontology.

Phase 1: Scope
The main scope of the proposed ontology in this paper is to express the complex knowledge of the cyber-security domain in a way that can be computationally traceable, machine processable and facilitate communication among software agents.

Phase 2: Elicitation
In order to obtain the necessary information to construct our knowledge model we studied various resources from the literature of cybersecurity, like catalogs, dictionaries and taxonomies of classes of attacks and malware, techniques used by hackers, target components, vulnerabilities exploited, consequences, etc. These sources include NISTs SCAP suite [28] (OVAL, CPE, CCE, CVE), CERT/CC Advisories, MAEC[26], CAPEC[27].

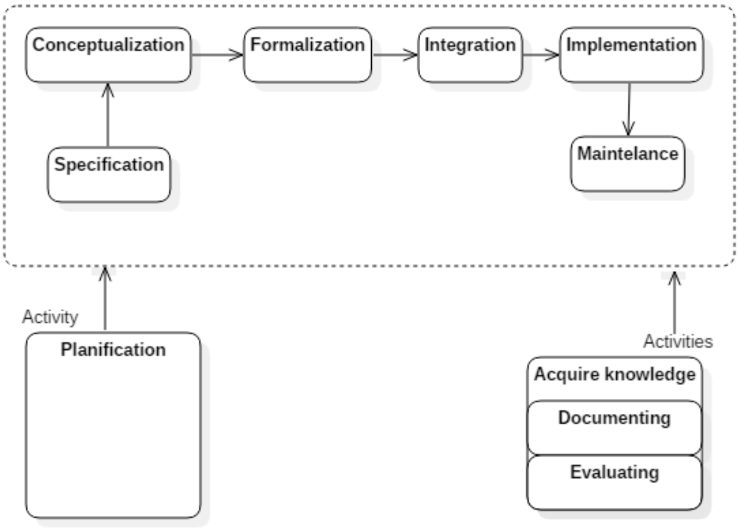


Fig. 3.1. *Phases of ontology development in METHONTOLOGY*

To acquire information about components of the ISO/OSI stack layers was studied the Internet Catalog of Assailable Technologies (ICAT), now called National Vulnerability Database (NVD) [31].

Phase 3: Conceptualization
After acquiring sufficient knowledge form the resources of cyber-security we were able to create our conceptual model by extracting the concepts (classes) and relations (properties) between them. The classes found are organized into a hierarchy of three levels, from the level of generalization they acquire. Classes from a lower level are sub-concepts of ones from superior levels.

In an ontology there are 3 types of properties: of objects, data and annotation. Object properties are relationships that links classes and objects. Data properties are attributes of classes that represents their structure. Annotation properties are sources of information that are attached to a class that says things about it. Facets of properties are restrictions that apply to properties, such as: data type, cardinality, quantifiers, hasValue restrictions.

Our ontological model comprises 106 classes, 38 object properties, 22 data properties and 3 annotation properties. In figures 3.2 and 3.3 are shown how classes and relations look in Protégé 4 editor.

Phase 4: Formalization
For the formal design of our conceptual model found so far we used a form of pseudo-code in which we expressed, in a top-down manner, our ontological system. We started with the complete model:

$$O = (C, P, A, I) \tag{3.1}$$

where C is the set of concepts, P the set of properties, A the set of axioms, and I the interpretation of the model. These can be further elaborated as:

$$C = (\cup_{t \in Type} C_t) \cup (\cup_{i \in Type} I_i) \tag{3.2}$$

$$P = (\cup_{p \in Type} P_p) \cup (\cup_{e \in Type} Rel_e) \tag{3.3}$$

$$A = (\cup_{a \in Type} A_a) \cup (\cup_{r \in Type} R_r) \tag{3.4}$$

where C is the set of all concepts with type t, I is the set of all instances with type i and P is the set of all properties with type p; Rel represents all relationships of type e , such as subsumption, equivalence and disjointness; A is a set of axioms and R a set of rules.

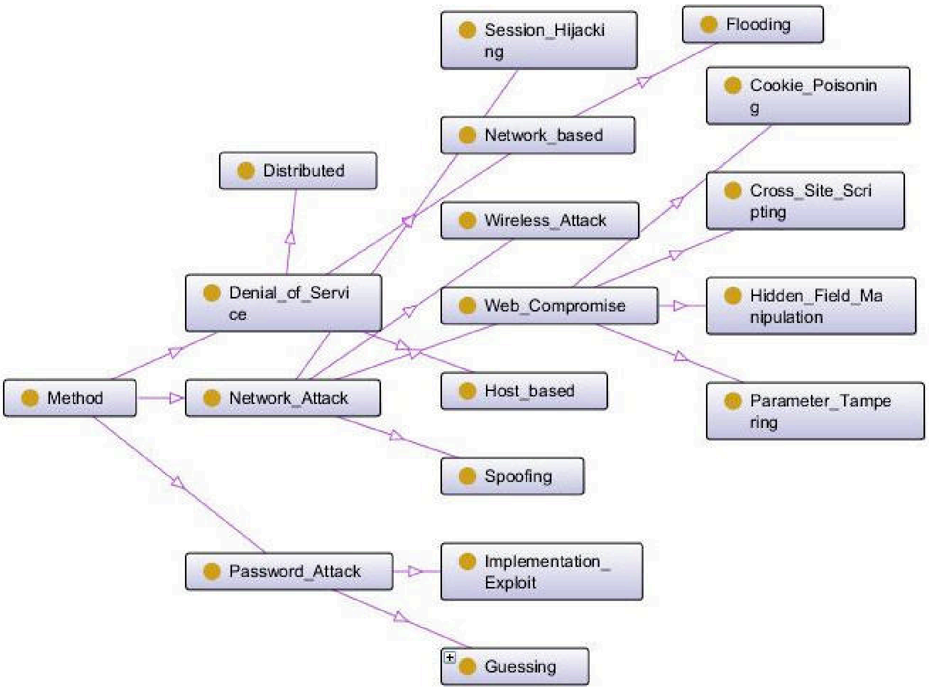


FIG. 3.2. *The owl:Method class and its subclasses*

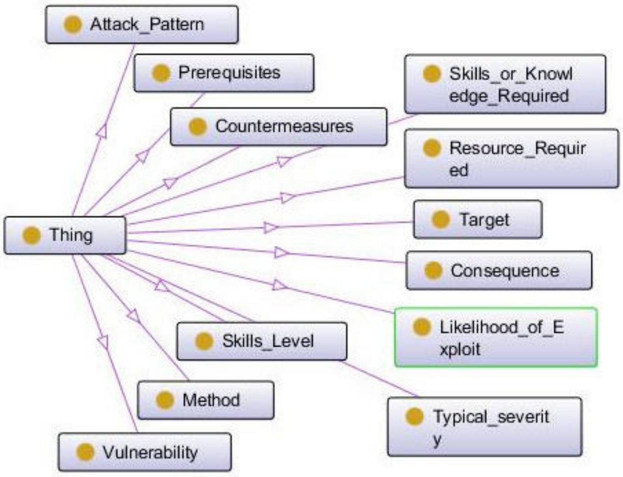


FIG. 3.3. *Top level classes of the ontology (OWLViz tool, Protege)*

Properties and relationships can be formally represented as:

$$P = (D : datatype, O : object, T : transitive, F : functional) \tag{3.5}$$

$$Rel = (\equiv : equivalence, \subseteq : subsumption, \cap : disjointnes) \tag{3.6}$$

Phase 5: Integration

This phase claims that ontologies must be created with the reuse goal in mind. To achieve this step for our ontology we chose a good formal language, OWL, and a development editor, so that when want to extend it to represent new knowledge we can easily do that by adding new axioms to the existing set.

Phase 6: Implementation
For the implementation we headed towards the state-of-art languages for ontologies construction, that is OWL second version, OWL 2.0. As the development editor we also chose currently best, that is Protégé 5 of Stanford University, California.

4. Evaluation. For the evaluation of our ontology we chose OntoClean [22], which we think it is the right methodology for our model. For other methodologies proposed in literature and a comparison of them reader is referred to [3].

Although in the literature there are stated more than 15 metrics for evaluation, we chose only 8 that we considered that are more relevant for our model, which will be discussed below.

Formal correctness: all information in the ontology must be accurate and valid according to existing standards of the domain modeled (in our case, cyber-security). OntoClean meta-properties rigid, identity and unity were applied to classes and properties of the ontology to ensure its correctness and checked for model specifications and subsumption relations violations using automated tools. Moreover, wrong patterns in the model had been detected and removed using SPARQL queries.

Consistency: according to [1] a model is consistent if all its relations are consistent and comply with its characteristics. We checked the consistency of our model by using the Pellet reasoner: the relations were verified in 66ms, class hierarchy in 264ms and inference was realized in 16ms.

Completeness: the proposed ontology by our work tries to be a large model that captures the domain as good as possible in order to increase the chances of the detection system to detect new and sophisticated attacks.

Expandability: it is possible to add new knowledge with minimum effort into our ontology using the process of semantic ontology alignment.

Clarity: each component of the ontology, i.e. concept, relation, axiom, rule is well stated and documented in order to be easily used, analyzed, understood etc.

Computational complexity, integrity, efficiency: the ontology was designed that does not generate and apply new rules each time a new situation occurs, to avoid redundant work of the system. Any change in the model may create new rules or regenerate existing instances. The semantic rules and constraints of the model are applied on concepts and properties, unlike signature-based techniques that have a less efficient way to capture contents of attacks and are prone to false positives.

Performance: the metrics for this criterion we chose from a bunch of many that were proposed in literature, as can be found in [30]: Detection rate (DR), Intrusion detection capability (CID), Area under ROC curve (AUC)[29].

Detection rate is defined as the ratio between number of correctly classified attacks and the total number of attacks.

$$CID = \frac{TP}{TP + FN} \tag{4.1}$$

CID is as an objective metric based on information theory that was proposed to solve the lacks and disadvantages of others existing. It takes into account all important aspects of detection capability, such as base rate B, positive predictive values (named also Bayesian detection rate - PPV), negative predictive values (NPV) and the probability of intrusions.

$$CID = -B(1-\beta)\log(PPV) - B(1-\beta)\log(1-NPV) - (1-B)(1-\alpha)\log(NPV) - (1-B)\alpha\log(1-PPV)$$

Receiver operating characteristics (ROC) curves are used on the one hand to visualize the relation between rates of detection and false positives of a classifier in its tuning phase, and also to compare the accuracy of several classifiers. Is computed with formula:

$$ROC = \frac{Sensitivity}{1 - Specificity} \tag{4.2}$$

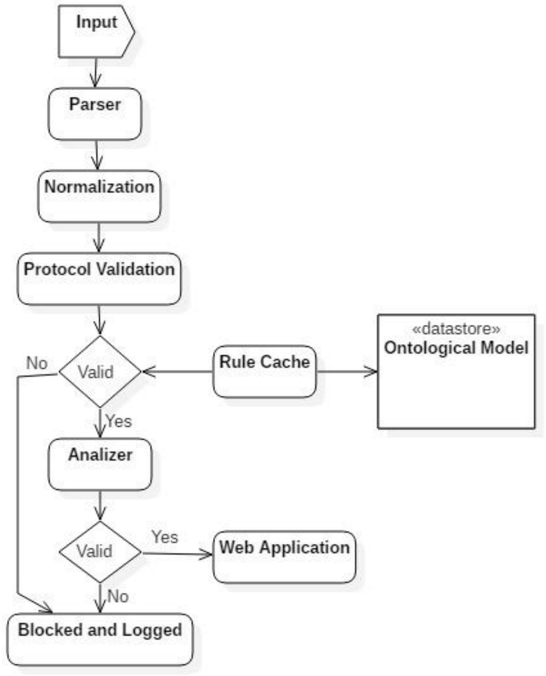


FIG. 5.1. *Ontology used by the detector*

where Sensitivity represents the fraction of true positives that are predicted as positives, and Specificity is the fraction of true negatives that are predicted as negatives. Area under the ROC curve is used as a summary of statistics [29].

Task orientation: this criterion ensures that the model fulfills the functional requirements for which it was developed. This will be proved during the course of the next section, where will be given details about the usage of the ontological model as part of the detection system.

5. Deployment. The proposed system is a novel approach for application of semantic technologies in the domain of information security. Various instances of attacks and vulnerabilities are tested using a prototype web application firewall. The ontology is stored into the firewalls knowledge base from where it is used by means of rules and inference to detect occurred situations. The use of semantic rules allows the system allows the model to be more time efficient because it provides substantial reduction in search space and yields low rates of false positives. The proposed system showed comparative detection rates to some of the best existing solutions today, like Snort and ModSecurity. In figure 5.1 is presented the architecture of the detection system, with its main modules.

Next we will try to explain how the process of detection is realized by the firewall. We will take as example a Cross Site Script (XSS) attack that a user injects into an application in an encoded form:

```
%3Cscript%3E%20alert(%22This%20is%20cross%20site%20)script%22)%20%3C%
2Fscript%3E%20site%20)script%22)%20%3C%2Fscript%3E
```

The input field is checked by the Parser for encoding, and in case yes then it is decoded; the above string decoded looks like:

```
<script>Alert("This is cross site scripting")</script>
```

After Normalization module, where it is transformed and rearranged by certain scales of the system, request is passed further to Protocol Validation and Analyzer modules.

In Protocol Validation and Analyzer the request is matched against the semantic rules that are generated

Table 6.1
Comparing our solution with existing systems

Metric Solution	Detection Rate	Detection Capability	Area under ROC
Ours	0.9050	0.8990	0.9071
Snort	0.9225	0.9031	0.9127
Suricata	0.8990	0.8750	0.8868
Bro	0.6780	0.5640	0.6688

by the ontological models from the KB for identification of malicious content in the input. Protocol Validation module is responsible for violations of protocol specifications, and the Analyzer for other attacks types. If the input content matches any of the generated rules then the request is blocked and is made a log with information about the attack found. Below is an example of a rule generated by the system from the ontological model.

```
[rule10: (?x rdf:type ex:HTTPRequest)  (?y rdf:typeex:ResponseHeaders)
        (?z rdf:type ex:ResponseSplitting)(?x ex:hasRequestHead) ?y)->(?x ex:hasAt ?z)]
```

6. Tests and Results. To test our ontology-based detection system on different types of attacks occurring in computer networks we used test datasets that are specifically created for this purpose. We chose KDDCup99 [23] since it is a medium size dataset that is more fit for our purpose. Other evaluation sets in this category are Kyoto2006+ [17], but this is too large and fit especially for IDSs of industrial scale.

Our system was compared to two state-of-art detectors existing today, Snort, Suricata and Bro [24]. Table 6.1 presents the results of evaluation of these systems based on three performance criterias stated in section 4.

The detection rate of the proposed system was very high, about 90%, very close to that of Snort and bigger than Suricata. The false alarm rate was 0.6%, also comparable to Snort.

7. Conclusions. In this work we proposed an ontology as a broad model of cyber-security domain, that tries to capture as much concepts and relations as possible in order to increase the performance of the detection system in which is used. It was constructed and evaluated using state-of-art methodologies in this purpose. For testing in detection of actual attacks in computer networks it was embedded into a web application firewall as its knowledge model, which consulted it each time to find out the nature of a new situation. The detection performance was compared to those of other 2 existing solutions today, and the results were comparable. The ontology can be downloaded from the authors drive account:

`https://drive.google.com/open?id=1NY7vBaoWQcI8QApP26SWaR8WY6bqJ1WK`

This paper represents only the beginning of our research in construction of Intrusion Detection and Prevention Systems (IDPS) to be used in various environments (hosts, LANs, wireless, etc). For now we limited only to an introduction into the domain and as contribution we created an ontological model using semantic technologies that can be used in detection activity. In the next chapter of our research we propose to move on and to actually implement an IDS that uses the proposed ontology in detecting the nature of situations from a host or network.

REFERENCES

[1] A.Razzaq, Z.Anwar, F.Ahmad, *Ontology for attack detection: An intelligent approach to web application security*, Computers & Security, Elsevier, vol.45 (2014).

[2] L.Obrst, P.Chase, R.Markeloff,*Developing an ontology for the cyber-security domain*, Semantic Technologies for Intelligence, Defense and Security (STIDS 2012)

[3] J.Hartman, P.Spyns, A.Giboin, D.Maynard, R.Cuel, *Methods for ontology evaluation*, EU-IST Network of Excellence (NoE), 2005

[4] F.Lopez, *Overview of methodologies for building ontologies*, International Joint Conference on Artificial Intelligence, 1999

[5] M.Fernandez, A.Gomez-Perez, N.Juristo, *METHONTOLOGY: From ontological art towards ontological engineering*, Association for the Advances in Artificial Intelligence, 1997

[6] K.Scarfone, P.Mell, *Guide to Intrusion Detection and Prevention Systems(IDPS)*, Reccomendations of the National Institute of Standards and Technology (NIST), Special Publication 2007

[7] F.Abdoli, M.Kahani, *Ontology-based Distributed Intrusion Detection System*, Proceedings of the 14th CSI Computer Conference (CSICC 09)

[8] N.Agarwal, S.Hussain, *A closer look at intrusion detection systems for web applications*, Hindawi Security and Communication Networks, 2018

[9] O.Can, M.Osman, E.Sezer, O.Bursa, B.Erdogdu, *An ontology-based approach for host intrusion detection systems*, 11th International Conference on Metadata and Semantic Research, Tallin, Estonia2017

[10] A.Razzaq, A.Hur, F.Ahmad, N.Haider, *Ontology-based application level intrusion detection system using Bayesian filter*, 2nd International Conference on Computer, Communication and Control, Budhapest, Hungary 2009

[11] P.Garcia-Teodoro, J.Diaz-Verdejo, G.Macia-Fernandez, E.Vazquez, *Anomaly-bases network intrusion detection: techniques, systems, challenges*, Elsevier, Computers & Security (2009)

[12] C.Tsai, Y.Hsu, C.Lin, W.Lin, *Intrusion detection by machine learning: A review*, Elsevier, Expert Systems with Applications, vol.36 (2009)

[13] M.Tavallaee, N.Stakhanova, A.Akbar,, *Towards credible evaluation of anomaly-based intrusion detection methods*, IEEE Transactions on Systems, Man and Cybernetics- Part C: Applications and reviews, vol.5 (2010)

[14] *Science of Cyber-security*, JASON Project, MITRE Corporation, McLean, Virginia (2010)

[15] M.Hoque, A.Mukit, A.N.Bikas, *An implementation of an intrusion detection system using a Genetic Algorithm*, International Journal of Network Security and Applications (IJNSA), vol.4 (2012)

[16] R.Sommer, V.Paxson, *Outside the closed world: On using machine learning for network intrusion detection*, IEEE Symposium on Security and Privacy (2010)

[17] P.McDaniel, B.Rivera, A.Swami, *Towards a Science of Secure Environments*, Journal of Security and Privacy, vol.12, pp.68-70 (2014)

[18] A.Oltramari, F.Cranor, J.Walls, *Building an ontology for cyber-security*, 9th International Conference on Semantic Technologies for Intelligence, Defense and Security, Fairfax, Virginia, USA (2014)

[19] S.Boubaker Ourida, *Implementation of an Intrusion Detection System*, International Journal of Computer Science, vol.9 (2012)

[20] Y.Lasheng, M.Chantal, *Agent-based distributed intrusion detection system*, Proceedings of Second International Symposium on Computer Science and Computational Technologies, Huangshang, China (2009)

[21] F.Vannel, N.Abdennadher, *Introduction to IoT*, https://docplayer.net/29316309-Introduction-to-iot-1.html

[22] N.Guarino, F.Welty, *Evaluating ontological decisions with OntoClean*, Communications on ACM, vol.45, pp.61-65

[23] P.Aggarwal, S.Sharma, *Analysis of KDD Dataset attributes - class wise for intrusion detection*, Procedia Compuer Science, vol.57, pp.842-851, Elsevier journal (2015)

[24] *2019 Open Source IDS tools - Snort, Suricata, Bro*, https://www.alienvault.com/blogs/security-essentials/open-source-intrusion-detection-tools-a-quick-overview, october 2018

[25] *MAEC: Malware Attribute Enumeration and Characterization*, https://maecproject.github.io/documentation/overview/

[26] *CAPEC: Common Attack Pattern Enumeration and Classification*, https://capec.mitre.org/

[27] *WASC Threat Classification*, http://projects.webappsec.org/w/page/13246978/Threat

[28] *SCAP: Security Content Automation Protocol*, https://www.open-scap.org/security-policies/scap-security-guide/

[29] M.Zweig, G.Campbell, *Receiver-operating Characteristics plots: A fundamental evaluation tool in clinical medicine*, Clinical Chemistry, vol.39, no.4, 1993

[30] A.Zamfira, H.Ciocarlie, *Developing an ontology for cyberoperations in computer networks*, Proceedings of 14th International Conference on Intelligent Computer Communication and Processing (ICCP'18)

[31] *NVD: National Vulnerabilities Database*, https://nvd.nist.gov/

Edited by: Dana Petcu
Received: Jul 28, 2019
Accepted: Oct 19, 2019

AIMS AND SCOPE

The area of scalable computing has matured and reached a point where new issues and trends require a professional forum. SCPE will provide this avenue by publishing original refereed papers that address the present as well as the future of parallel and distributed computing. The journal will focus on algorithm development, implementation and execution on real-world parallel architectures, and application of parallel and distributed computing to the solution of real-life problems. Of particular interest are:

Expressiveness:
- high level languages,
- object oriented techniques,
- compiler technology for parallel computing,
- implementation techniques and their efficiency.

System engineering:
- programming environments,
- debugging tools,
- software libraries.

Performance:
- performance measurement: metrics, evaluation, visualization,
- performance improvement: resource allocation and scheduling, I/O, network throughput.

Applications:
- database,
- control systems,
- embedded systems,
- fault tolerance,
- industrial and business,
- real-time,
- scientific computing,
- visualization.

Future:
- limitations of current approaches,
- engineering trends and their consequences,
- novel parallel architectures.

Taking into account the extremely rapid pace of changes in the field SCPE is committed to fast turnaround of papers and a short publication time of accepted papers.

INSTRUCTIONS FOR CONTRIBUTORS

Proposals of Special Issues should be submitted to the editor-in-chief.

The language of the journal is English. SCPE publishes three categories of papers: overview papers, research papers and short communications. Electronic submissions are preferred. Overview papers and short communications should be submitted to the editor-in-chief. Research papers should be submitted to the editor whose research interests match the subject of the paper most closely. The list of editors' research interests can be found at the journal WWW site (`http://www.scpe.org`). Each paper appropriate to the journal will be refereed by a minimum of two referees.

There is no a priori limit on the length of overview papers. Research papers should be limited to approximately 20 pages, while short communications should not exceed 5 pages. A 50–100 word abstract should be included.

Upon acceptance the authors will be asked to transfer copyright of the article to the publisher. The authors will be required to prepare the text in $\LaTeX\,2_\varepsilon$ using the journal document class file (based on the SIAM's `siamltex.clo` document class, available at the journal WWW site). Figures must be prepared in encapsulated PostScript and appropriately incorporated into the text. The bibliography should be formatted using the SIAM convention. Detailed instructions for the Authors are available on the SCPE WWW site at `http://www.scpe.org`.

Contributions are accepted for review on the understanding that the same work has not been published and that it is not being considered for publication elsewhere. Technical reports can be submitted. Substantially revised versions of papers published in not easily accessible conference proceedings can also be submitted. The editor-in-chief should be notified at the time of submission and the author is responsible for obtaining the necessary copyright releases for all copyrighted material.