An Ensemble Intrusion Detection Model For Internet of Things Network

Sarika Choudhary
Central University of Rajasthan
Nishtha Kesswani (nishtha@curaj.ac.in)
Central University of Rajasthan https://orcid.org/0000-0002-9044-4160
Sudhan Majhi
Indian Institute of Technology Patna

Research Article

Keywords: Internet of Things (IoT), Support vector machine, Deep Neural Network, Intrusion detection, Prevention

Posted Date: June 3rd, 2021

DOI: https://doi.org/10.21203/rs.3.rs-479157/v1

manuscript No.
(will be inserted by the editor)

An Ensemble Intrusion Detection Model for Internet of Things Network

Sarika Choudhary · Nishtha Kesswani* · Sudhan Majhi

Received: date / Accepted: date

Abstract The advancements of technology are playing a significant role in protecting the data from intruders. In this paper, a robust network intrusion detection system (IDS) is proposed for Internet of Things (IoT) using deep learning approaches. The type of intrusions we adopted in this work are distributed denial of service (DDoS) and replay attack. Our proposed work is divided into three sections, namely, node deployment, threat detection modelling, and prevention modelling. For detection, ensemble algorithm has been used, i.e., deep neural network (DNN) and support vector machine (SVM). SVM is used to identify the suspected route and DNN is used to identify the suspected node out of suspected routes. The chosen route ensures that it is prevented from attackers by incorporating the throughput and packet delivery ratio (PDR). The simulation results are obtained on the basis of accuracy, recall, precision, and F-measure to determine the effectiveness of the proposed approach. The precision, recall, F- measure, and accuracy of correctly identified intruders are 98.12%, 98.04%, 94.88%, and 98.68%, respectively, which is an improvement over the previous studies. The efficacy of the designed model for IoT is compared with the existing approaches.

Keywords Internet of Things (IoT) · Support vector machine · Deep Neural Network · Intrusion detection · Prevention.

S. Choudhary · Nishtha Kesswani *
Department of Computer Science,
Central University of Rajasthan,
Ajmer, Rajasthan, India-305817
S.Majhi
Indian Institute of Technology,
Patna, India
*E-mail: nishtha@curaj.ac.in

1 Introduction

With the development of communication technologies, billions of Internet of Things (IoT) devices are currently connected and it is expected there will be tremendous increase in the number of devices within the next four years [Hassija et al.(2019a)Hassija, Chamola, Saxena, Jain, Goyal, and Sikdar]. The main applications of IoT are home automation, smart car, smart cities, smart metering, health sensors etc. IoT-based networks are formed of low-power and lossy networks (LLNs), which are composed of various heterogeneous wireless technologies, such as radio-frequency identification (RFID) tags, sensors, actuators, etc. [Gubbi et al.(2013)Gubbi, Buyya, Marusic, and Palaniswami]. IoT devices are characterised by both their resource constraints and lossy communication links. Indeed, these objects have limited processing power, memory, and energy supply, in addition to a high loss rate, low throughput, limited frame size, and short communication ranges [Hui and Culler(2008), Ammar et al.(2018)Ammar, Russello, and Crispo]. The limitations raise several challenges such as scalability, routing, and security for industry and academic research community. Availability of services also opens the gate for the intruders, thus, the security is an important concern nowadays. It is a feature of any system that indicates whether a network is free from risk or not [Choudhary and Kesswani(2019)]. Total security is a utopia but there are other alternates by which a network can be secured.

Therefore, intrusion detection system (IDS) has been an important tool for the protection of IoT networks. It is used to detect intrusion based on behavior, architecture and analysis strategy. Implementing IDS in IoT devices is difficult due to its typical features such as resource-constrained devices, specific protocol stacks and standards. Therefore, various protocols such as routing protocol for low power and lossy network (RPL) [Winter et al.(2012)Winter, Thubert, Brandt, Hui, Kelsey, Levis, Pister, Struik, Vasseur, and Alexander], IPv6 over low power wireless personal area network (6LoWPAN) [Kushalnagar et al.(2007)Kushalnagar, Montenegro, and Schumacher], IEEE 802.15.4 [Montenegro et al.(2007)Montenegro, Kushalnagar, Hui, and Culler], message queuing telemetry transport (MQTT) [Quincozes et al.(2019)Quincozes, Emilio, and Kazienko], constrained application protocol (CoAP) [Shelby et al.(2014)Shelby, Hartke, Bormann, and Frank] can be utilized to enhance the security level.
Due to intrinsic resource and computational constraints, traditional security methods cannot directly be applied to secure IoT systems. Existing research on intrusion detection for the IoT is largely focused on rule-based detection techniques [Le et al.(2012)Le, Loo, Lasebae, Aiash, and Luo,Raza et al.(2013)Raza, Wallgren, and Voigt] but they failed to detect new type of attacks effectively. For an efficient detection of zero-day threats, anomaly-based detection techniques are essential especially in emerging IoT environments. In the last decades, anomaly-based intrusion detection and other classification problems have been solved with the idea of combining multiple classifiers [Aburomman and Reaz(2016)]. Also, in an IoT ecosystem, machine-learning algorithms can be useful to perform automated data analysis and provide meaningful interpre-

tations and predictions about the system, where many devices are constantly generating huge amount of data. Use of the machine learning for IoT security is especially very promising in detection of any outliers to normal activity in the system [Thamilarasu and Chawla(2019), Chaabouni et al.(2019)Chaabouni, Mosbah, Zemmari, Sauvignac, and Faruki, Zolanvari et al.(2019)Zolanvari, Teixeira, Gupta, Khan, and Jain].

Achieving security in IoT is difficult because of its architecture in which sensor nodes are accessible globally and the IPv6 border router (6BR) is also accessible. The establishment of IDS in IoT is a challenging issue due to the global access of the resources, constrained devices, connection through lossy links, and use of standard protocols such as RPL and CoAP. Hence, an attempt has been made in this paper to generate an intrusion detection system for detecting of malicious activities.

The present work describes a deep learning approach to determine the intrusion detection in IoT. The main motivation behind this research is to detect intruders which affect the network performance. In order to ensure the IoT network performance, throughput and PDR are derived. PDR is the ratio of packets successfully received to the total sent and throughput is the rate at which information is sent through the network. Both parameters are important to evaluate the network performance. In a IoT network, each node can easily join and leave the network anytime. This makes the IoT network susceptible to various attacks. However, the conventional techniques (such as firewall, DMZ (demilitarized zone), honey pot, rules and policies, user authorization etc.) are no longer effective and sufficient in protecting the network from intruders. Moreover, the common data encryption and authentication techniques seem to be insufficient in preventing the IoT network from malicious attackers. In such a case, the secure transmission of data packets is essential. Therefore, an intrusion detection system required which assures to protect the network from attackers.

The major contributions of this paper are as follows:

1. We propose a two phase model i.e., support vector machine (SVM) and deep neural network (DNN) for intrusion detection with minimum time and better accuracy.
2. A cosine similarity is introduced instead of Euclidian distance in RPL for route discovery process. By using similarity values of the nodes we made effective routes.
3. Simulation analysis and performance evaluation in terms of accuracy, precision, recall, F-measure, over 10-1000 simulation rounds and 50 to 150 nodes show that the proposed work proved to be efficient over other existing approaches.
4. Additionally, we compare our results against the existing work reflects the practical application of the proposed strategy for the deployment of a trustworthy and secure communication platform.

The paper contributes towards the development and deployment of a distributed maliciousness detection model.

Rest of the paper is organized as follows: Section 2 describes the related works. Section 3 gives the detailed description of the proposed work and algorithm. This section is followed by the performance analysis in Section 4. Section 5 explains the simulation results and performance evaluation. Conclusion of our work is mentioned in Section 6.

2 Related Work

Security and privacy have been significant concerns in IoT that have been posing barriers for its adoption and the device development [Sfar et al.(2018)Sfar, Natalizio, Challal, and Chtourou]. We are categorizing according to their detection method.

2.1 Anomaly, Signature and Specification-based IDS

Several studies focusing on IoT security have attempted to design IDS systems. Raza *et al.* [Raza et al.(2013)Raza, Wallgren, and Voigt] defined real-time IDS for IoT by implemented on the Contiki operating system and marked routing attacks such as selective forwarding and sinkhole. By using their technique, identify the malicious nodes that launch routing attacks. Kasinathan *et al.* [Kasinathan et al.(2013)Kasinathan, Costamagna, Khaleel, Pastrone, and Spirito] proposed a system that is signature-based IDS for IoT network . The main purpose of their paper was to detect attacks like DoS that have a low false positive rate. The need for this system was because although many attempts were made for security, IoT has been a target for malicious actors, and now it has become the necessity to shift our focus for early detection of intrusions so that the negative impacts on the system can be reduced. Pongle *et al.* [Pongle and Chavan(2015)] proposed a centralized and distributed architecture detecting routing attacks such as the wormhole attack. Jun *et al.* [Jun and Chi(2014)] presented a complex event-processing-based IDS for the IoT. It is more efficient than traditional IDS. This system is specification-based, and it uses complex event processing techniques for attack detection. However, it is CPU intensive as they use rule pattern repository.
Merlo *et al.* [Merlo et al.(2015)Merlo, Migliardi, and Caviglione] proposed measurement of energy at various levels for mobile devices to gain trade-off between the energy-based profile of malware and measurement precision. Yang *et al.* [Yang and Tang(2016)] proposed a model to generate GMM (Gaussian Mixture Model) that uses energy consumption frequency. It was based on the Mel frequency cepstral coefficient used for malware detection. In the paper, the authors have used a statistical approach for decision making based on power usage. Frequency of wavelength is also employed in this approach; thus, changes in the performance of CPUs would have an impact on the result, although the waveform visual form is unchangeable. Shaerpour *et al.* described

the advanced detection method based on malware's properties, their tracking, and energy consumption [Shaerpour et al.(2013)Shaerpour, Dehghantanha, and Mahmod]. Hamed *et al.* [Pajouh et al.(2016)Pajouh, Javidan, Khayami, Ali, and Choo] provided a two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in IoT network. They used the combination of naive bayes and K-nearest neighbor classification for an intrusion detection in KDD-Cup dataset. Their work shows highest accuracy to identify user to root attack(U2R), Remote to local attack (R2L) anomalies in the dataset.
Kolias *et al.* [Kolias et al.(2015)Kolias, Kambourakis, Stavrou, and Gritzalis] suggested a signature-based or anomaly-based scheme for attack the detection. In the signature-based system, it matches the traffic that is incoming with the known attacks saved in the database, whereas in the case of the anomaly-based system, it is a behavior-based detection. This approach is popular as it is highly accurate for detection, has low false rate alarm, but it is unable to detect novel attacks. Anomaly detection mainly captures new attacks, but, it lacks accuracy. Stephen *et al.* [Stephen and Arockiam(2017)] suggested a lightweight, hybrid, and centralized approach aiming to detect Hello Flood and Sybil attacks in IoT networks, which use the RPL as a routing protocol. Their system is based on an algorithm that uses detection metrics, such as a number of packets received and transmitted to validate the intrusion ratio (IR) by the IDS agent.
The features of IoT have been fascinating various other application areas like the HealthCare sector, smart cities, etc. [Catarinucci et al.(2015)Catarinucci, de Donno, Mainetti, Palano, Patrono, Stefanizzi, and Tarricone]. The unknown cyber-attacks have been a hindrance to the adoption of these services. The distributed system of IoT services has made the security of IoT difficult, and this is a challenging aspect. Practitioner states that in comparison to other systems, the phenomena of detecting attacks in IoT are quite different, and this is because of the specific tasks required by the system [Alrawais et al.(2017)Alrawais, Alhothaily, Hu, and Cheng]. This issue cannot be resolved by simply centralizing the cloud, scalability, or mobility.
Researchers suggested that this problem will not be solved either by detectors used for standalone attack or by traditional approaches and for this, a distributed intelligence approach like machine learning should be studied to cover up this gap [Hassija et al.(2019b)Hassija, Chamola, Saxena, Jain, Goyal, and Sikdar,Liang et al.(2019)Liang, Hatcher, Liao, Gao, and Yu].

2.2 Machine Learning-based IDS

Malware Detection is one of the challenging task, and there are several ongoing research to mitigate these [Faruki et al.(2014)Faruki, Bharmal, Laxmi, Ganmoor, Gaur, Conti, and Rajarajan]. Niyaz *et al.* [Javaid et al.(2016)Javaid, Niyaz, Sun, and Alam] have suggested that traditional systems failed to identify a slight change in attack from the earlier attacks. The deep learning tech-

nique used in big data systems can be used for cyber threats as these small changes are easily visible, for example, changes in image pixels. Damshenas *et al.* [Damshenas et al.(2013)Damshenas, Dehghantanha, and Mahmoud] suggested that the energy footprint is robust against malware detection and anti-forensic techniques. Diro *et al.* [Diro and Chilamkurti(2018)] had implemented a deep learning model for intrusion detection in social IoT's that was evaluated against shallow learning. In spite of demonstrating high detection accuracy, the implemented technique proved to be very time-consuming at both the training and testing stage as compared to the shallow learning technique. Caspi *et. al.* [Caspi(2017)] proposed that, as the number of attacks increases, traditional methods are unable to detect difficult breaches. Most of these attacks are the updated versions of the previous attacks. The new attacks are based on 1% previous attack and logic. Detection system based on a fuzzy mean algorithm and Quantum-behaved Particle Swarm Optimization algorithm for finding the global optimum solution. Lin *et al.* [Lin et al.(2015)Lin, Ke, and Tsai] proposed a CANN approach along with k-NN classifier to detect intrusion. First, clustering is done for creating training subsets. In the next step, the analysis of the main components is done using (PCA) for the selection of relevant features. Then the grouping is done, and k-NN classifier is used for classification. Azmoodeh *et al.* [Azmoodeh et al.(2018)Azmoodeh, Dehghantanha, Conti, and Choo] presented an approach to detect ransomware attacks by keeping the power consumption track of android devices. Their approach shows that it gives better results than SVM, k-NN, Neural Network, and Random Forest (RF). They had designed a ransomware detection protocol for IoT that could distinguish non-ransomware applications from ransomware applications only by utilizing the energy consumption parameter. Moustafa *et al.* [Moustafa et al.(2018)Moustafa, Turnbull, and Choo] did an in-depth analysis of the TCP/IP model to identify the set of features. They used HTTP, MQTT, DNS, and their flow identifiers to make an effective network intrusion detection system (NIDS). They used Bro-IDS for feature extraction to take the basic information of the protocols. They used AdaBoost method with naive bias (NB), decision tree (DT), and artificial neural network (ANN) to improve the performance of the approach in terms of accuracy, detection rate, and time processing. Li *et al.* [Li et al.(2018)Li, Zhao, Li, and Zhang] proposed an artificial intelligence-based two-stage intrusion detection empowered by software-defined technology. Firstly, to select the features, they used Bat algorithm with swarm division and binary differential mutation. After that, to classify the flows, they used random forest (RF) through adaptively altering the weights of samples by using the weighted voting mechanism. An ensemble approach designed by Aburomman *et al.*. [Aburomman and Reaz(2016)] took advantage of particle swarm optimization (PSO), local unimodal sampling (LUS) meta-optimization, and weighted majority algorithm (WMA) based intrusion detection to present safe and secure networking. Experimental results demonstrated the highest detection accuracy by LUS meta-optimized technique followed by PSO and least by WMA based technique. However, LUS method consumes a larger time for execution as compared to PSO based

method and hence was not a wiser consideration to design classifiers ensemble to develop and deploy an intrusion detection system.
The associated flaw of Aburomman *et al.* [Aburomman and Reaz(2016)], Diro *et al.* [Diro and Chilamkurti(2018)] and Azmoodeh *et al.* [Azmoodeh et al.(2018)Azmoodeh, Dehghantanha, Conti, and Choo] were addressed in the our study with the implementation of SVM to add vigour at classification stage while demonstrating higher performance parameters. In this context, time consumption and energy consumption evaluation were performed.
Multiple threat detection and prevention models have been proposed and are discussed in the literature survey as well. The existing works have primarily worked on threat specific detection and prevention models. This paper contributes to detecting the security threats in the network and the proposed two phase-detection model that is free from security threats.

3 Proposed Work

The proposed IDS aims at preventing the data loss during data transfer through a wireless network. The effectiveness for secure data transfer get challenged due to presence of malicious nodes (attacked points) within the IoT network. It significantly compromises the legitimate communication in IoTs. Hence, aim is to detect the malicious nodes from the network so that we can prevent network from the data loss. The proposed work is divided into three sections, namely node deployment, threat detection modelling, and prevention modelling.

The node deployment model assumed to be heterogeneous where each node has different attributes. The nodes are assumed to be wireless in nature. The number of nodes used in the simulation is 50-150, and the model is simulated for 10-1000 simulations. The nodes are deployed under the routing protocol for low power and lossy network. The nodes are positioned on x and y coordinate axis, and the nodes choose a source node and a destination node randomly. Every node with initial energy ϵ_0 and transmission range R_t can detect the next target node within the transmission radius. Every node has its transmission range which is 20% of the network size, considered in our study. The source to destination path delivery is based on the cosine similarity [Li et al.(2016)Li, Chen, Liu, and Min] between the nodes, and close similarity value results in adding the nodes in the route.

The network parameters of the simulation are listed in Table 1.

Fig. 1 demonstrates the route discovery process. It is seen in the given representation in Fig. 1 that the similarity between the nodes in the region is calculated using the similarity computation.

The cosine similarity is also called the cosine distance. Compared to Euclidean distance, cosine distance is more useful in this context as it calculates the difference between two vectors taking their direction into consideration. Euclidean distance is used to measure the absolute distance of each point in a space, where each point with coordinates is directly related. The cosine dis-

Table 1: Network Parameters.

Attributes	Values
Network Area	1000 * 1000 m^2
Number of Nodes	50 - 150
Iterations	10^3 simulations
Similarity Measures	cosine similarity
Transmission Range	200 m
Routing Protocol	RPL
Classifiers	SVM and DNN
Total Simulation Time	3350 sec

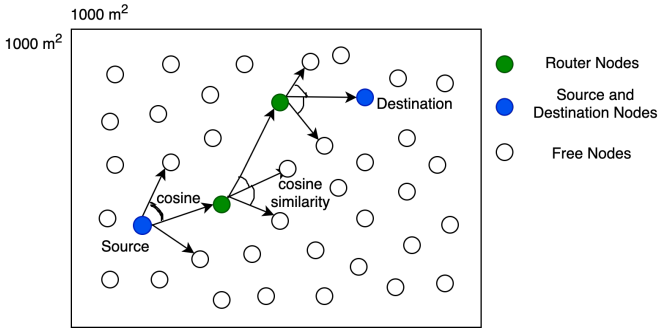


Fig. 1: Route discovery using similarity calculation.

tance measure is the angle between the vector space, more strongly reflecting differences in the direction, rather than the position. The cosine similarity is calculated by using the global positioning system co-ordinates. The GPS co-ordinates of a node is its location in the network that has been retrieved by using the node's position. When a node is plotted in the network then position has x and y value and these coordinates considered as its GPS coordinates. Let position of node N_1 is (x_1, y_1) be the source node and position of node N_2 is (x_2, y_2) whose similarity is to be calculated then the cosine similarity C_{sv} is calculated as:

$$C_{sv} = \frac{(x_1, y_1) * (x_2, y_2)}{||x_1, y_1|| * ||x_2, y_2||}. \tag{1}$$

The cosine similarity is in the range of [-1,1], the similarity value of -1 means that data are precisely opposite, 0 meaning independent, 1 meaning the same, and in-between values indicating common similarities or dissimilarities. The minimum similarity index value is added to the path. The process is repeated until the destination is not attained. A total of 1000 data packets are initialized at the time of simulation by each node.

Every node dumps some amount of data packets when the data is to be transferred from the node. Nodes also consume some amount of energy at the

time of receiving the data packet termed as E_{rx} and some amount of energy at the time of data transfer E_{tx}. Some aggregation energy is also consumed by each node when it starts to prepare the node to transfer the data, which is termed as E_{ax}. The total energy consumption C_{en} by each node is defined as :

$$C_{en} = E_{tx} + E_{rx} + E_{ax}. \tag{2}$$

Step by step implementation architecture of proposed work is as follows:

1. Initilaize nodes with geo-stationary co-ordinates (x,y).
2. Intialize QoS parameters (T_h = [], PDR = [], C_{en} = [], Delay = []) as empty.
3. Plot network.
4. Select source and destination node randomly.
5. Apply route discovery using RPL with cosine similarity.
6. Apply intrusion detection algorithm [DDoS and Replay Attack] see Algorithm 1
7. Evaluate energy consumption, T_h, PDR and Delay after every route discovery and data transfer.
8. Intialize detection algorithm with QoS parameters as input to the structure.
9. Count all non-classified nodes (suspected nodes).
10. Find maximum occurrence of non-classified nodes.
11. Block the intruded node(s) as preventing network from them.
12. Calculate accuracy, precision, recall, F-measure as performance evaluation parameters.

To check the robustness of the proposed algorithm, the probability of threat in the network is kept as 0.5, which means that in each simulation round, after every 1 ms, the probability of security breach is 50%. When a route is formed, the attack occurs in that phase only because the attacker would not like to waste its effort on an empty track. The attacker aims to harm the system with maximum payload. A dual-phase threat detection mechanism is designed to prevent the network. In this scenario, SVM is used to identify the suspected route and deep neural network (DNN) is used to identify the suspected node out of suspected routes. The algorithm uses throughput as a decision making architecture by training the classification algorithms. The training architecture calculates the relative similarity weight and passes to the classification layer in order to check whether the current evaluated throughput is related to intrusion or normal. If the throughput of a node drops or high then it will be used as decision maker to identify the intrusion or normal behavior.

SVM takes the throughput and PDR of every path as the input and uses a supervised learning mechanism with the polynomial kernel. If the classified data of SVM classification architecture is equal to the training data, the route is set to be free from intrusion and no further processing needs to be done on the nodes of the route. SVM sets the throughput and PDR in two-dimensional

array with the route ID as the associated label. As SVM is a binary classifier, it can only classify into two classes and hence each route is classified in a separate class. The classification data that was stored in the database is further used to train and test the data.

- If the attained PDR and throughput of that route are high then it shows that the path is clear and packets can be easily passed through the network.
- However, the lower PDR and throughput of the system shows that the path is suspected and requires an intruder detector system to free from the malicious nodes.

The classified route is then passed to DNN with energy consumption. In DNN, we took 20 hidden layers in the starting and varied till 100 and used sigmoid activation function.

Intrusion detection process is a two phase detection process.

The SVM approach first classifies the path followed by the detection of the node for each path using deep learning. After the two phases, the highest affected node will be counted as intruder. The notations used in Algorithm 1 are presented in Table 2.

Table 2: Notations used in Algorithm 1.

Notation	Description
R_f	Route frame
I_p	Input parameters to the SVM
T_h	Throughput
PDR	packet delivery ratio
T_e	Target elements of SVM
T_s	Training structure of SVM
C_r	Classified route based on training structure of SVM
C_e	Energy consumed by node
A_f	Activation function used in DNN
C_n	Classified node based on training structure of DNN
Tr_n	Training structure of nodes that are suspected from classified route
H_n	Hidden neuron count
S_r	Suspected routes
S_n	Suspected nodes

The proposed ensemble intrusion detection algorithm is as given in Algorithm 1:

Algorithm 1 Intrusion Detection Algorithm.

for each path in R_f **do**
  $I_p = T_h, PDR$
  $T_e = Route_{id}$
  Intialize SVM Structure
  $Kernel_{type} = \frac{Polynomial}{ax^2+bx+c=0}$
end for
$T_s = Plot_{kernel}(I_p, T_e, Kernel_{type})$
$Test_{data} = I_p$
$C_r = Simulate(Test_{data}, T_s)$
if C_r in each route structure does not match with the T_e **then**
  S_r++
end if
for each suspect route in $Suspected_{route}$ **do**
  for each node in $Suspected_{route}$ **do**
    $I_p(node) = C_e, Packet_loss$
    $T_e = Node_{id}$
    Intialize layers of Neural Network
    $Hidden_{layers} = 20 - 100$
    $A_f = Sigmoid$
  end for
end for
$Tr_n = Propagate(I_p, H_n, A_f)$
$Test_{data} = I_p$
$C_n = Simulate(Test_{data}, Tr_n)$
if C_n in each $route_{node}$ does not match with the T_e **then**
  S_n++
end if
Calculate maximum occurrence of nodes.

Algorithm 1 describes the identification of malicious nodes from the malicious path. Firstly, we initialized the network with default parameters of nodes. Then find out the source and destination node randomly and made a path between them by using RPL with minimum distance value (using cosine similarity index). We are taking random sources and destination nodes in the network because, in real life, the objects would be moving, we cannot fix them in the IoT network. We check each path in all route frames and take input parameters as throughput and PDR. Our target element is route IDs for each path. Next, we initialize SVM structure and kernel type (linear/ polynomial). Then train SVM with two features and save the training values. Now, choose test data as input parameters and classify the paths with SVM. If classified routes in each route structure match with the target elements (i.e., route ID), then the path is clear otherwise considered as a suspected route.

The proposed algorithm can be explained by the following example. Let there be 5 different routes (for 150 nodes in the network) as follows:

Table 3: Routes

Route ID	Nodes					
1	23	25	41	44	45	
2	31	36	39	120	115	
3	52	55	48	96	91	23
4	41	48	49	95	96	99
5	92	85	110	120	116	

Here, 23, 31, 52, 41 and 92 are the source nodes; 45, 115, 23, 99 and 116 are the destination nodes and others are intermediate hops. In the deployed network there are three types of nodes, first, Source nodes: from where data need to be transferred. Second is destination nodes: where data need to be transferred and act as receiver. Third is hope nodes: these are the intermediate nodes present between source and destination nodes. Throughput and PDR are input parameters for this algorithm and hence the training data would be as follows:

Table 4: Input Parameters

Route ID	Throughput	PDR
1	1245	.75
2	1148	.74
3	1152	.73
4	854	.62
5	896	.554

As a supervised learning approach, all the routes will be classified first. The performance of individual nodes is also stored. Like as for example, 1245 is the throughput of the route 1 (i.e., 23 25 41 44 45). The throughput of individual nodes are also stored other than source and destination. The identified suspect routes are passed to DNN. A series of nodes starting from 50 and goes up-to 150 in the simulation.

When we have a list of suspected routes, then for each suspected route, we classified the suspected node. Input parameters for testing of suspected nodes is energy consumption. In DNN, we took 20 hidden layers in the starting and varied till 100 and used sigmoid activation function. Now again, train our network with the energy consumption of each node in the path. Test data is input parameters of nodes. We classified the nodes by simulating the network. If classified node structure in each suspected route does not match with the target elements (i.e., node ID), then take that node as a suspected node.

When we have all the suspected nodes, then node(s) with the highest count is considered as an intruder. If a node is involved in the maximum number of misclassifications by the proposed algorithm, it is evident that it would be an intruder. Yes, there may be other intruder nodes, but as the simulation is check point-based and detecting all intruders at once would consume much time and would be power draining for resource-constraint nodes in the IoT. How accurately our algorithm detects intruder nodes is calculated by the performance parameters.

There are two kinds of behaviour for intrusion which may occur in the network. In the first behaviour, the intruder acts himself in the network and in the second behaviour, the intruder affects other nodes in the network. Our paper considers second behaviour. The nodes are physical and hence when the node is identified as intruder, it is either rectified, replaced or blocked. In this paper, the highest counted nodes will be blocked for prevention. After preventing network from attack, we have calculated throughput and packet delivery ratio and got higher values as compared to before prevention.
In the IDS, Throughput, PDR and energy consumption are used for decision making to evaluate the deployed network for secure data transfer.

- Route Property: Based on the throughput and PDR parameters every possible network path is evaluated and classified as either attacked path or genuine path.
- Route Node Property: Based on the energy consumption parameter, property of each node in the attacked path (as classified by SVM) is identified by deep learning architecture and each node is labelled as malicious or normal node. In case all the nodes in the attacked path exhibit high energy consumption, the node with highest parametric value is considered the intruder.

The effectiveness of the proposed algorithm is further evaluated by calculating the precision, recall, accuracy, and F-measure as discussed in the next section.

4 Performance Analysis

We have designed an algorithm that gives a generic solution for all kind of attacks. Two kinds of attacks have been considered in our network for evaluation process, i.e., distributed denial of service (DDoS) and replay attack. Whenever attack happens in the network it always tries to slow down the performance of the network. Every attacker node will dump packets more than usual, so, throughput and PDR are good parameters to consider for identifying malicious activity in the network.

The results are evaluated on the basis of throughput, packet delivery ratio and energy consumption. Analysis has been done with performance parameters like accuracy, recall, precision, and F-measure. Throughput is a measure of how

many units of information a system can process in a given amount of time. It is defined as:

$$T_h = \frac{P_r}{T_e}. \tag{3}$$

The total received packets at the destination, P_r, can be defined as:

$$P_r = P_s - P_d. \tag{4}$$

Where, T_h denotes throughput, P_r denotes received packets at the destination end, T_e denotes Total elapsed time, P_s denotes total sent packets and, P_d denotes dumped packets.

The ratio of packets that are successfully delivered to a destination compared to the number of packets that have been sent out by the sender called packet delivery ratio. PDR is defined as:

$$PDR = \frac{P_r}{P_s}. \tag{5}$$

Precision is the ratio of classified attacks to the number of attacks recorded which is represented as:

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

Recall is defined as the measure of number of attacks identified correctly when the intruders attacked on the system. It is also called True Positive Rate. The recall is given as:

$$Recall = \frac{TP}{TP + FN}. \tag{7}$$

F-measure is the harmonic average of the precision and recall. It defines the effectiveness of the developed technique in detecting the intruder attacked on the system. The F-measure is defined as:

$$F - measure = 2 * \frac{Precision * Recall}{Precision + Recall}. \tag{8}$$

Accuracy is defined as the percentage of true detection over total instances. The accuracy is given as:

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FN}. \tag{9}$$

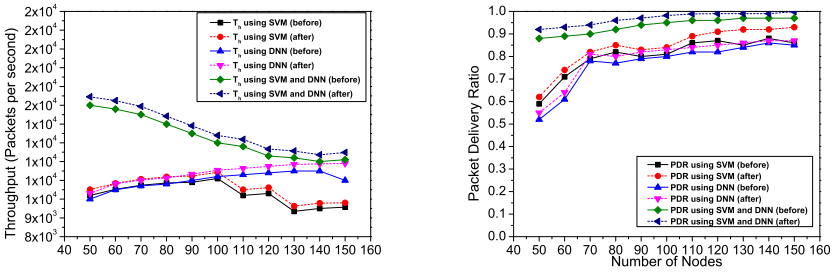
We have calculated throughput and PDR before and after the attack prevention. This also varies on basis of the network size. If the network size is less, then probability of finding attacks is more. Therefore, throughput and PDR increases with increase of the network size. Because, as the network size increases then the throughput and PDR rate decreases due to noise and packet losses.

5 Simulation Results and Discussion

The simulations has been conducted using MATLAB version R2016b with a 1.8 GHz Dual-Core Intel Core i5 processor, 8 GB of RAM and macOS Catalina operating system.
We have developed hybrid algorithm for intrusion detection and we have tried with individual mechanisms with either SVM applied individually for malicious path and node discovery or DNN applied separately for the same. This research work aims to target DDoS , DoS , Replay as they are of similar kind and behavior. Distributed Denial-of-Service (DDoS) forms the most common type of network attack in wireless networks. It is one of the fast elevating issues in the digital world. Therefore, the effectiveness of the proposed design is evaluated for intrusion detection, i.e., identifying attacked path and malicious node within the attacked path to offer a secure data transfer during DDoS attack. Evaluation against a single attack instance is not enough to justify the performance of the proposed design therefore another popular network attack, Replay attack is considered for experimentation. This is a frequently employed attack in which the capability of an attacker is restricted to one node and attack is characterized by repeated data transmission and delay that renders full gain over the network and information which was otherwise not easily accessible.

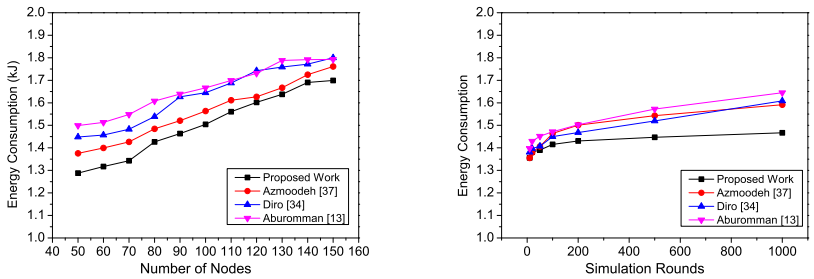
Throughput and PDR computation of the proposed work before and after preventing from the attacks shown in Fig. 2. Fig. 2a shows the throughput of the proposed work with respect of different nodes before and after attack prevention. The average throughput value using DNN before preventing from attacks is 10990, throughput using SVM is 10295 and SVM and DNN is 13290. Thus, the combined average value of SVM and DNN provides better results as compared to the average value of throughput using SVM and DNN individually. Dashed lines in the graph shows the throughput obtained after prevention from attacks. The average throughput value obtained using DNN is 11375, throughput using SVM is 10599 and combination of SVM and DNN is 13692. Thus, we observed that the combined average value of SVM and DNN still provides better results as compared to the average value of throughput using SVM or DNN after preventing from attacks.

Fig. 2b shows the PDR computation for different number of nodes before and after preventing from attacks. It is clearly seen that the average PDR using the DNN and SVM before preventing from attacks is 0.76 and 0.80. The PDR obtained through the proposed work using combined DNN and SVM is 0.94. Thus, PDR of the proposed work outperforms the individual machine learning techniques. Dashed lines in the graph shows the PDR computation for different number of nodes after preventing from attacks. It is clearly seen that the average PDR using the individual DNN and SVM is 0.79 and 0.84 respectively. The PDR obtained through the proposed work using combined DNN and SVM is 0.99.

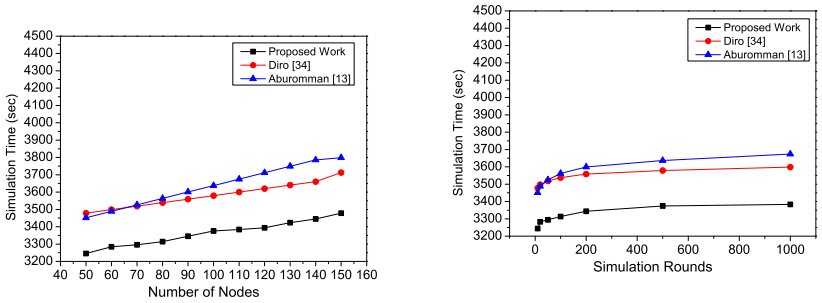


(a) Throughput computation. (b) Packet Delivery Ratio computation.

Fig. 2: Throughput and PDR computation before and after preventing from attacks.



(a) Energy consumption over number of nodes. (b) Energy consumption over simulation rounds.



(c) Time consumption over number of nodes. (d) Time consumption over simulation rounds.

Fig. 3: Energy and Time Consumption over number of nodes and number of simulation rounds.

SVM is a binary classifier and it maintains its stability up to a certain load point. When the weight either in terms of communicating nodes or expansion area increases, it gets too much of data near the kernel and as a result, it

may or may not choose the best values for training and hence the performance may go down all of a sudden and vice-versa. The behaviour of DNN can show surprising characteristics. For the same set of data, it might need less or more number of propagation layers but the learning gets settled at some point for sure. As shown in Fig. 2, DNN behaviour for throughput (or PDR) takes a leap behaviour until it did not settle down where as SVN started showing dramatic rise and fall on the attribute value.

Fig. 3a and Fig. 3b compares the energy consumption against the employed 50 to 150 nodes and simulation rounds varying from 10 to 1000. Fig. 3a shows that with increase in the number of nodes energy requirement increases for all the models, proposed as well as existing ones. However, it is observed that on an average minimal energy consumed by the proposed work i.e., 1.5 kJ as compared to 1.66 kJ, 1.63 kJ, 1.56 kJ and 1.6 kJ by [Aburomman and Reaz(2016)], [Diro and Chilamkurti(2018)], [Azmoodeh et al.(2018)Azmoodeh, Dehghantanha, Conti, and Choo] respectively. Similar observations are also seen when average values of energy consumption are analyzed against simulation rounds as shown in Fig. 3b. It is observed that with increase in simulation rounds, energy consumption rises gradually for all the cases. However, proposed work exhibited least energy consumption over all the simulation rounds performed for varied number of nodes.
The disadvantage of the previous intrusion detection models were time consumption due to the techniques that were used. We tried to overcome this disadvantage in our work. [Aburomman and Reaz(2016),Diro and Chilamkurti(2018)] used time constraint in the papers so we are comparing our results with their results. The Fig. 3c shows that execution time increases with the increase in the number of nodes. However, proposed model exhibited average lower time requirement of 3350.7s as compared to average simulation time of 3594.5s in [Aburomman and Reaz(2016)] and 3569.5s by [Diro and Chilamkurti(2018)]. Further, time consumption was also compared over simulation rounds varying from 10 to 1000. As shown in Fig. 3d, the proposed model required least time with an average value of 3319.5s against 3562.27s in [Aburomman and Reaz(2016)] and 3538.08s [Diro and Chilamkurti(2018)] Overall, the proposed model required lesser time for intrusion detection with enhanced detection accuracy that reflects the success of the proposed design.

Thus, higher throughput and PDR and less energy consumption enhances the further experimental results in the computation of precision, recall, accuracy, and F-measure. The proposed work has been evaluated considering accuracy, precision, recall and F- measure on Y-axis and number of nodes on X-axis.

The different number of nodes has been deployed in the range of 50 to 150. Fig. 4 shows the results of the precision with varied number of nodes. The average precision of the proposed work is 98.06 and that of past research is 85.16, 96.99, and 88.37 in [Azmoodeh et al.(2018)Azmoodeh, Dehghantanha, Conti, and Choo], [Diro and Chilamkurti(2018)], and [Aburomman and Reaz(2016)] model respectively. The improvement in precision percentage of the proposed work is 13%, 1%, and 9.8% respectively over the existing work. Thus, improved

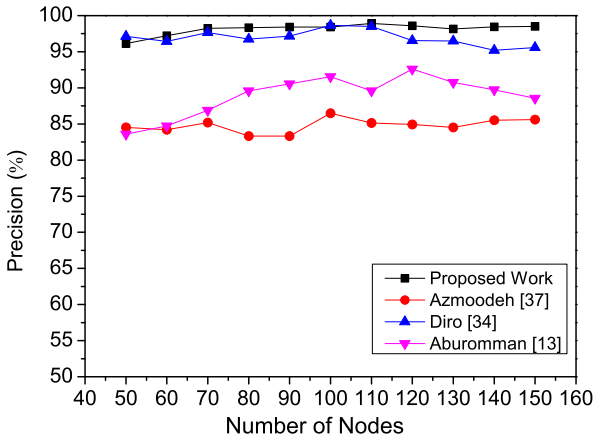


Fig. 4: Precision v/s Number of Nodes.

percentage of the precision proves that developed system better classifies the intruders in comparison to the state of art technique.

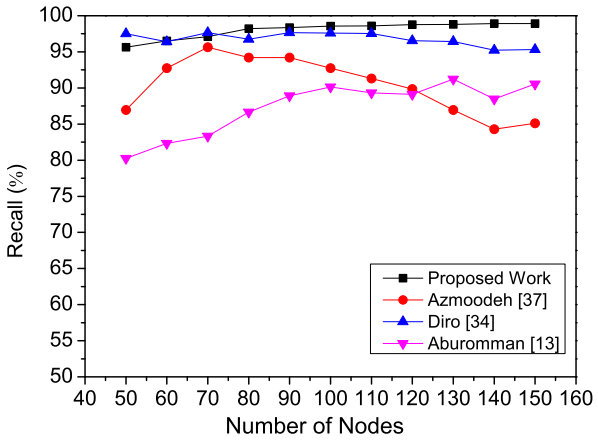


Fig. 5: Recall v/s Number of Nodes.

Fig. 5 shows the computation of Recall with a different number of nodes. It is clearly seen that number of attacks identified by the proposed technique better than other state of art approaches. The average value of the recall for the proposed work is 97.97 and that of existing approach is 90.37, 96.97, and 88.88

in [Azmoodeh et al.(2018)Azmoodeh, Dehghantanha, Conti, and Choo], [Diro and Chilamkurti(2018)], and [Aburomman and Reaz(2016)] respectively. Thus, percentage of recall improvement is 7.7%, 1%, and 9.2% respectively. Thus, intrusion detection efficiency of the developed technique better than other existing works.

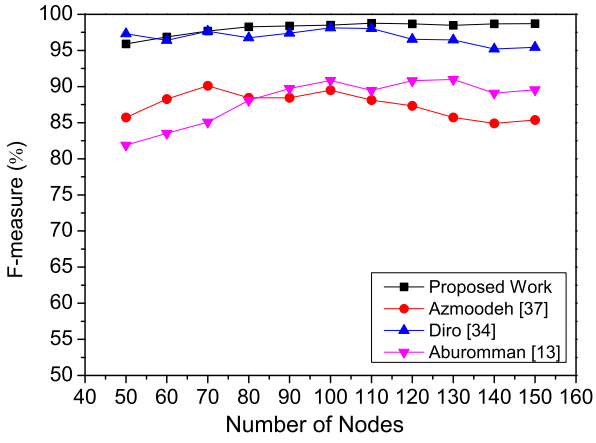


Fig. 6: F-measure v/s Number of Nodes.

F-measure is basically an average of recall and precision for different number of nodes is shown in Fig. 6. The proposed work shows improved value of the F-measure as compared to the previous works. There is a sharp rise as the number of nodes increases after 60. Later on, the value remains constant and then further falls after 70 nodes. The average obtained value for F-measure is 98.02 while that in [Azmoodeh et al.(2018)Azmoodeh, Dehghantanha, Conti, and Choo] is 87.64, in [Diro and Chilamkurti(2018)] is 96.98, and in [Aburomman and Reaz(2016)] is 88.68. Thus, overall F- measure has been improved by 10.5%, 1%, and 9.5% respectively.

Fig. 7 depicts the accuracy of the proposed work and state of art technique. It is clear from the graph that accuracy of the proposed algorithm is better than the past work. The average accuracy obtained through the proposed work is 98.59. However, in [Azmoodeh et al.(2018)Azmoodeh, Dehghantanha, Conti, and Choo], results show accuracy percentage of 90.89. The accuracy in [Diro and Chilamkurti(2018)] is 97.84, and in [Aburomman and Reaz(2016)] results have 94.30 percentage of accuracy, thus, overall accuracy of the proposed work shows an improvement by 7.8%, 0.7%, and 4.3% respectively.

The computed results have been compared with the state of art techniques for validation of results. The accuracy, precision, recall, and F-measure has been compared as shown in Fig. 8.

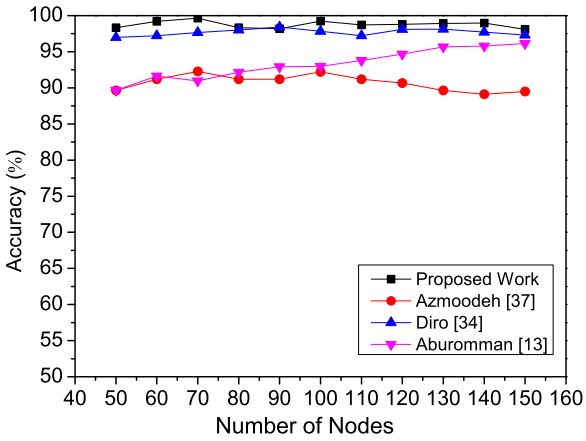


Fig. 7: Accuracy v/s Number of Nodes.

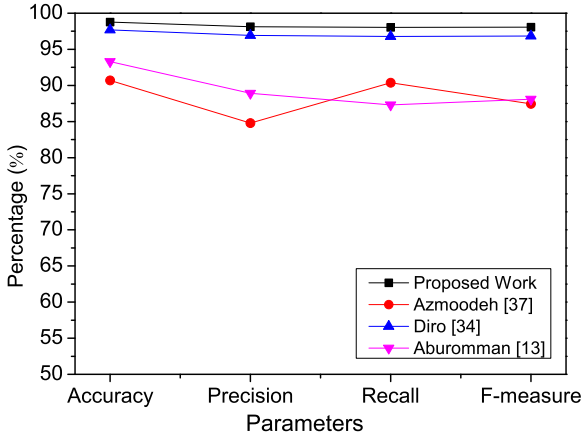


Fig. 8: Comparative performance of state-of-the-art techniques.

The proposed work has been compared with state of art approaches [Azmoodeh et al.(2018)Azmoodeh, Dehghantanha, Conti, and Choo], [Aburomman and Reaz(2016)], [Diro and Chilamkurti(2018)]. Aburomman *et al.* [Aburomman and Reaz(2016)] proposed an interesting architecture of the combination of PSO and SVM. A new fitness model was introduced in the paper but due to high sophistication and iterative delays of PSO, their proposed performance was low. Azmoodeh *et al.* [Azmoodeh et al.(2018)Azmoodeh, Dehghantanha,

Conti, and Choo] used only one parameter, i.e., energy consumption to detect the intrusion in the IoT network. We used throughput and PDR as an additional parameters including energy consumption. So, their performance is lower as compared to our proposed method. Diro *et al.* [Diro and Chilamkurti(2018)] proposed a distributed intrusion detection technique using deep neural network. Their performance was good but their approach have high false detection rate.

The average of the different parameters has been taken and results have been compared which indicates that proposed work shows better results. Overall, effective outcomes have been obtained in comparison to other state of art approaches.

6 Conclusion

The increasing trend of Internet based devices has raised the concerns of security and privacy in the real time environment. Our paper proposed a deep learning based intrusion detection system using SVM and DNN. Since, SVM is a binary classifier, only two routes at a time has been classified to demonstrate the appropriate route. The deployment model and cosine similarity measure have been introduced to understand the route similarity. The proposed work has been evaluated by measuring different parameters such as F-measure, Recall, Precision, and Accuracy. The effectiveness of the proposed architecture has been checked by comparing the results with other state of art techniques. The results indicate that Precision, Recall, F- measure and Accuracy have improved by 13%, 74%, 71% and 76% in comparison to other approaches.

Declarations

Funding : Not applicable
Conflict of interest : The authors declare that they have no conflict of interest.
Consent for publication : All the authors gave their consent for publication.

References

Aburomman and Reaz(2016). Aburomman AA, Reaz MBI (2016) A novel SVM-kNN-PSO ensemble method for intrusion detection system. Applied Soft Computing 38:360–372

Alrawais et al.(2017)Alrawais, Alhothaily, Hu, and Cheng. Alrawais A, Alhothaily A, Hu C, Cheng X (2017) Fog computing for the internet of things: Security and privacy issues. IEEE Internet Computing 21(2):34–42

Ammar et al.(2018)Ammar, Russello, and Crispo. Ammar M, Russello G, Crispo B (2018) Internet of Things: A survey on the security of IoT frameworks. Journal of Information Security and Applications 38:8–27

Azmoodeh et al.(2018)Azmoodeh, Dehghantanha, Conti, and Choo. Azmoodeh A, Dehghantanha A, Conti M, Choo KKR (2018) Detecting crypto-ransomware in IoT networks based on energy consumption footprint. Journal of Ambient Intelligence and Humanized Computing 9(4):1141–1152

Caspi(2017). Caspi G (2017) Introducing Deep Learning: Boosting Cybersecurity With An Artificial Brain

Catarinucci et al.(2015)Catarinucci, de Donno, Mainetti, Palano, Patrono, Stefanizzi, and Tarricone. Catarinucci L, de Donno D, Mainetti L, Palano L, Patrono L, Stefanizzi ML, Tarricone L (2015) An IoT-Aware Architecture for Smart Healthcare Systems. IEEE Internet of Things Journal 2(6):515–526

Chaabouni et al.(2019)Chaabouni, Mosbah, Zemmari, Sauvignac, and Faruki. Chaabouni N, Mosbah M, Zemmari A, Sauvignac C, Faruki P (2019) Network intrusion detection for IoT security based on learning techniques. IEEE Communications Surveys & Tutorials 21(3):2671–2701

Choudhary and Kesswani(2019). Choudhary S, Kesswani N (2019) A Survey: Intrusion Detection Techniques for Internet of Things. International Journal of Information Security and Privacy (IJISP) 13(1):86–105

Damshenas et al.(2013)Damshenas, Dehghantanha, and Mahmoud. Damshenas M, Dehghantanha A, Mahmoud R (2013) A survey on malware propagation, analysis, and detection. International Journal of Cyber-Security and Digital Forensics 2(4):10–30

Diro and Chilamkurti(2018). Diro AA, Chilamkurti N (2018) Distributed attack detection scheme using deep learning approach for Internet of Things. Future Generation Computer Systems 82:761–768

Faruki et al.(2014)Faruki, Bharmal, Laxmi, Ganmoor, Gaur, Conti, and Rajarajan. Faruki P, Bharmal A, Laxmi V, Ganmoor V, Gaur MS, Conti M, Rajarajan M (2014) Android security: a survey of issues, malware penetration, and defenses. IEEE communications surveys & tutorials 17(2):998–1022

Gubbi et al.(2013)Gubbi, Buyya, Marusic, and Palaniswami. Gubbi J, Buyya R, Marusic S, Palaniswami M (2013) Internet of Things (IoT): A vision, architectural elements, and future directions. Future generation computer systems 29(7):1645–1660

Hassija et al.(2019a)Hassija, Chamola, Saxena, Jain, Goyal, and Sikdar. Hassija V, Chamola V, Saxena V, Jain D, Goyal P, Sikdar B (2019a) A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures. IEEE Access 7:82721–82743

Hassija et al.(2019b)Hassija, Chamola, Saxena, Jain, Goyal, and Sikdar. Hassija V, Chamola V, Saxena V, Jain D, Goyal P, Sikdar B (2019b) A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures. IEEE Access 7:82721–82743

Hui and Culler(2008). Hui JW, Culler DE (2008) Extending IP to low-power, wireless personal area networks. IEEE Internet Computing 12(4):37–45

Javaid et al.(2016)Javaid, Niyaz, Sun, and Alam. Javaid A, Niyaz Q, Sun W, Alam M (2016) A deep learning approach for network intrusion detection system. In: Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS), ICST (Institute for Computer Sciences, Social-Informatics and . . . , pp 21–26

Jun and Chi(2014). Jun C, Chi C (2014) Design of complex event-processing IDS in internet of things. In: 2014 Sixth International Conference on Measuring Technology and Mechatronics Automation, IEEE, pp 226–229

Kasinathan et al.(2013)Kasinathan, Costamagna, Khaleel, Pastrone, and Spirito. Kasinathan P, Costamagna G, Khaleel H, Pastrone C, Spirito MA (2013) An IDS framework for internet of things empowered by 6LoWPAN. In: Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security, ACM, pp 1337–1340

Kolias et al.(2015)Kolias, Kambourakis, Stavrou, and Gritzalis. Kolias C, Kambourakis G, Stavrou A, Gritzalis S (2015) Intrusion detection in 802.11 networks: empirical evaluation of threats and a public dataset. IEEE Communications Surveys & Tutorials 18(1):184–208

Kushalnagar et al.(2007)Kushalnagar, Montenegro, and Schumacher. Kushalnagar N, Montenegro G, Schumacher C (2007) IPv6 over low-power wireless personal area networks (6LoWPANs): overview, assumptions, problem statement, and goals. Tech. rep.

Le et al.(2012)Le, Loo, Lasebae, Aiash, and Luo. Le A, Loo J, Lasebae A, Aiash M, Luo Y (2012) 6LoWPAN: a study on QoS security threats and countermeasures using intrusion detection system approach. International Journal of Communication Systems 25(9):1189–1212

Li et al.(2018)Li, Zhao, Li, and Zhang. Li J, Zhao Z, Li R, Zhang H (2018) AI-based Two-Stage Intrusion Detection for Software Defined IoT Networks. IEEE Internet of Things Journal 6(2):2093–2102

Li et al.(2016)Li, Chen, Liu, and Min. Li Z, Chen R, Liu L, Min G (2016) Dynamic Resource Discovery Based on Preference and Movement Pattern Similarity for Large-Scale Social Internet of Things. IEEE Internet of Things Journal 3(4):581–589

Liang et al.(2019)Liang, Hatcher, Liao, Gao, and Yu. Liang F, Hatcher WG, Liao W, Gao W, Yu W (2019) Machine Learning for Security and the Internet of Things: The Good, the Bad, and the Ugly. IEEE Access 7:158126–158147

Lin et al.(2015)Lin, Ke, and Tsai. Lin WC, Ke SW, Tsai CF (2015) CANN: An intrusion detection system based on combining cluster centers and nearest neighbors. Knowledge-based systems 78:13–21

Merlo et al.(2015)Merlo, Migliardi, and Caviglione. Merlo A, Migliardi M, Caviglione L (2015) A survey on energy-aware security mechanisms. Pervasive and Mobile Computing 24:77–90

Montenegro et al.(2007)Montenegro, Kushalnagar, Hui, and Culler. Montenegro G, Kushalnagar N, Hui J, Culler D (2007) Transmission of IPv6 packets over IEEE 802.15. 4 networks. Tech. rep.

Moustafa et al.(2018)Moustafa, Turnbull, and Choo. Moustafa N, Turnbull B, Choo KKR (2018) An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things. IEEE Internet of Things Journal

Pajouh et al.(2016)Pajouh, Javidan, Khayami, Ali, and Choo. Pajouh HH, Javidan R, Khayami R, Ali D, Choo KKR (2016) A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in IoT backbone networks. IEEE Transactions on Emerging Topics in Computing

Pongle and Chavan(2015). Pongle P, Chavan G (2015) Real time intrusion and wormhole attack detection in internet of things. International Journal of Computer Applications 121(9)

Quincozes et al.(2019)Quincozes, Emilio, and Kazienko. Quincozes S, Emilio T, Kazienko J (2019) MQTT Protocol: Fundamentals, Tools and Future Directions. IEEE Latin America Transactions 17(09):1439–1448

Raza et al.(2013)Raza, Wallgren, and Voigt. Raza S, Wallgren L, Voigt T (2013) SVELTE: Real-time intrusion detection in the Internet of Things. Ad hoc networks 11(8):2661–2674

Sfar et al.(2018)Sfar, Natalizio, Challal, and Chtourou. Sfar AR, Natalizio E, Challal Y, Chtourou Z (2018) A roadmap for security challenges in the Internet of Things. Digital Communications and Networks 4(2):118–137

Shaerpour et al.(2013)Shaerpour, Dehghantanha, and Mahmod. Shaerpour K, Dehghantanha A, Mahmod R (2013) Trends in android malware detection. Journal of Digital Forensics, Security and Law 8(3):2

Shelby et al.(2014)Shelby, Hartke, Bormann, and Frank. Shelby Z, Hartke K, Bormann C, Frank B (2014) RFC 7252: The constrained application protocol (CoAP). Internet Engineering Task Force

Stephen and Arockiam(2017). Stephen R, Arockiam L (2017) Intrusion detection system to detect sinkhole attack on RPL protocol in Internet of Things. International Journal of Electrical Electronics and Computer Science 4(4):16–20

Thamilarasu and Chawla(2019). Thamilarasu G, Chawla S (2019) Towards deep-learning-driven intrusion detection for the Internet of Things. Sensors 19(9):1977

Winter et al.(2012)Winter, Thubert, Brandt, Hui, Kelsey, Levis, Pister, Struik, Vasseur, and Alexander. Winter T, Thubert P, Brandt A, Hui J, Kelsey R, Levis P, Pister K, Struik R, Vasseur JP, Alexander R (2012) RPL: IPv6 routing protocol for low-power and lossy networks. Tech. rep.

Yang and Tang(2016). Yang H, Tang R (2016) Power consumption based Android malware detection. Journal of Electrical and Computer Engineering 2016

Zolanvari et al.(2019)Zolanvari, Teixeira, Gupta, Khan, and Jain. Zolanvari M, Teixeira MA, Gupta L, Khan KM, Jain R (2019) Machine Learning-Based Network Vulnerability Analysis of Industrial Internet of Things. IEEE Internet of Things Journal 6(4):6822–6834

Figures

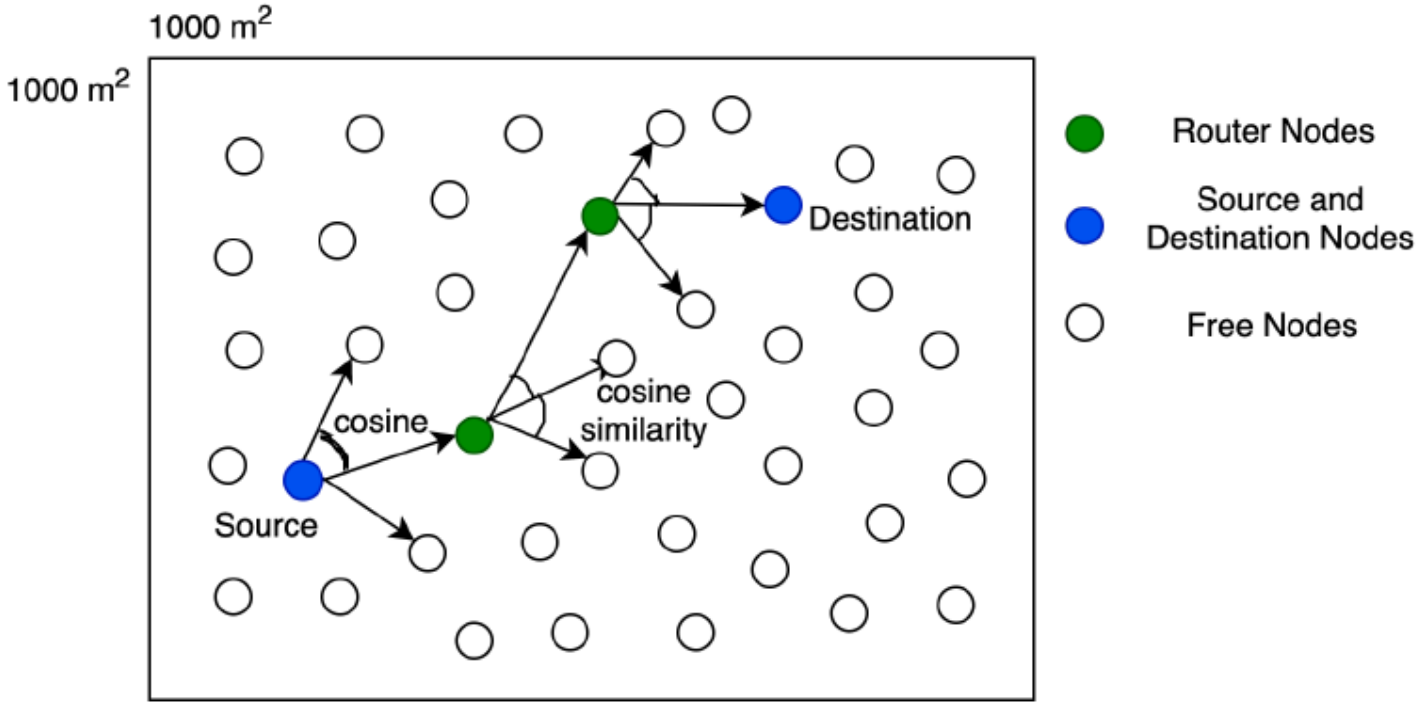
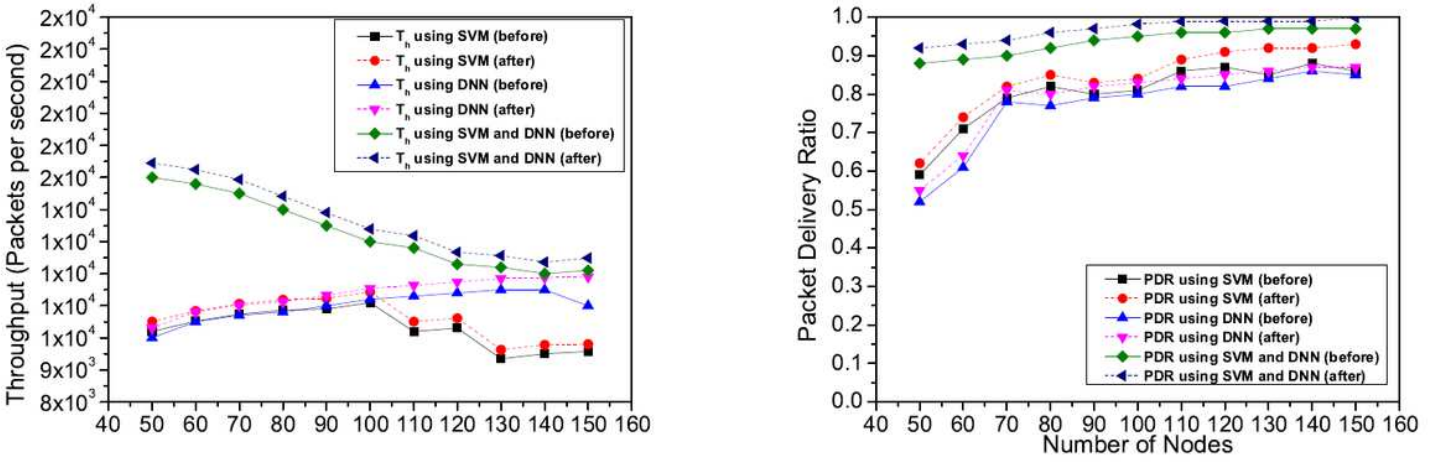


Figure 1

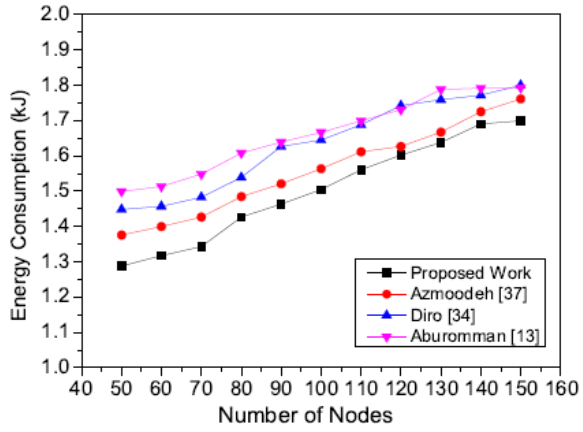
Route discovery using similarity calculation.



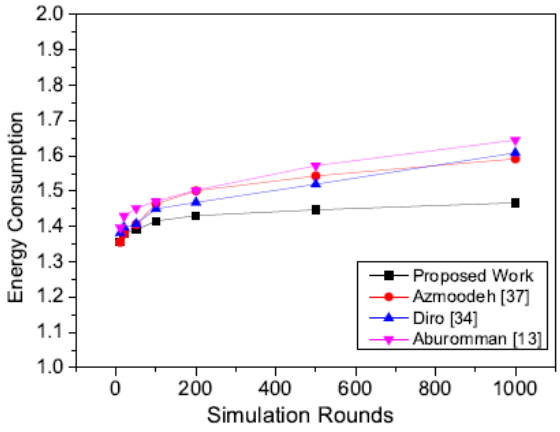
(a) Throughput computation. (b) Packet Delivery Ratio computation.

Figure 2

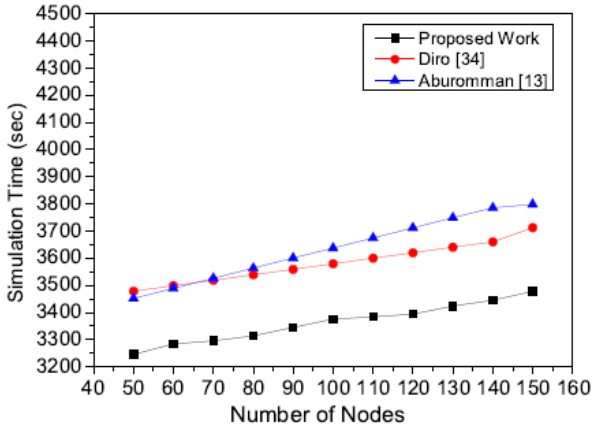
Throughput and PDR computation before and after preventing from attacks.



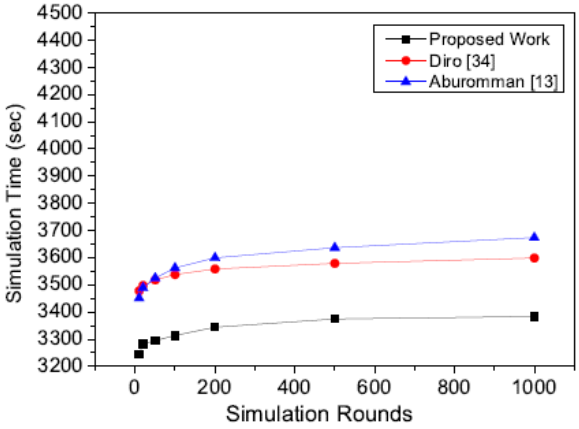
(a) Energy consumption over number of nodes.



(b) Energy consumption over simulation rounds.



(c) Time consumption over number of nodes.



(d) Time consumption over simulation rounds.

Figure 3

Energy and Time Consumption over number of nodes and number of simulation rounds.

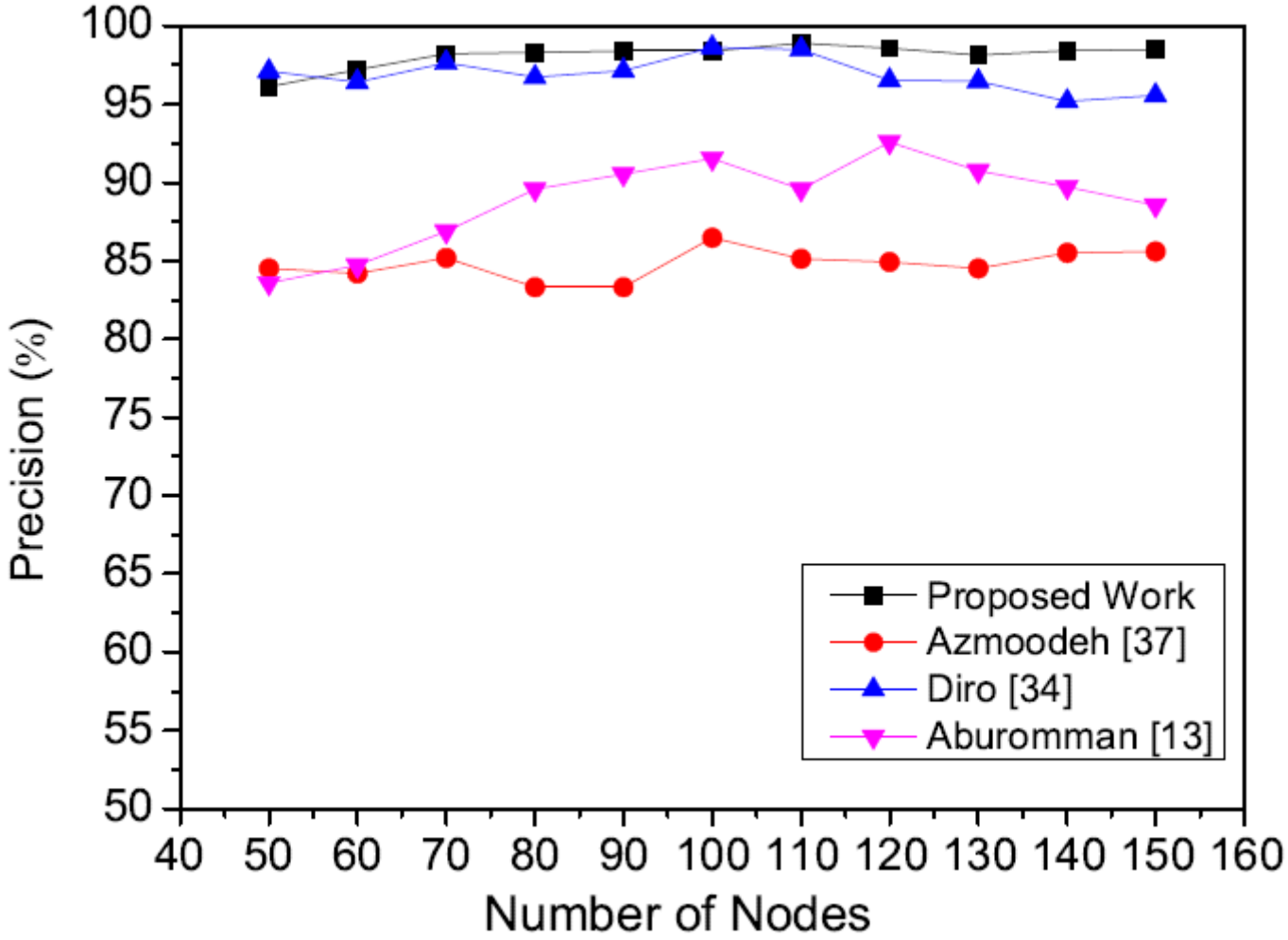


Figure 4

Precision v/s Number of Nodes.

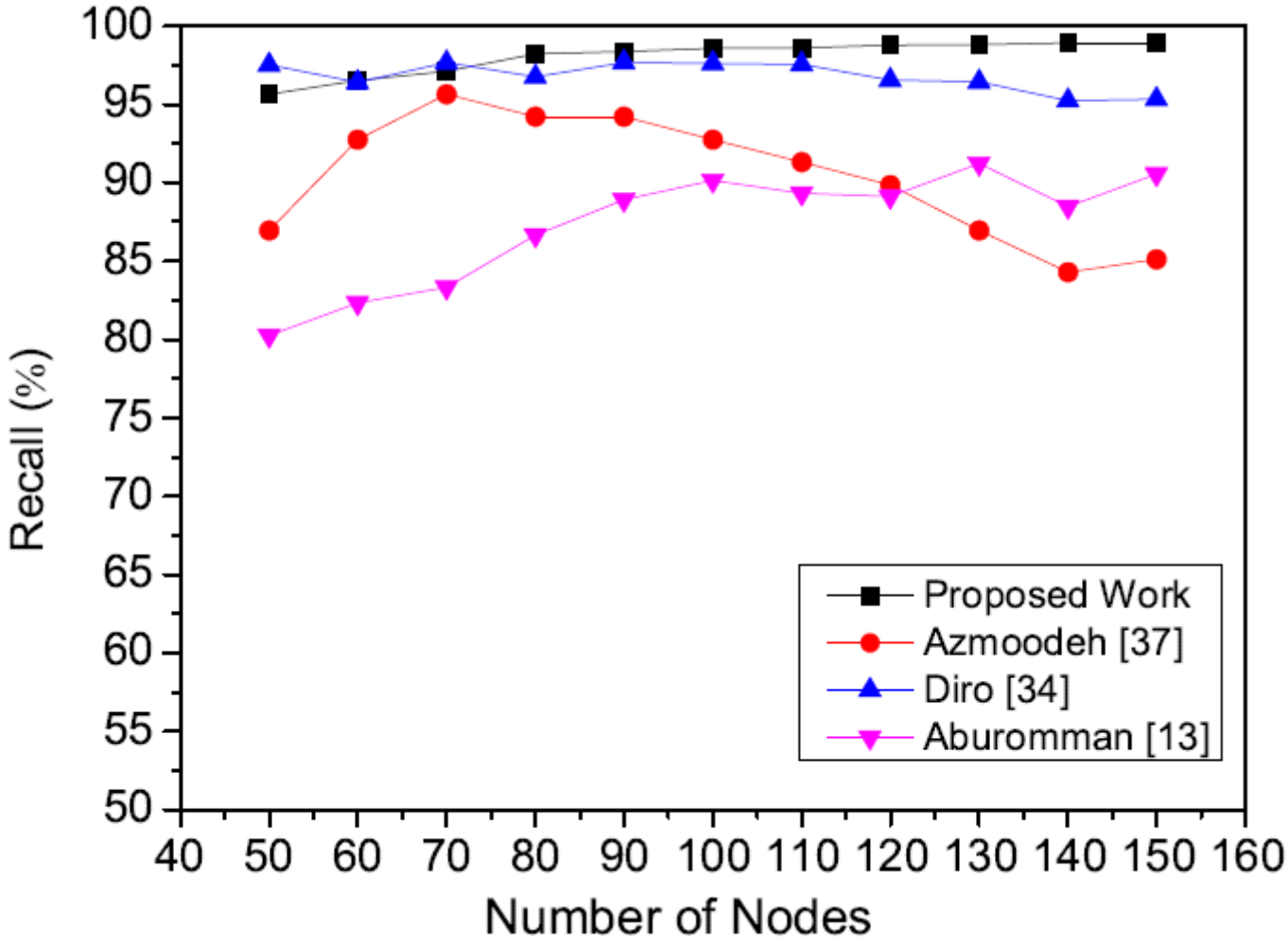


Figure 5

Recall v/s Number of Nodes.

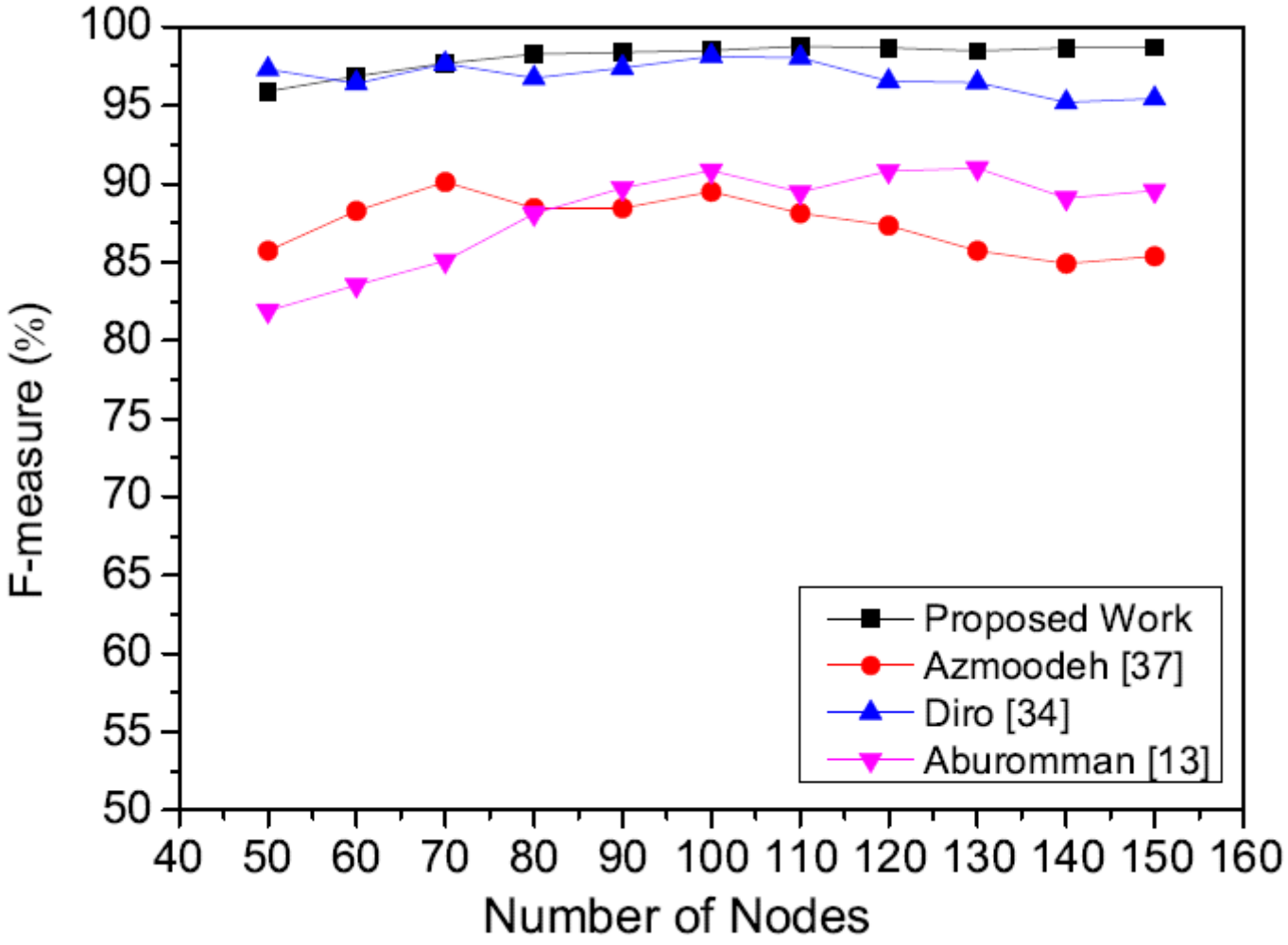


Figure 6

F-measure v/s Number of Nodes.

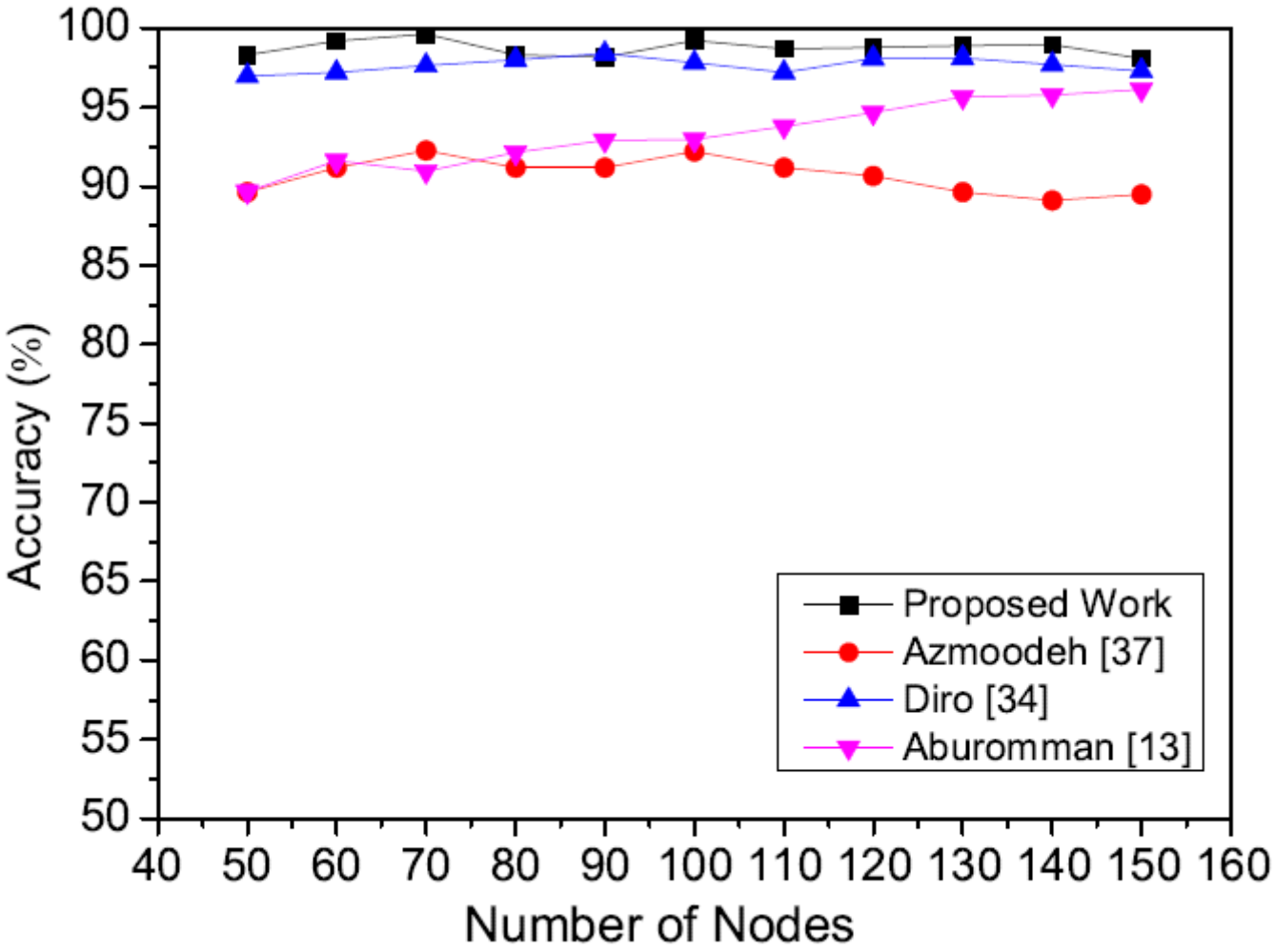


Figure 7

Accuracy v/s Number of Nodes.

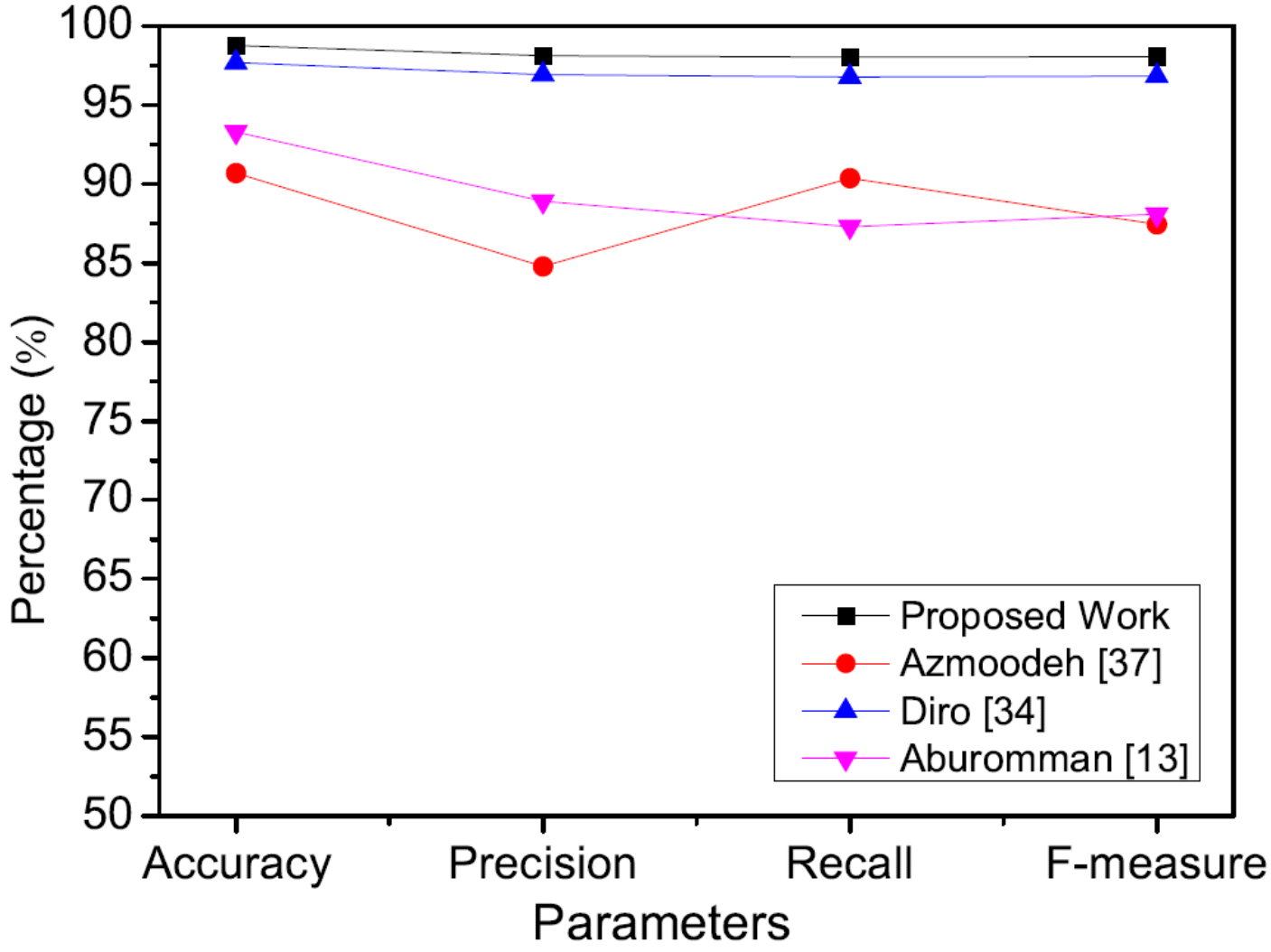


Figure 8

Comparative performance of state-of-the-art techniques.