A Hybrid Intrusion Detection Architecture for Internet of Things

Mansour Sheikhan
Department of Communication Engineering
Islamic Azad University, South Tehran Branch
Tehran, Iran
msheikhn@azad.ac.ir

Hamid Bostani
Department of Computer Engineering
Islamic Azad University, South Tehran Branch
Tehran, Iran
st_h_bostani@azad.ac.ir

***Abstract*—In computer networks, Internet of things (IoT) is an emerging paradigm wherein smart and resource-constrained objects can connect to Internet by using a wide range of technologies. Due to the insecure nature of Internet and also wireless sensor networks (WSNs), which are the main components of IoT, implementing security mechanisms in IoT seems necessary. To deal with intrusions which may occur in IoT, a novel intrusion detection architecture model for IoT is proposed in this paper. This model is based on MapReduce approach with the aim of distributed detection. To provide multi-faceted detection (from the Internet and WSNs sides), the proposed model consists of anomaly-based and misuse-based intrusion detection agents that use supervised and unsupervised optimum-path forest model for intrusion detection. The experimental results of simulated scenarios show the superior performance of proposed method in intrusion detection for IoT.**

Keywords—Internet of things; intrusion detection; optimum path forest; anomaly detection

I. INTRODUCTION

The Internet of things (IoT) is a worldwide network in which all heterogeneous objects around us (such as smart phones, laptops or smart sensors) can connect to the Internet by using a wide range of technologies. In other words, large number of smart interconnected devices in IoT results in valuable services to the society and individual citizens [1]. Moreover, IoT can be supported by satellite communication systems for the case of Internet of remote things (IoRT) in which Internet protocol version 6 (IPv6) should be supported over satellite [2].

The general architecture of IoT is shown in Fig. 1. As seen in Fig. 1, the IPv6 over low-power wireless personal area networks (6LoWPANs) [3] is a wireless sensor network (WSN) which allows the connection of resource-constrained devices, such as sensor nodes, to the Internet through the 6LoWPAN border router (6BR) [4, 5]. It is noted that the routing protocol for low power and lossy networks (RPL) [5] is a certain routing protocol for 6LoWPAN. The RPL, which is based on the construction of a destination-oriented directed acyclic graph (DODAG), is an IP-based distance vector and hop-by-hop routing protocol. RPL enables different operations such as the unidirectional traffic towards a DODAG root, bidirectional traffic between resource-constrained devices (i.e., 6LoWPAN nodes), and bidirectional traffic between resource-constrained devices and the DODAG root [5].

Along with the rapid growth of technology in computer networks such as IoT, security has become a critical challenge. The main security requirements for the IoT are as follows [6]: a) data confidentiality and authentication and b) privacy and trust among users and things. The communication in the IoT can be secured by using standard mechanisms such as cryptography and authentication techniques; however, these preventive mechanisms cannot detect all possible attacks, because of the nature of wireless communication. On the other hand, the resource-constrained devices are directly connected to unreliable Internet via IPv6 and 6LoWPAN networks in the IoT; so, they are vulnerable to intrusions (both from the Internet and WSNs) [4]. Therefore, an intrusion detection system (IDS) is required for detecting malicious activities in the IoT besides the standard security mechanisms.

IDS is an effective mechanism which gathers system activities or network traffic as input data with the aim of analyzing them for identifying malicious behaviors. IDSs are classified into the following categories: a) misuse-based (as the best method for detecting known attacks); b) anomaly-based (as the best method for detecting unknown attacks); and c) specification-based detection systems.

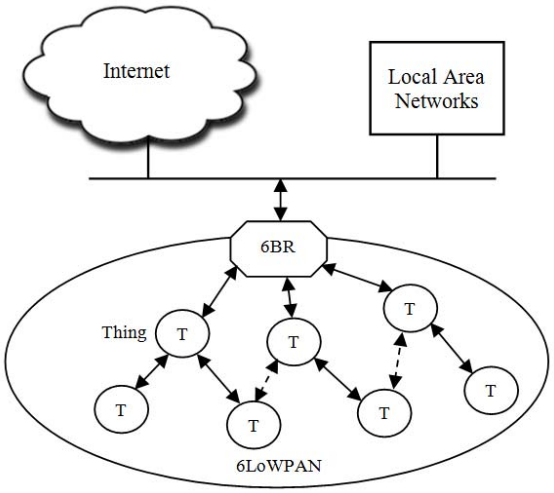

Fig. 1. General architecture of IoT.

978-1-5090-3455-2/16/$31.00 ©2016 IEEE

In the misuse-based detection systems, predefined attack patterns are modeled and maintained in the database of the attacks' signature. However, anomaly-based systems, which usually use statistical or machine learning methods, are based on finding the deviations from normal behaviors of network traffic or system activities. The specification-based systems work by the same way, as well. However, user guidance is required to extract legitimate system activities or network traffic for developing a model of normal behavior.

Designing IDS for IoT is still a new and on-going research subject and to the best knowledge of the authors, a few researchers in the security field work on this context. For example, Raza et al. [4] proposed a novel real-time IDS for IoT called SVELTE. They showed through simulated scenarios that SVELTE has a small overhead to deploy on the constrained nodes and can detect most of malicious nodes that launch sinkhole and/or selective-forwarding attacks. Kasinathan et al. [7] introduced a denial of service (DoS) attack detection architecture for 6LoWPAN. Their simulation results showed the capability of the proposed architecture in detecting DoS attacks. One of the main goals followed by employing an IDS in the IoT is fast security event-processing that results in detecting network attacks, immediately. For this purpose, June and Chi [8] designed a complex event-processing (CEP)-based IDS in the IoT environments to achieve better performance in real-time data computations.

IoT is a hybrid network which is composed of the Internet and the networks with heterogeneous nodes (e.g., 6LoWPAN). The traffic patterns of these networks are completely different. With the aim of providing multi-faceted detection (considering both traffic patterns from the Internet and WSNs sides), a hybrid distributed IDS is proposed for intrusion detection in this study for IoT. This model is based on the MapReduce approach that uses supervised and unsupervised OPF. For this purpose, we proposed a novel real-time intrusion detection framework based on anomaly-based detection for detecting insider (internal) attacks which may be happening in 6LoWAPN. The proposed method is focused only on detecting the malicious behaviors of sinkhole and selective-forwarding attacks (as the well-known routing attacks) in 6LoWPAN; however, it can be developed for detecting other attacks. On the other hand, a misuse-based intrusion detection engine is provided that is responsible for detecting cyber (external) attacks that may be occurred from Internet (or LANs) side.

The rest of this paper includes the following sections: the foundations of preliminaries are introduced briefly in Section 2. In Section 3, the processes of the proposed model are introduced in detail and the performance of the proposed model on simulated scenarios is reported in Section 4. Finally, the paper is concluded in Section 5.

II. PRELIMINARIES

A. Supervised and Unsupervised OPF

Supervised optimum-path forest (OPF) [9] algorithm is a graph-based machine learning method which reduces a pattern recognition problem into an optimal graph partitioning in a given feature space. In the OPF, each training sample is shown as a node in a complete weighted graph. The weighted arcs, which are defined by adjacency relations between samples, link all pairs of nodes in this graph. Suppose $G = (Z_1, A)$ as a complete weighted graph where Z_1 denotes the training samples. The samples in the training dataset are represented by the nodes of G, and each pair of samples is defined by its arc as $A = Z_1 \times Z_1$. In the training phase, some key samples from the training set, called prototypes, should be identified for each class in the classification problem. The closest nodes in the minimum spanning tree of G which have different labels in Z_1 are the prototypes of OPF, wherein some prototypes that minimizes the classification error make the optimum set of prototypes ($S^* \subset Z_1$) [9]. Then, the complete weighted graph will be partitioned into optimum-path trees (OPTs) by a competitive process between prototypes (as the roots of the OPTs) which introduces optimum paths to the remaining nodes of the graph [10]. The nodes of OPT will be strongly connected to their prototypes as compared to other prototypes in the OPF and consequently they have the same label as the OPT's root. Partitioning the OPF for computing OPTs is performed by minimization of $f_{\max}$ which assigns an optimum path $P^*(t)$ from the set of prototypes to every sample $t \in Z_1$ whose minimum cost $C(t)$ is calculated as follows [9]:

$$C(t) = \min_{\forall \pi_t \in (Z_1, A)} \{f_{\max}(\pi_t)\} \quad (1)$$

where $f_{\max}$ is a path-value function that assigns a path cost to each path π_t [9]. To classify each unlabeled sample such as t, we assume t as a part of the training set. The purpose of the classification phase is to find an optimum path $P^*(t)$ from S^* to t, and then labeling t with the class of its root. The optimum path can be found incrementally by evaluating the optimum cost as follows [9]:

$$C(t) = \min\{\max\{C(s), d(s,t)\}\}; \forall s \in Z_1 \quad (2)$$

where $C(s)$ is the minimum cost of s and $d(s,t)$ is the Euclidean distance from t to s. To see more details about the training and classification phases of OPF algorithm, refer to [9].

Unsupervised OPF, which is called optimum-path forest clustering (OPFC) [10], is almost same as supervised OPF. However, in the OPFC, each sample in the dataset (represented by a feature vector) is shown as a node in the k-nearest neighbors graph (G_{k-nn}) that is connected with its k best neighbors in a given feature space [11]. In OPFC, the arcs are weighted by the distance between each pair of nodes and the nodes are weighted by the probability density function (pdf) of each node that is based on the distance between the samples and their k-nearest neighbors [11]. When G_{k-nn} is created, the OPFC algorithm will find one sample (node) at each maximum pdf as a root of a dome or cluster which includes dense samples in the feature space. Then, an OPT will be created from each root to every node in the influence

zone (cluster) such that each OPT node will be strongly connected to its root as compared to other obtained roots in the G_{k-nn} [10]. Notably, as mentioned earlier, each $s \in Z$ (where Z is the training set) is weighted by a pdf that is defined as follows [10]:

$$p(\mathrm{s}) = \frac{1}{\sqrt{2\pi\sigma^2 \left|A_k(s)\right|}} \sum_{\forall t \in A_k(s)} \exp\left(\frac{-d^2(s,t)}{2\sigma^2}\right) \qquad (3)$$

where $\left|A_k(s)\right| = k$, $\sigma = d_f / 3$, and d_f is the maximum arc weight in G_{k-nn} . It is noted that $A_k(s)$ is the neighbor set of $s \in Z$. An OPFC model classifies a new sample to a special cluster (that was created in the OPFC algorithm), by finding a root which provides the optimum path to the new sample. To see more details about the OPFC algorithm refer to [10, 11]. Generally, supervised and unsupervised OPF are simple and fast classifiers which are parameter-independent and originally support multi-class problems [13]. Moreover, OPF does not make any assumption about the shape of classes; so, partial overlapping among the classes can be handled by the OPF [13]. In this study, the OPFC algorithm is used as an anomaly detection engine for detecting the insider attacks in IoT which may be happened by malicious things from WSN sides. Moreover, a new version of supervised OPF, called MOPF [14], is used in a misuse detection engine for identifying the cyber (external) attacks from Internet side. More details about the proposed method will be discussed in Section 3.

B. MapReduce Approach

MapReduce approach [15] which was firstly presented by Google, is an efficient solution for the big data problem. This approach employs algorithms that have parallelism capabilities in a parallel space. In this approach, a big dataset is split to smaller datasets and stored on different machines. These machines process the smaller datasets in parallel and finally, the results will be integrated. In the Map phase, input data is partitioned to smaller segments named chunk. Then, they are delivered to some machines, called mappers, that are responsible for the mapping operation [16]. Each mapper converts the content of the chunk to a sequence of key-value pairs by using the user-defined "map" function. On the other hand, in the Reduce phase, MapReduce framework performs sorting based on the keys and collects each key-value pair with the same key and sends them to the reducer node. Then, the user-defined "reduce" function accepts the mediate keys with a set of values representing the dimension of keys and merges the values by converting them to a smaller value [15].

III. PROPOSED MODEL

In this section, the proposed model is introduced. As mentioned earlier, with the aim of providing multi-faceted detection, we proposed a flexible model which can detect simultaneously both malicious behavior of 6LoWPAN and Internet (or LANs) sides. The general schema of the proposed model is shown in Fig. 2. As seen in Fig. 2, the proposed model consists of two modules where the Internet side module, which is called misuse detection module, provides misuse-based IDS for classifying the Internet (or LAN) traffic and consequently detecting cyber attacks. Moreover, a 6LoWPAN-side module, which is called anomaly detection module, can identify the insider attacks (i.e., sinkhole and selective-forwarding) by using the anomaly-based IDS that was projected based on 6LoWPAN traffic.

In the 6LoWPAN side, the anomaly detection module creates a sample for each source node by extracting four traffic-related features from the raw received packet of the source node at each time-slot Δw : a) packet receiving rate; b) packet dropping rate; c) average latency; and d) maximum hop-count. Then, the module projects a clustering model based on an unsupervised OPF algorithm for each source node by using its generated samples. The algorithm selects a cluster (or clusters) including a few samples and then labels the samples as anomalous for each projected model.

Notably, we design and implement a WSN based on the RPL routing protocol in this study for simulating the 6LoWPAN functionality (see Section 4). Generally, the structure of data packets in the simulations consists of two main parts (as shown in Fig. 3): a) data (fields) and b) data access interface (functions). In Fig. 3, *SrcID* and *SrcTimeStamp* represent the ID of source node and the time of packet sending by the source node, respectively. *RouterID* and *RouterTimeStamp* represent the ID of the last router node (before the current node) and the forwarding packet time, respectively. *HopCount* shows the number of hops taken by the packets and each router node increments it by *incHopCount( )* function. We assume that the router node cannot access the Data with the aim of manipulating values.

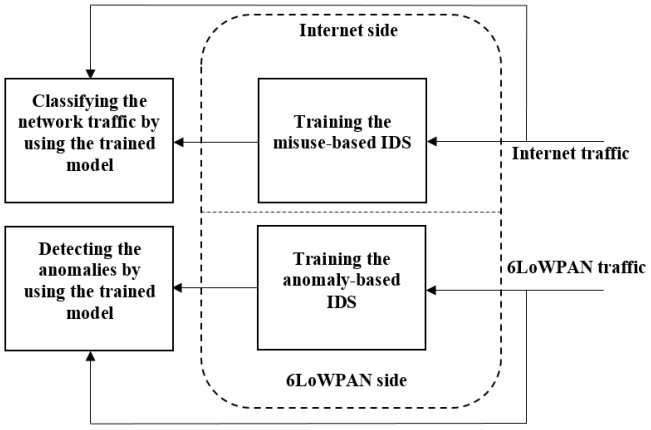

Fig. 2. General schema of proposed model.

Fields	Functions
- SequenceNumber	+ getSequenceNumber( )
- SrcID	+ getSrcID( )
- SrcTimeStamp	+ getSrcTimeStamp( )
- HopCount	+ incHopCount( )
- RouterID	+ getHopCount( )
- RouterTimeStamp	+ getRouteID( )
- ReceivingTimeStamp	+ getRouterTimeStamp( )
- Data	+ getReceivingTimeStamp( )

Fig. 3. Structure of data packets in the simulated WSN.

In extracting features for producing a new sample for each source node, such as *A*, the packet dropping rate is computed based on the following steps at each time-slot.

Step 1- Sort the received packets from node *A* based on its *SequenceNumber*.

Step 2- Calculate the sum of the distances between each two consecutive packets (based on *SequenceNumber*) and return the result as the packet dropping rate. For simplicity, we assume that each packet is sent only once.

Moreover, other features such as the maximum hop-count and the average latency are computed as follows:

$$\forall i \in L : MaxHopCount = Max\left(packet_j^i.getHopCount(\ ) \right) | \ j \in P^i \quad (4)$$

$$\forall i \in L : AverageLatency = \frac{\sum_{j \in P^i} packet_j^i.getRcvingTimeStamp(\ ) - packet_j^i.getSrcTimeStamp(\ )}{\left\| P^i \right\|} \quad (5)$$

where L is the set of source nodes in the network. P^i is the set of received packets from the i-th source in time-slot Δw and $\left\| P^i \right\|$ is the number of its members. By increasing the number of source nodes, the sequential projection of clustering models will be time-consuming that is not acceptable for a real-time model. The proposed anomaly detection method has the capability of parallelism, because projecting and using clustering models are independent processes. In this study, we inspired from MapReduce approach for improving the speed of projecting models and anomaly detection. In fact, we proposed an anomaly detection method based on the MapReduce architecture. In other words, if an appropriate platform (hardware/software) is prepared, then the model can run in parallel on a distributed space based on the MapReduce architecture. In this approach, the root node sends the values of extracted traffic features of the source nodes to corresponding reducer nodes for anomaly detection. According to this approach, we can also add a new reducer node to the proposed architecture that is a host of the proposed misuse detection module. Therefore, by sending the values of Internet traffic features from root node to the new reducer node, our framework can be employed for detecting the cyber attacks from the Internet side. The general architecture of the proposed intrusion detection model is shown in Fig. 4.

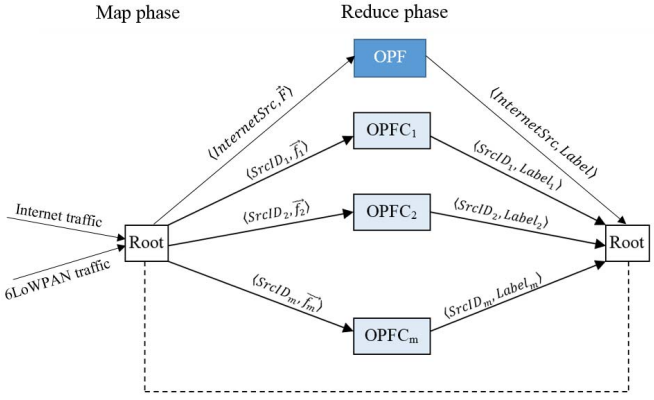

Fig. 4. General architecture of proposed model based on the MapReduce approach.

As seen in Fig. 4, the root node in anomaly detection module (i.e., the 6BR) extracts mentioned traffic-related features from the receiving raw packets (i.e., 6LoWPAN traffic) in each time-slot and creates a new sample for source nodes. Then, it sends the sample's information with key-value pair format to a node (i.e., the reducer) that is responsible to work with a special key. This format includes source ID as the key and feature vector as the value. Then, the reducer node projects a clustering model by using its samples which are received from the mapper node. As mentioned earlier, we assume that a cluster with fewer samples is anomaly; hence, if the new sample belongs to this cluster, it is classified as anomalous and otherwise, it is classified as a normal sample. So, the reducer node returns a new key-value pair with *⟨SID,Label⟩* format (in response to the incoming key-value pair) to the root node. It is noted that the key and the value are source ID's sample and its label (i.e., anomalous/normal), respectively. Notably, this scenario is also repeated for the misuse detection module.

IV. Experimental Results and Analysis

In this section, the simulation results of the proposed model are presented to show the robustness of the proposed method in IoT's intrusion detection field. Notably, for evaluating the anomaly detection module which is used in 6LoWAN side, we developed a special WSN that is based on the RPL protocol using .Net Framework technology and C#.Net programming. So, a flexible evaluation platform was provided for developing the proposed intrusion detection method and simulating the selective-forwarding and sinkhole attacks, as well. It is noted that in selective-forwarding attacks, which primarily disrupt the routing path, malicious nodes forward packets selectively to remove some packets based on the importance of data or randomly [5]. However, a malicious node represents itself to others as an optimal routing path in sinkhole attacks for attracting nearby nodes to route traffic through it [4]. The assumptions in the developed simulator are given in Table 1.

Moreover, the NSL-KDD [17] dataset was used instead of Internet (or LANs) traffic for evaluating the misuse detection modules in the proposed architecture (Fig. 4). In this study, 7,000 and 3,000 instances were randomly selected from the NSL-KDD as the training and test datasets, respectively. Table 2 lists the number of training and test instances.

TABLE I. Assumptions in the Developed Simulator in This Study

Parameter	Value/Type
Network scale	100 m × 100 m
Routing protocol	RPL
Transmission range	10 m
Packet size	127 bytes
DIO size	24 bytes
Δw	30 sec

TABLE II. SIZE OF TRAINING AND TEST DATASETS

Type of dataset	Total number of instances	Number of normal instances	Number of anomaly instances
Training	7,000	3,490	3,510
Test	3,000	1,526	1,474

Notably, the NSL-KDD features have significantly different ranges and various resolutions; therefore, most of the classifiers are not able to process data in this format. Therefore, it is essential to normalize the value of each feature to avoid data imbalance. In this study, we normalized the NSL-KDD features same as detailed in [18].

In the 6BR implementation, which was based on the MapReduce architecture, the MATLAB server was used as the reducer node for projecting the clustering and classification models with the aim of anomaly and misuse detection, respectively. We implemented anomaly and misuse detection methods that were based on the OPFC and MOPF algorithms, respectively, using MATLAB R2014a on a PC with an Intel(R) Core (TM) i5-4460, CPU 3.20 GHz, and 8 GB RAM.

In this study, the performance of the proposed method was evaluated in terms of detection rate (DR) and false alarm rate (FAR). As mentioned earlier, for evaluating the performance of the proposed model in terms of detecting the insider attacks, a simulated WSN's traffic was used in 6BR.

The screenshot of simulated WSN is shown in Fig. 5. In Fig. 5, sinkhole and selective-forwarding attacks were launched simultaneously by node 3 and node 10, respectively. Through this simulation, the performance of the proposed model was evaluated to deal with the joint occurrence of sinkhole and selective-forwarding attacks in the 6LoWPAN. The assumptions in these simulations are given in Table 3.

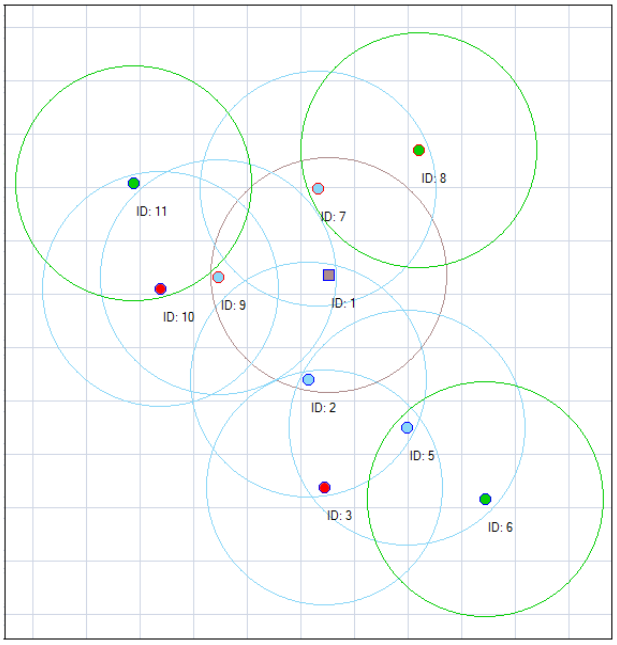

Fig. 5. Screenshot of 6LoWPAN simulation.

TABLE III. ASSUMPTIONS IN THE 6LOWPAN SIMULATION

Feature	Value
Number of source nodes	3 (IDs: {6, 8, 11})
Number of router nodes	6 (IDs: {2, 3, 5, 7, 9, 10})
Root's ID	{1}
Malicious nodes' ID	{3} (as the sinkhole agent) and {10} (as the selective-forwarding agent)
Simulation time (min)	30
Number of source nodes	3 (IDs: {6, 8, 11})

Moreover, the performance of the proposed model, in terms of detecting cyber attacks, was evaluated simultaneously by using NSL-KDD dataset. These evaluations were performed based on the proposed architecture (Fig. 4) that was explained in Section 3.

The performance of the proposed model in the mentioned experiment is reported in Table 4. As seen in Table 4, the performance of the anomaly detection module (as compared to other proposed methods such as SVELTE [4]) and misuse detection module is appropriate in terms of DR and FAR. The experimental results show that the acceptable performance of the proposed model in detecting both insider and cyber attacks, simultaneously.

Notably, for evaluating the performance of MOPF in detecting cyber attacks in the proposed model, the proposed misuse detection module (based on MOPF) was compared with other misuse detection systems that used the following classifiers (instead of MOPF): a) support vector machine (SVM); b) classification and regression tree (CART); and c) naïve Bayes (NB). These classifiers have been already implemented in MATLAB. The results of this comparison are reported in Table 5. As seen in Table 5, the MOPF classifier offers better DR and FAR as compared to SVM and NB.

TABLE IV. PERFORMANCE OF THE PROPOSED MODEL IN SIMULATED IOT

Method	DR (%)	FAR (%)
Anomaly detection module	80.95	5.92
Misuse detection module	96.20	1.44

TABLE V. PERFORMANCE COMPARISON OF DIFFERENT CLASSIFIERS IN MISUSE DETECTION MODULE

Classifier (used in misuse-based IDS)	DR (%)	FAR (%)
SVM	95.05	2.10
NB	81.00	9.37
CART	97.15	2.75
MOPF	96.20	1.44

On the other hand, the DR of CART is slightly better than MOPF (tantamount to 0.95 percent). However, the FAR of MOPF was improved by 1.31% as compared to that of CART.

V. Conclusion

The IoT is a worldwide network in which all heterogeneous objects around us can connect to the unreliable Internet by using a wide range of technologies. Since IoT is a hybrid network, which is composed of the Internet and the networks with heterogeneous nodes; hence, it provides accessibility to the Internet for all physical objects. Therefore, for the insecure nature of the Internet and also WSNs, which are the main components of IoT, implementing security mechanisms in IoT seems necessary. This paper proposed a novel hybrid architecture based on MapReduce for detecting both of insider and cyber attacks in IoT. The proposed model used a real-time anomaly detection module based on unsupervised OPF for detecting insider (internal) attacks which may be happened in 6LoWAPN. On the other hand, a misuse-based intrusion detection engine, which was based on supervised OPF, was responsible for detecting cyber (external) attacks that may be occurred from Internet (or LANs) side. The experimental results show the superior performance in simultaneous detection of the insider and cyber attacks in IoT.

References

[1] E. Borgia, "The Internet of things: Key features, applications and open issues," Computer Communications, vol. 54, pp. 1-31, Dec. 2014.

[2] M. De Sanctis, E. Cianca, G. Araniti, I. Bisio, and R. Prasad, "Satellite communications supporting Internet of remote things," IEEE Internet of Things Journal, vol. 3, pp. 113-123, Jan. 2016.

[3] T. Kushalnagar, G. Montenegro, and C. Schumacher, "IPv6 over low-power wireless personal area networks (6LoWPANs): Overview, assumptions, problem statement, and goals," RFC 4919, 2007.

[4] S. Raza, L. Wallgren, and T. Voigt, "SVELTE: Real-time intrusion detection in the internet of things," Ad Hoc Networks, vol. 11, pp. 2661-2674, Nov. 2013.

[5] L. Wallgren, S. Raza, and T. Voigt, "Routing attacks and countermeasures in the RPL-based Internet of things," International Journal of Distributed Sensor Networks, Article ID 794326, pp. 1-11, Jun. 2013.

[6] S. Sicari, A. Rizzardi, L.A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in Internet of things: The road ahead," Computer Networks, vol. 76, pp. 146-164, Jan. 2015.

[7] P. Kasinathan, C. Pastrone, M.A. Spirito, and M. Vinkovits, "Denial-of-service detection in 6LoWPAN based Internet of things," In: Proceedings of 9th International Conference on Wireless and Mobile Computing, Networking and Communications, Lyon, France, 2013.

[8] C. Jun and C. Chi, "Design of complex event-processing IDS in Internet of things," In: Proceedings of the 6th International Conference on Measuring Technology and Mechatronics Automation, Zhangjiajie, China, 2011.

[9] J.P. Papa, A.X. Falcão, and C.T.N. Suzuki, "Supervised pattern classification based on optimum-path forest," International Journal of Imaging Systems and Technology, vol. 19, pp. 120-131, Jun. 2009.

[10] C.R. Pereira, R.Y.M. Nakamura, K.A.P. Costa, and J.P. Papa, "An optimum-path forest framework for intrusion detection in computer networks," Engineering Applications of Artificial Intelligence, vol. 25, pp. 1226-1234, Sep. 2012.

[11] L.M. Rocha, F.A.M. Cappabianco, and A.X. Falcão, "Data clustering as an optimum-path forest problem with applications in image analysis," International Journal of Imaging Systems and Technology, vol. 19, pp. 50-68, Jun. 2009.

[12] K.A.P. Costa, L.A.M. Pereira, R.Y.M. Nakamura, C.R. Pereira, J.P. Papa, and A.X. Falcão, "A nature-inspired approach to speed up optimum-path forest clustering and its application to intrusion detection in computer networks," Information Sciences, vol. 294, pp. 95-108, Feb. 2015.

[13] W.P. Amorim and M.H. Carvalho, "Supervised learning using local analysis in an optimal-path forest," In: Proceedings of the 25th Conference on Graphics, Patterns and Images, Ouro Preto, Brazil, pp. 330-335, 2012.

[14] H. Bostani, Intrusion detection and identification of attacks in the Internet of things using a combination of machine learning methods, M.Sc. Thesis, Dept. Comp. Eng., Islamic Azad University-South Tehran Branch, 2015.

[15] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," In: Proceeding of 6th Symposium on Operating Systems Design and Implementation, San Francisco, USA, 2004.

[16] S. Aridhi, P. Lacomme, L. Ren, and B. Vincent, "A MapReduce-based approach for shortest path problem in large-scale networks," Engineering Applications of Artificial Intelligence, vol. 41, pp. 151-165, May 2015.

[17] M. Tavallaee, E. Bagheri, L. Wei, and A. Ghorbani, "NSL-KDD Data Set" (Available on http://www.unb.ca/research/iscx/dataset/iscx-NSL-KDD-dataset.html), [Accessed on 28 Feb. 2016]

[18] H. Bostani and M. Sheikhan, "Hybrid of binary gravitational search algorithm and mutual information for feature selection in intrusion detection systems," Soft Computing, pp. 1-18, Published online 28 Nov. 2015. DOI: 10.1007/s00500-015-1942-8