A Security Framework for Clone Detection in Internet of Things

[1]Kashimalla Naresh, M. Jayapal , B.Ravi Raja

Student, Department of Computer Science and Engineering,Malla Reddy college of Engineering and Technology, Hyderabad, Telangana. India

Associate Professor, Department of Computer Science and Engineering,Malla Reddy college of Engineering and Technology, Hyderabad, Telangana. India

Associate Professor, Department of Information Technology,ANURAG University, Hyderabad, Telangana. India

Abstract

In the context of Internet of Things (IoT), cloning is very serious threat which needs to be handled. The purpose of this project is to propose a security framework known as Multi-Model based Clone Detection Framework (MM-CDF). It also defines a novel clone detection method known as Probabilistic Clone Detection (PCD) which employs efficient probability analysis to detect clones in IoT scenarios. The proposed method can detect clones and take necessary steps to ensure security in IoT environments. The existing clone detection methods suffer in false positives in finding clones. The proposed system overcomes this problem with probabilistic analysis which reduces false positives and ensures that the clones are correctly identified. Since IoT use cases deploy thousands of connected devices, it is essential to have such solution. This project is realized with a prototype application that simulates the nodes in IoT and detection of clones with the aforementioned approach.

Keywords – Internet of Things (IoT), security, clone attacks, clone detection

1. INTRODUCTION

Internet of Things (IoT) is an emerging networking paradigm, in which a large number of interconnected devices communicate with each other to facilitate communications between people and objects. For example, a smart city is composed of several smart sectors, such as smart homes, smart hospitals, and smart cars, which are significant applications of IoT. In a smart home scenario, each IoT gadget is equipped with embedded sensors and wireless communication capabilities. The sensors are able to gather environmental information and communicate with each other, as well as the house owner and a central monitoring system. In a smart hospital scenario, which could be implemented using body sensor networks, patients wear implantable sensors that collect body signals and send the data to a local or remote database for further analysis. As another example, in

a smart traffic scenario embedded sensor in cars are able to detect accident events or traffic information, and collaboratively exchange such information.

In the context of Internet of Things (IoT), closing is very serious threat which needs to be handled. The purpose of this project is to propose a novel clone detection method known as correlation based clone detection which is based on correlation based technique. The proposed method can detect clones and take necessary steps to ensure security in IoT kind of scenarios. Our contributions in this paper are as follows.

1. An algorithm named Probabilistic Clone Detection (PCD) is proposed and implemented.
2. A prototype application is built to demonstrate proof of the concept.
3. The algorithm is evaluated and the results are compared with the state of the art.

The remainder of the paper is structured as follows. Section 2 provides review of literature. Section 3 presents the proposed system in detail. Section 4 presents implementation details. Section 5 presents experimental results while section 6 concludes the paper besides giving directions for future work.

2. RELATED WORK

In the case of static networks, a popular approach for detecting clones is witness finding. In essence, the idea behind witness finding is that the existence of clones must lead to location conflicts. More specifically, each node u collects the location information, L(v), of its neighbouring nodes, e.g., v, and sends the collected location claims to some selected nodes. Nodes receiving two location claims with the same ID v, but with two distinct locations, will serve as witness nodes, and witness the location conflict. The witness finding strategy not only detects the existence of clones, but also identifies the clone IDs [1].

Indeed, a centralized security monitoring solution is perfectly in line with the hierarchical architecture fostered by such technologies, which are currently being supported by key players, including among others Cisco and Orange. For instance, the current Lora WAN deployment being developed in the city of Rome concentrates all IoT sensor traffic collected by several tens of radio stations spread across the whole of the Rome municipality and relevant neighbours in a (logically) single centralized network server, which therefore appears to be a natural candidate to further host anomaly detection approaches such as MDSClone [1].

An inherent weakness among all of the witness finding-based approaches is the assumption of the knowledge of location information available for each node. A couple of solutions take alternative approaches to detect clones, such as the social fingerprint, pre-distributed keys, and random clustering methods [2]. Many researchers found in the literature [3]- [16] provided different methods for clone detection. In this paper, we proposed a correlation based approach for the same.

3. PROPOSED SYSTEM

The proposed system is meant for the proposal and implementation of a clone detection method. The detection method is named as MDSClone (extended). It is based on the concept of multi-dimensional scaling. It is distributed in nature and can scale to very big networks like IoT based networks. The efficiency of the MDSClone (extended) is evaluated against some existing methods.

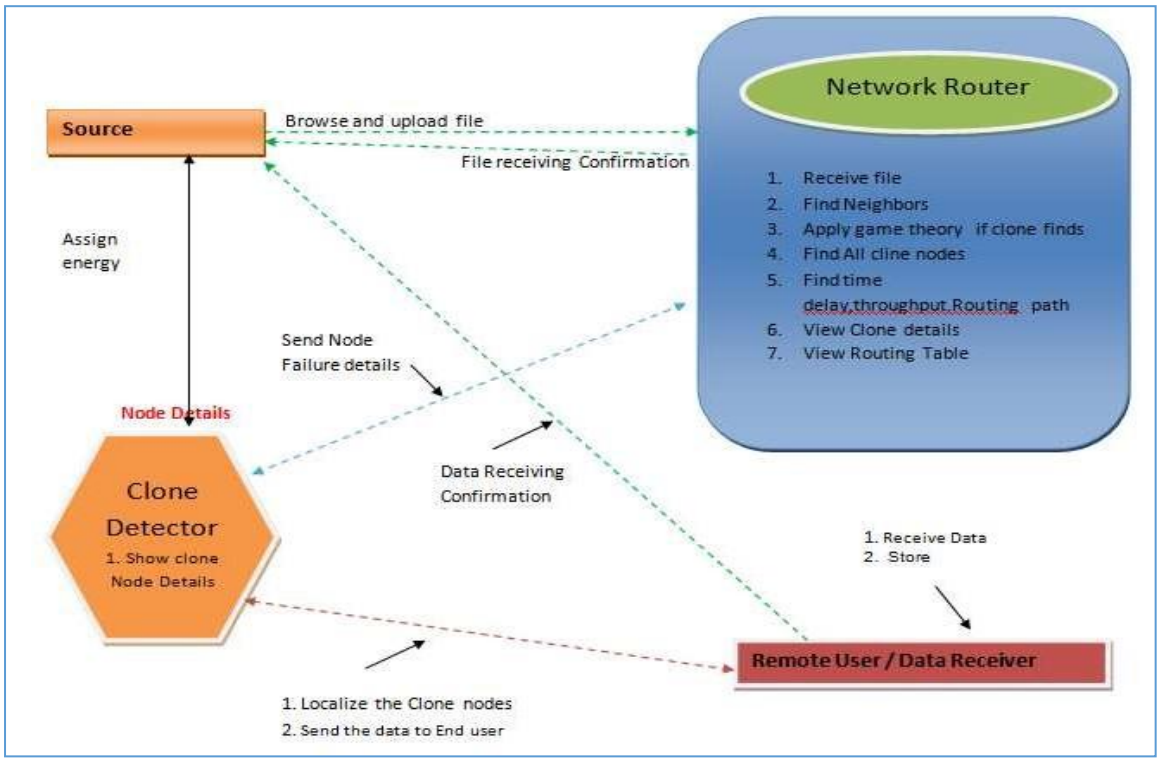

Figure 1: Architecture Diagram

As shown in the above figure 1 There are different components in the diagram there is a source component that sends the file to the network router component and there is remote user and data receiver component which receives data and store in it. And another component is clone detector its aim is to show clone node details ad send node failure details and localize the clone nodes and send the data to end user.

In the sender module, the Sender will browse the file, Initialize the nodes, distribute Mac address for every node and then upload to the particular Receivers (receiver1, receiver2, receiver3 and receiver4). And router will connect to the particular receiver. After receiving successfully, it will give response to the sender. The Sender can have capable of manipulating the data file. The Router module manages multiple nodes (node A, node B, node C, node D, node E, node F….) to provide data storage service. In a router we can view the node details, assign cost and view clones. The sender will upload data file to the router, the Router will select the smallest distance path and send to the particular receiver. If any clone is found in a particular node, the route replay will send to the Trusted Authority and then it will select another path.

In a router service provider can view the node information details and view the routing table details.

In the trusted authority module, the Trusted Authority is responsible for identify the intrusion in the network. If the router found any type of clones, then it transfers the flow to Trusted Authority. Then the Trusted Authority is

responsible for capturing the clones and identifies which type of clone (fake key clone, Destination IP clone and cost clone) and then response will send to the router. After getting a response from the TA, router will select other path and send to the particular receiver (receiver1, receiver2, receiver3 and receiver4). The Trusted Authority will make a list of failed node details and then all failed nodes are stored with tags such as node name, IP address, MAC address, node cost, time and date.

In the receiver module, there are an n-numbers of receivers are present (receiver1, receiver2, receiver3 and receiver4). All the receivers can receive the data file from the sender via router. The sender will send data file to router and router will select the lesser distance path and send to the particular receiver (receiver1, receiver2, receiver3 and receiver4), without changing any file contents. The receivers may try to receive data files within the router or network only. In the clone module, the clone can attack the node in three ways fake node clone, Destination IP clone and cost clone. Fake key clone means he will inject fake key to the particular node; IP clone means he will change the destination IP address to the particular node, cost clone means he will inject fake cost to the particular node.

3.1 Algorithm

An algorithm named Probabilistic Clone Detection (PCD) is proposed and implemented to improve the state of the art.

Algorithm: Probabilistic Clone Detection (PCD)

Input: Network N, Configuration C

Output: Detection of duplicate (cloned) nodes

1. Start
2. Initialize network vector N’
3. Initialize configuration vector C’
4. Initialize found to false
5. For each node n in N
6. Update C
7. End For
8. On arrival of new node
9. Register new node with the network
10. Update N and C
11. For each node n in N
12. For each configuration c in C
13. IF c is not associated with n Then

```
14.        Found=true
15.      Else
16.        Found = false
17.     End If
18.   End For
19.   IF found=false Then
20.      Add n to N'
21.     Add c to C'
22.   End If
23. End For
24. End
```

Algorithm 1: Probabilistic Clone Detection (PCD)

As shown in Algorithm 1, the nodes are verified with correlations. The suspected nodes are kept in N' and that is able to show the cloned nodes.

4. IMPLEMENTATION DETAILS

Implementation of the project is made with Java programming language and networking related classes. It encapsulates a networking application to demonstrates a network that will prevent cloning attacks.

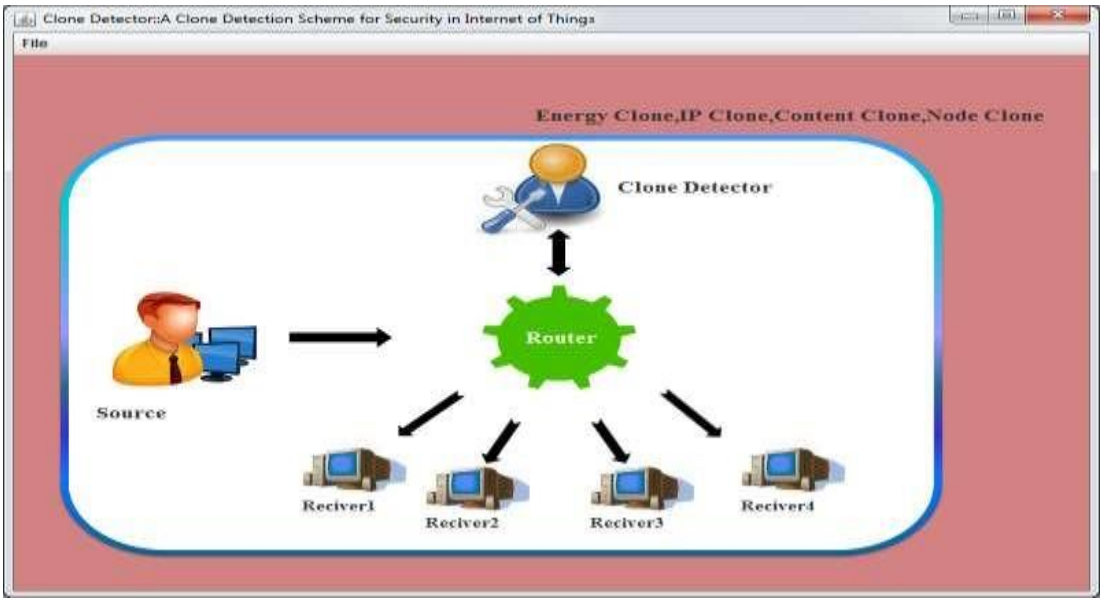

Figure 2: Clone Detection Scheme

As show in Figure 2, there is a router node and many receiver nodes. There is a source for sending data and there is a clone detector.

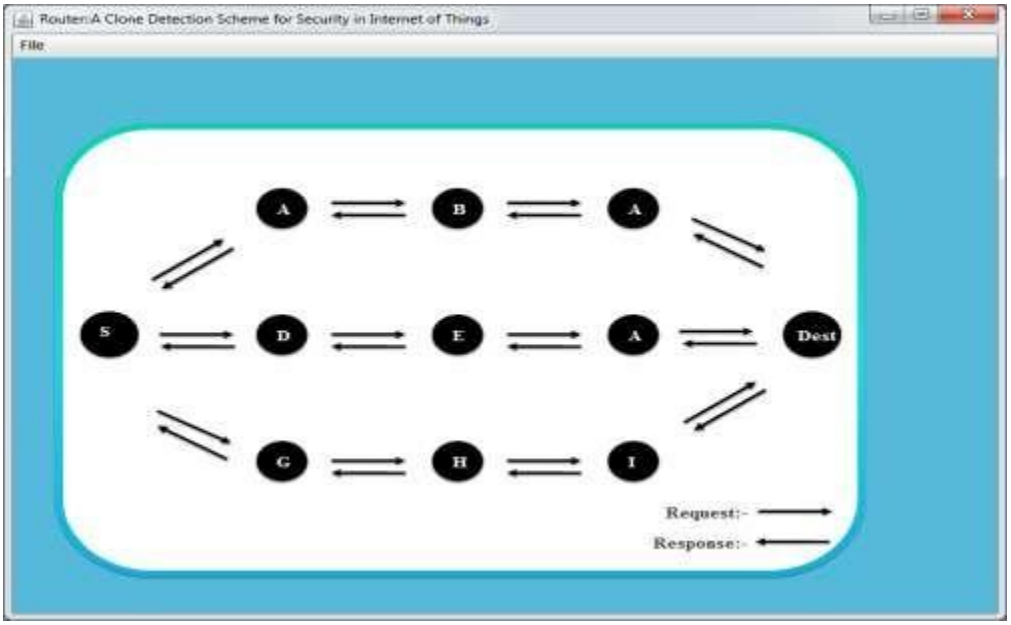

Figure 3: Router

As shown in the above figure 3 there is simulation between the source and destination. There is also a request and response between the source and destination as shown in the figure.

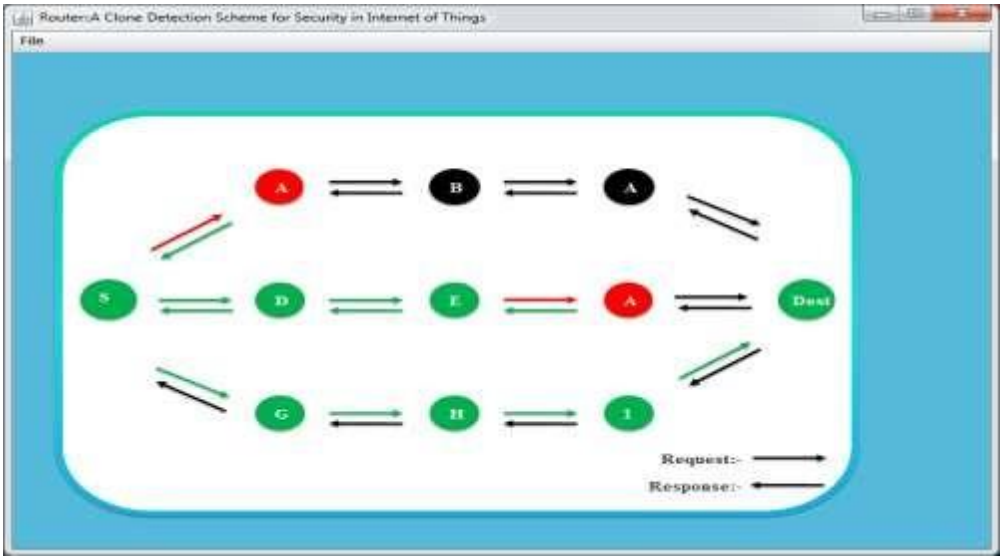

Figure 4: Router

As shown in the above figure 4 there is a runtime simulation between the source and destination in the file. There is also a request and response between the source and destination as shown in the figure.

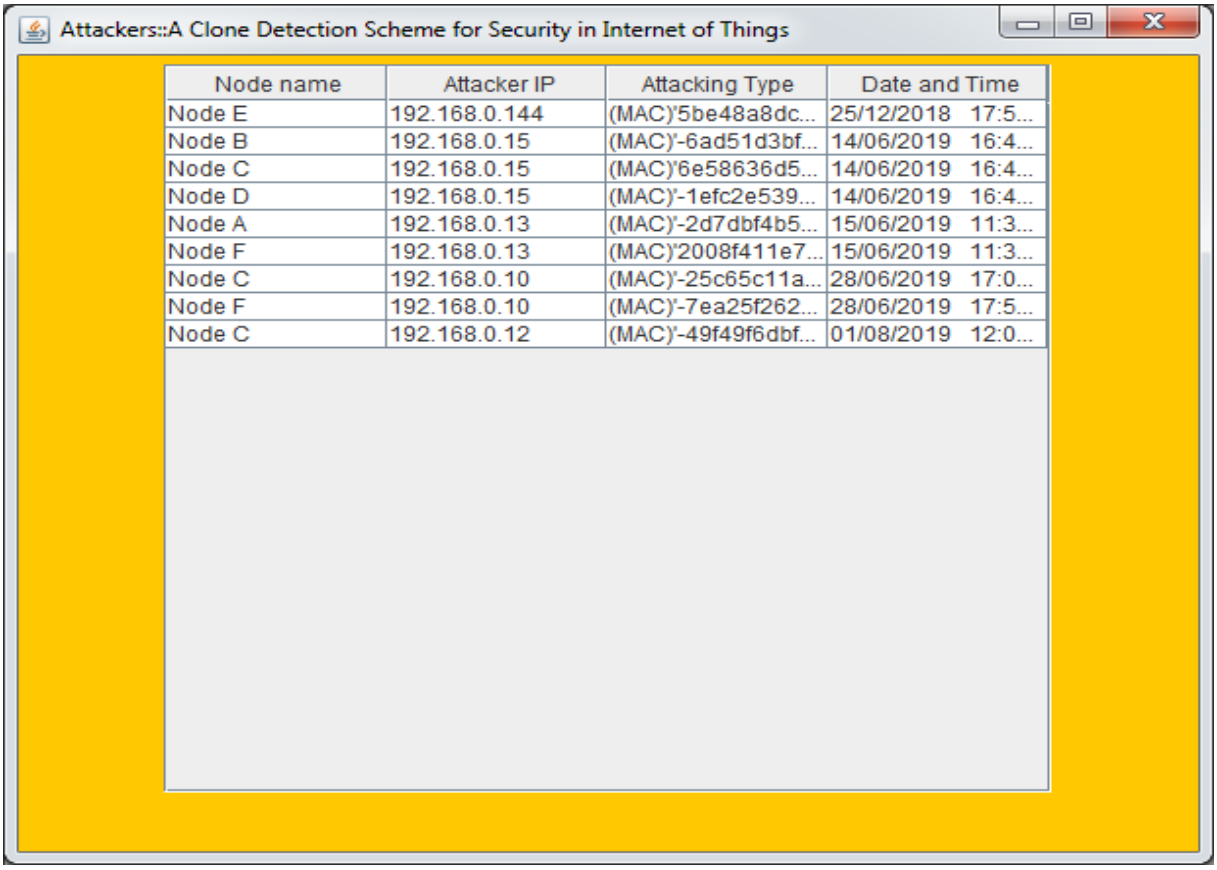

Figure 5: Attackers list

As show in Figure 5 there is table containing the attackers list and which node is attacked and also attackers IP and attackers type with including date and time.

5. EXPERIMENTAL RESULTS

Experiments are made with the prototype application. The application is evaluated with different experiments. The application's performance in terms of detection probability and computational time.

No. of Nodes	Detection Probability	
	Existing	Proposed
2000	0.9	1
4000	0.9	1
6000	0.85	1
8000	0.8	1
1000	0.75	1

Table 1: Detection Performance

As shown in table 1 the experimental results of existing and proposed are shown inn table above.

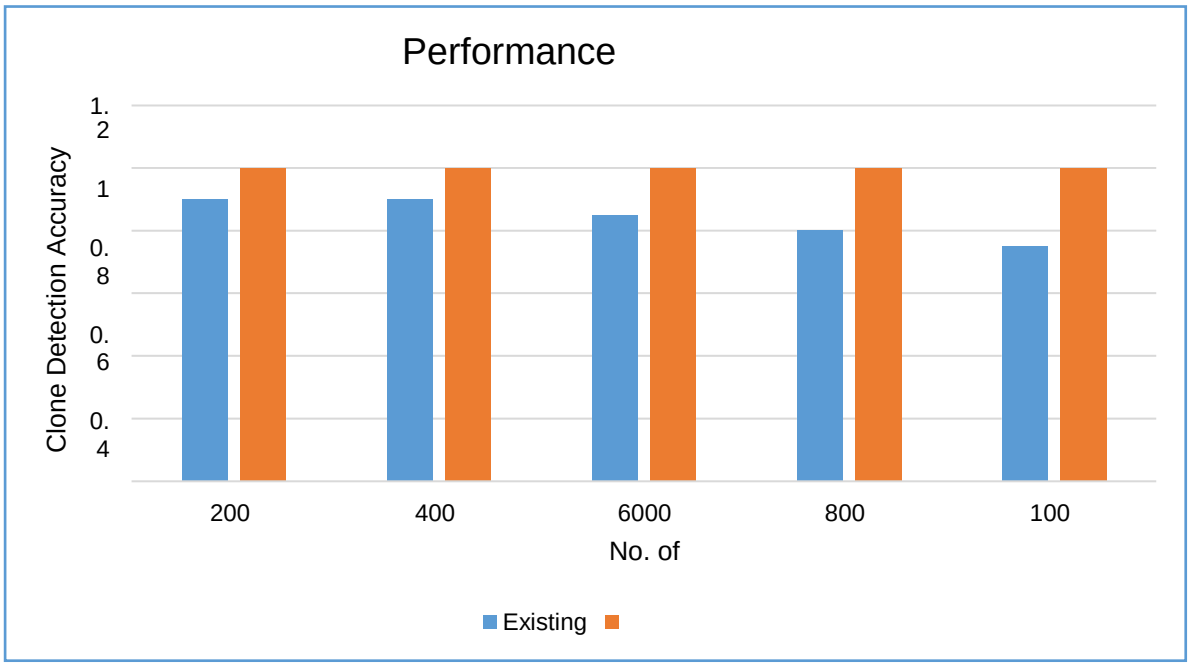

Figure 6: Detection Performance

As show in above Figure 6 there is comparison between the number of node in total in x-axis and detection probability in y-axis with including the existing and proposed difference bars.

No. of Nodes	Computation Time (seconds)		
	MDS - using TI Existing	MDS - using TI Proposed	Conventional MDS Existing
1000	0	5	0
4000	20	25	30
8000	40	45	150
10000	50	55	250

Table 2: Computational performance

As shown in table 2 the experimental results of MDS - using TI Existing, MDS - using TI Proposed and Conventional MDS Existing are shown in table above.

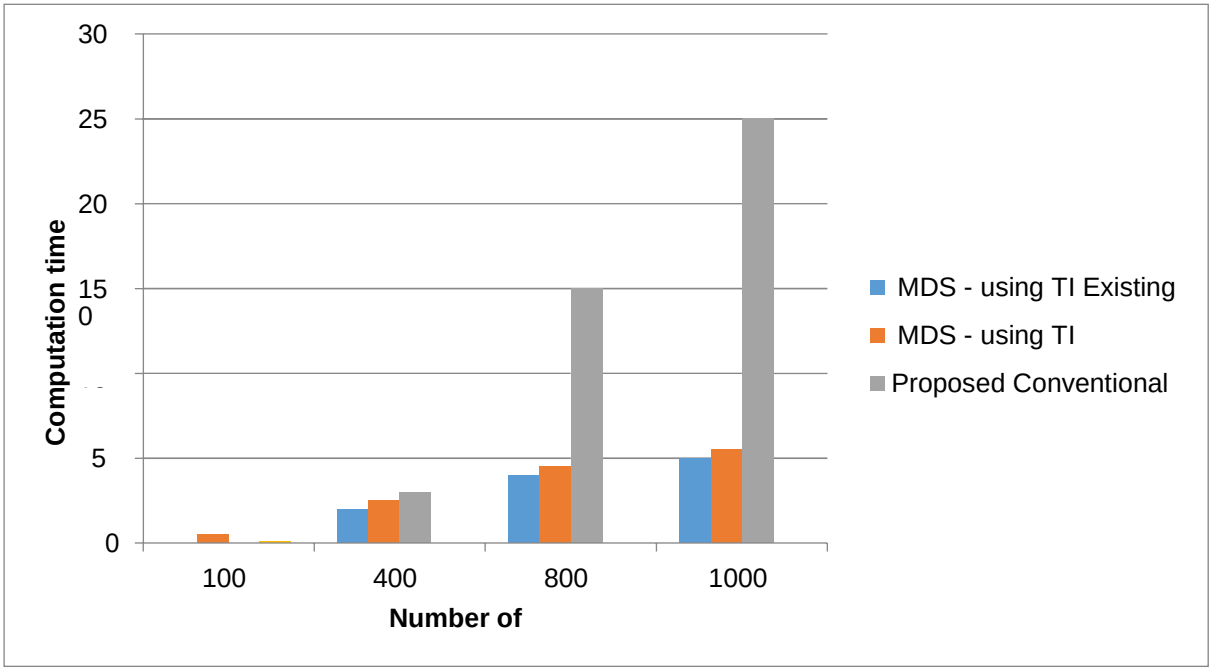

Figure 7: Computational performance

As show in above Figure 7 there is comparison between the number of nodes in x-axis and computation time (seconds) in y-axis with including the MDS – using it existing bar, MDS – using it proposed and Convectional MDS existing difference bars.

6. CONCLUSION AND FUTURE WORK

Multi-Model based Clone Detection Framework (MM-CDF) is proposed and implemented in this paper. It makes use of the abnormal behaviour of a node and suspect as clone and also uses the algorithm named as Probabilistic Clone Detection (PCD). When multiple nodes are involved in an IoT based use case, it is essential to safeguard it from different attacks. In this paper, the focus is on the detection of clone attack which is a commonly launched attack to have illegal access to IoT integrated application by hackers. The clone detection process makes use of correlations among the nodes in the existing configuration and the configuration that is based on the initial registration of nodes. Based on this the probability of the node being clone is determined. A prototype application is built to simulate number of nodes in a network where nodes behave like sender, receiver and router. In the process of analysis, the PCD algorithm finds possible duplicate (cloned) nodes in the system. The resuts revealed that the performance of the proposed system is better than existing. In future, we intend to realize this algorithm with a real test bed instead of a custom simulator.

References

[1] Po-Yen Lee, Chia-Mu Yu, Tooska Dargahi, Mauro Conti, and Giuseppe Bianchi. (2018). MDSClone: Multidimensional Scaling Aided Clone Detection in Internet of Things. *salford manches*, p1-17.

[2] A. Solanas, C. Patsakis, M. Conti, I. Vlachos, V. Ramos, F. Falcone, O. Postolache, P. Perez-martinez, R. Di Pietro, D. Perrea, and A. Martnez-Balleste, “Smart health: a context-aware health paradigm within

smart cities," IEEE Communications Magazine, vol. 52, no. 8, pp. 74–81, 2014.

[3] Y. Wang, G. Attebury, and B. Ramamurthy, "A survey of security issues in wireless sensor networks," IEEE Communications Surveys & Tutorials, vol. 8, no. 2, pp. 2–23, 2006.

[4] M. Conti, R. D. Pietro, L. V. Mancini, and A. Mei, "Distributed detection of clone attacks in wireless sensor networks," IEEE Transactions on Dependable and Secure Computing, vol. 8, no. 5, pp. 685–698, 2011.

[5] M. Conti, R. Di Pietro, and A. Spognardi, "Clone wars: Distributed detection of clone attacks in mobile wsns," Journal of Computer and System Sciences, vol. 80, no. 3, pp. 654–669, 2014.

[6] M. Conti, "Clone detection," in Secure Wireless Sensor Networks. Springer, pp. 75–100, 2016.

[7] A. K. Mishra and A. K. Turuk, "A comparative analysis of node replica detection schemes in wireless sensor networks," Journal of Network and Computer Applications, vol. 61, pp. 21–32, 2016.

[8] W. T. Zhu, J. Zhou, R. H. Deng, and F. Bao, "Detecting node replication attacks in wireless sensor networks: a survey," Journal of Network and Computer Applications, vol. 35, no. 3, pp. 1022–1034, 2012.

[9] Z. Chen, F. Xia, T. Huang, F. Bu, and H. Wang, "A localization method for the internet of things," The Journal of Supercomputing, pp. 1–18, 2013.

[10] O. Bello and S. Zeadally, "Intelligent device-to-device communication in the internet of things," IEEE Systems Journal, vol. 10, no. 3, pp. 1172–1182, 2016.

[11] C.-M. Yu, Y.-T. Tsou, C.-S. Lu, and S.-Y. Kuo, "Localized algorithms for detection of node replication attacks in mobile sensor networks," IEEE Transactions on Information Forensics and Security,, vol. 8, no. 5, pp. 754–768, 2013.

[12] C.-M. Yu, C.-S. Lu, and S.-Y. Kuo, "Efficient and distributed detection of node replication attacks in mobile sensor networks," in VTC'09. IEEE, 2009.

[13] K. Xing and X. Cheng, "From time domain to space domain: Detecting replica attacks in mobile ad hoc networks," in INFOCOM'10, 2010.

[14] Y. Shang, W. Ruml, Y. Zhang, and M. P. Fromherz, "Localization from mere connectivity," in MobiHoc'03. ACM, pp. 201–212, 2003.

[15] B. Parno, A. Perrig, and V. Gligor, "Distributed detection of node replication attacks in sensor networks," in IEEE Symposium on Security and Privacy. IEEE, pp. 49–63, 2005.

[16] B. Zhu, S. Setia, S. Jajodia, S. Roy, and L. Wang, "Localized multicast: efficient and distributed replica detection in large-scale sensor networks," IEEE Transaction on Mobile Computing, vol. 9, no. 7, pp. 913–926, 2010.