International Journal of Electrical and Computer Engineering (IJECE)
Vol. 11, No. 4, August 2021, pp. 3255~3266
ISSN: 2088-8708, DOI: 10.11591/ijece.v11i4.pp3255-3266

Collaborative intrusion detection networks with multi-hop clustering for internet of things

Ida Wahidah, Yudha Purwanto, Aditya Kurniawan
Department of Electrical Engineering, Telkom University, Indonesia

Article Info

Article history:

Received Jul 11, 2020
Revised Dec 19, 2020
Accepted Jan 13, 2021

Keywords:

Agent-based modeling
CIDN
Internet of things
JADE
Network security

ABSTRACT

Internet of things (IoT) is an emerging topic in so many aspects nowadays. The integration between devices and human itself is currently in large scale development. With the continuous applications of the IoT, the hidden problems such as security threats become one of the key considerations. Furthermore, limited power and computational capability of the devices in the system make it more challenging. Therefore, the needs of reliable and effective security system throughout the networks are highly needed. This research proposed a collaborative system based on JADE that consists of 3 types of agent, which are IoT server, controller, and node. Every agent will collaborate each other in terms of exchanging the intrusion detection results. The collaboration between the agents will provide more efficient and good performance. Four classification algorithms were used to model IDS functions. Then, the performance evaluation was done on the system with several parameters such as cost loss expectation, energy consumption, and metric of IDS efficiency. The result shows the number of reports sent by IoT controller were decreased up to 80% while preserving the security aspect.

Corresponding Author:

Ida Wahidah,
Department of Electrical Engineering
Telkom University
Telekomunikasi no. 1 st, 40257, Bandung, West Java, Indonesia
Email: idawahidah@telkomuniversity.ac.id

1. INTRODUCTION

The rise of IoT topics has been high lately as it has so many positive effects for daily routine. Generally, the IoT terms is used to denote the advanced connectivity between devices and services that goes beyond the traditional machine to machine and covers a variety of protocols and applications [1]. The word itself first mentioned by Kevin Ashton on his presentation in 1999 [2] and initially were called as "Ubiquitous Computing" by Mark Weiser in 1991 [3]. In generals, internet of things itself consist of 3 layer based on OSI model [4]. They are perception layer, network layer, and application layer. Perception layer or also known as sensor layer located on the bottom of this model because it interacts with physical devices such as RFID or sensor. Network layer works with the same functionality as transmission layer where it connects perception and application layer, it transmit the data between those two layers. Last, application layer or also known as business layer, is the highest position of this model, it accepts and processes the data from network layer to specific service. With the continuous development and application of the internet of things, it is own hidden problems increasingly appear [5], one of these is security. It is one of many aspects we should concern in internet of things area which has massive connectivity. Also, because of the intrusions have become much more sophisticated and hard to detect [6], a reliable and effective security system through the networks is highly needed either to enhance existing method or a new one.

Problems in network security not only come from those connected devices itself, but also from network architectural and system model. Especially in internet of things case, most of the device that connected to the network has small and limited processing power and tight energy restrictions [7]. One of the solutions is implementing an intrusion detection system (IDS) in order to prevent intruder from accessing the networks. IDS is a system that determines an intrusion into system through the observation of available information concerning the state of system, monitoring user activities, and reporting to a management station [8]. A typical IDS needs a high-performance computing power that runs in a large system or networks. These high specifications are useful for monitoring and analyzing the data that flows over the networks. In reality, the solution stated above is not suitable for IoT networks which is characterized by massive connectivity and limited processing power. Therefore, to solve these issues, it is necessary to distribute the computational load and intrusion detection resource in a distributed environment. One of the approaches is using collaborative intrusion detection network (CIDN).

The needs of CIDN in IoT networks is to improve the efficiency of existing standalone IDS become able to exchanges information between IDSs on the same network. With the concept of intrusion detection resource sharing and distributed workload, the system can analyze which actions to take. In our proposed method, clustering process on CIDN were used to reduce the reporting mechanism and to have a better result in terms of energy consumption and intrusion detection performance especially in IoT environment. So, in order to address the problem, the key contributions of this research are presented below.

- A design of CIDN architecture with a clustering process in IoT scenario.
- The proposed design must consider the energy restriction and security aspect in order to reach the low network overhead ratio indicated by a reduced reports packets from IoT controller to IoT server without degrading the security aspects.

Issues formulated in this paper include network security in IoT where the problem comes from the nature of IoT network itself which has massive connectivity causing more security gaps in the system. One of the solutions is to implement IDS in this system so the intrusion can be detected and handled. But with the large-scale network characteristic, intrusion handling in IDS will goes ineffective because it requires high-level computing capabilities and affects the system by its heavy workload. It is important to overcome these problems because in IoT, one of the main concerns is the data integrity. So, in order to solve the problem in a large-scale network and workload, collaborative scheme of IDS with clustering can be implemented for achieving a good performance in intrusion detection capabilities. In section 2, we will explain about our literature study.

2. RELATED WORKS

Collaborative work between IDSs in the networks has been studied on several research recently. Every system has its own characteristics depends on the scenario of the author wants. This idea of collaborative works of IDS in the same networks originally came from Fung's dissertation [9]. Vasilo Manolakis in his PhD thesis [10] also mentioned the collaborative works between IDSs but with different words (collaborative intrusion detection system). Between them, the terms of collaborative has the same meaning and function, which means collaborating each IDS resources to get higher performance between them. Further survey and experiments has been conducted by several researcher such as Trust-aware CIDS challenges [11], Bayesian method were used in CIDN model [12]. Also, there are several research that use artificial intelligence to boost the performance of CIDN such as enhancing CIDN using supervised intrusion sensitivity-based trus management model [13] and the concept of intelligent collaborative network intrusion detection [14]. In terms of the development of CIDN in IoT, there are several research that has been done in the scenario such as COLIDE [15]. Meanwhile, our agent-based modeling of the intrusion detection system was came from Orfila's research [16].

Our proposed system idea mainly came from Fung's [17] and Orfila's research [16]. Then we combine and adjust several parameters and scenario so that we can reach our goal. Some calculation was based on those researches meanwhile the other is not. Since the condition between both of the referenced research and ours is not the same, we try to compare our proposed system with others that does not implement our model but in the same condition. From that scenario, we hope to see the influence of our proposed system in terms of efficient communication between each node.

3. RESEARCH METHOD

3.1. System model

This research general system architecture consist of 3 parts, namely IoT node, IoT controller, and IoT server. IoT node takes a role in aggregating the sensing data of IoT-related units where it may consists of

single-board micro controller device (e.g Arduino UNO and ATmega). These devices usually comes in large number in any IoT environment (e.g smart farm and smart house). IoT controller functions as IoT node data aggregator and detection unit for the CIDN, whereas IoT server is the mastermind of the system. It acts as a data processor as well as security analyzer. Figure 1 shows the proposed system architecture.

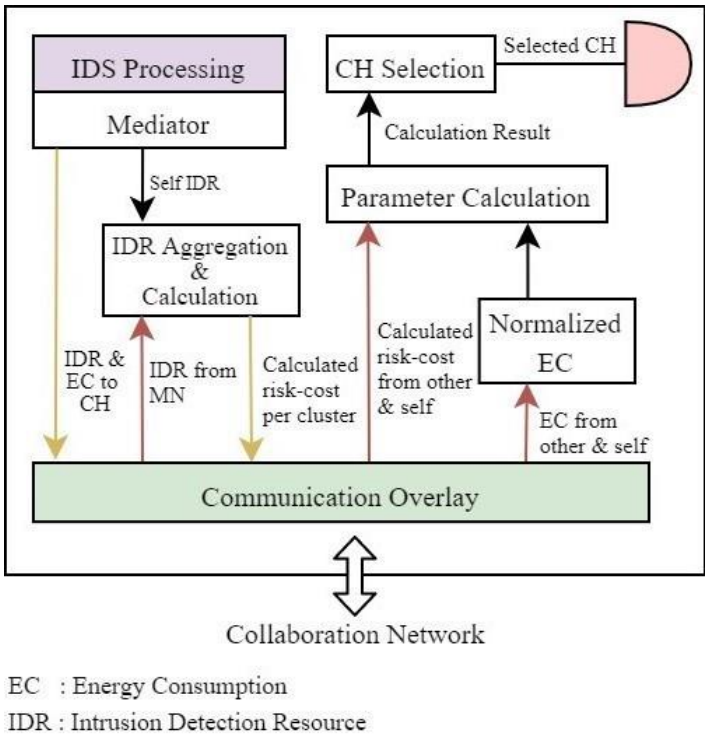


Figure 1. Proposed system architecture

The proposed clustering process starts when the IDS were done detecting incoming packets to their network interfaces. Then, every node in the cluster sends their data to the head and later calculate them. The outputs are the IDS and energy consumption which is then treated both as a ratio of security and energy. Then the final calculation of both parameters is done by simple addition. By selecting the smallest value of calculation, we determine the cluster head for the next period.

Figure 2 shows the topology of proposed clustered CIDN. There could be more than one CH in each layer, depends on how each IoT environment is regulated. On top of the figure there is an IoT server which is a single computer with high specifications. Inside each layer, there could be different approach in IDS implementation. This is because there is a possibility where one IoT environment consists of several different devices. So, in order to accomplish the main goal of CIDN in terms of collaborative work, this method distributes the intrusion detection resource throughout the system. The number of controllers depends on how large the system is. As shown in Figures 1 and 2 this research focused on several processes that occurs on IoT controller, where the system was modeled into agent-based design. The basic concept on how system works is as follows:

1. IoT nodes only sends traffic to IoT controller by aggregating their data to each CH.
2. IoT controller receives data and process it with IDS mechanism and discuss the results with other controller in same cluster to decide which node needs to send report to IoT server.
3. IoT server receives the report and could do next step action such as IRS (Intrusion Responsive System).

Based on [18-20], this approach modeled the IoT traffic as aggregated periodic with asynchronous sources, where the sampling rate is identical for every traffic sent by IoT nodes. Periodic traffic is much easier to model rather than event-driven traffic, because the data always sent every *t* time units. While asynchronous sources were chosen because in reality, IoT devices tend to send data independently in the actual environment. As stated in [19, 20], Poisson approximation can be used to model IoT traffic as long as the amount of the node is above the threshold *n* to keep the bias which is produced by the poor approximation process.

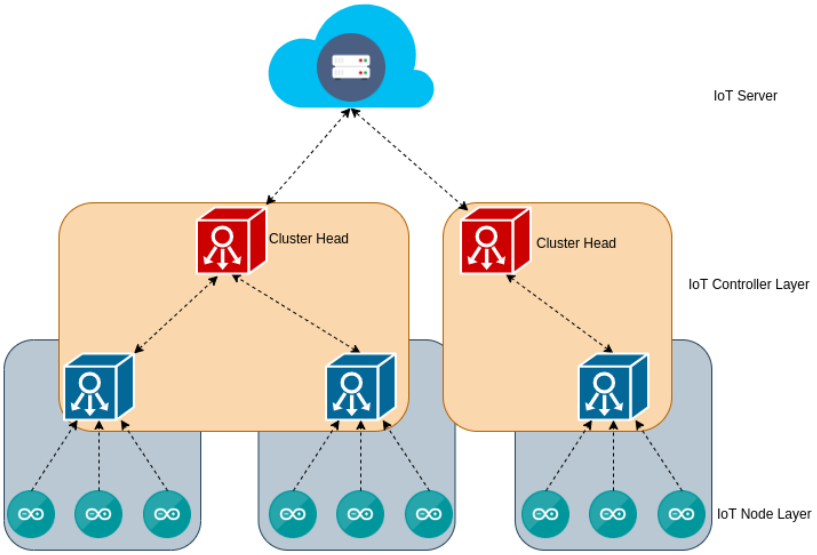


Figure 2. Proposed system topology

3.2. Agent-based modeling

In order to simulate the system, we use java agent development (JADE) framework as the simulation tools. JADE is a software environment written in JAVA and made to build agent systems for the management of networked information resource in compliance with foundation of intelligent physical agents (FIPA2000) specification for inter-operable multi-agent system (MAS) [21]. The system was modeled into agent-based modeling as several agents according to its role on the system. Only 2 parts of the system that were modeled, they are IoT server and controller. IoT server consist of 2 agents named main and container agent. Main agent functions as the main receiver for all of the report throughout the system while container agent does the communication with corresponding cluster. Meanwhile, the IoT controller were modeled to detection agent (DA). For each DA, they read the incoming packets generated by IoT nodes and classified them into whether normal or anomaly. Several classification algorithms are used for this process and were randomly distributed on each controller in the same cluster to model the intrusion detection process. Figure 3 shows the agent-based modeling of our proposed system.

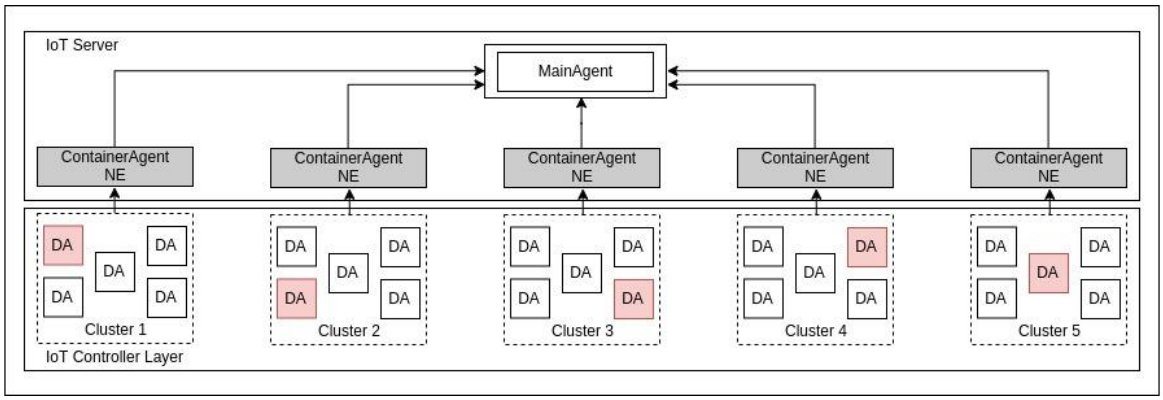


Figure 3. CIDN on agent-based modeling

3.3. Collaboration process

The collaboration on the proposed system located on how the cluster head will be selected. This process will involve all IoT controller in each cluster. Each of them exchanges the intrusion detection resource such as detection accuracy, false positive rate and false negative rate. Besides the intrusion detection resource, they also exchange their energy consumption. Both of those parameters will be used as reference

for the calculation of cluster head selection process. While the intrusion detection resource is used as the base for risk-cost calculation, the energy consumption used as the representative of IoT network characteristic where the energy restriction exists on several layer. This process is visualized in Figure 4(a). Meanwhile, the selection process will be done in order to select the lowest value of the addition between risk-cost calculation and energy consumption. All of this process aims to bring the lowest possible energy consumption while maintaining a good security aspect, which is represented by the value of IDS metric. This process is presented in Figure 4(b).

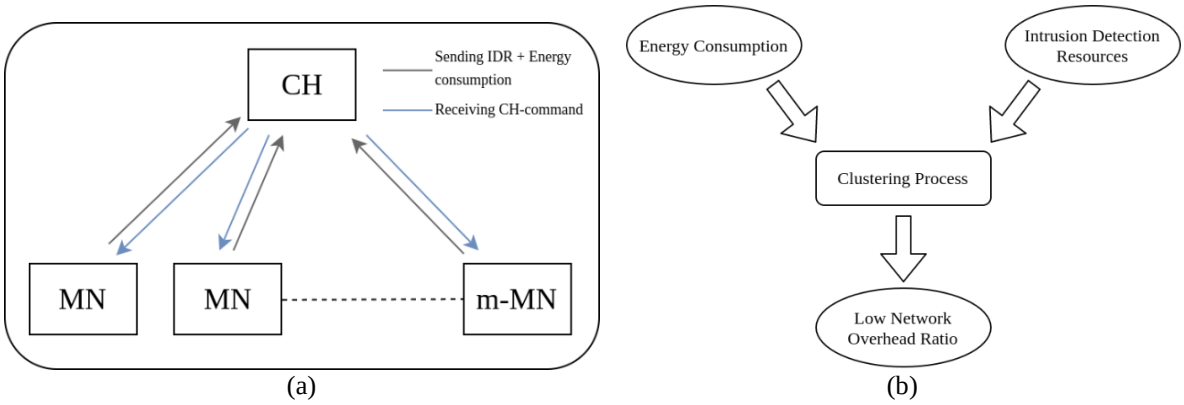


Figure 4. Collaboration: (a) Process visualization and (b) Parameter visualization

Table 1 shows the algorithm of the pseudo-codes of CH and CM respectively. The process listed below is the main contribution of this research. It functions as the procedure of collaboration process between each IoT controller in one cluster. The steps consist of how the resource will be processed and transferred between CH, CM, and IoT server on every period.

Table 1. Algorithm IoT controller task

Cluster Head Task	Cluster Member Task

t : period
t_{max} : maximum period
p : incoming packets from IoT node
nbr : neighbouring node
 	
t = 0
for every t **until** t_{max} **do**
  if t == 0 **then**
  trainIDS(p,t)
  t++
  else
  testIDS(p,t)
  t++
  end if
 	
  for every nbr **do**
  requestresource(nbr)
  end for
  while $\sum reply != \sum nbr$ **do**
  storeresult()
  end while
  if $\sum reply == \sum nbr$ **then**
  calculate$M$()
  calculateenergy()
  normalizeenergy()
  calculateR_{CHS}()
  end if
  selectlowestR_{CHS}()
  sendresult(IoTServer)
  CHack(SelectedCM)
end for
	t : period
	t_{max} : maximum period
	p : incoming packets from IoT node
	nbr : neighbouring node
	
	t = 0
	for every t **until** t_{max} **do**
	  if t == 0 **then**
	  trainIDS(p,t)
	  t++
	  else
	  testIDS(p,t)
	  t++
	  end if
	
	  depends on (incoming messages)
	  requestsource : sendresult(CH)
	  CHack : doCHtasks()
	end for

3.4. Clustering design

Clustering process in this system takes part as a mechanism to reduce the reporting node and selecting the best detection accuracy from each cluster. This research proposed method only occurs in the IoT controller, not IoT node. In order to minimize the required connection to the upper layer, only cluster head is tasked to report. Like other clustering algorithms, this method aims to select the best performed node with high detection accuracy and low energy consumption in order to reserve more energy for the network. Initial clustering was done by IoT server and it is static, so the members would not change over time. Static cluster produces more stability than dynamic, because with regards to intrusion detection process, the underlying network that provides data would play a vital role on the detection accuracy variation. The cluster consists of two main roles, i.e., cluster head and members. Each of them has different task in the clustering process as described below:

- Cluster head

As the main actor of the clustering activities, cluster head responsible for calculating each member node's risk cost by parsing the data sent by them. The reporting task between each layer in the system is also their responsibilities. Every period, cluster head must report to the Container Agent as a part of the IoT server about the result of the cluster head calculation.

- Cluster member

The cluster member is responsible for detecting their corresponding environment via IoT nodes. Every data traffic that originates from IoT nodes must be analyzed first in it.

3.5. First clustering

At first, in the simulation, each node was deployed with coordinate (x, y) in a coverage area. Then, K-Means clustering algorithm was implemented which created the cluster according to the input size. Each cluster's centroid on this process became a cluster head on the first period.

3.6. Cluster head selection

Selecting cluster head in this phase was done by considering 2 parameters, i.e., risk and energy consumption. In order to calculate the correct IoT controller for cluster head, this method uses a cost-based formula. This calculation is based on C. Fung's acquaintance management [17, 22] which is used to select a set of acquaintances from a list of candidates (A_i), hence the overall cost is minimized.

Since we did not consider the acquaintance management and assumed all of the nodes in the system are trusted nodes, the fixed cost of maintaining the acquaintance set ($M(A)$) is replaced by energy consumption of a node n (E_n). This replacement aims to select an IDS node in cluster i (C_i) that has the minimum risk and energy consumption. The mathematical expression of the proposed calculation is as (1):

$$\forall C_i | n = \{1 \dots \dots |C_i|\} = min\{R(IDS_n) + E_n\} \quad (1)$$

3.6.1. Energy modeling and calculation

In order to calculate the energy consumption, a complexity model is adjusted for each algorithm used in IoT controller and also its role in every sampling period whether it be the cluster head or member node. For our case, the complexity of an algorithm affects 2 processes in classification, i.e. training and testing phase. As stated above, this method uses 3 types of decision tree-based and a naive Bayes algorithm as shown in Table 2. The relationship between energy and complexity located on how the energy will be reduced. In this system, the energy consumption is set to follow the complexity pattern. So, if the complexity between each algorithm is different, the energy consumption will follow. But, because in the system only the complexity of each algorithm is different and since the testing process only occurs at the first deployment of IDS, the pattern will relatively the same on each algorithm. The result of this energy calculation is called as IDS process energy and denoted by e_{id}. In this model, some cost factors were used to determine the value of each process occurs on the IoT controller since each of them consumes energy. There are 4 main states on the controller, i.e., data transmitting and receiving, cluster head calculation, and the IDS process which uses the above-mentioned algorithm. Each state has its own energy consumption and is presented in Table 3.

Table 2. Selected algorithm complexity

Algorithm	Complexity	
	Training	Testing
J48 [23]	$O(m.n^2)$	$O(m)$
Random forest [23]	$O(m.n^2.p)$	$O(m)$
Naïve Bayes [24]	$O(m.n)$	$O(m)$
REPTree [23]	$O(m.n^2)$	$O(m)$

Table 3. Energy consumption ratio

State	Energy Consumption
Transmit	0.0324 J/packet [25]
Receive	0.01944 J/packet [25]
Cluster Head Process	673.4 nJ/task [26]
IDS Process	81.7 nJ/memory put [26]

Meanwhile, the energy model was denoted by:

$$E_n = e_{tt} + e_{tr} + e_{ch} + e_{id} \quad (2)$$

$$e_{tt} = C_{tt}.\sum messages_{sent} \quad (3)$$

$$e_{tr} = C_{tr}.\sum messages_{received} \quad (4)$$

$$e_{ch} = C_{ch}.\sum cluster_{member} \quad (5)$$

where:
E_n = Total energy consumption of node n
e_{tt} = Energy consumption for transmitting message in node n
C_{tt} = Cost for transmitting message in node n
e_{tr} = Energy consumption for receiving message in node n
C_{tr} = Cost for receiving message in node n
e_{ch} = Energy consumption for cluster head calculation in node n
C_{ch} = Cost for cluster head calculation in node n
e_{id} = Total energy consumption for intrusion detection process in node n

From the model above, the cluster head will consume more energy since it needs to initiate a communication among its member nodes and IoT server as well. Furthermore, the determination of the next cluster head (5) is done at the previous CH, making the energy consumption is larger than the member nodes. As a result of the additional process, the previous CH tends to choose different CH node on every consecutive sampling period.

3.6.2. Cost-loss based decision

As stated above, the addition of risk and energy consumption was used to determine the current cluster head. In this research, cost-based decision is based on the risk cost calculation in Fung's work [22]. However, since the circumstances between simulations are different, we use the Orfila's formula which is more suitable. In previous research, the parameter is called cost per unit loss [16] (M). Both Orfila's and Fung's formula are similar to some extent, that is, they decide whether and intrusion should raise and alarm or not. Orfila's risk cost calculation is expressed in (5).

$$M = \min\left\{(1-H), \frac{C}{L}(1-F)(1-p) + (1-H)p\right\} + \min\left\{Hp, \frac{C}{L}(F(1-p) + Hp)\right\} \quad (6)$$

where:
M = Expected cost per unit loss
H = Hit rate, denoted by $P(A|I)$ or True Positive rate
C = Response cost towards an intrusion
L = Damage cost caused by a missing intrusion
F = False alarm rate, denoted by $P(A|-I)$ or False Negative rate
p = Probability of an intrusion is happening

The ratio of C and L are obtained from the selection in the exploration process that has been done at first deployment. Each cluster has its own exploration results and will be selected randomly. The exploration process is done by calculating every risk value between 0 and 1 for the corresponding cluster head and choose one of them with considerations.

3.6.3. Final selection and calculation process

The final calculation for cluster head selection combines the value of expected cost per unit loss ($M(IDS_n)$) and energy consumption (E_n) on each IDS node. Nevertheless, since the value of energy consumption is still in energy units (Joule), a normalization must be done. This method uses min-max normalization that is written:

$$\widehat{E_n} = \frac{E_n - E_{min}}{E_{max} - E_{min}} \quad (7)$$

where:
E_n = Energy consumption of node n
E_{min} = Minimum energy consumption of node n
E_{max} = Maximum energy consumption of node n

This research assumed the value of E_{min} is 0, while E_{max} depends on node n initial energy. In other words, the energy consumption will not exceed the initial energy. After the normalization process, the final calculation of risk-based cluster head selection (R_{CHS}) in the system is expressed as:

$$R_{CHS} = M(IDS_n) + \widehat{E_n} \quad (8)$$

The collaboration process occurs when the intrusion detection resource was sent to the CH. It is then followed by the calculation of 2 parameters that has been received from MN in CH. The resulted lowest value can be used as consideration for selecting the next CH node. In order to measure the system performance, we use network overhead calculation to find out whether the system sends more control information packet or payload itself. The equation of this calculation is expressed as:

$$f_{nor}(t) = \frac{\sum packet_{ci}}{\sum(packet_p + packet_{ci})} \quad (9)$$

where:
$packet_{ci}$ = Number of control information packet in one period
$packet_p$ = Number of payloads in one period

4. RESULT AND DISCUSSION

The simulation designed with considering some parameters as shown in Table 4.

Table 4. Experimental parameters

Parameter	Value
IoT nodes	±75.000
Cluster amount	5
DA amount	25
DA initial energy	6 Joule
Observed period	10 rounds
Maximum coordinate (x, y)	(20,20)

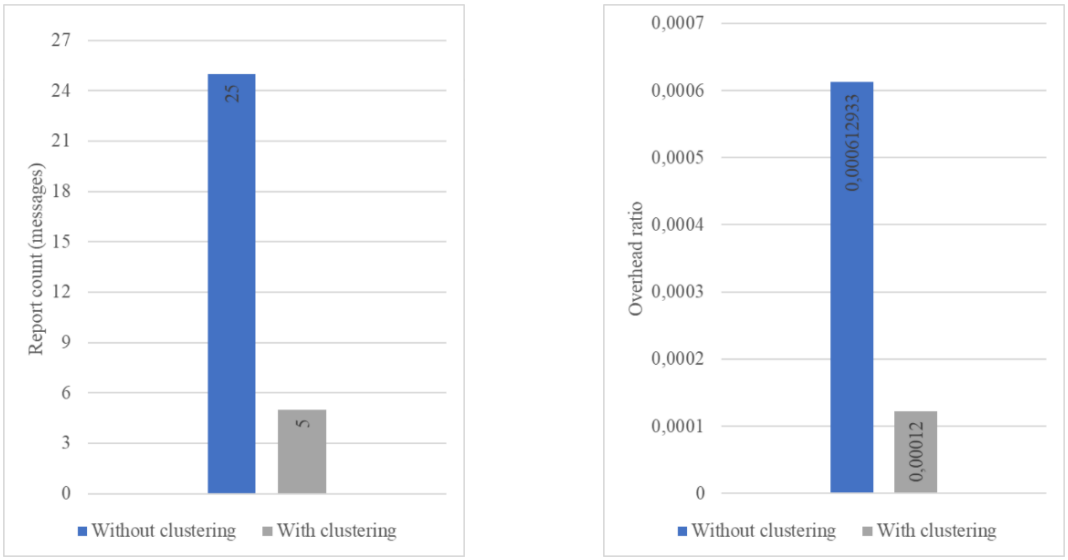
4.1. Network overhead ratio

In this performance parameter, this research tries to compare between the amount of control information and the total number of traffics sent on each DA. The aim of this performance parameter is to measure whether the system is sending more overhead or not. Higher result means that the system sending more overhead which is not good in terms of network capabilities. It is assumed that the reporting messages sent by CH to CA is packet control information. Also, on every period, the packets sent by DA to CA are only a control information without the payload. We define the network overhead ratio as the volume of control information, in this case is CIDN-related packets, sent to IoT server during one period of reporting. After calculating the ratio using (9), the result of the comparison is visualized on Figure 5(a) and Figure 5(b).

As shown in Figure 5, the proposed system is more efficient than the others. It reduces the report counts up to 80% since we use 5 clusters with total 25 DA. The reduction of report counts of DA is heavily influenced by the number of clusters created in the system. Because each of them consists of 1 CH and only CH can send packets to the IoT server, the number of reports equals the number of clusters. In contrast, for the non-clustering scenario, the amount of reporting nodes is equal to the number of Da launched by assuming the above-stated condition.

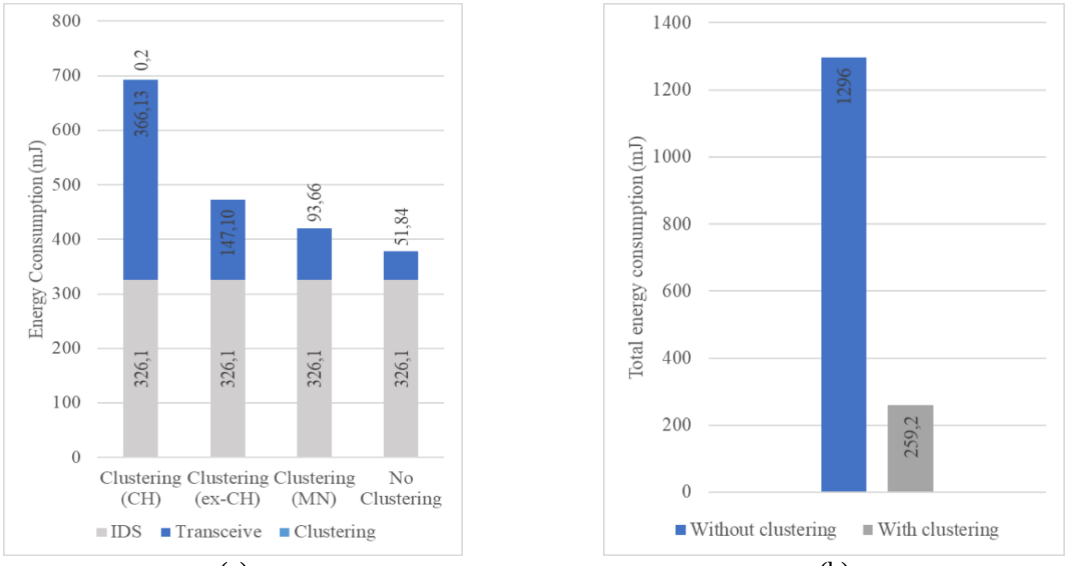
The average energy consumption comparison between the proposed system and other scenarios is also influenced by how the whole process is handled. In our system, there are 4 energy parameters included in the cluster head calculation and 3 parameters for member nodes. Meanwhile, in the testing scenario without clustering process, the CH calculation is relatively similar with member nodes, where the parameters included are e_{id}, e_{tt}, and e_{tr}. Hence, in the testing scenario, the most influential parameter towards the

energy consumption is e_{id}, because the sending and receiving report process only happen 1 time respectively. Figure 6(a) shows the comparison of energy consumption for each state in the proposed system as well as compared scenarios. Meanwhile, Figure 6(b) shows the average energy consumption of transmit and receive process per period for both systems.



(a) (b)

Figure 5. Comparison of performance parameter between 2 scenarios: (a) Report count over 2 scenarios, (b) Network overhead ratio over 2 scenarios



(a) (b)

Figure 6. Average energy consumption comparison: (a) Average energy consumption per state, (b) Average transceiving energy consumption

From Figure 6(a), it shows that in proposed system, especially in CH, the average energy consumption is higher than on the ex-CH, MN and testing scenario. It is because in CH, there is an addition of parameter e_{ch} to calculate the cluster head selection process. Also, the energy consumption differences between the proposed and other systems are significant. This is because the addition of cluster head selection process in our system, as well as numerous transmission process. On the other hand, the amount of total energy used in transmit and receive on both scenarios are far adrift. This is affected by the total number of

messages sent and received as seen in 5. Based on these results, in terms of the total energy spent on each scenario to communicate with IoT server, the proposed system obviously has better results due to the reduction on the number of reporting nodes.

4.2. Metric of IDS value

The security performance itself can be proved from the low-scored value of cost per unit loss (M) and with metric of economic value. This metric measures the value of an IDS and evaluates the efficiency of it. However, this metric is not relevant for evaluating the effectiveness of the IDS. One sample metric of IDS efficiency can be called IDS value, acting as the reference to the capability of detecting an intrusion. If an IDS is perfect at detecting intrusion, then its value is 1, otherwise it is $0 < V < 1$ based on the improvement of its predictive system. The comparative results of CHs V in all cluster are shown in Figure 7. This metric can be calculated using (4)-(6) on [16].

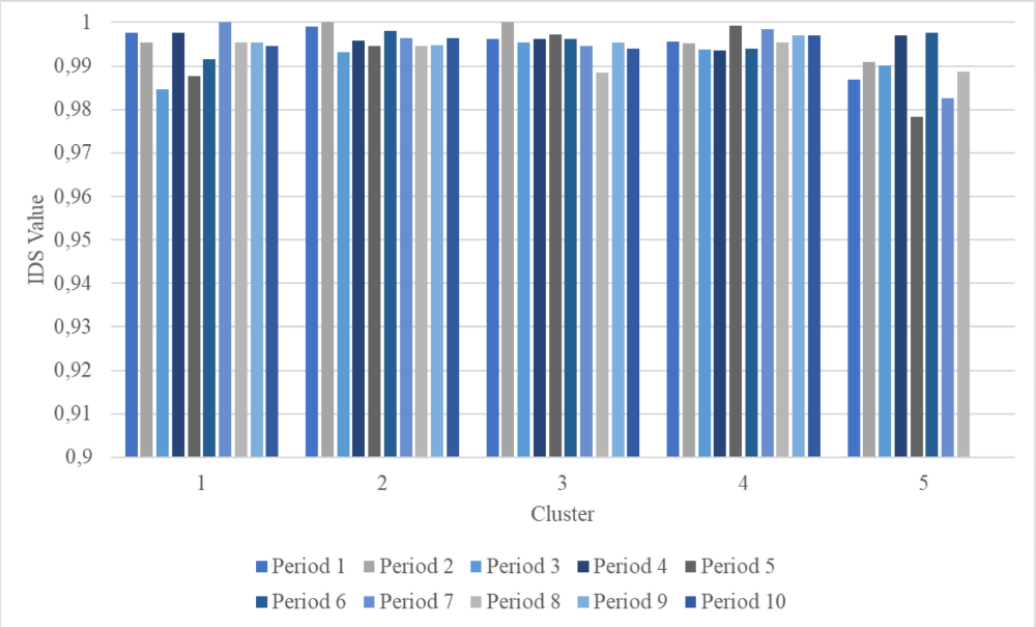


Figure 7. IDS value of selected CH per cluster

As shown in Figure 7, the x-axis shows each cluster and period while y-axis shows the value of IDS efficiency metrics (IDS value). Out of the total 5 clusters, each of them got the value nearly 1. This proves that in our system, each IDS node chosen to be CH has reliable security performance. The value of IDS efficiency is also highly influenced by the performance of individual IDS process. So, if the corresponding IDS could perform better at detecting False Positive (FP), True Positive (TP), and False Negative (FN), which are the main impact factors of the IDS efficiency calculation, then the score would be better.

5. CONCLUSION

The proposed system was proved to be able to create a collaborative intrusion detection network with multi-hop clustering. In addition, a performance evaluation was done in terms of overhead ratio as well as energy consumption. From the performance results, the reports sent by IoT controller layer to IoT server is reduced by 80%. Since it was able to reduce the number of reports to the IoT server, the total energy that were used to communicate will be also reduced by the similar ratio. The proposed cluster head selection algorithm was able to select the lowest energy consumption while maintaining security performance represented by IDS value ~1 and low expected cost per loss (M) parameter.

This method is tested in IoT network with density approximately ±75.000 IoT nodes. Varying number of nodes can affect the performance of IDS in terms of its detecting capabilities, yet in accordance with the algorithm complexity. Since in this experiment we did not measure the exact time of each period, the number of end nodes can affect its performance values.

REFERENCES

[1] A. Ghasempour, "Internet of things in smart grid: Architecture, applications, services, key technologies, and challenges," *Inventions*, vol. 4, no. 1, 2019, Art. no. 22, doi: 10.3390/inventions4010022.

[2] K. Ashton, "That Internet of Things Thing," *RFID journal*, vol. 22, no. 7, pp. 97–114, 2009.

[3] M. Weiser, "The computer for the 21st century," *ACM SIGMOBILE Mob. Comput. Commun. Rev.*, 1999, doi: 10.1145/329124.329126.

[4] P. Sethi and S. R. Sarangi, "Internet of Things: Architectures, Protocols, and Applications," *Journal of Electrical and Computer Engineering*, vol. 2017, pp. 1-25, 2017, doi: 10.1155/2017/9324035.

[5] X. F. Wang, "Research on Security Issues of the Internet of Things," *4th Int. Conf. Mater. Sci. Inf. Technol. MSIT 2014 989-994*, vol. 994, pp. 4261–4264, 2014, doi: 10.4028/www.scientific.net/AMR.989-994.4261.

[6] E. Vasilomanolakis, S. Karuppayah, M. A. X. M. Uhlh, and M. Fischer, "Taxonomy and Survey of Collaborative Intrusion Detection," vol. 47, no. 4, pp. 1–33, 2015.

[7] Z. A. Khan and P. Herrmann, "A Trust Based Distributed Intrusion Detection Mechanism for Internet of Things," *2017 IEEE 31st Int. Conf. Adv. Inf. Netw. Appl.*, 2017, pp. 1169–1176, doi: 10.1109/AINA.2017.161.

[8] M. Shingte, F. Siddiqui, and A. Yewale, "Review Of Distributed Intrusion," *Int. J. Res. Eng. Appl. Manag.*, vol. 1, no. 11, pp. 1-7, 2016.

[9] C. Fung, "Collaborative intrusion detection networks and insider attacks," *J. Wirel. Mob. Networks, Ubiquitous Comput. Dependable Appl.*, vol. 2, no. 1, pp. 63–74, 2011.

[10] E. Vasilomanolakis, "On Collaborative Intrusion Detection," Technische Universität Darmstadt, 2016.

[11] E. Vasilomanolakis, S. M. Habib, P. Milaszewicz, R. S. Malik, and M. Mühlhäuser, "Towards trust-aware collaborative intrusion detection: Challenges and solutions," in *IFIP Advances in Information and Communication Technology*, 2017, pp. 94–109, doi: 10.1007/978-3-319-59171-1_8.

[12] C. J. Fung, Q. Zhu, R. Boutaba, and T. Başar, "Bayesian decision aggregation in collaborative intrusion detection networks," in *Proceedings of the 2010 IEEE/IFIP Network Operations and Management Symposium, NOMS 2010*, 2010, pp. 349–356, doi: 10.1109/NOMS.2010.5488489.

[13] W. Li, W. Meng, L. F. Kwok, and H. H. S. IP, "Enhancing collaborative intrusion detection networks against insider attacks using supervised intrusion sensitivity-based trust management model," *J. Netw. Comput. Appl.*, vol. 77, pp. 135–145, 2017, doi: 10.1016/j.jnca.2016.09.014.

[14] C. Gruhl, F. Beer, H. Heck, B. Sick, U. Buehler, A. Wacker, and S. Tomforde, "A Concept for Intelligent Collaborative Network Intrusion Detection," *ARCS 2017; 30th International Conference on Architecture of Computing Systems*, Vienna, Austria, 2017, pp. 1-8.

[15] J. Arshad, M. A. Azad, M. M. Abdellatif, M. H. Ur Rehman, and K. Salah, "COLIDE: A collaborative intrusion detection framework for Internet of Things," *IET Networks*, vol. 8, no. 1, pp. 3–14, 2018. doi: 10.1049/iet-net.2018.5036.

[16] A. Orfila, J. Carbó, and A. Ribagorda, "Autonomous decision on intrusion detection with trained BDI agents," *Comput. Commun.*, vol. 31, no. 9, pp. 1803–1813, 2008, doi: 10.1016/j.comcom.2007.11.018.

[17] C. Fung, "Design and Management of Collaborative Intrusion Detection Networks by," in *2013IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*, 2013, pp. 955–961.

[18] V. Gupta, S. K. Devar, N. H. Kumar, and K. P. Bagadi, "Modelling of IoT Traffic and Its Impact on LoRaWAN," in *2017 IEEE Global Communications Conference, GLOBECOM 2017-Proceedings*, 2017, pp. 1-6, doi: 10.1109/GLOCOM.2017.8254512.

[19] T. Hosfeld, F. Metzger, and P. E. Heegaard, "Traffic modeling for aggregated periodic IoT data," in *21st Conference on Innovation in Clouds, Internet and Networks, ICIN 2018*, 2018, doi: 10.1109/ICIN.2018.8401624.

[20] F. Metzger, T. Hobfeld, A. Bauer, S. Kounev, and P. E. Heegaard, "Modeling of Aggregated IoT Traffic and Its Application to an IoT Cloud," *Proceedings of the IEEE*. vol. 107, no. 4, pp. 679–694, 2019. doi: 10.1109/JPROC.2019.2901578.

[21] F. Bellifemine, A. Poggi, and G. Rimassa, "JADE a FIPA2000 compliant agent development environment," in *Proceedings of the International Conference on Autonomous Agents*, 2001, pp. 216–217.

[22] C. Fung and R. Boutaba, "Intrusion Detection Networks : A Key to Collaborative Security," *CRC Press*, 2013.

[23] J. Su and H. Zhang, "A Fast Decision Tree Learning Algorithm," *American Association for Artificial Intelligence*, vol. 6, 2006, pp. 500–505.

[24] G. H. John and P. Langley, "Estimating Continuous Distributions in Bayesian Classifiers," in *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 1995, pp. 338–345.

[25] M. Costa, T. Farrell, and L. Doyle, "On Energy Efficiency and Lifetime in Low Power Wide Area Network for The Internet of Things," in *2017 IEEE Conference on Standards for Communications and Networking (CSCN)*, 2017, pp. 258–263, doi: 10.1109/CSCN.2017.8088631/.

[26] T. K. Tan, "Energy Macromodeling of Embedded Operating Systems," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 4, no. 1, pp. 231–254, 2005.

BIOGRAPHIES OF AUTHORS

Ida Wahidah received the B.Eng., M.Eng., and D.Eng. degrees in electronics and informatics engineering from Institut Teknologi Bandung, Indonesia in 1998, 2005, and 2014 respectively. She is currently a Senior Lecturer with the School of Electrical Engineering, Telkom University, Indonesia. She has authored and coauthored more than 50 research papers in national and international journals and conferences. Her research interests include compressive sensing theory and applications, body sensor networks, public protection and disaster relief, as well as video watermarking. Her research has been supported by the Indonesian Higher Education Ministry and Korean security industry. She is a member of the IEEE and IEICE society.

Yudha Purwanto is a lecturer and researcher in Electrical Engineering Department of Telkom University. He received his Doctoral degree in School of Electrical Engineering from Institut Teknologi Bandung with specialized research on Network Queue in 2019. His research interest is mainly about Network Security.

Aditya Kurniawan is post-graduate student from Electrical Engineering Department of Telkom University. He received his B.Eng. from Telkom University in 2018. He recently doing research about Network Security and his main research interest is Internet of Things.