Indonesian Journal of Electrical Engineering and Informatics (IJEEI)
Vol. 12, No. 3, September 2024, pp. 711~725
ISSN: 2089-3272, DOI: 10.52549/ijeei.v12i3.5533

ML-ACID: a Modified Machine Learning Algorithm Coupled With a Novel Ant Colony Approach for Intrusion Detection in IOT

Hamza Belkhiri[1], Abderraouf Messai[1], Yehya Belhadad[2], Beylot Andre-Luc[3], Salaheddine Sadouni[1]
[1]LSIACIO Laboratory, Constantine 1 Frères Mentouri University Constantine, Algeria.
[2]Telecommunications ETA, Department of electronics, University Mohamed El Bachir El Ibrahimi of Bordj Bou Arreridj, Algeria.
[3]IRIT/ENSEEIHT The National Institute of Electrical engineering, Electronics, Computer science Fluid mechanics and Telecommunications and Networks, National Polytechnic Institute, Toulouse, France.
[1]LSIACIO Laboratory, Constantine 1 Frères Mentouri University Constantine, Algeria.

Article Info

Article history:

Received Apr 7, 2024
Revised Aug 31, 2024
Accepted Sep 25, 2024

Keyword:

Ant
ACO
Machine learning
Intrusion detection. SDN
OpenFlow

ABSTRACT

Software Defined Networks is becoming increasingly important in IoT because it allows devices to communicate more easily it provides the flexibility and centralized management, however in recent years these networks have witnessed a widespread spread of cyber-attacks that has a significant and negative impact on the availability of services. In this paper, we propose a novel approach for intrusion detection in Software Defined Networks for IoT. our work inspired by the self-defense mechanism of ant colonies. The approach uses a self-adaptable colony fingerprint and based on multiple parameters, it makes the detection of intrusions easy and filters out every other legitimate communication within the network. A machine learning model is used to provide basic predictions about the communication that later drives the evolution of the colony in terms of self-defence. The whole approach is implemented in a simple switch using Ryu-controller and analyses OpenFlow datagrams. The meta-heuristic implication of using ant colony optimization improved approach provides the system with reliability and high performance of detecting and blocking threats. in the end interesting results based on several scenarios shows the usability of our approach.

Corresponding Author:

Hamza Belkhiri,
Department of Electronics,
University Frères Mentouri Constantine 1,
Constantine, Algeria.
Email: hamza.bekhiri@umc.edu.dz

1. INTRODUCTION

The emerging use of the internet and the proliferation of network connected devices from the constant use of smart-phones, laptops all the way to the use of smart home devices and IoT. The importance of these is in constant increase in our daily lives, however, both the constant evolution and the increase use of such connected devices bring with it its own challenges.

The company Cisco [1], predicted that in 2021 there would be around 27.1 billion network devices worldwide, now that number of connected IoT devices is 13.1 billion globally [2]. Considering that the IoT is a incentive for many areas of application, such as, smart homes, smart farms, healthcare, etc. and each of which has diverse requirements, and therefore there must be a technology capable of meeting the heterogeneous requirements, for that we find technology Software Defined Networking (SDN) is essential for the development of IoT because it provides the flexibility and centralized management that is necessary for a network.

SDN as a new approach or technique for managing computer networks, a new paradigm shifts in network architectures [3]. such use opens up new possibilities towards both monitoring and control over network activity in an advanced manner.

Despite the tremendous advantages that software-defined networking for IoT but they have collided with one of the most important security and privacy challenges. especially in light of the diversity of different attacks such as; Distributed Denial of Service (DoS-DDoS) attacks This may not only affect usability, but also affect users' lives [4].

In this work, we rely on combining the use of a novel meta-heuristic based on Ant Colony behaviour in real life, and machine learning to achieve a solution to detect both DoS-DDoS attacks within a software defined network. Some existing works try to combine Ant Colony Optimization and machine learning [5],[6] however, in our approach we try to mimic the colony self-defence mechanism in recognizing and neutralizing intrusions, it can also be extended to other types of attacks.

Ants in real life belong to only one colony of ants, by time the ants in a colony tend to recognize each other using smell. The smell in the form of pheromone fingerprint that is unique to each colony [7],[8]. What happens is whenever an ant does not belong to the colony the ants within the colony would recognize the intrusion and neutralize it by using the smell. Another infraction that might be deemed wrong but is useful in the case of identity theft in networking happens within the ant colony is whenever a member ant
traverses the path of another colony it acquires the second colony fingerprint and loses its original smell. In this case also the ant is treated as intrusion.

Our work inspired from the colony self-defence mechanism of ant colonies and implements a solution seamless and advanced intrusion detection within a software defined network using a similar approach. The implemented approach relies on an auto adaptable pheromone fingerprint that helps identify both legitimate and malicious traffic within the network. The results show the efficacy of the approach in detecting and the auto-evolution of the network controller. Moreover, the approach is not only limited to these types of attacks and can be extended to other types.

Also, the inclusion of a machine learning model that drives the evolution of the mechanism and the colony auto-adaptable fingerprint makes it easy to assist the colony in detecting several types of intrusion. This paper is organized in four main sections, the first section presents related works that treats the same problem. The second section presents the background to our approach based on the behaviour of real ants then a system design is proposed to tackle the problem. Finally, system deployment is presented along with test based on scenarios formulated for our specific problem.

2. RELATED WORKS

Most works related to the use of Ant Colony inspired algorithms they apply the algorithms on feature selection to optimize the results of machine learning. In most of the works related to the use of Aco algorithms in the field of intrusion detection in network security, they try to apply the algorithm as a first stage to the machine/deep learning system so to help select the optimal features to optimize the controller ability to detect intrusion in the case of DDoS and DoS.

J. Ramprasath and V. Seethalakshmi in [9], The DDoS attacks were handled in an SDN environment where ant colony optimization particle swarm optimization (ACO-PSO) was used to frame the traffic state and then identify the traffic in normal traffic and abnormal traffic. Mitigation is applied if illegal traffic is identified, otherwise packets will be forwarded.

In [10] Liang et al. they proposed a network traffic prediction model based on Support Vector Regression (SVR), that through develop a detection method by ant colony optimization (ACO) and chaos theory. Parameters of the machine learning model were selected based on the use of the Ant Colony Optimization (ACO) algorithm. The experimental results show that the proposed method has a high accuracy rate.

In another work, S. Jaiswal [11], the intrusion detection system supervises is applied to expose the intruders, they focus on use K-Nearest Neighbour KNN classifier with ACO (ant colony optimization) were applied to detect on Normal and malicious traffic. In their work, it is applied on multiple classifiers, as the misclassified features can be combined and used for improving the time, while the ACO optimizes the category until the features which take less time to compute are accurately classified.

HH Chen and SK Huang. In [12], a technique is proposed to counter low-rate distributed denial-of-service (LDDoS) attacks by an ant-colony-optimization based meta-heuristic technique. The core idea in this work, and due to compatibility with the emerging Software Defined Network (SDN) is to applied multi-agent algorithm the control plane and the data plane to monitor and manage the network topology. When comparing distributed detection and identification ant colony system (DDIACS) framework with existing methods, we see that the framework is capable of addressing LDDoS with a detection rate of about 89 percent.

Aanshi Bhardwaj et al. In [13] an approach has been proposed to detect DDoS attacks in the cloud environment. The authors focused on using the deep neural network (DNN) model, with they were based on the use of ant colony optimization (ACO) to obtain the effective classification of DNN. The experimental results showed that after conducting experiments on the CICIDS2017 data set, the method has high detection and accuracy rates of 95.74 percent and 98.25 percent, respectively.

D. Yuvaraj et al. In [14] in another work, DDoS attacks have been dealt by nature inspired evolutionary Algorithm (ACO). In this work, using the input of essential parameters (time and conduction) the optimization algorithm is trained, when compared with the current Artificial Neural Network (ANN) algorithm, the experimental results showed superior performance of proposed ACO algorithm.

As DDoS attack techniques evolve, networks defence has become very difficult because of conventional detection methods. For this, integrate machine learning with SDN can lead to a self-network capable of dealing with these attacks. In recent years, many studies have been done to secure SDN using machine-learning techniques. In this section, a simple survey of the machine learning techniques used in dealing DDoS attacks is presented.

A. S. Jose, L et al. [15], system is designed to detect floods attack, (TCP SYN flooding attacks, HTTP request flooding attacks, UDP flooding attacks and ICMP flooding attacks) in SDN network. They mentioned that after using the feature selection module with the statistical method ANOVA (Analysis of Variance) F-Test, the feature sets for classification were arrived. After they performed a performance evaluation for each of the classifiers that used the obtained features, the feature set which gives the best performance in detection was determined.

Ö. Tonkal, H et al [16] Machine learning algorithms were used to classify normal or offensive data traffic in SDN by constructing the dataset. In their work, they report that the data set was obtained after they drew on the NCA algorithm to detect the most relevant features by selecting the feature, then the dataset is classified by; k-Nearest Neighbor (kNN), Decision Tree (DT), Artificial Neural Network (ANN), and SVM algorithms. The experimental results show that the proposed method has a high accuracy rate.

Aye, Thandar, Kyaw et al. [17] used two machine-learning algorithms to detect DDoS attacks in the SDN. After generating traffic packets using the Scapy tool, a polynomial SVM is applied to compare to existing linear SVM. After the feature extraction phase, Classification performance is compared for linear and polynomial SVM models. Experimental results show that polynomial SVM algorithm achieves better accuracy compared to linear SVM by 3 percent.

Huseyin Polat et al, [18] proposed a model using machine learning-based models to detect DDoS attacks. In this work, the two datasets are created with and without feature selection methods. After training the two datasets–they are tested using Support Vector Machine (SVM), Naive Bayes (NB), Artificial Neural Network (ANN), and K-Nearest Neighbors Classification (KNN).

Experimental results show a High accuracy rate of DDoS attack detection was achieved by using the wrapper feature selection with a KNN classifier (98.3%).

Ranyelson N. Carvalho et al. [19] The DataPlane-M approach is proposed for detecting DDoS attacks in an SDN environment. DataPlane-M uses white box switches to handle input flow and ML libraries to run ML models at the data plane. When evaluating the proposed work using KNN, SVM and RF machine learning algorithms, it was compared with statistical-based solutions.

The experimental results show that DataPlane-ML is = 23 percent faster than statistical-based solutions while providing better accuracy.

K. S. Sahoo et al in [20] The SDN centralized control aspect is exploited to detect malicious attack traffic. The proposed work aims to design an SVM model that achieves a more accurate classification with the possibility of embedding the form inside control plane. This work is based on kernel principal component analysis (KPCA) using Genetic Algorithm (GA) to assist the Support Vector Machine (SVM). Genetic algorithm (GA) and (KPCA) contribute to the optimizing different SVM parameters and feature differences.

After comparing the proposed model with the single-SVM model, the experimental results showed that the proposed model achieves a more accurate classification.

W. Zhijun, X et al [21] a mechanism has been proposed to Low-Rate DDoS Attack Detection Based on Factorization Machine in Software Defined Network, the main idea of the paper is to establish a feature dataset, after extracting the four features related to the flow rules, and based on FM machine learning algorithms based low-rate DDoS attacks detection is implemented. Experimental results showed. After comparing the proposed work with the detection methods that detect such attacks, the mechanism has a high accuracy rate and AUC value with an effective line of defence.

J. A. Perez-Diaz et al. [22] LR-DDoS attacks were dealt with in an SDN environment, by designing and implementing a modular and flexible security architecture. The authors stated that the intrusion detection system (IDS) designed into the architecture underwent training using machine learning (ML) models (Support Vector Machines , Random Forest, REP Tree, J48, MultiLayer Perceptron, and Random Tree when

evaluating their performance using the Canadian Institute of Cyber-security (CIC) DoS dataset. The results of the evaluation gave that the proposed approach achieves a detection rate of 95 percent, with the ability to mitigate attacks.

L. Tan, Y. et al [23] A mechanism was used to detect and defend DDoS attacks in the SDN environment, the proposed framework is based on studying asymmetry characteristics of the flows and the suspicious flows, they mentioned first; that at the data plane a DDoS attack was launched and is implemented by counting packets in messages on switches. Then the features that can characterize the attack flows more accurately are selected.

Finally, the K-Means algorithm is applied in the training phase, and the KNN algorithm is used to detect after comparing the distance between the detection point and various clustering centers. We use a combined machine learning algorithm based on K-Means and KNN. The experimental results showed that using a combined machine-learning algorithm based on K-Means and (KNN) has the ability to know the asymmetry of the flows and the suspicious flows well. In addition, the proposed framework provides an effective defence and mitigation strategy for DDoS attacks.

K. M. Sudar et al. In [24] a machine learning algorithm (Decision tree support vector machine (SVM) was used to counteract DDoS attacks. In this paper, the KDD99 data set was divided into two groups, the training data set and the evaluation data set. After training and conducting an evaluation of the proposed model. The experimental results showed the effectiveness of the proposed approach with a better accuracy of SVM compared to the decision tree technique.

3. BACKGROUND

The concept of our approach is based on the behaviour of ants in real life. Pheromone is a substance that drive the behaviour of several animals and insects, the ants in this case uses pheromone for multiple tasks. Some tasks are related to the path that every ant browse, other tasks based on pheromone are for defence and rank recognition. Ants in real life recognize each other based on pheromone fingerprint. Pheromone represents for ants the smell for humans. Like searching for an optimum path to find the food [25], ant colonies in real life have an internal defence mechanism to detect intruder ants based also on pheromone that ants can smell. Instead of the pheromone being deposited, in this case to recognize intruders or foreign ants. each ant develops a distinct pheromone odour related to the colony from the constant contact with other colony members.

The smell and odour of each ant is sustained [26] and developed overtime; this helps the colony to keep track of the colony members without the need for fancy mechanism, some other animals make use of this recognition pattern. Such a distinct pheromone odour acts like a fingerprint, each colony develops over time a distinct fingerprint that each ant member of the colony recognizes and acts upon. The fingerprint helps not only recognize same colony members, also ranking within the colony is recognized based on pheromone fingerprint. In our approach, the pheromone fingerprint acts as a communication link between hosts that helps us decide along with the colony fingerprint, i.e. in the normal case of no attack occurs, it receives positive feedback from the machine learning model, then it contributes with the colony fingerprint to allows ants to communicate and vice versa in the case of an attack. The following sections explain the basic approach that inspired our system.

3.1. Pheromone fingerprint evolution

The pheromone fingerprint of a colony starts to develop over time from the constant contact of the colony member ants. Whenever an ant crosses the path of the colony, it does not only deposits pheromones trace itself but also contracts some of the existing pheromone on its body. At the same time, whenever a close contact occurs with other ants, both ants get and deposits some pheromones on each other's. All this results in the auto-generation of a distinct pheromone fingerprint to the colony member ants which will develop over time.

3.2. Pheromone fingerprint loss or staining

A distinct property of pheromone as chemical substance, is its evaporation over time. Each ant would lose pheromone odour over time but in small quantities. Therefore, intruders will be more likely detected by the colony members and colony members would recognize each other without difficulty. Another property of the pheromone fingerprint is that not only do the ants get pheromone staining from the colony itself but whenever an ant crosses the path used by another colony it might get stained by their odour. This staining can cause in some (rare) cases to the ant losing its colony fingerprint or worse gaining another colony fingerprint, which leads to the execution of the ant or deportation from the colony until it loses the other colony fingerprint by evaporation.

3.3. Intruders detection mechanism

As explained in the previous two sections, there are two scenarios when an ant is judged as an intruder/Possible intruder. The first case, the ant is a real intruder that is member of an offensive colony therefore the fingerprint is completely different and the ant is treated as an intruder. The second case is when an ant is a member of the colony [27]. However, this time if the ant might have followed a path stained by another colony's pheromone (or smell) the ant is stained and no longer has the distinct fingerprint of its original colony. Therefore, the ant is treated as an intruder.

3.4. The inspiration

In today's network security, most attacks happen usually by an attacker taking control of some of the networks legitimate hosts infecting them with a virus. In the case where a host is outside of a network, especially in the terms of IoT, an external host can be identified in some cases and neutralized. In other cases, some scenarios might allow us to simply block all incoming traffic. However, the tricky part is when an attack is coming from a friendly host whether part or not of the network.

To address this, we modelled the ant's behaviour in detecting the intruder. A host is treated as a legitimate host in the case where the fingerprint of its colony applies to its behaviour in communicating with other hosts.

The fingerprint of the colony develops over time and only in case an attack is detected, the fingerprint is increased to make the host with an unwanted behaviour to be blocked by a controller. To achieve this, the proposed algorithm is augmented with a general machine learning model that detects anomalies within the network traffic. Thus, providing intel about the traffic and about the network hosts. A filter by the algorithm can be used in order not to treat the external hosts by the colony to limit the number of hosts treated.

Each host is represented by an ant and each ant has a special variable called trust that helps us decide along with the colony fingerprint if we should trust the host or not.

The edges between each ant represent links of communications within the network. Whenever a message occurs within the network, a machine learning model informs the colony about its prediction and the colony acts accordingly to update both pheromone fingerprint of the colony and the link between the hosts communicating. The trust of each ant is updated dynamically, each occurring communication gets screened for possible intruders by examining the auto-adaptable fingerprint.

The convergence and the important feature of the algorithm is observed whenever an ant (host) is detected as non-legitimate, the colony detects further on every communication coming from the host as non-legitimate. Only when enough positive feedback from the machine learning model that helps to reset the trust and pheromone of the communications on the link between an intruder and another host, then the colony resets its fingerprint and allows the ants to communicate again. However, it takes a bit time for the colony's fingerprint to reset and allow the ants (hosts) to communication again.

4. SYSTEM DESIGN

In our approach, instead of choosing an optimum path as in the famous ACO algorithm. In our system, the ants get to choose whether to trust or not another ant as explained above. However, to achieve a meaningful decision, we augmented the system with a machine learning module. The ML module provides the ants with intel about the traffic which allows the convergence of the algorithm overtime to filter out intruders automatically.

4.1. Machine Learning (ML module)

The machine learning module is based on a dataset collected using Mininet and python scripts to generate both a set of a legitimate traffic and a set of DDoS traffic. Both the dataset and the machine learning module can be perfected to enhance the behaviour of the system, the decision tree algorithm is used to decide whether a traffic is legitimate or not. The decision tree algorithm from the python SciKit library [28] is called within the controller whenever a new OpenFlow traffic arrived.

4.2. Ant Colony Intrusion Detection (ACID module)

The ACID (Ant Colony Intrusion Detection) module is based on the following class diagram. A colony is a set of ants where each couple of ants is linked with a communication link; each ant represents a network host. The communication links forms a network, a network is a complete graph. The graph contains the set of all the links in the network.

The controller receives a message and sends information to the colony to create a new link if it does not exist. The colony searches if the ant representing a host inside the message received exists and the creates a new link with all the other existing ants.

The following class diagram fig:1 presents the main components of the ACID module. The colony is a set of ants and a set of links, an ant represents a single host. A link represents a possible communication link between two ants, at first, each new ant arriving into the colony establishes a new link with all the colony members. The links contains the needed information about the communication and is update each time a new message (on the link) arrives. The colony has the following parameters Alpha, Beta, Evaporation.

4.2.1. Pheromone update

The pheromone helps us with making an auto adaptable variable to help decide to whether trust or not a host. The pheromone itself characterizes a communication link instead of the ant itself. However, it also affects a single variable

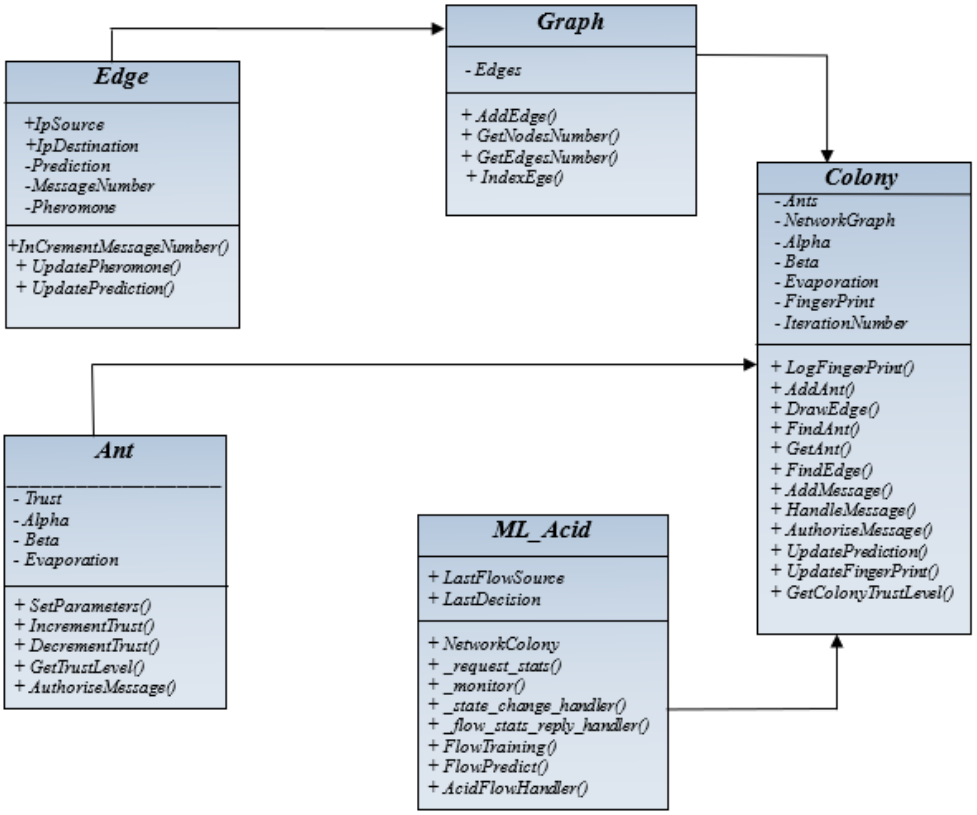

Figure 1. class diagram.

within each ant that characterizes its trust level in the colony, on each communication the trust is updated based on the prediction from the trained machine learning model to trust or not the host. The pheromone update rule is based on the following formula.

Formula for pheromone update when Trust is ≥ 0,5

$$\Phi = \Phi \times \left(\frac{(1-\varepsilon)}{\mathrm{Md}}\right) + \left(\frac{\mathrm{Trust}}{\mathrm{Md}}\right) \quad (1)$$

Formula for pheromone update when Trust is ≤ 0,5

$$\Phi = \Phi \times \left(\frac{(1-\varepsilon)}{\mathrm{Md}}\right) - \left(\frac{\mathrm{Trust}}{\mathrm{Md}}\right) \quad (2)$$

Equations (eq:1,eq:2) represent Formulas for pheromone update where:

. ε: represents the Evaporation rate.

. Md: Message Difference |incoming - outgoing| between the number of incoming and outgoing flows on a particular edge of the network.

Both the trust leveller and the pheromone on each communication link helps in the final decision. In the case where the ant has a low trust level and the communication on that link used to be a legitimate traffic, even though the decision is affected but the overall link pheromone also affects the decision outcome.

The trust is updated based on the nature of the traffic detected by the machine learning module as presented in the following equations (eq:3, eq:4). The maximum value of the trust of a single ant is 100 while the minimum value is -100.

- when trust < 100 $\text{Trust}=\text{trust}+0{,}001$ (3)
- when trust > -100 $\text{Trust}=\text{trust}-0{,}001$ (4)

The trust level τ represents the rate of trust of a single ant, and is based on ant trust value as follows (eq:5).

$$\tau=\frac{\frac{(100+\text{Trust})}{2}}{100} \quad (5)$$

Eq:5 formula for the trust level update rules.

4.2.2. Pheromone evaporation

The colony at its initialization get the pheromone evaporation rate as an input, the evaporation rate is used in the previous update formula (eq:1, eq:2) to allow as in real life ants the pheromone is lost overtime, the evaporation happens at the same time the update occurs. The evaporation rate is calculated and subtracted from the overall pheromone whether from the edge or the ant itself.

4.2.3. Fingerprint update

At every message exchanged on the network, the colony fingerprint is updated based on every single decision made by the ants to trust or not another one.

The update occurs in the following manner:

. At the initial phase of the colony, the fingerprint is set to 0 meaning: trust all

. Whenever a new traffic arrives, the colony gets the UpdatePrediction function is called from within the controller and is supplied with an intel about the traffic from the trained machine learning model.

. Based on the prediction about the traffic if legitimate or not: the trust of the source ant is either incremented or decremented, the pheromone on the link is updated and the fingerprint is updated according to the following formulas (eq:6, eq:7).

Where:

. π: represents the FingerPrint. **. ε:** represents the Evaporation rate.

. Mn: represents the Message Number. . **It:** represents Iteration Number.

$$\pi=\frac{\pi\times(1-\varepsilon)}{\text{It}-\text{Mn}+1}+\frac{\text{CTL}}{\text{It}-\text{Mn}+1} \quad (6)$$

Eq:6 Fingerprint Increment equation

$$\pi=\frac{\pi\times(1-\varepsilon)}{\text{It}-\text{Mn}+1} \quad (7)$$

Eq:7 Fingerprint Decrement equation

Whereas: CTL represents the colony trust level and is calculated based on the trust level of all ants and iteration number, on the following formula (eq:8).

$$\text{CTL}=\sum_{i=0}^{n}\tau i/\text{It} \quad (8)$$

To minimize the evaporation rate of the fingerprint, the rate is divided by the difference between the total iteration number and the current link message number. Otherwise, the fingerprint will tend to evaporate faster and always allow all kind of traffic to pass through. A colony trust level can be calculated as the average of all its ants' members individual trust level.

4.2.4. Fingerprint test

The decision to trust or not the host is only augmented by the machine learning model, instead of letting the machine learning model decide, in our approach the ants make the decision to trust or not another

one based on the following formula. The formula helps us achieve a convergence between the pheromone and the fingerprint as presented in the following equation.

$$\delta=(\alpha \times \Phi) - \left(\beta \times \left(\frac{\tau}{\mathrm{Md}}\right)\right) \quad (9)$$

Eq 9. formula for the decision.

Whereas:

. Md= Message Difference |incoming -outgoing| between the number of incoming and outgoing flows on a particular edge of the network.

. Φ= pheromone **. τ** = Trust Level

The ant representing the destination host of each traffic calculate a decision parameter following the previous formulae based on the source host trust level, the pheromone on the link between the two ants, and the last prediction on the link provided by the machine learning model. The decision parameter represents the pheromone fingerprint sensed by an ant on each encounter with another ant. The parameter is compared against the fingerprint of the whole colony which allows us to achieve an inversed convergence. The more the trust the higher the parameter is and vice versa, the colony fingerprint allows us to draw a line between which communication an ant trust or DDOS traffic figure.

4.3. ML-ACID design

The ML-ACID algorithm (short for machine learning ant colony intrusion detection) we propose follows the next diagram. The system is based on a software defined network, a controller is implemented so it takes advantage of our novel ant colony inspired algorithm and a machine learning model that aims to detect the intrusion and update the colony with intel about every communication occurring within the network. The following flowchart details the process of receiving a new flow inside a software defined network and processing it through machine learning (ML) and Ant Colony Intrusion Detection (ACID).

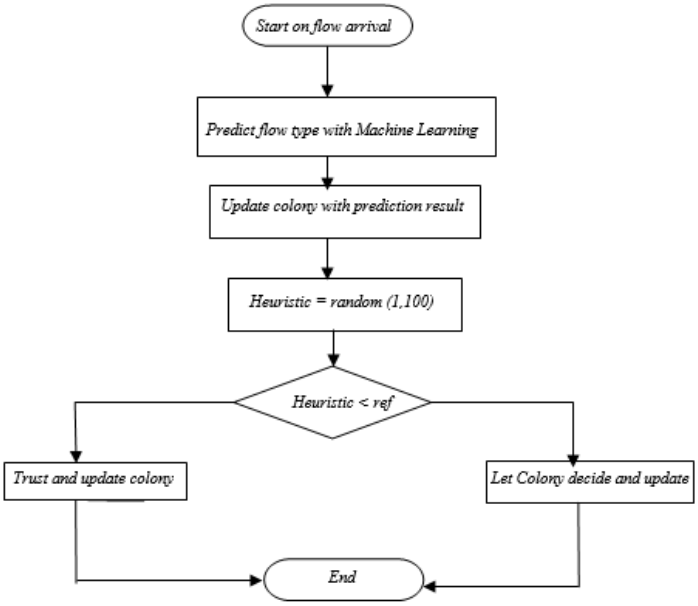

Figure. 2 ML-ACID main flowchart.

A controller (using Ryu-controller) is implemented with double monitoring functions. The first treats each flow of an OpenFlow and pass it to a pretrained model to predict if it is about legitimate traffic or not and then passes the intel into the colony, while the later analyses each traffic instance against the colony pheromone fingerprint so it decides to authorize or block the source host of the traffic communication.

5. SYSTEM DEPLOYMENT AND EVALUATION

The system deployed in two remote virtual servers, one for running our Ryu-Controller running the ML-ACID algorithm.

The second server responsible for the simulation of a network using Mininet, this server connects to our remote controller and we run the simulation of traffic using several HPING and PING commands to simulate possible DDoS and DoS attacks.

Based on the following section, a set of tests is designed to make a comparison between scenarios.

The system is deployed in on two remote servers according to the following schema. A network links the two servers so all communications occur over the network, the first server takes the task of simulating the network to be attacked using Mininet. While the second represents software defined switch controller based on Ryu-controller library.

To test the system, the servers are controlled through SSH communication, the same system can be implemented on local but the implementation done on servers provide better performance while testing the system. Another benefit of using remote server is the simulated traffic takes long to execute, before any of the following scenarios, the network is reset along with the controller to provide a better evaluation of the system.

5.1. First scenario

In this scenario we create a single legitimate traffic in a newly launched ryu- controller switch running based on a single Ping of a 100000 packets.

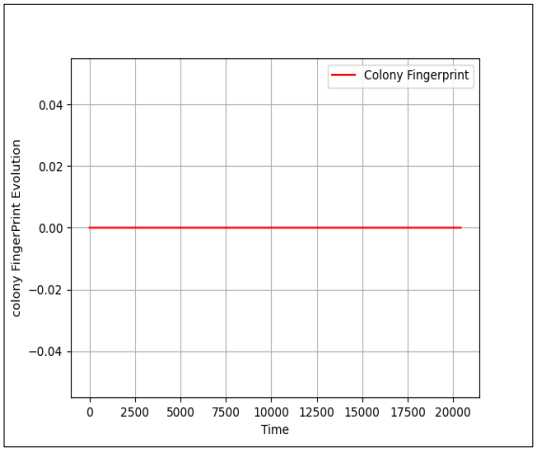

Figure 3. colony FingerPrint evolution as a function of the time.

Figure 3 illustrates the fingerprint evolution as a function of the time, we find the fingerprint remains constant over time and this due to legitimate traffic.

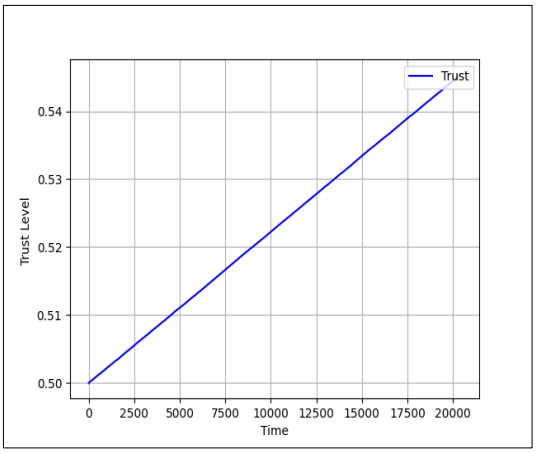

Figure 4. host 1 trust to host 2

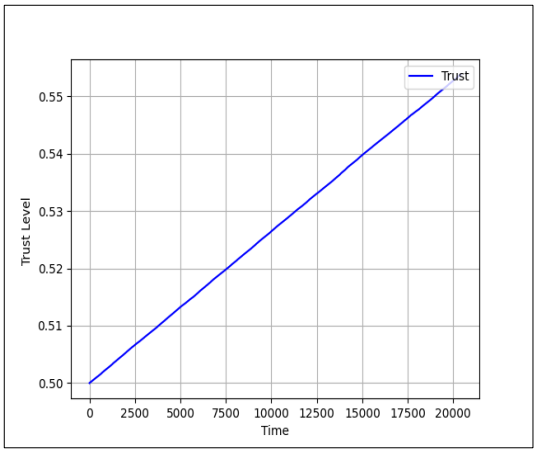

Figure 5. host 2 trust to host 1

In figure (fig:4 ,5), both graphs show the expected behaviour, where we see evolution in trust, and this is due the traffic consist only of legitimate traffic.

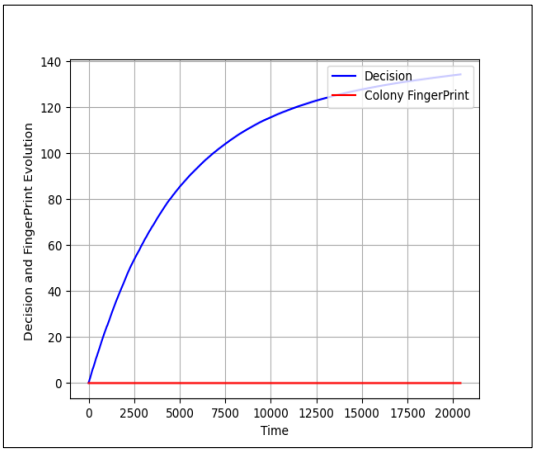

Figure 6. Decision and FingerPrint evolution as a function of the time.

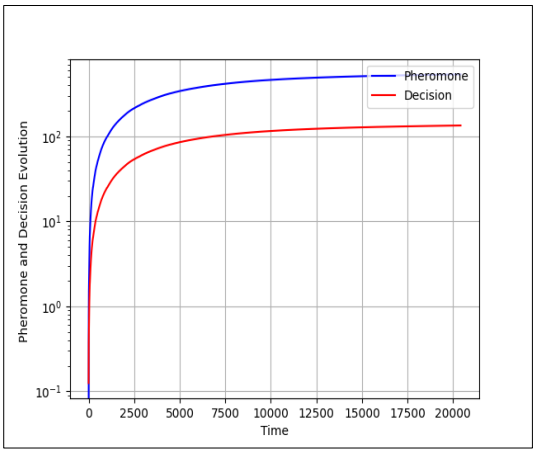

Figure 7. Pheromone and decision evolution as a function of the time.

In figure 6, we see the decision is increments steady, while the colony fingerprint evolution remains non-existent, which means no attacks. The wider the gap between the colony fingerprint and the decision, it allows to always trust the incoming and outgoing traffic. As shown in (fig:7) the evolution of the decision follows the same pattern as the pheromone since the pheromone plays a central role in the decision equation. the steady increments in pheromone can be also explained, on the machine learning model informed the colony about its prediction (in this case positive feedback) and the colony acts accordingly to update both pheromone fingerprint, and then contribute to decision making.

5.2. Second scenario

In this scenario, the controller is initiated with a legitimate traffic like the first scenario. A second illegitimate traffic is passed using Hping flood command to simulate a DoS traffic consisting of a million packet. The same source host and destination host is used to simulate a sudden change in traffic behaviour. Host 1 will send a 100000 packet using ping to host 2 then a host 1 floods host 2.

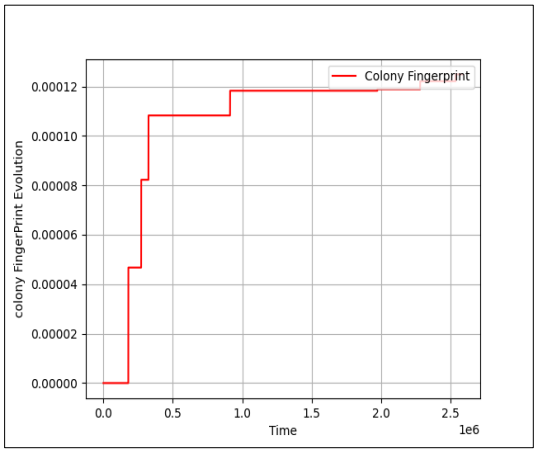

Figure 8. colony FingerPrint evolution as a function of the time.

Figure 8 shows in the first, the fingerprint remains constant due to legitimate traffic, then it a sudden increment, this is due to the detect an attack, and the colony fingerprint keeps increasing over time because illegitimate traffic.

The decision in (fig:11,12) shows slightly an incremental behaviour following the legitimate traffic, then after a bit time and due to the machine learning model informed the colony about its prediction (in this case negative feedback). the decision curve down to minimise the gap with the colony fingerprint to allow the illegitimate traffic to be detected.

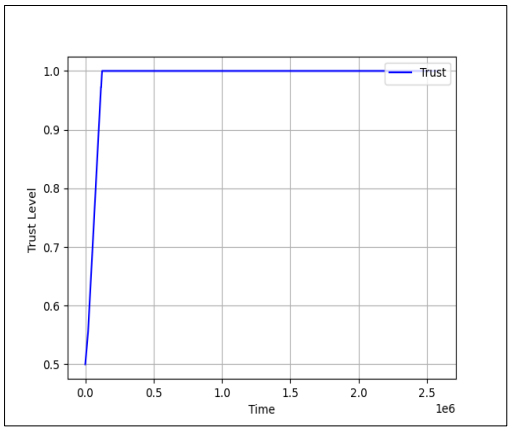

Figure 9. host 1 trust to host 2

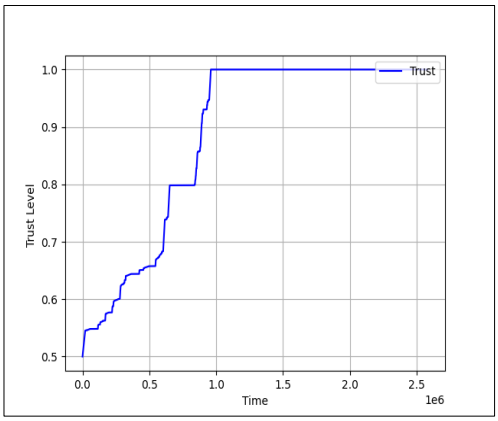

Figure 10. host 2 trust to host 1

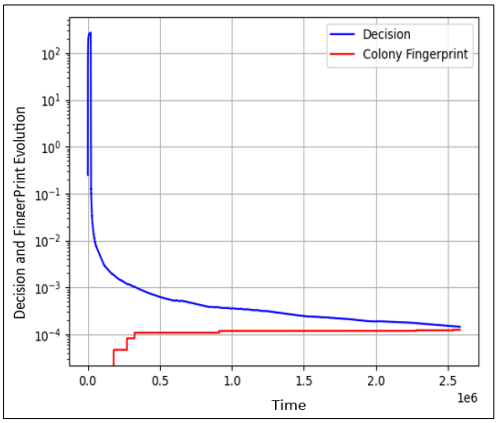

Figure 11. Decision and FingerPrint evolution as a function of the time.

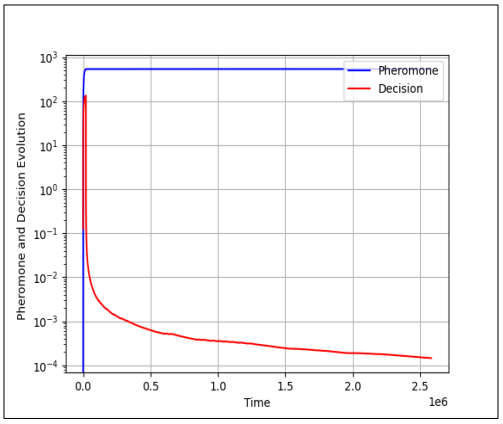

Figure 12. Pheromone and decision evolution as a function of the time.

As shown in the figure, the evolution of the decision in this case did not intersect with the colony fingerprint therefore the trust is not decremented figure 9,10 and this behaviour is due to the meta-heuristic behaviour of our approach.

5.3. Third scenario

In this scenario, the victim host is changed to show that the behaviour depends on every single traffic link on the network. Therefore, like the second scenario a legitimate traffic initiates the controller using a simple ping consisting of a 100000 packet from host 1 to host 2. A second traffic is initiated shortly after the first one ends from host 1 to host 3 changing the destination host of the illegitimate traffic where host 1 even though it started sending legitimate traffic to host 2 it floods host 3 with a million packet.

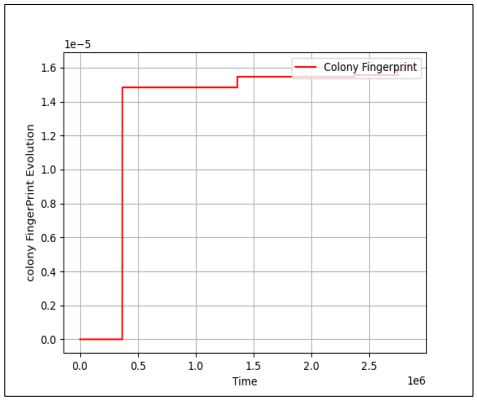

Figure 13. colony FingerPrint evolution as a function of the time.

As expected in fig:13 the Colony fingerprint shows multiple evolution points, and the reason the presence the illegitimate traffic.

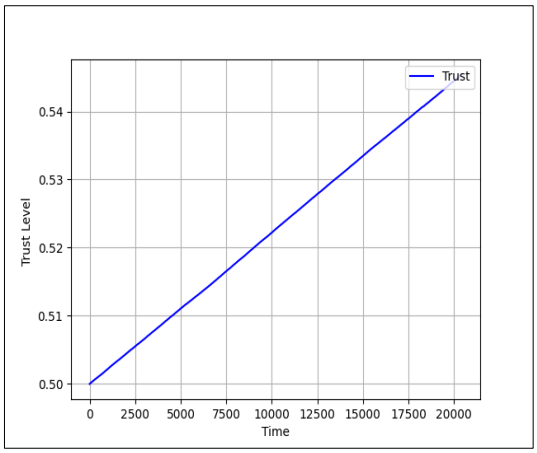

Figure 14. host 2 trust to host 1

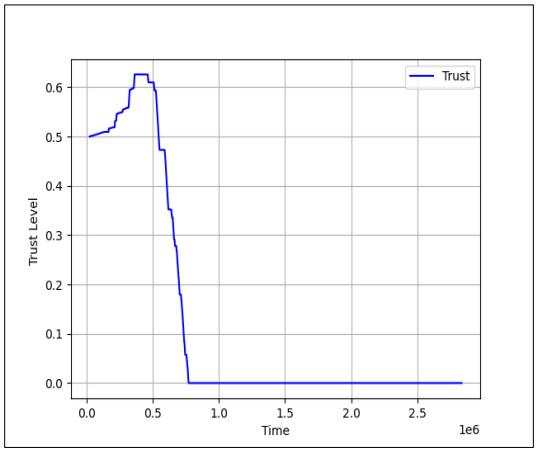

Figure 15. host 3 trust to host 1

As compared to the trust level evolution, in Figure (fig:15) shows the evolution of trust level from host 3 to host 1.
Since the nature of traffic is illegitimate, we see clearly a slow increment and then a steady decrement of trust until it reaches the minimum trust level of 0.

Both figures (fig:14,15) shows the independent behaviour in how attacks are handled by the ML-ACID, as if an illegitimate host only targets a single host inside a network, it will not be trusted only by that host as in the case with a DOS attack and not the entire networks.

On the other end, if a set of hosts are hijacked inside of a network to target a single or even multiple other hosts such in the case of DDOS. In this case, every illegitimate traffic is isolated and treated as a single threat.

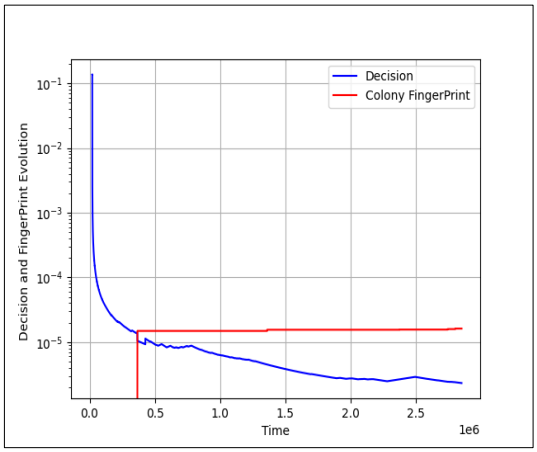

Figure 16. Decision and FingerPrint evolution as a function of the time.

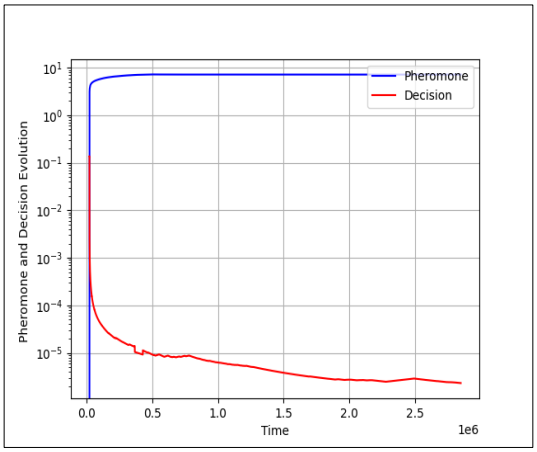

Figure 17. Pheromone and decision evolution as a function of the time.

we see in figure 16, the decision rapidly curves down, due to sending legitimate and illegitimate traffic at almost the same time. also, we see the colony fingerprint exceeds the decision at a point,
at that point, an attack is detected. while the figure 17, we find the pheromone remains constant due to the presence of attack. we see the independent behaviour in how ML-ACID handles attacks, when sending both traffic legitimate and illegitimate at almost the same time the attacks are detected faster compared to the previous scenario.

5.4. Fourth scenario

In this scenario, both the victim host and the attacker source are changed to show that the behaviour depends on every single traffic link on the network even if illegitimate traffic starts without initiating the controller. Therefore, like the third scenario a legitimate traffic initiates the controller using a simple ping

consisting of a 100000 packet from host 1 to host 2. A second traffic is initiated shortly after the first one ends where host 4 floods host 5 with a million traffic.

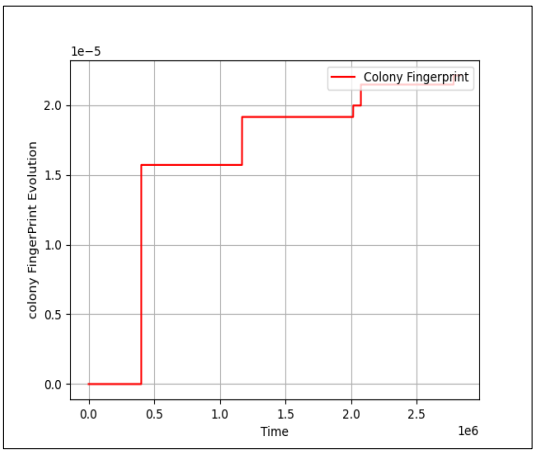

Figure 18. colony FingerPrint evolution as a function of the time.

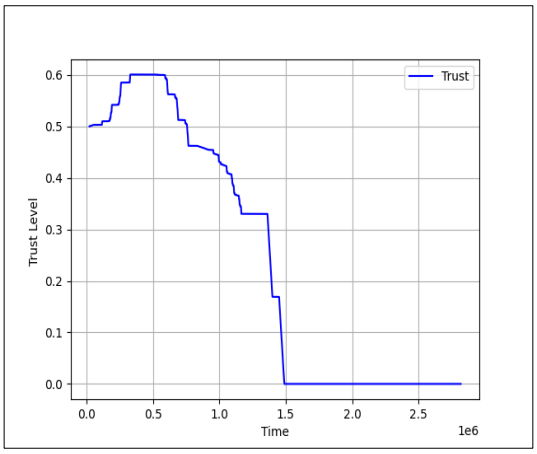

Figure 19. host 5 trust to host 4

Figures (fig:18, fig:19, fig:20, fig:21) shows the same behaviour of the third scenario despite the fact the only illegitimate traffic exists on the link between host 4 and host 5. Therefore, our approach can clearly detect even early illegitimate traffic on the network.

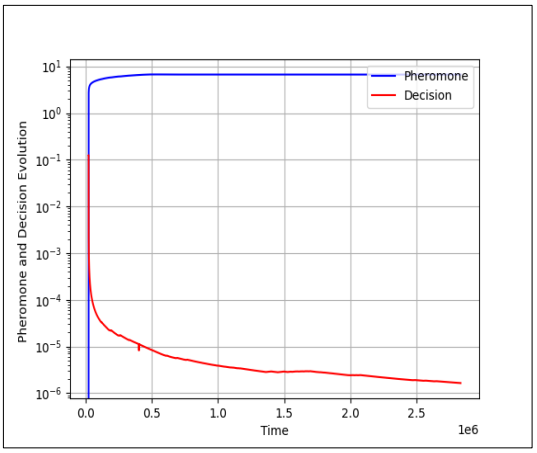

Figure 20. colony Fingerprint evolution as a function of the time.

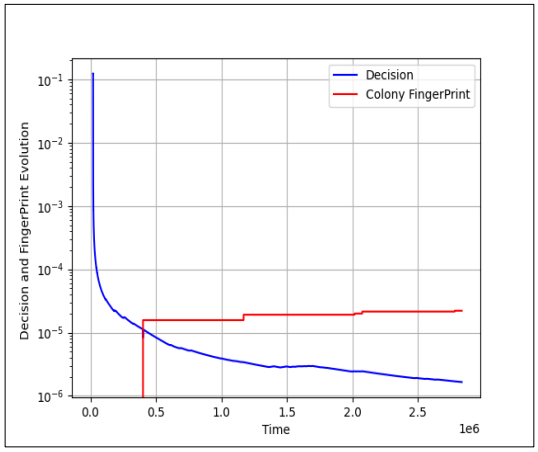

Figure 21. Pheromone and decision evolution as a function of the time.

As explained previously, the decision Figures 20 at a higher point making a bigger gap with the colony fingerprint. Since the traffic is of illegitimate nature, there exists a point where the colony fingerprint exceeds the decision making the module rejects the traffic from there on.
While the pheromone evolution remains constant due to the presence of the attack, until there is positive feedback from the machine learning model that helps to reset the trust and pheromone.

6. CONCLUSION

In this study, we proposed an effective system for detecting both Dos, DDoS attacks in SDN for IoT using two Artificial intelligence techniques. Combining both machine learning and a novel ant colony based meta-heuristic. A simple machine learning model provides the system with basic detection capabilities to drive the evolution of the system detection capability. The meta-heuristic behaviour of artificial ants representing each a host within the network, that tries to mimic ants in real life to detecting intruders.

Several scenarios were proposed to show that the proposed system has an independent behaviour in how it handles the traffic. Thus, providing the system with a capability to both detect Dos and DDOS attacks within the network. The adaptive evolution of the colony fingerprint allows for the fast detection of illegitimate traffic compared to other methods as shown in the results. Moreover, the approach is not only limited to these types of attacks and can be extended to other types. However, in case of an attack, our

proposed approach takes a bit time to reset the trust and pheromone and also the colony's fingerprint for allows the ants (hosts) to communicate again.

While the system is well-suited for a wide range of IoT environments, future research should focus on addressing the above-mentioned challenge and addressing scalability challenges and exploring the integration of more advanced machine learning techniques to further enhance detection accuracy.

REFERENCES

[1] Cisco. Global 2021 forecast highlights. https://www.cisco.com/c/dam/m/en us/solutions/service-provider/vni forecast-highlights/ pdfGlobal 2021 Forecast Highlights.pdf, 2019.

[2] Nick G. How many iot devices are there in 2022? https://techjury.net/blog/how-many-iot-devicearethere/, 2022.

[3] Hamza Belkhiri, Abderraouf Messai, André-Luc Beylot, and Farhi Haider. Denial of service attack detection in wireless sensor networks and software defined wireless sensor networks: A brief review. In International Conference On Big Data and Internet of Things, pages 100–115. Springer, 2022.

[4] Hamza Belkhiri, Abderraouf Messai, Mohamed Belaoued, and Farhi Haider. Security in the internet of things: recent challenges and solutions. In International Conference on Electrical Engineering and Control Applications, pages 1133–1145. Springer, 2021.

[5] Mohamed Takieddine Seddik, Ouahab Kadri, Chakir Bouarouguene, and Houssem Brahimi. Detection of flooding attack on obs network using ant colony optimization and machine learning. Computación y Sistemas,25(2):423–433, 2021.

[6] Jeyaram. Intrusion detection system based on combined support vector machine with ant colony optimization. Journal on Software Engineering, 11(4), 2017.

[7] Steven Johnson. Emergence: The connected lives of ants, brains, cities, and software. Simon and Schuster, 2002.

[8] Tristram D Wyatt. Pheromones and animal behavior: chemical signals and signatures. Cambridge University Press, 2014.

[9] Ramprasath and V Seethalakshmi. Improved network monitoring using software-defined networking for ddos detection and mitigation evaluation. Wireless Personal Communications, 116(3):2743–2757, 2021.

[10] Yonglin Liang and Lirong Qiu. Network traffic prediction based on svr improved by chaos theory and ant colony optimization. International journal of future generation communication and networking, 8(1):69–78, 2015.

[11] Sakchi Jaiswal, Khushboo Saxena, Amit Mishra, and Shiv K Sahu. A knn-aco approach for intrusion detection using kddcup'99 dataset. In 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), pages 628–633. IEEE, 2016.

[12] Hsia-Hsiang Chen and Shih-Kun Huang. Lddos attack detection by using ant colony optimization algorithms. Journal of Information Science & Engineering, 32(4), 2016.

[13] Aanshi Bhardwaj, Veenu Mangat, and Renu Vig. Hybrid deep neural architecture for detection of ddos attacks in cloud computing. In Intelligent Systems, Technologies and Applications, pages 71–86. Springer, 2021.

[14] D Arivudainambi, Varun Kumar KA, and S Sibi Chakkaravarthy. Lion ids: A meta- heuristics approach to detect ddos attacks against software-defined networks. Neural Computing and Applications, 31(5):1491–1501, 2019.

[15] Ancy Sherin Jose, Latha R Nair, and Varghese Paul. Towards detecting flooding ddos attacks over software defined networks using machine learning techniques. REVISTA GEINTEC-GESTAO INOVACAO E TECNOLOGIAS, 11(4):3837–3865, 2021.

[16] Özgür Tonkal, Hüseyin Polat, Erdal Başaran, Zafer Cömert, and Ramazan Kocaoğlu. Machine learning approach equipped with neighbourhood component analysis for ddos attack detection in software-defined networking. Electronics, 10(11):1227, 2021.

[17] Aye Thandar Kyaw, May Zin Oo, and Chit Su Khin. Machine-learning based ddos attack classifier in software defined network. In 2020 17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), pages 431–434. IEEE, 2020.

[18] Huseyin Polat, Onur Polat, and Aydin Cetin. Detecting ddos attacks in software-defined networks through feature selection methods and machine learning models. Sustainability, 12(3):1035, 2020.

[19] Ranyelson N Carvalho, Lucas R Costa, Jacir L Bordim, and Eduardo AP Alchieri. Detecting ddos attacks on sdn data plane with machine learning. In 2021 Ninth International Symposium on Computing and Networking Workshops (CANDARW), pages 138–144. IEEE, 2021.

[20] Kshira Sagar Sahoo, Bata Krishna Tripathy, Kshirasagar Naik, Somula Ramasubbareddy, Balamurugan Balusamy, Manju Khari, and Daniel Burgos. An evolutionary svm model for ddos attack detection in software defined networks. IEEE Access, 8:132502–132513, 2020.

[21] Wu Zhijun, Xu Qing, Wang Jingjie, Yue Meng, and Liu Liang. Low-rate ddos attack detection based on factorization machine in software defined network. IEEE Access, 8:17404–17418, 2020.

[22] Jesus Arturo Perez-Diaz, Ismael Amezcua Valdovinos, Kim-Kwang Ray-mond Choo, and Dakai Zhu. A flexible sdn-based architecture for identifying and mitigating low-rate ddos attacks using machine learning. IEEE Access, 8:155859–155872, 2020.

[23] Liang Tan, Yue Pan, Jing Wu, Jianguo Zhou, Hao Jiang, and Yuchuan Deng. A new framework for ddos attack detection and defense in sdn environment. IEEE Access, 8:161908–161919, 2020.

[24] K Muthamil Sudar, M Beulah, P Deepalakshmi, P Nagaraj, and P Chinnasamy. Detection of distributed denial of service attacks in sdn using machine learning techniques. In 2021 International Conference on Computer Communication and Informatics (ICCCI), pages 1–5. IEEE, 2021.
[25] Frank Wenzel. Smell and repel: resin-based defense mechanisms and interactions between Australian ants and stingless bees. PhD thesis, Universität Würzburg, 2011.
[26] Lisa Jean Moore and Mary Kosut. Buzz: Urban beekeeping and the power of the bee. NYU Press, 2013.
[27] B Chandra Mohan and R Baskaran. A survey: Ant colony optimization based recent research and implementation on several engineering domain. Expert Systems with Applications, 39(4):4618–4627, 2012.
[28] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vander-plas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duch-esnay. scikit -learn: Machine learning in Python. Journal of Machine Learning Research, 12:2825–2830, 2011.