Journal of Intelligent Systems and Internet of Things
Vol. 12, No. 01, PP. 129-143, 2024
DOI: https://doi.org/10.54216/JISIoT.120110

An adaptive distributed intrusion detection system in local network: Hybrid classification methods

Amjad Hijazi[1,*], Nizar Alhafez[2], Iyad Al-khayat[3]

[1] PhD Student at the Department of Computer Networks and Systems, Faculty of Information Technology Engineering, Damascus University, Damascus, Syrian Arab Republic
[2] Professor at Faculty of Information Technology Engineering, Damascus University, Damascus, Syrian Arab Republic
[3] Assistant professor at Faculty of Information Technology Engineering, Damascus University, Damascus, Syrian Arab Republic
Emails: amjad01.hijazi@damascusuniversity.edu.sy ;nizar.alhafez@damascusuniversity.edu.sy; iyad.khayat@damascusuniversity.edu.sy
[*]Corresponding Author: amjad01.hijazi@damascusuniversity.edu.sy

Abstract

In the realm of cybersecurity, the incessant evolution of network attacks necessitates advanced and robust intrusion detection systems (IDS). The major issues with these systems are numerous: false positive/negative alarms, delayed response and detection time, size of processed data, adaptability to future threats, scalability of the system, difficulty in detecting distributed attacks, and downtime (fault tolerance). We propose a system that introduces a distributed framework aimed at enhancing network security by effectively identifying subtle deviations from normal network behavior. This is achieved through transfer learning based on artificial neural networks, and support vector machine (SVM), capitalizing on their complementary strengths in recognizing complex patterns and addressing high-dimensional datasets. To validate the efficacy of the proposed approach, the NSL-KDD dataset is utilized within a distributed IDS architecture. It consists of several intrusion detection nodes representing subnetworks. A node consists of two agents that work collaboratively. A way is proposed to avoid interference between analysis agents: the network agents manager monitors the functioning of the nodes and displays the results of each vulnerability-detecting node in each subnet separately. Such communication between agents should reduce FPAS (false positive alarms) significantly. The Detection engine extracts relevant features of network attacks to solve the problem of SVM in processing huge sizes of data and detect adaptive future threats to detect famous distributed denial of services (DDOS) attacks in real-time. The system is highly scalable by increasing the number of intrusion detection system nodes if necessary. Central processing is avoided to circumvent a system failure situation, where processing and decision-making take place at the detection node level within each subnet.

Received: August 19, 2023 Revised: November 19, 2023 Accepted: March 01, 2024

Keywords: distributed intrusion detection system; network security; agents, denial of service; artificial neural networks; support vector machine.

1. Introduction:

The role of intrusion detection systems is to identify whether the traffic has patterns that represent a network attack or normal traffic. In the event of a classification error, this leads to false positive and negative alarms [1]. Intrusion detection techniques are categorized as follows: anomaly detection and misuse detection [2].

IDS (intrusion detection system) based on the misuse detection model can detect attacks that have signatures and generate fewer false positives (FPA). IDS has a major drawback, however, that novel attacks will go undetected until signatures for those intrusions are known to the IDS. While IDSs based on anomaly detection models have a better chance of detecting novel intrusions, however, they are slow due to exhaustive monitoring and the use of many resources. Also, their rate of generating false positive alarms is higher [2].
There are three types of IDS: network intrusion detection system (NIDS), host intrusion detection system (HIDS) and multi IDS, including HIDS or NIDS, classified into three architectures: distributed intrusion detection system (DIDS), collaborative intrusion detection system (CIDS) and collaborative intrusion detection Network (CIDN) [3], Overall, all three architectures are effective ways to improve the accuracy and efficiency of intrusion detection by leveraging the collective knowledge and resources of multiple IDS systems. The choice between them depends on the specific needs and requirements of the organization or network being protected.
Most distributed intrusion detection systems (DIDS) that use a planned framework with various mobile agents and well-supported coordinator agents can resolve the current issues in intrusion detection systems.

The remaining of this work is structured as follows:

- Section 2 details the related literature done on single IDS and distributed intrusion detection.
- Section 3 describes the proposed system architecture for the IDS to work, and the communication model of the analysis and detection agent for the agent interaction with other entities.
- Section 4 debates the performed experiments and the corresponding results.
- Section 5 concludes our contributions and findings.

2. Related Work

We studied a wealth of literature that relies on independent intrusion detection systems, both as a single IDS and as a distributed IDS. The most important evaluation criteria and methodology they used are [1]: detection rate (DR), accuracy (ACC), false positive alarm rate (FAR), detection time, and finally fault tolerance (FT).
Most research uses central processing (to analyze data and decide whether there is an attack or not) which is at risk of failing if the intrusion detection system fails. Moreover, central processing is not able to detect malicious events occurring at different places at the same time such as a (DDOS) [4]. Some research suggests [5] to transfer data from one place to another over the network for the analysis and detection process or to transfer the analysis and detection process to the location of the data. These actions would cause a load on the network, increase the rate of negative and positive false alarms, and lead to a delay in response time.
In all of the reference studies, there is a large chain of agents working together to detect attacks. The size of the payload depends on the mechanism of exchanging data between agents [6], which causes an increase in response time. Most of the proposed solutions give good results in terms of accuracy and detection rate but do not attention to false alarm rates and detection time. also in distributed systems, the system failure condition is not handled. Table 1 resumes our study of the literature.

Table1: Related works on intrusion detection system.

REF	Approach	Type of IDS		Data set	Detected attack	Number of Features	Effectiveness			Efficiency	Classification Type		FT
		Single	DIDS and how many agents/units				DR %	ACC %	FAR %	Detection time (s)	Binary (0 or 1)	Multiclass	
[7]	ANN(Sensitivity analysis) as feature selection +SVM (RBF) as classifier.	ok	*	NSL-KDD	*	17	94.02	96.88	*	*	*	*	*
[17]	Rule-based for extracting features then three classifiers work in parallel (GRU, CNN and Random Forest).	ok	*	NSL-KDD	*	41	*	87.28	*	When using RF 16.32	*	*	*
[8]	Embedded function using CNN and DNN as Feature Extraction and Few-Shot Learning for classification.	ok	*	NSL-KDD	DoS (Teardrop, Smurf), Probe (Satan,Portsweep, saint), U2R (Rootkit, Buffer overflow Load module) and R2L (Xsnoop, HTTP tunnel).Other attack types tested were: normal, generic, fuzzers, reconnaissance, shellcode, worms, backdoor and exploits)	*	DOS 94.07 R2L 75.93 U2R 81.50 Probe 92.65	92.34	For all attacks 3.95	*	*	ok	*
[1]	G-ABC for optimizing features in the first stage and DNN for Classification.	ok	*	NSL-KDD	DOS Probe U2R R2L	*	99.85 99.85 99.86 99.79	99.42 98.89 99.01 99.81	*	*	*	ok	*

[9]	Misuse detection and Migration agents between hosts in subnetworks and controlled by management servers.	*	Ok / 5	*	Evasion Techniques, DOS, Client Side, IP Reputation, Shell Codes, Brute Force, Fragmented Packets, Test Rules	*	*	93	33	*	*	*	ok
[10]	Implementation of misuse detection, Central supervisor agent controls and manages MAS-DIDS Agents.	*	Ok /5	*	DOS, U2R, R2L	*	72 for Dos, 63.11 for U2R, 81 for R2L	*	11 for DOS, 18 U2R and 8.3 for R2L	*	*	*	*
[11]	The central server works to extract features and publish incoming traffic to all nodes to detect anomalies, using a voting algorithm (weighted majority algorithm) to classify traffic.	*	Ok/7	NSL-KDD	DOS, U2R, R2L, Probe.	*	98	*	0.09	*	ok	*	*
[12]	Anomaly detection based on naïve bayes and random forest.	*	Ok/5	CIC_IDS001	DOS, PortScan, Ping scan, Brute force	*	*	97	0.21	6.23	*	Ok	*
	random forest.							85	0.43	24.87			
[13]	Multi-agent-based DIDS used an ensemble data mining approach between (SVM, ANN, and Random forest).	*	Ok/4	KDD-CUP	DOS, U2R, R2L, Probe.	*	96.1	DOS 99.9 Probe 92.1 U2R 96.4 R2L 91.1	0.12	*	*	OK	*

3. The proposed System Architecture

This paper covers a detection engine design and implementation using a machine learning algorithm, called Hybrid ANN-SVM, for anomaly detection. We provide a general framework for testing a hybrid intrusion detection engine that works in a fully distributed IDS (DIDS) fashion in real time.
The proposed system consists of two phases, the first phase is the detection engine with a transfer learning-based approach, and the second phase covers a distributed intrusion detection mechanism.

3.1 Detection engine design and implementation

Any decision-making system that deals with a large amount of data requires efficient preparatory processing of the data. Then the system builds the initial structure of the neural network classifier using the full features to compare with later on. The selection of the features affecting the classification process in several stages is based on several techniques to select the most relevant features from the dataset to classify the network traffic. In features extraction we start by calculating the variance values of the features, and eliminating features with zero variances, then finding the highest correlation between the features (by analyzing the activation contribution of each of the input dimensions (features)) to determine the important features in the classification process [14], The selection of features helps in selecting the minimum subset of features necessary for intrusion detection and thus reducing the amount of data processed by the system to detect intrusion more quickly and effectively, and based on the above, the final structure of the neural network classifier for the proposed system was built. Figure 1 depicts the Detection Engine.
The basic principle of machine learning was applied to obtain a model that was trained on a data set (NSL-KDD) to detect abnormal behavior (Training Model Offline) [15].

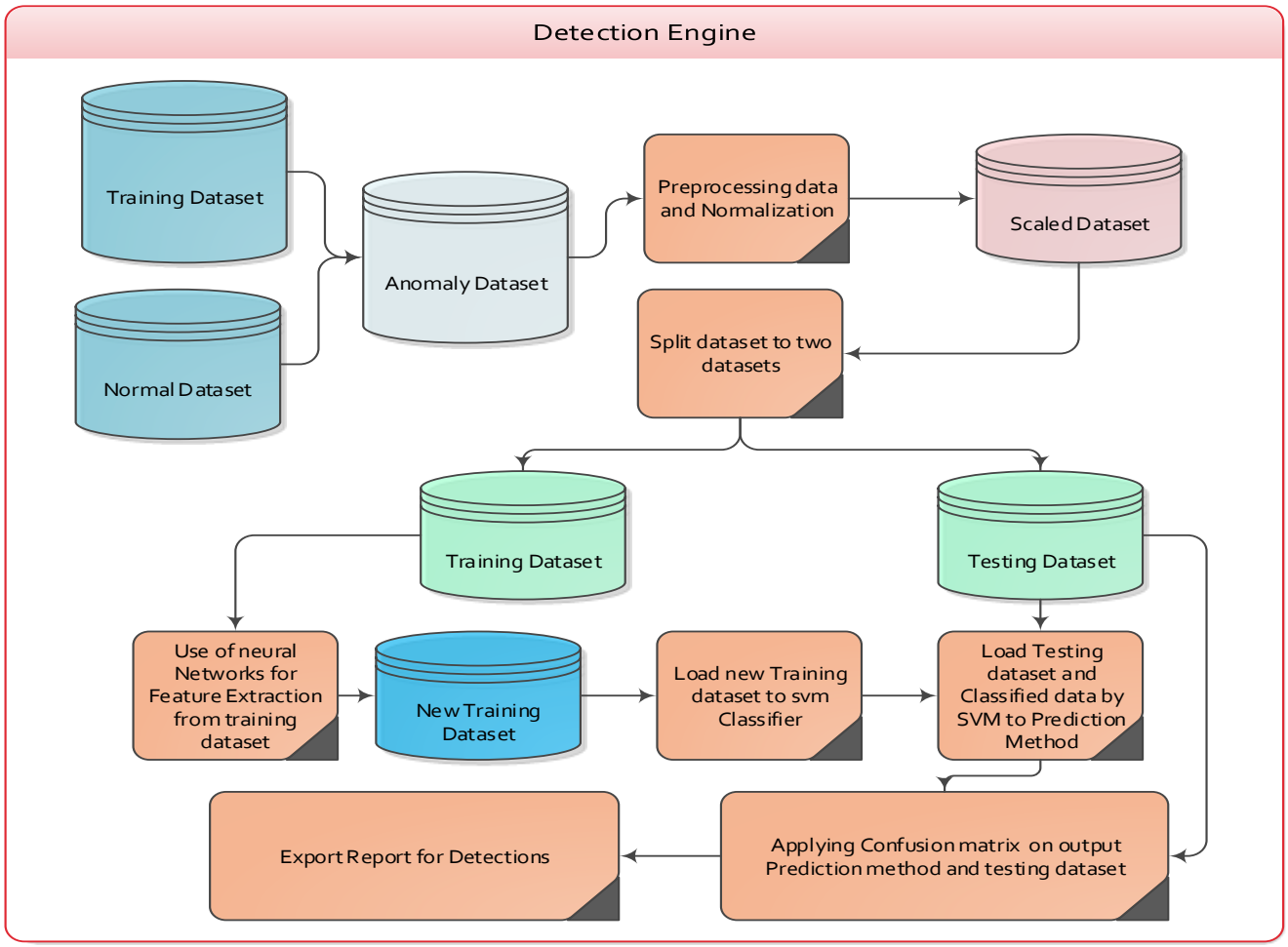

Figure 1: Detection Engine.

3.2. Distributed intrusion detection system architecture

the proposed architecture is a decentralized system, IDS Node contains two collaboration agents that will collect the network data, and analyze and identify the intrusions. The JADE tool forms nodes representing an intrusion detection system [16]. The (CICFlow Meter) tool captures network packets. Figure 2. depicts a communication diagram between two intrusion detection nodes that are part of a distributed system.

Each node has two agents. First, a sniffer agent based on the CICFlow Meter tool for extracting features from network traffic packets and storing them in the dataset for analysis within every time window of (20) seconds. Second, an analysis agent based on a detection engine analyzes and decides whether there are network attacks or not. For the analysis agent, data for analysis is received according to its chronological arrival, without the need for an algorithm for distributing the data between the analysis agents within the nodes. Analysis agent works with parallel behavior, either as agents manager or as analysis agent.

The duties of the agents manager are: to monitor the agents' work within the nodes, to trigger an alarm if the agent is out of service, and to receive analysis results from analysis agents via an agent communication protocol (ACL).

Figure 3 depicts the architecture of the IDS node with four layers. Each layer represents the tasks of the agents that are defined in Figure 2

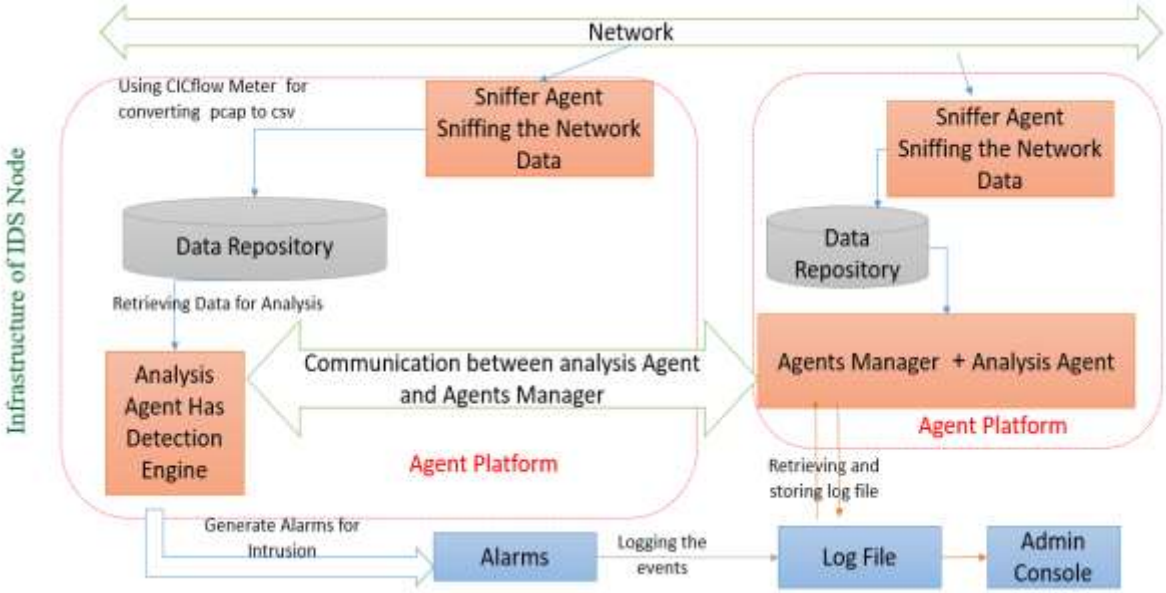

Figure 2: Communication between Two IDS nodes.

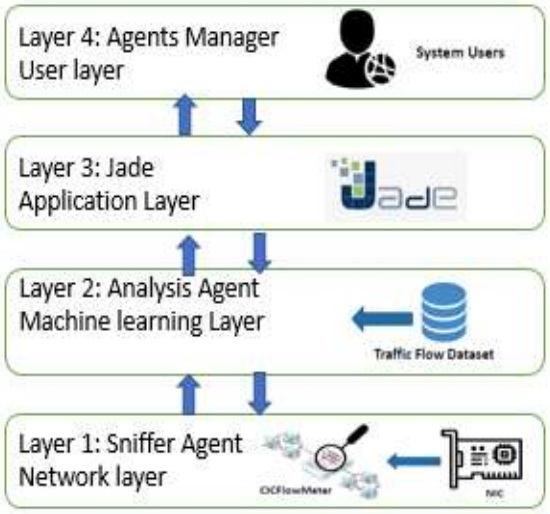

Figure 3: Architecture of IDS Node.

3.2.1 Scheduler algorithm between Analysis Agents

We consider solving the problem of centralization at the node level in case the Agents Manager fails to elect a new agent manager, which monitors the work of the node and performs the task of an analysis agent in the network branch where it is located.

The test environment has the main container including all agents (Sniffer and Analysis) so that each Analysis Agent joined the main container gets a number to represent the priority for working with two behaviors in parallel.

The proposed architecture helps to operate with transparency and accessibility for agents and to modify their behavior in real-time, without the analysis agents interfering with each other. Before starting the scheduling algorithm, we must distinguish between the following cases of agents. Table 2. depicts the description of the behavior Analysis Agent.

Table 2. The behavior of the Analysis Agent.

STATUS OF ANALYSIS AGENT	BEHAVIOR	DESCRIPTION
Active = 1	Parallel Behavior: Agents Manager + IDS	Analysis agent in node IDS works with two Behaviors
Active = 0 (Passive)	Cyclic Behavior IDS	Analysis agent in node IDS works with cyclic analysis Behavior

Scheduler Algorithm

The algorithm of the scheduler is as follows:

- If there isn't any Analysis Agent in the main container then
  Make this Analysis Agent as the main agent to work with parallel behavior
  Change the status of this agent to Active =1.
- Else
  Make this Analysis Agent work as cyclic Behavior, and set the status of this agent to Active =0.
- Each Analysis Agent sends a message (keep_alive_IDS) to the Agents Manager every 30 s.
- Each Analysis Agent receive message (ok_Keep_Alive) from Agents Manager every 30 s.
- If (30) seconds have passed and the Analysis Agent didn't receive (ok_Keep_Alive) from the Agents Manager
  - The Analysis Agents within the IDS node elect a new Agents Manager based on the priority of the Analysis Agent joining the container.
  - Change the status of the new Analysis Agent to active =1.
  - Change the status of the old Agents Manager to active =0.
  - Broadcast message (Voting_Agents_manager) from new Agents Manager to all Analysis Agents
  - Every Analysis Agent sends a message (OK_Voting_Agents_manager) to approve and communicate with the new Agents Manager.
- All Analysis Agents send message (Send_Analysis_status) to Agents Manager every 60s.
- Every Analysis Agent receives a message (Recive_Analysis_Status) every 60s.
- If (60) seconds have passed and the Analysis Agent didn't receive (Recive_Analysis_Status) from the Agents manager
  - The Analysis Agents within the IDS node elect a new Agents Manager based on the priority of the Analysis Agent joining the container.
  - Change the status of the new Analysis Agent to active =1.
  - Change the status of the old Agents Manager to active =0.
  - Broadcast message (Voting_Agents_manager) from new Agents Manager to all Analysis Agents
  - Every Analysis Agent sends a message (OK_Voting_Agents_manager) to approve and communicate with the new Agents Manager.

The message (Send Analysis Status) contains information about detection (Accuracy, Detection rate, false alarm rate, detection time, number of packets that have been classified as an attack, and number of packets that have been classified as normal traffic).

Table 3. depicts the exchanged messages between the Analysis Agents and the Agents Manager, and the time for each. Figure 4 depicts the communication diagram between agents.

Table 3: Exchanged messages between the Analysis Agents and Agents Manager.

NUM OF MSG	MESSAGE	STATUS	TIME (S)	SOURCE	DESTINATION
1	Keep_Alive_IDS	Cyclice_Send	30	IDS Node	Agents manager
2	Ok_Keep_ALive	Cyclice_Recive	30	Agents Manager	IDS node
3	Send_Analysis_Status	Cyclice_Send	60	IDS Node	Agents Manager
4	Recive_Analysis_Status	Cyclice_Recive	60	Agents Manager	IDS node
5	Voting_Agents_manger	OneShot_send	If lost, the Agents Manger sent once to all IDS Node	New Agents Manager	All IDS node
6	OK_Voting_Agents_manger	OneShot_send	If voting, the new Agents Manger sent once to all IDS node	All IDS Node	New Agents Manager

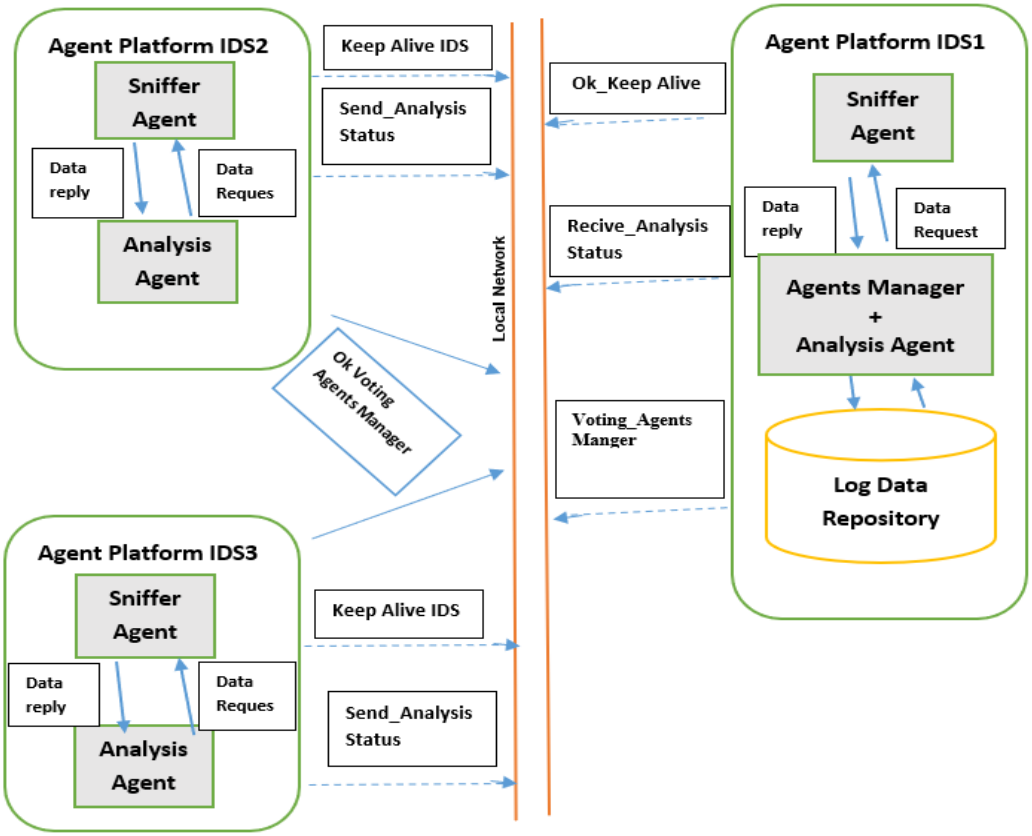

Figure 4: Communication diagram between agents.

3.2.2 The experimental layout of the system and testbed environment

Figure 5 depicts a network environment set up in which various workstations were deployed by connecting them with switches. We have four subnets, and in each subnet, an IDS node was deployed through the JADE platform. All the packets to the outside networks pass through the DMZ (router and firewall) in subnet-4.

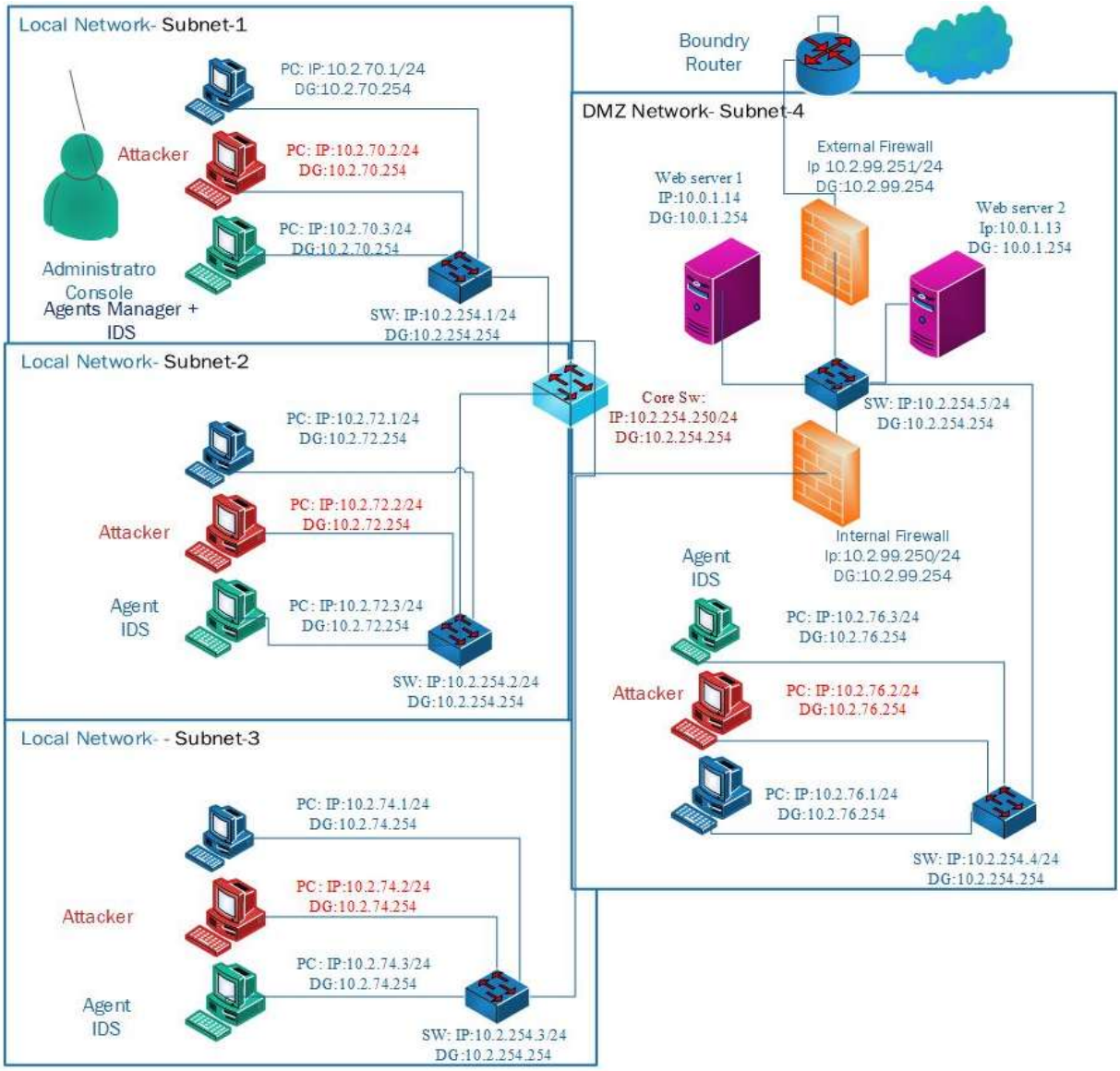

Figure 5: The experimental layout.

Table 4. describes the status of the IDS node.

Table 4: Description of the status of the IDS node

NODE	PRIORITY	STATUS OF NODE	TASK BEHAVIOUR
Node1	1	Active	Parallel Behaviour: Agents Manager + IDS
Node2	2	Passive	IDS
Node3	3	Passive	IDS
Node4	4	Passive	IDS

4. Results and Discussion

Table 5. depicts network attacks applied to the proposed approach.

Table 5: Type of attacks applied on the experimental layout.

NUM	TYPE OF ATTACK	ATTACK ON NETWORK RESOURCES	ATTACKS ON WORKSTATION RESOURCES
1	Dos	UDP Flooding Attack	TCP SYN Flooding
		ICMP Flooding Attack	LOW and Slow Attack
2	U2R		Buffer over Flow
3	Probe		NMAP Stealth Scan

4.1 We consider all DOS attacks (SYN flooding, low and slow, ICMP flooding and UDP flooding).

Table 6: depicts Dos Results

Attack Type	Node	Type of node	Type of traffic	Accuracy	Detection Rate	False Alarm Rate	Time(S)	Attack Session	Normal Session
DDOS-Syn Flooding attack	We consider Syn flooding Attack from an attacker in subnets (1,2,3) to the web server (10.0.1.14) in DMZ Subnet.								
	Node1	Agents Manager	Normal	98.21	98.18	0	1.4	0	985
	Node2	IDS	Normal	99.2	98.62	0	1.7	0	1061
	Node3	IDS	Dos	96.01	90.91	0.54	3.778	998	7100
	Node4	IDS	Normal	98.22	98.59	0	1.2	0	722
	Node1	Agents Manager	Normal	99.51	98.19	0	1.3	0	873
	Node2	IDS	Normal	98.12	99.16	0	1.1	0	579
	Node3	IDS	Dos	96.01	90.91	0.54	3.778	998	7100
	Node4	IDS	Dos	95.29	96.71	0.01	1.56	99	1380
	Node1	Agents Manager	Dos	94.22	96.89	0.05	1.983	246	2030
	Node2	IDS	Normal	98.23	98.45	0	1.82	0	1252
	Node3	IDS	Dos	96.01	90.91	0.54	3.778	998	7100
	Node4	IDS	Dos	95.29	96.71	0.01	1.56	99	1380
	Node1	Agents Manager	Dos	94.22	96.89	0.05	1.983	246	2030
	Node2	IDS	Dos	95.22	96.99	0.12	2.3	500	4000
	Node3	IDS	Dos	96.01	90.91	0.54	3.778	998	7100
	Node4	IDS	Dos	95.29	96.71	0.01	1.56	99	1380
DOS Attack (Low and Slow attack)	We consider Attack from subnet-2 to web server (10.0.1.14) in DMZ subnet-4.								
	Node1	Agents Manager	Normal	99.11	99.01	0	1.7	0	721
	Node2	IDS	Dos	97.58	96.99	0.02	2.3	500	1985
	Node3	IDS	Normal	99.13	99.66	0	0.9	0	501
	Node4	IDS	Normal	99.4	99.54	0	0.7	0	223
	Node1	Agents Manager	Normal	99.51	98.19	0	1.3	0	873
	Node2	IDS	Dos	97.58	96.99	0.02	2.3	500	1985
	Node3	IDS	Normal	99.97	98.9	0	1.2	0	601
	Node4	IDS	Dos	99.1	99.7	0.01	1.9	460	1754
	We consider Attack from subnet-3 to network resource (switch 10.2.254.250).								

DOS attack (ICMP Flooding attack)	Node1	Agents Manager	Normal	99.91	99.1	0	1.22	0	1120
	Node2	IDS	Normal	99.88	99.4	0	1.1	0	943
	Node3	IDS	Dos	99.5	99.7	0.01	0.4	1215	410
	Node4	IDS	Normal	99.27	98.59	0	0.6	0	517
DOS attack (UDP-Flooding attack)	We consider Attack from subnet-4 to network resource (switch 10.2.254.250).								
	Node1	Agents Manager	Normal	99.2	98.19	0	1.3	0	873
	Node2	IDS	Normal	99.93	99.16	0	1.1	0	579
	Node3	IDS	Normal	99.9	99.71	0	1.2	0	601
	Node4	IDS	Dos	99.7	99.4	0.03	1.9	157	5000

Figure 6 depicts a diagram of the best test DDOS attack done on the internal and external nodes.

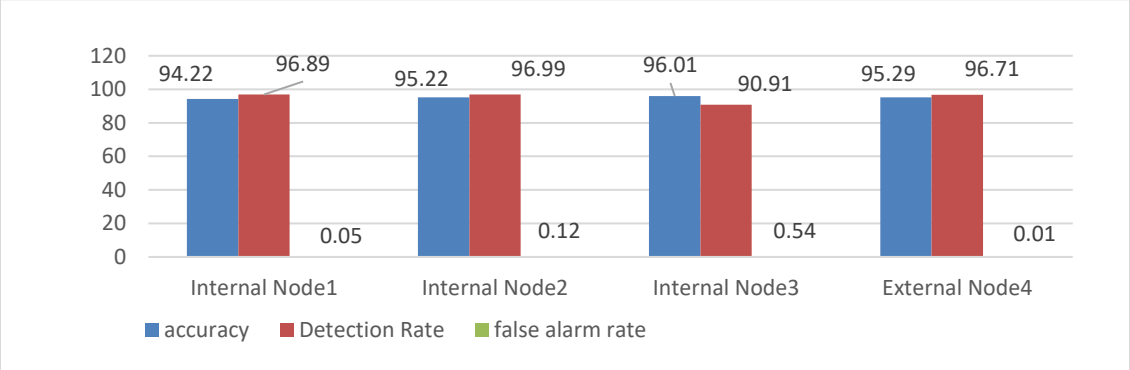

Figure 6: best results for DDOS attacks.

We note that the average attack detection time distributed by the nodes is estimated at (2.40) seconds, the average detection rate is (95.37%), the average accuracy of attack detection is (95.18%), and the average rate of false alarms is (False Alarm Rate) is estimated at (0.18%).

4.2 User to Root attack (Buffer Overflow)

The attacker from the subnet-2 attacks the host representing the IDS node with the Agents Manager located in the subnet-1 to make it out of service and make the second Analysis Agent from the subnet-2 perform the tasks of the Agent Manager. This ensures that the tasks are scheduled using the scheduling algorithm (adaptive behavior).
Table 7. depicts the attack.

Table 7: Buffer overflow attack

Node	Type of node	Type of traffic	Accuracy	Detection Rate	False Alarm Rate	Time(S)	Attack Session	Normal Session
Node1	Agents Manager	U2R	99.11	99.5	0.01	2.2	28	112
Node2	IDS	U2R	99.8	99.4	0.01	2.3	28	96
Node3	IDS	Normal	99.13	99.66	0	0.9	0	501
Node4	IDS	Normal	99.4	99.54	0	0.7	0	223
Node2	Agents Manager	Normal	99.2	99.16	0	0.8	0	579
Node3	IDS	Normal	99.9	99.3	0	1.2	0	601
Node4	IDS	Normal	99.7	99.8	0	0.5	0	312

.

4.3 Probe attack (Nmap Stealth Scan)

The attack is carried out from the subnet-2 on the server located in subnet-4. The attacker probes the ports to explore them. Table 8. depicts the attack.

Table 8: Nmap stealth Scan.

Node	Type of node	Type of traffic	Accuracy	Detection Rate	False Alarm Rate	Time(S)	Attack Session	Normal Session
Node2	Agents Manager	Probe	97.9	99.4	0.02	1.1	1000	88
Node3	IDS	Normal	99.13	99.66	0	0.8	0	501
Node4	IDS	Normal	99.4	99.54	0	0.5	0	71
Node2	Agents Manager	Normal	99.2	99.16	0	0.7	0	311
Node3	IDS	Normal	99.9	99.3	0	0.5	0	612
Node4	IDS	Probe	99.7	99.8	0.01	0.9	927	66

Table 9. shows a comparison with other techniques. Where the most important criteria used in the comparison that indicate the effectiveness of distributed intrusion detection systems have been adopted.

Table 9: Comparison with other DIDS techniques.

REF	Approach	Type of IDS		Data set	Detected attack	Number of Features	Effectiveness			Efficiency	Classification Type		F.T
		Single	DIDS and how many agents/units				DR %	ACC %	FAR %	Detection time (s)	Binary (0 or 1)	Multiclass	
[9]	Misuse detection and Migration agents between hosts in subnetworks and controlled by management servers.	*	Ok /5	*	Evasion Techniques Denial of Service Client Side Attacks IP Reputation Shell Codes Brute Force Fragmented Packets Test Rules	*	*	93	33	*	*	*	ok
[10]	Implementation of misuse detection, Central supervisor agent controls and manages	*	Ok /5	*	DOS, U2R, R2L	*	72 for Dos, 63.11 for U2R, 81 for R2L	*	11 for DOS, 18 U2R and 8.3	*	*	*	*

	MAS-DIDS Agents.								for R2L				
[11]	The central server works to extract features and publish incoming traffic to all nodes to detect anomalies, using a voting algorithm (weighted majority algorithm) to classify traffic.	*	Ok/7	NSL-KDD	DOS, U2R, R2L, Probe.	*	98	*	0.09	*	ok	*	*
[12]	Anomaly detection based on naïve bayes and random forest.	*	Ok/5	CIC_IDS001	DOS, Port Scan, Ping scan, Brute force	*	*	97	0.21	6.23	*	Ok	*
	random forest.							85	0.43	24.87			
[13]	Multi-agent based DIDS used ensemble data mining approach between (SVM,ANN and Random forest	*	Ok/4	KDD-CUP	DOS, U2R, R2L, Probe.	*	96.1	DOS 99.9 Probe 92.1 U2R 96.4 R2L 91.1	0.12	*	*	Ok	*
Our proposal	DIDS based on Anomaly detection using Transfer Learning for extract features by ANN and SVM for Classification.	*	Ok/2	NSL-KDD	DDOS	11	95.37	95.18	0.18	2.40	*	Ok	Ok
					DOS		99.7	99.1	0.01	1.9			
					Buffer over flow		99.4	99.8	0.01	2.3			
					Nmap stealth scan		99.8	99.7	0.01	0.9			

5. Conclusion

This work proposed a distributed intrusion detection system based on Anomaly Detection to detect attacks. The model was improved using hybrid machine learning ANN-SVM.
The system has been tested online, in a real network environment based on nodes. In each node, two agents are working collaboratively, to achieve a high speed of response. In addition, the data is analyzed locally within the detection node. This reduces the load on the network. The results of the analysis are transmitted using the ACL protocol between agents implemented in the platform JADE. The proposed system also can expand dynamically without affecting its work, by creating a detection node within any new sub-network that is added to the network topology. Agents Manager downtime has also been addressed, by applying an algorithm for scheduling and node prioritization to perform the tasks of the Agent Manager in that state (this is the solution to the problem of centralization at the node level). Within the tests carried out on the system, four major DOS attacks were used, and one of U2R, and one of the Probe attacks were achieved.
Future work includes multiple steps: testing the proposed approach on a new set of network attacks, such as Probe Attacks and U2R attacks; training and testing the proposed approach on new datasets; and development of a solution, that can address encrypted network traffic.

References

[1] N. Gulia, K. Solanki, S. Dalal, A. Dhankhar, O. Dahiya and N. Salman, "Intrusion Detection System using G-ABC with deep neural network in cloud environment," in *Hindawi scientific programmer*, vol. 2023, Apr. 2023, doi: 10.1155/2023/7210034.

[2] P. Vanin, T. Newe, L.L. Dhirani , E. O'Connell, D. O'Shea, B. Lee and M. Rao, "A Study of Network Intrusion Detection Systems UsingArtificial Intelligence/Machine Learning," in *Applied* sciences, vol. 12, Nov. 2022, doi: 10.3390/app122211752

[3] M. Ozlap, C. Karakuzu and A. Zengin, "Distributed intrusion Detection System: A Review," in *International Symposium on Innovative Technologies in Engineering and Science*, Nov. 2019, doi: 10.33793/acperpro.02.03.18.

[4] S. Othman, N. Alsohaybe, F. Ba-alwi and A. Zahary, "Survey on Intrusion Detection System Types," in *International Journal of Cyber-Security and Digital Forensics*, vol. 7, Oct. 2018.

[5] U. Akyazi and A. Uyar, "Distributed intrusion detection using mobile agents against DDoS attacks," in *IEEE Xplore*, *International Symposium on Computer and Information Sciences*, Oct. 2008, doi: 10.1109/ISCIS.2008.4717920.

[6] G. Vigna and C. Kruegel, "Host-based Intrusion Detection," in *UC Santa Barbara Computer science*, 2006.

[7] B. S. N. Murthy, K. Srinivas, S. Jena, A. Sandeep, M. Naidu, M. Ravi and K. Sudheer. "Network Intrusion Detection using Supervised Machine Learning Technique with Feature Selection," in *Mathematical Statistician and Engineering Applications (MSEA*), vol. 71, 2022, doi: 10.17762/msea.v71i4.1115.

[8] Y. Yu and N. Bain, "An Intrusion Detection Method Using Few-Shot Learning," in *IEEE Access*, vol. 8, pp. 49730-49740, Mar. 2020, doi: 0.1109/ACCESS.2020.2980136.

[9] Y. Mehmood, M. Shibli, A. Kanwal and R. Masood, "Distributed Intrusion Detection System using Mobile Agents in Cloud Computing Environment", in *IEEE Xplore, Conference on Information Assurance and Cyber* Security *(CIACS)*, Feb. 2016, doi: 10.1109/CIACS.2015.7395559.

[10] O. Achbarou, M. El Kiram,O. Bourkoukou and S. Elbouanani, "A New Distributed Intrusion Detection System Based on Multi-Agent System for Cloud Environment," in *International Journal of Communication Networks and Information Security (IJCNIS)*, vol. 10, pp. 526-533, Dec. 2018, doi: 10.17762/ijcnis.v10i3.3546.

[11] S. Khonde and U. Venugopal, "Hybrid Architecture for Distributed Intrusion Detection System, " in *international information and engineering technology association, Ingenierie des Systemes d'Information*.vol. 24, No. 1, pp. 19-28, Feb. 2019, doi: 10.18280/isi.240102.

[12] M. Idhammad, K, Afdel and M. Belouch, "Distributed Intrusion Detection System for Cloud Environmentsbased on Data Mining techniques," *in Science Direct, Procedia Computer Science*, vol. 127, pp. 35-41, 2018, doi: 10.1016/j.procs.2018.01.095.

[13] R. A.M, I. Ahmad and R. Khan, "An adaptive distributed intrusion detection system architecture using multi agents," in *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, No. 6, PP. 4951-4960, Dec. 2019, doi: 10.11591/ijece.v9i6.pp4951-4960.

[14] D. Roy, K.Sri.R. Murty and C.K. Mohan, "Feature Selection using Deep Neural Networks," in *IEEE Xplore, International Joint Conference on Neural Networks (IJCNN)*, Jul. 2015, doi: 10.1109/IJCNN.2015.7280626.

[15] S. Choudhary, N. Kesswani, "Analysis of KDD-Cup'99, NSL-KDD and UNSW-NB15 Datasetsusing Deep Learning in IoT," in *ScienceDirect, Procedia computer science, International Conference on Computational Intelligence and Data Science*, vol. 167,pp. 1561–1573, 2020, doi: 10.1016/j.procs.2020.03.367

[16] JADE Board. (2005). JADE Security Add-On GUIDE. Administrator's guide of the Security add-on, Version 28-February-2005, JADE 3.3

[17] A. Andalib and V. Vakili, "An Autonomous Intrusion Detection System Using Ensemble of Advanced Learners," in *IEEE Xplore, Iranian Conference on Electrical Engineering (ICEE)*, Nov. 2020, doi: 10.1109/ICEE50131.2020.9260808.