International Journal of Advanced Research in Engineering Technology & Science
ISSN: 2349-2819
Impact Factor: 7.10
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Email: editor@ijarets.org Volume-12, Issue-12 December – 2025 www.ijarets.org

ANALYSIS ON IDS BASED ON FUZZY LOGIC APPROACH AND BAGGING MODELS

Khwahish Indoria
Department of Computer Science and Engineering
B K Birla Institute of Engineering and Technology, Pilani (Raj) – India

Dr. Nimish Kumar (Professor and Head)
Department of Computer Science and Engineering
B K Birla Institute of Engineering and Technology, Pilani (Raj) – India

ABSTRACT

Cloud computing has evolved as a breakthrough technology in this age of digital transformation. It provides users with access to computer resources via the internet that is flexible, scalable, and available on demand. A growing number of organisations across a wide range of industries are turning to cloud infrastructure for the management of their data, services, and applications because of the cost-effectiveness, availability, and simplicity of deployment that cloud infrastructure offers. However, the same characteristics that make cloud computing appealing—shared resources, remote accessibility, and elasticity—also make it vulnerable to a broad variety of cyber attacks. A fuzzy logic approach is used to the data that has been pre-processed during the second phase of the research project. This technique is used to determine the link between the characteristics in order to discover anomalous behaviour. On the basis of the pre-processed data, the fuzzy logic approach generates a greater number of fuzzy rules; nevertheless, it is possible that some of these fuzzy rules are redundant. It is thus required to use an optimisation strategy in order to minimise the number of fuzzy rules in order to discover the effective fuzzy rules. Using the bagging technique, these fuzzy criteria are used to identify a distributed denial of service assault that is a TCP-SYN flood

Keywords: *IDS, Fuzzy, Logic, Approach, Bagging, Models*

INTRODUCTION

Cloud computing has evolved as a breakthrough technology in this age of digital transformation. It provides users with access to computer resources via the internet that is flexible, scalable, and available on demand. A growing number of organisations across a wide range of industries are turning to cloud infrastructure for the management of their data, services, and applications because of the cost-effectiveness, availability, and simplicity of deployment that cloud infrastructure offers. However, the same characteristics that make cloud computing appealing—shared resources, remote accessibility, and elasticity—also make it vulnerable to a broad variety of cyber attacks. The Distributed Denial of Service (DDoS) assault, more precisely the

Transmission Control Protocol (TCP) SYN flooding attack, is one of the most persistent and devastating of these attacks.

Its purpose is to deplete server resources and render services inaccessible to users who are authorised to use them. As the level of complexity and frequency of assaults of this kind continues to rise, the need for security systems that are resilient, intelligent, and adaptable becomes more important than it has ever been.

Cloud Computing

Throughout the course of its history, cloud computing has been defined and understood in a variety of ways by a wide range of scholars. According to the National Institute of Standards and Technology (NIST), cloud computing is defined as "a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources, such as networks, servers, storage, applications, and services that can be rapidly provisioned and released with minimal management effort or service provider interaction." This is the standard definition of cloud computing.

The cloud community is provided with three distinct kinds of services, which are referred to as Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS), as shown by this description. Private clouds, public clouds, hybrid clouds, and community clouds are the four categories that may be used to classify the various deployment methods for cloud computing.

In addition to providing the greatest degree of flexibility and administrative control of IT resources, Infrastructure as a Service (IaaS) provides computational capabilities on demand via a pay-as-you-go approach. Highly scalable, dynamic, and adaptable services are provided by the company.

For the purpose of providing cloud customers with automated administrative duties, the Infrastructure as a Service (IaaS) service model is developed with a Graphical User Interface (GUI) and an Application Programming Interface (API). Digital Ocean, Amazon Web Services (AWS), Microsoft Azure, Google Compute Engine (GCE), Rackspace, and Cisco Meta cloud are some of the service providers that are available. The Platform as a Service (PaaS) model was developed so that programmers could construct, test, operate, and maintain programs, as well as integrate them with online services and databases, without having to install the underlying infrastructures, such as hardware and operating systems.

Attacks in Cloud Environment

The exponential rise of cloud computing has led to an increase in the likelihood of assaults, and as a result, security has emerged as a major issue on a multinational scale. Cloud environments are characterised by a pay-per-use-on-demand approach, which allows users to share information technology resources via the use of the Internet.

Cloud computing, which is a kind of computing that is dependent on the Internet, is vulnerable to a variety of assaults that may cause the cloud community to be distressed. These attacks can interrupt and deny cloud-

based services, which are essential for key activities. It is a complicated environment that gives users a continual plethora of advantages to alter the organisations to store and share the data, apps, and workloads. Cloud computing has a big potential to boost productivity and to cut expenses, but it is also a difficult environment.

At the same time as the levels of utilisation of public clouds are quickly increasing, there are a great deal of new security dangers that are associated with them. Additionally, it always results in an increased body of sensitivity that is susceptible to being put in jeopardy. As a result of the sensitive nature of cloud computing, the upkeep of data becomes the most significant challenge. There are a lot of different sectors that are interested in uploading their massive amounts of data into public cloud storage. The sensitive information that is uploaded to the cloud is susceptible to a variety of security threats, including those pertaining to availability, integrity, and confidentiality.

Distributed Denial of Service (DDOS) Attack

Only one device connected to the Internet (one network connection) is required for a denial of service attack. A distributed denial of service attack is a subtype of a distributed denial of service attack. To overwhelm the online services and resources, a distributed denial of service attack (DDoS) is a wide category of assault that interrupts the usual traffic of a server that is the target of the attack.

A distributed denial of service assault (DDoS) is a malicious effort to interrupt the regular traffic of a targeted server by flooding a website with phoney Internet traffic.

This is accomplished by using a botnet, which is a collection of individual devices that are linked to the internet and are used to overload a website. A malicious assault is being launched with the intention of rendering a certain online service inaccessible to users for the length of the attack. Through the use of several hacked computer systems as sources of traffic, distributed denial of service assaults are targeted.

A distributed denial of service attack that is effective is an event that is extremely observable and has an effect on an entire user base of internet users. This is one of the reasons why hacktivists, cyber vandals, and extortionists choose to use it as their weapon of choice. It is possible for distributed denial of service assaults to occur in brief bursts or in recurring strikes; but, regardless of the nature of the attack, the effect on a website or company may endure for days, weeks, or even months, while the organisation works to recover. DDoS attacks may be incredibly detrimental to any organisation that operates online because of this. Distributed denial of service attacks may result in a number of negative outcomes, including the loss of income, the need for companies to spend a significant amount of money on compensations, and the harm to the long-term reputation of genuine users.

A distributed denial of service attack (DDoS) is one of the most popularly rising attacks for hacking. In this attack, hackers are able to flood an IP address with hundreds or thousands of messages, typically through the use of botnets or through a coordinated hacktivist effort.

REVIEW OF LITERATURE

Gong, C., Liu, J., Zhang, Q., Chen, H., & Gong, Z. (2010), The ability to share a single physical host, resources, and application across several users is made possible by virtualisation, which is a fundamental component of cloud computing. The architecture of virtual clouds is susceptible to distributed denial of service attacks, in particular SYN flood attacks, which deplete the server's resources and render it inaccessible to users who are authorised to use it. When it comes to cloud computing, the effect of this kind of assault may spread across a vast number of resources, which can result in losses that are twice as high. In this work, the SYN flood attack in virtual cloud is discussed, and it is shown how new information retrieved from the TCP/IP header may be used to identify attack techniques. For this reason, several machine learning techniques, including neural networks, naïve bases, decision trees, and k-means, are used in order to evaluate the performance evaluation. It has been shown via the findings that the machine learning algorithms have demonstrated an effective performance in the classification and detection of SYN flood attacks using extracted characteristics.

Ravi, Nagarathna & Shalinie, S. & Lal, Chhagan & Conti, Mauro. (2020), SDN, which stands for software-defined network, is a network architecture that creates a programmable network by separating the control plane and the data plane. The controller that is located in the control plane is responsible for running network modules and establishing rules for the packets that are to be sent in the switches that are located in the data plane. Even though it has a number of benefits, software-defined networking (SDN) might fail if the controller is overloaded with a large number of TCP SYN packets. It is possible to generate a SYN flood by resorting to malicious spoofing of IP or MAC addresses or by utilising flash crowd. The solutions that are currently available to combat SYN flood against the controller are not capable of handling MAC spoofing-based SYN flood in an effective manner, and these solutions are also unable to differentiate between malicious traffic and flash crowd-based traffic. We propose a unique method that we name AEGIS, which detects and mitigates SYN flood with regard to the controller in SDN. This mechanism is designed to overcome several constraints that are present in previous solutions. In the controller, AEGIS is running, and it performs frequent checks to determine whether or not there is a performance lag in the controller as a result of a continuing SYN flood. AEGIS considers the occurrence of a performance reduction to be a sign of a SYN flood, and it determines if the deterioration is the result of faked addresses or flash crowding. After the cause has been identified, the proper course of action to mitigate the problem is initiated. AEGIS is evaluated in both testbed and emulation environments, and the findings of the assessment are compared against systems that are considered to be state-of-the-art technology. It has been determined via the performance assessment of AEGIS that it is capable of identifying malicious SYN with a level of accuracy of 97.78%. Furthermore, when there is no SYN flood, it takes 0.0637 seconds for AEGIS to establish a successful TCP connection. This is 53.81% shorter than the time that the state-of-the-art solution takes, which is further evidence that AEGIS is a lightweight solution.

Wang, Haining & Zhang, Danlu. (2002), In order to identify SYN flooding assaults, we offer a technique that is both simple and reliable. We identify SYN flooding assaults at leaf routers, which are routers that link end sites to the Internet. This is done rather than monitoring the ongoing traffic at the front end (such

as a firewall or proxy) or on the target server itself. Our detection system is impervious to flooding assaults due to its statelessness and minimal computing cost, which contribute to the mechanism's simplicity. Additionally, the mechanism itself is resistant to flooding attacks. The protocol behaviour of TCP SYN-FIN (RST) pairings serves as the foundation for our detection approach, which is an example of the Sequential Change Point Detection . In order to render the detection mechanism insensitive to the site and access pattern, a non-parametric Cumulative Sum (CUSUM) technique is used. This approach makes the detection mechanism far more relevant to a wider range of situations and makes its implementation significantly simpler. By using trace-driven simulations, the effectiveness of this detection system is shown and proven. Based on the findings of the assessment, it can be concluded that the detection mechanism has a low detection latency and very high detection accuracy. Furthermore, because of its close proximity to the sources of flooding, our technique not only triggers alerts when it detects continuous SYN flooding assaults, but it also discloses the location of the flooding sources without the need to resort to costly IP traceback.

John, Joshua & Okonkwo, Obikwelu & Akawuku, Godspower & Onyagu, Chika. (2023), Despite the fact that cloud computing is often regarded as their most prevalent technology. Despite the fact that it provides a number of benefits, end users continue to face a number of difficulties, particularly those that are associated with its security flaws. These include that it is susceptible to malicious assaults, such as Distributed Denial of Service attacks (DDoS), which block users from accessing the internet. The purpose of this article is to provide a knowledge of distributed denial of service assaults (DDoS) in cloud computing environments and to provide techniques to mitigate these attacks. This will be accomplished by using the many solutions that have been proposed to detect and prevent DDoS attacks from causing damage to network communication. The following are some keywords: Zombies, Cloudflare, Firewall, Gateway, and Webserver.

Fichera, Silvia & Galluccio, Laura & Grancagnolo, Salvatore & Morabito, Giacomo & Palazzo, Sergio. (2015), Distributed denial of service attacks, often known as DDoS assaults, are a sort of attack that are typically conducted out against web servers. TCP SYNFLOOD attacks are one. It is necessary for TCP SYNFLOOD to use the standard TCP Three-Way Handshake protocol in order to consume resources on the server that is being targeted. It is possible to block the server's resources and render the server unusable by using this method. In order to do this, the attacker transmits a number of bogus SYN packets, making it seem as if it is attempting to establish a number of TCP connections; however, it does not completely complete the Three-Way Handshake. Blocking the resources of the server that is being attacked in this manner is completely pointless. The use of firewalls and intrusion detection systems has proven an effective method for defending against assaults of this kind in conventional networks. Nevertheless, these remedies are ineffective since they are susceptible to being violated. When it comes to addressing TCP SYNFLOOD assaults, Software Defined Networks (SDNs) provide additional characteristics such as network programmability that significantly improve the effectiveness of solutions.

OBJECTIVES OF THE STUDY

1. To study on Distributed Denial of Service (DDOS) Attack
2. To study on Attacks in Cloud Environment

RESEARCH METHOD

Design of IDS Based On Fuzzy Logic Approach and Bagging Models

The identification of attacks in cloud networks is a difficult responsibility. Prior to arriving at a conclusion about the assault, it is necessary to take into account a great deal of related information.

The decision-making process requires more time because of this reason, and it is not possible to find the best choice just employing attack behaviours. In order to arrive at a sound judgement in a much shorter amount of time, the fuzzy logic approach is used for the purpose of making decisions quickly. The findings of the principal component analysis (PCA) based features from the first phase are used for the implementation of the fuzzy logic approach.

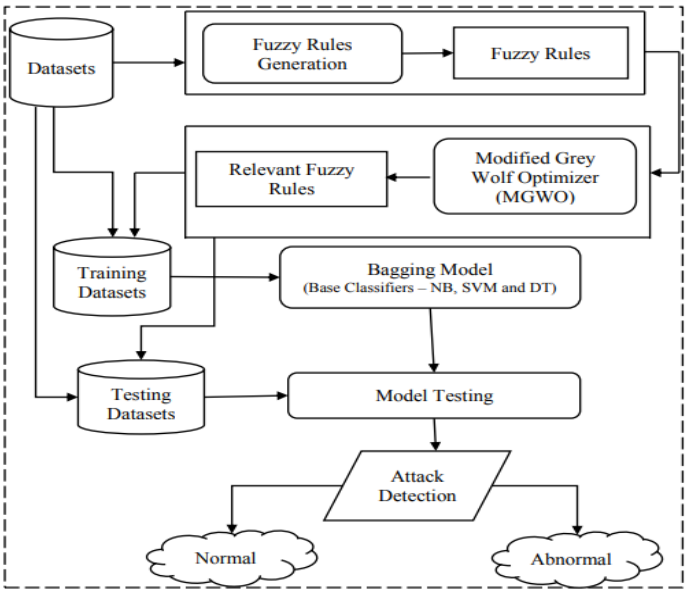

Figure 1 Architectural Design of IDS Using Fuzzy Logic and Bagging Approach

A fuzzy logic approach is used to the data that has been pre-processed during the second phase of the research project. This technique is used to determine the link between the characteristics in order to discover anomalous behaviour. On the basis of the pre-processed data, the fuzzy logic approach generates a greater number of fuzzy rules; nevertheless, it is possible that some of these fuzzy rules are redundant. It is thus required to use an optimisation strategy in order to minimise the number of fuzzy rules in order to discover the effective fuzzy rules. Using the bagging technique, these fuzzy criteria are used to identify a distributed denial of service assault that is a TCP-SYN flood. Based on fuzzy logic and bagging, the architectural architecture of the intrusion detection system is shown in Figure 1

Evaluation of MGWO on Benchmark Test Functions

R is the tool that is used to create the suggested MGWO. The parameter settings that are included in Table 1 are used to carry out the execution. With the number of population being around twenty, which is designated as the number of wolves, and the maximum iteration being approximately one hundred, the input size may be structured either row wise or column wise. The issue criteria may be used to determine how these parameter settings should be adjusted.

Table 1 MGWO Parameter Settings

Parameter Settings	Value in Numbers
numVar(The number variables are determined with positive integer.)	50(it may also called as data size)
num Population	20
maxIter	100

A list of the benchmark test functions that were used in the process of MGWO's performance evaluation can be found in Table 1. In terms of criteria, they are classified as mixed, such as uni-model or multi-model. Although the ranges of each benchmark test function are distinct from one another, the dimensions remain the same.

Table 2 Benchmark Test Functions of Optimization Algorithm

Function Label	Test Function	Nature	Dimension	Range
F1	Step	Uni-Modal	30	[-1.28,1.28]
F2	Rosenbrock	Uni-Modal	30	[-30,30]
F3	Schwefel	Uni-Modal	30	[-10,10]
F4	Sphere	Uni-Modal	30	[-100,100]
F5	Schaffer	Uni-Modal	30	[-100,100]
F6	Ackley	Multi-Modal	30	[-32, 32]
F7	Alpine	Multi-Modal	30	[-50.0, 50.0]
F8	Griewank	Multi-Modal	30	[-600, 600]
F9	Trid	Multi-Modal	30	[-100, 100]
F10	Rastrigin	Multi-Modal	30	[-5.12, 5.12]
F11	Salomon	Multi-Modal	30	[-100,100]

Table 3 Statistical Results of MGWO

Function Label	Test Function	MGWO		GWO		PSO		CS	
		Mean	STD	Mean	STD	Mean	STD	Mean	STD
F1	Step	0.012	0.014	0.005	0.005	0.113	0.096	3.939	2.575
F2	Rosenbrock	0.040	0.134	0.012	0.026	0.437	0.318	3.990	2.158
F3	Schwefel	4.906	0.968	4.341	1.710	5.679	1.418	5.214	2.446
F4	Sphere	0.016	0.029	0.007	0.007	0.216	0.164	4.007	2.888
F5	Schaffer	2.349	3.210	2.989	3.007	2.301	2.383	4.077	2.404
F6	Ackley	0.002	0.001	0.016	0.029	0.410	0.444	4.140	2.810
F7	Alpine	0.001	0.001	0.003	0.003	1.411	1.549	3.540	3.026
F8	Griewank	0.175	0.178	0.037	0.053	0.226	0.233	3.399	2.281
F9	Trid	4.702	3.761	3.021	3.188	9.773	0.966	3.631	2.459
F10	Rastrigin	0.478	0.698	0.196	0.350	1.507	0.951	4.165	2.753
F11	Salomon	0.267	0.313	0.772	0.660	0.526	0.410	4.794	2.590

In order to evaluate the performance of MGWO, benchmark test functions are used, and the results of these tests are shown in Table 3. Comparative analysis is performed on the performance of MGWO, GWO, PSO, and CS. Due to the fact that the MGWO approach in question performs very well on benchmark test functions, it has been included into IDS for the purpose of fuzzy rule reduction.

RESULT

Design of IDS Based On Fuzzy Logic Approach and Bagging Models

In order to carry out the IDS framework for the detection of a TCP-SYN flood distributed denial of service attack that is taking place in a cloud environment, a bagging model is used during the second phase of the process. It is possible to acquire the input for the bagging model by considering the outcome of the combination of features that has been optimised via the use of fuzzy rules. It has been designed using the R platform, and the configuration of the system consists of a 64-bit Windows operating system, a central processing unit (CPU) from Intel Core i3, and four gigabytes of random access memory (RAM).

1. Performance Evaluation Measures

In each dataset, the training section accounts for 75% of the total, while the testing portion accounts for 25%. It is necessary to carry out this analysis on every dataset. In order to evaluate the performance of the

model and establish whether or not it is successful, evaluation measures like as accuracy, precision, recall, specificity, and f-measure are used.

Table 4 Confusion Matrix Table–Bagging Models

Dataset / Parameter	Public Dataset						Private Dataset		
	NSL-KDD			CIC-DDoS2019					
	NB	SVM	DT	NB	SVM	DT	NB	SVM	DT
TP	9953	10684	10326	10129	11327	13707	12607	11721	15321
FP	458	353	487	967	528	89	2054	913	1246
TN	4396	4036	4164	4187	3276	1132	6033	7387	4134
FN	735	469	565	829	981	1184	368	1041	361
Total Number of Records	15542			16112			21062		

Table 4 presents the results of the examination into the accuracy and performance of the bagging model that was carried out. A value of 0.9232 was obtained by the model via the use of bagging NB, a value of 0.9471 was obtained through bagging SVM, and a value of 0.9323 was obtained through bagging DT. Following the results, bagging SVM shown greater performance in terms of the attack detection rate. This was the result of the findings. This graph, which can be seen in Figure 4.6, provides an illustration of the comparison of the accuracy performance.

Table 5 Accuracy Comparison of Bagging Models

	Bagging NB	Bagging SVM	Bagging DT
NSL-KDD	**0.9232**	**0.9471**	**0.9323**
CIC-DDoS2019	0.8885	0.9063	0.9210
Private	0.8850	0.9072	0.9237

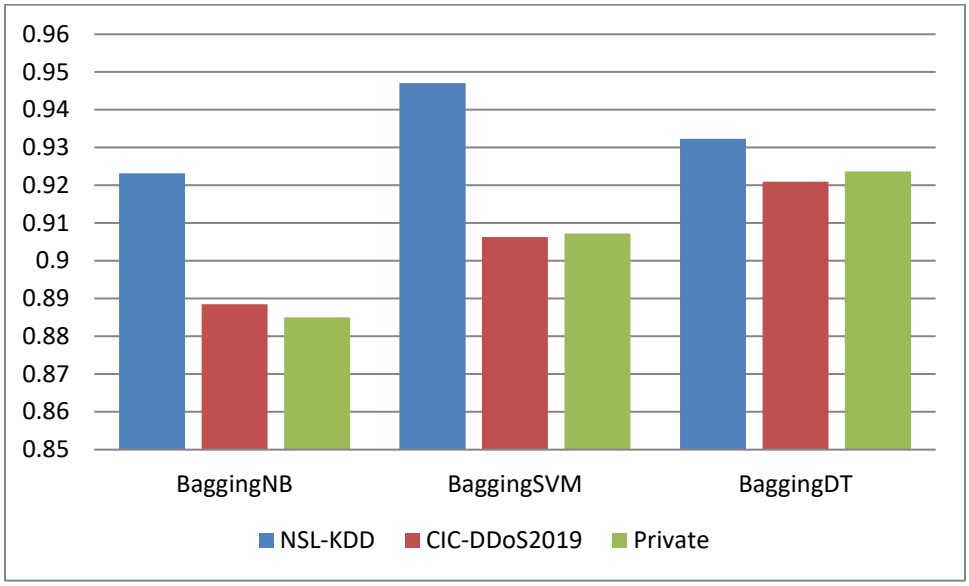

Figure 2 Accuracy Comparison of Bagging Models

The results of the precision performance tests performed on the bagging models that were investigated are shown in tabular form in Table 5 Bagging NB models have a precise detection of attack of 0.9560, whereas bagging SVM models have a calculation of 0.9680 and bagging DT models have a calculation of 0.9935. The bagging NB model has the highest precision detection of attack. When compared to the other models, the bagging DT model performed very well in terms of accuracy. Figure 3 is an illustration that displays the comparison of the two systems' exact performance of the two systems.

Table 6 Precision Comparison of Bagging Models

	Bagging NB	**Bagging SVM**	**Bagging DT**
NSL-KDD	**0.9560**	**0.9680**	0.9550
CIC-DDoS2019	0.9129	0.9555	**0.9935**
Private	0.8599	0.9277	0.9248

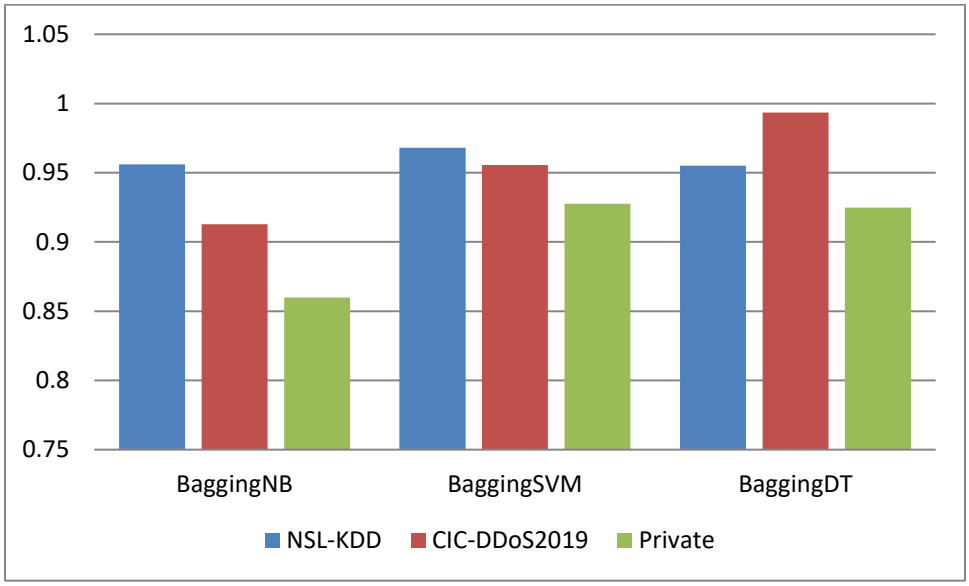

Figure 3 Precision Comparison of Bagging Models

The results of the recall measure carried out using the bagging model are shown in Table 7. Bagging NB has a recall value of 0.9716, bagging SVM has a recall value of 0.9579, and bagging DT has a recall value of 0.9770. The model effectively identifies the attack. In comparison to the other models, bagging NB and bagging DT fared the best. Figure 4 illustrates the comparison of recall measures for the various versions of the bagging model.

Table 7 Recall Comparison of Bagging Models

	Bagging NB	**Bagging SVM**	**Bagging DT**
NSL-KDD	0.9312	**0.9579**	0.9481
CIC-DDoS2019	0.9243	0.9203	0.9205
Private	**0.9716**	0.9184	**0.9770**

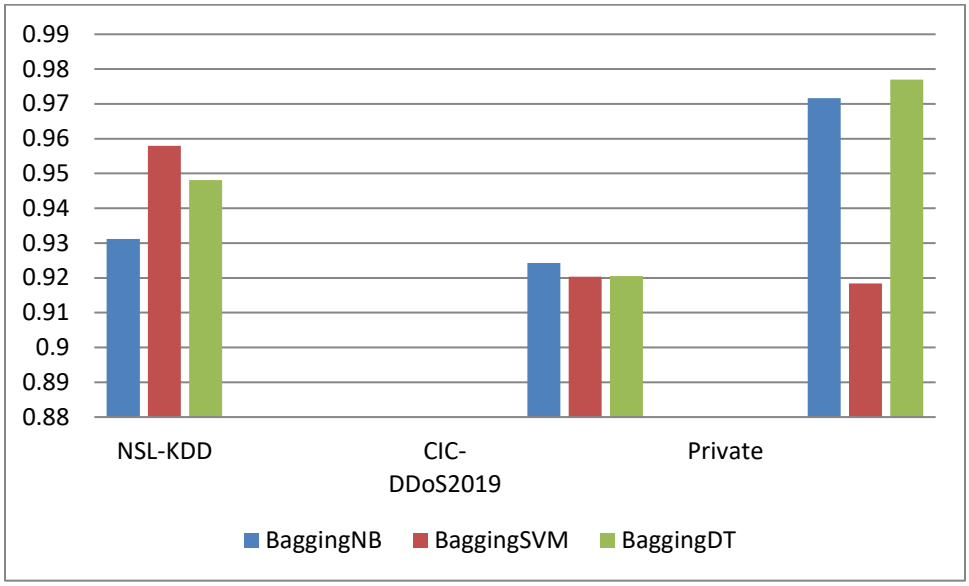

Figure 4: Recall Comparison of Bagging Models

A comparison of the specificity measurements that are linked with the bagging models is shown in Table 8. The model indicates that the specificity value for bagging NB is 0.9056, the specificity value for bagging SVM is 0.9196, and the specificity value for bagging DT is 0.9271. The bagging DT model is successful in obtaining the highest possible degree of efficacy when compared to the specificity of other models. A graphical depiction of the comparison of specificity metrics for the various bagging models is shown in Figure 5 [148].

Table 8 Specificity Comparison of Bagging Models

	Bagging NB	**Bagging SVM**	**Bagging DT**
NSL-KDD	**0.9056**	**0.9196**	0.8953
CIC-DDoS2019	0.8124	0.8612	**0.9271**
Private	0.7460	0.8900	0.7684

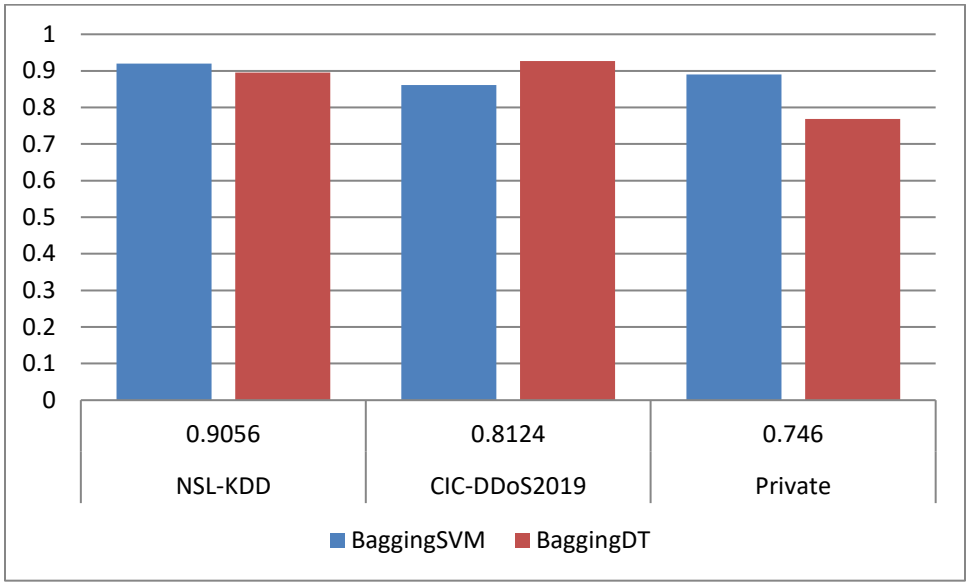

Figure 5 Specificity Comparison of Bagging Models

CONCLUSION

The cloud intrusion detection system (IDS) is built in three steps to recognise a TCP-SYN flood DDoS attack on three separate datasets: The bagging DT model is successful in obtaining the highest possible degree of efficacy when compared to the specificity of other models. A graphical depiction of the comparison of specificity metrics for the various bagging model A fuzzy logic approach is used to the data that has been pre-processed during the second phase of the research project. This technique is used to determine the link between the characteristics in order to discover anomalous behaviour. On the basis of the pre-processed data, the fuzzy logic approach generates a greater number of fuzzy rules; nevertheless, it is possible that some of these fuzzy rules are redundant. It is thus required to use an optimisation strategy in order to minimise the number of fuzzy rules in order to discover the effective fuzzy rules

REFERENCES

[1] Albahar, M. A., Thanoon, M. I., & Albahr, A. A. (2021). An ensemble model for detecting coronavirus disease-19 from chest X-ray images. *International Journal of Medical Research & Health Sciences, 10*(8), 1–12. https://www.ijmrhs.com/medical-research/an-ensemble-model-for-detecting-coronavirus-disease19-from-chest-xray-images.pdf

[2] Al-Hawawreh, M. S. (2017). SYN flood attack detection in cloud environment based on TCP/IP header statistical features. In *Proceedings of the 2017 8th International Conference on Information Technology (ICIT)* (pp. 107–112). IEEE. https://doi.org/10.1109/ICITECH.2017.8080006

[3] Anonymous. (2014). A multicriterion fuzzy classification method with greedy attribute selection for anomaly-based intrusion detection. *Procedia Computer Science, 34*, 55–62. https://www.researchgate.net/publication/264827714

[4] Berguiga, A., & Harchay, A. (2021). An IoT-based intrusion detection system approach for TCP SYN attacks. *Computers, Materials & Continua, 71,* 3839–3851. https://doi.org/10.32604/cmc.2022.023399

[5] Bogdanoski, M., Toshevski, A., Bogatinov, D., & Bogdanoski, M. (2016). A novel approach for mitigating the effects of the TCP SYN flood DDoS attacks. *World Journal of Modelling and Simulation, 12*(3), 217–230. https://www.researchgate.net/publication/307919097_A_novel_approach_for_mitigating_the_effects_of_the_TCP_SYN_flood_DDoS_attacks

[6] Fernández, A., & Herrera, F. (2016). Evolutionary fuzzy systems: A case study in imbalanced classification. In *Fuzzy logic and information fusion* (Studies in Fuzziness and Soft Computing, Vol. 339, pp. 269–290). Springer. https://sci2s.ugr.es/sites/default/files/ficherosPublicaciones/2085_2016-Fernandez-EFS_imbalanced.pdf

[7] Fichera, S., Galluccio, L., Grancagnolo, S., Morabito, G., & Palazzo, S. (2015). OPERETTA: An OPEnflow based REmedy to mitigate TCP SYN flood attacks against web servers. *Computer Networks, 92,* 89–100. https://doi.org/10.1016/j.comnet.2015.08.038

[8] Gong, C., Liu, J., Zhang, Q., Chen, H., & Gong, Z. (2010). The characteristics of cloud computing. In *2010 39th International Conference on Parallel Processing Workshops (ICPPW)* (pp. 275–279). IEEE. https://doi.org/10.1109/ICPPW.2010.45

[9] Govindarajan, M., & Chandrasekaran, R. M. (2012). Intrusion detection using an ensemble of classification methods. In *Proceedings of the World Congress on Engineering and Computer Science (WCECS 2012), Vol. I* (pp. 459–464). IAENG. https://www.iaeng.org/publication/WCECS2012/WCECS2012_pp459-464.pdf

[10] IBM Support. (2024). SYN flood attack—Detection and prevention. *IBM Knowledge Center*. https://www.ibm.com/support/pages/syn-flood-attack-detection-and-prevention

[11] John, J., Okonkwo, O., Akawuku, G., & Onyagu, C. (2023). Understanding and mitigating cloud computing security attacks: A case of DDoS. *Newport International Journal of Scientific and Experimental Sciences, 4,* 25–35. https://doi.org/10.59298/NIJSES/2023/10.4.1000

[12] Kocher, G., & Kumar, G. (2021). Analysis of machine learning algorithms with feature selection for intrusion detection using UNSW-NB15 dataset. *Maharaja Ranjit Singh Punjab Technical University, Department of Computational Sciences.* https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3784406

[13] Kumar, N., & Sharma, S. (2013). Study of intrusion detection system for DDoS attacks in cloud computing. In *2013 IFIP Wireless and Optical Communications Conference (WOCN)* (pp. 1–5). IEEE. https://doi.org/10.1109/WOCN.2013.6616255

[14] Kumarasamy, S., & Gowrishankar, A. (2012). An active defense mechanism for TCP SYN flooding attacks. *arXiv preprint.* https://doi.org/10.48550/arXiv.1201.2103

[15] Pillai, P. M., & Krishnan, N. (2013). Layered approach for intrusion detection systems based on genetic algorithm. *International Journal of Computer Applications, 100*(3), 17–22. https://www.researchgate.net/publication/259849351

[16] Radjaa, B., Bensaid, N., Abba Ari, A. A., Maglaras, L., Saidi, H., Lwahhab, A. M. A., & Benfriha, S. (2024). Toward a real-time TCP SYN flood DDoS mitigation using adaptive neuro-fuzzy classifier and SDN assistance in fog computing. *Security and Communication Networks, 2024*, Article 6651584. https://doi.org/10.1155/2024/6651584

[17] Ramadhan, G., Kurniawan, Y., & Kim, C.-S. (2016). Design of TCP SYN flood DDoS attack detection using artificial immune systems. In *Proceedings of the 6th International Conference on System Engineering and Technology (ICSEngT)* (pp. 72–76). IEEE. https://doi.org/10.1109/ICSEngT.2016.7849626

[18] Ravi, N., Shalinie, S., Lal, C., & Conti, M. (2020). AEGIS: Detection and mitigation of TCP SYN flood on SDN controller. *IEEE Transactions on Network and Service Management, 17*(4), 2527–2540. https://doi.org/10.1109/TNSM.2020.3037124

[19] Saidi, H., Labraoui, N., Abba Ari, A. A., & Bouida, D. (2020). Remote health monitoring system of elderly based on fog-to-cloud (F2C) computing. In *Proceedings of the 2020 International Conference on Intelligent Systems and Computer Vision (ISCV)* (pp. 1–7). IEEE. https://www.researchgate.net/publication/345426355_Remote_health_monitoring_system_of_elderly_based_on_Fog_to_Cloud_F2C_computing

[20] Saranya, T., Sridevi, S., Deisy, C., Chung, T. D., & Khan, M. K. A. A. (2020). Performance analysis of machine learning algorithms in intrusion detection system: A review. *Procedia Computer Science, 171*, 1251–1260. https://doi.org/10.1016/j.procs.2020.04.133

[21] Shah, S. Q. A., Khan, F. Z., & Ahmad, M. (2021). Mitigating TCP SYN flooding based EDOS attack in cloud computing environment using binomial distribution in SDN. *Computer Communications, 182*, 236–246. https://doi.org/10.1016/j.comcom.2021.11.008

[22] Tama, A. B., & Lim, S. (2021). Ensemble learning for intrusion detection systems: A systematic mapping study and cross-benchmark evaluation. *Computer Science Review, 39*, 100357. https://www.researchgate.net/publication/348049939

[23] Wikipedia. (2025). *SYN cookies.* In *Wikipedia.* https://en.wikipedia.org/wiki/SYN_cookies

[24] Wikipedia. (2025). *SYN flood.* In *Wikipedia.* https://en.wikipedia.org/wiki/SYN_flood

[25] Zhang, Z., Kong, S., Yang, A., & Xiao, T. (2024). A network intrusion detection method based on bagging ensemble. *Journal of Computer Networks, 16*(7), 850. https://www.researchgate.net/publication/382103122