International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Securing the Internet of Things: A Machine Learning Approach to Mitigate DoS Threats with an Intrusion Detection System

Saumya Mishra [a,1], Manoj Kumar [b,2] , Aditi Paul [c,3], Somnath Sinha [d,4,*]

Submitted: 07/02/2024 **Revised**: 15/03/2024 **Accepted**: 21/03/2024

Abstract: This study addresses the threat of Denial of Service (DoS) attacks within the Internet of Things (IoT) and introduces a Hybrid Intrusion Detection System (IDS) designed for detecting Cross-Layer DoS assaults. Comparative analysis with a single IDS reveals a substantial reduction in false positive rates. The Hybrid IDS integrates various machine learning algorithms to prevent overfitting or underfitting, functioning in two stages—Anomaly detection and Signature detection. The initial stage (Anomaly Detection) produces an Output of First Stage which becomes input to the Second Stage (Signature Detection). The Output of the Second Stage gives the final attack classes. Notably, the study creates an adapted dataset by simulating multiple assault environment in the NetSim Simulator, emphasizing the concurrent selection of the best feature set and critical feature using an innovative technique. Additionally, the research includes a comparative analysis of testing datasets under varying attacker nodes, network nodes, and processing time efficiency scenarios. This further validates the proposed Hybrid IDS's effectiveness in mitigating DoS attacks in the IoT.

Keywords: *Internet of Things, Cross-Layer IDS, DoS attacks, Hybrid IDS, Machine Learning*

1.Introduction

The pervasive rise of the Internet of Things (IoT) has permeated virtually every facet of technological progress, spanning personal to professional domains. The surge of smart applications, encompassing homes, offices, education, health, transportation, food, clothing, wearables, entertainment, and gadgets, owes its existence to the unification of heterogeneity under the umbrella of IoT. However, this technological leap forward brings with it the imperative need to safeguard connected devices from internet-based attacks. Securing a multitude of heterogeneous devices poses a complex challenge, requiring a robust and secure backbone to thwart potential threats [1].

Complicating matters, IoT protocols are specifically crafted for low-power lossy networks, limiting their capacity to accommodate extensive security algorithms with heavy overhead. The diversity within the network further complicates the creation of a singular security framework. As time progresses, the security threats associated with IoT have become a focal point for researchers. Some attacks, rather than exploiting keys or algorithms, unleash destructive Denial of Service (DoS) attacks that can incapacitate an entire communication system, resulting in substantial time and financial losses, potentially leading to economic disasters for nations. DoS attacks exhibit at multi-layers of IoT, such as Flooding attacks with respect to Network and RPL layers, necessitating the need to address multiple-layer attacks simultaneously to avoid overwriting detection methods [2].

The era of research demonstrates how commonplace machine learning-based algorithms, soft computing methods, sampling, and deep learning are extensively used in intrusion detection systems (IDS) for Internet of Things security. These IDSs primarily emphasis over detecting bizarre behaviour within an IoT network and triggering spooks. Having said that, the implementation of detection system can be susceptible to false alarm if enforced rigorously. Real-time functionality of a detection system presents another challenge, requiring the consideration of all conceivable traces, making the creation of a solely hosted IDS challenging in networks requiring higher detection accuracy. Hybrid IDS, emerging as a novel research avenue, aims to prevail over this challenge by incorporating manifold layers of filtration to comprehend the genuine comportment of a network.

To resolve this gap current study proposes a Cross-layer Hybrid IDS for recognition of DoS attacks in multiple layers of IoT and utilizing machine learning-based tools to minimise false alarm rate. Machine learning algorithms offer the advantage of learning from extensive datasets, obviating the need for resizing attack traces. Additionally, certain machine learning algorithms exhibit strong predictive capabilities, especially in the presence of high

[a,c] Banasthali Vidyapith, Vanasthali Rd, Aliyabad, Radha Kishnpura, Rajasthan 304022, India
[b]Indraprastha College for Women, Delhi University, Delhi, India
[d] CHRIST (Deemed to be University), Bangalore, Karnataka-560029, India
[1] saumyamishra2709@gmail.com; [2]manojkumar@ip.du.ac.in; [3]aditi23.mca@gmail.com; [4]ssin.mca@gmail.com;
** corresponding author*

variances in datasets. Acknowledging the pitfalls of relying on a single machine learning tool, the proposed approach adopts an ensemble approach, integrating multiple machine learning tools to formulate a robust attack detection model. The efficiency of this proposed model is further analysed through a comparative assessment, considering both balanced and imbalanced test datasets.

2. Literature Review

The literature review presents a comprehensive overview of recent advancements in the field of Intrusion Detection Systems (IDS) with a focus on Internet of Things (IoT) networks. The evaluated papers show the progress and difficulties in protecting IoT environments from cyber-attacks. They include a wide range of techniques, strategies, and applications.

Saranya and Valarmathi [3] conducted recent research in which they thoroughly investigated many cross-layer methods based on machine learning techniques that have previously been proposed to handle problems and challenges arising from the multiplicity of IoT. The primary concerns—scalability, interoperability, security, privacy, mobility, and energy consumption—are also discussed and examined.

An anomaly-based IDS for IoT was proposed by Bajaj et al. [4] using a stack-ensemble model that included DT, LR, SVM, and KNN. The ensemble method notably addressed single-layer and cross-layer assaults, exhibiting increased accuracy without overfitting. The paper highlights how difficult it may be to choose base models that are suitable for a variety of attack datasets and recommends future research on hybrid IDS architecture for improved detection capabilities.

BoT-IoT and KDD Cup 1999 datasets were exploited by Nimbalkar and Kshirsagar [5] to concentrate on feature selection for IDS. Their system achieved high accuracy and detection rates for DDoS and DoS attacks. The paper highlights the use of bio-inspired algorithms for optimizing feature selection and suggests extending the approach to identify optimal features for a broader range of attacks.

Anthi et al. [6] introduced a novel three-layer IDS architecture targeting IoT devices. The system demonstrated effective classification of normal behavior, detection of wireless attacks, and classification of deployed attacks. Evaluation results on real IoT device data revealed high F-measures for each core function, showcasing the proposed architecture's ability to distinguish between benign and malicious activities.

Aliya and Aiman [7] conducted a comprehensive survey evaluating existing IDS approaches for IoTs. The study identified advantages and disadvantages of current IDS methods and emphasized the need for real-time evaluation, specifically tailored for IoT environments. The study urges the creation of intelligent intrusion detection systems (IDS) that can adjust to shifting network circumstances in IPv6-connected Internet of Things.

To construct a Network Intrusion Detection System (NIDS) for Internet of Things networks, Moustafa et al. [8] investigated feature extraction from TCP/IP protocols. Accuracy, detection rate, and processing time were all improved above previous methods by their ensemble approach, which combined AdaBoost with DT, NB, and ANN. Characterising valid and suspicious events in IoT network data is possible using the features that have been suggested.

A two-stage intrusion detection system, proposed by Amouri et al. [9], infers the nodes' position by analysing network behaviour and the gathered CCI (Correctly Classified Instances). The system showed encouraging results in differentiating between malicious and regular nodes, despite constraints in the unpredictability of the input. The suggested method can be implemented and offer efficient intrusion detection in situations when node data access is restricted.

The work of [10] presents a two-level detection technique using decision-tree-based classifiers and an algorithmic super node, addressing the difficulties of mobile ad hoc networks (MANETs). Even in dynamic situations with limited data availability, the suggested method effectively identifies rogue nodes. The findings also apply to wireless sensor networks (WSNs), providing a unique intrusion detection system for these kinds of situations.

A cross-layer IDS based on neural networks was described by Canbalaban and Sen [11] for RPL-based IoT networks. With the use of link-layer characteristics, the proposed IDS achieves excellent detection rates and minimal false positives for specific RPL attacks using binary and multiclass categorization. Being the first cross-layer intrusion detection system in RPL, this study is notable for revealing the importance of link-layer properties in intrusion detection.

For real-time control systems, Kwon et al. [12] presented a hybrid anomaly detection technique that combines behavior-based and signature-based methods. The approach, which combined a CAE (Composite Autoencoder) with statistical filtering, performed better than behavior-based detection alone. The findings point to possible applications that go beyond water treatment for a variety of Industrial Control Systems (ICSs), including enhanced detection precision and faster processing.

A Hybrid Intrusion Detection System (HIDS) for the Internet of Things was introduced by Khraisat et al. [13]. It combines a One Class Support Vector Machine with a C5 classifier. Comparing the proposed HIDS to established SIDS and AIDS procedures, the former showed reduced false positive rates and greater detection rates. Because HIDS is ensemble-based, it can identify known as well as zero-day attacks with high accuracy, protecting IoT settings from a variety of threats.

IoT-Sentry, a cross-layer intrusion detection system optimised for standardised IoT networks, was introduced by Malik et al. [14]. Detecting five distinct assaults with no additional overhead was accomplished by the system using a unique cross-layer IoT dataset. The study makes a public dataset available for assessing intrusion detection systems in Internet of Things situations, in addition to its contribution to intrusion detection.

NetFlow-based feature sets were proposed by Sarhan et al. [15] as a solution to the problem of standardising feature sets for NIDS. The study showed that the bigger 43-feature set might perform better than proprietary feature sets by comparing two versions with varying feature counts. To support more thorough assessments of ML-based traffic classifiers, the suggested NetFlow-based feature sets are meant to enable equitable comparisons among various NIDS datasets.

Sinha et.al [16] proposed a lightweight IDS for detection of Sybil attack using Fuzzy-Neural Network based approach. The system can achieve an accuracy of up to 100% with 0% false positive. The authors prove the system to be lightweight and can be implemented in distributed manner.

3.Proposed Model

In this study a two-stage Intrusion Detection System (IDS) is described along with performance analysis of the proposed cross-layer attack detection method.

The proposed method is a hybrid of anomaly and signature-based detection approaches. The first stage is a binary classification model comprised of Decision Tree (DT), K-Nearest Neighbour (KNN), and Support Vector Machine (SVM) using an anomaly-based intrusion detection system (AIDS). When compared to individual models, the stack ensemble integration of these models, as shown in Fig. 1, substantially improves detection accuracy during testing.

The second stage aims to classify the attack samples (from first stage) with improve the accuracy using a signature-based IDS. Five machine learning tools are highlighted here: KNN, Gradient Boosting (GB), Decision Tree (DT), Random Forest (RF), and Logistic Regression (LR) for increased accuracy.

In the first stage, anomaly detection(attack/non-attack) system detects the occurrence of any anomaly or attack using the binary classification and produces an output as attack or non-attack labels. This output is then passed into signature detection in the second stage, producing the classes of attacks using multiclass classification models. The percentage accuracy of Output Stage-1 and Output stage-2 serves as a vital metric, reflecting the efficiency of the proposed model (Fig. 1).

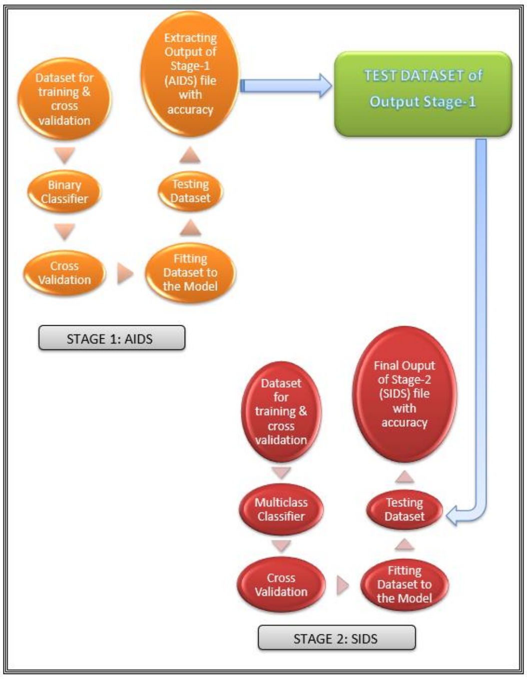

Dataset for training & cross validation
Binary Classifier
Cross Validation
Fitting Dataset to the Model
Testing Dataset
Extracting Output of Stage-1 (AIDS) file with accuracy
TEST DATASET of Output Stage-1
STAGE 1: AIDS
Dataset for training & cross validation
Multiclass Classifier
Cross Validation
Fitting Dataset to the Model
Testing Dataset
Final Ouput of Stage-2 (SIDS) file with accuracy
STAGE 2: SIDS

Fig.1. Ensemble-Based Two-Stage Intrusion Detection System Model

The proposed model (Fig. 1) encompasses following steps which are designing attacks, generating traces, data cleaning, merging datasets, data transformation, and feature selection using the Pearson correlation coefficient technique. A compact view of the steps is represented in Fig. 2.

Step 1: Attack Design and Trace Generation

Utilizing NetSim Simulator for Cross-Layer DoS Attacks: Generating cross-layer dataset is done by designing attack simulation at multiple layers of IoT using NetSim simulator. The attacks considered are DIS Flooding (Network Layer) and TCP SYN Flooding (Transport Layer). Detailed trace prompting information is provided in the "Implementation" section.

Step 2: Data Cleaning

Refining generated traces for standard training dataset: cleaning the accumulated trace files involves identifying mislaid values, N/A fields, and unrelated or single data. Data cleaning is essential to eliminate unnecessary columns and rows for creating a standardized training dataset. The data cleaning process for cross-functional attacks, involving both ordinary and invasion traces, is embellished in the implementation section.

Step 3: Merging Datasets to Single File

Combining attack and non-attack trace data to address inter-tier attack scenarios: A target column-*class type* is introduced here, with attack set to 1 and non-attack set to 0 for Step 1 and 0,1,2 for Step 2. This specifically created an entire cross-layer dataset by combining TCP SYN Flood and DIS Flooding traces for both the Steps.

Step 4: Data Transformation

Using One-hot encoding for converting categorical data: In direct conversion for example source and destination nodes, data columns are converted into numerical values. Simply substituting numerical values for the node numbers is common approach of data transformation. One-hot encoding is used to transform the numerous categorical variables such as *packet types* and *control packet types* into numerical values.

Step 5: Pearson Correlation Coefficient and Feature Selection

Reducing redundancy through correlation analysis: After one-hot encoding, employing the Pearson correlation coefficient technique to select relevant features and minimize redundant ones is executed on the dataset. A total of 12 features are chosen based on their correlation with the target column.

Step 6: Cross-Validation and Sub-dataset Selection

Enhancing efficiency through sub-datasets and cross-validation: Generating two sub-datasets for each type of attack to prevent efficiency issues caused by an increased dataset size is the most important part of dataset generation and segregation for training the models. A cross-validation technique is conducted on these sub-datasets during training to choose the most effective ones. This process results in four datasets, two for each layer attack.

Step 7: Designing Stack-Ensemble

Integrating base models into a unified stacking-ensembler: At this Step base models are combined into a single stacking-ensembler for training. The AIDS is designed by stacking of Decision Tree (DT), K-Nearest Neighbour (KNN), and Support Vector Machine (SVM) together into a single stack ensembler wherein the SIDS is designed by ensembling KNN, Gradient Boosting (GB), Decision Tree (DT), Random Forest (RF), and Logistic Regression (LR) models. Stack-ensemble helps to improve multiple weak model’s efficiency using voting approach. In the current study this approach improves the proposed intrusion detection system’s (IDS) overall performance.

Step 8: Comparative Analysis

Assessing efficiency across varied scenarios: Step 8 encompasses a rigorous comparative analysis for evaluating the model’s efficiency in both balanced and imbalanced test datasets. The analysis extends to varying scenarios, including the varying attacker nodes, the total number of nodes into the network, and processing time efficiency. This comprehensive approach ensures a thorough assessment of the proposed IDS model's robustness and performance across diverse scenarios within IoT environments.

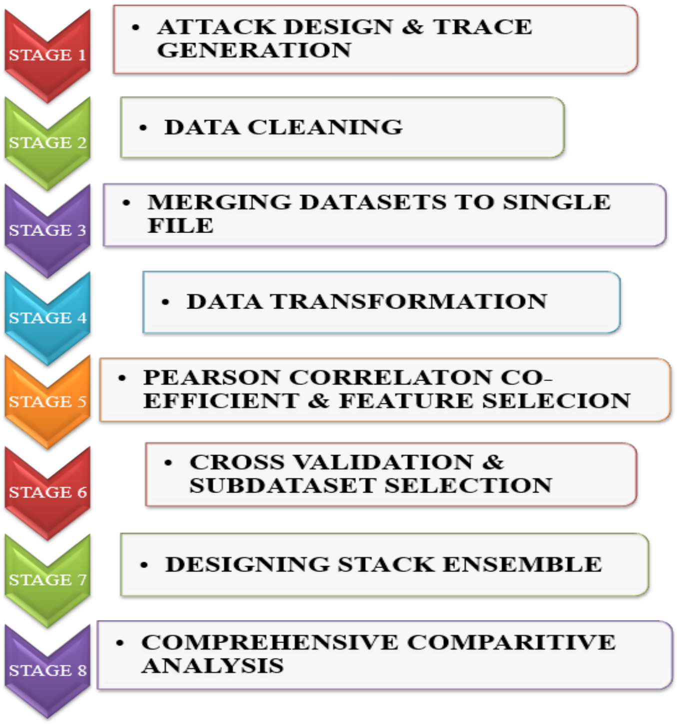

STAGE 1
• ATTACK DESIGN & TRACE GENERATION
STAGE 2
• DATA CLEANING
STAGE 3
• MERGING DATASETS TO SINGLE FILE
STAGE 4
• DATA TRANSFORMATION
STAGE 5
• PEARSON CORRELATON CO-EFFICIENT & FEATURE SELECION
STAGE 6
• CROSS VALIDATION & SUBDATASET SELECTION
STAGE 7
• DESIGNING STACK ENSEMBLE
STAGE 8
• COMPREHENSIVE COMPARITIVE ANALYSIS

Fig. 2. Methodology of the Proposed Intrusion Detection System

4.Implementation

The two-stage IDS is implemented using NetSim simulator(For designing attack and generating datasets) and Python IDE(for designing stack ensembler using ML models as discussed in Step 7).Two attacks as considered in the current study were designed using the Netsim simulator. Ten nodes and three attacker nodes were used in the SYN Flood invasion (Fig. 3), whereas ten nodes and two attacker nodes were used in the DIS Flooding assault (Fig. 4). The attacker nodes are shown as red nodes in the figures. Adhoc Links (Adhoc Link 1) were part of the network topology, which connected every node to the terminal node via a router and the 6LOWPAN Gateway. Wireless nodes might stick to a constant bit rate (CBR) via purple connection (App1_CBR) or communicate directly through wireless link.

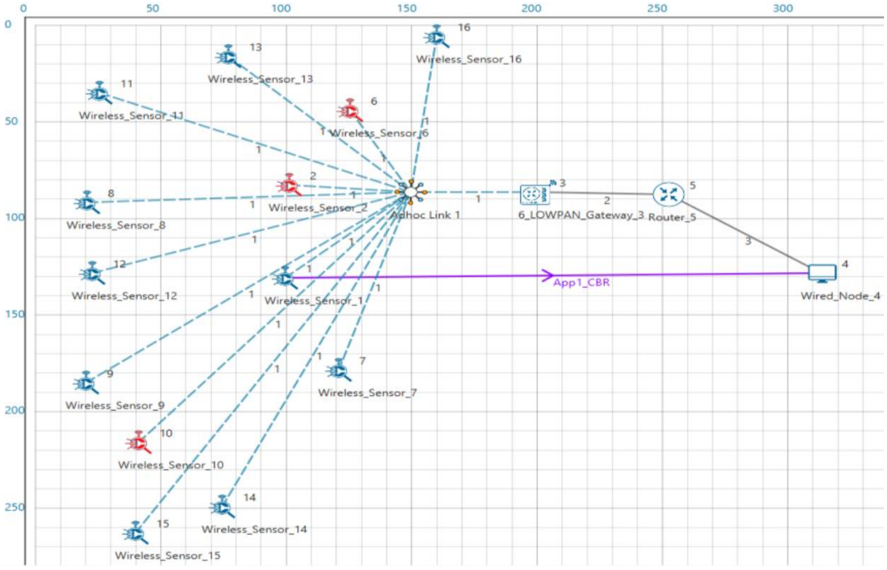


Fig. 3.SYN Flooding Assault with 3 Attacker Nodes and 10 Nodes

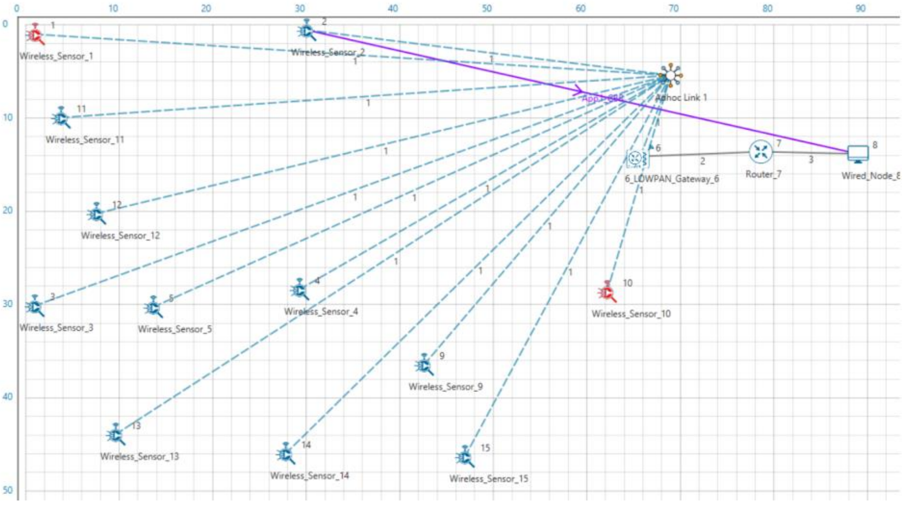
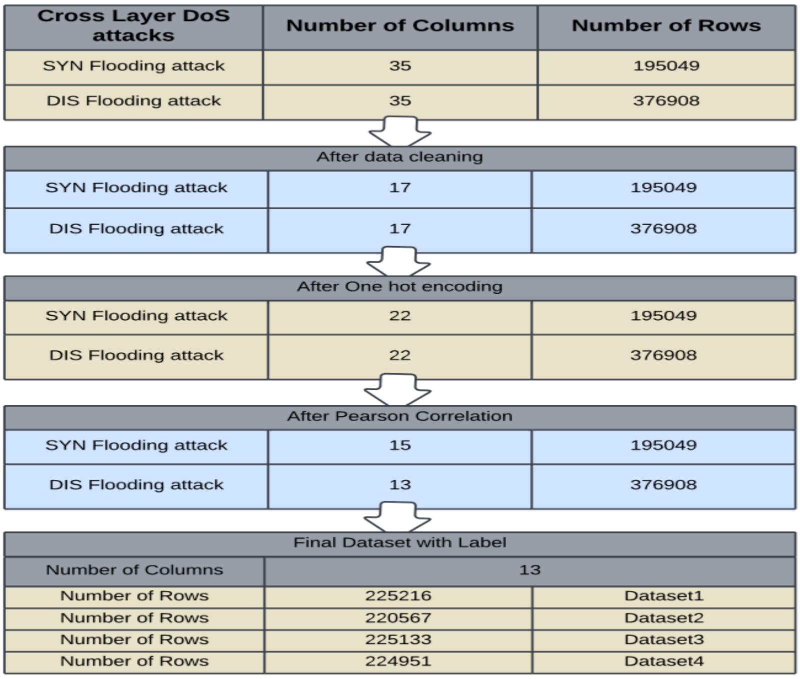


Fig. 4.DIS Flooding Assault with 2 Attacker Nodes and 10 Nodes

Dataset Generation:

The datasets for each of the attacks were generated by running each simulation for 50, 70, and 100 seconds. The generated trace files for for the SYN Flooding Assault had 35 features with 195,049 rows, while trace file for the DIS Flooding Assault had 35 features with 376,908 rows. After deleting the duplicate rows and N/A fields, a dataset containing 17 features was produced. Data transformation was done using one-hot encoding and highly correlated features were sorted using Pearson correlation method. Fig.s 5 and 6, this show a final set of 13 features, with the target variable (Class Type) positioned in the 13th column. Since one hot encoding adds extra bits for each categorical columns, number of columns increases due to this methos. Correlation method generated different features for each attack and here feature extraction technique is employed to come up with a common set of features for both the attack that give maximum accuracy. After merging two attack datasets, four subsets are generated as shown in Fig. 5. These datasets are passed through cross-validation in the ML models so as to get the best datasets. Result analysis section shows the relevant graph to understand the procedure.



Cross Layer DoS attacks	Number of Columns	Number of Rows
SYN Flooding attack	35	195049
DIS Flooding attack	35	376908

After data cleaning		
SYN Flooding attack	17	195049
DIS Flooding attack	17	376908

After One hot encoding		
SYN Flooding attack	22	195049
DIS Flooding attack	22	376908

After Pearson Correlation		
SYN Flooding attack	15	195049
DIS Flooding attack	13	376908

Final Dataset with Label		
Number of Columns	13	
Number of Rows	225216	Dataset1
Number of Rows	220567	Dataset2
Number of Rows	225133	Dataset3
Number of Rows	224951	Dataset4

Fig. 5. Refined Feature Set from Steps I to V

After the dataset is generated ML models are trained with the datasets.

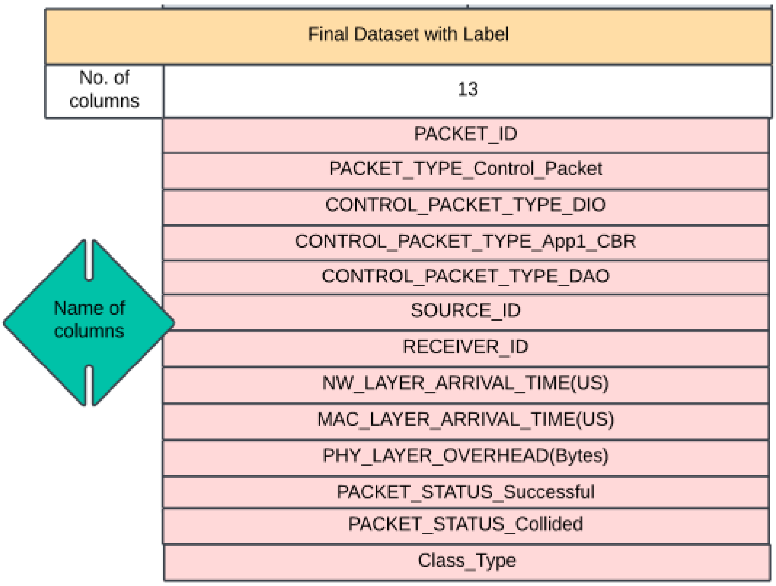
implementation a comparative analysis has been performed under hypothetical situations to understand the model's efficiency. For this different attack environment were created by increasing attacker nodes and legitimate nodes also. The number of legitimate nodes were 10, 20 and 40, whereas attacking nodes considered as 2 and 3 only. Similarly in another scenario, number of legitimate nodes is kept to 10 while number of attacking nodes have been raised to 3 and 4 from 2 and 3.

Fig. 7 shows the final dataset generated with the ratio of attack and non-attack as 1:1,1:2 and 1:3. Both 1:2 and 1:3 are imbalanced dataset where the attacker nodes are much higher than normal nodes. 1:1 is a balanced dataset with same number of classes.

Designing stack ensemblers:

We began with the cross-validation scores of each of the four datasets. The CV scores and test accuracy of the dataset are then evaluated. Our strategy was to picking the best dataset from the four. For this multiple training and testing trials were made by permuting the four datasets. After achieving the best and second best datasets from the CV technique next step is to design the stack ensembler for each stage. The Python StackingClassifier() function of sklearn, is used for implementing binary classification model (AIDS) in **Stage 1**. Here the ML models are ensembled and trained with Dataset 3 which in our case gives the best results in Stage 1. Dataset 4 with high CV score has also shown comparable accuracy to Dataset 3. The details values are analysed through graphs in result section.

Stage 2 follows the same procedures as Stage 1



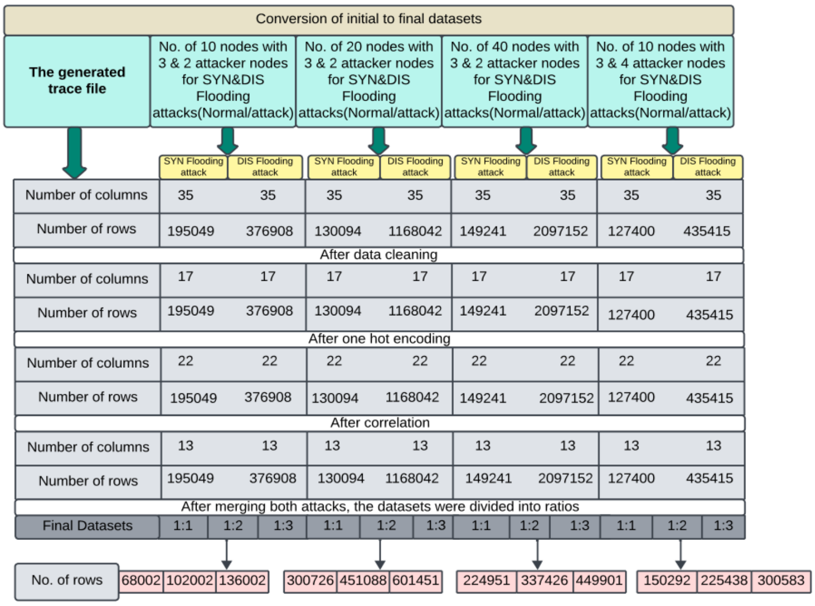
Final Dataset with Label

No. of columns	13

Name of columns

PACKET_ID
PACKET_TYPE_Control_Packet
CONTROL_PACKET_TYPE_DIO
CONTROL_PACKET_TYPE_App1_CBR
CONTROL_PACKET_TYPE_DAO
SOURCE_ID
RECEIVER_ID
NW_LAYER_ARRIVAL_TIME(US)
MAC_LAYER_ARRIVAL_TIME(US)
PHY_LAYER_OVERHEAD(Bytes)
PACKET_STATUS_Successful
PACKET_STATUS_Collided
Class_Type

Fig. 6. Final Feature set



Conversion of initial to final datasets

The generated trace file	No. of 10 nodes with 3 & 2 attacker nodes for SYN&DIS Flooding attacks(Normal/attack)		No. of 20 nodes with 3 & 2 attacker nodes for SYN&DIS Flooding attacks(Normal/attack)		No. of 40 nodes with 3 & 2 attacker nodes for SYN&DIS Flooding attacks(Normal/attack)		No. of 10 nodes with 3 & 4 attacker nodes for SYN&DIS Flooding attacks(Normal/attack)	
	SYN Flooding attack	DIS Flooding attack	SYN Flooding attack	DIS Flooding attack	SYN Flooding attack	DIS Flooding attack	SYN Flooding attack	DIS Flooding attack
Number of columns	35	35	35	35	35	35	35	35
Number of rows	195049	376908	130094	1168042	149241	2097152	127400	435415
After data cleaning								
Number of columns	17	17	17	17	17	17	17	17
Number of rows	195049	376908	130094	1168042	149241	2097152	127400	435415
After one hot encoding								
Number of columns	22	22	22	22	22	22	22	22
Number of rows	195049	376908	130094	1168042	149241	2097152	127400	435415
After correlation								
Number of columns	13	13	13	13	13	13	13	13
Number of rows	195049	376908	130094	1168042	149241	2097152	127400	435415

After merging both attacks, the datasets were divided into ratios

Final Datasets	1:1	1:2	1:3	1:1	1:2	1:3	1:1	1:2	1:3	1:1	1:2	1:3
No. of rows	68002	102002	136002	300726	451088	601451	224951	337426	449901	150292	225438	300583

Fig. 7. Conversion of initial to final datasets

5. Result Analysis

In this section a detail analysis of the proposed model is represented. Starting from Cross-validation to final outcome of the model, each step of Fig. 1 and 2 is evaluated and graphically depicted.

Fig. 8 is the representation of CV scores of the four datasets used for training and testing od the proposed model. Here it is evident that Dataset 3 has the highest accuracy than other three datasets and hence Dataset 3 is chosen as the training dataset for both the stages. However, Dataset 4 also have a close accuracy to Dataset 3. Thus Dataset 4 is used for testing of the model's efficiency.

Once we get the best training dataset, the two-stage model is now trained with it. Both the models (AIDS, SIDS) are trained with dataset 3 and training accuracy is shown in Fig. 6 and 7 respectively.

From Fig. 9 it is obvious that Stacking Classifier used for AIDS in stage 1 has a greater accuracy (96.86%) than KNN and SVM and almost similar accuracy to DT.

Similarly Fig. 10 shows a higher accuracy (97.04%) for Stacking Classifier in Stage 2 than other individual ML models.

After both the models are trained, the next part is to test each model for analysing their efficiencies. The AIDS is tested with dataset 4 and the output is passed to the SIDS at stage 2. The final accuracy at stage 2 is shown in Fig. 11 where Stacking Classifier has an accuracy of 95.97% with execution time 19.291 second. KNN has a little higher accuracy (96.2%) than SC but the execution time of KNN is the largest (25.304). This shows that SIDS using SC is more efficient than other ML models.

Since the current study analyses the model's efficiency in various attack environments multiple attacker nodes and legitimate nodes are considered and model is tested with these.

Fig. 12 shows a decrease in accuracy of stage 2 when number of nodes increases. this is obvious as the testing datasets now have more variation than the trained model which cannot detect new data. As the number of nodes increase execution time also increases (Fig. 13) and reaches up to 104 seconds with 40 nodes.

Fig.s 14-16 show the accuracy variations of stage 2 with increased number of attackers. The testing were done with 1:1, 1:2 and 1:3 test datasets and for each dataset the accuracy decreases with increased number of attackers.

Fig.s 17-19 show the execution time increases with increasing number of attackers in all datasets(1:1,1:2 and 1:3). This is also obvious as with increased attackers the datasets gets more variations which cannot be detected by the model.

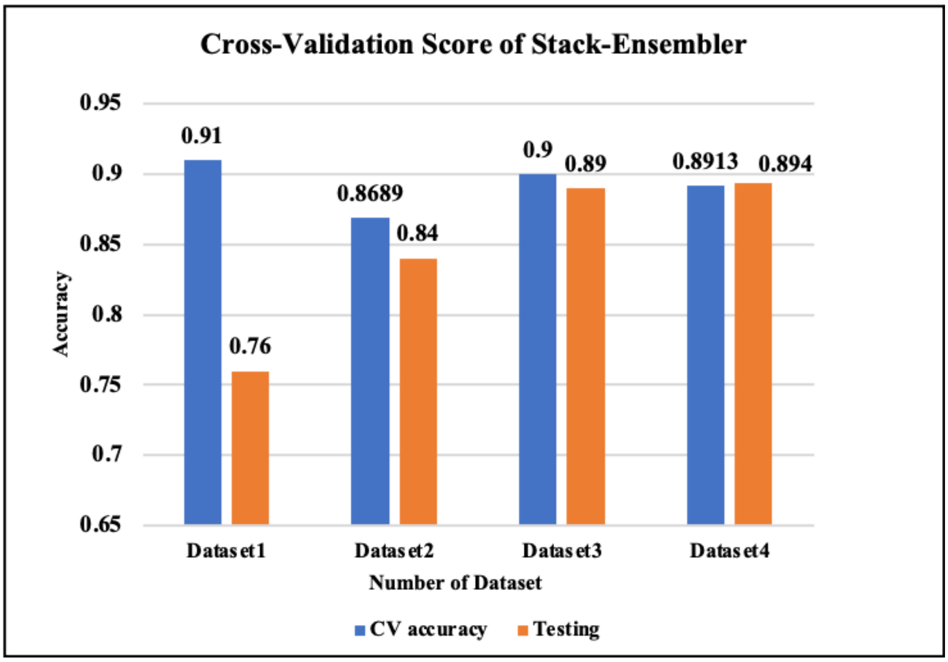


Fig. 8. Cross-validation scores of four datasets- Ensemble Testing

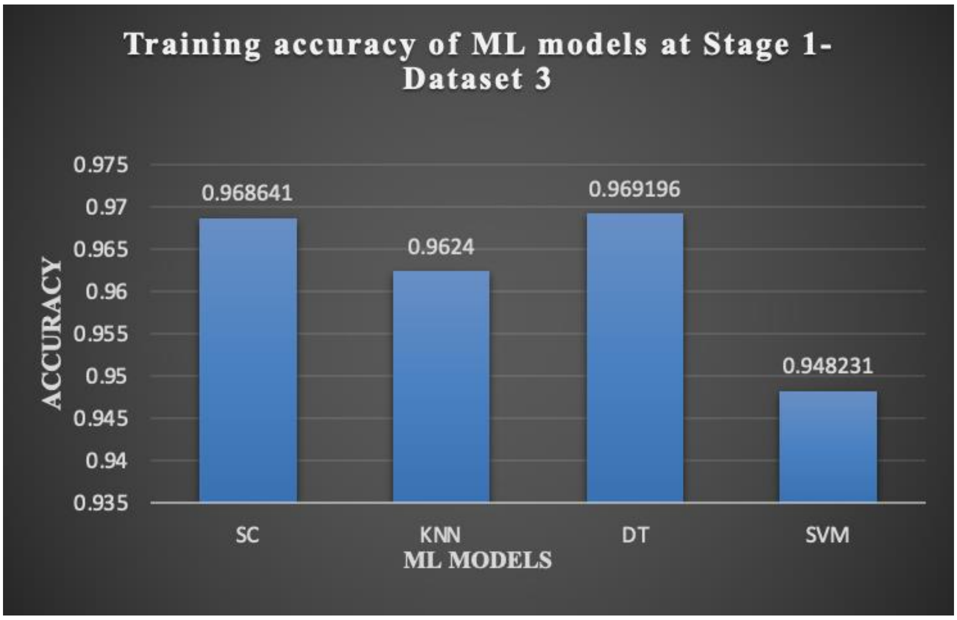


Fig. 9. Training Accuracy: Stage-1 ML Models

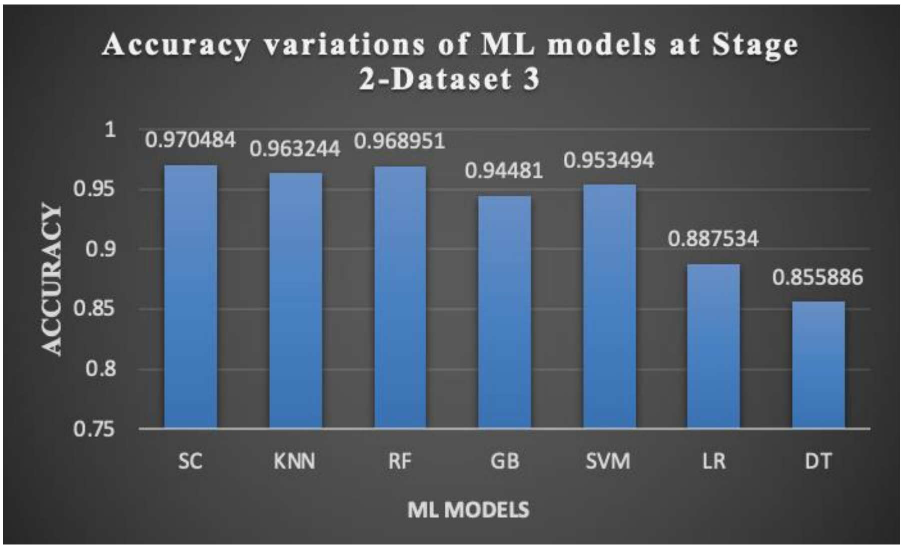


Fig. 10. Training Accuracy: Stage-2 ML Models

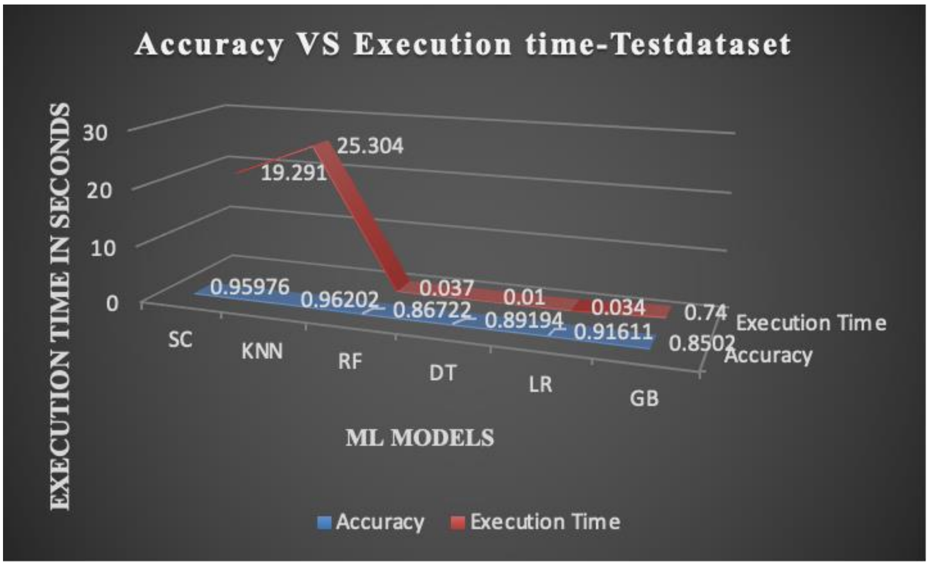


Fig. 11. Accuracy VS Execution Time of ML models

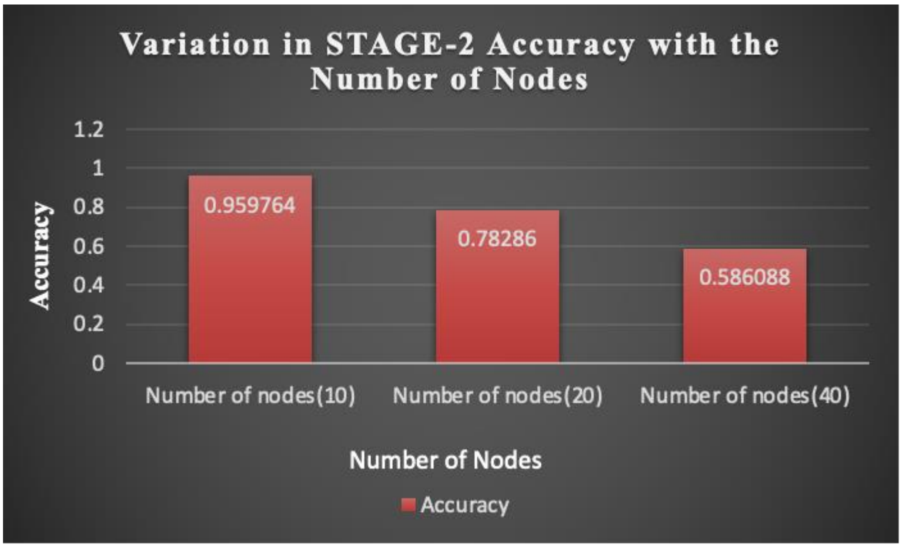


Fig. 12. Stage-2 Accuracy variations with number of nodes

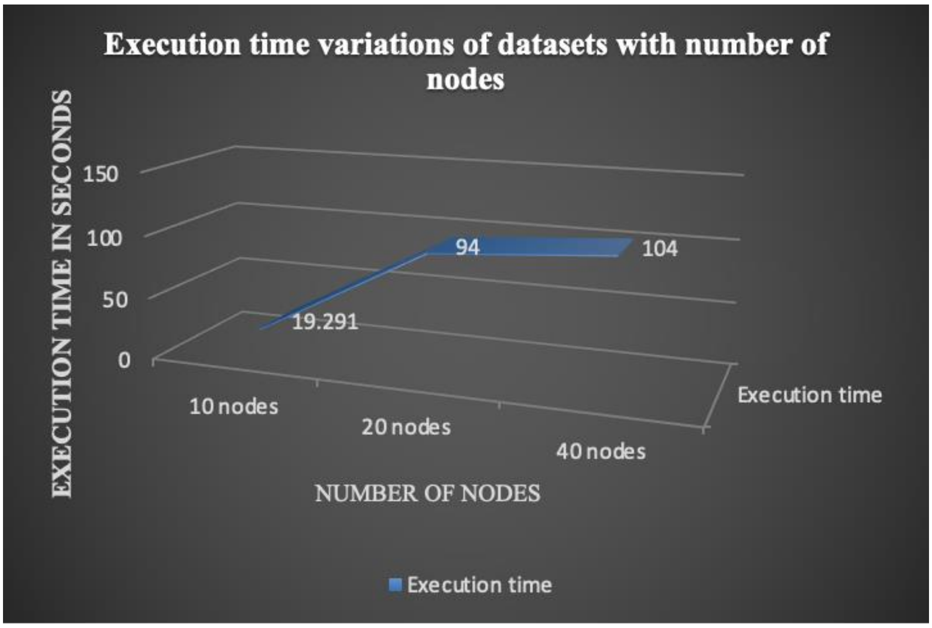


Fig. 13. Variations of execution time with increasing nodes

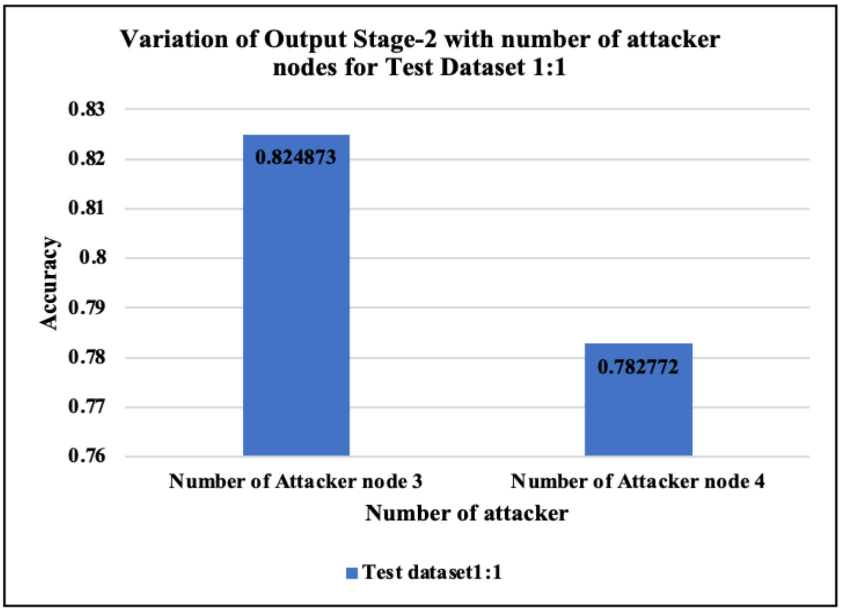


Fig. 14. Accuracy variation with number of attacker nodes for Test Dataset 1:1

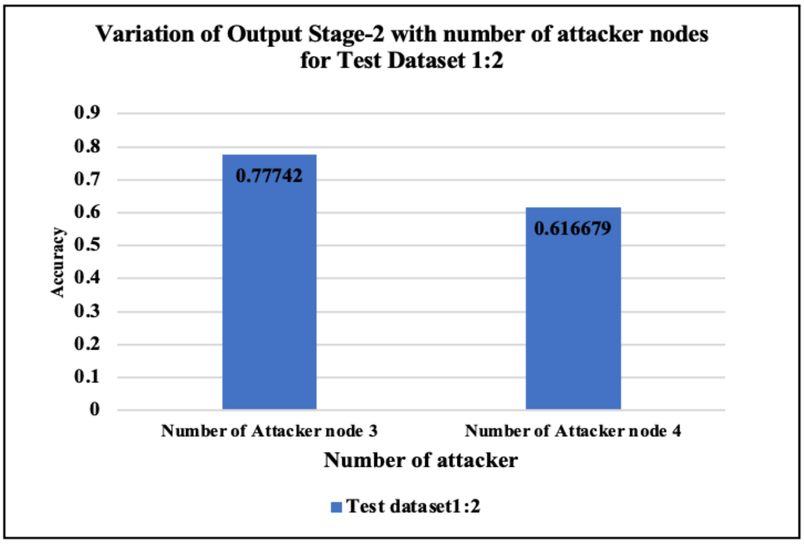


Fig. 15. Accuracy variation with number of attacker nodes for Test Dataset 1:2

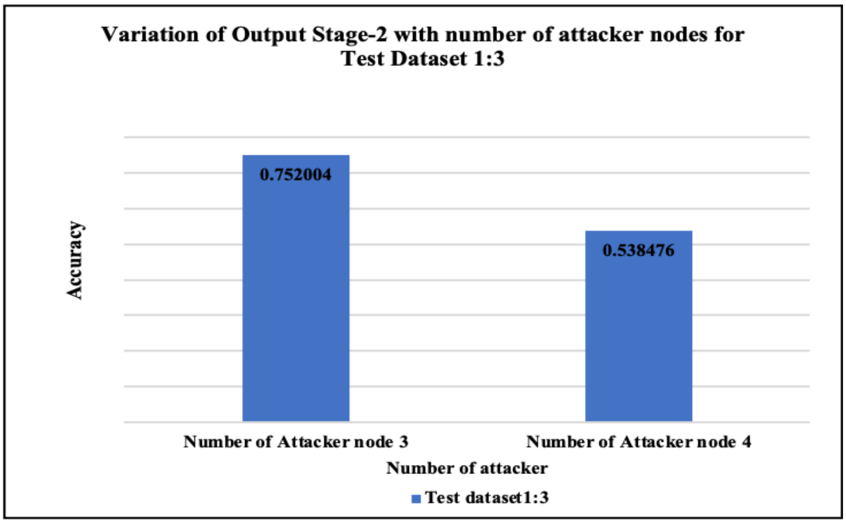


Fig. 16. Accuracy variation with number of attacker nodes for Test Dataset 1:3

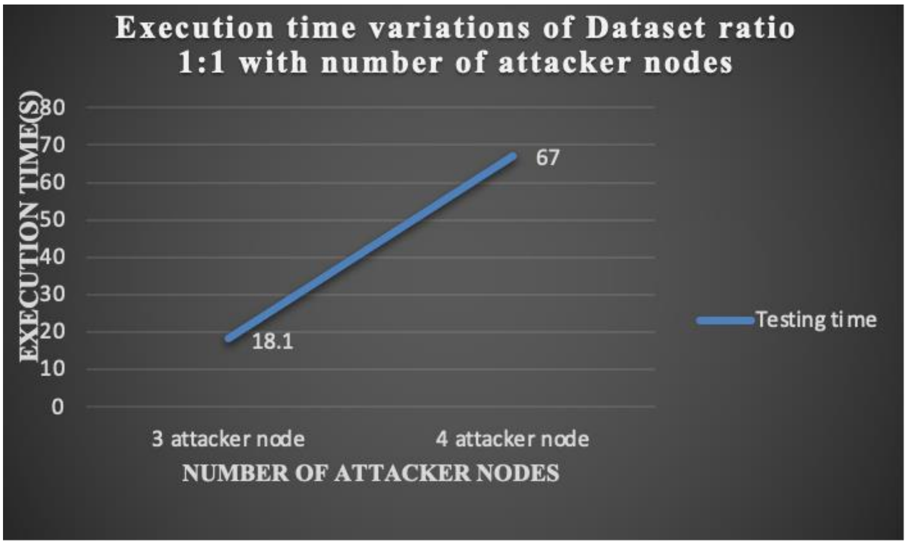


Fig. 17. Variations in execution time with Numbers of Attacker Nodes in dataset 1:1

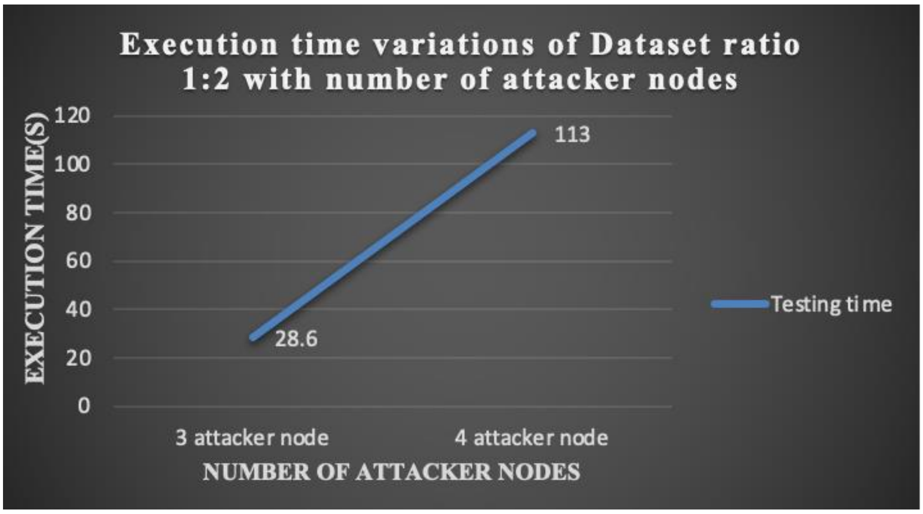


Fig. 18. Variations in execution time with Numbers of Attacker Nodes in dataset 1:2

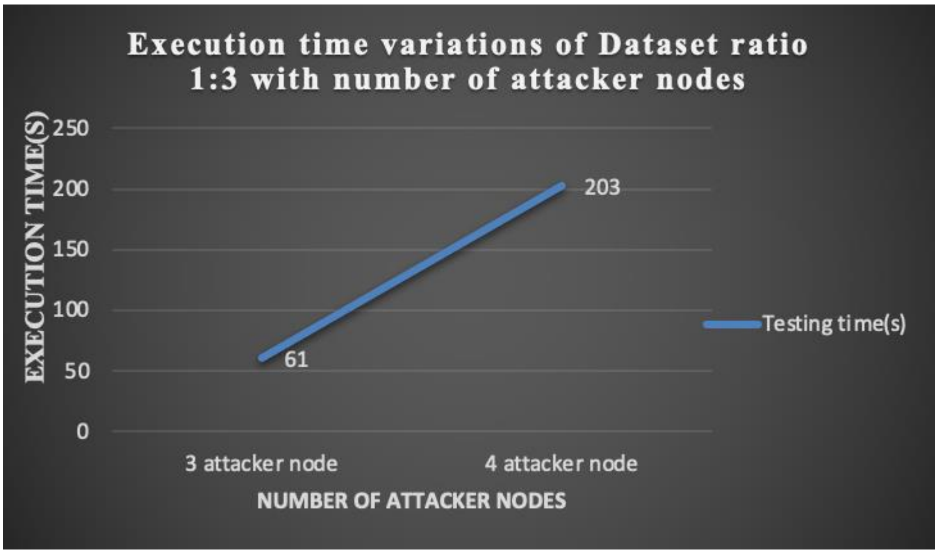


Fig. 19. Variations in execution time with Numbers of Attacker Nodes in dataset 1:3

6.Conclusion and Future Work

As a conclusion, our exploration introduces a Hybrid Intrusion Detection System (IDS) designed for detecting Denial of Service (DoS) attacks across diverse layers into Internet of Things (IoT). Notably, our proposed study stands out for its unique ability to construct datasets for various attacks, particularly emphasizing cross-layer DoS attacks within the IoT. Leveraging the NetSim simulator, we specifically engineered SYN flooding attacks at the transport layer (TCP protocol) and DIS flooding attacks at the network layer (RPL protocol). The proposed IDS designed by AIDS and SIDS have been able to perform with an accuracy of up to 96%.

At stage 1, the proposed model identifies attacks with up to 95% accuracy, and at stage 2, it categorizes attacks with up to 96% accuracy, showcasing robust detection capabilities. The efficiency of our model is evident in its rapid evaluation, taking only 18 to 19 seconds—markedly quicker than KNN with nearly identical accuracy. The primary focus hereof is to provide a thorough comparative assessment, considering various attacker and network node scenarios, along with processing time efficiency. These outcomes, derived from thorough training and test runs, are pivotal to enhancing the resilience of our approach against diverse attacks, and our Cross-Layer testing approach adds a nuanced dimension to the revelation of DoS attacks within the IoT framework.

In future it would be valuable to assess the accomplishment of the intended model using existing datasets from diverse sources. This analysis would provide insights into the model's adaptability across different datasets and could involve incorporating additional layers to evaluate its effectiveness against a broader range of attacks in the Internet of Things (IoT).

Declarations

Author contribution. Saumya Mishra has developed the methodology , Manoj has done a rigorous literature review on the problem domain, Somnath Sinha has implemented the model, Aditi Paul has analysed the results.

Funding statement. This is not funded by any agency.

Conflict of interest. The authors declare no conflict of interest.

Additional information. N.A

References

[1] Galeano-Brajones, J., Carmona-Murillo, J., Valenzuela-Valdés, J. F., & Luna-Valero, F. (2020). Detection and mitigation of DoS and DDoS attacks in IoT-based stateful SDN: An experimental approach. *Sensors, 20*(3), 816.

[2] R. Qamar, B.A. Zardari, A.A. Arain, Z. Hussain, and A. Burdi, "A Comparative Study of Distributed Denial of Service Attacks on The Internet Of Things By Using Shallow Neural Network," *Quaid-E-Awam University Research Journal of Engineering, Science & Technology, Nawabshah.*, *20*(01), 61-73, 2022.

[3] K. Saranya, and A. Valarmathi, "A Comparative Study on Machine Learning based Cross Layer Security in Internet of Things (IoT)," IEEE , pp. 267-273, December, 2022. *[International Conference on Automation, Computing and Renewable Systems (ICACRS,2022])*

[4] P. Bajaj, S. Mishra, and A. Paul, "Comparative Analysis of Stack-Ensemble-Based Intrusion Detection System for Single-Layer and Cross-layer DoS Attack Detection in IoT," *SN Computer Science*, *4*(5), 562, 2023.

[5] P. Nimbalkar, and D. Kshirsagar, "Feature selection for intrusion detection system in Internet-of-Things (IoT)." *ICT Express*, *7*(2), 177-181, 2021.

[6] E. Anthi, L. Williams, M. Słowińska, G. Theodorakopoulos, and P. Burnap, "A supervised intrusion detection system for smart home IoT devices," *IEEE Internet of Things Journal*, *6*(5), 9042-9053, 2019.

[7] T. Aliya, and E. Aiman, "A Survey on Recent Approaches in Intrusion Detection System in IoTs," IEEE,*[15th International Wireless Communications & Mobile Computing Conference (IWCMC).,2019]*.

[8] N. Moustafa, B. Turnbull, and K.K.R. Choo, "An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things," *IEEE Internet of Things Journal*, *6*(3), 4815-4830, 2018.

[9] Amouri, V.T. Alaparthy, and S.D. Morgera, "Cross layer-based intrusion detection based on network behavior for IoT," IEEE, pp. 1-4 [19th Wireless and Microwave Technology Conference (WAMICON), April, 2018].

[10] Amouri, S. D. Morgera, M. A. Bencherif, and R. Manthena, "A cross-layer, anomaly-based IDS for WSN and MANET," Sensors, 18(2), 651, 2018.

[11] E. Canbalaban, and S. Sen, "A cross-layer intrusion detection system for RPL-based Internet of Things." In *Ad-Hoc, Mobile, and Wireless Networks: 19th International Conference on Ad-Hoc Networks and Wireless, ADHOC-NOW 2020, Bari, Italy, October 19–21, 2020, Proceedings 19*, Springer International Publishing, pp. 214-227, 2020.

[12] H.Y. Kwon, T. Kim, and K.M. Lee, "Advanced intrusion detection combining signature-based and behavior-based detection methods," *Electronics*, *11*(6), 867, 2022.

[13] Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, and A. Alazab, "A novel ensemble of hybrid intrusion detection system for detecting internet of things attacks," Electronics, 8(11), 1210, 2019.

[14] M. Malik, M. Dutta, and J. Granjal, "IoT-sentry: A cross-layer-based intrusion detection system in standardized Internet of Things," *IEEE Sensors Journal*, *21*(24), 28066-28076, 2021.

[15] M. Sarhan, S. Layeghy, and M. Portmann, "Towards a standard feature set for network intrusion detection system datasets," *Mobile networks and applications*, 1-14, 2022.

[16] S. Sinha, A. Paul, "Neuro-Fuzzy Based Intrusion Detection System for Wireless Sensor Network," Wireless Pers Commun 114, 835–851 (2020). https://doi.org/10.1007/s11277-020-07395-y