

DDoS Detection in Smart Home IoT Networks: An Experimental Comparison of MLP and AE-MLP with Correlation-Based Feature Selection

Nurul Afifah^{1*}, Septiani Kusuma Ningrum², RA Siti Safa Marefa³, Jihan Badiatus Shaliha⁴, Hikmah Maharani⁵, Ilma Sari⁶, Andi Daniella⁷

Department of Computer Engineering, Faculty of Computer Science, Universitas Sriwijaya

**nurul@unsri.ac.id*

ABSTRACT

The growing use of Internet of Things (IoT) devices in smart homes increases the network attack surface and exposes them more to Distributed Denial of Service (DDoS) attacks. We evaluate a baseline Multilayer Perceptron (MLP) and an Autoencoder-Multilayer Perceptron (AE-MLP) hybrid model for binary DDoS detection, using flow-based traffic collected in a controlled smart home testbed environment. We converted CICFlowMeter PCAP traffic samples to flow level data. Data preparation was performed using a data-cleaning pipeline, correlation-based feature selection and Min-Max normalization. Both models are trained on the same experimental setting and data split. The accuracy of the MLP is 98.1% and the F1-score is 98.3%, which is higher than the AE-MLP with an accuracy of 68.7% and an F1-score of 71.5% before feature selection. The feature selection was really useful for both models. In general, MLP achieved the best performance with accuracy of 99.5% and F1-score of 99.5%, followed by AE-MLP with accuracy of 99.2% and F1-score of 99.3%. The performance of the algorithm on the testbed was enhanced with feature selection based on correlation. The results should be interpreted taking into account the presence of metadata fields that might limit the generalization to unexplored situations and the use of a restricted dataset.

Keywords: IoT, Smart Home, DDoS, MLP, AE-MLP, Feature Selection.

1. INTRODUCTION

The rapid development of IoT has transformed conventional residential environments into smart home ecosystems consisting of various sensors, actuators, consumer devices, and cloud-based services [1], [2], [3]. Devices such as IP cameras, smart plugs, smart bulbs, and smart locks continuously exchange telemetry data and control messages via local networks and the Internet, enabling automation, monitoring, and remote control of devices. However, the heterogeneous nature of IoT devices and their limited computational resources, coupled with limitations in firmware updates and suboptimal security configurations, contribute to an expanded attack surface within home networks [4], [5]. One significant threat is DDoS an attack that exploits compromised IoT devices to generate massive amounts of malicious traffic directed at a target [6]. In a smart home environment, such attacks can disrupt service availability, degrade device communication quality, and hinder services that rely on connectivity, and cloud [7], [8], [9]. These conditions highlight the need for a

high performance detection mechanism that also accounts for resource constraints on IoT gateways or aggregator devices [10], [11]. Various machine learning and deep learning approaches have been developed to detect DDoS attacks on IoT networks [12], [13]. One relevant method is AE-MLP, which combines MLP for classification with the ability of the autoencoder to learn feature representations. Research [14] The results of the study [14] shows that AE-MLP has got 98.34% accuracy, 97.91% precision, 98.48% recall and 98.18% F1-score in identifying DDoS attacks. Although it delivers high performance, the effectiveness of AE-MLP is heavily influenced by the characteristics and quality of the input features used. Study [15] employed a voting-based feature selection approach combined with an MLP and a Genetic Algorithm (MLP-GA), achieving an accuracy of 98.8%. These results indicate that feature selection may be very important to reduce the superfluous features as well as to improve the efficacy of the categorization.

Although various studies have reported high detection performance, several research gaps remain [16]. First, some studies still use public or synthetic datasets, which do not fully represent the traffic characteristics of a physical smart home testbed, such as inter-device communication patterns, background traffic, variations in packet delivery times, and noise generated by household applications. Second, performance improvements achieved through more complex architectures have the potential to increase computational requirements and inference latency, which may limit their application in IoT environments with limited resources. Third, the contributions of preprocessing and feature selection stages to model performance improvements have not always been analyzed separately. This makes it difficult to determine whether performance improvements stem primarily from the model architecture or from the quality of feature representation. Fourth, comparative studies that systematically compare MLP as a baseline model with AE-MLP as a hybrid deep learning model particularly using datasets obtained directly from physical smart home environments remain limited. To address this gap, this study presents a controlled experimental evaluation of MLP and AE-MLP for DDoS attack detection in smart home environments.

MLP is used as the baseline model, while AE-MLP is used as the hybrid deep learning approach [17], [18]. The main focus of this research is to analyze the impact of correlation matrix-based feature selection on the performance of both models and to evaluate the differences in the models' responses to the quality of input features. The dataset was constructed from a physical smart home testbed consisting of a smart door lock, an IP camera, a smart plug, and a smart bulb; network traffic was then converted into flow-based features using CICFlowMeter. Both models were evaluated using consistent datasets, preprocessing procedures, and experimental configurations to ensure a controlled comparison. The main contributions of this study are:

- The development of a primary DDoS dataset based on a physical smart home testbed obtained from a controlled environment involving various IoT devices, namely a smart door lock, an IP camera, a smart plug, and a smart bulb.
- A quantitative analysis of the impact of feature selection on the performance of MLP and AE-MLP before and after feature selection, thereby determining the contribution of feature quality to the success of DDoS detection.

The structure of this paper is as follows. Section II describes the smart home testbed, attack scenarios, data collection procedures, the preprocessing pipeline, and

the model architecture. Section III describes the experimental results, including hyperparameter tuning, performance comparisons, feature selection impact analysis, research limitations, and deployment implications. In Section IV we summarize the main results and discuss possible future research directions.

2. MATERIALS AND METHODS

The experiments were conducted using a smart home testbed environment consisting of a smart door lock, an IP camera, a smart plug, and a smart bulb to generate both normal traffic and DDoS attack traffic. All network communication activities were captured in PCAP format and then processed using CICFlowMeter to create a flow-based dataset. The dataset subsequently underwent data cleaning and feature reduction using correlation-based feature selection. Following that, built and assessed MLP and AE-MLP models in various experimental setups, such as diverse dataset splits, optimizers, learning rates, and epoch counts. The models are evaluated on the basis of Accuracy, Precision, Recall and F1-Score. The research technique workflow is shown in Figure 1.

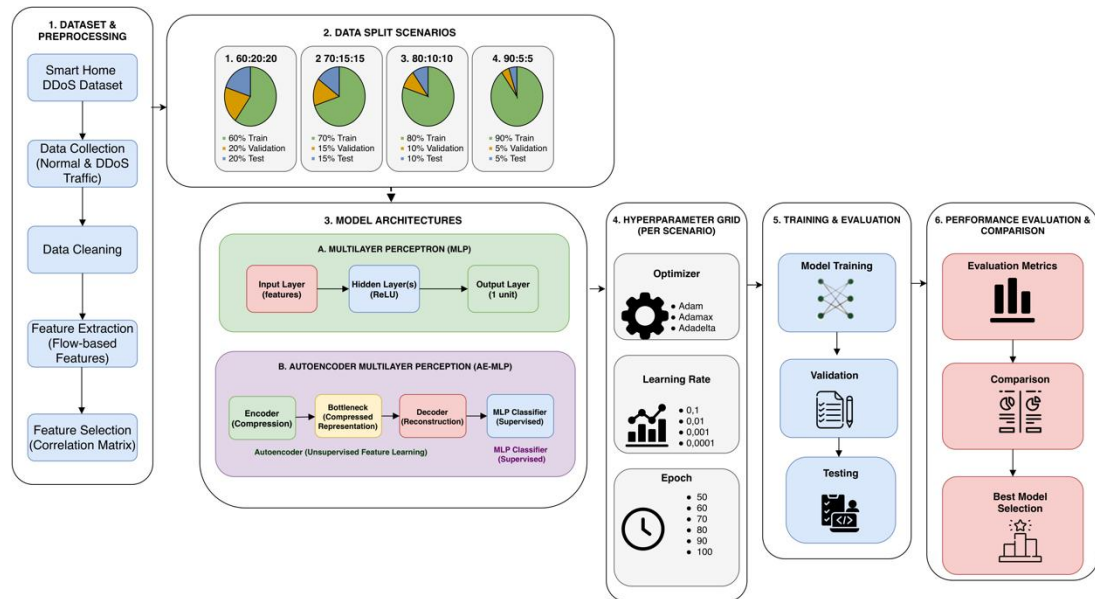


FIGURE 1. Research Methodology

2.1 SMART HOME TESTBED & ATTACK SCENARIO

The network topology used in the DDoS *attack* scenario of this study was designed to simulate an IoT environment vulnerable to DDoS attacks. This network consists of various devices connected to each other via a *router* or *access point* serving as the communication hub [19], [20], [21]. Connected devices include computers, *Wi-Fi* devices, *indoor* IP cameras, *smart plugs*, LED lights, and *smart locks*. Each device in this network can communicate one- ly with one another via an application connected through the internet, as shown in Figure 2.

Nurul Afifah, Septiani Kusuma Ningrum, RA Siti Safa Marefa, Jihan Badiatus Soleha,
Hikmah Maharani, Ilma Sari, Andi Daniella
DDoS Detection in Smart Home IoT Networks: An Experimental Comparison of MLP
and AE-MLP with Correlation-Based Feature Selection

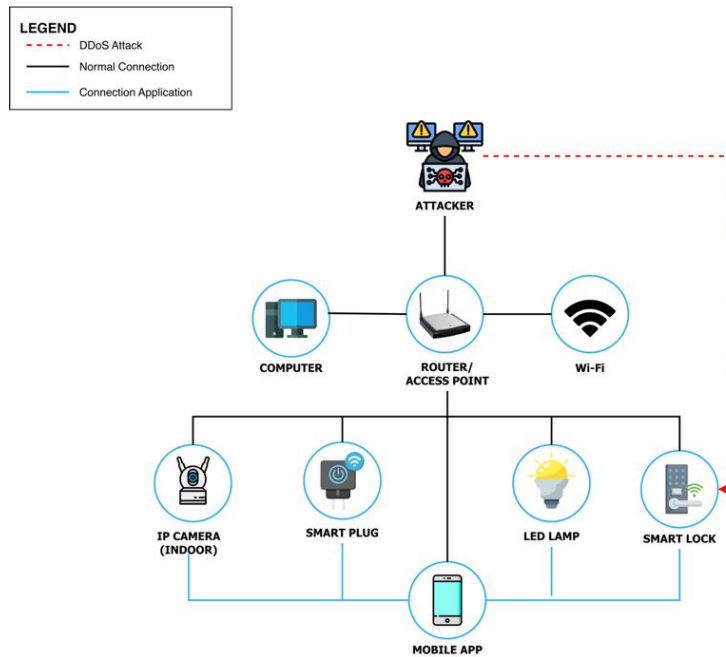


FIGURE 2. Proposed Topology

Data collection for this study was conducted using various *smart home* devices interconnected within the network. The appearance of a series of log entries in rapid succession over a very short time interval indicates network scanning activity aimed at identifying active devices or services.

1	06/16-15:33:14.868076	[**]	[1:1917:6]	SCAN UPnP service discover attempt	[**]	[Classification: Detection of a Network Scan]
2	06/16-15:33:31.110438	[**]	[1:1917:6]	SCAN UPnP service discover attempt	[**]	[Classification: Detection of a Network Scan]
3	06/16-15:33:32.125653	[**]	[1:1917:6]	SCAN UPnP service discover attempt	[**]	[Classification: Detection of a Network Scan]
4	06/16-15:33:33.140173	[**]	[1:1917:6]	SCAN UPnP service discover attempt	[**]	[Classification: Detection of a Network Scan]
5	06/16-15:33:34.154673	[**]	[1:1917:6]	SCAN UPnP service discover attempt	[**]	[Classification: Detection of a Network Scan]
6	06/16-15:35:15.548255	[**]	[1:1917:6]	SCAN UPnP service discover attempt	[**]	[Classification: Detection of a Network Scan]
7	06/16-15:35:31.115738	[**]	[1:1917:6]	SCAN UPnP service discover attempt	[**]	[Classification: Detection of a Network Scan]
8	06/16-15:35:32.131034	[**]	[1:1917:6]	SCAN UPnP service discover attempt	[**]	[Classification: Detection of a Network Scan]
9	06/16-15:35:33.141600	[**]	[1:1917:6]	SCAN UPnP service discover attempt	[**]	[Classification: Detection of a Network Scan]
10	06/16-15:35:34.149345	[**]	[1:1917:6]	SCAN UPnP service discover attempt	[**]	[Classification: Detection of a Network Scan]
11	06/16-15:37:15.560337	[**]	[1:1917:6]	SCAN UPnP service discover attempt	[**]	[Classification: Detection of a Network Scan]
12	06/16-15:37:31.130410	[**]	[1:1917:6]	SCAN UPnP service discover attempt	[**]	[Classification: Detection of a Network Scan]

FIGURE 3. Snort Analysis

Figure 3 shows that this activity was detected by Snort via the rule “SCAN UPnP service discovery attempt” with Signature ID (SID) 1917:6, and was subsequently classified as “Detection of a Network Scan” with Priority 3. Although SSDP traffic may occur as part of a legitimate service discovery process, the high frequency of packet transmission within a short period is a characteristic commonly found during the reconnaissance phase prior to a cyberattack. These detection logs are subsequently used as one of the validation sources in the dataset labeling process. Information regarding the time of the incident, the type of attack, and the IP addresses involved is used to ensure that the collected network traffic data has been accurately labeled

before entering the preprocessing and machine learning model training stages. Consequently, the resulting dataset has better labeling quality and can improve the reliability of the proposed DDoS detection model evaluation process.

2.2 DATASET

The data set used in this study was collected from a controlled smart home testbed built by the COMNETS Smart Home research group, Universitas Sriwijaya [22]. The testbed emulated an IoT-enabled home network with a smart door lock, an indoor PTZ IP camera, a smart power strip and an LED smart bulb. These devices were connected to each other using a TP-Link TL-WR840N router as the wireless gateway and access point. Computers, laptops and mobile devices were used to drive the IoT devices, generate traffic, monitor network activity and capture traffic data. Two traffic scenarios were collected: normal traffic and DDoS traffic. In the normal-traffic scenario, the smart-home devices generated normal communication including device-to-application interaction, service discovery, control commands, and background network activity. Normal traffic was captured for approximately 5 minutes using Wireshark. This resulted in a PCAP file of 2448 packets and size of about 9938 KB. In the DDoS-traffic scenario, a host from the local network discovered active devices and sent a large number of connection requests to a targeted IoT device. The repeated TCP SYN packets and retransmissions that can cause packet loss, latency and disruption of service characterized the attack traffic. The DDoS capture had 2,474,659 packets and was approximately 1,660,593 KB in PCAP file size. Traffic labeling was supported by traffic traces and Snort alerts. The PCAP files were broken into manageable chunks using tcpdump. Next, they were converted into flow-level records using CICFlowMeter. The experimental dataset was combined with the generated CSV documents. The final dataset comprised of 1,832 flow records, of which 778 (42.47%) were normal flows and 1,054 (57.53%) were DDoS flows. The field “Label” was encoded as 0 for normal traffic and 1 for DDoS traffic.

TABLE 1
Feature Categories

Field category	Fields	Experiment
Flow identifier and metadata	Flow ID, Src IP, Timestamp	Included as preprocessed model-input attributes
Flow-behavior attributes	Flow Duration, Fwd Pkt Len Min, Bwd Pkt Len Min, Bwd Packets/s, Pkt Len Min, and other selected CICFlowMeter features	Used for correlation-based feature selection and model training
Target variable	Label	normal = 0 and DDoS = 1

As shown in Table 1, the input feature set included flow identifiers and metadata fields Flow ID, Src IP, Timestamp and flow-behavior attributes derived from CICFlowMeter, such as flow-duration stats, packet-length stats, and directional

packet-rate features. In the experiment presented in this paper, the identifier and metadata fields were maintained as preprocessed model input attributes. The Label field was used solely as the target for binary classification.

2.3 DATA PARTITIONING AND FEATURE SELECTION

The experiments were performed on a computer with a processor AMD Ryzen 3 7320U and 8 GB RAM using TensorFlow in Jupyter Notebook. In the experimentation phase, we experimented with different training testing validation ratios and analyzed the effect of data partitioning on the efficiency of DDoS detection. The ratios studied were 50:25:25, 60:20:20, 70:15:15, 80:10:10 and 90:5:5 respectively. The same preprocessing of data and hyper parameter search was applied for the MLP and AE-MLP models in each partition configuration for training and testing. We compared experimental parameters on the validation split and computed final evaluation metrics on the test split. The most relevant attributes to the binary class label were identified using correlation-based feature selection [23]. As shown in Figure 4, The selected numerical features are normalized by min-max normalization to the range of 0 to 1. For fair comparison the same configuration was used for pre-processing MLP and AE-MLP. In the experiment, the input attributes (Flow ID, Src IP and Timestamp) were kept unchanged. Thus the observed performance could be a function of both traffic related patterns and testbed specific meta data. The limitation of this study is openly discussed in the problem.

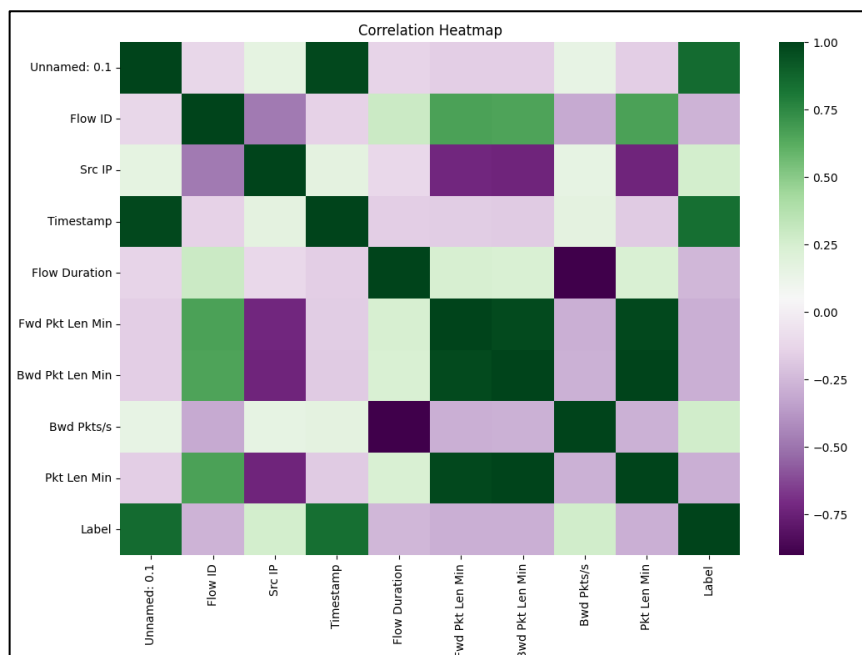


FIGURE 4. Heatmap Visualization of Selected Feature Correlations

2.4 PROPOSED ARCHITECTURE

In this study, two artificial neural network architectures were used to detect DDoS attacks: MLP as the baseline model and AE-MLP as the proposed model. Both models

were developed using the same dataset, so the difference in performance is solely influenced by the model architecture and the combination of hyperparameters used. As shown in the figure, the input data consists of n preprocessed features that are passed through several fully connected hidden layers. Each hidden layer uses the ReLU activation function to learn the nonlinear relationships between features. The representation obtained in the last hidden layer is then fed to the output layer which is a single unit to give a class prediction. This architecture allows the MLP to learn directly from the input features to the attack label, without the need for an additional latent representation stage, as shown in Figure 5.

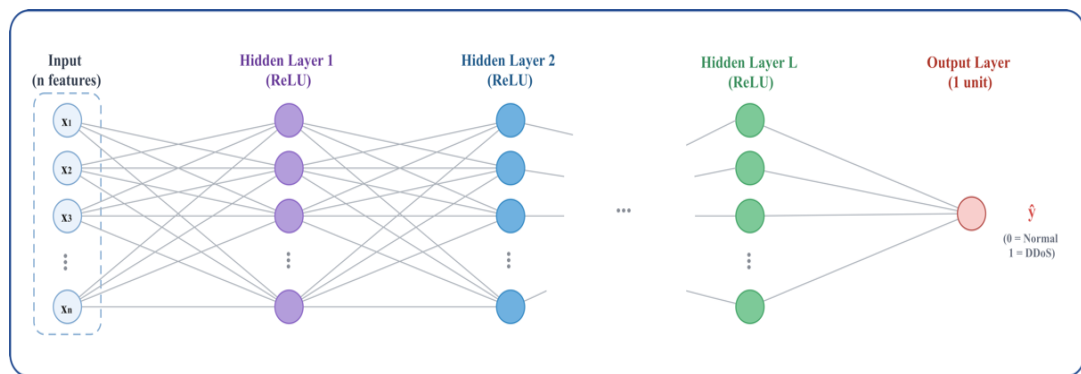


FIGURE 5. MLP Architecture

In the initial stage shown in Figure 6, n input features are processed by several encoder layers using the ReLU activation function to perform gradual compression until reaching a bottleneck, which represents a lower-dimensional latent representation. This representation is then passed through a decoder layer to reconstruct the input features. The reconstructed output is then used as the input representation for the MLP classification section, which consists of several hidden layers with ReLU activation and one output unit to distinguish between Normal (0) and DDoS (1) traffic. Thus, AE-MLP not only performs classification but also first utilizes an Autoencoder to learn a more concise feature representation before it is used by the MLP in the classification process.

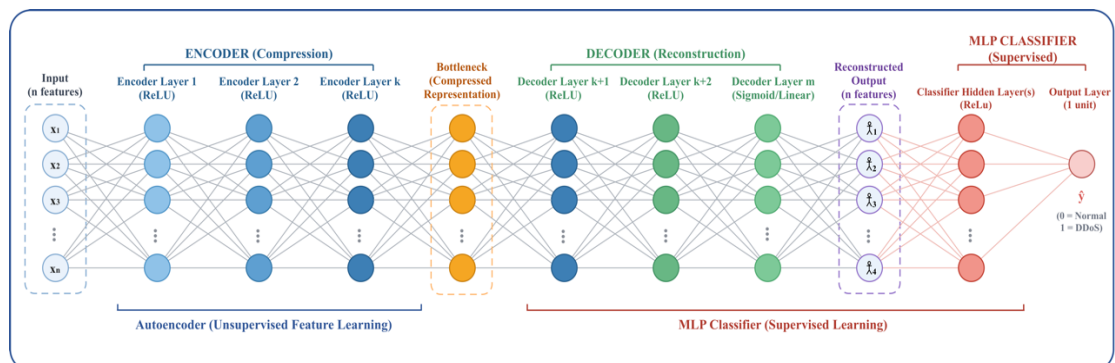


FIGURE 6. AE-MLP Architecture

The MLP classifier consists of two fully connected hidden layers with 64 and 32 neurons, respectively. The two hidden layers were activated by ReLU function and

used dropout rate of 0.2 with L2 regularization. The output layer was a single sigmoid neuron for binary classification. The AE-MLP autoencoder was constructed for reconstruction of the input with 128, 64 and 32 neurons in the encoder and with a symmetric decoder with 64 and 128 neurons. An MLP classifier was constructed for the 32 neuron bottleneck representation. The MLP processes features directly through a series of fully connected layers, whereas the AE-MLP adds an encoding bottleneck decoding stage to learn feature representations before classification is performed by the MLP. Therefore, a comparison of the two models allows the study to evaluate whether learning representations via an Autoencoder provides additional benefits compared to a direct MLP approach in detecting DDoS traffic.

TABLE 2.
Hyperparameter Search Space for MLP and AE-MLP

Hyperparameter	Values
Data partition	50:25:25, 60:20:20, 70:15:15, 80:10:10, 90:5:5
Optimizer	Adam, Adamax, Adadelta
Learning rate	0.1, 0.01, 0.001, 0.0001
Epochs	10, 20, 30, 40, 50, 60, 70, 80, 90, 100

Table 2 shows 5 ratio of data partitioning for training, validation and testing dataset respectively, i.e., 50:25:25, 60:20:20, 70:15:15, 80:10:10 and 90:5:5. Three algorithms were tested namely Adam, Adamax and Adadelta . This was to study the effect of the optimization algorithms on the convergence of the model and the efficiency of the classification . We also consider four learning rates 0.1, 0.01, 0.001 and 0.0001. For training, the number of epochs was varied in factors of ten from 10 to 100. We train the two models on identical hyperparameter spaces. Experimental results showed best and reliable performance in comparative study with data split 60:20:20, adam optimizer and learning rate 0.001. Therefore, we have performed detailed analysis of MLP and AE-MLP before and after applying correlation driven feature selection in this environment. We trained the MLP classifier with Adam optimizer and binary cross-entropy loss with batch size of 64. The autoencoder part of AE-MLP was trained with the Adam optimizer and the mean squared error (MSE) as reconstruction loss. The MLP classifier was trained on the 32 dimensional bottleneck representation after autoencoder training with binary cross-entropy loss and batch size 64.

2.5 EVALUATION METRICS

The evaluation was based on the classification results on the test data which are presented in a Confusion Matrix having four main parts: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). We use these four components to compute the various evaluation metrics used in this study [24], [25]. The four key metrics used to assess model performance were Accuracy, Precision, Recall, and F1-score, which are represented in Equations (1)–(4).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1 - Score = \frac{2 \cdot (Precision \times Recall)}{Precision + Recall} \quad (4)$$

3. RESULT

The MLP and AE-MLP models were trained for 50, 60, 70, 80, 90 and 100 epochs before and after the feature selection. Five different split ratio for training, validation and testing for experiment are 50:25:25 60:20:20 70:15:15 80:10:10 90:5:5. For the last comparison evaluation we used 60:20:20 split for better and more stable results on the validation set while having enough data for the model training, hyperparameter tuning and independent evaluation. The data will be divided into 60% training, 20% validation and 20% testing. The results of the confusion matrix evaluation of the MLP and AE-MLP models before and after the feature selection are shown in Tables 3 and 4.

TABLE 3.
Model Evaluation Results Before Feature Selection

Model	Metric	Epoch					
		50	60	70	80	90	100
MLP	<i>Acc</i>	96.70	97.80	98.10	97.30	98.10	98.10
	<i>Prec</i>	95.80	97.60	98.60	96.70	98.60	98.60
	<i>Rec</i>	98.60	98.60	98.10	98.60	98.10	98.10
	<i>F1</i>	97.20	98.10	98.30	97.60	98.30	98.30
AE-MLP	<i>Acc</i>	67.60	67.80	68.70	68.10	67.30	67.60
	<i>Prec</i>	74.50	77.20	75	75	74.90	76.40
	<i>Rec</i>	66.30	62.60	68.20	66.80	64.90	63
	<i>F1</i>	70.20	69.10	71.50	70.70	69.50	69.10

TABLE 4.
Model Evaluation Results After Feature Selection

Model	Metric	Epoch					
		50	60	70	80	90	100
MLP	<i>Acc</i>	97.30	99.20	99.20	99.20	99.50	98.90
	<i>Prec</i>	95.90	99	99.50	100	99.50	100
	<i>Rec</i>	99.50	99.50	99	98.60	99.50	98.10
	<i>F1</i>	97.70	99.30	99.30	99.30	99.50	99
AE-MLP	<i>Acc</i>	97.50	97.50	98.10	99.20	98.10	99.20
	<i>Prec</i>	96.70	99.50	98.10	99.50	99.50	99.50
	<i>Rec</i>	99	96.20	98.60	99	97.20	99
	<i>F1</i>	97.90	97.80	98.30	99.30	98.30	99.30

Before the feature selection, the direct MLP classifier gave better results than the AE-MLP. In the considered epoch settings, the MLP achieved an accuracy of 96.7%-98.1% and the F1-score of 97.2%-98.3%, while the AE-MLP received an accuracy of 67.3%-68.7% and the F1-score of 69.1%-71.5%. The large performance gap suggests that the autoencoder stage has not learned a sufficiently discriminative representation to the downstream classifier on the whole feature space. One possible explanation is that the reconstruction criterion tends to preserve the global variation of the input rather than the class separating information. If the input data includes irrelevant, redundant or noisy variables, the bottleneck may keep the variation that is important for the reconstruction, but at the same time lose the information that is important for the distinction between normal and DDoS flows.

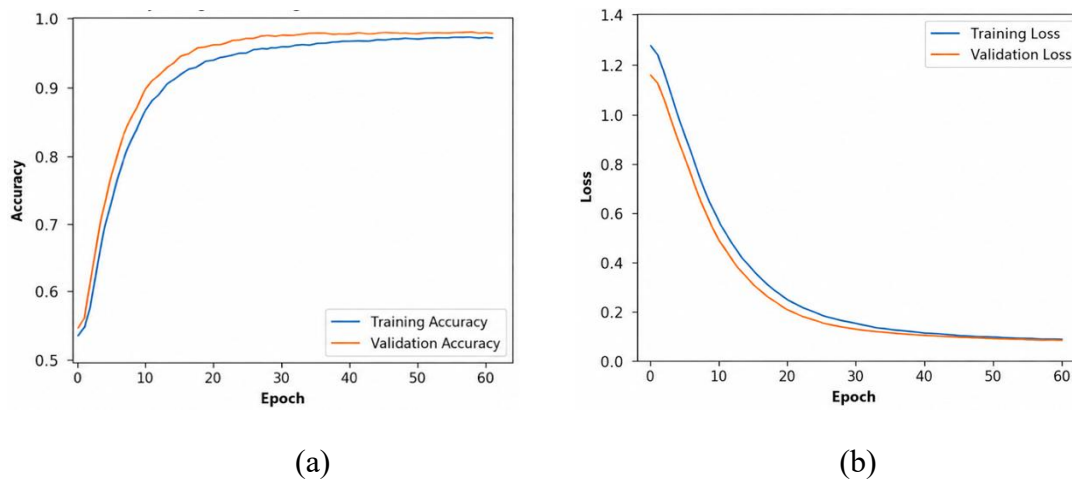


Figure 7. (a) Accuracy; (b) Loss

Conversely, MLP directly optimize the classification objective and thus may be better at exploiting discriminative relationships in this setting. The performance gap narrowed substantially after correlation-based feature selection after the correlation feature selection. After feature selection, MLP achieved the best overall result, with a maximum accuracy and F1-score of 99.5%. AE-MLP achieved a maximum accuracy of 99.2% and F1-score of 99.3%. The improvement was particularly striking for AE-MLP, with the best F1-score rising from 71.5% to 99.3%. Thus, the results suggest that the autoencoder produced a more discriminative representation after the input was reduced to a smaller selected subset of attributes. Because the selected set included metadata fields, the observed improvement cannot be attributed solely to traffic-behavior features. After feature selection, the MLP was the best performer since no marginal benefit was seen in the more complex architecture of AE-MLP. In the testbed described here, it seems more important to enhance the quality of the input features than to add a representation learning stage based on an autoencoder.

3.1 LIMITATIONS

In interpreting the data, the limitations of this study should be considered. First, the traffic characteristics obtained do not accurately capture the diversity of devices,

communication patterns, and network conditions of more complex smart home environments, as the dataset was built from a controlled smart home testbed with limited variety of devices and attack scenarios. Second, the ability of the model to detect different variations or other types of cyberattacks has not been evaluated since the study is limited to the binary detection of DDoS traffic, i.e., normal and DDoS traffic. Third, the evaluation was performed on MLP and AE-MLP setups with pre-defined hyper-parameters, hence it is still possible to enhance the performance using different architectures or optimization techniques. Fourth, the feature set used in this study included Flow ID, Src IP, and Timestamp, which may encode information related to specific devices, traffic-capture sessions, or attack timing in the controlled testbed. Consequently, the high predictive performance reported in this study may partially reflect the ability of the models to exploit environment-specific identifiers in addition to traffic-behavior patterns. Therefore, the results should not be interpreted as evidence that the proposed models will generalize unchanged to unseen devices, IP-address assignments, network environments, or attack sessions.

Future work should repeat the experiments after excluding identifier and metadata fields and should evaluate the models using device-disjoint, time-disjoint, and cross-environment data partitions. Furthermore, the evaluation was done under a single data collection setting and a single experimental split. Effects of random initialization, split variability and statistical uncertainty were not quantified via repeated runs, confidence intervals or cross-validation. Thus, the reported metrics should be considered as point estimates for tested testbed and experimental configuration rather than final estimates of general performance. Moreover, the models were trained and evaluated on a desktop-based experimental setup, but their computational requirements were not benchmarked on a resource-constrained IoT gateway or edge device. Hence, the reported classification results do not establish the deployment feasibility in terms of inference latency, memory consumption, CPU utilization or energy usage.

4. CONCLUSION

In this work, we studied the performance of MLP and AE-MLP for binary DDoS detection based on flow-based traffic gathered from a controlled smart-home IoT testbed. Results show that the correlation-based feature selection significantly increased the performance of both the models. Without feature selection, MLP outperformed AE-MLP with a maximum accuracy of 98.1% and F1-score of 98.3% while AE-MLP achieved a maximum accuracy of 68.7% and F1-score of 71.5%. The best overall performance was obtained by MLP with the highest accuracy and F1-score of 99.5% after feature selection. AE-MLP also presented a significant improvement with a maximum accuracy of 99.2% and the F1-score of 99.3%.

The results show that for the considered dataset, the quality and relevance of the input features were more important for detection performance than the added architectural complexity of the autoencoder. However, although AE-MLP demonstrated the greatest relative improvement after feature selection, it did not show better performance than the simpler MLP model. Hence, MLP is selected as the model in this work due to the best detection performance with a simpler architecture. The results are limited to the controlled smart-home testbed, binary normal vs. DDoS classification and the traffic scenarios considered. Future work should evaluate the

approach on unseen devices, different smart home environments, different variants of DDoS attacks, and resource constrained edge/gateway devices. Future work should also take inference latency, memory footprint, CPU utilization and generalization across environments into account.

ACKNOWLEDGEMENTS

This research was fully supported by the Faculty of Computer Science and the COMNETS Research Group at Universitas Sriwijaya.

REFERENCES

- [1] I. Lee, "Internet of Things (IoT) cybersecurity: Literature review and iot cyber risk management," *Futur. Internet*, vol. 12, no. 9, p. 157, 2020, doi: 10.3390/FI12090157.
- [2] Z. Chiba, N. Abghour, K. Moussaid, O. Lifandali, and R. Kinta, "A Deep Study of Novel Intrusion Detection Systems and Intrusion Prevention Systems for Internet of Things Networks," 2022. doi: 10.1016/j.procs.2022.10.124.
- [3] A. Jamalipour and S. Murali, "A Taxonomy of Machine-Learning-Based Intrusion Detection Systems for the Internet of Things: A Survey," *IEEE Internet Things J.*, vol. 9, no. 12, pp. 9444–9466, 2022, doi: 10.1109/JIOT.2021.3126811.
- [4] R. Alghamdi and M. Bellaiche, "Evaluation and Selection Models for Ensemble Intrusion Detection Systems in IoT," *Internet of Things*, vol. 3, no. 2, pp. 285–314, 2022, doi: 10.3390/iot3020017.
- [5] E. Muhati and D. B. Rawat, "Hidden-Markov-Model-Enabled Prediction and Visualization of Cyber Agility in IoT Era," *IEEE Internet Things J.*, vol. 9, no. 12, pp. 9117–9127, 2022, doi: 10.1109/JIOT.2021.3056118.
- [6] K. Choudhary and A. G. Dinker, "Detection of DDoS Attack Using Machine Learning Models," *2025 Int. Conf. Networks Cryptol.*, pp. 1515–1521, 2025, doi: 10.1109/netcrypt65877.2025.11102586.
- [7] J. Rejito, D. Stiawan, A. Alshaflut, and R. Budiarto, "Machine learning-based anomaly detection for smart home networks under adversarial attack," *Comput. Sci. Inf. Technol.*, vol. 5, no. 2, pp. 122–129, 2024, doi: 10.11591/csit.v5i2.p122-129.
- [8] M. Wang, N. Yang, and N. Weng, "Securing a Smart Home with a Transformer-Based IoT Intrusion Detection System," *Electron.*, vol. 12, no. 9, 2023, doi: 10.3390/electronics12092100.
- [9] M. Amru *et al.*, "Network intrusion detection system by applying ensemble model for smart home," *Int. J. Electr. Comput. Eng.*, vol. 14, no. 3, pp. 3485–3494, 2024, doi: 10.11591/ijece.v14i3.pp3485-3494.
- [10] N. Chavan, M. Kukreja, G. Jagwani, N. Nishad, and N. Deb, "DDoS Attack Detection and Botnet Prevention using Machine Learning," *8th Int. Conf. Adv. Comput. Commun. Syst. ICACCS 2022*, vol. 1, pp. 1159–1163, 2022, doi:

- 10.1109/ICACCS54159.2022.9785247.
- [11] R. R. Papalkar and A. S. Alvi, "Analysis of Defense Techniques for DDOS Attacks in IoT – A Review," *ECS Trans.*, vol. 107, no. 1, pp. 3061–3068, 2022, doi: 10.1149/10701.3061ecst.
 - [12] U. Islam *et al.*, "Detection of Distributed Denial of Service (DDoS) Attacks in IOT Based Monitoring System of Banking Sector Using Machine Learning Models," *Sustain.*, vol. 14, no. 14, p. 8374, 2022, doi: 10.3390/su14148374.
 - [13] M. H. Ali *et al.*, "Threat Analysis and Distributed Denial of Service (DDoS) Attack Recognition in the Internet of Things (IoT)," *Electron.*, vol. 11, no. 3, 2022, doi: 10.3390/electronics11030494.
 - [14] L. Almahadeen *et al.*, "Enhancing Threat Detection in Financial Cyber Security Through Auto Encoder-MLP Hybrid Models," *Int. J. Adv. Comput. Sci. Appl.*, vol. 15, no. 4, pp. 924–933, 2024, doi: 10.14569/IJACSA.2024.0150495.
 - [15] U. S. Chanu, K. J. Singh, and Y. J. Chanu, "A dynamic feature selection technique to detect DDoS attack," *J. Inf. Secur. Appl.*, vol. 74, p. 103445, May 2023, doi: 10.1016/j.jisa.2023.103445.
 - [16] P. Musikawan, Y. Kongsorot, I. You, and C. So-In, "An Enhanced Deep Learning Neural Network for the Detection and Identification of Android Malware," *IEEE Internet Things J.*, vol. 10, no. 10, pp. 8560–8577, 2023, doi: 10.1109/IIOT.2022.3194881.
 - [17] Y. Wei, J. Jang-Jaccard, F. Sabrina, A. Singh, W. Xu, and S. Camtepe, "AE-MLP: A Hybrid Deep Learning Approach for DDoS Detection and Classification," *IEEE Access*, vol. 9, pp. 146810–146821, 2021, doi: 10.1109/ACCESS.2021.3123791.
 - [18] M. Wang, Y. Lu, and J. Qin, "A dynamic MLP-based DDoS attack detection method using feature selection and feedback," *Comput. Secur.*, vol. 88, p. 101645, Jan. 2020, doi: 10.1016/j.cose.2019.101645.
 - [19] M. Flores, D. Heredia, R. Andrade, and M. Ibrahim, "Smart Home IoT Network Risk Assessment Using Bayesian Networks," *Entropy*, vol. 24, no. 5, p. 668, 2022, doi: 10.3390/e24050668.
 - [20] N. Butt *et al.*, "Intelligent Deep Learning for Anomaly-Based Intrusion Detection in IoT Smart Home Networks," *Mathematics*, vol. 10, no. 23, 2022, doi: 10.3390/math10234598.
 - [21] K. Al-Begain, M. Khan, B. Alothman, C. Joumaa, and E. Alrashed, "A DDoS Detection and Prevention System for IoT Devices and Its Application to Smart Home Environment," *Appl. Sci.*, vol. 12, no. 22, 2022, doi: 10.3390/app122211853.
 - [22] D. Stiawan and N. Afifah, "Smart Home Attack Topology (IPv4)," Feb. 2025, *Zenodo*. doi: 10.5281/zenodo.14807844.
 - [23] V. S. Pandi, S. D, and R. Geetha, "Role of Feature Selection and Dimensionality Reduction in Hybrid Learning Systems for Threat Detection," *Hybrid Mach. Learn. Tech. Data Encryption, Anom. Detect. Zero-Day Attack Prev.*, pp. 66–91, 2025, doi: 10.71443/9788197933608-03.
 - [24] M. S. Akhtar and T. Feng, "Malware Analysis and Detection Using Machine Learning Algorithms," 2022, *mdpi.com*. doi: 10.3390/sym14112304.

- [25] J. Singh, D. Thakur, T. Gera, B. Shah, T. Abuhmed, and F. Ali, "Classification and Analysis of Android Malware Images Using Feature Fusion Technique," *IEEE Access*, vol. 9, pp. 90102–90117, 2021, doi: 10.1109/ACCESS.2021.3090998.