International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 04 | April - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

DoS Attack Detection using Edge Machine Learning

Prof. A. H. Kagne, Harshal Gogawale, Bhavesh Chopade, Rishabh Indalkar, Avinash Bhite

Prof. Ashvini H. Kagne, Comp Engin, SAE (ashvinikagne.sae@sinhgad.edu)

Harshal Satish Gogawale, Comp Engin, SAE (harshalgogawale.sae.comp@gmail.com)

Bhavesh Hanumant Chopade, Comp Engin, SAE (bhaveshchopade.sae.comp@gmail.com)

Rishahb Santosh Indalkar, Comp Engin, SAE (rishabhindalkar.sae.comp@gmail.com)

Avinash Sunil Bhite, Comp Engin, SAE (avinashbhite.sae.comp@gmail.com)

ABSTRACT

The Denial-of-Service (DoS) attacks are among the most common and malicious types of cybersecurity attacks. These attacks attempt to disrupt the normal traffic and function of a targeted server or network. Although the launch of Internet Protocol version six (IPv6) addressed the issue of IPv4's address depletion, but also mandated the use of Internet Control Message Protocol version six (ICMPv6) messages in newly introduced features such as the Neighbour Discovery Protocol (NDP). This has exacerbated existing network attacks including ICMPv6-based Denial of Service (DoS) attacks and its variant form Distributed Denial of Service (DDoS) attack. Intrusion Detection Systems (IDS) aimed at tackling security issues raised by ICMPv6-based DoS and DDoS attacks have been reviewed by researchers and a general classification of existing IDSs was proposed as anomaly-based and signature-based. However, it is incredibly hard to see the overall picture of IDSs based on Machine Learning (ML) techniques with such a classification, as there is a lack of a more detailed view of the ML approach, classifiers, feature selection techniques, datasets, and different evaluation metrics.

Key Words: Edge Machine Learning, Denial of Service, Attack, Detection, Long Short Term Memory(LSTM).

1. INTRODUCTION

The global permanent deployment of IPv6 has attracted the interest of researchers to review the security issues raised by numbers of attacks and one of them is the DoS attack using ICMPv6 messages [1]. ICMPv6 has been given a vital role by the designers of IPv6 as compared to its previous version IPv4 [2]. example, the NDP which uses ICMPv6 messages has been introduced by IPv6 as a new protocol for The Stateless Address Auto Configuration (SLAAC), discovering link-layer addresses, routers discovery, and Duplicate Address Detection (DAD) processes [3]. However, these features are subjected to exploitation by the attackers to perform DoS attacks [4]. Further, many to one dimension, which is an intrinsic characteristic of a DDoS attack, is still possible in IPv6 using ICMPv6 Echo messages [

To the best of our knowledge, there is no review paper specifically focused on IDSs based on ML techniques for ICMPv6-based DoS and DDoS attacks detection. The ML-based intrusion detection for ICMPv6-based DoS and DDoS attacks is a relatively new field, and research in this area is gaining momentum. Therefore, the main contributions of this article are:

(i) the review and classification of existing ML-based IDSs for the detection of ICMPv6-based DoS and DDoS attacks, (ii) the identification of open challenges as future research directions, (iii) proposed blockchain applicability in the ensemble framework as one of the possible solutions to these challenges, and (iv) the classification of ICMPv6 vulnerabilities that are revealed by exploitation techniques and not addressed by previous reviews.

2. LITERATURE SURVEY

Sr. No.	Reference Name	Seed Idea / Work Description	Publish Year
1.	**IEEE : Ngoc Soung Huynh** DoS Attack Detection Using Edge Machine Learning	Develops lightweight algorithms to prevent DoS attacks on Edge devices	January 2023
2.	**Science Direct : S.V. Jansi Rani** Detection of DDoS attacks in D2D communications using machine learning	Detecting and preventing SYN and Slow loris DDoS attacks in D2D communication networks.	November 2022
3.	**IEEE : R. Bharathi** Investigation on Efficient Machine Learning Algorithm for DDoS Attack Detection	This paper explores detecting complex DDoS attacks in IoT networks using machine learning.	April 2023

3. RELATED WORK

The role of ML in the intrusion detection field is to build a model for a multinomial classifier problem that can classify network events as normal or attack events, as in the case of DoS and DDoS attacks. Recent research shows that, compared to traditional IDS solutions, researchers have shifted their focus towards developing ML-based IDS solutions [36]. In the literature, ML-based IDS models achieved interesting results; from 86.53% [37] to over 99% [38] in detection accuracy and a significant decrease in false-positive from around 4% [39] to 0.01% [40].

Class Type:

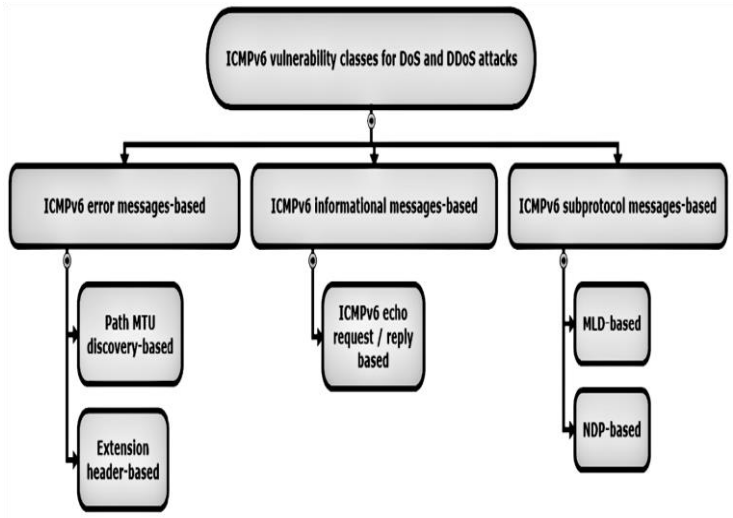
The classification approach using ML techniques can be categorized as single, ensemble, or hybrid depending on the number and the way in which different techniques work together to solve a problem

1) SINGLE

IDS model under this category can be designed by utilizing only one technique such as clustering, classification, or association. In recent research, classifiers such as SVM, NN, Decision Trees (CART, C 4.5, and J48), NB, and KNN have been used to design single classifier-based IDS models.

2) HYBRID

In this category, an IDS model typically combines two or more functional components with the intuition to improve performance as an advantage over a single classifier approach. Classifiers implementation in this approach works in two stages, first one aims at optimizing the learning



3) ENSEMBLE

Ensemble models add another dimension to achieve performance intuitive benefits by combining the opinions of multiple learners. Ensemble methods, with access to multiple processors, are the ideal choice for training and testing time efficiency because they are inherently parallel in nature. The ensemble model's implementation can be accomplished in two ways, one is training multiple classifiers on the same dataset and the other is training a single classifier on multiple datasets. After the training phase, the data item is assigned to the class to which

the majority of classifiers point at the time of testing.

4. PERFORMANCE MATRIX

The Cross-validation and supplied test set approaches have been applied to evaluate the performance of an IDS by obtaining the results in the form of different performance evaluation metrics [45]. Experimental results using these metrics have been used by researchers to compare their results with already existing approaches.

True Positive (TP): An attack traffic instance correctly classified as belonging to attack class.

False Positive (FP): A normal traffic instance incorrectly classified as belonging to attack class.

True Negative (TN): A normal traffic instance correctly classified as a normal class instance.

False Negative (FN): An attack traffic instance incorrectly classified as a normal class instance.

Detection Accuracy (DA): Detection accuracy measures the ratio of correct predictions over the total number of instances evaluated.

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

Error Rate (ER): Also referred to as misclassification error, measures the ratio of incorrect predictions over the total number of instances evaluated.

$$ER = \frac{FP + FN}{(TP + FP + TN + FN)} \quad (2)$$

True Positive Rate (TPR): The intrusions which are correctly classified as an attack are also known as sensitivity.

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

False Positive Rate (FPR): Often referred to as false alarm. These are the normal patterns that were incorrectly classified as an attack.

$$FPR = \frac{FP}{(FP + TN)} \quad (4)$$

True Negative Rate (TNR): The normal patterns that were correctly predicted as normal are also known as specificity.

$$TNR = \frac{TN}{(TN + FP)} \quad (5)$$

Precision (P): The positive patterns that are correctly predicted from the total predicted patterns in a positive class.

$$P = \frac{TP}{(TP + FP)} \quad (6)$$

Recall (R): The fraction of positive patterns that are correctly classified.

$$R = \frac{TP}{(TP + TN)} \quad (7)$$

F-Measure (FM): This metric represents the harmonic mean between recall and precision values.

$$FM = \frac{2 * P * R}{(P + R)} \quad (8)$$

Matthews Correlation Coefficient (MCC): This metric measures the correlation between the predicted results and the real data.

$$\mathbf{MCC} = \frac{(TP.TN) - (FP.FN)}{\sqrt{(TP + FP).(TP + FN).(TN + FP).(TN + FN)}}$$

5. METHOD OF IMPLEMENTATION

1.Traning phase

a. The training phase of a machine learning model involves several steps, including data processing, feature scaling, and handling missing values.

b. The model architecture includes LSTM layers, a dense layer, and a loss function.

c. The input shape preparation ensures the data conforms to the expected shape.

d. The final hidden state is outputted using the final hidden state.

e. The training process involves fitting the model on training data using the fit method, with validation data used to monitor performance and adjust hyper parameters if needed.

f. The loss function measures the model's performance in multi-class classification tasks

2.Testing

a. The evaluation process involves preprocessing test data into the same format as the training data, applying the same feature scaling transformation, and evaluating the model's performance using metrics like accuracy, precision, recall, and F1-score.

b. The confusion matrix is also analyzed to understand classification performance and detect biases.

3. Detection

a. The system uses a trained LSTM model for real-time detection of DoS attacks, with the system configured to generate alerts or trigger automated responses.

b. Post-detection analysis refines the model's accuracy by analyzing false positives/negatives.

c. Continuous monitoring and retraining mechanisms are implemented to adapt to new attack patterns and evolving network conditions.

- In our day-to-day life the usage of mobile phones has increased in restricted areas such as exam venues, places of important meetings, offices, conference halls, prisons, etc., and the hidden wireless cameras intrial rooms and hotels, public toilets.
- The radio frequency signals are transmitted from wireless cameras and mobile phones during the video transmission, incoming calls and outgoing calls, and text messages from one gadget to another. The detector will detect the transmitted signal and then it is given as input to AT mega 8 microcontrollers.
- As soon as the Arduino microcontroller receives the signal, it will turn ON the beep alarm and the information will be displayed on the LCD and also sends messages like mobile detected with location, room number, etc. to the mobile number stored in the microcontroller by using the GSM module.
- This system will be used to detect the mobile phones

and the wireless hidden camera present in a room by the radio frequency signals that are transmitted by them.

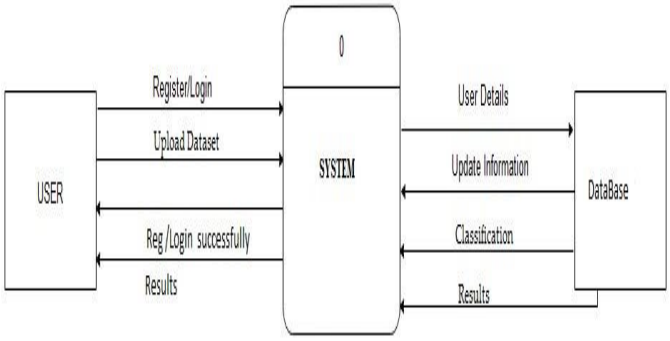
Machine Learning Model Development:

Real-time network traffic data captured using Wireshark can be parsed with DPKT module in Python or CICFlowMeter traffic flow analyzer to extract features on packet metadata and packet flow statistics [2]. In this study, we extracted two features from captured network traffic data: (1) frame length, and (2) packet inter-arrival time. We used the preprocessed dataset to train Support Vector Machine (SVM) and Logistic Regression (LR) models.

Both SVM and LR are relatively lightweight machine learning models that are robust for datasets with fewer features. SVM is more lightweight than most models in terms of training time, model size, inference time and ease of implementation, especially on constrained environments such as micro-controllers. SVM models have also proven robust when analyzing larger numbers of features (up to 34) as experimented in study [12]. LR models are easy to implement for binary as well as multi-classification problems to account for different types of DDoS attacks at the same time. Compared to other models, it is also less prone to overfitting.

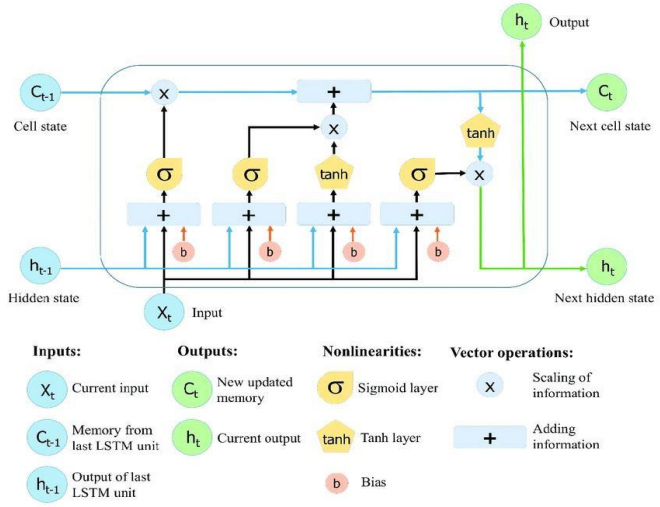
Besides real-time captured data, we also used DoS attack data from the CICIoT2023 dataset to train the aforementioned models. This dataset contains more types of DoS attacks experimented on a wide range of IoT devices. As we only experimented with a simplified De-authentication DoS attack on a small Wi-Fi network, the CICIoT2023 dataset assists in testing the applicability and generalizability of our models over the most common and recent types of DoS attacks. The features that we used for analyzing the CICIoT2023 Dataset are flow duration, frame length, protocol type, time to live, flow rate, total length of frames in flow, and inter-arrival time (IAT).

6.SYSTEM ARCHITECTURE



\

7. METHODOLOGY



Long Short-Term Memory (LSTM) is a Recurrent Neural Network (RNN) architecture developed by Hochreiter and Schmidhuber to overcome the limitations of traditional RNNs in handling sequential data. LSTM introduces a unique memory cell concept, controlled by three critical components: the input gate, forget gate, and output gate.

These gates collectively determine how information flows into, out of, and is retained within the memory cell. The memory cell in LSTM is the core element that sets it apart from traditional RNNs. It can store and preserve information over long sequences, making it ideal for tasks with extended dependencies. The input gate regulates what information should be added to the memory cell at the current time step, while the forget gate determines what information from the previous memory cell state should be discarded. The output gate controls what information from the memory cell should be passed as the output at the current time step, considering both the current input and the previous

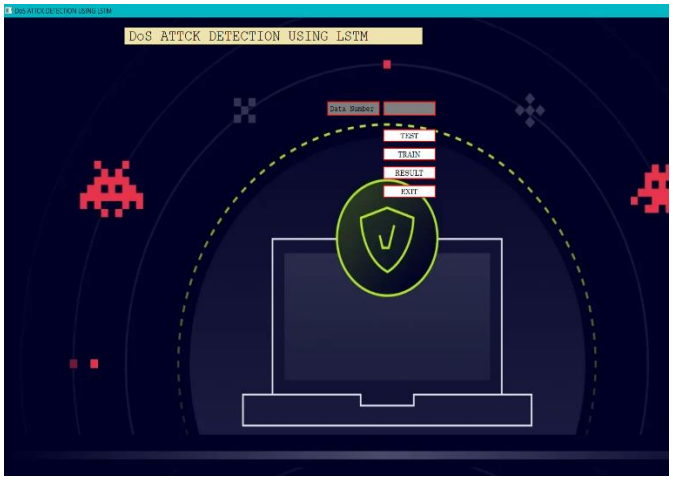
state.

These gates allow LSTM networks to selectively capture and store information from past time steps, discard irrelevant details, and output contextually relevant information.

This capability enables LSTMs to effectively learn and utilize long-term dependencies in sequential data.

LSTM networks find applications in various domains, including language translation, speech recognition, time series forecasting, and text generation. They leverage their memory cell and gating mechanisms to overcome the vanishing gradient problem and effectively capture long-term dependencies in sequential data, making them a powerful tool for a wide range of applications involving time series, speech, and text data.

8. System Components



GUI (Graphical User Interface) Setup

- Built using PyQt5 and Tkinter.
- Provides buttons for:
  - TRAIN – Trains the model.
  - TEST – Tests the model with selected input.
  - RESULT – Visualizes attack statistics.
  - EXIT – Quits the application.

1. Training the Model (TRAIN Button)

- Loads a dataset (dataset_sdn.csv) containing network traffic data.
- Preprocesses the data:
  - Drops non-numeric/irrelevant columns (dt, src, dst, label).
  - Converts categorical data using one-hot encoding.
  - Scales features with Standard Scaler.
- Builds an LSTM model using TensorFlow Keras:
  - 1 LSTM layer with 100 units and ReLU activation.
  - 1 Dense output layer.
  - Trained to minimize mean squared error between predictions and true labels (label = 0 for normal, 1 for attack).
- Trains on 10000 samples, reshaped for LSTM input.

2. Testing the Model (TEST Button)

- Accepts a sample index number from the user.
- Loads the pre-trained model (LSTM.keras) and test data (X.dat, y.dat) from disk.
- Feeds selected test data sample into the LSTM model.
- Predicts whether the sample is:
  - 0 → Normal
  - 1 → DoS Attack
- Displays the result in the console.

3. Results and Visualizations (RESULT Button)

- Uses matplotlib and seaborn to:
  - Show how many requests came from each IP address.
  - Show the number of malicious vs normal requests per IP.
  - Analyze requests by protocol.
  - Pie chart of benign vs malicious request distribution.

Files Used:

- dataset_sdn.csv: Network traffic dataset.
- X.dat, y.dat: Pickled feature and label data for test input.
- LSTM.keras: Pickled trained model.
- Images (e.g., gray.jpg, white.jpg, y1.jpg): Used for background styling of the GUI.

9. Result of LSTM Model

For LSTM, the model's performance during training and validation is depicted in Figure 22. Over time, the training and validation loss decreases, while the validation accuracy remains high, indicating that the model generalises well to new data. Table 9 displays the class-wise performance metrics of an LSTM model trained with a particular dataset.

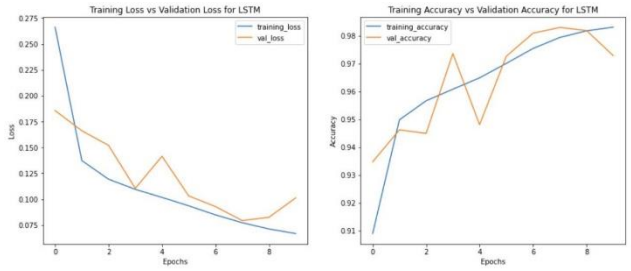


FIGURE 22. Training and validation performance LSTM.

TABLE 9. Class-wise performance metrics of LSTM.

Class	Accuracy	Precision	Recall	F1
Benign	99.93	85.99	99.03	92.05
TFTP	99.39	98.89	95.422	97.12
UDP+SSDP	99.62	99.85	98.39	99.12
LDAP	99.80	98.11	99.92	99.00
NetBIOS+Portmap	99.91	99.98	99.62	99.80
SYN	99.86	99.06	99.68	99.37
MSSQL	99.51	11.68	95.16	20.81
HULK	99.52	99.76	97.15	98.44
RUDY	99.93	98.62	99.29	98.95
GoldenEye	99.57	86.79	98.36	92.21
SQLi	99.99	99.93	99.60	99.77
MITM	99.97	95.61	98.14	96.86
Dictionary	99.98	93.24	98.71	95.89

The model's accuracy ranges from 99.39% to 99.99%, indicating that it detects various types of network traffic with high precision. Precision measures the model's ability to distinguish between genuine positives and false positives. The precision values range from 11.68% (for MSSQL) to 99.98% (for Dictionary), indicating that the model accurately identifies positive samples for the majority of classes, with the exception of MSSQL. The recall metric measures the model's ability to correctly identify all affirmative samples, with values ranging from 95.16 to 99.92% (for MSSQL and LDAP, respectively). The high recall values indicate that the model is capable of identifying the majority of positive samples for all classes. The F1 score, which is the harmonic mean of precision and recall, represents the model's performance as a whole. F1 values range between 20.81% (MSSQL) and 99.77% (SQLi). The low F1 value for MSSQL suggests that the model needs to identify this traffic class. The LSTM model's confusionmatrix is depicted in Figure23.The diagonal values represent the number of properly classified samples for each class, whereas the off-diagonal values represent misclassifications. The confusion matrix indicates that the model has difficulty correctly identifying MSSQL class samples, which is consistent with the class's low precision and F1 score. Except for MSSQL, the LSTM model appears to function well in detecting various types of network traffic. In conclusion, the LSTM model demonstrates a high degree of accuracy, precision, recall, and F1 score when identifying different categories of network traffic. While there is room for development in identifying MSSQL traffic, the model's

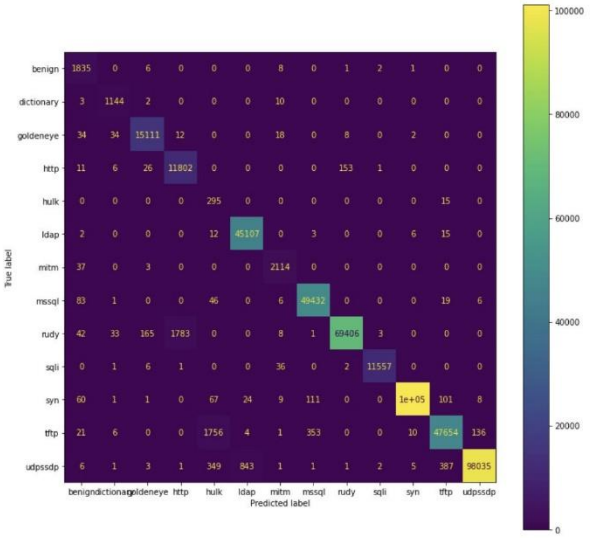


FIGURE 23. LSTM confusion matrix.

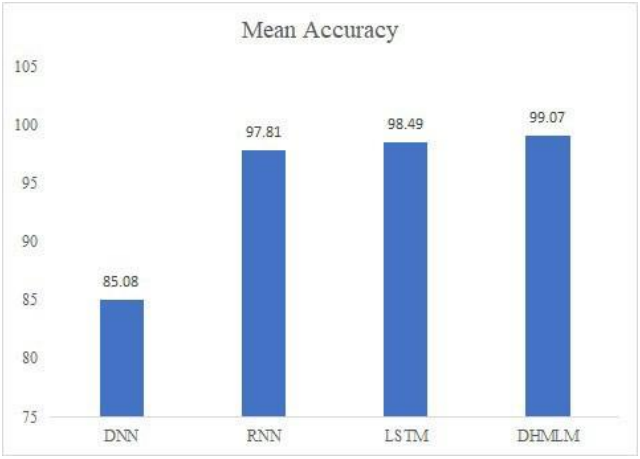


FIGURE 24. Mean accuracies of investigated approaches.

generalisation ability suggests it could be a useful classification tool for network traffik.

10. CONCLUSION

This review presents the current state of the art of ML-based IDSs for the detection of ICMPv6-based DoS and DDoS attacks. Based on the classifier's design we present a classification of existing ML-based IDSs into single and hybrid categories for the detection of ICMPv6-based DoS and DDoS attacks. Table 3 gives an overall idea about the ML approach, classifier used, and feature selection techniques along with the result produced by each model. Although existing ML-based IDS models such as [50] and [61] have promised high DA in the detection of ICMPv6 Echo messages-based DDoS anomalies, the growing scale of the problem with respect to the analysis of traffic produced by DDoS attacks can have a significant impact on the performance of ML-based IDSs using sequential hardware. Therefore, IDS models based on ML techniques is still in its infancy stage for the detection of ICMPv6-based DoS and DDoS attacks.

11. REFERENCES

[1] A. Conta, S. Deering, and M. Gupta, *Internet Control Message Protocol (ICMPv6)*, document RFC 4443, IETF, 2006.

[2] T. Narten, W. Simpson, E. Nordmark, and H. Soliman, *Neighbor Discovery for IP Version 6 (IPv6)*, document RFC 4861, IETF, 2007.

[3] A. K. Al-Ani, M. Anbar, A. Al-Ani, and D. R. Ibrahim, ''Match-prevention technique against denial-of-service attack on address resolution and duplicate address detection processes in IPv6 link-local network,'' *IEEE Access*, vol. 8, pp. 27122–27138, 2020.

[4] R. M. A. Saad, M. Anbar, and S. Manickam, ''Rule-based detection technique for ICMPv6 anomalous behaviour,'' *Neural Comput. Appl.*, vol. 30, no. 12, pp. 3815–3824, Dec. 2018.

[5] O. E. Elejla, M. Anbar, and B. Belaton, ''ICMPv6-based DoS and DDoS attacks and defense mechanisms: Review,'' *IETE Tech. Rev.*, vol. 34, no. 4, pp. 390–407, Jul. 2017.

[6] O. E. Elejla, B. Belaton, M. Anbar, and A. Alnajjar, ''Intrusion detection systems of ICMPv6-based DDoS attacks,'' *Neural Comput. Appl.*, vol. 30, no. 1, pp. 45–56, Jul. 2018.

[7] A. H. Bdair, R. Abdullah, S. Manickam, and A. K. Al-Ani, ''Brief of intrusion detection systems in detecting ICMPv6 attacks,'' in *Proc. 6th Comput. Sci. Technol. (ICCST)*, Kota Kinabalu, Malaysia, vol. 603. Singapore: Springer, 2020, pp. 199–213.

[8] O. E. Elejla, B. Belaton, M. Anbar, B. Alabsi, and A. K. Al-Ani, ''Comparison of classification algorithms on ICMPv6-based DDoS attacks detection,'' in *Computational Science and Technology* (Lecture Notes in Electrical Engineering), vol. 481. Singapore: Springer, 2019, pp. 347–357.

[9] N. K. Kasabov and Q. Song, ''DENFIS: Dynamic evolving neural-fuzzy inference system and its application for time-series prediction,'' *IEEE Trans. Fuzzy Syst.*, vol. 10, no. 2, pp. 144–154, Apr. 2002.

[10] Google. (2020). *Percentage of Users that Access Google Over IPv6*.

[11] J. B. Ard, ''Internet protocol version six (IPv6) at UC Davis: Traffic analysis with a security perspective,'' M.S. thesis, Univ. California, Davis, CA, USA, 2012.

[12] R. C. Baishya and D. K. Bhattacharyya, ''A complete detection and mitigation framework to protect a network from DDoS attacks,'' *IETE J. Res.*, pp. 1–18, Apr. 2019,

[13] A. K. Al-Ani, M. Anbar, S. Manickam, C. Y. Wey, Y.-B. Leau, and A. Al-Ani, ''Detection and defense mechanisms on duplicate address detection process in IPv6 link-local network: A survey on limitations and requirements,'' *Arabian J. Sci. Eng.*, vol. 44, no. 4, pp. 3745–3763, Apr. 2019.

[14] Miniwatts Marketing Group. (Mar. 2020). *World Internet Usage and Population Statistics*. [Online]. Available: https://internetworldstats. com/stats.htm

[15] S. Deering and R. Hinden, *Internet Protocol, Version 6 (IPv6) Specification*, document RFC 2460, IETF, 1998.

[16] Internet Society. (Jun. 2012). *World IPv6 Launch*. [Online]. Available: https://www.worldipv6launch.org/

[17] S. Krishnan and S. Frankel, *IP Security (IPsec)*, document RFC 6071, IETF, 2011.

[18] J. Mirkovic and P. Reiher, ''A taxonomy of DDoS attack and DDoS defense mechanisms,'' *ACM SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 2, pp. 39–53, Apr. 2004.

[19] O. Osanaiye, K.-K.-R. Choo, and M. Dlodlo, ''Distributed denial of service (DDoS) resilience in cloud: Review and conceptual cloud DDoS mitigation framework,'' *J. Netw. Comput. Appl.*, vol. 67, pp. 147–165, May 2016.

[20] F. A. Barbhuiya, S. Biswas, and S. Nandi, ''Detection of neighbor solicitation and advertisement spoofing in IPv6 neighbor discovery protocol,'' in *Proc. 4th Int. Conf. Secur. Inf. Netw. (SIN)*, New York, NY, USA, 2011, pp. 111–118.

[21] J. Gao and Y. Chen, ''Detecting DOS/DDOS attacks under IPv6,'' in *Proc. Int. Conf. Cybern. Informat.*, vol. 163. New York, NY, USA: Springer, 2014, pp. 847–855.

[22] H. Scott and V. Erick, ''Local network security,'' in *IPv6 Security* (Network Technology Series), 1st ed. Indianapolis, IN, USA: Pearson, 2008, pp. 183–193.

[23] Supriyanto, I. H. Hasbullah, R. K. Murugesan, and S. Ramadass, ''Survey of Internet protocol version 6 link local communication security vulnerability and mitigation methods,'' *IETE Tech. Rev.*, vol. 30, no. 1, pp. 64–71, 2013.

[24] J. McCann, S. Deering, and J. Mogul, *Path MTU Discovery for IP Version 6*, document RFC 1981, IETF, 1996.----------

[25] N. C. Arjuman and S. Manickam, ‘‘A review on ICMPv6 vulnerabilities and its mitigation techniques: Classification and art,’’ in *Proc. Int. Conf. Comput., Commun., Control Technol. (ICT)*, Kuching, Malaysia, Apr. 2015, pp. 323–327.

[26] C. E. Martin and J. H. Dunn, ‘‘Internet protocol version 6 (IPv6) protocol security assessment,’’ in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Orlando, FL, USA, Oct. 2007, pp. 1–7.

[27] R. Vida and L. Costa, Eds., *Multicast Listener Discovery Version 2 (MLDv2) for IPv6*, document RFC 3810, IETF, 2004.

[28] G. Bansal, N. Kumar, S. Nandi, and S. Biswas, ‘‘Detection of NDP based attacks using MLD,’’ in *Proc. 5th Int. Conf. Secur. Inf. Netw. (SIN)*, New York, NY, USA, 2012, pp. 163–167.

[29] J. Ullrich, K. Krombholz, H. Hobel, A. Dabrowski, and E. Weippl, ‘‘IPv6 security: Attacks and countermeasures in a nutshell,’’ in *Proc. 8th USENIX Conf. Offensive Technol. (WOOT)*. San Diego, CA, USA: USENIX Association, 2014, p. 5.

[30] M. Anbar, R. Abdullah, R. M. Saad, E. Alomari, and S. Alsaleem, ‘‘Review of security vulnerabilities in the IPv6 neighbor discovery protocol,’’ in *Information Science and Applications* (Lecture Notes in Electrical Engineering), vol. 376. Singapore: Springer, 2016, pp. 603–612.

[31] M. Anbar, R. Abdullah, R. M. A. Saad, and I. H. Hasbullah, ‘‘Review of preventive security mechanisms for neighbour discovery protocol,’’ *Adv. Sci. Lett.*, vol. 23, no. 11, pp. 11306–11310, Nov. 2017.

.