

A GRU-based approach for botnet detection using deep learning technique

Suchetha G.¹, Pushpalatha K.²

¹Department of Information Science and Engineering, Sahyadri College of Engineering and Management,
Visvesvaraya Technological University, Belagavi, India

²Department of CSE (Artificial Intelligence and Machine Learning), Sahyadri College of Engineering and Management,
Visvesvaraya Technological University, Belagavi, India

Article Info

Article history:

Received Jun 22, 2024

Revised Oct 29, 2024

Accepted Nov 11, 2024

Keywords:

Botnet

Gated recurrent unit

K-nearest neighbors

Long short-term memory

Random forest

SelectKBest

ABSTRACT

The increasing volume of network traffic data exchanged among interconnected devices on the internet of things (IoT) poses a significant challenge for conventional intrusion detection systems (IDS), especially in the face of evolving and unpredictable security threats. It is crucial to develop adaptive and effective IDS for IoT to mitigate false alarms and ensure high detection accuracy, particularly with the surge in botnet attacks. These attacks have the potential to turn seemingly harmless devices into zombies, generating malicious traffic that disrupts network operations. This paper introduces a novel approach to IoT intrusion detection, leveraging machine learning techniques and the extensive UNSW-NB15 dataset. Our primary focus lies in designing, implementing, and evaluating machine learning (ML) models, including K-nearest neighbors (KNN), random forest (RF), long short-term memory (LSTM), and gated recurrent unit (GRU), against prevalent botnet attacks. The successful testing against prominent Bot-net attacks using a dedicated dataset further validates its potential for enhancing intrusion detection accuracy in dynamic and evolving IoT landscapes.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Suchetha G.

Department of Information Science and Engineering, Sahyadri College of Engineering and Management

Visvesvaraya Technological University

Belagavi-590018, India

Email: suchethag87@gmail.com

1. INTRODUCTION

The rise of botnets poses one of the most significant challenges to cybersecurity, especially within the rapidly expanding internet of things (IoT) ecosystem [1]. These networks of compromised devices enable a wide range of cybercrimes, including distributed denial of service (DDoS) attacks, data breaches, and spam campaigns. Traditional cybersecurity measures have proven insufficient, especially given the volume and diversity of IoT devices, many of which have limited security, making them prime targets for botnet infections. In response to these challenges, the cybersecurity community has increasingly adopted machine learning (ML) and deep learning (DL) techniques, such as K-nearest neighbors (KNN), random forest (RF) [2], long short-term memory (LSTM) [3], and gated recurrent unit (GRU). While these models show potential, they face significant limitations. Traditional ML models often struggle with the variability of IoT network traffic, leading to reduced accuracy. Meanwhile, existing DL models, though powerful, require substantial computational resources, making them less feasible for real-time applications in resource-constrained IoT environments [4].

Improving feature selection is crucial for enhancing model performance. Addressing these challenges is essential for advancing botnet detection systems. This research aims to close these gaps by developing a

scalable botnet detection framework that integrates GRU for sequential data processing with optimized feature selection. By focusing on improving the scalability and accuracy of intrusion detection systems (IDS) in IoT environments, this study contributes to the broader effort to enhance cybersecurity measures against the evolving threat of botnets. Despite advancements, challenges such as scalability, computational efficiency, and real-time applicability in diverse IoT environments persist. Our study builds on prior research [5], [6] by addressing these constraints, ultimately aiming to develop a more robust and scalable botnet detection model that can be effectively deployed in real-world IoT scenarios.

The UNSW-NB15 [7] dataset is used in the proposed approach to detect botnets in IoT scenarios. The dataset offers a thorough depiction of network traffic scenarios and consists of 49 variables divided into nine classes. To improve model applicability, data preprocessing techniques include feature selection, clamping extreme values, log transformation, and lowering cardinality in categorical features. This research employs the SelectKBest method with Chi-Square scoring to identify critical features, optimizing the GRU model's ability to detect botnets effectively. The effectiveness of machine learning algorithms like GRU, LSTM, and RF in real-time intrusion detection is assessed. The approach seeks to mitigate the growing threat of botnet assaults in IoT contexts and enhance the efficacy of current IDS by combining cutting-edge models and feature selection approaches.

Researchers have been looking into several techniques to improving IDSs. According to the study, it was discovered that applying XGBoost for feature selection improved IDS performance on the UNSW-NB15 dataset, resulting in higher accuracy and lower complexity, particularly for binary classification tasks [8]. An integrated rule-based IDS demonstrated good accuracy and lower false alarm rates on both the UNSW-NB15 dataset and a real-time dataset from NIT Patna, but it was limited in its capacity to detect zero-day attacks due to its reliance on predetermined signatures [9]. Furthermore, a two-stage anomaly-based strategy employing recursive feature removal and RFs had promise, but it did not significantly increase detection, particularly with decision trees (DT) and Naive Bayes (NB) classifiers [10]. Mohy-Eddine *et al.* [11] created a network intrusion detection system (NIDS) for IoT contexts utilizing a KNN classifier along with feature selection approaches such as PCA, univariate statistical tests, and the Genetic Algorithm. This model achieved 99.99% accuracy and lowered forecast time from over 51,000 seconds to less than one minute, although it had limitations in binary classification, dataset coverage, and sensitivity to noisy data. Venkatachalam and Jacob [12] used recursive feature elimination (RFE) with RF on the UNSW-NB15 dataset, lowering features from 45 to 4 while attaining 98.3% accuracy. However, because of the narrow feature set, our technique may have missed complicated patterns. Another approach [13], explores methods for detecting botnets, focusing on behavior-based analysis and flow-based features. The suggested multilayer framework, which uses a variety of classification methods, exhibits good accuracy rates. One more approach [14], tackles the scant knowledge of the behavior of harmful botnets, exposing their substantial contribution to undesired internet traffic and a variety of victim domains.

Our study demonstrates the effectiveness of a GRU-based deep learning model for botnet detection, enhanced by SelectKBest with the Chi-square test. GRUs outperform traditional machine learning models in binary classification by recognizing long-term dependencies in sequential data. This research aims to develop a scalable and efficient botnet detection system that improves accuracy, reduces false positives, and ensures applicability in resource-constrained IoT environments. By addressing computational efficiency and real-time detection, this study strengthens cybersecurity against evolving botnet threats. The paper is organized as follows: section 2 covers the methodology, section 3 discusses results and conclusions, and section 4 presents the final conclusion.

2. METHOD

To enhance the detection of botnets in IoT environments, we have developed a novel cybersecurity system using a GRU-SelectKBest deep learning model. This section describes the systematic steps involved in detecting botnets using a GRU-based approach. The techniques include data collection and preprocessing, feature selection, model architecture design, training, and evaluation. Figure 1 depicts the suggested design, ensuring a thorough understanding of the underlying processes.

2.1. Dataset

The UNSW-NB15 dataset was chosen for this study due to its detailed representation of modern network vulnerabilities, which is crucial for evaluating IDSs in IoT environments. This dataset includes 49 features categorized into 9 groups, offering a balanced complexity that is both relevant and challenging for testing advanced machine learning and deep learning models. Unlike older datasets like KDDCUP99 [15] and NSLKDD [16], which are constrained by outdated attack scenarios, UNSW-NB15 captures a broader range of network behaviors, making it more applicable to current IoT security challenges [17]. Additionally, compared to newer datasets such as Bot-IoT, UNSW-NB15 provides a more comprehensive environment for evaluating

model performance, balancing complexity and real-world applicability [18], [19]. This choice informed the preprocessing steps, feature selection methods, and evaluation processes, ensuring the robustness and relevance of our findings [20], [21].

2.2. Data preprocessing

To prepare the UNSW-NB15 dataset for effective modeling and analysis, several preprocessing steps were undertaken:

- Feature selection and dropping: irrelevant or highly correlated features were removed to prevent overfitting and streamline the dataset, enhancing the model's ability to generalize to new data.
- Clamping extreme values: mitigated skewness by clamping numeric features, balancing data distribution and reducing outlier influence.
- Applying log transformation: normalized high cardinality and skewed features through logarithmic transformation to enhance model suitability.
- Reducing cardinality in categorical features: reduced high cardinality in categorical features by grouping less frequent labels, simplifying the dataset without losing essential information.
- Validation and visualization: validated preprocessing by visually inspecting feature representations before and after transformation, ensuring readiness for model training.

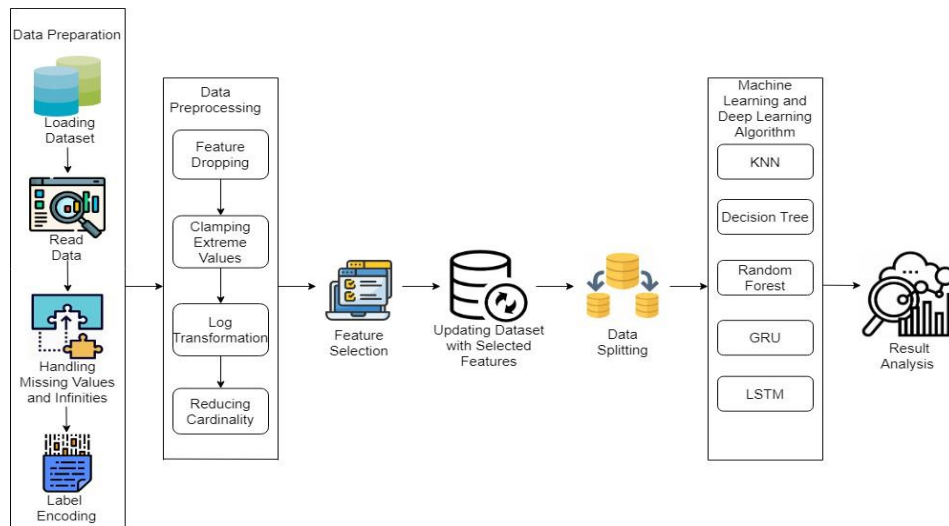


Figure 1. Proposed architecture diagram

2.3. Feature selection

SelectKBest [22] with Chi-Square scoring was chosen as the feature selection method because of its effectiveness in managing high-dimensional data, especially in IoT situations where feature relevance varies significantly. This method was selected over others because the Chi-Square test is particularly well-suited for categorical data, which is prevalent in the UNSW-NB15 dataset. By evaluating each feature's relevance to the target variable, this method ensures that only the most informative features are retained, reducing dimensionality while preserving the dataset's integrity for model training. The Chi-Square statistic, which compares observed and expected frequencies, was used to rank characteristics based on their predictive ability. SelectKBest improves model accuracy and lowers the chance of overfitting by dimensionality reducing the dataset while maintaining important features. The chi-square statistic used in feature selection has the (1).

$$\chi^2 = \frac{\sum(O_m - E_m)^2}{E_m} \quad (1)$$

Were, O_m represents the observed frequency, and E_m the expected frequency under independence. Given the high dimensionality of the UNSW-NB15 dataset, effective feature selection is crucial for identifying relevant attributes, such as network traffic, packet payloads, and protocol headers [22]. SelectKBest with Chi-Square scoring allows analysts to assess the correlation between variables and network intrusions, helping to identify features that strongly correlate with intrusion instances for model training and evaluation [23].

2.4. GRU architecture

The GRU, a type of recurrent neural network (RNN), was chosen for its efficiency in processing sequential data, essential for detecting botnet activities. This model, implemented with the Keras Sequential API, uses GRU to capture long-term dependencies through its memory units and more efficient architecture than LSTM, while still maintaining robust performance. This architecture, combined with dropout techniques to prevent overfitting, ensures the model's suitability for real-time applications in resource-constrained IoT environments. Additionally, GRU's gating mechanism allows it to capture long-term dependencies in data effectively, providing a good balance between model complexity and performance. A GRU unit's principal parts are as (2)-(5).

$$z_t = \sigma(W_z \cdot [h_{t-1}, X_t] + b_z) \quad (2)$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, X_t] + b_r) \quad (3)$$

$$h'_t = \tanh(W_h \cdot [r_t \odot h_{t-1}, X_t] + b_h) \quad (4)$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot h'_t \quad (5)$$

In order to improve data quality for model training, Algorithm 1 describes a systematic method to botnet detection that focusses preprocessing, feature selection using SelectKBest, and label encoding. StandardScaler post data splitting is used for feature scaling in order to speed up model convergence. OneHotEncoder converts categorical labels into numerical format. The 128-unit GRU layer of the GRU model, which was constructed using Keras's Sequential API, is used for sequential data processing. A dropout layer with a rate of 0.5 randomly eliminates input units during training to avoid overfitting. A densely connected layer with 64 units and a rectified linear unit (ReLU) activation function are used to introduce non-linearity [24], [25].

Algorithm 1. Feature selection, data splitting, feature scaling, and label encoding

Require: Feature matrix (X), Label vector (y), Number of features to select (k), Number of classes

Ensure: Transformed features X_{selected} , Encoded labels y_{encoded} , Split datasets (X_{train} , X_{val} , X_{test} , Y_{train} , Y_{val} , Y_{test})

- 1: Perform Feature Selection:
- 2: Use SelectKBest with chi-squared scoring to select the top k features.
- 3: Fit SelectKBest on the features (X) and labels (y).
- 4: Transform the features (X) to obtain the selected features (X_{selected}).
- 5: Split the Data:
- 6: Split the dataset into training, validation, and testing sets.
- 7: Scale the Features:
- 8: Scale the features using StandardScaler.
- 9: Fit the scaler on the training set and transform all sets of features.
- 10: Encode the Labels:
- 11: Perform one-hot encoding on the labels using OneHotEncoder.
- 12: Fit the encoder on the training labels and transform all sets of labels.

The Algorithm 2 describes how the GRU-based model is built, trained, evaluated, and saved, assuring maximum performance for botnet detection through organised compilation and validation. Training occurs over 200 epochs with a batch size of 32, utilizing validation data to prevent overfitting. Evaluation on the testing dataset provides metrics like test loss and accuracy, assessing the model's ability to classify unknown data. In neural networks, ReLU function $h = \max(0, a)$ is preferred for classification due to its constant gradient for positive inputs, mitigating the vanishing gradient issue of sigmoid functions and enabling faster learning.

To enhance the model's performance, several feature engineering techniques were employed. This included normalization to scale the data, one-hot encoding for categorical variables, and the application of SelectKBest for feature selection using the chi-squared test. These techniques were essential in reducing dimensionality, improving model accuracy, and reducing overfitting by selecting the most relevant features.

Algorithm 2. Model building, compilation, training, evaluation, and model saving

Require: Transformed features X_{selected} , Encoded labels y_{encoded} , Number of classes

Ensure: Trained GRU model, Test loss, Test accuracy

- 1: Build the Model:
- 2: Build a Sequential model using Keras.
- 3: Add a GRU layer with 128 units and the input shape based on the selected features.
- 4: Add a Dropout layer with a dropout rate of 0.5.
- 5: Add a Dense layer with 64 units and ReLU activation function.
- 6: Add a Dense output layer with units equal to the number of classes and softmax activation.
- 7: Compile the Model:
- 8: Compile with Adam (learning rate 0.001) and categorical cross-entropy loss.

```

9: Train the Model:
10:     Reshape the data for the GRU input and train for 200 epochs (batch size 32) using
validation data.
11: Evaluate the Model:
12:     Evaluate the trained model on the testing data to obtain the test loss and accuracy
and save the model

```

3. RESULTS AND DISCUSSION

The experiments in this study were conducted using Python 3.12.0 in Jupyter Notebook 7.0.6 on a Windows 10×64 machine with an Intel Core i5 processor and 16 GB of RAM. The machine learning models were developed and evaluated using the Scikit-Learn library. Our proposed GRU-based deep learning model for botnet detection was tested on the UNSW-NB15 dataset and compared with several models, including RF, DT, KNN, and LSTM. The GRU model, combined with the SelectKBest feature selection method, outperformed these models in key metrics such as accuracy, precision, recall, and F1-score. A crucial step in our approach was the feature selection process, where the SelectKBest method was employed to identify the most relevant features from the UNSW-NB15 dataset. Figure 2 illustrates the top 20 features selected based on the chi-square scores, which were crucial in optimizing the performance of the GRU model. These features significantly contributed to reducing the model's complexity while maintaining high accuracy, as they represented the most informative attributes for botnet detection. The ability to effectively filter out less relevant features not only improved the model's accuracy but also played a role in reducing overfitting, which is often a challenge in machine learning models dealing with high-dimensional data.

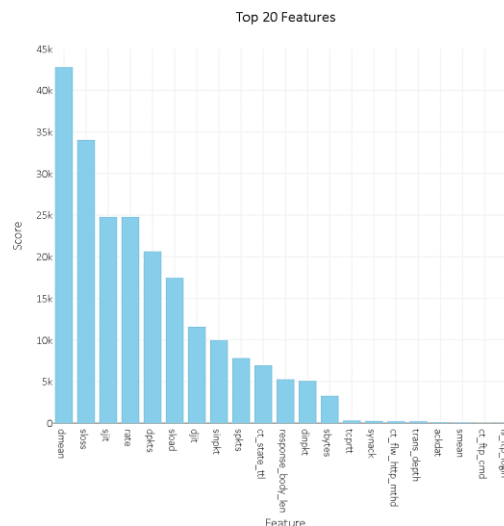


Figure 2. Top 20 features from SelectKBest

Our findings align with earlier research that highlights the effectiveness of deep learning models in IDSs. For instance, Yuan *et al.* [4] employed CNNs and LSTM models for detecting DDoS attacks, achieving high accuracy. However, our GRU-based model, enhanced by the SelectKBest feature selection, surpasses these results, particularly in the context of handling the complexities of IoT network traffic. This suggests that GRU networks are particularly well-suited for processing sequential data with high-dimensional features, as found in the UNSW-NB15 dataset. When comparing our results with Kasongo and Sun [8], who used XGBoost for feature selection and classification on the UNSW-NB15 dataset, our GRU model demonstrated higher accuracy and lower error rates. This performance difference likely arises from the GRU's ability to maintain information over longer sequences, crucial for accurately identifying sophisticated botnet behaviors in the dataset. Moreover, our model showed robustness, likely due to the effective feature selection process and dropout regularization applied during training, unlike some other studies that struggled with overfitting issues.

The proposed GRU model achieved an accuracy of 97.39%, significantly surpassing the RF model's accuracy of 96.538% and the LSTM model's accuracy of 96.94%. The GRU model's precision and recall were also higher, with precision at 98.10% and recall at 97.12%, indicating the model's effectiveness in correctly identifying botnet-related activities. The confusion matrices shown on Figure 3 represents the data of different ML models and DL models. Figures 3(a) to 3(d) represents the predicted values of LR, KNN, DT,

and RF with the improvement of accuracy, whereas Figures 3(e) and 3(f) represents confusion matrix of the LSTM and the proposed model GRU having the highest accuracy showing superior performance by minimizing both false positives and false negatives, leading to the highest F1-score of 97.61%. The receiver operating characteristic (ROC) curve and loss graph analysis was conducted to assess the performance of a classification model and also to provide comprehensive insights into the model’s performance, as shown in Figure 4. The ROC curve for the GRU model in Figure 4(a) had the highest area under the curve (AUC), further reinforcing its superior performance. To provide a comprehensive comparison, Table 1 summarizes the performance metrics of different models, including LR, KNN, DT, RF, LSTM, and GRU. The GRU model outperformed all other models across key metrics, including accuracy, precision, recall, and F1-score, underscoring its robustness and effectiveness in botnet detection.

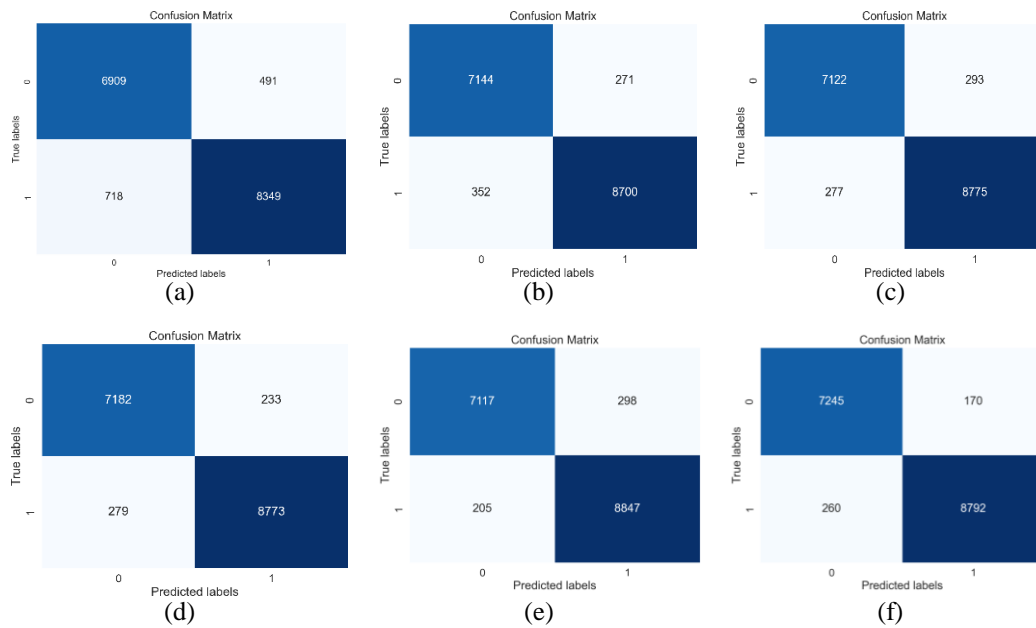


Figure 3. Confusion matrix of; (a) logistic regression (LR), (b) KNN, (c) DT, (d) RF, (e) LSTM, and (f) GRU

A key strength of our approach is the integration of the GRU model with the SelectKBest feature selection method, which reduces dimensionality while maintaining high performance. The chi-square statistic in SelectKBest ensures that only relevant features are retained, enhancing accuracy and minimizing overfitting. However, the dropout layer with a rate of 0.5 helps prevent overfitting during training, as indicated by the steady decline in validation loss over the epochs as shown in Figure 4(b). GRU outperformed LSTM, despite LSTM’s strength in managing long-term dependencies, due to GRU’s simpler architecture and computational efficiency, which allowed it to generalize better to the UNSW-NB15 dataset. Future research should explore optimizations that make the model more suitable for such environments.

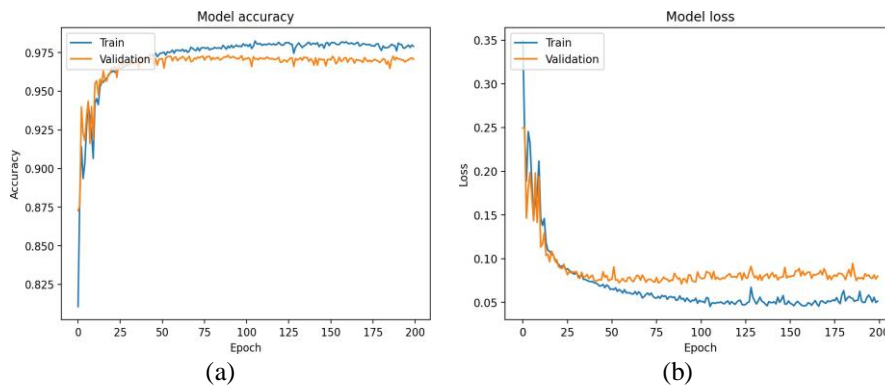


Figure 4. Performance graphs of the GRU model on the UNSW-NB15 dataset, (a) ROC plot and (b) loss graph

Table 1. Comparison of proposed model's performance with other deep learning models investigated in study

ML models	Accuracy (%)	Recall	Precision	F1-score (%)	Error-rate
LR	92.65	0.9208	0.9444	93.2484	0.0734
KNN	96.2167	0.9611	0.9697	96.5433	0.0378
DT	96.5385	0.9693	0.9676	96.8543	0.0346
RF	96.8908	0.9691	0.9741	97.1647	0.0311
LSTM	96.9454	0.9773	0.9674	97.2358	0.0305
GRU	97.3887	0.9712	0.9810	97.612	0.0261

4. CONCLUSION

Our work shows that a GRU-based deep learning model was used to detect botnets on the UNSW-NB15 dataset. The model's performance was evaluated against various machine learning and deep learning models, including RF, LR, KNN, DT, and LSTM. Feature selection was performed using SelectKBest with the chi-squared statistical test. The GRU model, combined with SelectKBest, outperformed all other models, achieving an accuracy of 97.38%, a recall of 97.12%, and a precision of 98.10%. This surpasses the performance of the LSTM model, which had the highest recall and precision among the other models, and the RF model, which also showed strong results. The results suggest that the GRU-based model is highly effective for real-time intrusion detection in IoT environments, with the potential to enhance cybersecurity frameworks by improving detection rates and reducing false positives. Future work will focus on applying this approach to other botnet-related datasets to further validate its robustness and integrating it into real-time network security systems to enhance protection against evolving cyber threats.




REFERENCES

- [1] N. Hoque, D. K. Bhattacharyya, and J. K. Kalita, "Botnet in DDoS attacks: trends and challenges," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2242–2270, 2015, doi: 10.1109/COMST.2015.2457491.
- [2] Irfan, I. M. Wildani, and I. N. Yulita, "Classifying botnet attack on internet of things device using random forest," *IOP Conference Series: Earth and Environmental Science*, vol. 248, Apr. 2019, doi: 10.1088/1755-1315/248/1/012002.
- [3] D. Tran, H. Mac, V. Tong, H. A. Tran, and L. G. Nguyen, "A LSTM based framework for handling multiclass imbalance in DGA botnet detection," *Neurocomputing*, vol. 275, pp. 2401–2413, Jan. 2018, doi: 10.1016/j.neucom.2017.11.018.
- [4] X. Yuan, C. Li, and X. Li, "DeepDefense: identifying DDoS attack via deep learning," in *2017 IEEE International Conference on Smart Computing (SMARTCOMP)*, May 2017, pp. 1–8. doi: 10.1109/SMARTCOMP.2017.7946998.
- [5] M. Feily, A. Shahrestani, and S. Ramadass, "A survey of botnet and botnet detection," in *2009 Third International Conference on Emerging Security Information, Systems and Technologies*, 2009, pp. 268–273. doi: 10.1109/SECURWARE.2009.48.
- [6] R. Doshi, N. Apthorpe, and N. Feamster, "Machine learning DDoS detection for consumer internet of things devices," in *2018 IEEE Security and Privacy Workshops (SPW)*, May 2018, pp. 29–35. doi: 10.1109/SPW.2018.00013.
- [7] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*, Nov. 2015, pp. 1–6. doi: 10.1109/MilCIS.2015.7348942.
- [8] S. M. Kasongo and Y. Sun, "Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset," *Journal of Big Data*, vol. 7, no. 1, p. 105, Dec. 2020, doi: 10.1186/s40537-020-00379-6.
- [9] V. Kumar, D. Sinha, A. K. Das, S. C. Pandey, and R. T. Goswami, "An integrated rule based intrusion detection system: analysis on UNSW-NB15 data set and the real time online dataset," *Cluster Computing*, vol. 23, no. 2, pp. 1397–1418, Jun. 2020, doi: 10.1007/s10586-019-03008-x.
- [10] S. Meftah, T. Rachidi, and N. Assem, "Network based intrusion detection using the UNSW-NB15 dataset," *International Journal of Computing and Digital Systems*, vol. 8, no. 5, pp. 477–487, Jan. 2019, doi: 10.12785/ijcds/080505.
- [11] M. Mohy-eddine, A. Guezzaz, S. Benkirane, and M. Azrou, "An efficient network intrusion detection model for IoT security using K-NN classifier and feature selection," *Multimedia Tools and Applications*, vol. 82, no. 15, pp. 23615–23633, Jun. 2023, doi: 10.1007/s11042-023-14795-2.
- [12] K. Venkatachalam and P. Jacob, "UNSW-NB15 dataset feature selection and network intrusion detection using deep learning," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 7, no. 5S2, pp. 443–446, 2019.
- [13] W. N. H. Ibrahim *et al.*, "Multilayer framework for botnet detection using machine learning algorithms," *IEEE Access*, vol. 9, pp. 48753–48768, 2021, doi: 10.1109/ACCESS.2021.3060778.
- [14] M. Abu Rajab, J. Zarfoss, F. Monrose, and A. Terzis, "A multifaceted approach to understanding the botnet phenomenon," in *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, Oct. 2006, pp. 41–52. doi: 10.1145/1177080.1177086.
- [15] M. Tavallae, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, Jul. 2009, pp. 1–6. doi: 10.1109/CISDA.2009.5356528.
- [16] S. Revathi and A. Malathi, "A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection," *International Journal of Engineering Research & Technology (IJERT)*, vol. 2, no. 12, pp. 1848–1853, 2013.
- [17] Y. Yin *et al.*, "IGRF-RFE: A hybrid feature selection method for MLP-based network intrusion detection on UNSW-NB15 dataset," *Journal of Big Data*, vol. 10, no. 1, Feb. 2023, doi: 10.1186/s40537-023-00694-8.
- [18] J. M. Peterson, J. L. Leevy, and T. M. Khoshgofaar, "A review and analysis of the bot-IoT dataset," in *2021 IEEE International Conference on Service-Oriented System Engineering (SOSE)*, Aug. 2021, pp. 20–27. doi: 10.1109/SOSE52839.2021.00007.




- [19] A. Özgür and H. Erdem, "A review of KDD99 dataset usage in intrusion detection and machine learning between 2010 and 2015," Apr. 14, 2016. doi: 10.7287/peerj.preprints.1954v1.
- [20] N. Moustafa and J. Slay, "The significant features of the UNSW-NB15 and the KDD99 data sets for network intrusion detection systems," in *2015 4th International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS)*, Nov. 2015, pp. 25–31. doi: 10.1109/BADGERS.2015.014.
- [21] M. Zeeshan *et al.*, "Protocol-based deep intrusion detection for DoS and DDoS attacks using UNSW-NB15 and bot-IoT data-sets," *IEEE Access*, vol. 10, pp. 2269–2283, 2022, doi: 10.1109/ACCESS.2021.3137201.
- [22] N. R. Abid-Althaqafi and H. A. Alsalamah, "The effect of feature selection on the accuracy of X-platform user credibility detection with supervised machine learning," *Electronics*, vol. 13, no. 1, Jan. 2024, doi: 10.3390/electronics13010205.
- [23] N. Chavan, M. Kukreja, G. Jagwani, N. Nishad, and N. Deb, "DDoS attack detection and botnet prevention using machine learning," in *2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Mar. 2022, pp. 1159–1163. doi: 10.1109/ICACCS54159.2022.9785247.
- [24] E. Biglar Beigi, H. Hadian Jazi, N. Stakhanova, and A. A. Ghorbani, "Towards effective feature selection in machine learning-based botnet detection approaches," in *2014 IEEE Conference on Communications and Network Security*, Oct. 2014, pp. 247–255. doi: 10.1109/CNS.2014.6997492.
- [25] G. Kocher and G. Kumar, "Analysis of machine learning algorithms with feature selection for intrusion detection using UNSW-NB15 dataset," *International Journal of Network Security & Its Applications*, vol. 13, no. 1, pp. 21–31, Jan. 2021, doi: 10.5121/ijnsa.2021.13102.

BIOGRAPHIES OF AUTHORS



Suchetha G.    received the B.E. and M.Tech. degree in CSE from Visvesvaraya Technological University, India. Currently, she is working as an assistant professor in the Department of Information Science and Engineering at Sahyadri College of Engineering and Management, India. Her area of interests includes cybersecurity, AI, ML, and IoT. She has published research papers in various reputed journals and conferences. She is a life member of the CSI and Data Science Association. She can be contacted at email: suchethag87@gmail.com.



Dr. Pushpalatha K.    is working as Professor in the Department of Artificial Intelligence and Machine Learning, Sahyadri College of Engineering and Management, India. She has completed her Ph.D. in image processing from NITK, Surathkal. She has 15 years of teaching experience. She has published research papers in SCI indexed journals and conferences. Her areas of interest are biomedical signal processing and brain imaging. She can be contacted at email: pushpak@gmail.com.