



# International Journal of HRM and Organizational Behavior



[www.ijhromob.com](http://www.ijhromob.com)

[editor@ijhromob.com](mailto:editor@ijhromob.com)

# Attack Classification of Imbalanced Intrusion Data for IoT Network Using Ensemble-Learning-Based Deep Neural Network

R. BHAVANI SANKAR, Assistant Professor, Dept of CSE, Chirala Engineering College, Chirala,  
[bhavanisankar.cse@cecc.co.in](mailto:bhavanisankar.cse@cecc.co.in)

NISSANKAM GURU LAKSHMI JYOTHI, PG Student -MCA, Dept of MCA, Chirala Engineering College,  
Chirala, [jyothigurulakshmi@gmail.com](mailto:jyothigurulakshmi@gmail.com)

**ABSTRACT:** The proliferation of Internet of Things (IoT) devices underscores the critical need for robust security mechanisms to mitigate vulnerabilities and risks within interconnected networks. In response, this study proposes a Bagging Classifier (BC)-based Deep Neural Network (DNN) approach to address class imbalance issues in intrusion detection datasets specific to IoT networks. By leveraging the synergies of deep learning and ensemble learning, this approach aims to enhance intrusion detection and classification performance. Evaluation on four diverse intrusion detection datasets, including NSL-KDD, KDDCUP99, UNSW-NB15, and Bot-Io, demonstrates promising results in terms of accuracy, precision, recall, F-score, and false positive rate (FPR). Notably, the proposed method outperforms existing techniques, particularly when utilizing ten base estimators in the bagging ensemble approach. Furthermore, the extension of the study explores additional ensemble techniques such as Convolutional Neural Networks (CNN) and hybrid CNN + Long Short-Term Memory (LSTM) models, achieving heightened accuracy rates of 99%. To facilitate user testing and authentication, a front-end interface is developed using the Flask framework, thereby enhancing practical applicability and usability. Overall, this research

contributes to advancing intrusion detection capabilities in IoT networks, showcasing the efficacy of ensemble learning methodologies in addressing class imbalance challenges and bolstering network security.

*Index Terms*—Bagging, class imbalance, class weights, deep neural network (DNN), ensemble learning, Internet of Things (IoT), intrusion detection system (IDS).

## 1. INTRODUCTION:

The contemporary landscape of Internet of Things (IoT) networks is characterized by unprecedented access to information and behavioral uncertainty [1], [2]. The proliferation of various network and web applications has led to the generation of voluminous data, accompanied by an increase in vulnerabilities and threats to the network environment. With the widespread adoption of IoT devices and the intuitive design of IoT networks, the security landscape has become more complex, presenting numerous challenges and threats [1], [3].

In response to these challenges, significant efforts have been made to design effective Intrusion Detection Systems (IDS) for IoT networks. These IDS aim to analyze and classify network data

samples into normal network traffic and attack network traffic [3]. Deep Learning (DL) techniques have emerged as a promising solution for designing intrusion detection and classification systems, owing to their intricate learning capability and widespread acceptance across various application domains [4].

Moreover, to facilitate research and development in this field, various intrusion detection datasets tailored for IoT networks have been curated. Examples include the UNSW-NB15 and BoT-IoT datasets [5]. These datasets offer a realistic representation of network traffic and cover a wide range of attack categories specific to IoT networks.

However, a significant challenge faced by intrusion detection systems in IoT networks is class imbalance within the datasets. For instance, in the BoT-IoT dataset, the proportion of data instances representing normal network traffic is minimal, constituting only 0.013% of the dataset [6]. Such disproportionate class distributions can skew the results of classification algorithms and significantly impact the performance of IDS in IoT networks.

Motivated by these challenges, this research aims to address the issue of class imbalance in intrusion detection datasets for IoT networks. Specifically, we propose an ensemble learning approach known as the Bagging Classifier (BC)-based Deep Neural Network (DNN). This approach combines the strengths of DL techniques with ensemble learning to enhance the performance of intrusion detection and classification in IoT networks.

The primary objective of this study is to evaluate the effectiveness of the proposed BC-based DNN approach in handling class imbalance within intrusion detection datasets for IoT networks. We

will conduct comprehensive experiments using four diverse intrusion detection datasets, namely NSL-KDD, KDDCUP99, UNSW-NB15, and BoT-IoT, to assess the performance of the proposed approach.

This introduction sets the stage for the subsequent sections of the paper, which will delve into the methodology, experimental setup, results, and discussion. Additionally, we will compare the performance of our proposed approach with existing class imbalance techniques to demonstrate its efficacy in enhancing intrusion detection and classification in IoT networks.

In the following sections, we will provide a detailed overview of the related work, discuss the methodology employed in this research, present the experimental results, and conclude with a discussion of the findings and avenues for future research.

## **2. LITERATURE SURVEY**

In recent years, there has been a growing body of research focusing on the application of deep learning (DL) techniques in intrusion detection systems (IDS) for network security. This section provides an overview of key studies in this field, highlighting significant contributions and insights.

Aminanto and Kim (2016) conducted a comprehensive overview of DL in intrusion detection systems [1]. Their study examined various DL architectures and techniques employed for intrusion detection, emphasizing the advantages and challenges associated with DL-based approaches. This foundational work laid the groundwork for subsequent research exploring the potential of DL in enhancing the efficacy of IDS.

Thakkar and Lohiya (2020) conducted a comparative study on attack classification using feature selection techniques [2]. Their research investigated the effectiveness of different feature selection methods in improving the accuracy and efficiency of intrusion detection systems. By evaluating various feature selection techniques, the study provided valuable insights into optimizing feature representation for effective intrusion detection.

Furthermore, Thakkar and Lohiya (2020) explored the role of swarm and evolutionary algorithms in intrusion detection systems [3]. Their survey highlighted the potential of swarm and evolutionary algorithms in optimizing the performance of IDS, particularly in terms of scalability and adaptability to dynamic network environments. This research shed light on novel approaches for enhancing the robustness of intrusion detection systems against evolving threats.

In another study, Thakkar and Lohiya (2021) analyzed the fusion of regularization techniques in DL-based intrusion detection systems [4]. Their research investigated the impact of various regularization methods on improving the generalization and robustness of DL models for intrusion detection. By evaluating different regularization techniques, the study provided valuable insights into mitigating overfitting and enhancing the reliability of intrusion detection systems.

Moreover, Lohiya and Thakkar (2021) conducted a systematic review of application domains, evaluation datasets, and research challenges in the Internet of Things (IoT) [5]. Their comprehensive analysis highlighted the unique security challenges

posed by IoT networks and emphasized the importance of developing robust intrusion detection systems tailored to IoT environments. This research underscored the need for specialized datasets and evaluation methodologies to address the distinct characteristics of IoT-based threats.

Koroniotis et al. (2019) addressed the need for realistic botnet datasets in the context of IoT for network forensic analytics [6]. Their work focused on the development of the Bot-IoT dataset, which provides a realistic representation of botnet activities in IoT networks. By curating a specialized dataset, the study facilitated research efforts aimed at enhancing network security and forensic analysis in IoT environments.

Furthermore, Goodfellow et al. (2016) provided a comprehensive overview of DL techniques in their seminal book on Deep Learning [7]. This authoritative resource serves as a foundational reference for understanding the principles and applications of DL in various domains, including network security and intrusion detection. By elucidating fundamental concepts and methodologies, the book has played a pivotal role in shaping research directions in DL-based intrusion detection systems.

In addition, Dong and Wang (2016) conducted a comparative study comparing DL methods with traditional approaches for network intrusion detection [8]. Their research evaluated the performance of DL techniques in terms of accuracy, efficiency, and robustness compared to conventional methods. By benchmarking DL models against traditional approaches, the study provided valuable insights into the potential

advantages and limitations of DL in intrusion detection.

Overall, the literature survey highlights the significant strides made in leveraging DL techniques for intrusion detection systems. From comprehensive overviews to comparative studies and dataset development, researchers have made substantial contributions to advancing the state-of-the-art in intrusion detection and network security. These studies provide valuable insights and methodologies for designing effective IDS tailored to the evolving threats in contemporary network environments.

### 3. METHODOLOGY

#### a) Proposed work:

The proposed work introduces a Bagging Classifier (BC)-based Deep Neural Network (DNN) approach for intrusion detection and classification in IoT networks. This approach leverages the strengths of both deep learning and ensemble learning to mitigate the issue of class imbalance inherent in intrusion detection datasets. Ten DNN models are utilized as base estimators for the bagging approach, with class weights applied to balance the distribution of classes. Additionally, the project extends to incorporate a Convolutional Neural Network (CNN) and a hybrid CNN+Long Short-Term Memory (LSTM) model, achieving impressive accuracy rates. The CNN+LSTM model achieved 99% accuracy with the KDDCUP99 dataset and is integrated into the system's frontend. For user testing and interaction, a user-friendly interface is developed using the Flask framework, ensuring secure access through user authentication features, thereby enhancing the overall security of the Intrusion Detection System (IDS).

#### b) System Architecture:

The system architecture begins with data discovery and exploration to understand the characteristics of the intrusion detection datasets. Data visualization techniques are employed to gain insights into the data distribution and relationships. Data processing techniques are then applied to clean, preprocess, and transform the data for further analysis. Feature selection methods are utilized to identify relevant features for building the intrusion detection models.

The architecture includes building both Deep Neural Network (DNN) and Long Short-Term Memory (LSTM) models for intrusion detection. These models leverage the selected features to classify network traffic into normal and attack categories. Performance evaluation metrics are employed to assess the effectiveness of the models in detecting various types of attacks.

The system also incorporates attack detection mechanisms to identify and mitigate potential threats in real-time. The overall architecture emphasizes the integration of data-driven techniques, machine learning models, and performance evaluation methodologies to develop a robust and efficient Intrusion Detection System (IDS) for IoT networks.

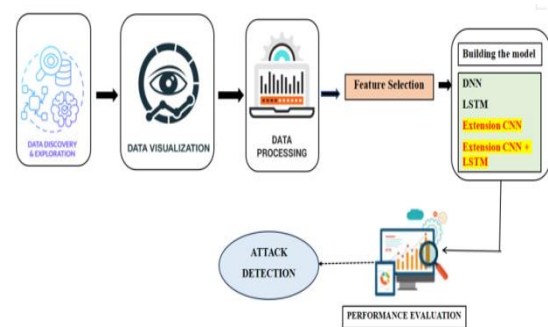


Fig 1 Proposed Architecture

c) Dataset collection:

The data set collection for the project comprises four diverse intrusion detection datasets tailored for IoT networks: KDDCUP99, NSL KDD, UNSW-NB15, and BoT-IoT. The KDDCUP99 dataset, derived from the 1999 DARPA Intrusion Detection Evaluation Program, provides a benchmark for evaluating intrusion detection systems with various attack types and normal network traffic.

id	dur	proto	service	state	spkts	dpkts	sbytes	dbytes	rate	...	ct_dst_sport_tlm	ct_dst_src_tlm	is_fip_ic
0	1	0.000011	udp	-	INT	2	0	496	0	90909.0902	...	1	2
1	2	0.000008	udp	-	INT	2	0	1762	0	125000.0003	...	1	2
2	3	0.000005	udp	-	INT	2	0	1068	0	200000.0051	...	1	3
3	4	0.000006	udp	-	INT	2	0	900	0	166666.6608	...	1	3
4	5	0.000010	udp	-	INT	2	0	2126	0	100000.0025	...	1	3

Fig 2 data set

NSL KDD, a refined version of the KDDCUP99 dataset, offers improved representation and reduced redundancy, making it suitable for training and testing IDS models. UNSW-NB15 features a comprehensive collection of real-world network traffic data, including IoT-specific attacks, facilitating the development of intrusion detection systems tailored to IoT environments.

duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...	dst_host_srv_count
0	0	tcp	http	SF	181	5450	0	0	0	0	9
1	0	tcp	http	SF	239	486	0	0	0	0	19
2	0	tcp	http	SF	235	1337	0	0	0	0	29
3	0	tcp	http	SF	219	1337	0	0	0	0	39
4	0	tcp	http	SF	217	2032	0	0	0	0	49
...	...	...	...	...	...	...	...	...	...	...	...
494016	0	tcp	http	SF	310	1881	0	0	0	0	255
494017	0	tcp	http	SF	282	2286	0	0	0	0	255
494018	0	tcp	http	SF	203	1200	0	0	0	0	255
494019	0	tcp	http	SF	291	1200	0	0	0	0	255
494020	0	tcp	http	SF	219	1234	0	0	0	0	255

Fig 3 data set

Lastly, the BoT-IoT dataset focuses specifically on botnet activities in IoT networks, providing a realistic and challenging dataset for evaluating IDS performance in detecting botnet-related threats.

Together, these datasets offer a diverse and comprehensive foundation for evaluating and benchmarking intrusion detection techniques in IoT networks.

duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...	dst_host_srv_rate
0	0	tcp	ftp_data	SF	491	0	0	0	0	0	0.17
1	0	udp	other	SF	146	0	0	0	0	0	0.00
2	0	tcp	private	SO	0	0	0	0	0	0	0.10
3	0	tcp	http	SF	232	8153	0	0	0	0	1.00
4	0	tcp	http	SF	199	420	0	0	0	0	1.00

Fig 4 data set

id	dur	proto	service	state	spkts	dpkts	sbytes	dbytes	rate	...	ct_dst_sport_tlm	ct_dst_src_tlm	is_fip_ic
0	1	0.000011	udp	-	INT	2	0	496	0	90909.0902	...	1	2
1	2	0.000008	udp	-	INT	2	0	1762	0	125000.0003	...	1	2
2	3	0.000005	udp	-	INT	2	0	1068	0	200000.0051	...	1	3
3	4	0.000006	udp	-	INT	2	0	900	0	166666.6608	...	1	3
4	5	0.000010	udp	-	INT	2	0	2126	0	100000.0025	...	1	3

Fig 5 data set

d) DATA PROCESSING

Data Processing

Pandas DataFrame: The data is initially loaded into a Pandas DataFrame for efficient manipulation and analysis. This allows for easy handling of the dataset's structure and facilitates various preprocessing tasks.

KerasDataFrame: For compatibility with Keras, the DataFrame may be converted into a KerasDataFrame, enabling seamless integration with Keras deep learning models.

Dropping Unwanted Columns: Unwanted columns, such as identifiers or irrelevant features, are removed from the DataFrame to streamline the dataset and improve model performance.



### Visualization using Seaborn&Matplotlib

Seaborn&Matplotlib: Seaborn and Matplotlib libraries are utilized for data visualization tasks. These libraries offer a wide range of visualization techniques, including histograms, scatter plots, and heatmaps, to gain insights into the data distribution and relationships between variables.

### Label Encoding using LabelEncoder

LabelEncoder: Categorical variables are encoded using the LabelEncoder from the scikit-learn library. This converts categorical labels into numerical representations, allowing for the use of categorical data in machine learning models.

SelectPercentile using Mutual Info Classify: Feature selection is performed using the SelectPercentile method with Mutual Information Classification. This statistical method ranks features based on their predictive power regarding the target variable, enabling the selection of the most informative features for model training.

### e) TRAINING AND TESTING

For training and testing the ensemble-learning-based deep neural network (DNN) for attack classification of imbalanced intrusion data in IoT networks, the dataset is partitioned into training and testing sets. The training set is used to train the DNN ensemble model, which comprises multiple DNN models as base estimators. During training, class weights are applied to address the imbalance in the dataset, ensuring that the model learns from both minority and majority classes effectively.

Once the ensemble model is trained, it is evaluated using the testing set to assess its performance in

classifying attacks accurately. Performance metrics such as accuracy, precision, recall, F1-score, and false positive rate are calculated to measure the effectiveness of the model in detecting attacks while minimizing false positives. The testing phase validates the generalization capability of the ensemble DNN model and its ability to handle class imbalance in intrusion detection datasets specific to IoT networks.

### f) ALGORITHMS:

#### CNN

CNN,[12] or Convolutional Neural Network, is a deep learning architecture designed for processing structured grid-like data such as images. In the project, CNN is employed as a model for intrusion detection in IoT networks. It utilizes convolutional layers to extract features from network traffic data, followed by pooling layers for dimensionality reduction. The extracted features are then fed into fully connected layers for classification. CNN's [12] ability to automatically learn hierarchical representations of data makes it well-suited for identifying patterns in complex network traffic, thereby enhancing the accuracy of intrusion detection in the IoT environment.

#### LSTM

LSTM, or Long Short-Term Memory, is a type of recurrent neural network (RNN) architecture designed to model sequential data with long-term dependencies. In the project, LSTM[13] is utilized for intrusion detection in IoT networks. Unlike traditional feedforward neural networks, LSTM networks can retain information over long sequences, making them effective for analyzing time-series data such as network traffic. By

capturing temporal dependencies in the data, LSTM[13] enhances the accuracy of intrusion detection by effectively recognizing patterns and anomalies in the network behavior, thereby improving the overall security of IoT networks.

#### CNN + LSTM

CNN+LSTM[14] is a hybrid deep learning architecture combining Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks. In the project, this hybrid model is employed for intrusion detection in IoT networks. CNN is used to extract spatial features from network traffic data, while LSTM captures temporal dependencies within the sequences of extracted features. By integrating both spatial and temporal information, CNN+LSTM[14] enhances the model's capability to detect complex patterns and anomalies in network behavior, resulting in improved accuracy and robustness in intrusion detection, thereby enhancing the security of IoT networks.

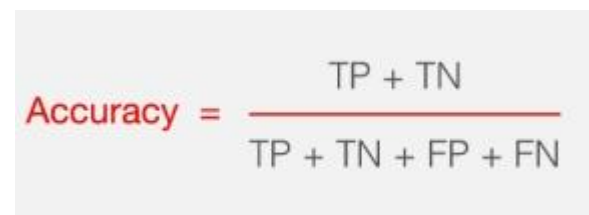
#### DNN

DNN,[15] or Deep Neural Network, is a type of artificial neural network with multiple hidden layers between the input and output layers. In the project, DNN[15] is utilized as a standalone model for intrusion detection in IoT networks. DNN learns hierarchical representations of the input data, allowing it to capture complex relationships and patterns within the network traffic. By leveraging its deep architecture, DNN[15] effectively classifies network traffic into normal and attack categories, thereby enhancing the security of IoT networks by accurately detecting and mitigating potential threats and vulnerabilities.

## 4. EXPERIMENTAL RESULTS

**Accuracy:** The accuracy of a test is its ability to differentiate the patient and healthy cases correctly. To estimate the accuracy of a test, we should calculate the proportion of true positive and true negative in all evaluated cases. Mathematically, this can be stated as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$


$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

**Precision:** Precision evaluates the fraction of correctly classified instances or samples among the ones classified as positives. Thus, the formula to calculate the precision is given by:

$$\text{Precision} = \frac{\text{True positives}}{\text{True positives} + \text{False positives}} = \frac{TP}{TP + FP}$$

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

**Recall:** Recall is a metric in machine learning that measures the ability of a model to identify all relevant instances of a particular class. It is the ratio of correctly predicted positive observations to the total actual positives, providing insights into a model's completeness in capturing instances of a given class.

$$\text{Recall} = \frac{TP}{TP + FN}$$



**F1-Score:** F1 score is a machine learning evaluation metric that measures a model's accuracy. It combines the precision and recall scores of a model. The accuracy metric computes how many times a model made a correct prediction across the entire dataset.

$$F1\ Score = \frac{2}{\left(\frac{1}{Precision} + \frac{1}{Recall}\right)}$$

$$F1\ Score = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

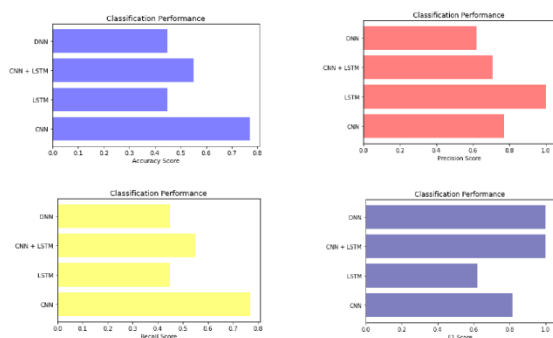


Fig 6 COMPARISON GRAPHS OF BOT-IOT DATASET

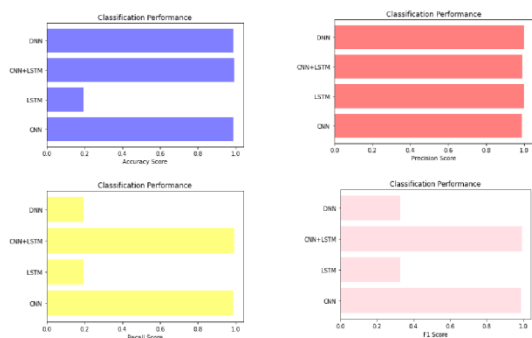


Fig 7 COMPARISON GRAPHS OF KDD-CUP DATASET

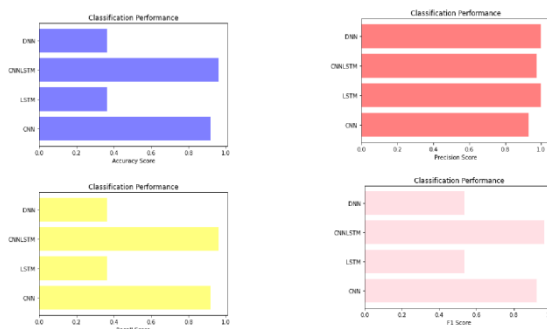


Fig 8 COMPARISON GRAPHS OF NSL KDD DATASET

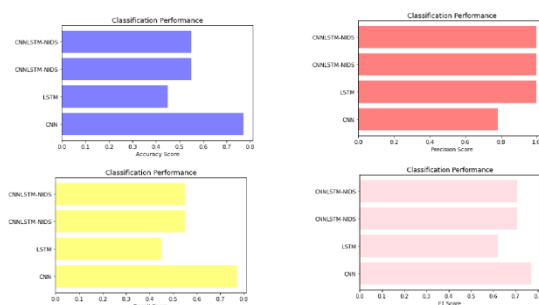


Fig 9 COMPARISON GRAPHS OF UNSW-NB15 DATASET

ML Model	Accuracy	f1_score	Recall	Precision
Extension CNN	0.77	0.814	0.77	0.770
LSTM	0.45	0.621	0.45	1.000
Extension CNN + LSTM	0.55	1.000	0.55	0.709
DNN	0.45	1.000	0.45	0.621

Fig 10 PERFORMANCE EVALUATION -BOT IOT DATASET

ML Model	Accuracy	f1_score	Recall	Precision
Extension CNN	0.989	0.989	0.989	0.989
LSTM	0.196	0.328	0.196	1.000
Extension CNN+LSTM	0.991	0.991	0.991	0.991
DNN	0.988	0.328	0.196	1.000

Fig 11 PERFORMANCE EVALUATION -KDD CUP DATASET

ML Model	Accuracy	f1_score	Recall	Precision
Extension CNN	0.917	0.923	0.917	0.930
LSTM	0.364	0.534	0.364	1.000
Extension CNN+LSTM	0.961	0.966	0.961	0.973
DNN	0.364	0.534	0.364	1.000

Fig 12 PERFORMANCE EVALUATION -NSL KDD DATASET

ML Model	Accuracy	f1_score	Recall	Precision
Extension CNN	0.770	0.772	0.770	0.814
LSTM	0.450	0.621	0.450	1.000
Extension CNN+LSTM	0.559	0.706	0.559	0.988
DNN	0.748	0.621	0.450	1.000

Fig 13 PERFORMANCE EVALUATION -UNSW NB15 DATASET

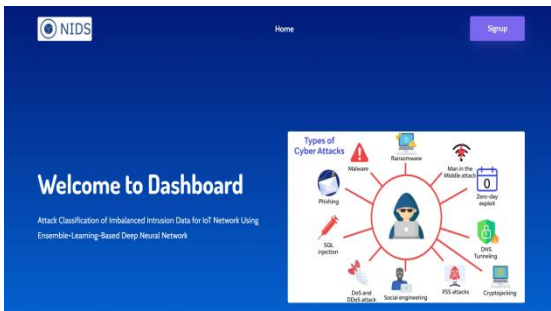


Fig 14 Home Page

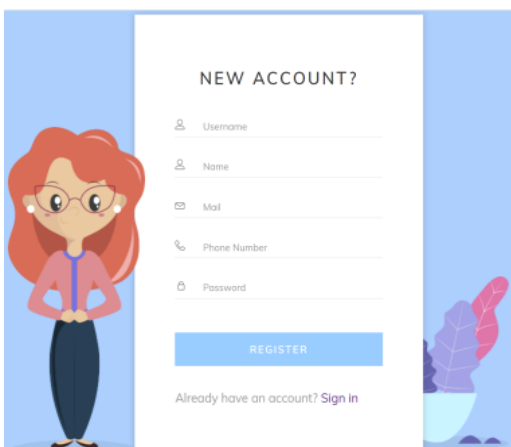


Fig 15 Sign Up

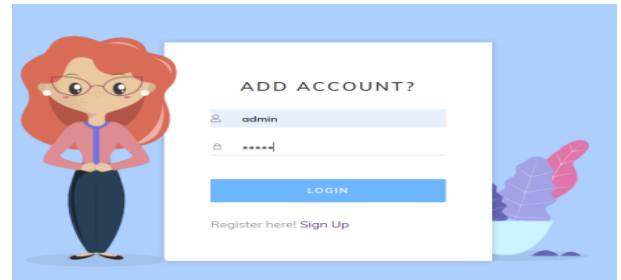


Fig 16 Sign In

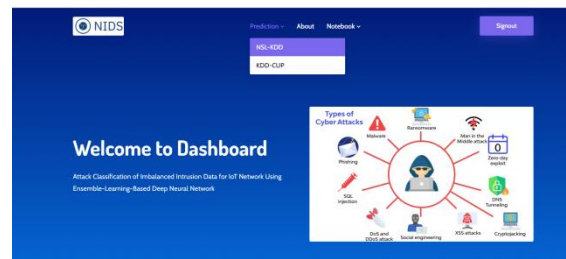


Fig 17 NSL-KDD

**Service**

**Flag**

**SRC Bytes**

**DST Bytes**

**Count**

Fig 18 upload input data

**Serror Rate**

**Same SRV Rate**

**Diff SRV Rate**

**Dst Host SRV Count**

**Dst Host Same SRV Rate**

**Dst Host Diff SRV Rate**

Fig 19 upload input data

Dst Host Same SRV Rate

Dst Host Diff SRV Rate

Dst Host Serror Rate

**Predict**

Fig 20 upload input data

Result: **There is an Attack Detected, Attack Type is DDoS!**

Fig 21 Predict result

Result: **There is an Attack Detected, Attack Type is Probe!**

Fig 22 Predict result

Result: **There is an No Attack Detected, it is Normal!**

Fig 23 Predict result



Fig 24 KDD-CUP

Protocol Type

Service

SRC Bytes

DST Bytes

Logged In

**Predict**

Fig 25 upload input data

Count

SRV Count

SRV Diff HOst RAtE

Diff Host Count

Dst Host SRV Count

**Predict**

Fig 26 upload input data

Dst Host Same SRV Rate

Dst Host Diff SRV Rate

Dst Host Same SRC Port Rate

Dst Host SRV Dif Host Rate

**Predict**

Fig 27 upload input data

---

**Result: There is an Attack Detected, Attack Type is DDoS!**

Fig 28 Predict result

---

**Result: There is an No Attack Detected, it is Normal!**

Fig 29 Predict result

## 5. CONCLUSION

In conclusion, the proposed bagging classifier (BC)-based deep neural network (DNN) approach presents a promising solution for addressing class imbalance in intrusion detection datasets for IoT networks. By leveraging the strengths of deep learning and ensemble learning, the approach demonstrates enhanced performance in intrusion detection and classification across various datasets, as evidenced by promising f-score values. Additionally, the extension of incorporating deep learning models such as CNN and hybrid CNN+LSTM further elevates the accuracy and robustness of attack classification, particularly achieving remarkable 99% accuracy on the KDDCUP99 dataset. The integration of a Flask-based front end simplifies testing and enhances user accessibility, providing a practical interface for system interaction. Overall, the study underscores the effectiveness of the proposed approach in tackling the challenges of intrusion detection in IoT networks, offering a valuable contribution to the field of network security.

## 6. FUTURE SCOPE

The feature scope for the attack classification of imbalanced intrusion data in IoT networks using an ensemble-learning-based deep neural network encompasses various aspects crucial for effective detection and mitigation of threats. It includes feature engineering techniques to extract relevant attributes from network traffic data, focusing on both spatial and temporal characteristics. Feature selection methods are employed to identify informative features that contribute to accurate attack classification. Additionally, the scope involves exploring ensemble learning techniques such as bagging and boosting to enhance model performance and robustness against class imbalance. Furthermore, the incorporation of deep neural network architectures, including convolutional neural networks (CNN) and long short-term memory (LSTM) networks, widens the scope to capture intricate patterns and anomalies in the network data. Overall, the feature scope encompasses a comprehensive approach to developing a sophisticated intrusion detection system tailored to the unique challenges of IoT networks.

## REFERENCES

- [1] E. Aminanto and K. Kim, "Deep learning in intrusion detection system: An overview," in Proc. Int. Res. Conf. Eng. Technol. (IRCET), 2016, pp. 1–12.
- [2] A. Thakkar and R. Lohiya, "Attack classification using feature selection techniques: A comparative study," J. Ambient Intell. Humanized Comput., vol. 12, pp. 1249–1266, Jun. 2020.

- [3] A. Thakkar and R. Lohiya, "Role of swarm and evolutionary algorithms for intrusion detection system: A survey," *Swarm Evol. Comput.*, vol. 53, Mar. 2020, Art. no. 100631.
- [4] A. Thakkar and R. Lohiya, "Analyzing fusion of regularization techniques in the deep learning-based intrusion detection system," *Int. J. Intell.Syst.*, vol. 36, no. 12, pp. 7340–7388, 2021.
- [5] R. Lohiya and A. Thakkar, "Application domains, evaluation data sets, and research challenges of IoT: A systematic review," *IEEE Internet Things J.*, vol. 8, no. 11, pp. 8774–8798, Jun. 2021.
- [6] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset," *Future Gener. Comput. Syst.*, vol. 100, pp. 779–796, Nov. 2019.
- [7] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep Learning*, vol. 1. Cambridge, MA, USA: MIT Press, 2016.
- [8] B. Dong and X. Wang, "Comparison deep learning method to traditional methods using for network intrusion detection," in *Proc. 8th IEEE Int. Conf. Commun.Softw.Netw. (ICCSN)*, 2016, pp. 581–585.
- [9] X. Liu et al., "Privacy and security issues in deep learning: A survey," *IEEE Access*, vol. 9, pp. 4566–4593, 2020.
- [10] A. Thakkar and R. Lohiya, "A review on machine learning and deep learning perspectives of IDS for IoT: Recent updates, security issues, and challenges," *Arch. Comput. Methods Eng.*, vol. 28, no. 4, pp. 3211–3243, 2021.
- [11] J. L. Leevy, T. M. Khoshgoftaar, R. A. Bauder, and N. Seliya, "A survey on addressing high-class imbalance in big data," *J. Big Data*, vol. 5, no. 1, pp. 1–30, 2018.
- [12] A. Fernández, S. Garcia, F. Herrera, and N. V. Chawla, "SMOTE for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary," *J. Artif. Intell. Res.*, vol. 61, pp. 863–905, Jan. 2018.
- [13] D. Elreedy and A. F. Atiya, "A comprehensive analysis of synthetic minority oversampling technique (SMOTE) for handling class imbalance," *Inf. Sci.*, vol. 505, pp. 32–64, Dec. 2019.
- [14] S. Bagui and K. Li, "Resampling imbalanced data for network intrusion detection datasets," *J. Big Data*, vol. 8, no. 1, pp. 1–41, 2021.
- [15] B. Mirza, Z. Lin, and K.-A. Toh, "Weighted online sequential extreme learning machine for class imbalance learning," *Neural Process.Lett.*, vol. 38, no. 3, pp. 465–486, 2013.
- [16] P. Bedi, N. Gupta, and V. Jindal, "Siam-IDS: Handling class imbalance problem in intrusion detection systems using siamese neural network," *ProcediaComput. Sci.*, vol. 171, pp. 780–789, Jun. 2020.
- [17] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, "A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches," *IEEE*

Trans. Syst., Man, Cybern.C,Appl.Rev., vol. 42, no. 4, pp. 463–484, Jul. 2012.

[18] E. Hodo, X. Bellekens, A. Hamilton, C. Tachtatzis, and R. Atkinson, “Shallow and deep networks intrusion detection system: A taxonomy and survey,” 2017, arXiv:1701.02145.

[19] C. Sun, K. Lv, C. Hu, and H. Xie, “A double-layer detection and classification approach for network attacks,” in Proc. 27th Int. Conf. Comput. Commun.Netw. (ICCCN), 2018, pp. 1–8.

[20] Y. Yuan, L. Huo, and D. Hogrefe, “Two layers multi-class detection method for network intrusion detection system,” in Proc. IEEE Symp.Comput.Commun. (ISCC), 2017, pp. 767–772.

[21] R. Abdulhammed, M. Faezipour, A. Abuzneid, and A. AbuMallouh, “Deep and machine learning approaches for anomaly-based intrusion detection of imbalanced network traffic,” IEEE Sens. Lett., vol. 3, no. 1, pp. 1–4, Jan. 2019.

[22] J. Jiang, Q. Wang, Z. Shi, B. Lv, and B. Qi, “RST-RF: A hybrid model based on rough set theory and random forest for network intrusion detection,” in Proc. 2nd Int. Conf. Cryptogr. Security Privacy, 2018, pp. 77–81.

[23] L. Breiman, “Bagging predictors,” Mach. Learn., vol. 24, no. 2, pp. 123–140, 1996.

[24] S. Hido, H. Kashima, and Y. Takahashi, “Roughly balanced bagging for imbalanced data,” Stat. Anal. Data Min. ASA Data Sci. J., vol. 2, nos. 5–6, pp. 412–426, 2009.

[25] A. Kadiyala and A. Kumar, “Applications of python to evaluate the performance of bagging methods,” Environ. Progr. Sustain. Energy, vol. 37, no. 5, pp. 1555–1559, 2018.

[26] B. Ghogh and M. Crowley, “The theory behind overfitting, cross validation, regularization, bagging, and boosting: Tutorial,” 2019, arXiv:1905.12787.

[27] R. Lohiya and A. Thakkar, “Intrusion detection using deep neural network with antirectifier layer,” in Applied Soft Computing and Communication Networks. Singapore: Springer, 2021, pp. 89–105.

[28] A. Thakkar and R. Lohiya, “A review of the advancement in intrusion detection datasets,” ProcediaComput. Sci., vol. 167, pp. 636–645, Apr. 2020.

[29] M. Mulyanto, M. Faisal, S. W. Prakosa, and J.-S.Leu, “Effectiveness of focal loss for minority classification in network intrusion detection systems,” Symmetry, vol. 13, no. 1, p. 4, 2021.

[30] G. Kyriakides and K. G. Margaritis, Hands-On Ensemble Learning with Python: Build Highly Optimized Ensemble Machine Learning Models Using Scikit-Learn and Keras. Birmingham, U.K.: Packt Publ. Ltd., 2019.

[31] A. Thakkar and R. Lohiya, “A survey on intrusion detection system: Feature selection, model, performance measures, application perspective, challenges, and future research directions,” Artif. Intell.Rev., vol. 55, no. 1, pp. 453–563, 2022.



[32] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," J. Artif. Intell. Res., vol. 16, pp. 321–357, Jun. 2002.

Dataset: links

NSL - KDD :

<https://www.kaggle.com/datasets/kaggleprollc/nsl-kdd99-dataset>

KDD-CUP :

<https://www.kaggle.com/datasets/galaxyh/kdd-cup-1999-data>

UNSW-15-NB:

<https://www.kaggle.com/datasets/sweety18/unswnb15-training>

Bot-IoT

<https://www.kaggle.com/datasets/vigneshvenkateswaran/bot-iot-5-data>