International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 22 No. 3 (2026)

ijerst.editor@gmail.com

editor@ijerst.com

ISSN 2319-5991 www.ijerst.com

Peer Reviewed, Referred & Indexed Journal

Research Paper

Automated Network Intrusion Detection for Internet of Things Security Enhancements

RANGU SHASHIDHAR[1]
PG Scholar, Department of Computer Networks and Information Security,
Nalla Mallareddy Engineering College, AUTONOMOUS INSTITUTION, DIVYA NAGAR,
MALKAJGIRI-MEDCHAL (DIST), HYDERABAD-500088
Email: rangushashidhar2@gmail.com

Dr. M. Raju[2]
Professor and head, Department of Computer science and Engineering
Nalla Mallareddy Engineering College, AUTONOMOUS INSTITUTION
(Approved by AICTE, New Delhi and Affiliated to JNTUH)
Accredited by NAAC with "A" Grade
DIVYA NAGAR, MALKAJGIRI-MEDCHAL (DIST), HYDERABAD-500088 Telangana
Email: hodcse@nmrec.edu.in

Abstract

As interconnected devices increasingly transmit personal and sensitive data, security attacks are becoming more sophisticated and prevalent, highlighting the critical need for effective security solutions in Internet of Things (IoT) environments. An automated Network Intrusion Detection (NID) system plays a vital role in notifying system administrators of security breaches, acting as an efficient tool for protecting IoT networks from various threats. This study utilizes the UNSW-NB 15 dataset to enhance intrusion detection accuracy by addressing performance challenges and class imbalances within the data. We employ a combination of feature selection techniques, including Filter Method, Wrapper Method, and an Embedded approach using Lasso and Random Forest with Recursive Feature Elimination (RFE), alongside Pearson Correlation Coefficient (PCC). To tackle class imbalance, we apply the Synthetic Minority Over-sampling Technique (SOMTE). Various algorithms are implemented, including Random Forest, Decision Tree, AdaBoost, Bernoulli Naive Bayes, K-Nearest Neighbors, and Logistic Regression. Notably, the Stacking Classifier, which combines Boosted Decision Trees, Bagging with Random Forest, and LightGBM, demonstrates high performance in accurately detecting intrusions, significantly improving detection rates and reducing false alarms.

Index Terms— Internet of Things, network intrusion detection, cybersecurity, machine learning, feature selection, SMOTE, ensemble learning, stacking classifier, UNSW-NB15.

I. INTRODUCTION

The Internet of Things has transformed computing environments by connecting large numbers of devices, sensors, embedded systems, and services through communication networks. These devices generate and exchange significant amounts of information, including sensitive and operational data. The increasing connectivity of IoT environments consequently expands the attack surface available to adversaries. The project source identifies unauthorized access, data compromise, availability problems, privacy violations, and service disruption as important consequences of successful attacks against connected systems.

A Network Intrusion Detection System (NIDS) monitors network activity and analyzes traffic to identify suspicious, malicious, or policy-violating behavior. Traditional intrusion detection approaches can become difficult to maintain as network behavior and attack techniques evolve. Machine learning (ML) and deep learning (DL) have consequently become important approaches for automatically identifying patterns associated with attacks [1], [2], [16], [20]. The literature included in the project demonstrates the application of recurrent neural networks, gated recurrent units, long short-term memory networks, convolutional models, and ensemble approaches to intrusion detection.

The choice of dataset is also important for evaluating an intrusion detection system. The UNSW-NB15 dataset was developed to provide contemporary normal and malicious network activity and includes multiple attack categories and network features. The dataset contains 2,540,044 records and 49 features plus the class label; predefined training and testing partitions contain 175,341 and 82,332 records, respectively.

Despite the progress of ML-based NIDSs, several challenges remain. First, network datasets can contain irrelevant or redundant attributes that increase computational requirements and potentially reduce classification quality. Second, intrusion datasets may exhibit class imbalance, causing classifiers to favor majority classes. Third, individual classifiers may have different strengths and weaknesses across attack types. Finally, false alarms can reduce the practical usefulness of intrusion detection systems because excessive alerts place an additional burden on administrators.

This work therefore proposes an automated NID architecture combining feature selection, class balancing, multiple classifiers, and stacking-based ensemble learning. The stated objective is to improve detection accuracy while minimizing false alarms in IoT environments.

The main contributions of this work are:

1. Development of an automated ML-based NID framework for IoT environments.
2. Integration of multiple feature-selection strategies to identify relevant network attributes.
3. Use of SMOTE to mitigate class imbalance.
4. Comparative use of several conventional ML classifiers.
5. Integration of classifiers through a stacking-based ensemble.

6. Development of a Flask-based user interface for practical interaction with the trained detection model.

II. RELATED WORK

A. Intrusion Detection in IoT

Intrusion detection has long been considered an important component of computer and network security. Denning's early work established intrusion detection as a mechanism for identifying abnormal or suspicious behavior in computer systems [9]. With the growth of IoT, intrusion detection has become more challenging because IoT environments involve heterogeneous devices, constrained resources, dynamic traffic patterns, and diverse attack surfaces.

Recent research has investigated ML and DL techniques for IoT intrusion detection. Sharma *et al.* studied anomaly-based IoT attack detection using deep learning [2], while Sharma *et al.* also surveyed deep-learning-based anomaly detection techniques for IoT [6]. A broader systematic study by Ahmad *et al.* examined ML and DL approaches for network intrusion detection [16]. These studies demonstrate the increasing importance of automated learning-based security mechanisms.

This work literature also identifies approaches based on RNN, GRU, LSTM, and attention mechanisms. GRU-based detection has been investigated for Software-Defined Networks, while LSTM and attention mechanisms have been applied to anomaly detection [11], [12]. The source project observes that these approaches can achieve strong classification performance but notes that reliance on individual architectures may limit the ability to exploit complementary model strengths.

B. Feature Selection

Feature selection is important because network datasets can contain attributes that contribute little to classification. Selecting relevant features can reduce dimensionality computational requirements and unnecessary model complexity.

This wrok proposes three categories of feature selection: filter methods, wrapper methods, and embedded methods. The embedded approach incorporates Lasso, Random Forest, and Recursive Feature Elimination (RFE).

Abu Alghanam *et al.* proposed an improved pigeon-inspired optimization feature-selection approach combined with ensemble learning for IoT intrusion detection [1]. Their work evaluated the approach on multiple intrusion-detection datasets, including UNSW-NB15, and considered metrics such as F-score, accuracy, AUC, false-positive rate, and true-positive rate.

Feature selection is particularly relevant to IoT because reducing the number of input attributes can potentially lower computational overhead and support deployment on resource-constrained devices.

C. Class Imbalance and SMOTE

Intrusion datasets frequently contain unequal numbers of normal and malicious samples. A classifier trained directly on

such data may become biased toward the majority class. SMOTE addresses this problem by generating synthetic examples for minority classes rather than simply duplicating existing minority samples [31]. The original SMOTE study demonstrated its usefulness for classification under imbalanced conditions.

The proposed project uses Synthetic Minority Over-sampling Technique to increase the representation of minority intrusion classes and reduce training bias. The source specifically states that oversampling is intended to improve the detection of rare intrusions.

D. Ensemble Learning

Different ML algorithms may capture different characteristics of network traffic. Ensemble learning attempts to combine these complementary capabilities. The proposed system uses a Stacking Classifier that combines predictions from multiple models. The project describes an ensemble involving boosted decision trees, Random Forest bagging, and LightGBM.

Recent research also supports the use of stacking for IoT intrusion detection. A recent stacked-ensemble IDS study evaluated its approach on NB-AIoT and UNSW-NB15 and reported strong classification performance, while emphasizing feature selection and computational efficiency.

III. PROBLEM STATEMENT AND RESEARCH GAP

The rapid growth of interconnected IoT devices has resulted in increasingly complex network environments. According to the source project, conventional security mechanisms face difficulties in handling sophisticated attacks and the dynamic characteristics of IoT networks. High false-alarm rates and class imbalance further reduce the effectiveness of existing intrusion detection systems.

Existing research has demonstrated the effectiveness of individual ML and DL architectures, but the project identifies several limitations:

- Dependence on a limited set of neural architectures.
- Insufficient exploitation of ensemble learning.
- Potential bias caused by imbalanced data.
- Incomplete optimization of feature selection.
- Difficulty maintaining robust detection across diverse intrusion patterns.

The research gap addressed by this work is therefore the integration of feature selection + class balancing + multiple ML classifiers + stacking ensemble learning into one automated IoT intrusion detection framework.

IV. PROPOSED SYSTEM

A. System Overview

The proposed system is an automated NID framework designed to classify network traffic as benign or malicious. The overall processing pipeline consists of:

Dataset → Data Pre-processing → Label Encoding → Feature Selection → SMOTE → Train/Test Split → ML Models →

Stacking Ensemble → Prediction → User Interface

The system contains data-loading, pre-processing, visualization, label-encoding, feature-selection, oversampling, model-generation, authentication, and prediction modules.

B. Dataset

The proposed system is based on the UNSW-NB15 dataset. The dataset was created using a combination of real modern normal activities and synthesized contemporary attack behaviors. It contains nine attack categories: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms. It contains 49 features plus the class label.

C. Data Preprocessing

Data preprocessing is performed before model training. The project describes duplicate removal, cleaning, normalization, and transformation of categorical attributes.

Categorical attributes are converted into numerical representations through label encoding or related encoding operations. Numerical values are normalized so that attributes with substantially different scales do not disproportionately influence model learning.

The sample implementation in the source additionally demonstrates numerical scaling using RobustScaler, categorical encoding, conversion of the target into binary labels, and dimensionality reduction using PCA in an example preprocessing workflow.

D. Feature Selection

Three broad feature-selection approaches are incorporated:

1) Filter Method

Filter techniques evaluate features using statistical characteristics or relationships with the target variable. They are computationally efficient because feature evaluation occurs independently of the final classifier.

2) Wrapper Method

Wrapper techniques evaluate subsets of features using a predictive model. Although potentially more computationally expensive than filter methods, they can identify feature subsets that work well with a specific classifier.

3) Embedded Method

The proposed embedded strategy combines Lasso, Random Forest, and RFE. The project uses these techniques to identify relevant features and improve the quality of the model input.

Feature selection is expected to reduce unnecessary information and improve the efficiency of the downstream classification stage.

E. Class Balancing Using SMOTE

After feature preprocessing, class imbalance is addressed using SMOTE. Let xi be a minority-class sample and xnn be one of its nearest minority-class neighbors. A synthetic sample can be represented conceptually as

$$x_{new} = x_i + \lambda(x_{nn} - x_i$$

where

$$0 \leq \lambda \leq 1.$$

This creates synthetic minority observations between existing minority samples rather than merely replicating observations. The objective is to provide classifiers with a more representative training distribution.

The original SMOTE methodology specifically targets classification problems in which minority examples are underrepresented.

The applies this concept to intrusion data so that rare attacks receive greater representation during model training.

V. MACHINE-LEARNING MODELS

The proposed framework evaluates several classification algorithms.

A. Random Forest

Random Forest is an ensemble method based on multiple decision trees. Each tree produces a prediction and the ensemble combines these predictions. The project uses Random Forest to classify network traffic and identify potentially malicious patterns.

Its ensemble structure can reduce the sensitivity of the final prediction to an individual decision tree and can provide a robust baseline for intrusion classification.

B. Decision Tree

A Decision Tree recursively partitions the feature space according to feature values. The resulting tree structure provides an interpretable classification process. In the proposed NID system, Decision Trees are used to classify network traffic and identify suspicious behavior.

C. AdaBoost

AdaBoost combines weak classifiers into a stronger ensemble by repeatedly emphasizing observations that were incorrectly classified in previous iterations. The proposed system uses AdaBoost to improve sensitivity to attack patterns.

D. Bernoulli Naive Bayes

Bernoulli Naive Bayes is a probabilistic classifier suited to binary-valued features. It is included because of its computational simplicity and ability to provide rapid classification.

E. K-Nearest Neighbors

KNN classifies an observation according to the labels of nearby observations in feature space. The project employs KNN to identify unusual network patterns by comparing incoming samples with previously observed samples.

F. Logistic Regression

Logistic Regression estimates the probability of a binary outcome from one or more predictors. In the proposed system, it is used to estimate whether network activity represents an intrusion.

VI. STACKING-BASED ENSEMBLE MODEL

The central extension of the proposed system is the Stacking Classifier. Instead of relying on one classifier, stacking uses predictions from multiple base models as input to a higher-level model.

The architecture described in the project combines:

- Boosted Decision Trees,
- Random Forest-based bagging, and
- LightGBM.

Conceptually, if the predictions from K base classifiers are

$p1(x),p2(x),...,pK(x),$

the meta-classifier receives

$P(x)=[p1(x),p2(x),...,pK(x)]$

and produces the final classification

$y\hat{}=g(P(x))$.

The advantage of this approach is that different classifiers can learn complementary representations of network behavior. The source project reports that combining model predictions through stacking improves overall attack-detection performance and robustness.

VII. SYSTEM ARCHITECTURE AND IMPLEMENTATION

A. System Architecture

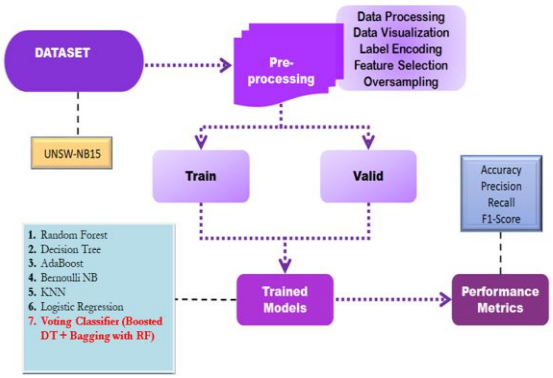


Fig.1.system architecture

B. Software Environment

The project specifies the following environment:

- **Programming language:** Python
- **Frontend:** Flask
- **Backend/development environment:** Jupyter Notebook
- **Database:** SQLite3
- **Frontend technologies:** HTML, CSS, JavaScript, Bootstrap 4
- **Development environment:** Anaconda

The hardware requirements identified in the project include a Windows operating system, an Intel i5 or higher processor, at least 8 GB RAM, and approximately 25 GB of local storage.

C. User Interface

A Flask-based frontend is included to facilitate user testing and prediction. The project also includes user registration and authentication so that users can access the prediction functionality.

The functional requirements include:

1. Data collection.
2. Data preprocessing.

3. Training and testing.
4. Model generation.
5. Prediction.

VIII. PERFORMANCE EVALUATION

A network intrusion detection system should not be evaluated using accuracy alone. The project identifies performance evaluation across the implemented algorithms.

The principal evaluation measures can be expressed using the confusion matrix:

Actual / Predicted	Normal	Attack
Normal	TN	FP
Attack	FN	TP

where:

- **TP:** attacks correctly identified as attacks,
- **TN:** normal traffic correctly identified as normal,
- **FP:** normal traffic incorrectly classified as attacks,
- **FN:** attacks incorrectly classified as normal.

Accuracy

Accuracy=TP+TN+FP+FNTP+TN

Precision

Precision=TP+FPTP

Recall

Recall=TP+FNTP

F1-Score

F1=2×Precision+RecallPrecision×Recall

False Positive Rate

FPR=FP+TNFP

These metrics are important because an intrusion detector must detect attacks while avoiding excessive false alarms. The literature included in the project also evaluates intrusion detection systems using measures such as F-score, accuracy, AUC, FPR, and TPR.

IX. RESULTS AND DISCUSSION

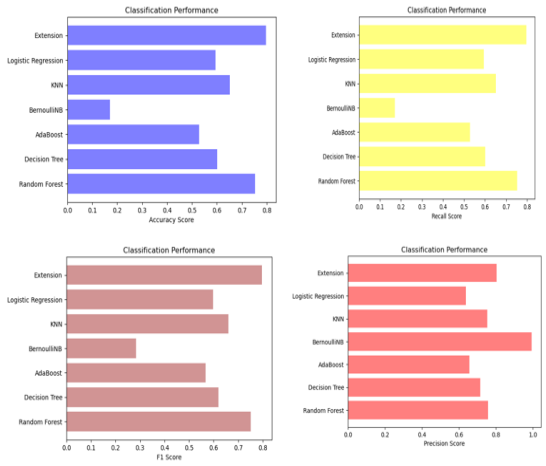


Fig.2. Comparison Graph – Filter FS

Fig.3. Comparison Graph –Wrapper Based FS

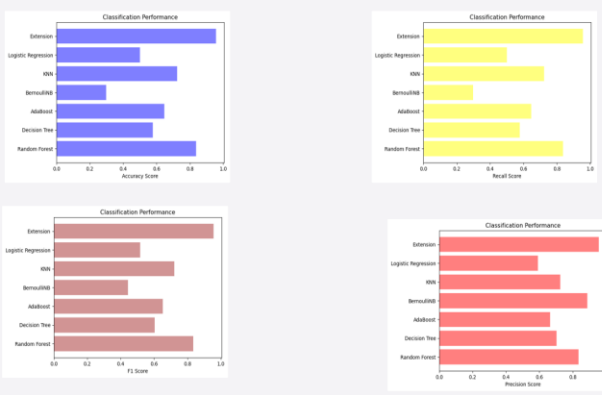


Fig.4. Comparison Graph –Embedded Based FS

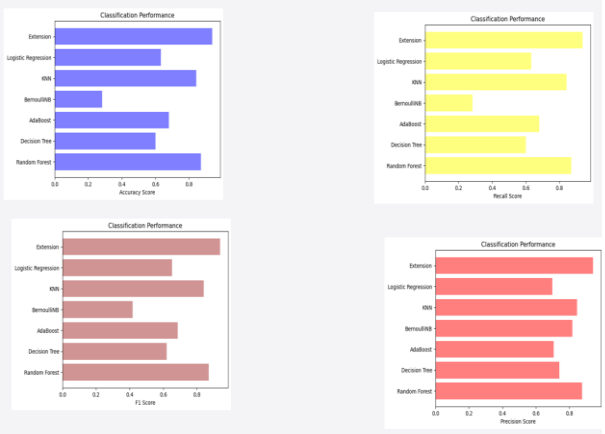
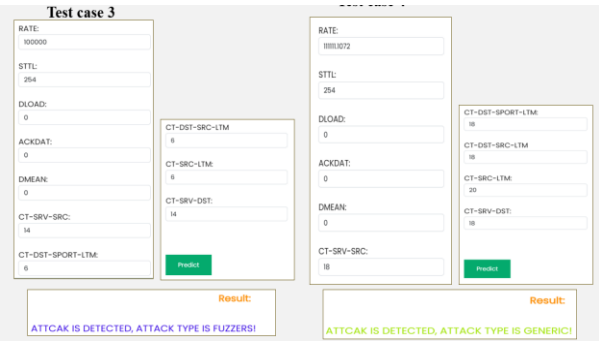


Fig.5.Comparison Graph – PCC Based FS



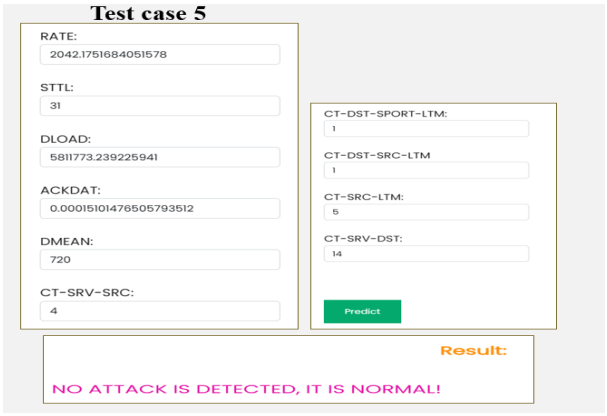


TABLE I

Summary of The Proposed Approach

Component	Proposed Approach
Dataset	UNSW-NB15
Data preprocessing	Cleaning, encoding, normalization
Feature selection	Filter, Wrapper, Embedded
Embedded techniques	Lasso, Random Forest, RFE
Class balancing	SMOTE
Individual classifiers	RF, DT, AdaBoost, Bernoulli NB, KNN, LR
Ensemble	Stacking Classifier
Ensemble components	Boosted DT, RF Bagging, LightGBM
Interface	Flask
Authentication	User signup/login
Primary objective	Improved intrusion detection and reduced false alarms

The combination of feature selection and class balancing is important because it addresses two major sources of model degradation before classification. The ensemble stage then attempts to exploit the complementary strengths of different models.

The project's reported advantage of stacking is consistent with recent IoT intrusion-detection research, where ensemble learning and feature selection have been used together to improve detection performance while reducing computational overhead.

The use of UNSW-NB15 also provides a contemporary benchmark containing multiple attack categories rather than relying exclusively on older intrusion datasets. The dataset was specifically designed to address limitations associated with older benchmark datasets and includes modern normal and synthesized attack traffic.

X. Advantages of The Proposed System

The proposed framework provides several advantages.

1. Improved Feature Representation

Using multiple feature-selection approaches can reduce irrelevant information and improve the quality of the input data. The project identifies Filter, Wrapper, Lasso, Random Forest, and RFE-based approaches as mechanisms for obtaining relevant features.

2. Reduced Class Bias

SMOTE increases minority-class representation and is intended to reduce bias toward majority classes. This is particularly important when rare attack categories are difficult to detect.

3. Multiple Classification Perspectives

The framework evaluates several fundamentally different classifiers rather than depending on a single algorithm.

4. Ensemble Robustness

Stacking combines predictions from several models, potentially improving robustness compared with an individual classifier.

5. Practical User Interface

The Flask interface provides a practical mechanism through which users can interact with the trained detection system.

6. Scalability Considerations

The undertaking identifies scalability as a non-functional requirement, with the system expected to accommodate increasing numbers of IoT devices and changing data volumes.

XI. LIMITATIONS

Although the proposed system provides an integrated approach, several limitations should be acknowledged.

First, the evaluation is primarily based on a benchmark dataset. Real IoT networks may exhibit traffic distributions and attack behaviors that differ from the training data.

Second, the project document does not provide complete numerical experimental results for every classifier. Consequently, quantitative comparisons cannot be independently reproduced from the supplied document alone.

Third, the source's sample implementation includes code based on the NSL-KDD dataset in addition to the proposed UNSW-

NB15-oriented system. Therefore, a publication-ready implementation should clearly separate exploratory/sample code from the final UNSW-NB15 experimental pipeline.

Fourth, SMOTE can alter the distribution of training data. Its use should therefore be restricted to the training partition to avoid data leakage during evaluation.

Finally, real-time deployment on resource-constrained IoT or edge devices requires additional evaluation of memory usage, latency, throughput, and energy consumption.

XII. SECURITY AND DEPLOYMENT CONSIDERATIONS

The system is designed with several operational requirements. Performance requires efficient processing of incoming IoT data with low latency, while reliability requires continuous monitoring and error handling. Usability requires an interface that allows administrators to interpret results without excessive technical complexity. Security requirements include access control, secure data transmission, and protection against unauthorized access.

For deployment in a real IoT environment, the trained model could be positioned at an edge or gateway layer. This would allow suspicious traffic to be analyzed before it reaches more sensitive internal services. Research has also investigated intrusion detection at the edge, including the use of specialized edge hardware for IoT security [22].

However, deployment should consider computational constraints. IoT devices frequently have lower processing and memory capabilities than conventional servers. Feature selection and lightweight classification can therefore be important for practical deployment.

XIII. FUTURE WORK

The source project identifies several future research directions. These include the application of CNN and RNN models, anomaly-detection approaches, hybrid models, and unsupervised learning techniques.

Future work can therefore extend the proposed framework in the following directions:

1. **Deep-learning integration:** CNN, RNN, LSTM, and GRU models could be integrated with the current ensemble architecture.
2. **Online learning:** Models could be updated continuously as new traffic patterns become available.
3. **Unsupervised detection:** Unsupervised and self-supervised approaches could identify previously unseen attacks.
4. **Adversarial robustness:** Adversarial training could be investigated to improve resistance to intentionally manipulated traffic.
5. **Edge deployment:** The model could be optimized for edge hardware and resource-constrained IoT gateways.
6. **Explainable AI:** Explainability techniques could help administrators understand why particular traffic is classified as malicious.

7. **Real-time evaluation:** Future experiments should measure inference latency, throughput, memory consumption, and energy requirements.
8. **Cross-dataset validation:** Testing on additional datasets would provide stronger evidence of generalization beyond UNSW-NB15.

XIV. CONCLUSION

In conclusion, the proposed automated Network Intrusion Detection (NID) system for Internet of Things (IoT) environments addresses the pressing challenges of security threats and data integrity. Through the implementation of advanced feature selection techniques, such as Filter and Wrapper Methods alongside an Embedded approach using Lasso with Random Forest and Recursive Feature Elimination (RFE), we enhance the dataset's quality and relevance for training machine learning models. The integration of the Synthetic Minority Over-sampling Technique (SOMTE) effectively mitigates class imbalance, allowing for more accurate detection of intrusions. Among the algorithms evaluated, the Stacking Classifier demonstrates superior performance, combining the strengths of Boosted Decision Trees, Bagging with Random Forest, and LightGBM to achieve a high level of detection accuracy. This robust mechanism not only improves the detection rates but also significantly reduces false alarm rates, addressing critical concerns in IoT security. Ultimately, this system serves as a valuable tool for system administrators, providing an effective solution for safeguarding IoT networks against an ever-evolving landscape of cyber threats while ensuring the integrity and confidentiality of sensitive data.

In future work, we aim to further enhance the automated Network Intrusion Detection (NID) system by exploring advanced techniques such as deep learning algorithms, including convolutional neural networks (CNNs) and recurrent neural networks (RNNs), to improve detection accuracy. Additionally, we will investigate the integration of anomaly detection methods and hybrid models that combine multiple algorithms for better adaptability to diverse attack patterns. Exploring unsupervised learning techniques could also provide valuable insights into emerging threats within IoT environments.

REFERENCES

[1] O. Abu Alghanam, W. Almobaideen, M. Saadeh, and O. Adwan, “An improved PIO feature selection algorithm for IoT network intrusion detection system based on ensemble learning,” *Expert Systems with Applications*, vol. 213, Art. no. 118745, Mar. 2023.

[2] B. Sharma, L. Sharma, C. Lal, and S. Roy, “Anomaly based network intrusion detection for IoT attacks using deep learning technique,” *Computers & Electrical Engineering*, vol. 107, Art. no. 108626, Apr. 2023.

[3] I. Debicha, R. Bauwens, T. Debatty, J.-M. Dricot, T. Kenaza, and W. Mees, “TAD: Transfer learning-based multi-adversarial detection of evasion attacks against network intrusion detection systems,” *Future Generation Computer Systems*, vol. 138, pp. 185–197, Jan. 2023.

[4] W. Wang, S. Jian, Y. Tan, Q. Wu, and C. Huang, "Robust unsupervised network intrusion detection with self-supervised masked context reconstruction," *Computers & Security*, vol. 128, Art. no. 103131, May 2023.

[5] W. D. Xiong, K. L. Luo, and R. Li, "AIDTF: Adversarial training framework for network intrusion detection," *Computers & Security*, vol. 128, Art. no. 103141, May 2023.

[6] B. Sharma, L. Sharma, and C. Lal, "Anomaly detection techniques using deep learning in IoT: A survey," in *Proc. Int. Conf. Computational Intelligence and Knowledge Economy (ICCIKE)*, 2019, pp. 146–149, doi: 10.1109/ICCIKE47802.2019.9004362.

[7] E. Altulaihan, M. A. Almaiah, and A. Aljughaiman, "Cybersecurity threats, countermeasures and mitigation techniques on the IoT: Future research directions," *Electronics*, vol. 11, no. 20, Art. no. 3330, Oct. 2022.

[8] R. A. Nafea and M. Amin Almaiah, "Cyber security threats in cloud: Literature review," in *Proc. Int. Conf. Information Technology (ICIT)*, 2021, pp. 779–786.

[9] D. E. Denning, "Intrusion detection revisited," *ACM Transactions on Information Systems Security*, vol. 5, no. 4, pp. 412–417, 1987.

[10] M. Al-Zewairi, S. Almajali, and A. Awajan, "Experimental evaluation of a multi-layer feed-forward artificial neural network classifier for network intrusion detection system," in *Proc. Int. Conf. New Trends in Computing Sciences (ICTCS)*, 2017, pp. 167–172.

[11] M. V. O. Assis, L. F. Carvalho, J. Lloret, and M. L. Proença, "A GRU deep learning system against attacks in software defined networks," *Journal of Network and Computer Applications*, vol. 177, Art. no. 102942, Mar. 2021.

[12] P. Lin, K. J. Ye, and C. Z. Xu, "Dynamic network anomaly detection system by using deep learning techniques," in *Proc. IEEE Cloud Computing (CLOUD)*, 2019, pp. 161–176.

[13] T. H. Hai and L. H. Nam, "A practical comparison of deep learning methods for network intrusion detection," in *Proc. Int. Conf. Electrical, Communication and Computer Engineering (ICECCE)*, 2021, pp. 1–6.

[14] A. Meliboev, J. Alikhanov, and W. Kim, "Performance evaluation of deep learning based network intrusion detection system across multiple balanced and imbalanced datasets," *Electronics*, vol. 11, no. 4, Art. no. 515, Feb. 2022.

[15] R. Ravinder Reddy, K. Ayyappa Reddy, C. Madan Kumar, and Y. Ramadevi, "Detection of network anomaly sequences using deep recurrent neural networks," in *Smart Computing Techniques and Applications*, vol. 2, 2021, pp. 605–615.

[16] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," *Transactions on Emerging Telecommunications*

Technologies, vol. 32, no. 1, Art. no. e4150, Jan. 2021.

[17] Y. Fu, Y. Du, Z. Cao, Q. Li, and W. Xiang, "A deep learning model for network intrusion detection with imbalanced data," *Electronics*, vol. 11, no. 6, Art. no. 898, Mar. 2022.

[18] P. Rajesh Kanna and P. Santhi, "Unified deep learning approach for efficient intrusion detection system using integrated spatial–temporal features," *Knowledge-Based Systems*, vol. 226, Art. no. 107132, Aug. 2021.

[19] P. R. Kanna and P. Santhi, "Hybrid intrusion detection using MapReduce based black widow optimized convolutional long short-term memory neural networks," *Expert Systems with Applications*, vol. 194, Art. no. 116545, May 2022.

[20] M. Ozkan-Okay, R. Samet, Ö. Aslan, and D. Gupta, "A comprehensive systematic literature review on intrusion detection systems," *IEEE Access*, vol. 9, pp. 157727–157760, 2021.

[21] A. Patel, H. Alhussian, J. M. Pedersen, B. Bounabat, J. C. Júnior, and S. Katsikas, "A nifty collaborative intrusion detection and prevention architecture for smart grid ecosystems," *Computers & Security*, vol. 64, pp. 92–109, Jan. 2017.

[22] S. Hosseininoorbin, S. Layeghy, M. Sarhan, R. Jurdak, and M. Portmann, "Exploring edge TPU for network intrusion detection in IoT," *Journal of Parallel and Distributed Computing*, vol. 179, Art. no. 104712, Sep. 2023.

[23] M. A. Khan, "HCRNNIDS: Hybrid convolutional recurrent neural network-based network intrusion detection system," *Processes*, vol. 9, no. 5, Art. no. 834, May 2021.

[24] B. Cao, C. Li, Y. Song, Y. Qin, and C. Chen, "Network intrusion detection model based on CNN and GRU," *Applied Sciences*, vol. 12, no. 9, Art. no. 4184, Apr. 2022.

[25] Z. Wang, K. W. Fok, and V. L. L. Thing, "Machine learning for encrypted malicious traffic detection: Approaches, datasets and comparative study," *Computers & Security*, vol. 113, Art. no. 102542, Feb. 2022.

[26] J. Zhang, Y. Ling, X. Fu, X. Yang, G. Xiong, and R. Zhang, "Model of the intrusion detection system based on the integration of spatial–temporal features," *Computers & Security*, vol. 89, Art. no. 101681, Feb. 2020.

[27] S. B. Bastola, S. Shakya, and S. Sharma, "Distributed denial of service attack detection on software defined networking using deep learning," in *Proc. Int. Conf. Advanced Computing, Communications and Informatics (ICACCI)*, 2021, pp. 13–16.

[28] P. Zhao, Z. Fan, Z. Cao, and X. Li, "Intrusion detection model using temporal convolutional network blend into attention mechanism," *International Journal of Information Security and Privacy*, vol. 16, no. 1, pp. 1–20, Oct. 2021.

[29] M. Aldwairi and D. Alansari, "N-grams exclusion and inclusion filter for intrusion detection in internet of energy big

data systems," *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 3, Art. no. e3711, Mar. 2022.

[30] A. I. Siam *et al.*, "Secure health monitoring communication systems based on IoT and cloud computing for medical emergency applications," *Computational Intelligence and Neuroscience*, vol. 2021, pp. 1–23, Dec. 2021.

[31] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002, doi: 10.1613/jair.953.

[32] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proc. 2015 Military Communications and Information Systems Conference (MilCIS)*, Canberra, Australia, 2015, pp. 1–6, doi: 10.1109/MILCIS.2015.7348942.