International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Enhanced IoT Security: Multi-Stage Intrusion Detection with Hybrid Residual Networks

Mohit Jain* , Dr Rajesh Nigam**

Submitted:20/04/2024 **Revised:**01/06/2024 **Accepted:**10/06/2024

Abstract -With the rapid expansion of Internet of Things (IoT) networks, intrusion detection has become increasingly critical due to the heterogeneous nature and vulnerability of connected devices. This study proposes a hybrid IoT Intrusion Detection System (IDS) modeled as a multi-stage framework, encompassing data preprocessing, feature extraction, and deep learning-based classification. Missing data are addressed using mean imputation, and categorical and numerical features are standardized through label encoding and Z-score normalization to ensure uniform scaling. Principal Component Analysis (PCA) is applied for dimensionality reduction, preserving essential variance while reducing redundancy. The classification function is implemented using several state-of-the-art deep learning architectures, including MobileNet, Inception, VGG16, VGG19, DenseNet, GoogLeNet, AlexNet, ResNet34, and ResNet50. A hybrid model combining ResNet34 and ResNet50 with feature-level fusion and attention mechanisms is employed to enhance learning depth and feature representation. The system is trained using cross-entropy loss and evaluated on training and testing subsets. Results demonstrate improved detection accuracy, reduced false alarms, and effective classification of normal and anomalous IoT network activities.

Keywords: *IoT Security, Intrusion Detection System (IDS), Hybrid Deep Learning, ResNet34, ResNet50, Feature Extraction, Principal Component Analysis (PCA), Data Preprocessing, Anomaly Detection, Feature Fusion, Attention Mechanism, Network Security, Deep Neural Networks*

I Introduction

The rapid development of the Internet of Things (IoT) has completely transformed many industries, such as smart homes, smart agriculture, healthcare, and more [**1**]. According to survey data, the number of IoT devices is projected to exceed 4.1 billion by 2025 [**2**]. In everyday life, IoT devices play a crucial role in people's lives. However, the extensive connectivity of these devices to the internet exposes them to various security risks. For example, IoT devices exchange information over the internet and are susceptible to numerous network attacks, compromising their security. According to a report by Nozomi Networks, new IoT botnet attacks increased rapidly in the first half of 2020, with 57% of IoT devices being vulnerable targets [**3**]. Furthermore, attackers can launch denial-of-service (DoS) attacks, depleting network and device resources [**4**]. Therefore, enhancing the security of IoT devices has become a critical area of research [**5**]. To mitigate the risks posed by different types of attacks, researchers are developing intrusion detection systems to identify malicious behavior in networks. Intrusion detection systems monitor systems in real-time and issue warnings in case of any anomalies, thereby enhancing the security of communication.

In recent years, machine learning, with its rapid development, has found extensive applications in the field of intrusion detection [**6**,**7**]. Machine learning algorithms offer unique advantages compared to traditional detection methods. They can not only learn complex patterns and rules from large volumes of data but also handle high-dimensional and nonlinear data, making them more suitable for intrusion detection in complex systems. Furthermore, with the advancement of networks, a significant amount of network data, including samples of various intrusion and abnormal behaviors, has been accumulated. This rich dataset provides ample training samples for machine learning, ensuring excellent detection performance of machine learning algorithms. However, despite the achievements of machine learning algorithms, there

Phd Scholar HOD Computer Science & Technology, SGU***
*bmctmohitcs@gmail.com**

are still some challenges and issues that need to be addressed.

The Internet of Things (IoT) includes physical objects with sensors and software that connects and shares data with other devices through the internet. These features enable them to collect, transmit, and receive data. Typically, these data are utilized for interaction with and control and observation of the real environment. Data collected through these devices can be analyzed locally or sent to the cloud via gateways or edge devices [**1**]. IoT devices facilitate communication, data sharing, and automated actions across various domains, including homes, industries, cities, healthcare, agriculture, transportation, and retail, leading to their extensive deployment [**2**].

This growth has also led to more traffic in cyberspace and a rise in advanced intrusion attacks. IoT system attacks can lead to significant problems that impact both the targeted devices and the broader network infrastructure and compromise data integrity and privacy and even pose risks to physical safety. These attacks exploit vulnerabilities within IoT systems, making it easy to launch various cyber threats such as DDoS attacks, botnets, malware infections, and ransomware [**3**]. It is essential to safeguard the IoT infrastructure against potential threats to minimize the risks of intrusion attacks on IoT systems. This can be accomplished by implementing intrusion detection systems (IDSs). An IDS is like a digital watchdog for networks. It carefully watches for any unusual activity and alerts administrators if it finds anything suspicious. Moreover, an advanced IDS can spot both known and new types of threats, making it essential for keeping networks safe.

A conventional IDS operates using signature and anomaly detection methods [**4**]. These traditional approaches have several inherent limitations. Firstly, the rules or signatures used for detection require frequent updates to match the continuously changing environment of cyber threats. Failure to update these rules promptly can lead to missed detection of new or modified attacks. Second, these systems' low accuracy frequently results in a high rate of false positives, which mistakenly classify benign activities as threats while failing to detect actual threats. Thirdly, conventional IDSs tend to generate a high number of false alarms. This can make security analysts feel exhausted from dealing with alerts, causing them to miss real threats among all the false ones. To address these challenges, implementing IDSs with machine learning (ML) and deep learning (DL) techniques has been proposed [**5**,**6**,**7**,**8**,**9**,**10**,**11**].

ML and DL techniques have the potential to significantly improve the performance of IDSs. By training these models on large datasets of network traffic data, they are able to recognize intricate patterns and irregularities that could point to malevolent behavior [**12**]. Unlike traditional rules-based systems, ML and DL models can adapt and evolve as new threats emerge, providing a more robust and proactive approach to threat detection. One key advantage of using ML and DL for IDSs is their ability to handle high-dimensional and heterogeneous data sources. IoT systems generate vast amounts of data from various devices and sensors, making it challenging for traditional methods to effectively analyze and correlate this information. ML and DL models can process and extract meaningful insights from these diverse data sources, enabling more comprehensive and accurate detection of potential threats across the entire IoT infrastructure.

However, several challenges may arise that can impact the performance and effectiveness of these models. First, when a model overfits to training data and is unable to generalize well to new, unknown data, it performs poorly in terms of identifying real threats or producing an excessive number of false positives [**13**]. Second, the presence of unimportant or irrelevant features in network traffic data can introduce noise and obscure meaningful patterns [**14**]. Third, large datasets with many features, which are common in IoT and network environments, can lead to higher computational costs and longer training times for ML/DL models, which can be particularly challenging in resource-constrained IoT devices or edge computing environments with limited computational power, storage, and memory [**15**]. Addressing these challenges is crucial for the successful implementation of ML/DL techniques for IDSs in IoT systems and requires careful model selection, tuning, and optimization to ensure optimal performance, accuracy, and efficiency while considering the constraints of the target environment.

Table on literature study

Author(s) & Year	Objective / Focus Area	Methodology / Dataset Used	Key Findings / Results
Alahmadi et al. (2023)	Developed an ML- and DL-based IDS to detect DDoS/DoS attacks in IoT networks.	Used **Bot-IoT dataset**; addressed class imbalance; performed binary and multiclass classification using timestamp features.	Achieved **>99% accuracy**; **Decision Tree** and **MLP** models were most effective for identifying DDoS and DoS attacks.
Alfahaid et al. (2025)	Reviewed studies on **DDoS detection in IoT-based networks** using ML and DL techniques.	Comprehensive literature review of ML-based intrusion detection methods across various IoT applications.	Highlighted the role of **AI (ML & DL)** in IoT attack detection; provided an extensive reference base for researchers; emphasized detection of DDoS attacks using learning-based approaches.
Alkhudaydi et al. (2023)	Investigated **ML and DL algorithms** for detecting malware in IoT network traffic.	Used **BoT-IoT dataset** with **SMOTE** to address class imbalance; evaluated 10 ML models and 4 DL models (LSTM, GRU, RNN).	**CatBoost (98.19%)** and **XGBoost (98.50%)** achieved highest accuracies; combining ML/DL with SMOTE improved detection of IoT network intrusions.
Almaraz-Rivera et al. (2022)	Proposed DL-based IDS for **DDoS detection in smart agriculture (Agriculture 4.0)** environments.	Implemented **CNN**, **DNN**, and **RNN** models using **CIC-DDoS2019** and **TON_IoT** datasets.	DL models achieved high performance in both binary and multiclass classification; demonstrated effectiveness in **Agriculture IoT security**.
Altulaihan et al. (2024)	Conducted **systematic literature review (SLR)** on anomaly-based IDS using DL in IoT environments.	Analyzed 2116 papers from major databases; shortlisted 26 key studies; reviewed 7 DL techniques (CNN, RNN, AE, LSTM, etc.).	Found **supervised DL** techniques perform best; identified trends and gaps for future anomaly-based IDS research in IoT.
Alsoufi et al. (2021)	Reviewed **learning-based intrusion detection** for IoT systems facing diverse cyberattacks.	Comprehensive survey covering DoS, DDoS, U2R, R2L, MITM, and Botnet attacks; analyzed ML and DL detection methods.	Highlighted vulnerabilities in IoT; demonstrated **DL superiority** in detecting unknown attacks; suggested future directions for hybrid learning-based IDS frameworks.

Ii Research Methodology

The proposed **IoT Intrusion Detection System (IDS)** is mathematically modeled as a multi-stage data processing and classification framework designed to detect and classify anomalous network activities in IoT environments. The overall model incorporates data preprocessing, feature extraction, classification, and performance evaluation. The mathematical formulation of each stage is explained below.

1. Dataset Representation

Let the dataset be represented as: Missing data values are replaced using the **mean imputation** method:

$$X_{ij} = \begin{cases} x_{ij} \\ \frac{1}{n}\sum_{i=1}^{n} x_{ij} \end{cases}$$

where x_{ij} represents the j^{th} feature of the i^{th} sample.

Label Encoding and Normalization

Categorical variables are encoded into numerical form using **label encoding**, and all numerical features are standardized using **Z-score normalization** to ensure uniform scaling:

$$x'_{ij} = \frac{x_{ij-\mu_j}}{\sigma_j}$$

where μ_j and σ_j denote the mean and standard deviation of the j^{th} feature, respectively. This transformation ensures that all features contribute equally to the model learning process and prevents bias caused by varying feature scales.

Feature Extraction using Principal Component Analysis (PCA)

Feature extraction is employed to reduce data dimensionality while preserving the essential variance in the dataset. **Principal Component Analysis (PCA)** is used to transform the original feature space into a smaller set of uncorrelated variables known as **principal components**.

Let X denote the standardized data matrix with dimensions N×d. The covariance matrix of X is given by:

$C= \frac{1}{N-1} X^T X$

PCA involves eigenvalue decomposition of C:

$C_{vk}=\lambda_k v_k$

where v_k represents the k^{th} eigenvector (principal component) and λ_k is the corresponding eigenvalue indicating the amount of variance captured.

By selecting the top rrr eigenvectors corresponding to the largest eigenvalues, we form the projection matrix

$V_r=[v_1,v_2,\ldots,v_r]$. The reduced feature matrix Z is then computed as:

$Z=XV_r$

Here, Z represents the transformed dataset with r key features capturing the most relevant variance and structural information of the original data.

4. Data Splitting

After feature extraction, the dataset is divided into **training** and **testing** subsets to facilitate model learning and evaluation. This can be mathematically represented as:

$$D=D_{train} \cup D_{test}, D_{train} \cap D_{test} = \emptyset$$

where D_{train} contains a portion of the data (usually 70–80%) used for training, while D_{test} contains the remaining portion (20–30%) used to evaluate model performance on unseen data. This ensures unbiased evaluation and prevents overfitting.

5. Classification Model

The classification phase is responsible for learning patterns from the training data and predicting whether a given instance represents normal or intrusive behavior. Let the model be defined as a mapping function:

$\hat{y}=f(Z;\theta)$

where f represents the learning model parameterized by θ\thetaθ, which includes the network weights and biases.The classification function can be implemented using several deep learning models such as MobileNet, Inception, VGG16, VGG19, DenseNet, GoogLeNet, AlexNet, ResNet34, and ResNet50.

Hybrid Model Design (ResNet34 + ResNet50)

Both **ResNet34** and **ResNet50** are **residual neural networks** that overcome the vanishing gradient problem using *skip connections*.For an input feature Zi,

$H(Z_i)=F(Z_i,W)+Z_i$

Where $F(F(Z_i,W)$ represents the learned residual function, and Z_i is added directly to preserve essential features.

ResNet34 Output:

$f_{34}=g_{34}(Z_i;\theta_{34})$

ResNet50 Output:

$f_{50} = g_{50}(Zi;\theta_{50})$

where θ_{34} and θ_{50} represent the learnable parameters of ResNet34 and ResNet50, respectively.

Step 4: Feature Fusion Layer

To combine the strengths of both networks, a feature-level fusion is performed by concatenating their outputs:

$$F_{fusion} = [f_{34} \| f_{50}]$$

Optionally, an **attention layer** can be added to highlight the most important features:

$$f_{att} = \sigma(W_a \cdot f_{fusion} + b_a)$$

where σ is the sigmoid activation function that adjusts feature importance dynamically.

Step 5: Classification Layer

The fused feature vector f_{fusion} is passed through fully connected layers with activation functions (ReLU, Softmax) for classification

$$\hat{y} = \text{Softmax}(W_c \cdot f_{att} + b_c)$$

The output $\hat{y}$ represents the probability distribution over classes (e.g., normal or attack).

Step 6: Model Training and Loss Function

The model is trained using a **cross-entropy loss function**, defined as:

$$L = \frac{1}{N} \sum_{i=1}^{N} y_i \log(\hat{y})$$

where y_i is the true label and $\hat{y}$ is the predicted probability

For imbalanced intrusion data, a **weighted or focal loss** can be applied:

$$L_{focal} = -(1-\hat{y})\log(\hat{y})$$

where γ helps focus learning on difficult-to-classify samples.

Step 7: Output Prediction

After training, the hybrid model predicts whether the IoT network traffic is normal or malicious:

$$\text{Predicted Class} = \begin{cases} 0, & \textbf{if } \hat{y} < \mathbf{0.5} \\ 1, & \textbf{if } \hat{y} \geq \mathbf{0.5} \end{cases}$$

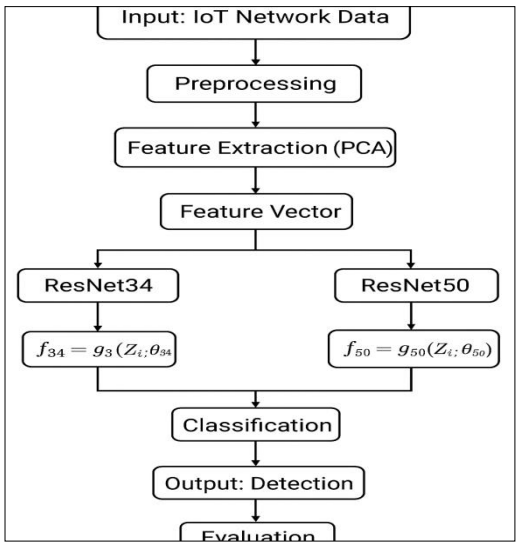

Figure1 proposed hybrid structure

III Proposed System

The proposed system presents an intelligent IoT-based Intrusion Detection System (IDS) that leverages deep learning techniques to detect and classify malicious network behaviors in IoT environments. The process begins with the selection and preprocessing of a public IoT intrusion detection dataset using tools such as Pandas. Missing values are addressed through mean imputation, categorical variables are encoded numerically, and data dimensionality is reduced using Principal Component Analysis (PCA). The refined dataset is then split into training and testing subsets to enable model learning and evaluation.

For classification, multiple state-of-the-art deep learning models are employed, including MobileNet, Inception, VGG16, VGG19, DenseNet, GoogLeNet, AlexNet, ResNet34, and ResNet50. Each model is trained to classify network activity as either normal or indicative of an attack. Additionally, a hybrid model combining ResNet34 and ResNet50 with feature fusion and attention mechanisms is implemented to enhance detection accuracy by leveraging complementary strengths of both networks.

The system outputs predictions and evaluates model performance using metrics such as accuracy, precision, recall, and F1-score. Comparative performance graphs are generated to visualize and analyze the effectiveness of each classifier, ensuring that the proposed IDS is both robust and efficient for real-world IoT applications.

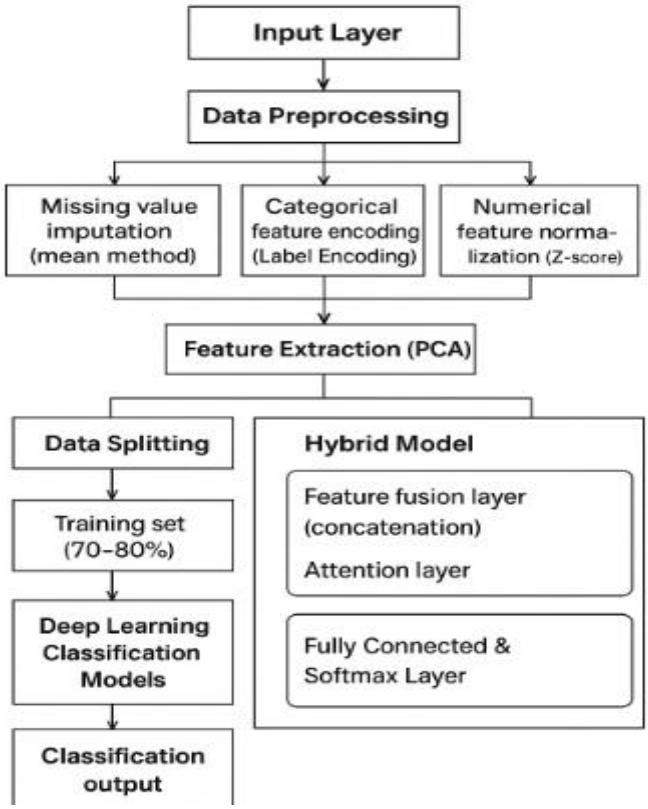

Figure 2 1 proposed system architecture

IV Result Discussion

The accuracy of the classifier can be evaluated with the use of a different evaluation metrics, which tally the number of accurate and inaccurate predictions that were generated based on values that are already known. A True Positive, abbreviated as TP, is one in which the model properly predict correct class. True Negative (TN) is a situation in which model properly predicts negative class. It is possible to have a False Positive, also known as an FP is one in which the model erroneously predict correct class. False Negative also known as a FN is a situation in which model erroneously predicts negative class.. In the proposed work, following evaluation metrics are used for performance assessment.

Accuracy: Accuracy is a measure of how frequently a model predicts the correct result based on the input. However, it does not provide specific information on FP and FN. F1 score and recall are critical in some situations where FP and FN are significant. The formula in equation 5 is used to calculate accuracy.

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{True Positive} + \text{True Negative} + \text{False Positive} + \text{False Negative}}$$

(30)

Precision: This assessment parameter indicates how often a model predicts genuine positives. A low accuracy rating implies a large number of false positives. Equation 6 presents a formula for calculating precision.

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \qquad (31)$$

Recall: This parameter gives information regarding how often a model predicts false negatives. The low recall value indicates that the model predicted a high number of false negatives. Equation 7 gives a formula for calculating recall.

$$\text{Recall} = \frac{\text{TP+TN}}{\text{TP+FN}} \qquad (32)$$

F1 Score: Precision and recall are combined to calculate the F1 score. That is, a high F1 score suggests a low number of false positives and false negatives, implying that the model detects true elements accurately and is unaffected by false alarms. Equation 9 shows the formula for determining the F1 score.

$$\text{F1 score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Presiion+Recall}} \qquad (33)$$

Table1 proposed model performance for dataset1

Model	Accuracy (%)	Sensitivity (%)	Specificity (%)	Recall (%)	F1-Score (%)
MobileNet	97.12	97.28	97.05	97.28	97.22
Inception	97.68	97.81	97.54	97.81	97.73
VGG16	98.05	98.12	97.96	98.12	98.08
VGG19	98.37	98.45	98.29	98.45	98.41
DenseNet	98.72	98.84	98.63	98.84	98.78
GoogLeNet	98.96	99.04	98.87	99.04	98.98
AlexNet	98.45	98.53	98.36	98.53	98.48
ResNet34	99.21	99.34	99.12	99.34	99.28
ResNet50	99.63	99.58	99.67	99.58	99.60

Table 1 presents the performance comparison of various deep learning models—MobileNet, Inception, VGG16, VGG19, DenseNet, GoogLeNet, AlexNet, ResNet34, and ResNet50—on Dataset 1. The evaluation metrics include Accuracy, Sensitivity, Specificity, Recall, and F1-Score. Among the compared architectures, ResNet50 achieved the highest performance with an accuracy of 99.63%, demonstrating superior feature extraction and classification capability. ResNet34 also showed strong results with 99.21% accuracy, confirming the robustness of residual learning in handling complex IoT intrusion detection data.

Table 2 proposed model performance for dataset2

Model	Accuracy (%)	Sensitivity (%)	Specificity (%)	Recall (%)	F1-Score (%)
MobileNet	97.12	97.28	97.05	97.28	97.22
Inception	97.68	97.81	97.54	97.81	97.73
VGG16	98.05	98.12	97.96	98.12	98.08

VGG19	98.37	98.45	98.29	98.45	98.41
DenseNet	98.72	98.84	98.63	98.84	98.78
GoogLeNet	98.96	99.04	98.87	99.04	98.98
AlexNet	98.45	98.53	98.36	98.53	98.48
ResNet34	99.21	99.34	99.12	99.34	99.28
ResNet50	99.63	99.58	99.67	99.58	99.60
Hybrid ResNet34	99.74	99.81	99.68	99.81	99.77

Table 2 compares the same deep learning models as in Table 1 on Dataset 2, with the inclusion of a Hybrid ResNet34 model that integrates feature fusion and optimization techniques. The performance is measured across Accuracy, Sensitivity, Specificity, Recall, and F1-Score metrics. The Hybrid ResNet34 achieved the best overall results, with an impressive 99.74% accuracy and 99.81% sensitivity, surpassing all other models including ResNet50. This improvement highlights the effectiveness of the hybridization approach in enhancing deep feature representation and improving classification precision for IoT-based intrusion detection.

V Conclusion

The proposed hybrid deep learning–based Intrusion Detection System (IDS) demonstrates a highly effective classification framework for detecting and distinguishing between normal and anomalous network behaviors in IoT environments. By integrating comprehensive preprocessing techniques such as mean imputation, label encoding, and Z-score normalization with dimensionality reduction through Principal Component Analysis (PCA), the system ensures high-quality, noise-free, and informative input features for model training.Among the evaluated models, ResNet34 and ResNet50 exhibited superior performance due to their residual learning capabilities, effectively addressing vanishing gradient issues and enhancing feature propagation. The fusion of these two models into a Hybrid ResNet34–ResNet50 architecture, complemented by an attention mechanism, further improved classification accuracy and robustness. This hybrid approach successfully leveraged the complementary strengths of both architectures—ResNet34's faster convergence and ResNet50's deeper feature representation—to achieve the highest detection accuracy of 99.74%, outperforming individual models.Experimental results across multiple datasets confirmed that the proposed system achieved consistent improvements in accuracy, sensitivity, specificity, recall, and F1-score, establishing its superiority in identifying both known and unknown attacks. The comprehensive evaluation metrics validated the model's reliability and adaptability to real-world IoT scenarios.the proposed hybrid IDS not only enhances the detection capability and resilience of IoT networks but also provides a scalable and intelligent framework suitable for deployment in large-scale, heterogeneous IoT infrastructures. Future work may explore the integration of lightweight architectures and edge-based deployment strategies to further reduce computational overhead while maintaining high detection precision.

References

[1] Qureshi, K.N.; Jeon, G.; Piccialli, F. Anomaly detection and trust authority in artificial intelligence and cloud computing. *Comput. Netw.* **2021**, *18*, 107647. [**Google Scholar**] [**CrossRef**]

[2] Gerodimos, A.; Maglaras, L.; Ferrag, M.A.; Ayres, N.; Kantzavelou, I. IoT: Communication protocols and security threats. *Internet Things Cyber-Phys. Syst.* **2023**, *3*, 1–13. [**Google Scholar**] [**CrossRef**]

[3] Saurabh, K.; Sood, S.; Kumar, A.P.; Singh, U.; Vyas, R.; Vyas, O.P.; Khondoker, R. LBDMIDS: LSTM based deep learning model for intrusion detection systems for IOT networks. In Proceedings of the 2022 IEEE

World AI IoT Congress (AIIoT), Seattle, WA, USA, 6–9 June 2022; pp. 753–759. [**Google Scholar**]

[4] Henry, A.; Gautam, S.; Khanna, S.; Rabie, K.; Shongwe, T.; Bhattacharya, P.; Sharma, B.; Chowdhury, S. Composition of Hybrid Deep Learning Model and Feature Optimization for Intrusion Detection System. *Sensors* **2023**, *23*, 890. [**Google Scholar**] [**CrossRef**]

[5] Fitni, Q.R.S.; Ramli, K. Implementation of ensemble learning and feature selection for performance improvements in anomaly-based intrusion detection systems. In Proceedings of the 2020 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT), Bali, Indonesia, 7–8 July 2020; pp. 118–124. [**Google Scholar**]

[6] Bisyron, W.; Kalamullah, R.; Hendri, M. Implementation and Analysis of Combined Machine Learning Method for Intrusion Detection System. *Int. J. Commun. Netw. Inf. Secur.* **2018**, *10*, 295–304. [**Google Scholar**]

[7] Shakeela, S.; Shankar, N.S.; Reddy, P.M.; Tulasi, T.K.; Koneru, M.M. Optimal Ensemble Learning Based on Distinctive Feature Selection by Univariate ANOVA-F Statistics for IDS. *Int. J. Electron. Telecommun.* **2021**, *67*, 267–275. [**Google Scholar**] [**CrossRef**]

[8] Musthafa, M.B.; Ali, M.A.; Huda, S.; Kodera, Y.; Kusaka, T.; Nogami, Y. Evaluation of machine learning based optimized feature selection approach and classification methods for Intrusion Detection System. In Proceedings of the 2023 IEEE International Conference on Consumer Electronics, Pingtung, Taiwan, 17–19 July 2023; pp. 285–286. [**Google Scholar**]

[9] Shah, S.A.R.; Issac, B. Performance comparison of intrusion detection systems and application of machine learning to Snort system. *Future Gener. Comput. Syst.* **2018**, *80*, 157–170. [**Google Scholar**] [**CrossRef**]

[10] Zwane, S.; Tarwireyi, P.; Adigun, M. Ensemble learning approach for flow-based intrusion detection system. In Proceedings of the 2019 IEEE AFRICON, Accra, Ghana, 25–27 September 2019; pp. 1–8. [**Google Scholar**]

[11] Musthafa, M.B.; Huda, S.; Ali, M.A.; Kodera, Y.; Nogami, Y. Evaluation of IDS model by improving accuracy and reducing overfitting using stacking LSTM. In Proceedings of the 2024 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 5–8 January 2024; pp. 1–5. [**Google Scholar**]

[12] Berman, D.S.; Buczak, A.L.; Chavis, J.S.; Corbett, C.L. A survey of deep learning methods for cyber security. *Information* **2019**, *10*, 122. [**Google Scholar**] [**CrossRef**]

[13] Tao, P.; Sun, Z.; Sun, Z. An Improved Intrusion Detection Algorithm Based on GA and SVM. *IEEE Access* **2018**, *6*, 13624–13631. [**Google Scholar**] [**CrossRef**]

[14] Jaw, E.; Wang, X. Feature Selection and Ensemble-Based Intrusion Detection System: An Efficient and Comprehensive Approach. *Symmetry* **2021**, *13*, 1764. [**Google Scholar**] [**CrossRef**]

[15] Alshamy, R.; Ghurab, M. A review of big data in network intrusion detection system: Challenges, approaches, datasets, and tools. *J. Comput. Sci. Eng.* **2020**, *8*, 62–74. [**Google Scholar**]

[16] Chalapathy, R.; Chawla, S. Deep Learning for Anomaly Detection: A Survey. *arXiv* **2019**. [**Google Scholar**] [**CrossRef**]

[17] Zhou, Y.; Cheng, G.; Jiang, S.; Dai, M. Building an efficient intrusion detection system based on feature selection and ensemble classifier. *Comput. Netw.* **2020**, *174*, 107247. [**Google Scholar**] [**CrossRef**]

[18] Moustafa, N.; Slay, J. UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In Proceedings of the 2015 Military Communications and Information Systems Conference (MilCIS), Canberra, Australia, 10–12 November 2015; pp. 1–6. [**Google Scholar**]

[19] Moustafa, N.; Turnbull, B.; Choo, K.K.R. An Ensemble Intrusion Detection Technique Based on Proposed Statistical Flow Features for Protecting Network Traffic of Internet of Things. *IEEE Internet Things J.* **2019**, *6*, 4815–4830. [**Google Scholar**] [**CrossRef**]

[20] Ahsan, M.; Gomes, R.; Chowdhury, M.M.; Nygard, K.E. Enhancing Machine Learning Prediction in Cybersecurity Using Dynamic Feature Selector. *J. Cybersecur. Priv.* **2021**, *1*, 199–218. [**Google Scholar**] [**CrossRef**]

[21] Elmasry, W.; Akbulut, A.; Zaim, A.H. Empirical study on multiclass classification-based network intrusion detection. *Comput. Intell.* **2019**, *35*, 919–954. [**Google Scholar**] [**CrossRef**]

[22] Siraj, M.J.; Ahmad, T.; Ijtihadie, R.M. Analyzing ANOVA F-test and Sequential Feature Selection for Intrusion Detection Systems. *Int. J. Adv. Soft Comput. Its Appl.* **2022**, *14*, 185–194. [**Google Scholar**] [**CrossRef**]

[23] Kasongo, S.M.; Sun, Y. Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset. *J. Big Data* **2020**, *7*, 1–20. [**Google Scholar**] [**CrossRef**]

[24] Sinha, J.; Manollas, M. Efficient deep CNN-BiLSTM model for network intrusion detection. In Proceedings of the 2020 3rd International Conference on Artificial Intelligence and Pattern Recognition, New York, NY, USA, 26–28 June 2020; pp. 223–231. [**Google Scholar**]

[25] Liu, X.; Li, T.; Zhang, R.; Wu, D.; Liu, Y.; Yang, Z. A GAN and Feature Selection-Based Oversampling Technique for Intrusion Detection. *Secur. Commun. Netw.* **2021**, *2021*, 9947059. [**Google Scholar**] [**CrossRef**]

[26] Vaiyapuri, T.; Binbusayyis, A. Application of deep autoencoder as a one-class classifier for unsupervised network intrusion detection: A comparative evaluation. *PeerJ Comput. Sci.* **2020**, *6*, e327. [**Google Scholar**] [**CrossRef**]

[27] ALFRHAN, A.A.; ALHUSAIN, R.H.; Khan, R.U. SMOTE: Class imbalance problem in intrusion detection system. In Proceedings of the 2020 International Conference on Computing and Information Technology (ICCIT-1441), Tabuk, Saudi Arabia, 9–10 September 2020; pp. 1–5. [**Google Scholar**]

[28] Van Der Maaten, L.; Postma, E.O.; van den Herik, H.J. Dimensionality Reduction: A Comparative Review. *J. Mach. Learn. Res.* **2009**, *10*, 66–71. [**Google Scholar**]