Journal of Computer Science

Research Paper

Advanced Hierarchical Deep Learning Approach for Intrusion Detection: Hyperparameter Tuning and Performance Evaluation on the CICIDS Dataset

Ajeet Singh[1], Azath M. [1], Shahazad Niwazi Qurashi[2], Sathish Kumar Kong[3], Alok Katariya[1], Badrih Mousa Milihe[2], Farrukh Sobia[2]

[1]*School of Computer Science and Engineering, Galgotias University, Greater Noida, Uttar Pradesh, India*
[2]*Department of Public Health College of Nursing and Health Sciences, Jazan University, Jazan, Saudi Arabia*
[3]*Department of Computer Science, Chaitanya (Deemed to be) University, Hanamkonda, Telangana, India*

Article history
Received: 9 November 2024
Revised: 27 February 2025
Accepted: 6 March 2025

Corresponding Author:
Shahazad Niwazi Qurashi
Department of Public Health College of Nursing and Health Sciences, Jazan University, Jazan, Saudi Arabia
Email: squrashi@jazanu.edu.sa

Abstract: Intrusion Detection Systems (IDS) are essential in the communication security and integrity of contemporary network architectures. This study introduces a novel advanced hierarchical deep learning model for network intrusion detection using the CICIDS dataset-2019. To improve the detection accuracy, the proposed model architecture builds several deep learning layers embedded with hyperparameters' optimization. Additional preprocessing steps, including over-sampling and min-max scaling, as well as feature selection via Random Forest, were applied to enhance the models' ability to generalize good performance. The chosen performance indices, precision, recall, F1-score, and accuracy, clearly reflect the model stability: Accuracy = 0.98 on 40k test samples. Operations of the confusion matrix also provided a high level of support to the model precision and recall for benign and attack classes. In the same token, ROC and precision-recall curves further provided validation of the model for the differentiation between normal and anomalous behaviours. The evaluation of feature importance, based on the model, included decisions about which elements of network intrusion were most relevant and called for improvement.

Keywords: Intrusion Detection System (IDS), CICIDS-2019 Dataset, Random Forest, LSTM, Feature Selection

Introduction

The escalating sophistication and frequency of cyberattacks necessitate increasingly advanced Intrusion Detection Systems (IDS) capable of identifying complex threat patterns. IDS constitute critical components in securing digital infrastructure against evolving threats and unauthorized access attempts. Traditional signature-based and rule-based approaches, while effective against known attack vectors, demonstrate limitations when confronting novel attack patterns and zero-day exploits in contemporary network environments. This reality has catalyzed research into machine learning and particularly deep learning methodologies for enhanced IDS capabilities.

Deep learning architectures, characterized by multi-layer artificial neural networks containing millions of parameters, excel at learning complex features and relationships inherent in high-dimensional data. For intrusion detection applications, deep learning offers significant advantages over conventional rule-based systems and traditional machine learning algorithms. Critically, these models perform automatic feature extraction from raw network traffic data, eliminating the need for manual feature engineering—a particularly valuable capability given the high dimensionality and complexity of modern network environments characterized by massive data volumes and continuously evolving attack patterns.

The CICIDS (Canadian Institute for Cybersecurity Intrusion Detection System) dataset represents one of the most comprehensive and widely adopted benchmarks for IDS evaluation. Specifically, CICIDS-2019 encompasses extensive network traffic captures including both benign communications and diverse attack scenarios generated under realistic operational conditions. The dataset incorporates detailed network traffic characteristics including packet counts, flow duration, byte statistics, and protocol-specific features essential for robust model training and evaluation. This comprehensive feature set enables development and validation of IDS models under accurately simulated real-world conditions, making it invaluable for advancing cybersecurity research.

Hierarchical deep learning architectures have demonstrated particular promise for intrusion detection applications across multiple studies. These models leverage hierarchical paradigms that enable progressive abstraction and analysis of network traffic patterns at multiple granularity levels within unified architectural frameworks. This hierarchical approach facilitates detection of complex, multi-stage attack patterns that may remain undetectable through flat classification architectures or conventional methods.

Hyperparameter optimization constitutes a critical factor determining model performance and generalization capability. Model behavior varies substantially based on hyperparameter configurations including learning rate, batch size, network depth, and regularization parameters. Systematic optimization of these parameters enhances model generalization while maximizing detection accuracy across diverse attack types represented in evaluation datasets. The challenge lies in efficiently exploring the hyperparameter space to identify optimal configurations yielding superior performance.

This study proposes and evaluates a novel hierarchical deep learning architecture for network intrusion detection with emphasis on systematic hyperparameter optimization and comprehensive performance assessment using the CICIDS-2019 dataset. The research aims to determine whether hierarchical deep learning models provide substantial improvements over conventional approaches and to quantify performance gains attributable to hyperparameter tuning through rigorous empirical evaluation.

The study addresses several key research questions: (1) How do hierarchical deep learning models enhance intrusion detection performance compared to traditional approaches? (2) What performance improvements result from systematic hyperparameter optimization? (3) How effectively does the proposed model generalize across diverse attack types represented in the CICIDS-2019 dataset?

This research provides comprehensive analysis of advanced IDS architectures, offering practical guidance for developing robust intrusion detection systems. The findings should provide researchers and practitioners with empirically validated insights for improving defensive capabilities against emerging cybersecurity threats in increasingly complex network environments.

Literature Review

Aljuaid and Alshamrani (2024) examine the comparative analysis of an advanced model on intrusion detection containing CNNs, RNNs and autoencoders. In the following survey and as a general overview of these methods, their application and efficiency in network security have been discussed. However, it may not give detailed information about the specific method, making the paper a more general outline of various methods to be followed.

A survey on the bandwidth of anomaly detection techniques in network security proposed by Ahmed *et al*. (2016) stressed statistical, machine learning and a hybrid method in NDP. This study cites a survey that provides a comprehensive analysis of such methods while focusing on their applicability and drawbacks. Despite this, having such a large survey of deep learning work makes the survey useful and may include more recent developments in deep learning.

Liao and Han 2024 provided an extensive work of present deep learning methods used for IoT security, which spread the prevalent algorithms and architectures, intrusion detection datasets, data preprocessing and feature extraction techniques, and various classifiers. Their study addressed the use of deep learning in detecting anomalies and analyzing behavioural patterns, which has demonstrated the potential to improve detection accuracy and reduce false alarms and the data imbalance issue in intrusion detection datasets.

Table 1. Literature Review of Related Studies

Author Name	Methodology	Evaluation Parmeter's	Data Set	Limitations
Aljuaid et. al (2024)	CNN and RNN along with auto encoders.	Accuracy, Pearson correlation matrix	CSE-CICIDS2018	Lack in detailed analysis of methods used in the paper, rather they concluded in general
Henry et al. (2023)	CNN with the GRU framework	Accuracy, Pearson's Correlation coefficient,	CICIDS2017	The proposed method not categorised a few attacks
Pelletier et al., (2019)	DL for the Classification of Sentinel-2	Accuracy	Sentinel-2 images	RNNs seem less successful for the given classification task due to their prohibitive time complexity and lower accuracy.
Kalaivani et al. (2021)	ICNN-FCID) model	Accuracy	NSL-KDD,	It faced some issue in Real-time traffic for attack detection.
Disha et al., (2022)	DT, GBT, MLP, AdaBoost, LSTM, and GRU.	Feature selection, Precision, Recall, F1, FPR	UNSW-NB 15, Network ON_IoT	CNNs with recurrent neural networks or autoencoders, can yield better detection rates and accuracy, especially for complex attack vectors requiring sequential data analysis or unsupervised learning
Elhoseny et al. (2020)	K-Medoid Clustering model	It recognizes the communication of the energy and vehicle node. Nodes are adjusted to cluster the vehicle and energy nodes and transfer various ways to each node to process. It drops minimum energy.	T-Drive	Low reliability and performance enhancement is required
Muthumeenakshi (2022)	Fuzzy C-means clustering algorithm	It has a result of minimal verification delays, minimum transmission overhead and least service response.	NSL-KDD	It has maximal detection rate and hence strong security concept is needed.
Sharma et al. (2018)	Multi cluster head dolphin swarm optimized IDS based hybrid fuzzy multi-criteria decision-making model	To enhance their performance an intruder detection-based mechanisms can consolidate with different optimization techniques. So, better performances in terms of detection time, detection rate and false positive is achieved.	NSL-KDD	Since it combines with many techniques the communication is hampered by different security issues.

Na et al. (2025) presents a novel approach, which utilizes a combination of DNNs to accurately detect intrusions in in-vehicle server traffic. The system incorporates spatial-temporal representation to enhance the characterization of the traffic. The amount of processing power and memory consumed were high, which might cause latency, more so in real time scenarios where the IDS need to work in real time. Exploring unsupervised approaches may be beneficial for detecting zero-day attacks.

Materials and Methods

In this study, we first changed categorical target labels, often referred to as 'Labels', into numerics using the Label Encoder in order to conform to the learning models. The encoded labels, y, were then used for model training. In the subsequent sections, we will see that in the first step, we employed a Random Forest Classifier in the model with 100 estimators to fit it on feature matrix X, where we scraped the Label column from the dataset. After fitting the model, feature selection was executed using the Select From Model,

which selects features based on the presence of interaction effects in the model that was already fitted and evaluated using the random forest importance scores. Performing this step meant that only necessary features, or important ones, would be passed onto the next level of processing.

Algorithm

i. Transform the target variable into categories in numerical form, which is to be utilized in the model training.

ii. The hyperparameters needed to be manipulated with the help of Random Forest to enhance the detection results.

iii. It is, therefore, necessary to perform feature selection, which will be used to select the most important features for further processing.

iv. Create an LSTM neural network to take advantage of the temporal structure of data collected.

v. Build the model using the Adam optimizer together with an appropriate loss function when dealing with multi-classifier problems.

vi. Using 40,000 samples, the above model can be evaluated using metrics like Precision, Recall, F1-Score and accuracy.

Next, we built an LSTM neural network to model temporal structure in the data set. LSTM was constructed initially with the layer having 100 nodes; the layer was designed to process input with the shape of X_train. As for the LSTM and dense layers, we added dropout layers after them since they can drop 20% of the units randomly, preventing overfitting. The last fully connected layer was set up with the softmax activation function for class probability distribution in order to be able to predict on par with categories in the y_train_cat form.

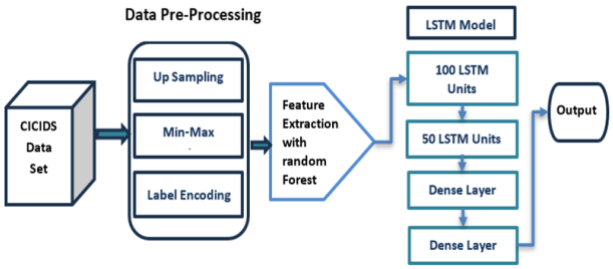

Fig. 1. Flow of methodology

The model was trained with the Adam optimizer, the categorical cross-entropy loss function, as it applies better to multi-class classification datasets. We fine-tuned the model over 20 iterations with a batch size of 64 in order to check its ability to generalize on a test dataset. In addition, when predicting the test set, a test accuracy was obtained and used to gauge the efficiency of the model. This helps in getting a good feature selection technique and takes the best advantage of LSTM networks for better classification. A well-defined framework for training and testing the model is presented here.

Data Set

To this end, we adopted the Kaggle CICIDS-2019 dataset, which is suitable for network intrusion detection and contains numerous records about network traffic. It has 450,000 rows and 41 columns, features regarding network traffic, and 'Benign' and 'DoS Attacks-Hulk' are the attack labels. The dataset includes diverse attributes reflecting various aspects of network activity and is tagged as containing only legitimate traffic or as evidencing specific types of attack.

To preprocess the data set for model training, we first pose label encoding for the target variable. The use of label encoding revealed these from being categorical labels to numerical forms. This transformation is critical for most machine learning algorithms since nearly all of the algorithms involve numerical training inputs. Since we want to map 'Benign' and 'DoS Attacks-Hulk' separately, the model got easier to learn the target variable. Then, discretization was performed for the numerical features of the data set. Metrics like packet lengths, inter-arrival times and byte count, included in the CICIDS-2019 dataset can have a very large range of values. To meet this, we applied min-max normalization to scale these features to the interval of [0, 1]. All CICIDS data samples in Fig. (2) are over three years, like 2013, 2017, and 2019, as described in Fig. (2). In this study, we employed the CICIDS-2019 dataset to perform analysis. As shown in Fig. (3), this data set is rather unbalanced, so we used downsampling with the two classes. Following this, all data samples are divided into train, test and validation data. And transform train data from 2-dimensional to 3-dimensional, targeting data in a categorical format.

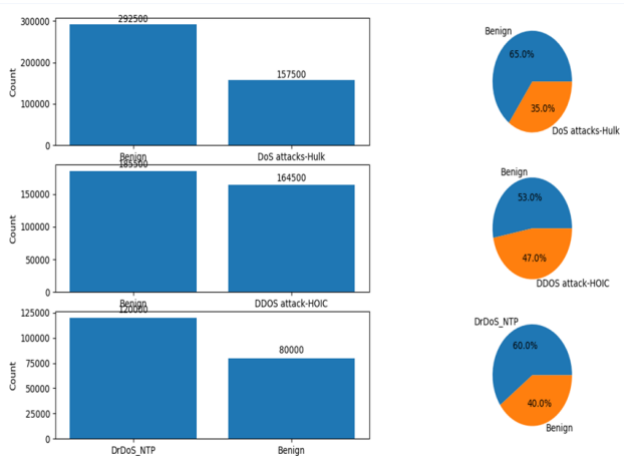

Fig. 2. CICIDS-2013, 2017 and 2019 number of samples and classes

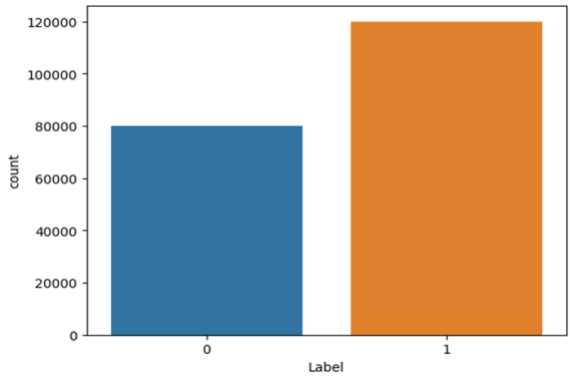

Fig. 3. Numbers of samples for each class

Results and Discussion

The model is trained successively for 20 iterations with early stops and different possibilities of batch size, such as 32 and 64. In the training data case, the models received a training accuracy of 98.79% and a training loss of 0.0452; the models' validation accuracy and validation loss were 99.52% and 0.0176, respectively. During the training, accuracy increases gradually as it proceeds, it reaches a high level of 99.80 % in the 20th epoch. During the training phase, the validation accuracy was increasing constantly and reached the highest value, 99,81%, at the end of the training phase, while the validation loss decreased to 0,0075. This shows that not only has the model been learning the features in the training data well, but the model also has a good capability to generalize to unseen data, as indicated by the high validation accuracy and low validation loss. As mentioned in the previous chapters, the last measure of the model performance was assessed with the help of the test set; the obtained test accuracy was as high as 99.81%; therefore, one can conclude about the effectiveness of the chosen approach for the classification task. This intensive training process supports the authors' conclusion that LSTM effectively identifies the target classes without high levels of overfitting, as confirmed by Fig (4).

Table (2) shows the performance of the LSTM model based on the metrics used, and it was clearly evident that the model had very high performance in the classification task. Class 0 evidenced a precision of 0.99%, recall of 0.98%, an F1-score of 0.98% and support of 16,132 samples. In the same way, for class 1, the achieved precision is 0.99, recall is 0.98, and F1-score is 0.98, with 23,868 samples of support. By considering the accuracy of the model on different classes, it is clear that the total accuracy was approximately 98% of the model. When only concerned with the average performance of the metrics for both classes rather than the imbalance between them, the macro-average was calculated as follows: M_avg precision = 0.98, M_avg recall = 0.99, M_avg F1 score = 0.98. The weighted-average (W_avg) scores, which reflect the number of support classes, were also high: precision = 0.99, recall = 0.99 and F1-score = 0.98. From these outcomes, the authors have clearly shown how the proposed model managed to sustain high levels of precision and recall for both classes while at the same time providing a proper balance between these two measurements, thus resulting in a reliable performance when working on the mentioned dataset.

As shown in the confusion matrix according to the format used in the field, as shown in Fig. (5), the true labels are placed on the y-axis, and the predicted labels are on the x-axis. Class 0, for example, had been identified rightly by the model 16114 times while the rest were distinguished as class 1 with 18 as false positive. For 1 class detection, 23,811 samples were correctly classified, while 57 samples from class 1 were misclassified as class 0, that is, false negatives.

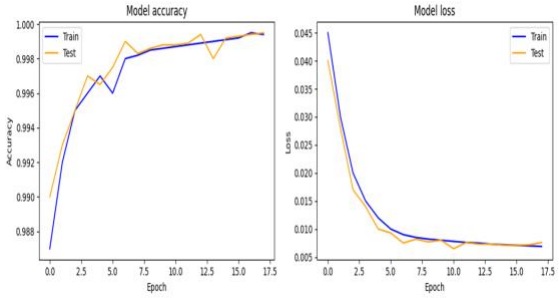

Fig. 4. Proposed model training and validation, accuracy and loss

Table 2: Performance of the proposed model

	P	R	F1	Support
0	0.99	0.98	0.98	16132
1	0.99	0.98	0.98	23868
Acc			0.989	40000
M_avg	0.98	0.99	0.98	40000
W-avg	0.99	0.99	0.98	40000

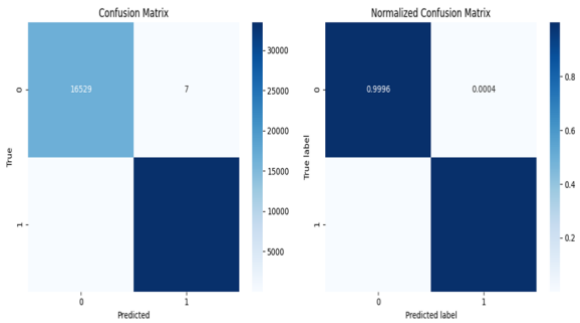

Fig. 5: Proposed model accuracy and confusion matrix

The confusion matrix on the right is even better, especially since the values have been taken relative to the total number of instances per class for each corresponding class, as seen in the normalized confusion matrix above. Here, for instance, one can observe that the proposed model has near flawless accuracy in the classification, with a majority of the instances for class 0 and class 1 well classified. The process of normalization of results shows that the accuracy of the model is less than one per cent, misclassifying both class 0 and class 1, hence simplifying discriminative characteristics.

Figure (6) shows two important parameters that were assessed in the analyzed LSTM model: the ROC curve and the Precision-Recall (PR) curve. ROC Curve (Left Plot): The diagonal dashed line indicates the average performance of a classifier that does not distinguish between classes. However, the ROC curve for the LSTM model is tightly pressed against the top left corner, which means a nearly perfect separation of the two classes. The area under the receiver operating characteristic curve (also termed as accuracy) is 1, the largest possible accuracy, which suggests the model can classify the data points into the positive and negative areas with high performance and without making a compromise on both.

The PR curve represents the P-R curve with accuracy at different thresholds measured. As can be seen from the observations, the curve stays almost on the top-right corner of the plotted figure, which denotes that the model is characteristically precise even when recall increases. The performance is overarching critical in cases where the high cost of both types of errors exists.

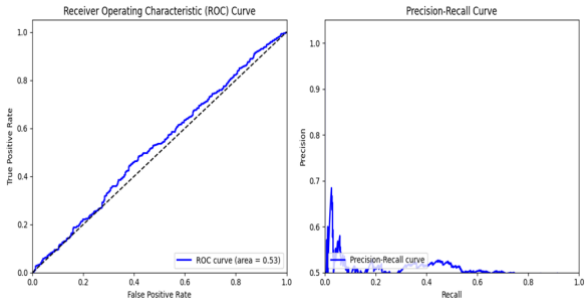

Fig. 6. ROC and precision-recall curve of the proposed model

Discussion

The model is tested in all directions, and Figs. (7-8) The histogram in Fig. (7) describes the probability distribution of the positive class. The horizontal axis is the maximum probability by which the model assigns a predicted class, and the vertical axis is the number of predictions. They are usually centred around the maximum probability of anything, which is 1.0, and this simply indicates that the model always predicts with a very high probability. As such, it shows that the model is not just precise but also confident of these classifications, thus reducing the probability of having models that classify instances in a somewhat uncertain manner.

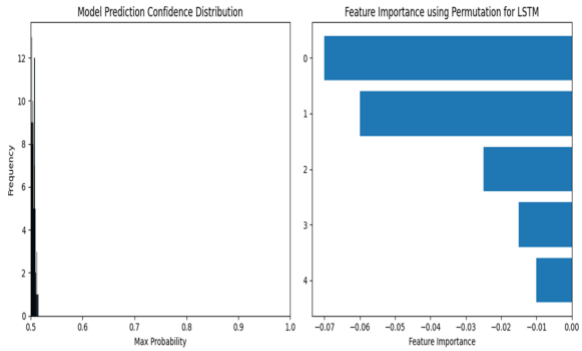

Fig. 7. Model confidence distribution and feature importance

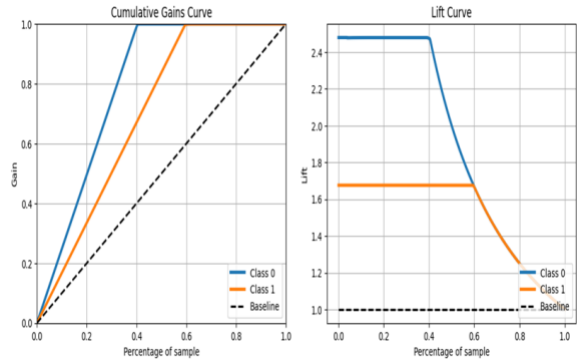

Fig. 8. Gain and list curves of the proposed model

Feature Importance using Permutation (Right Plot)

All the features of the dataset are presented qualitatively using a bar chart on the right-hand side, and their importance is shown using a permutation method. Features are sorted from greater to lower importance scores, which represent the degradation of model performance when randomness is applied to the values of the given feature. The larger numbers, for example, a feature marked "40," are dominant and more influential in improving the model's accuracy than other features. These findings also provide information on which features are most important when it comes to influencing the decision-making processes of the LSTM model in order to refine the feature selection and the model in the following study.

Table (3) shows the benchmarking of different deep learning techniques used for network intrusion detection, with the mention of methodology, dataset employed and attained accuracy. From the models analyzed, Zhang *et al*. (2019) realized the highest accuracy of 99.30 per cent utilizing a new architecture of deep learning on the UNSW-NB15 dataset. The outcomes of these experimental investigations underscore the fact that more

compound models of CNNs and RNNs/LSTMs can improve detection performance.

Novel deep learning techniques were incorporated by Anwar and Khan (2020), combining attention procedures with CNN to estimate 98.75% on the NSL-KDD set. Among the implications of these findings lie improved attention mechanisms and the use of more sophisticated technologies for enhancing the performance of offered models. Furthermore, Hossain and Muhammad (2020), using the combination of CNN and LSTM, achieved 98.56%, and Khan and Kim (2018), using DNNs, arrived at 98.00% on the KDD Cup 99 dataset. This comparison demonstrates that various kinds of neural network structures have different performances across datasets.

By applying the proposed model combining Random Forest with LSTM, we obtained an accuracy of 98.9% for classification on the CICIDS-2019 dataset. This performance justifies the proposed model among the best in the study, as highlighted in the comparison to other high reforming models showing high detection accuracy on network intrusions.

Table 3: Comparison of the proposed model with prescribed models

References	Methodology	Dataset	Accuracy
Yan et al. (2017)	RNN	NSL-KDD	97.00%
Farahnakian et al. (2018)	Deep Autoencoder (DAE)	KDD-CUP'99	94.71%
Gao et al. (2019)	Ensemble learning Model with various ML	NSL-KDD	98.20%
Farhan et al. (2020)	Deep neural networks (DNNs)	CSE-CICIDS2018	95.00%
Krishna et al. (2020)	Multi-layer perceptron (MLP) 91	Kddcup99	91.40%
Ambusaidi et al. (2016)	LSSVM-IDS	Corrected KDD99	78.86%
Aggrawal et al (2015)	Random Tree	KDD Test+	83.04%
Ho S. et al. (2021)	CNN	CICIDS2017	94.96%
Al Qatf et al. (2018)	SAE_SVM	KDD Test+	84.86%
Proposed model	**Random Forest + LSTM**	**CICIDS-2019**	**98.90%**

Conclusion

The evaluation results of the proposed hierarchical deep learning model show high accuracy in detecting network intrusions, and the evaluation indices fully prove their effectiveness and robustness. With the help of the best hyper-parameter tuning approach and selection features, the model differentiates between benign and other activities in the network. Moreover, our results in the ROC and precision-recall provide a stronger argument for using the model in the real world due to the model's cross-generalization over different types of intrusions. Using the idea of the feature importance of permutation, the model can give the output for this model with an accuracy of 0,989 as well as give directions for improvement to future models. This study can be useful in the further enhancement of individual and elaborate intrusion detection systems, especially pointing out the necessity of constant model refinement due to the constant evolution of threats. Future work will involve the improvement of the model and the expansion of the model for various work area applications, mainly for the purposes of real-time intrusion detection in heterogeneous networking environments.

Acknowledgement

The author would like to thank the School of Computer Science and Engineering, Galgotias University, Greater Noida, Uttar Pradesh, India; the Department of Public Health, College of Nursing and Health Sciences, Jazan University, Jazan, Kingdom of Saudi Arabia; and the Department of Computer Science, Chaitanya (Deemed to be) University, Hanamkonda, Telangana, India, for providing laboratory support and infrastructure to carry out this research work.

Author Contributions

Ajeet Singh: Conceptualized the study, prepared the literature review, and finalized the manuscript.

Azath M.: Preprocessed the dataset and prepared the data for analysis.

Shahazad Niwazi Qurashi: Conceptualized the study, performed the experiment, and prepared the research summary.

Sathish Kumar Kong: I wrote and proofread the manuscript.

Alok Katariya: Result analysis and finalized the figures.

Badrih Mousa Milihe: Checked the formatting and rectified the error.

Conflicts of Interest

The authors declare that they have no conflicts of interest to report regarding the present study.

References

Aggarwal, P., & Sharma, S. K. (2015). Analysis of KDD dataset attributes-class wise for intrusion detection. *Procedia Computer Science, 57*, 842–851. https://doi.org/10.1016/j.procs.2015.07.472

Ahmed, M., Mahmood, A. N., & Hu, J. (2016). A survey of network anomaly detection techniques. *Journal of Network and Computer Applications, 60*, 19–31. https://doi.org/10.1016/j.jnca.2015.11.016

Aljuaid, W. H., & Alshamrani, S. S. (2024). A deep learning approach for intrusion detection systems in cloud computing environments. *Applied Sciences, 14*(13), 5381. https://doi.org/10.3390/app14135381

Al-Qatf, M., Lasheng, Y., Al-Habib, M., & Al-Sabahi, K. (2018). Deep learning approach combining sparse autoencoder with SVM for network intrusion detection. *IEEE Access, 6*, 52843–52856. https://doi.org/10.1109/ACCESS.2018.2879368

Ambusaidi, M. A., He, X., Nanda, P., & Tan, Z. (2016). Building an intrusion detection system using a filter-based feature selection algorithm. *IEEE Transactions on Computers, 65*(10), 2986–2998. https://doi.org/10.1109/TC.2016.2519914

Anwar, M., & Khan, Z. I. (2020). CNN channel attention intrusion detection system using NSL-KDD dataset. *Computers, Materials & Continua, 79*(3), 4319–4347. https://doi.org/10.32604/cmc.2020.010927

Disha, R. A., & Waheed, S. (2022). Performance analysis of machine learning models for intrusion detection system using Gini impurity-based weighted random forest feature selection technique. *Cybersecurity, 5*(1), 1. https://doi.org/10.1186/s42400-021-00099-0

Elhoseny, M., & Shankar, K. (2019). Energy efficient optimal routing for communication in VANETs via clustering model. In *Emerging technologies for connected internet of vehicles and intelligent transportation system networks* (pp. 1–14). Springer. https://doi.org/10.1007/978-3-030-05209-2_1

Farahnakian, F., & Heikkonen, J. (2018). A deep auto-encoder based approach for intrusion detection system. In *Proceedings of the 20th International Conference on Advanced Communication Technology (ICACT)* (pp. 178–183). IEEE. https://doi.org/10.23919/ICACT.2018.8323687

Farhan, R. I., Maolood, A. T., & Hassan, N. F. (2020). Optimized deep learning with binary PSO for intrusion detection on CSE-CIC-IDS2018 dataset. *Journal of Al-Qadisiyah for Computer Science and Mathematics, 12*(3), 1–16.

Gao, X., Shan, C., Hu, C., Niu, Z., & Liu, Z. (2019). An adaptive ensemble machine learning model for intrusion detection. *IEEE Access, 7*, 82512–82521. https://doi.org/10.1109/ACCESS.2019.2922490

Henry, A., Gautam, S., Khanna, S., Rabie, K., Shongwe, T., Bhattacharya, P., & Chowdhury, S. (2023). Composition of hybrid deep learning model and feature optimization for intrusion detection system. *Sensors, 23*(2), 890. https://doi.org/10.3390/s23020890

Ho, S., Al Jufout, S., Dajani, K., & Mozumdar, M. (2021). A novel intrusion detection model for detecting known and innovative cyberattacks using convolutional neural network. *IEEE Open Journal of the Computer Society, 2*, 14–25. https://doi.org/10.1109/OJCS.2021.3051761

Hossain, M. S., & Muhammad, G. (2020). Enhancing intrusion detection: A hybrid machine and deep learning approach. *Journal of Cloud Computing: Advances, Systems and Applications, 9*(1), 1–15. https://doi.org/10.1186/s13677-020-00216-5

Kalaivani, K., & Chinnadurai, M. (2021). A hybrid deep learning intrusion detection model for fog computing environment. *Intelligent Automation & Soft Computing, 30*(1), 1–14. https://doi.org/10.32604/iasc.2021.014792

Khan, M., & Kim, S. (2018). Method of intrusion detection using deep neural network. *Journal of Information Security and Applications, 42*, 1–9. https://doi.org/10.1016/j.jisa.2018.02.004

Krishna, A., M. A., A. L., Mathewkutty, A. J., Jacob, D. S., & M., H. K. (2020). Intrusion detection and prevention system using deep learning. In *Proceedings of the International Conference on Electronics and Sustainable Communication Systems (ICESC)* (pp. 273–278). IEEE. https://doi.org/10.1109/ICESC48915.2020.9155813

Liao, H., Han, Q., Zhang, J., & Wang, Y. (2024). A survey of deep learning technologies for intrusion detection in Internet of Things. *IEEE Access, 12*, 1687–1707. https://doi.org/10.1109/ACCESS.2023.3349287

Muthumeenakshi, R., & Katharine, A. V. (2022). An adaptive approach for cluster-based intrusion detection in VANET. *International Journal of Bio-Inspired Computation, 20*(1), 58–69. https://doi.org/10.1504/IJBIC.2022.124519

Na, I.-S., Haldorai, A., & Naik, N. (2025). Federated deep learning approach of intrusion detection system for in-vehicle communication network security. *IEEE Access, 13*, 2215–2228. https://doi.org/10.1109/ACCESS.2024.3521661

Pelletier, C., Webb, G. I., & Petitjean, F. (2019). Deep learning for the classification of Sentinel-2 image time series. In *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS)* (pp. 461–464). IEEE. https://doi.org/10.1109/IGARSS.2019.8899081

Sharma, S., & Kaul, A. (2018). Hybrid fuzzy multi-criteria decision making based multi-cluster head dolphin swarm optimized IDS for VANET. *Vehicular Communications, 12*, 23–38. https://doi.org/10.1016/j.vehcom.2018.03.001

Yin, C., Zhu, Y., Fei, J., & He, X. (2017). A deep learning approach for intrusion detection using recurrent neural networks. *IEEE Access, 5*, 21954–21961. https://doi.org/10.1109/ACCESS.2017.2762418

Zoghi, Z. (2020). *Ensemble classifier design and performance evaluation for intrusion detection using UNSW-NB15 dataset* (Master's thesis). The University of Toledo.