ISSN: 2357- 0954
DOI:10.21608/EJMTC.2025.353334.1301

Original Article

Reinforcing cyber security : Defensive machine learning based intrusion detection system for NSL-KDD

Ahmed Hossam Eid[1], Ahmed M. Mattar[2], Mohamed Hussein[3]

[1]Department of Mathematics, [2]Department of Computers and Artificial Intelligence, [3]Communications Department, Milittary Technical College, Cairo, Egypt

Keywords:

Cyber attacks, cyber security, intrusion detection, machine learning, NSL-KDD, random forest.

Corresponding Author:

Ahmed M. Mattar, Department of Computers and Artificial Intelligence, Military Technical College, Cairo, Egypt, **Tel:** 01008988994, **E-mail:** a.mattar@ieee.org

Received : 15 January 2025

Accepted : 26 February 2025

Abstract

In the world of digital transformation, intrusion detection has proven to be valuable in protecting the assets of organizations. In this paper, we propose a new machine learning-based technique; Random Forest (RF) to be implemented as intrusion detection system (IDS), to act as defensive frontier for organizations. However, creating an efficient IDS faces a number of challenges, these challenges summarize in accuracy (mirrored as false positive rate) and training time. Choosing the right machine learning classifier, to work with the right type of network data is important. Detection accuracy can be enhanced by tuning the classifier towards optimal variables. While, training time can be enhanced by correct pre-processing of network data and selecting the features that are most dominant in correlation with the desired output. We examined several machine learning techniques, we applied several data pre-processing steps on NSL-KDD, also, hyper parameter tuning (manipulation) was performed to optimize classifier performance, finally, feature selection techniques were utilized to reduce training time and enhance overall performance. Random Forest has proven to be the most effective machine learning classifier to be used with NSL-KDD, we achieved the highest accuracy of 99.7% and training time of 30.25 second using only 7 features.

I. INTRODUCTION

The exponential evolution of technology has brought observable advancements in communication and digital transformation, non the less, it has also given opportunity for the rise of sophisticated cyber threats. Large companies understand the importance of cyber security, but small or startup companies does not realize that importance.

Organization growth is tightly coupled to more interconnectivity and more data, thus, the need for data protection increases. Intrusion Detection can play a vital role in preserving the assets of organizations by monitoring the events that are happening in pursuit of threats[1]. Intrusion detection systems (IDS) are the systems that automate the process of intrusion detection.

In the early days of cyber security, IDSs used threshold, or rule based formulas to detect cyber-attacks. This setting worked for a while, but the more hackers are smart and persistent, the more this setting proved to be in-effective.

Machine learning (ML) can enrich IDS by elevating its privileges to detect sophisticated or elusive attacks. Also, ML-based IDS has proven to be effective against new (zero-day) undiscovered attacks via smart anomaly detection.

NSL-KDD[2] a prominent cyber security dataset, is now considered the new corner stone in cyber security. Unlike its predecessor KDD CUP 99 that contained a lot of redundant data and imbalance, NSL-KDD still attracts the attention of researchers till today, due to its structure and format.

This research focuses on applying machine learning techniques to the NSL-KDD dataset to build a robust intrusion detection system. Our approach involves comprehensive data preprocessing steps, including sampling, parameter tuning, and feature selection, to ensure high-quality input for model training. By systematically training and testing the machine learning models, this work aims to enhance the accuracy of intrusion detection while minimizing false alarms. The ultimate goal is to contribute to the development of smarter and more adaptive IDS solutions capable of addressing emerging cybersecurity challenges.

The rest of the paper is organized as follow; section II represents research work in the field, section III explains the design aspects of the proposed research work. Section IV offers a thorough performance evaluation of proposed research, section V concludes the paper.

II. RELATED WORK

This section summarizes previous research done on NSL-KDD in cyber security[3] proposed classification using Extreme learning machine (ELM). The model was trained and tested on several ratios of NSL-KDD.

In[4], an anomaly IDS using machine learning and information gain-rank (IG-R) is proposed to improve the detection accuracy of intrusions on NSL-KDD. The network security lab-knowledge discovery dataset (NSL-KDD) is used to train and test the proposed IDS. The authors were able to reduce the number of features to only 6, achieving the highest accuracy against other individual ML classifiers.

The authors in[5] adopted Cross-Industry Standard Process for Data Mining (CRISP-DM) as the framework. The primary goal of this research is to conduct a comparative analysis of classification techniques to identify normal and anomaly records within network data.

The work in[6] utilized three machine learning classifiers; Support vector Machine (SVM), Naïve Bayes (NB) and K-Nearest Neighbor classifiers (KNN). These algorithms were tested using NSL KDD dataset by blending Chi-square and Extra Tree feature selection method against 10% and 20% data from the original distribution of the dataset, from which 80% were allocated for training and 20% for testing. The highest accuracy was achieved via KNN and 27 features.

Rani *et al.*[7] developed a Network Intrusion Detection System (NIDS) that addresses class imbalance issues by employing a Deep Neural Network (DNN) as the classifier. The preprocessing stage involves normalization and data transformation by adjusting the entropy function. Training includes grouping labels and assigning weights to further improve performance.

Mohamed *et al.*[8] proposed an anomaly detection framework based on the Deep SARSA technique to enhance accuracy in identifying anomalies in imbalanced datasets. The framework utilizes Deep Reinforcement Learning (DRL) as an anomaly detector, The blend of DRL, SARSA, and DNN demonstrated improved detection accuracy.

Cui *et al.*[9] introduced a novel multi-model integrated detection and prevention system called GMM-WGAN (Gaussian Mixture Model-Wasserstein Generative Adversarial Network). The system first utilizes a weighted autoencoder-based t select features. Next, an imbalance analysis module, GMM-WGAN, is developed by combining a clustering algorithm with the Wasserstein generative adversarial network (WGAN) model. Finally, a classification module is created using a mixture of Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks (CNN-LSTM).

Kasongo *et al.*[10] developed an Intrusion Detection System (IDS) framework using various machine learning methods, particularly focusing on recurrent neural networks (RNNs).

Liu *et al.*[11] proposed the Broad Learning System (BLS), a substitute to deep neural networks. The BLS is enhanced with LU decomposition.

Zhou *et al.*[12] introduced a more efficient hybrid Harris Hawk optimization technique to improve intrusion detection capabilities. This enhanced method is employed as a feature selection strategy , class imbalance is handled by using a combination of K-Nearest Neighbors (KNN) and deep denoising autoencoder (KNN-DDAE). Finally, a Deep Neural Network (DNN) is used for extensive classification tasks, enhancing detection accuracy.

Keshak *et al.*[13] proposed an LSTM-based model to develop an Intrusion Detection System (IDS) for detecting cyberattacks.

The authors in[14] performed similar work to this proposed research but they differ in the feature selection technique; they used Symmetrical uncertainty. Also, they performed several data preprocessing steps, different than what we used. As will be clear later, our results are better than theirs.

III. PROPOSED RESEARCH

This section highlights the design aspects of the proposed model. First, a brief explanation of the used dataset; NSL-KDD, followed by discussion of proposed model.

A. NSL-KDD DATASET

The experimental work performed in this paper use NSL-KDD as benchmark dataset for cyber security. The proposed model is trained and tested upon this dataset, rather than its predecessor KDD CUP 99. The dataset consist of two main files; KDDTrain+ (125,973 records) and KDDTest+ (22,544 records) with an overall traffic of 148,517 record. The dataset contains 77,052 normal traffic and 71464 attack traffic. The attack traffic resembles 38 cyber-attacks; these attacks are categorized into four main attack categories, Table I depicts NSL-KDD attack categories.

Table I: NSL-KDD Attack Types

Attack Category	Training Set	Testing Set	Total	Total
DoS	45,927	2,421	48,348	48,348
Probe	11,656	7,456	19,112	19,112
U2R	114	1,436	1,550	1,550
R2L	934	1,520	2,454	2,454
Normal	67,341	9,711	77,052	77,052
Total	125,972	22,544	148,517	148,517

B. Data Preprocessing

- This study focuses on developing an efficient intrusion detection model by careful data preparation, model training, and testing.
- We utilized many preprocessing methods to prepare the data. Initially, we merged the train and test data into one dataset and split the data into by ratio 80% for training and 20% for testing.
- Next, we applied sampling technique to the dataset in order to handle the imbalance.
- Afterwards, we conducted several parameter tweaking in order to optimize the model's working configuration and achieve optimal performance.
- We tried a number feature selection techniques to identify the most dominant features from the dataset, thus, enhancing the model's efficiency and training time.
- After preparing the data, we employed machine learning techniques to train the model for the classification of normal and suspicious/abnormal activities in network traffic.
- Ultimately, we evaluated the model to assess its accuracy, effectiveness in detecting cyber-attacks, and lower the frequency of false alarms.

This approach illustrates the importance of integrating effective reliable data preparation with intelligent model training to enhance intrusion detection.

IV. EXPERIMENTAL RESULTS

This section will cover the various aspects of evaluating the performance of proposed model.

A. Performance Metrics

Since the research here involves machine learning, it is only valid to utilize the widely used confusion matrix. Most of the academic research in machine learning uses accuracy, precision, recall, and F1-score, and the same will be used here.

Table II: Confusion matrix for evaluation of classifiers

		Prediction	
		Normal	Attack
Actual	Normal	TN	FP
	Attack	FN	TP

True Positive (TP) - The number of times that malicious instances were correctly found.

True Negative (TN) - The quantity of legitimate instances that accurately classified.

False Positive (FP) - The number of instances that are incorrectly identified as attacks however in fact they are legitimate activities.

False Negative (FN) - The number of instances that are incorrectly classified as legitimate activities however in fact they are malicious.

The proportion of correctly categorized instances to all instances is calculated by the equation's metrics of accuracy, as given in equation (1)[15,16].

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

“Recall,” or the True Positive Rate as delineated in equation (2), denotes the ratio of correctly detected positive occurrences to the total instances anticipated to be true positive[15,16].

$$Recall\ (True\ Positive\ Rate) = \frac{TP}{TP + FN} \quad (2)$$

“Precision”, defined by equation (3), denotes the proportion of accurately detected positive instances among the total predicted positive instances[15,16].

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

The "F1-score", presented in equation (4), furnishes the harmonic meaning of both recall and precision. This metric proves invaluable in addressing challenges related to specific/individual class[15,16].

$$F1 - Score = 2 \times (\frac{Precision \times Recall}{Precision + Recall}) \quad (4)$$

The “FPR” measures the probability of a system raising a false alert, as given in equation (5)[15,16].

$$FPR = \frac{FP}{FP + TN} \quad (5)$$

B. Testbed setup

As an initial investigation, we use the most commonly known machine learning techniques; Random Forest (RF), Decision Tree (DT), Bayes Networks (BN), K-Nearest Neighbor (KNN), and Logistic Regression (LogR). Careful consideration was made to choose machine learning techniques belonging to multiple categories (i.e., different data exploration behaviour).

NSL KDD standard dataset was used, it contains

41 features, and two files; one for training and the other for testing. These techniques tested on Intel(R) Core™ i7 –4510U CPU @ 2.00 GHZ, 8GB RAM and coding & analysis are done by Weka 3.8.6. The individual performance of these techniques is represented in Table III below.

Table III: Machine learning techniques performance

Tech.	Acc%	Pre%	Rec%	F1-Score%
RF	99.6	99.6	99.6	99.6
DT	99.49	99.5	99.5	99.5
BN	92.98	94.6	93	93.6
KNN	99.23	99.2	99.2	99.2
SVM	97.68	97.7	97.7	97.7
LogR	97.41	97.3	97.4	97.4

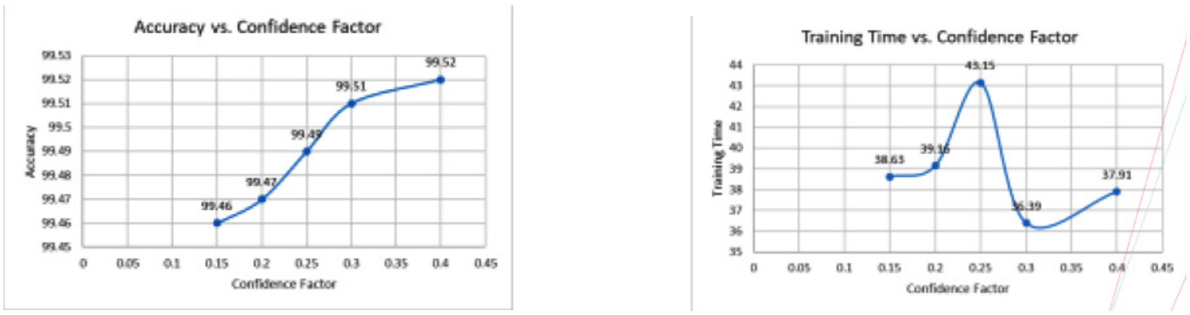
It is clear from the results that tree techniques; RF and DT, have the best performance on NSL-KDD among other techniques. Thus, further investigation will be performed on them. Table IV illustrates the performance of RF and DT after applying sampling technique on the NSL-KDD.

Table IV: ML techniques performance with sampling

Tech.	Acc%	Training Time (s)	Testing Time (s)
RF	99.8	90.81	1.81
DT	99.63	26.87	0.24

It is very clear from Table IV that performance of RF and DT is enhanced by nearly 0.2% and 0.163 respectively. This enhancement in performance is due to sampling. Also, the training time was radically decreased by ratio of 35.19% and 37.73% of RF and DT from their original performance without sampling (140.12s and 43.15s)

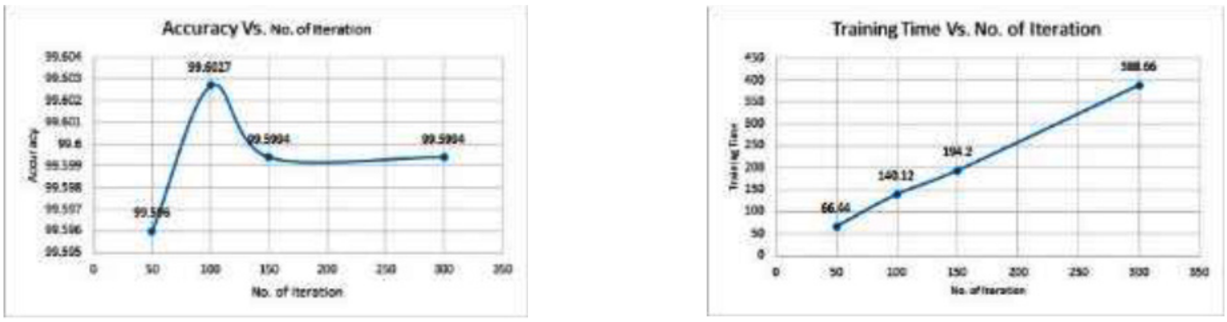
Reverting NSL-KDD to its original format to analyze the parameter tuning individually we get the finding in Fig. 1 and Fig. 2 for DT and RF respectively. Tuning the parameters resulted in changes regarding accuracy and training time.



(a) (b)

Fig. 1: Decision Tree Parameter Tuning

Regarding DT, the default value for the confidence factor parameter is 0.25, we started our series of experiments by the default value and branched in both directions, i.e., increasing and decreasing the factor, in order to observe the effect of confidence factor on technique accuracy. Thus, the findings were, increasing the confidence factor increased accuracy and decreased training time. While, decreasing it resulted in a decrease in both accuracy and training time.



(a) (b)

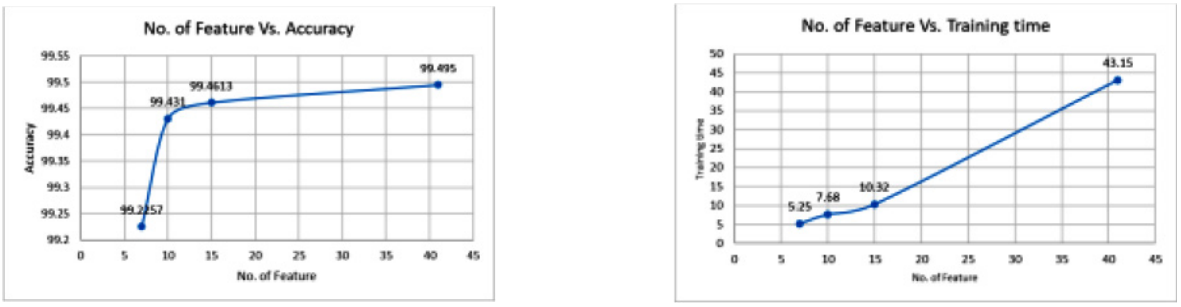
Fig. 2: Random forest parameter tuning

As for RF, the default value for iterations is 100, any change in it deteriorated the accuracy. Also, it is obvious that there is a tight correlation between the change of iterations and training time.

Table V: ML techniques performance with parameters tuning

Tech.	Acc%	Training Time (s)
RF	99.596	66.44
DT	99.52	37.19

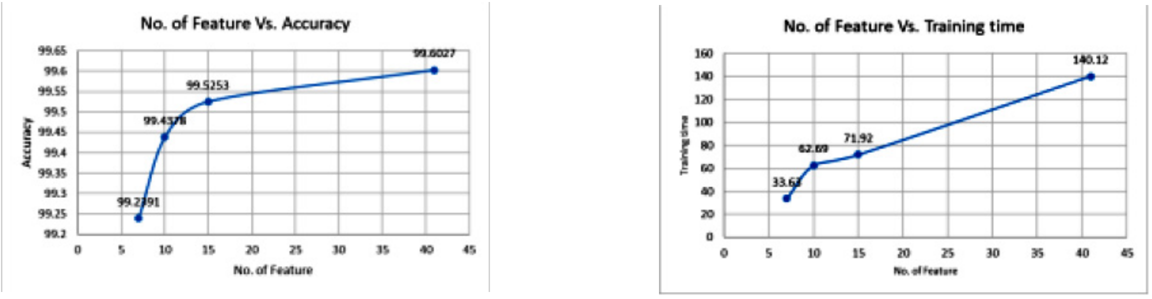
It is obvious from Table V that employing parameter tuning does affect the overall performance of DT for accuracy and training time, and it is also clear that the accuracy of RF is decreased by 0.004% which is very small value against a time enhancement with 52.58%.Reverting NSL-KDD to its original format to solely analyze the feature selection, we get the finding in Fig. 3 and Fig. 4 for DT and RF respectively. The selected features resulted in changes regarding accuracy and training time.



(a) (b)

Fig. 3: Decision Tree Feature selection

Regarding DT, the original number of feature of NSL-KDD is 41 feature, increasing the number of features for training, increases accuracy and increases training time. While, decreasing them results in a significant decrease in both accuracy and training time. The same analysis goes for RF as depicted in Fig. 4.



(a) (b)

Fig. 4: Random Forest Feature selection

Table VI: ML techniques performance with feature selection

Tech.	No. of features	Acc%	Training Time (s)
RF	41	99.6027	140.12
	7	99.239	33.63
DT	41	99.495	43.15
	7	99.225	5.25

It is obvious from Table V that employing feature selection affect the overall performance RF and DT, using ONLY 7 features reduced the accuracy by 0.3637% and 0.27% for RF and DT respectively. On the other hand, there was a significant reduction in training time by 76% and 84.39% for RF and DT.

Table VII: Overall ML techniques

Tech.	# of features	Pre-processing	Parameter Tuning	Feature Selectio	Acc%	Training Time (s)
RF	41	-	-	-	99.603	140.12
	7	resampling	√	Ranker	99.748	30.25
DT	41	-	-	-	99.495	43.15
	7	resampling	√	Ranker	99.626	7.73

Table VII represents the final findings after applying pre-processing, parameter tuning, and feature selection. It illustrates the position that integrating effective reliable data preparation with intelligent model training can enhance intrusion detection. The accuracy of RF is improved by 0.15% and the training time is also reduced by 78.4%, while the accuracy of DT has increased by 0.13%, also training time has been reduced by 82.1%. This can be helpful in sensitive systems, critical systems, Internet of Things (IoT) systems, and smart systems.

Table VIII: Performance Evaluation

Tech.	Acc%	Pre%	Rec%	F1-Score%	Training Time (s)
[3]	94.58	96.51	-	97.05	52.3
[4]	99.7	-	-	99.7	-
[5]	80	-	-	-	-
[6]	99	98	97	99	-
[7]	85.56	-	-	-	-
[8]	84.36	84.71	-	84.40	1140
[9]	86.59	88.55	-	86.88	-
[10]	84.03	-	-	98.99	109.4
[11]	81.21	96.59	-	80.81	330
[12]	86.79	89.46	-	87.56	183.19
[13]	81.17	92.1	-	81.56	-
[14]	99.67	-	-	-	-
DT	99.63	99.6	99.6	99.6	7.73
RF	99.75	99.7	99.7	99.7	30.25

Table VIII confirms our proposed research, that RF outperforms other classifiers in defensive cyber security against network intrusions. It is viable to discuss our proposed work to the other research work that is comparable to our results, as a result we are going to highlight potential differences between us and the work in[4,6,14].

The author in[6] achieved 99% accuracy, but this result is biased because he uses only 10% and 20% from the original dataset for training and testing his proposed machine-learning model.

In[4] training and testing were conducted using only the six selected attributes based on the IG algorithm, their accuracy metric seems comparable to our proposed model but is lesser than our proposed accuracy. This is justifiable because we use seven features not six as in[4]. This incurs the fact that given more knowledge about the network traffic means better understanding, and ultimately better results

The authors in[14] used RF and J48 (DT) on NSL-KDD. They differ than this proposed work in data preprocessing methodologies and utilized feature selection technique. Their final performance is fairly close to us, but ours is better in accuracy and training time.

V. CONCLUSION AND FUTURE WORK

The analysis proposed in this study on NSL-KDD, using solid data preprocessing and training of machine learning classifier; RF, has provided valuable insights. The findings indicate that tree-based classifiers (i.e., RF and DT) outperforms classifiers belonging to other categories. The optimization of classifiers (RF and DT) using sampling for data preprocessing and ranker for feature selection prove to be the optimal state on NSL-KDD dataset, on the other hand, technique parameter tuning did prove to be useful, individually, in enhancing performance, but not entirely sufficient when combined with other smart aspects, like feature selection. RF's performance resulted in 99.75% accuracy and 78% decrease in training time, on the other hand, DT's performance resulted in 99.63% accuracy and 82.1% reduction in training time.

The collective investigation of sampling, data splitting, parameter tuning, and feature selection has proven their importance in providing higher accuracy and less training time.

VI. REFERENNCES

[1] K. A. Scarfone and P. M. Mell, "Guide to Intrusion Detection and Prevention Systems (IDPS)," National Institute of Standards and Technology, Gaithersburg, MD, NIST SP 800-94, 2007. doi: 10.6028/NIST.SP.800-94.

[2] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ottawa, ON, Canada: IEEE, Jul. 2009, pp. 1–6. doi: 10.1109/CISDA.2009.5356528.

[3] U. C. Akuthota and L. Bhargava, "Network intrusion classification for IoT networks using an extreme learning machine," Eng. Res. Express, vol. 6, no. 2, p. 025217, Jun. 2024, doi: 10.1088/2631-8695/ad4cb5.

[4] A. H. Aljammal, I. Al-Oqily, M. Obiedat, A. Qawasmeh, S. Taamneh, and F. I. Wedyan, "Anomaly intrusion detection using machine learning-IG-R based on NSL-KDD dataset," Bull. Electr. Eng. Inform., vol. 13, no. 6, pp. 4466–4474, Dec. 2024, doi: 10.11591/eei.v13i6.7308.

[5] Y. Yuliana, D. H. Supriyadi, M. R. Fahlevi, and M. R. Arisagas, "Analysis of NSL-KDD for the Implementation of Machine Learning in Network Intrusion Detection System," J. Inform. Inf. Syst. Softw. Eng. Appl. INISTA, vol. 6, no. 2, pp. 80–89, Feb. 2024, doi: 10.20895/inista.v6i2.1389.

[6] S. Mehari and Prof. Dr. A. K. Acharya, "Comparative Analysis of Intrusion Detection Attack Based on Machine Learning Classifiers," Int. J. Comput. Sci. Netw. Secur., vol. 24, no. 10, pp. 115–124, Oct. 2024, doi: 10.22937/IJCSNS.2024.24.10.14.

[7] M. Rani and Gagandeep, "Effective network intrusion detection by addressing class imbalance with deep neural networks multimedia tools and applications," Multimed. Tools Appl., vol. 81, no. 6, pp. 8499–8518, Mar. 2022, doi: 10.1007/s11042-021-11747-6.

[8] S. Mohamed and R. Ejbali, "Deep SARSA-based reinforcement learning approach for anomaly network intrusion detection system," Int. J. Inf. Secur., vol. 22, no. 1, pp. 235–247, Feb. 2023, doi: 10.1007/s10207-022-00634-2.

[9] J. Cui, L. Zong, J. Xie, and M. Tang, "A novel multi-module integrated intrusion detection system for high-dimensional imbalanced data," Appl. Intell., vol. 53, no. 1, pp. 272–288, Jan. 2023, doi: 10.1007/s10489-022-03361-2.

[10] S. M. Kasongo, "A deep learning technique for intrusion detection system using a Recurrent Neural Networks based framework," Comput. Commun., vol. 199, pp. 113–125, Feb. 2023, doi: 10.1016/j.comcom.2022.12.010.

[11] Y. Liu, K. Zhang, and Z. Wang, "Intrusion detection of manifold regularized broad learning system based on LU decomposition," J. Supercomput., vol. 79, no. 18, pp. 20600–20648, Dec. 2023, doi: 10.1007/s11227-023-05403-z.

[12] P. Zhou, H. Zhang, and W. Liang, "Research on hybrid intrusion detection based on improved Harris Hawk optimization algorithm," Connect. Sci., vol. 35, no. 1, p. 2195595, Dec. 2023, doi: 10.1080/09540091.2023.2195595.

[13] M. Keshk, N. Koroniotis, N. Pham, N. Moustafa, B. Turnbull, and A. Y. Zomaya, "An explainable deep learning-enabled intrusion detection framework in IoT networks," Inf. Sci., vol. 639, p. 119000, Aug. 2023, doi: 10.1016/j.ins.2023.119000.

[14] Farnaaz, Nabila and M. A. Jabbar. "Random Forest Modeling for Network Intrusion Detection System." Procedia Computer Science 89 (2016): 213-217.

[15] E. Shehab, S. Elsaid, and A. Mattar. A Comprehensive Key Features Analysis and Recommendations based Cyber Intrusion Detection for Satellite-Terrestrial Networks. 2024 6th Novel Intelligent and Leading Emerging Sciences Conference (NILES), Giza, Egypt, (2024).

[16] S. Elsaid, E. Shehab, A. Mattar, A. Azar, and I. Hameed. Hybrid intrusion detection models based on GWO optimized deep learning. Discover Applied Sciences 6(1): 1-34 (2024).