Enhancing Multi-Agent Reinforcement Learning Intrusion Detection Systems Using Random Forest Q-Value Estimation

Ricky Aurelius Nurtanto Diaz
Faculty of Engineering, Udayana University, Bali, Indonesia | Department of Computer Systems, Institute of Technology and Business STIKOM Bali, Bali, Indonesia
ricky@stikom-bali.ac.id (corresponding author)

I. Ketut Gede Darma Putra
Department of Information Technology, Faculty of Engineering, Udayana University, Bali, Indonesia
ikgdarmaputra@unud.ac.id

Made Sudarma
Department of Electrical Engineering, Faculty of Engineering, Udayana University, Bali, Indonesia
msudarma@unud.ac.id

I. Made Sukarsa
Department of Information Technology, Faculty of Engineering, Udayana University, Bali, Indonesia
sukarsa@unud.ac.id

I. Wayan Budi Sentana
Department of Information Systems, Politeknik Negeri Bali, Indonesia
budisentana@pnb.ac.id

Ni Luh Gede Pivin Suwirmayanti
Department of Computer Systems, Institute of Technology and Business STIKOM Bali, Bali, Indonesia
pivin@stikom-bali.ac.id

Received: 24 March 2025 | Revised: 29 April 2025 and 5 May 2025 | Accepted: 10 May 2025

 DOI: https://doi.org/10.48084/etasr.11113

ABSTRACT

Intrusion Detection Systems (IDSs) analyze network traffic and system activity to identify anomalies or suspicious attack patterns. Various artificial intelligence-based approaches have been explored, including Deep Learning (DL) and Multi-Agent Reinforcement Learning (MARL) to increase their accuracy. This study combines MARL with Random Forest (RF) for Q-value estimation and utilizes two agents, a Detector and a Classifier. The proposed method was evaluated on three public datasets, including UNSW-NB15, NSL-KDD, and UKM-IDS20. The experimental results showed that the Detector Agent achieved higher accuracy (99.95%) compared to the Classifier Agent (80.63%) for the UNSW-NB15 dataset. On the NSL-KDD dataset, both agents performed similarly, with the Detector Agent achieving 99.82% accuracy and the Classifier Agent 99.80%. In contrast, for the UKM-IDS20 dataset, the Classifier Agent slightly outperformed the Detector Agent, with accuracies of 99.98% and 99.94%, respectively. These findings demonstrate the effectiveness of MARL-based IDS and highlight variations in agent performance across different datasets.

Keywords-IDS; MARL; RF; q-value estimation

I. INTRODUCTION

Intrusion Detection Systems (IDSs) have experienced rapid development along with the growing need for more adaptive and resilient security systems in the face of cyber threats [1-3]. Initially, IDSs relied on signature-based and fixed-rule methods to recognize attacks, but this approach has limitations in identifying undocumented threats. Therefore, new techniques are developed to increase accuracy and performance. IDSs work by analyzing network traffic and system activity to detect anomalies or suspicious attack patterns. IDSs consist of two main categories: Network-based IDSs (NIDSs), which operate by monitoring network traffic, and Host-based IDSs (HIDSs), which function at the individual device level [4-6]. Traditional IDSs often have limitations in dynamically recognizing new attack patterns, resulting in the need for more flexible and artificial intelligence-based approaches [7-10]. IDS methods based on machine learning (ML) and deep learning (DL) have been widely deployed in recent years [4-6, 11]. Several studies have shown that ML and DL models can improve attack detection with more complex and adaptive pattern analysis. For example, XGBoost has been combined with LSTM, RNN, and GRU [1]. In [12], a CNN was combined with Random Forest (RF) for an IDS in a fog computing environment [12], and in [13] a hybrid CNN with LSTM was used for this purpose.

To increase IDS effectiveness, various Deep Learning (DL) approaches have been explored, one of which is Multi-Agent Reinforcement Learning (MARL), which has been extensively researched and developed. MARL allows multiple agents to work collaboratively to identify and address cyberthreats more effectively. With reinforcement learning-based strategies, MARL can adjust detection policies based on evolving attack patterns. However, one of the main challenges in MARL is the estimation of Q-values, which is often unstable and can lead to suboptimal detection strategies[14-15]. Several studies have applied the MARL approach, including applying MARL to big data streaming and cloud computing to ensure that potential intrusions are identified and addressed as they occur [14, 16, 17]. Other implementations are related to network optimization tasks by applying two MARL agents for route optimization and DDoS protection [16, 17]. Some studies put little emphasis on the MARL parameters, which greatly affect its adaptability. In [18], a multi-agent approach was used in Software-Defined Networks (SDN) to improve multipath routing optimization, as well as to identify and prevent DDoS traffic. The results of this study demonstrated a considerable enhancement in network performance with a multiagent approach compared to the more conventional single-agent method.

Deep Reinforcement Learning (DRL) is superior to supervised learning in cybersecurity, especially in detecting unknown attacks (zero-day attacks) without relying on previously documented attack patterns. In addition, DRL in IDS can reduce false positives and increase detection accuracy, especially in the face of attacks such as DDoS, polymorphic malware, and IoT-based threats, which are difficult to detect with traditional methods [19]. Moreover, combining DL methods such as Bi-LSTM with GRU has shown the highest accuracy (99.999%), making it one of the most accurate models in detecting cyberattacks [20].

As discussed earlier, MARL has an advantage in adaptive detection capabilities but still faces challenges in estimating unstable Q values [13, 15], which can lead to less-than-optimal attack detection. This study aimed to develop a MARL-based IDS model optimized with RF Q-value Estimation (RF-QE) to decrease false positives, increase detection accuracy, and improve the learning stability of MARL agents in detecting cyber threats. Using RF as an ensemble learning method, this study aims to improve the estimation of Q-values by addressing the problem of overestimation that often occurs in MARL. Therefore, it is hoped that the IDS system optimized with this method can increase its resilience and adaptability in detecting cyber attacks that are increasingly complex and developing dynamically. Accuracy, precision, recall, and F1-score were used to evaluate the effectiveness of this proposed IDS and to understand how MARL optimization with RF-QE can improve IDS performance compared to the original MARL method.

II. PROPOSED METHOD

The proposed approach consists of three primary components: Dataset preprocessing, MARL, and RF. The flow diagram in Figure 1 presents a hybrid approach that integrates MARL with RF-QE to develop an adaptive classifier to detect different types of cyber threats.

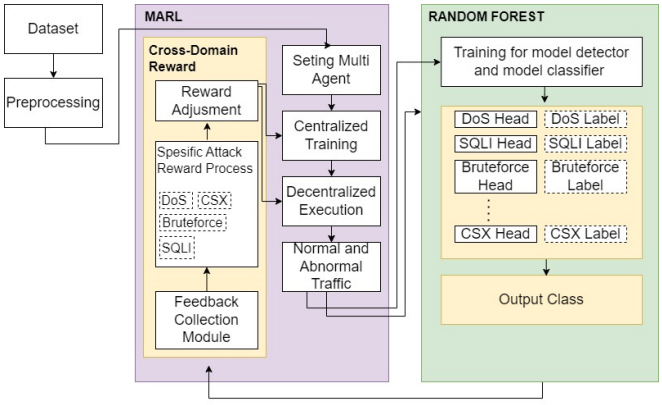

Fig. 1. The proposed method.

A. Dataset Preprocessing

The model starts with raw data collection, which undergoes preprocessing to remove noise, normalize values, and extract relevant features. The cleaned dataset is then used to train the MARL model.

B. Multi-Agent Reinforcement Learning (MARL) Framework

MARL employs multiple agents, namely a Detector Agent to detect and classify normal and abnormal traffic and a Classifier Agent to determine the type of attack. The Detector Agent detects whether traffic is normal or abnormal by taking action for normal = 0 and abnormal = 1. This is learned in the training process from the dataset labeled 0 and 1 for each type of traffic. On the other hand, the Classifier Agent will only be active if the Detector Agent detects an anomaly (abnormal traffic). Then, the Classifier Agent will run an action to determine the type of attack, such as DoS or Backdoor.

The core component of this phase is the MARL Q-value estimation, which is formed by RF and a cross-domain reward mechanism, which evaluates the agent's actions based on attack categories. Q-value is an estimate of the total reward that an agent can receive when he performs an action under a certain state and follows the best strategy afterward. In this case, there are two Q-values, one for the Detector Agent (QDetector) and one for the Classifier Agent (QClassifier). To produce a Q-value, Random Forest trains and works with combined inputs, namely state (derived from the dataset feature vector) and action, derived from data labels, namely 0 or 1 (for QDetector) and attack types such as DoS or Backdoor (for QClassifier). Based on the Q-value generated by the Random Forest, the agent then performs actions as a detector or classifier. Furthermore, the reward process is carried out by giving a value of +1 for correct predictions and -1 for incorrect predictions from both existing agents. The MARL model uses a reward adjustment module that provides feedback based on the impact of detected anomalies. The rewards are adjusted using:

$$R_t = \propto . R_{previous} + \beta . Attack\ Impact \quad (1)$$

where R_t is the updated reward at time t, $\propto$ is the weight assigned to previous rewards, $R_{previous}$ is the accumulated reward from past decisions, and β is the attack impact factor, which assigns different penalties or rewards based on the severity of an attack.

Agents operate in a Centralized Training and Decentralized Execution (CTDE) framework. In centralized training, multiple agents learn collaboratively using a shared knowledge base, and in decentralized execution, each agent independently detects attacks during real-time deployment. The Feedback Collection Module ensures that agents receive adaptive training based on previous experiences. The attack types included in the training are from the datasets used.

C. Random Forest

In training MARL to identify traffic anomalies, an RF model is employed to estimate the Q-value. The model is trained using labeled data, associating network traffic features with attack types. The classification consists of multiple decision trees, where each tree votes for a class. The final classification is performed using majority voting:

$$P(C_k) = \frac{1}{N} \sum_{i=1}^{N} f_i(\chi) \quad (2)$$

where $P(C_k)$ is the probability of class k, N is the total number of trees in the forest, and $f_i()$ is the prediction of the i-th decision tree for input χ.

III. RESULT AND DISCUSSION

To evaluate the performance of the proposed model, three different datasets were used with predefined dataset and model parameters. Next, the performance of the model was based on the multi-agent performance, and the overall performance was compared with some of the previously discussed related studies.

A. Dataset Description

Public IDS datasets were used, which are often employed in research on IDS development. UNSW-NB15 and NSL-KDD are popular datasets, and UKM-IDS20 is a relatively new dataset. Table I provides details on these datasets.

TABLE I. DATASET DETAILS

Dataset	Source	Number of features	Number of instances
UNSW-NB15 [21-25]	University of New South Wales	45	82332 rows
NSL-KDD [26]	University of New Brunswick- Kaggle Dataset	44	125972 rows
UKM-IDS20 [27]	National University of Malaysia - Kaggle Dataset	48	10308 rows

The attack types found on all datasets were used for the multiclass classification process. Table II lists the attack types in each dataset.

TABLE II. DATASET ATTACKS LIST

Dataset	Attack list
UNSW-NB15	Normal, Backdoor, DoS, Exploits, Fuzzers, Generic, Shellcode, Worms, Analysis, and Reconnaissance
NSL-KDD	Normal, Imap, Ipsweep, Land, Loadmodule, Multihop, Neptune, Nmap, Pod, Phf, Portsweep, Rootkit, Satan, Smurf, Spy, Teardrop, Warezclient, Warezmaster, Back, Buffer_Overflow, Ftp_Write, Guess_Passwd
UKM-IDS20	Normal, ARP Poisoning, Beef HTTP Exploits, Mass HTTP Requests, Metasploit Exploits, Port Scanning, TCP Flood, UDP Data Flood

B. Parameters

The model parameters used in this study consist of five parameters. The parameters for MARL environments consist of discount_factor, epsilon, and num_episodes (the amount of training on all data and its subsets to achieve a converged Q-value), and the parameters for RF consist of test_size (the data ratio for testing) and random_state parameters. The default number of trees from the random forest regressor was used, which is 100 trees. For the data distribution, 70% of the data was used for training and 30% for testing (test_size = 0.3). Table III shows the parameter values used.

TABLE III. MODEL PARAMETERS

Parameters	Value
discount_factor	0.9
epsilon	0.9
num_episodes	5
test_size	0.3
random_state	42

C. Performance Evaluation

In the implementation, two agents were used in MARL, namely a Detector and a Classifier Agent. The Detector Agent is tasked with making predictions based on binary classifiers, where the value is 0 for normal traffic data and 1 for anomaly traffic data. Suppose the Detector Agent finds an anomaly from the existing data. In that case, the Classifier Agent will

continue the detection process to predict the type of attack that occurs (multiclass classifier process). The results of these two agents are illustrated in Tables IV, V, and VI, accompanied by a comparison graph.

TABLE IV. MODEL PERFORMANCE ON UNSW-NB15 DATASET

Agent type	Performance scores on MARL RF-QE			
	Acc.	Prec.	Rec.	F1
Detector Agent	99.95%	99.95%	99.95%	99.95%
Classifier Agent	80.63%	80.34%	80.63%	80.36%

As shown in Table IV, in the test with the UNSW-NB15 dataset, the Detector Agent had better accuracy than the Classifier Agent, namely 99.95% compared to 80.63%.

Then, a model experiment on the NSL-KDD dataset showed that the Detector Agent was slightly superior to the Classifier Agent, with accuracies of 99.82% and 99.80%, respectively. Table V illustrates the performance of this multi-agent model on the NSL-KDD dataset.

TABLE V. MODEL PERFORMANCE ON NSL-KDD DATASET

Agent type	Performance scores on MARL RF-QE			
	Acc.	Prec.	Rec.	F1
Detector Agent	99.82%	99.82%	99.82%	99.82%
Classifier Agent	99.80%	99.84%	99.80%	99.81%

Finally, the proposed model was tested on a newer dataset but with a much smaller amount of data compared to the previous two. The tests on the UKM-IDS20 dataset showed that the Classifier Agent had a slightly better accuracy compared to the Detector Agent, with 99.98% versus 99.94%. Table VI shows the performance of the multi-agent model on the UKM-IDS20 dataset.

TABLE VI. MODEL PERFORMANCE ON UKM-IDS20 DATASET

Agent type	Performance scores on MARL RF-QE			
	Acc.	Prec.	Rec.	F1
Detector Agent	99.94%	99.94%	99.94%	99.94%
Classifier Agent	99.98%	99.98%	99.98%	99.98%

D. Comparison with Related Works

The proposed model was then compared with previously discussed studies. Table IV shows a comparison of performance results. These results show that the proposed MARL RF-QE model performed better on the NSL-KDD dataset. On the contrary, when using the UNSW-NB15 dataset, the proposed model was superior in terms of accuracy to the XGBoost-GRU (Multiclass Classification) algorithm [1] but inferior to CNN-IDS with RF [12].

TABLE VII. MODEL PERFORMANCE COMPARISON WITH RELATED WORKS

Ref	Year	Models	Dataset	Accuracy
[1]	2023	XGBoost-LSTM (Binary Classification)	NSL-KDD	88.13%
		XGBoost-Simple-RNN (Binary Classification)	UNSW-NB15	87.07%
		XGBoost-LSTM (Multiclass Classification)	NSL-KDD	86.93%
		XGBoost-GRU (Multiclass Classification)	UNSW-NB15	78.40%
[14]	2024	MARL based on the DQN algorithm	NSL-KDD	97.44%
[12]	2023	CNN-IDS with RF	UNSW-NB15	97.50%
Proposed model	2025	MARL Enhanced by RF-QE (Multiclass Classification)	NSL-KDD	99.80%
		MARL Enhanced by RF-QE (Multiclass Classification)	UNSW-NB15	80.63%
		MARL Enhanced by RF-QE (Multiclass Classification)	UKM-IDS20	99.98%

IV. CONCLUSION

This study applied MARL using RF for Q-value estimation, using two agents, namely a Detector and a Classifier Agent. In the test case with the UNSW-NB15 dataset, the Detector Agent had better accuracy than the Classifier Agent (99.95% versus 80.63%). In evaluation using the NSL-KDD dataset, the Detector Agent was slightly superior to the Classifier Agent, with an accuracy score of 99.82% compared to 99.80%. In the UKM-IDS20 dataset, the Classifier Agent had slightly better accuracy compared to the Detector Agent, with 99.98% compared to 99.94%. Comparing the results with related works, the proposed model was superior in experiments on the NSL-KDD dataset. However, on the UNSW-NB15 dataset, the proposed model was superior in terms of accuracy to the XGBoost-GRU (Multiclass Classification) [1] but inferior to CNN-IDS with RF [12]. This study has limitations in the use of three IDS datasets with tests that still use split data from them. In future research, the model can be further developed and tested with other datasets, including IoT datasets, and its accuracy will also be tested using real-time network activity data.

REFERENCES

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

[2] S. Alzughaibi and S. El Khediri, "A Cloud Intrusion Detection Systems Based on DNN Using Backpropagation and PSO on the CSE-CIC-IDS2018 Dataset," *Applied Sciences*, vol. 13, no. 4, Feb. 2023, Art. no. 2276, https://doi.org/10.3390/app13042276.

[3] S. More, M. Idrissi, H. Mahmoud, and A. T. Asyhari, "Enhanced Intrusion Detection Systems Performance with UNSW-NB15 Data Analysis," *Algorithms*, vol. 17, no. 2, Feb. 2024, Art. no. 64, https://doi.org/10.3390/a17020064.

[4] Y. C. Wang, Y. C. Houng, H. X. Chen, and S. M. Tseng, "Network Anomaly Intrusion Detection Based on Deep Learning Approach," *Sensors*, vol. 23, no. 4, Feb. 2023, Art. no. 2171, https://doi.org/10.3390/s23042171.

[5] D. Javeed, M. S. Saeed, I. Ahmad, P. Kumar, A. Jolfaei, and M. Tahir, "An Intelligent Intrusion Detection System for Smart Consumer Electronics Network," *IEEE Transactions on Consumer Electronics*, vol.

69, no. 4, pp. 906–913, Nov. 2023, https://doi.org/10.1109/TCE.2023.3277856.

[6] S. Rajasoundaran, S. V. N. S. Kumar, M. Selvi, K. Thangaramya, and K. Arputharaj, "Secure and optimized intrusion detection scheme using LSTM-MAC principles for underwater wireless sensor networks," *Wireless Networks*, vol. 30, no. 1, pp. 209–231, Jan. 2024, https://doi.org/10.1007/s11276-023-03470-x.

[7] G. Logeswari, S. Bose, and T. Anitha, "An Intrusion Detection System for SDN Using Machine Learning," *Intelligent Automation & Soft Computing*, vol. 35, no. 1, pp. 867–880, 2023, https://doi.org/10.32604/iasc.2023.026769.

[8] E. Altulaihan, M. A. Almaiah, and A. Aljughaiman, "Anomaly Detection IDS for Detecting DoS Attacks in IoT Networks Based on Machine Learning Algorithms," *Sensors*, vol. 24, no. 2, Jan. 2024, Art. no. 713, https://doi.org/10.3390/s24020713.

[9] Md. A. Hossain and Md. S. Islam, "Ensuring network security with a robust intrusion detection system using ensemble-based machine learning," *Array*, vol. 19, Sep. 2023, Art. no. 100306, https://doi.org/10.1016/j.array.2023.100306.

[10] F. A. Vadhil, M. L. Salihi, and M. F. Nanne, "Machine learning-based intrusion detection system for detecting web attacks," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 13, no. 1, Mar. 2024, Art. no. 711, https://doi.org/10.11591/ijai.v13.i1.pp711-721.

[11] S. Songma, T. Sathuphan, and T. Pamutha, "Optimizing Intrusion Detection Systems in Three Phases on the CSE-CIC-IDS-2018 Dataset," *Computers*, vol. 12, no. 12, Nov. 2023, Art. no. 245, https://doi.org/10.3390/computers12120245.

[12] K. Kaliyaperumal, C. Murugaiyan, D. Perumal, G. Jayaraman, and K. Samikannu, "Combined Ensemble Intrusion Detection Model using Deep learning with Feature Selection for Fog Computing Environments," *Acta Scientiarum. Technology*, vol. 45, Aug. 2022, Art. no. e60551, https://doi.org/10.4025/actascitechnol.v45i1.60551.

[13] R. H. Altaie and H. K. Hoomod, "An Intrusion Detection System using a Hybrid Lightweight Deep Learning Algorithm," *Engineering, Technology & Applied Science Research*, vol. 14, no. 5, pp. 16740–16743, Oct. 2024, https://doi.org/10.48084/etasr.7657.

[14] F. Louati, F. B. Ktata, and I. Amous, "Big-IDS: a decentralized multi agent reinforcement learning approach for distributed intrusion detection in big data networks," *Cluster Computing*, vol. 27, no. 5, pp. 6823–6841, Aug. 2024, https://doi.org/10.1007/s10586-024-04306-9.

[15] J. Popper and M. Ruskowski, "Using Multi-Agent Deep Reinforcement Learning For Flexible Job Shop Scheduling Problems," *Procedia CIRP*, vol. 112, pp. 63–67, 2022, https://doi.org/10.1016/j.procir.2022.09.039.

[16] S. Finistrella, S. Mariani, and F. Zambonelli, "Multi-Agent Reinforcement Learning for Cybersecurity: Classification and survey," *Intelligent Systems with Applications*, vol. 26, Jun. 2025, Art. no. 200495, https://doi.org/10.1016/j.iswa.2025.200495.

[17] R. Agila, R. Estrada, and K. Rohoden, "Resource Allocation with Graph Neural Networks-Multi Agent Reinforcement Learning for 6G HetNets," *Procedia Computer Science*, vol. 241, pp. 24–31, 2024, https://doi.org/10.1016/j.procs.2024.08.006.

[18] D. K. Dake, J. D. Gadze, G. S. Klogo, and H. Nunoo-Mensah, "Multi-Agent Reinforcement Learning Framework in SDN-IoT for Transient Load Detection and Prevention," *Technologies*, vol. 9, no. 3, Jun. 2021, Art. no. 44, https://doi.org/10.3390/technologies9030044.

[19] M. Sewak, S. K. Sahay, and H. Rathore, "Deep Reinforcement Learning in the Advanced Cybersecurity Threat Detection and Protection," *Information Systems Frontiers*, Aug. 2022, https://doi.org/10.1007/s10796-022-10333-x.

[20] S. Soliman, W. Oudah, and A. Aljuhani, "Deep learning-based intrusion detection approach for securing industrial Internet of Things," *Alexandria Engineering Journal*, vol. 81, pp. 371–383, Oct. 2023, https://doi.org/10.1016/j.aej.2023.09.023.

[21] M. Sarhan, S. Layeghy, N. Moustafa, and M. Portmann, "NetFlow Datasets for Machine Learning-Based Network Intrusion Detection Systems," in *Big Data Technologies and Applications*, vol. 371, Z. Deze, H. Huang, R. Hou, S. Rho, and N. Chilamkurti, Eds. Springer International Publishing, 2021, pp. 117–135.

[22] N. Moustafa, G. Creech, and J. Slay, "Big Data Analytics for Intrusion Detection System: Statistical Decision-Making Using Finite Dirichlet Mixture Models," in *Data Analytics and Decision Support for Cybersecurity*, I. Palomares Carrascosa, H. K. Kalutarage, and Y. Huang, Eds. Springer International Publishing, 2017, pp. 127–156.

[23] N. Moustafa, J. Slay, and G. Creech, "Novel Geometric Area Analysis Technique for Anomaly Detection Using Trapezoidal Area Estimation on Large-Scale Networks," *IEEE Transactions on Big Data*, vol. 5, no. 4, pp. 481–494, Dec. 2019, https://doi.org/10.1109/TBDATA.2017.2715166.

[24] N. Moustafa and J. Slay, "The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set," *Information Security Journal: A Global Perspective*, vol. 25, no. 1–3, pp. 18–31, Apr. 2016, https://doi.org/10.1080/19393555.2015.1125974.

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

[26] 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, Canada, Jul. 2009, pp. 1–6, https://doi.org/10.1109/CISDA.2009.5356528.

[27] M. S. Al-Daweri, S. Abdullah, and K. A. Z. Ariffin, "An adaptive method and a new dataset, UKM-IDS20, for the network intrusion detection system," *Computer Communications*, vol. 180, pp. 57–76, Dec. 2021, https://doi.org/10.1016/j.comcom.2021.09.007.