Deep Reinforcement Learning-based Asymmetric Convolutional Autoencoder for Intrusion Detection

Yuqin Dai[1,*], Xinjie Qian[2] and Chunmei Yang[3]

[1]*School of Electronic Information and Artificial Intelligence, Yibin Vocational & Technical College, Yibin 644000, China*
[2]*College of Digital Economy, Yibin Industry Polytechnic College, Yibin 644000, China*
[3]*School of Changning County Vocational and Technical School, Yibin 644000, China*
E-mail: Daiyuqin20241105@163.com; Qianxinjie20241031@163.com; 15082648030@163.com
[*]*Corresponding Author*

Received 04 March 2025; Accepted 23 April 2025

Abstract

In recent years, intrusion detection systems (IDSs) have become a critical component of network security, due to the growing number and complexity of cyber-attacks. Traditional IDS methods, including signature-based and anomaly-based detection, often struggle with the high-dimensional and imbalanced nature of network traffic, leading to suboptimal performance. Moreover, many existing models fail to efficiently handle the diverse and complex attack types. In response to these challenges, we propose a novel deep learning-based IDS framework that leverages a deep asymmetric convolutional autoencoder (DACA) architecture. Our model combines advanced techniques for feature extraction, dimensionality reduction, and anomaly detection into a single cohesive framework. The DACA model is designed to effectively capture complex patterns and subtle anomalies in network traffic while significantly reducing computational complexity. By employing this

Journal of ICT Standardization, Vol. 13_1, 67–92.
doi: 10.13052/jicts2245-800X.1314

architecture, we achieve superior detection accuracy across various types of attacks even in imbalanced datasets. Experimental results demonstrate that our approach surpasses several state-of-the-art methods, including HCM-SVM, D1-IDDS, and GNN -IDS, achieving high accuracy, precision, recall, and F1-score on benchmark datasets such as NSL-KDD and UNSW-NB15. The results emphasize how effectively our model identifies complex and varied attack patterns. In conclusion, the proposed IDS model offers a promising solution to the limitations of current detection systems, with significant improvements in performance and efficiency. This approach contributes to advancing the development of robust and scalable network security solutions.

Keywords: Intrusion detection system, asymmetric convolutional autoencoder, network security, attack detection, feature extraction.

1 Introduction

With the growth of mobile internet, the widespread adoption of devices such as smartphones and IoT devices has made wireless networks crucial for daily activities and work. However, this also brings significant security challenges. Network attacks are becoming increasingly complex, with attackers using various techniques to steal data, disrupt systems, or cause service outages [1]. Traditional security measures are limited in handling intelligent, evolving attacks. Intrusion detection systems (IDSs) monitor traffic in real-time to detect abnormal behavior, but traditional methods like rule-based detection and signature matching struggle with new or zero-day attacks [2]. As attack techniques evolve, encryption and obfuscation bypass traditional IDSs, reducing detection efficiency. Statistical methods can detect anomalies but require large datasets and predefined models, making them unsuitable for dynamic environments [3]. Thus, efficient and dynamic IDS design has become a key issue.

With the rapid advancement of artificial intelligence, advanced algorithms are increasingly integrated into intrusion detection systems, ushering in a new era in cybersecurity. Deep reinforcement learning (DRL), which combines deep learning's feature extraction and reinforcement learning's strategy optimization, has shown great potential in detecting threats in complex, dynamic network environments [4–6]. Unlike traditional methods, DRL-based systems can learn from data and interact with the environment to optimize detection strategies without relying on predefined rules [7–9]. DRL systems not only improve accuracy against known attacks but also handle novel threats, offering adaptability and robustness. Additionally, DRL dynamically

adjusts detection strategies based on real-time network changes, optimizing decisions under various attack scenarios [10, 11]. These systems continue to improve detection accuracy and efficiency through continuous interaction, making them highly intelligent and self-improving. As a result, DRL-based mobile network intrusion detection systems provide a more efficient and flexible solution to complex security threats [12, 13]. Studies show that DRL-based IDSs have achieved significant results, reducing false positives and false negatives while handling novel attacks [14, 15]. Large-scale training and testing further demonstrate their efficiency and scalability, solidifying their practical application. Thus, DRL-based IDSs are becoming a promising technological direction.

This paper presents an innovative approach that combines deep reinforcement learning with adaptive feature extraction. The proposed solution first utilizes a deep asymmetric convolutional autoencoder (DACA) to automatically extract relevant features from network traffic, and then employs the DDPG algorithm to optimize intrusion detection decisions. This approach allows the system to dynamically adapt to changes in the network environment, accurately identify attack traffic, and exhibit high detection efficiency and accuracy, especially in the case of unknown intrusions.

To achieve efficient and precise mobile network intrusion detection, the key contributions of this paper are as follows:

- A novel intrusion detection approach is introduced, combining DACA with deep reinforcement learning, enabling automatic extraction of key features from network traffic. This significantly enhances the accuracy and robustness of the detection system, making it more effective in identifying both known and unknown attacks.
- An adaptive intrusion detection mechanism based on the DDPG algorithm is designed, allowing the system to dynamically optimize detection strategies. This mechanism enables the system to respond flexibly to changing network environments and evolving attack patterns, offering superior detection performance in real-time scenarios.

2 Related Work

2.1 Conventional Approaches to Intrusion Detection

As network attack techniques evolve, signature-based detection methods struggle to detect new attack types, such as zero-day and polymorphic attacks. This has led to increased interest in machine learning for IDSs.

Support vector machines (SVMs) are widely used in IDSs, particularly for binary classification tasks, where they classify samples by finding an optimal hyperplane to distinguish normal from malicious traffic [16, 17]. SVMs are known for their strong generalization ability, but face challenges with imbalanced data, high computational expense, and the need for fine-tuning to achieve optimal performance [18–20].

Decision trees use a tree structure to classify data based on a series of rules. In IDSs, they learn rules from network traffic features to categorize traffic as normal or malicious [21]. The key advantage of decision trees lies in their interpretability, as they provide clear insight into decision-making processes. However, they tend to overfit, particularly with complex datasets and small sample sizes. Random forests address this by using an ensemble approach, where multiple trees are trained and their outputs are combined, leading to better accuracy and increased robustness [22–24]. Random forests handle high-dimensional features well and can manage nonlinear relationships between features, providing strong noise resistance and high accuracy in IDS applications.

The k-nearest neighbor (KNN) algorithm classifies samples based on their proximity to other training samples [25]. The KNN algorithm is simple to implement and does not require a training phase, making it suitable for dynamic data. However, it has high computational costs, particularly with large datasets, and is sensitive to class imbalance. Naive Bayes, based on Bayes' theorem, calculates conditional probabilities of features to predict data categories [26–28]. Its advantage is high computational efficiency, making it suitable for large datasets. However, naive Bayes assumes feature independence, which may not be true for real-world network traffic, leading to suboptimal performance in complex attack scenarios.

Despite their successes, traditional machine learning methods have limitations. They often rely on manual feature selection and extraction, which is time-consuming and complex, especially with large, high-dimensional datasets. Methods like SVMs and decision trees also face the "curse of dimensionality," reducing training and prediction efficiency when dealing with high-dimensional data [29–31]. Additionally, traditional methods require large amounts of labeled data for training, posing a significant challenge when dealing with new attack types.

Despite these challenges, traditional machine learning methods remain crucial in IDSs, particularly when fast and effective detection is needed. With the development of deep learning and reinforcement learning, many researchers are now combining traditional methods with newer techniques

to leverage the strengths of both approaches, opening new possibilities for future IDS designs.

2.2 Deep Learning-based Intrusion Detection

In the domain of IDSs, DRL is emerging as a solution for addressing dynamic and unknown attacks. Unlike traditional static models, DRL optimizes strategies through interaction between an agent and its environment, allowing it to adapt to continuously changing attack patterns. The application of DRL in IDSs includes various approaches, such as a DQN, policy gradient methods, the DDPG, and proximal policy optimization (PPO), among others. Below are some specific methods and applications.

A DQN combines Q-learning with deep neural networks to manage high-dimensional state spaces. In IDSs, a DQN uses deep networks to approximate Q-values and learns attack patterns from network traffic [32, 33]. However, DQNs face challenges in high-dimensional spaces, such as low learning efficiency and local optima. To address this, recent research has introduced DDQNs (double deep Q-networks), which reduces Q-value overestimation by using two Q-networks, thus offering more stable results in fluctuating network environments [34, 35].

The DDPG is designed for continuous action spaces, using deep networks to approximate both policy and Q-value functions. It has shown strong performance in high-frequency attack scenarios, such as traffic scheduling or bandwidth control [36, 37], where it can adjust fine-grained detection strategies. Recent studies have expanded on DDPG by incorporating multi-agent systems for large-scale IDSs, where agents collaborate to optimize detection strategies in distributed network environments. PPO, a policy-gradient method, stabilizes training through a stable update strategy, addressing the high variance problem in reinforcement learning. It has been widely applied in IDSs, showing high detection accuracy across complex network environments [38, 39]. Multi-agent reinforcement learning (MARL) involves multiple agents working together to optimize strategies and improve detection. In IDSs, MARL simulates cooperation between detection nodes, enhancing detection performance, especially in large-scale distributed networks [40–42]. Another promising direction in IDS research is the integration of reinforcement learning (RL) with generative adversarial networks (GANs). GANs are used to generate malicious attack samples that augment the training datasets, which significantly improves the system's ability to detect previously unseen attacks [43]. The generator creates synthetic attack samples,

while the discriminator evaluates them, thereby enhancing the model's robustness against novel intrusion attempts. Adaptive reward mechanisms are crucial in DRL-based IDSs, as different attack types exhibit distinct network behaviors. Some studies propose dynamically adjusting the reward function to improve detection accuracy and system adaptability to changing threats [44, 45]. Similarly, policy integration methods, such as combining a DQN with PPO, can optimize detection by integrating multiple strategies, enhancing overall robustness [46].

Despite advancements, DRL in IDSs faces challenges like computational efficiency in high-dimensional state spaces, reward mechanism design, and policy stability. Balancing exploration and exploitation, avoiding overfitting, and improving model training efficiency remain critical issues. Moreover, the issue of handling imbalanced datasets in DRL-based IDS is becoming increasingly important, with recent studies proposing the use of semi-supervised learning techniques to improve model performance when labeled data is scarce. As technology progresses, DRL is expected to be a key research direction for future IDS development.

3 Method

3.1 Overview of Our Network

This study proposes a network intrusion detection system based on adaptive learning algorithms, which consists of three core modules: the feature extraction module, the anomaly detection module, and the learning optimization module. In the feature extraction module, we employ the DACA to dynamically assess and extract critical features from network traffic data. In the anomaly detection module, we use DDPG to classify normal and attack traffic based on the extracted features. The learning optimization module dynamically adjusts the detection strategy through reinforcement learning to ensure the system efficiently detects both known and unknown intrusions at the optimal pace. Through the collaborative operation of these three modules, the system is able to provide an accurate and personalized intrusion detection experience. Figure 1 illustrates the overall architecture.

3.2 Deep Asymmetric Convolutional Encoder

3.2.1 Autoencoder

An autoencoder is an unsupervised learning method used for dimensionality reduction and feature learning. It maps input data to a lower-dimensional

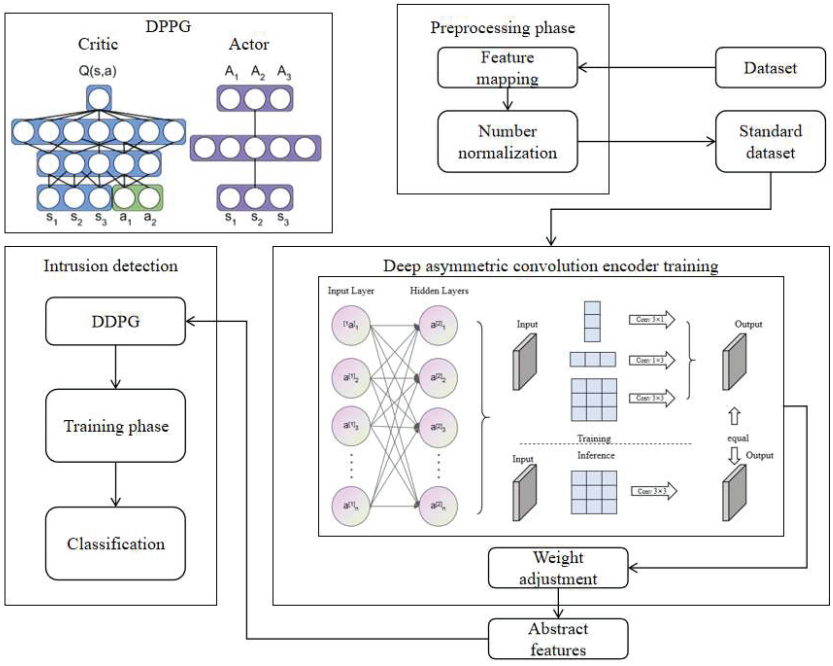

Figure 1 Overall workflow of the proposed model.

latent space for reconstruction. In intrusion detection, the autoencoder compresses high-dimensional network traffic features, extracts latent patterns, and removes noise, improving the efficiency of subsequent detection models. The architecture of the autoencoder is shown in Figure 2.

The autoencoder consists of an encoder and a decoder. The encoder maps the input to a latent space, generating a low-dimensional representation mmm. This is achieved using a nonlinear activation function:

$$\mathrm{h} = \mathrm{f}(\mathrm{x}; \mathrm{W_e}, \mathrm{b_e}) \tag{1}$$

where f is the activation function of the encoder (e.g., ReLU or Sigmoid), and $\mathrm{W_e}$ and $\mathrm{b_e}$ are the weights and biases of the encoder, respectively.

The decoding process is also achieved through a nonlinear activation function, represented as:

$$\widehat{\mathrm{x}} = \mathrm{g}(\mathrm{h}; \mathrm{W_d}, \mathrm{b_d}) \tag{2}$$

where g is the activation function of the decoder, and $\mathrm{W_d}$ and $\mathrm{b_d}$ are the weights and biases of the decoder.

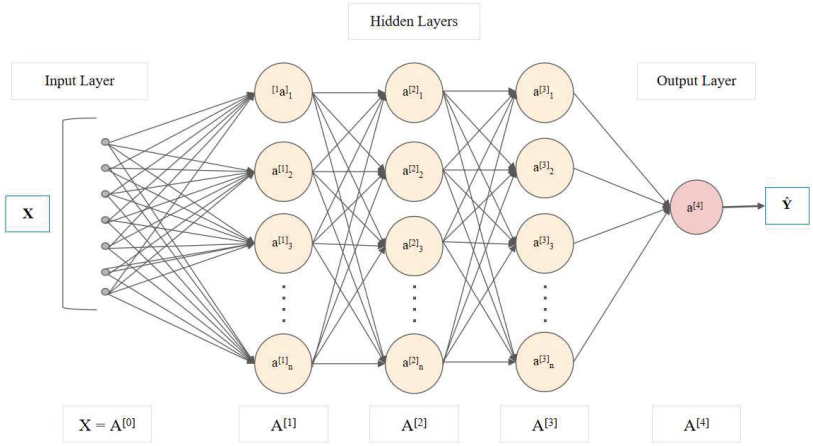

Figure 2 Architecture of the autoencoder.

To train the autoencoder, we minimize the difference between the input data x and the reconstructed data $\widehat{\mathrm{x}}$. A commonly used loss function is the MSE, expressed as:

$$\mathrm{L_{AE}} = \frac{1}{\mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{N}} \|\mathrm{x_i} - \widehat{\mathrm{x}}_\mathrm{i}\|^2 \tag{3}$$

where N is the number of training samples, $\mathrm{x_i}$ represents the input data, and $\widehat{\mathrm{x}}_\mathrm{i}$ denotes the reconstructed data.

Additionally, regularization terms are often added during the training process to prevent overfitting. An L2 regularization term can be expressed as:

$$\mathrm{L_{reg}} = \lambda(\|\mathrm{W_e}\|_2^2 + \|\mathrm{W_d}\|_2^2) \tag{4}$$

where λ is the regularization coefficient, and $\mathrm{W_e}$ and $\mathrm{W_d}$ are the weights of the encoder and decoder, respectively.

The final objective is to minimize the total loss function:

$$\mathrm{L_{total}} = \mathrm{L_{AE}} + \mathrm{L_{reg}} \tag{5}$$

By optimizing this loss function, the autoencoder learns effective feature representations that compress the dimensionality of network traffic data while preserving important intrusion detection information.

The autoencoder reduces computational burden and removes noise from the data, allowing detection models to focus on key intrusion-related features. This unsupervised approach is particularly useful for network traffic data,

which is difficult to label manually. By using the autoencoder, useful latent patterns are extracted, supporting more effective intrusion detection.

3.2.2 Deep asymmetric convolutional autoencoder (DACA)

The DACA combines the strengths of CNNs and AEs, making it highly effective for feature learning in network traffic data. The DACA excels at extracting multi-scale features and capturing complex patterns through its asymmetric structure, which enhances its adaptability to complex network traffic. The DACA architecture consists of multiple convolutional layers followed by the encoder part of an autoencoder. The convolutional layers first extract local features from the raw input data, and the autoencoder then compresses and learns the feature representations. Unlike traditional autoencoders, the DACA employs an asymmetric structure, where each layer uses different kernel sizes and depths. This asymmetric design enables the model to capture features at various scales, making it more effective in expressing complex data patterns.

In the input stage, convolution operations extract local features from the raw data. The first convolutional layer output is expressed as:

$$\mathrm{h_1} = \mathrm{Conv}(\mathrm{x}; \mathrm{K_1}) + \mathrm{b_1} \tag{6}$$

where Conv denotes the convolution operation, $\mathrm{K_1}$ is the first-layer convolution kernel, and b1 is the bias term. The output h1 is then passed to the next convolution layer.

Next, the second convolutional layer further extracts features, with the convolution kernel $\mathrm{K_2}$, as expressed by:

$$\mathrm{h_2} = \mathrm{Conv}(\mathrm{h_1}; \mathrm{K_2}) + \mathrm{b_2} \tag{7}$$

The DACA's layers use different kernel sizes to capture data at multiple scales. As layers deepen, the model extracts more abstract features.

For the n-th convolutional layer, this can be written as:

$$\mathrm{h_n} = \mathrm{Conv}(\mathrm{h_{n-1}}; \mathrm{K_n}) + \mathrm{b_n} \tag{8}$$

where $\mathrm{h_{n-1}}$ denotes the output generated by the preceding layer.

After convolution, the output is passed to the autoencoder's encoder, which compresses the features into a lower-dimensional space:

$$\mathrm{h} = \mathrm{f}(\mathrm{h_n}; \mathrm{W_e}, \mathrm{b_e}) \tag{9}$$

where $\mathrm{W_e}$ and $\mathrm{b_e}$ are the encoder's weights and biases, and $\mathrm{h_n}$ is the feature representation obtained from the convolutional layers.

To improve performance, batch normalization (BN) is used during training, expressed as:

$$\widehat{\mathrm{h}}_{\mathrm{n}} = \frac{\mathrm{h}_{\mathrm{n}} - \mu}{\sigma} \cdot \gamma + \beta \tag{10}$$

Optimizing this loss function allows the DACA to learn effective feature representations from network traffic data, which are then used for intrusion detection models. The DACA's ability to capture features at various scales and adapt to dynamic environments makes it a robust tool for intrusion detection, improving accuracy and reliability.

3.3 DDPG Reinforcement Learning Model Design

This study introduces an adaptive intrusion detection model built on the DDPG algorithm, aimed at enhancing decision-making in dynamic network environments. As a reinforcement learning method for continuous action spaces, the DDPG allows the model to autonomously learn optimal detection strategies and uncover unknown intrusions. The algorithm has two key components: the actor network, which generates action decisions, and the critic network, which evaluates their value. To improve learning stability and efficiency, the DDPG incorporates techniques like experience replay and target networks, allowing the model to learn from past experiences and stabilize updates.

In our model, the state space is composed of feature representations of network traffic, extracted by DACA. The state vector $\mathrm{S_t}$ can be represented as:

$$\mathrm{s_t} = [\mathrm{h_1}, \mathrm{h_2}, \ldots, \mathrm{h_n}] \tag{11}$$

where $\mathrm{h_1}, \mathrm{h_2}, \ldots, \mathrm{h_n}$ are the high-dimensional features extracted by the DACA.

The action space is discrete, consisting of the actions "Attack" and "Normal," i.e.:

$$\mathrm{a_t} \in \{\mathrm{Attack},\ \mathrm{Normal}\} \tag{12}$$

The policy network μ in DDPG generates continuous actions $\mathrm{a_t}$ for a given state $\mathrm{s_t}$, and the value network Q evaluates the quality of the action. The policy network can be represented as:

$$\mu(\mathrm{s_t}; \theta^{\mu}) = \mathrm{Actor}(\mathrm{s_t}) \tag{13}$$

where θ^{μ} represents the parameters of the policy network.

The value network Q aims to evaluate the expected cumulative future rewards from taking action $\mathrm{a_t}$ at state $\mathrm{s_t}$. The value function can be calculated as:

$$\mathrm{Q}(\mathrm{s_t}, \mathrm{a_t}; \theta^{\mathrm{Q}}) = \mathrm{r_t} + \gamma \mathbb{E}[\mathrm{Q}(\mathrm{s_{t+1}}, \mu(\mathrm{s_{t+1}}; \theta^{\mu}); \theta^{\mathrm{Q}})] \tag{14}$$

where γ is the discount factor, controlling the importance of future rewards. To reduce estimation errors, the DDPG uses target networks μ' and Q' to stabilize the learning process

$$\mathrm{Q}'(\mathrm{s_t}, \mathrm{a_t}; \theta^{\mathrm{Q}'}) = \mathrm{r_t} + \gamma \mathrm{Q}'(\mathrm{s_{t+1}}, \mu'(\mathrm{s_{t+1}}; \theta^{\mu'}); \theta^{\mathrm{Q}'}) \tag{15}$$

Through this update strategy, the DDPG continually adjusts both the policy and value functions to find the optimal detection strategy.

The DDPG also incorporates experience replay to improve data utilization and reduce correlations between consecutive samples.

By continuously optimizing the parameters in both networks, the DDPG improves the precision and stability of intrusion detection. Most importantly, the DDPG adapts to real-time changes in network traffic, progressively improving its ability to identify unknown intrusions.

In summary, the DDPG model leverages deep reinforcement learning to adjust detection strategies through interaction with the environment. With techniques like experience replay and target networks, the DDPG enhances learning efficiency and stability, as a result, it offers a flexible approach to intrusion detection in ever-changing mobile network environments.

4 Experiment

4.1 Experimental Environment

The experiments were conducted on a system equipped with an Intel Core i7-10700K CPU, 32 GB RAM, and an NVIDIA RTX 3080 GPU to ensure efficient processing and fast model training. The proposed intrusion detection model was implemented using Python, with deep learning libraries such as TensorFlow and Keras for model development. The datasets were processed and fed into the model using Pandas for data manipulation and Scikit-learn for feature preprocessing. The model training was performed in a Python 3.8 environment, with an emphasis on optimizing training time and computational efficiency. The experiments were run on an Ubuntu 20.04 operating system, ensuring a stable and reproducible environment for evaluating the model's performance.

4.2 Dataset

The NSL-KDD dataset [47], designed as an improved successor to the KDD Cup 1999 dataset, addresses challenges such as data redundancy and class imbalance. It features 125,973 samples for training and 22,544 for testing, encompassing 41 attributes like connection duration, protocol type, and failed login attempts. Incorporating attack categories such as DoS, R2L, U2R, and Probe, this dataset remains a prominent benchmark for IDS evaluations, despite being introduced over 20 years ago.

The UNSW-NB15 dataset [48], developed by the University of New South Wales, provides more modern and diverse network traffic data. Generated using the IXIA traffic generator, it includes 2,540,044 instances and 49 features, covering both continuous and discrete attributes. The dataset contains a wider range of attack types, such as DoS, DDoS, Exploits, Fuzzers, and Malware, with a training set of 175,341 samples and a testing set of 82,332 samples. It reflects contemporary network threats, offering a more complex and challenging evaluation environment for IDS models.

4.3 Data Preprocessing

In the data preprocessing phase, several essential steps are taken to prepare the datasets for model training and evaluation. First, IP addresses and port numbers are removed to prevent bias, ensuring the model focuses on network traffic patterns instead of host-specific data, which helps avoid overfitting. Unnecessary white spaces in categorical attributes are also eliminated to maintain consistency and ensure correct label assignments. Categorical labels, such as "normal" and various attack types (e.g., "DoS", "Probe"), are encoded into numeric values, allowing for efficient processing during training. Additionally, Min–Max scaling is applied to normalize features with different numerical ranges, ensuring equal contribution from all features to the model's learning process. These preprocessing steps result in clean, consistent, and normalized data, which is well-suited for effective model training.

4.4 Model Architecture and Hyperparameter Selection

4.4.1 Selection of network architecture and layer depth

In our model, the architecture of the DACA plays a critical role in feature extraction from network traffic data. We selected a relatively deep network structure to capture both low-level and high-level patterns in the

data, ensuring effective detection of a wide variety of attack types. The model consists of three convolutional layers followed by two fully connected layers. The first two convolutional layers use smaller kernels to capture local patterns, while the third layer has larger kernels to capture more abstract features. This layered approach allows the model to progressively extract more complex features from raw network traffic data. The depth of the network was selected based on the trade-off between model complexity and computational efficiency. Too many layers can lead to overfitting, while too few layers might not capture sufficient feature complexity.

4.4.2 Convolution kernels and network parameters

The convolutional layers utilize 3×3 and 5×5 kernels in the first two layers to capture fine-grained patterns, while a 7×7 kernel is used in the third layer to detect broader features. These kernel sizes were selected based on prior research, where smaller kernels detect simple patterns and larger kernels capture more complex data relationships. The model employs 32 filters in the first layer, 64 in the second, and 128 in the third, balancing model complexity and performance. For the fully connected layers, 128 and 64 units are used. The ReLU activation function is applied throughout the network, promoting faster convergence and reducing the likelihood of vanishing gradients.

4.5 Evaluation Metrics

The effectiveness of the proposed intrusion detection model is evaluated using standard metrics: accuracy, precision, recall, F1-score, and AUC. Accuracy reflects the overall performance, while precision and recall evaluate the model's effectiveness in correctly identifying attacks and minimizing false positives. The F1-score, which balances precision and recall, is especially useful for imbalanced datasets. The AUC evaluates how well the model distinguishes between attack and normal traffic. Together, these metrics offer a comprehensive evaluation of the model's effectiveness in detecting various attacks.

5 Result

5.1 Comparison of Different Methods

In the experimental results presented in Table 1, our proposed method demonstrates significant advantages. Leveraging the deep feature extraction capability of the DACA module, our model automatically learns efficient

Table 1 Comparison of different methods on datasets

Method	Dataset	Accuracy	Precision	Recall	F1-score
Ours	NSL-KDD Dataset	0.935	0.92	0.928	0.924
FCM-SVM [49]		0.905	0.89	0.895	0.892
FN-GNN [50]		0.91	0.895	0.905	0.9
1D-CNN-IDS [51]		0.895	0.88	0.885	0.882
NSGA2-LR [52]		0.87	0.855	0.86	0.858
GNN -IDS [53]		0.905	0.895	0.9	0.897
Ours	UNSW-NB15 Dataset	0.93	0.915	0.92	0.918
FCM-SVM [49]		0.875	0.865	0.87	0.868
FN-GNN [50]		0.88	0.87	0.875	0.873
1D-CNN-IDS [51]		0.86	0.855	0.86	0.858
NSGA2-LR [52]		0.85	0.845	0.85	0.847
GNN -IDS [53]		0.88	0.875	0.88	0.878

and representative features from raw network traffic, improving recognition accuracy. Our method achieved 0.935 accuracy, outperforming traditional methods like FCM-SVM (0.905) and 1D-CNN-IDS (0.895). This demonstrates how the combination of deep feature extraction and reinforcement learning enhances adaptability to complex intrusion patterns. The model also attained a precision of 0.92 and a recall of 0.928, indicating both high accuracy and strong detection capability. In terms of F1-score (0.924), our model showed better robustness, especially in noisy and imbalanced datasets. On the UNSW-NB15 dataset, our method achieved 0.93 accuracy, outperforming FCM-SVM (0.875) and 1D-CNN-IDS (0.86). The model's precision (0.915) and recall (0.92) further highlight its strong performance in distinguishing between normal and abnormal traffic. These results confirm that our model can effectively handle various attack patterns and maintain high performance in dynamic network environments. Overall, by combining the adaptive strategy optimization of deep reinforcement learning with the DACA's feature extraction capabilities, our model demonstrates high detection accuracy and strong generalization. These results validate that our method outperforms traditional machine learning and other deep learning approaches, making it well-suited for dynamic and complex network intrusion detection tasks.

Table 2 presents various methods in terms of training time, inference time, detection time, and memory usage. Our model demonstrates high efficiency across all key metrics. For training time, our method takes 145.32 seconds on the NSL-KDD dataset, outperforming others. In terms of inference time, it achieves 0.314 ms/instance, significantly lower than other methods,

Table 2 Detection time comparison

Method	Dataset	Training Time (s)	Inference Time (ms/instance)	Detection Time (s)	Memory Usage (MB)
Ours	NSL-KDD	145.32	0.314	25.42	119.75
FCM-SVM		160.89	1.218	40.35	110.2
FN-GNN		185.66	1.104	35.12	129.56
1D-CNN-IDS		210.74	1.501	45.68	139.82
NSGA2-LR		180.33	0.802	37.28	124.63
GNN -IDS		220.27	2.008	50.47	150.21
Ours	UNSW-NB15	180.65	0.333	27.48	130.15
FCM-SVM		200.44	1.302	42.17	119.31
FN-GNN		210.89	1.205	38.97	134.87
1D-CNN-IDS		240.15	1.808	48.76	144.21
NSGA2-LR		205.87	0.856	39.64	130.28
GNN -IDS		230.47	2.101	50.18	154.93

highlighting its real-time processing capability. For detection time, it takes 25.42 seconds, showing superior performance in dynamic environments. Regarding memory usage, our model consumes 119.75 MB, which is lower than several other methods, indicating better resource efficiency. By combining the deep asymmetric convolutional autoencoder (DACA) and the deep deterministic policy gradient (DDPG), our model optimizes both feature extraction and decision-making processes. This reduces computational and memory costs while maintaining high performance. Results on the UNSW-NB15 dataset further emphasize its stability, flexibility, and efficient use of resources. Overall, our method is highly competitive in terms of real-time performance and resource consumption, with computational complexity suitable for practical deployment.

To further analyze our model's decision mechanism, we visualize the feature importance patterns between normal and attack traffic, with results shown in Figure 3.

The radar chart reveals that attack traffic demonstrates significantly higher discriminative weights in source bytes (Src Bytes) and protocol type (Protocol) features (0.95 and 0.70 respectively), indicating these characteristics play a crucial role in attack detection, particularly as the abnormal surge in source bytes aligns with typical DoS attack patterns. In contrast, normal traffic shows stronger reliance on connection flag features (0.85), reflecting stable connection patterns in regular network activities. The similar importance levels of destination bytes (Dst Bytes) for both traffic types (0.65 vs 0.80) suggest

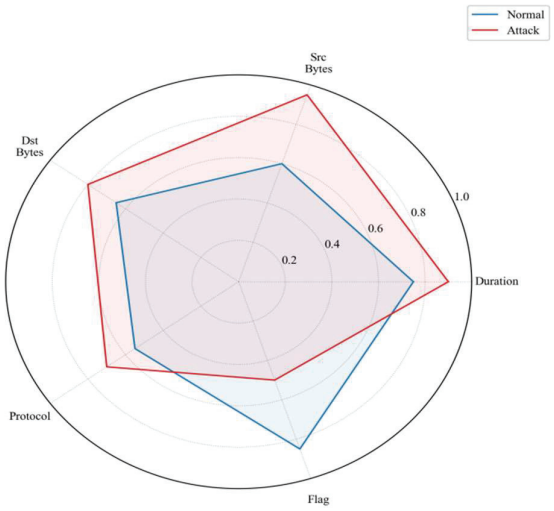

Figure 3 PCA visualization of the NSL-KDD dataset before and after applying our method.

some attacks may mimic normal data reception behavior, increasing detection challenges. The overall feature distribution demonstrates the model's effectiveness in distinguishing differential patterns between attack and normal traffic, providing interpretable evidence for the detection performance. The visualization clearly shows how different network features contribute to intrusion detection, with source-related metrics being particularly significant for identifying malicious activities while connection states remain more relevant for normal traffic classification.

5.2 Ablation Study

To further validate the contribution of each module to our method, we conducted ablation experiments by removing different modules and comparing their performance. The experimental results are shown in Table 3. When the autoencoder was removed (no autoencoder), the model's performance significantly declined, particularly in accuracy and F1-score, highlighting the autoencoder's critical role in feature extraction for high-dimensional and complex data. Next, removing the convolutional layers (no convolution) also led to a performance drop, emphasizing their importance in capturing local features and spatsly (no autoencoder and no convolution), the performance worsened notably, especially in F1-score and recall, proving that the combination of these two modules is crucial. Additionally, removing the

Table 3 Comparison of learning efficiency of different models on ASSISTments and MATH datasets

Method	Dataset	Accuracy	Precision	Recall	F1-score
Ours (full model)	NSL-KDD	0.935	0.92	0.928	0.924
No autoencoder		0.873	0.853	0.868	0.86
No convolution		0.868	0.848	0.853	0.85
No autoencoder and no convolution		0.838	0.828	0.828	0.826
No DDPG		0.878	0.863	0.868	0.866
Ours (full model)	UNSW-NB15	0.93	0.915	0.92	0.918
No autoencoder		0.868	0.853	0.868	0.863
No convolution		0.873	0.863	0.868	0.866
No autoencoder and no convolution		0.833	0.818	0.823	0.82
No DDPG		0.878	0.863	0.868	0.865

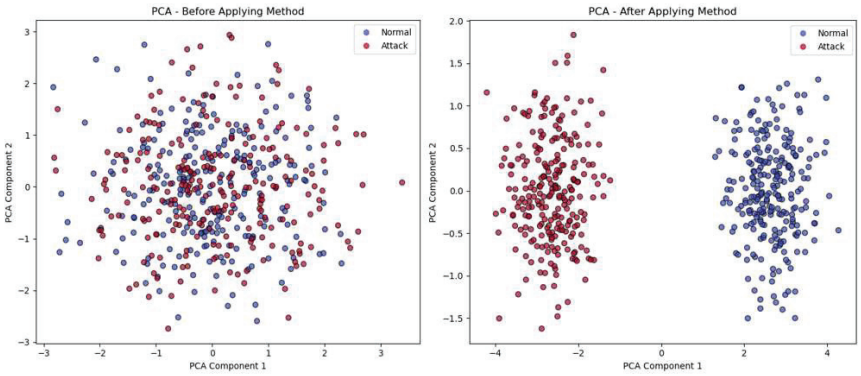

Figure 4 PCA visualization of the NSL-KDD dataset before and after applying our method.

reinforcement learning module (no DDPG) caused a slight decline in accuracy, suggesting that while the DDPG helps optimize the strategy, traditional feature extraction methods still allow for reasonable performance. Overall, the results demonstrate that the autoencoder and convolutional layers are key to the model's effectiveness, while the DDPG module has a smaller impact on the final accuracy, further validating our approach.

5.3 PCA Visualization Before and After Applying Our Method

Figure 4 shows the PCA visualization results for the NSL-KDD dataset before and after applying our method. On the left, the data points are mixed,

with Normal and Attack instances overlapping, indicating that the original feature representation lacks clear separability. However, after applying our method (shown on the right), the Normal and Attack classes are clearly separated into distinct clusters. This demonstrates that our approach has learned a more effective feature representation, improving class discriminability. The Normal instances form a tight group, while the Attack instances are isolated, highlighting the impact of deep convolutional encoding and reinforcement learning (DDPG) in refining the feature space. This improved separation reduces class overlap and enhances classification accuracy, showcasing the effectiveness of our method in transforming raw network traffic data into a more suitable feature representation for intrusion detection.

6 Conclusions

This study presents a novel intrusion detection model that combines the DACA and reinforcement learning to address the limitations of traditional methods. Our model demonstrates superior performance on two benchmark datasets, NSL-KDD and UNSW-NB15, outperforming existing approaches in accuracy, precision, recall, and F1-score. Utilizing convolutional layers, reinforcement learning, and a deep autoencoder design, the framework efficiently detects both known and unknown intrusion types while minimizing incorrect alerts. Additionally, its adaptability to dynamic environments makes it highly effective for real-time intrusion detection applications.

However, the model has some limitations. Its training time and computational complexity are high, especially with large-scale datasets. While detection time per instance is reasonable, the training phase demands significant computational resources, limiting its suitability for real-time environments. Additionally, its performance depends on the quality and quantity of labeled training data. In cases of limited or imbalanced data, the model may struggle with generalization, reducing detection accuracy. This highlights the need for better techniques to handle imbalanced datasets or integrate semi-supervised learning. Future research could focus on optimizing the model's architecture to reduce computational overhead, possibly through lightweight convolutional autoencoders or transfer learning. Addressing imbalanced datasets through semi-supervised learning could also improve robustness in data-scarce situations.

In summary, the proposed model advances intrusion detection, offering high accuracy in detecting various attacks. By combining deep learning and reinforcement learning, it shows promise for future network security

applications. However, overcoming its current limitations is essential for large-scale real-time deployment, and this work lays the groundwork for further advancements in network security research.

Funding

Yibin Vocational & Technical College 2024 General Project of Institutional Research, "Research on Intrusion Detection of New Energy Vehicle Internet Connected System Based on LSTM-Attention", Project No.: 24ZRYB-01.

References

[1] M. S. Elsayed, N.-A. Le-Khac, S. Dev, and A. D. Jurcut, "Ddosnet: A deep-learning model for detecting network attacks," in *2020 IEEE 21st International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, 2020: IEEE, pp. 391–396.

[2] X. Han, Y. Liu, Z. Zhang, X. Lü, and Y. Li, "Sparse auto-encoder combined with kernel for network attack detection," *Computer Communications*, vol. 173, pp. 14–20, 2021.

[3] Y. K. Saheed, A. I. Abiodun, S. Misra, M. K. Holone, and R. Colomo-Palacios, "A machine learning-based intrusion detection for detecting internet of things network attacks," *Alexandria Engineering Journal*, vol. 61, no. 12, pp. 9395–9409, 2022.

[4] Y.-F. Hsu and M. Matsuoka, "A deep reinforcement learning approach for anomaly network intrusion detection system," in *2020 IEEE 9th international conference on cloud networking (CloudNet)*, 2020: IEEE, pp. 1–6.

[5] J. Wang, F. Li, and L. He, "A Unified Framework for Adversarial Patch Attacks against Visual 3D Object Detection in Autonomous Driving," *IEEE Transactions on Circuits and Systems for Video Technology*, 2025.

[6] J. Wang, F. Li, S. Lv, L. He, and C. Shen, "Physically Realizable Adversarial Creating Attack against Vision-based BEV Space 3D Object Detection," *IEEE Transactions on Image Processing*, 2025.

[7] M. Lopez-Martin, B. Carro, and A. Sanchez-Esguevillas, "Application of deep reinforcement learning to intrusion detection for supervised problems," *Expert Systems with Applications*, vol. 141, p. 112963, 2020.

[8] L. Zhang, J. Liu, Y. Wei, D. An, and X. Ning, "Self-supervised learning-based multi-source spectral fusion for fruit quality evaluation: A case

study in mango fruit ripeness prediction," *Information Fusion*, vol. 117, p. 102814, 2025.

[9] R. B. Ping and W. Z. Yue, "Strategic Focus, Tasks, and Pathways for Promoting China's Modernization through New Productive Forces," *Journal of Xi'an University of Finance and Economics*, vol. 1, pp. 3–11, 2024.

[10] K. Sethi, Y. V. Madhav, R. Kumar, and P. Bera, "Attention based multi-agent intrusion detection systems using reinforcement learning," *Journal of Information Security and Applications*, vol. 61, p. 102923, 2021.

[11] H. Zhang et al., "Cross-modal knowledge transfer for 3D point clouds via graph offset prediction," *Pattern Recognition*, vol. 162, p. 111351, 2025.

[12] G. Karatas, O. Demir, and O. K. Sahingoz, "Deep learning in intrusion detection systems," in *2018 international congress on big data, deep learning and fighting cyber terrorism (IBIGDELFT)*, 2018: IEEE, pp. 113–116.

[13] L. A. Nguyen, I. Miciæ, N.-T. Nguyen, and S. Stanimiroviæ, "Depth-bounded fuzzy bisimulation for fuzzy modal logic," *Cybernetics and Systems*, pp. 1–18, 2023.

[14] H. Alavizadeh, H. Alavizadeh, and J. Jang-Jaccard, "Deep Q-learning based reinforcement learning approach for network intrusion detection," *Computers*, vol. 11, no. 3, p. 41, 2022.

[15] H. Liu and B. Lang, "Machine learning and deep learning methods for intrusion detection systems: A survey," *applied sciences*, vol. 9, no. 20, p. 4396, 2019.

[16] R. Abdulhammed, H. Musafer, A. Alessa, M. Faezipour, and A. Abuzneid, "Features dimensionality reduction approaches for machine learning based network intrusion detection," *Electronics*, vol. 8, no. 3, p. 322, 2019.

[17] C. X. Jing and W. Qing, "The Logic and Pathways of New Productive Forces Driving High-Quality Development," *Journal of Xi'an University of Finance and Economics*, vol. 37, no. 1, pp. 12–20, 2024.

[18] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 1, p. e4150, 2021.

[19] A. Aruna Kumari, A. Bhagat, and S. Kumar Henge, "Classification of Diabetic Retinopathy Severity Using Deep Learning Techniques on Retinal Images," *Cybernetics and Systems*, pp. 1–25, 2024.

[20] M. Sannidhan, J. E. Martis, R. S. Nayak, S. K. Aithal, and K. Sudeepa, "Detection of antibiotic constituent in Aspergillus flavus using quantum convolutional neural network," *International Journal of E-Health and Medical Communications (IJEHMC)*, vol. 14, no. 1, pp. 1–26, 2023.

[21] E. Alhajjar, P. Maxwell, and N. Bastian, "Adversarial machine learning in network intrusion detection systems," *Expert Systems with Applications*, vol. 186, p. 115782, 2021.

[22] O. Almomani, M. A. Almaiah, A. Alsaaidah, S. Smadi, A. H. Mohammad, and A. Althunibat, "Machine learning classifiers for network intrusion detection system: comparative study," in *2021 International Conference on Information Technology (ICIT)*, 2021: IEEE, pp. 440–445.

[23] K. Zheng and Z. Li, "An Image-Text Matching Method for Multi-Modal Robots," *Journal of Organizational and End User Computing (JOEUC)*, vol. 36, no. 1, pp. 1–21, 2024.

[24] W. Han, T. Zhang, J. Khan, L. Wang, and C. Tu, "Going global in the digital era: how digital finance affects Chinese OFDI," *Journal of Organizational and End User Computing (JOEUC)*, vol. 36, no. 1, pp. 1–22, 2024.

[25] J. Alsamiri and K. Alsubhi, "Internet of things cyber attacks detection using machine learning," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 12, 2019.

[26] Z. K. Maseer, R. Yusof, N. Bahaman, S. A. Mostafa, and C. F. M. Foozy, "Benchmarking of machine learning for anomaly based intrusion detection systems in the CICIDS2017 dataset," *IEEE access*, vol. 9, pp. 22351–22370, 2021.

[27] C. Quan and B. Lu, "Enhancing Innovation Management and Venture Capital Evaluation via Advanced Deep Learning Techniques," *Journal of Organizational and End User Computing (JOEUC)*, vol. 36, no. 1, pp. 1–22, 2024.

[28] P. Li, X. Peng, C. Zhang, and T. Baležentis, "Financial Cycle With Text Information Embedding Based on LDA Measurement and Nowcasting," *Journal of Organizational and End User Computing (JOEUC)*, vol. 36, no. 1, pp. 1–25, 2024.

[29] A. O. Alzahrani and M. J. Alenazi, "Designing a network intrusion detection system based on machine learning for software defined networks," *Future Internet*, vol. 13, no. 5, p. 111, 2021.

[30] Y. Xin et al., "Machine learning and deep learning methods for cybersecurity," *Ieee access*, vol. 6, pp. 35365–35381, 2018.

[31] J. Huang, X. Yu, D. An, X. Ning, J. Liu, and P. Tiwari, "Uniformity and deformation: A benchmark for multi-fish real-time tracking in the farming," *Expert Systems with Applications*, vol. 264, p. 125653, 2025.
[32] M. Farooq, R. A. Khan, and S. Z. Zahoor, "Q-learning and deep Q networks for securing IoT networks, challenges, and solution," in *Cognitive Machine Intelligence*: CRC Press, 2024, pp. 158–175.
[33] T. Lyu et al., "Optimized CNNs for Rapid 3D Point Cloud Object Recognition," *arXiv preprint arXiv:2412.02855*, 2024.
[34] M. Injadat, A. Moubayed, A. B. Nassif, and A. Shami, "Multi-stage optimized machine learning framework for network intrusion detection," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1803–1816, 2020.
[35] L. Zhao, B. Deng, L. Wu, C. Liu, M. Guo, and Y. Guo, "Deep Reinforcement Learning for Adaptive Stock Trading: Tackling Inconsistent Information and Dynamic Decision Environments," *Journal of Organizational and End User Computing (JOEUC)*, vol. 36, no. 1, pp. 1–27, 2024.
[36] Y. Wu, D. Wei, and J. Feng, "Network attacks detection methods based on deep learning techniques: a survey," *Security and Communication Networks*, vol. 2020, no. 1, p. 8872923, 2020.
[37] K. A. Taher, B. M. Y. Jisan, and M. M. Rahman, "Network intrusion detection using supervised machine learning technique with feature selection," in *2019 International conference on robotics, electrical and signal processing techniques (ICREST)*, 2019: IEEE, pp. 643–646.
[38] V. Praveena et al., "Optimal deep reinforcement learning for intrusion detection in UAVs," *Computers, Materials & Continua*, vol. 70, no. 2, pp. 2639–2653, 2022.
[39] P. P. Shinde and S. Shah, "A review of machine learning and deep learning applications," in *2018 Fourth international conference on computing communication control and automation (ICCUBEA)*, 2018: IEEE, pp. 1–6.
[40] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, pp. 1–22, 2019.
[41] H. Pang et al., "Electronic Health Records-Based Data-Driven Diabetes Knowledge Unveiling and Risk Prognosis," *arXiv preprint arXiv:2412.03961*, 2024.
[42] A. Kodipalli, S. L. Fernandes, S. K. Dasar, and T. Ismail, "Computational framework of inverted fuzzy C-means and quantum convolutional

neural network towards accurate detection of ovarian tumors," *International Journal of E-Health and Medical Communications (IJEHMC)*, vol. 14, no. 1, pp. 1–16, 2023.

[43] N. Sultana, N. Chilamkurti, W. Peng, and R. Alhadad, "Survey on SDN based network intrusion detection system using machine learning approaches," *Peer-to-Peer Networking and Applications*, vol. 12, no. 2, pp. 493–501, 2019.

[44] A. Jamalipour and S. Murali, "A taxonomy of machine-learning-based intrusion detection systems for the internet of things: A survey," *IEEE Internet of Things Journal*, vol. 9, no. 12, pp. 9444–9466, 2021.

[45] Y. Zeng, M. Qiu, D. Zhu, Z. Xue, J. Xiong, and M. Liu, "DeepVCM: A deep learning based intrusion detection method in VANET," in *2019 IEEE 5th intl conference on big data security on cloud (BigDataSecurity), IEEE intl conference on high performance and smart computing,(HPSC) and IEEE intl conference on intelligent data and security (IDS)*, 2019: IEEE, pp. 288–293.

[46] S. K. Gupta, M. Tripathi, and J. Grover, "Hybrid optimization and deep learning based intrusion detection system," *Computers and Electrical Engineering*, vol. 100, p. 107876, 2022.

[47] 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*, 2009: Ieee, pp. 1–6.

[48] 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)*, 2015: IEEE, pp. 1–6.

[49] A. N. Jaber and S. U. Rehman, "FCM–SVM based intrusion detection system for cloud computing environment," *Cluster Computing*, vol. 23, no. 4, pp. 3221–3231, 2020.

[50] D.-H. Tran and M. Park, "FN-GNN: A novel graph embedding approach for enhancing graph neural networks in network intrusion detection systems," *Applied Sciences*, vol. 14, no. 16, p. 6932, 2024.

[51] M. Arsalan, M. Mubeen, M. Bilal, and S. F. Abbasi, "1D-CNN-IDS: 1D CNN-based Intrusion Detection System for IIoT," in *2024 29th International Conference on Automation and Computing (ICAC)*, 2024: IEEE, pp. 1–4.

[52] C. Khammassi and S. Krichen, "A NSGA2-LR wrapper approach for feature selection in network intrusion detection," *Computer Networks*, vol. 172, p. 107183, 2020.
[53] Z. Sun, A. M. Teixeira, and S. Toor, "GNN-IDS: Graph Neural Network based Intrusion Detection System," in *Proceedings of the 19th International Conference on Availability, Reliability and Security*, 2024, pp. 1–12.

Biographies

Yuqin Dai was born in Sichuan, China, in 1994. From 2012 to 2016, she studied at Southwest University of Science and Technology and received her Bachelor's degree in 2016. From 2016 to 2019, she continued her studies at the same university and received her Master's degree in 2019. Currently, she works in Yibin Vocational & Technical College. She has published a total of four papers. Her research interests include cyber security, artificial intelligence, and big data.

Xinjie Qian was born in Anhui, China, in 1981. He is an associate professor. He studied at Sichuan University from 2000 to 2004 and received his Bachelor's degree in 2004. From 2008 to 2011, he continued his studies at

Sichuan University and obtained his Master of Engineering degree in 2011. He worked at Yibin Vocational & Technical College from 2004 to 2024. Since January 2025, he has been working at Yibin Industry Polytechnic College. He has published over 30 papers. His research interests include software development and big data.

Chunmei Yang was born in Sichuan, China, in 1992. She studied at Chongqing University of Arts and Sciences from 2012 to 2016 and received her Bachelor's degree in 2016. From 2012 to 2016, she worked at Xujia Middle School. Since 2019, she has been working at Changning Vocational and Technical School. Her research interests include software development.