LLM-Based Immune Detection Method for Unknown Network Attacks in ICS Under Few-Shot Conditions

Hao Wu[1], Jiangchuan Chen[1,✉], Wengang Ma[1], Ping He[2], Xiaolong Lan[1]

Tao Li[1] and Junjiang He[1]

[1] School of Cyber Science and Engineering, Sichuan University, Chengdu, 610065, China
[2] Information Technology Department at Sichuan Chuangang Gas Co., Ltd.
chenjiangchuan@stu.scu.edu.cn

Abstract. The rapidly evolving landscape of unknown network attacks has significantly expanded the range of cyber threats. However, existing intrusion detection systems (IDS) primarily rely on large amounts of known attack samples for model training and can only effectively detect known network attacks, particularly in industrial control system (ICS) environments, where obtaining attack samples is extremely difficult. In this paper, inspired by artificial immune systems (AIS) and large language models (LLM), we propose an LLM-based immune detection method for identifying unknown network attacks in ICS under few-shot conditions. The artificial immune system, as a biologically inspired intelligent algorithm, inherently possesses the ability to identify unknown threats. Meanwhile, LLM, with its strong reasoning ability, can deeply explore the latent spatial feature information even with limited train samples. Specifically, we first map network attack data to the antigen space of the artificial immune system. Then, we design a specialized prompt template to guide the LLM in learning and analyzing the spatial distribution features of nonself antigens, thereby capturing the latent space feature distribution information. Finally, we generate immune space detectors under the guidance of LLM and activate them through tolerance mechanisms. Extensive experiments on multiple datasets demonstrate that our method exhibits superior performance in detecting both known and unknown cyberattacks, significantly outperforming current mainstream IDS research achievements.

Keywords: Intrusion Detection System, Large Language Model, Artificial Immune System, Unknown Cyber Attacks

1 INTRODUCTION

The rapid emergence of advanced technologies such as IoT and generative AI has driven unprecedented developments in various communication, computer systems, and networks. However, this technological advancement has been accompanied by an exponential growth in cybersecurity threats, posing severe challenges to critical industrial infrastructure. Daily attack volumes have intensified significantly, with AV-Test

✉*Corresponding author*

reporting 450,000 new incidents daily [1], and Kaspersky documenting 828,000 daily emerging technology-based attacks in 2024, representing a two-fold increase since 2022 [2]. Intrusion detection systems (IDS) have become central to industrial control system (ICS) security research. Current IDS solutions, primarily reliant on known attack signatures, demonstrate inadequate capabilities against novel threats [3][4]. Kaspersky ICS CERT confirms this vulnerability, reporting that only 23.5% of ICS intrusion detection systems successfully mitigated threats in Q2 2024[5].

Signature-based IDS, widely deployed in industry, has demonstrated efficient identification of network attacks [6]. However, these systems essentially employ closed-set inference mechanisms, and their detection capabilities entirely depend on the attack samples included in the train set. With limited training data, they fail to extract sufficient attack features, compromising their ability to identify threats [7]. While anomaly-based approaches can detect anomalies by modeling normal behavior, they typically rely on hyperparameter models utilizing mathematical constructs such as probability density functions and feature space partitioning [8]. Under few-shot conditions, hyperparameter optimization becomes challenging, leading to unstable model performance [9]. In real industrial environments, improving detection capabilities will inevitably lead to increased false positive rates, which is generally undesirable.

To address the challenges, potential solutions lie in artificial immune systems and large language models. Artificial immune systems, as a biologically inspired intelligent learning algorithm, inherently recognize unknown threats [10]. Meanwhile, LLMs demonstrate strong reasoning abilities and can deeply explore feature information even under data scarcity conditions [11]. In 2023, Huang et al. enhanced unknown detection by integrating artificial immune systems with differential evolution theory [12]. In 2024, Bai et al. demonstrated superior detection for unknown cyber attacks with their approach combining LLMs and synchronous attention mechanisms [13].

Based on this foundation, we apply large language models to address the challenge of unknown attack detection under few-shot conditions. First, we design specialized prompt templates, utilizing LLM to fit limited data and deeply explore the feature distribution in the sample space. Then, we generate immune detectors under the guidance of the LLM and activate them through tolerance computation. Finally, we detect network attacks in the immune space using immune detectors, achieving both known and unknown attack detection. The main contributions of this study are as follows:

• We propose a prompt-based LLM feature fitting method. Unlike traditional approaches that rely on large-scale train data, our method achieves deep spatial distribution feature extraction under few-shot conditions.

• We develop an LLM-based immune detector generation method. In contrast to conventional random detector generation, our approach achieves comprehensive coverage of unknown attack sample spaces with limited train samples.

• We conduct extensive experiments on two datasets to validate the effectiveness of our proposed method. The experimental results demonstrate superior performance in identifying both known and unknown attacks compared to traditional methods.

The paper is organized as follows. Section 2 reviews recent advances in unknown attack detection systems and LLM-based data augmentation. Section 3 details our

proposed method and Section 4 evaluates its performance through extensive experiments. Finally, Section 5 concludes the paper.

2 RELATED WORK

In this section, we review recent progress in intrusion detection systems for unknown network attacks and the application of large language models for data generation.

2.1 IDS for Unknown Network Attacks

Intrusion detection systems for unknown network attacks are mainly classified into machine learning-based, deep learning-based, and emerging artificial immune system-based approaches.

In 2021, Liu et al. developed a hybrid model combining signature-based and artificial immune approaches with a two-round recognition mechanism, effectively overcoming limitations of traditional methods against complex unknown attacks [14]. In 2022, Aoudni et al. introduced HMM_TDL, integrating Hidden Markov Models with transudative deep learning for zero-day attack detection in cloud environments through a three-stage detection mechanism [15]. In 2023, Nguyen et al. proposed a hybrid detection framework combining Soft-Ordered Convolutional Neural Networks (SOCNN) with Local Outlier Factor (LOF) and isolation-based Nearest Neighbor Ensemble (iNNE), effectively addressing unknown DoS/DDoS attack detection in IoT environments [16]. Also in 2023, Yang et al. introduced MDGWO-NSA, a novel framework with adaptive regulation capabilities that combines unsupervised clustering-based heuristic dimensionality reduction, hybrid-partitioned negative selection algorithm (NSA), and improved grey wolf optimizer, significantly enhancing detection of unknown network attacks[17]. In 2024, Li et al. proposed HAD-IDS, integrating NN-LSTM with GAN to establish behavioral baselines, providing an effective framework for large-scale unknown network attack detection in IoT [18].

2.2 LLM-based Data Generation

Large language models exhibit two principal paradigms in data generation: generating semantically rich textual content and producing strictly formatted structured data including tabular and time-series information.

In 2023, Kholgh et al. proposed PAC-GPT, a reliable network data generation framework based on GPT-3, addressing the scarcity of real datasets in network security through Flow Generator and Packet Generator modules that capture network packet sequence patterns and generate individual packets [19]. In 2024, Zhou et al. introduced a universal time-series data generation method for edge intelligence, achieving flexible control over generation results through a self-trained fine-tuned model with a two-stage generation process incorporating abstract and detailed guiding signals [20]. In 2025, Banday et al. developed a context-enhanced LLM tabular data generation method that

addresses insufficient semantic context in feature names by combining three prompting approaches: expert-guided, LLM-guided, and novel mapping [21].

3 PROPOSED METHOD

To address the limitation of traditional methods in detecting unknown attacks with small samples, we design an LLM-based immune detection method for unknown network attacks (the framework of the system shown in **Fig. 1**). This method consists of three key components: antigen presentation, LLM-based few-shot spatial feature fitting and LLM-based immune detector generation.

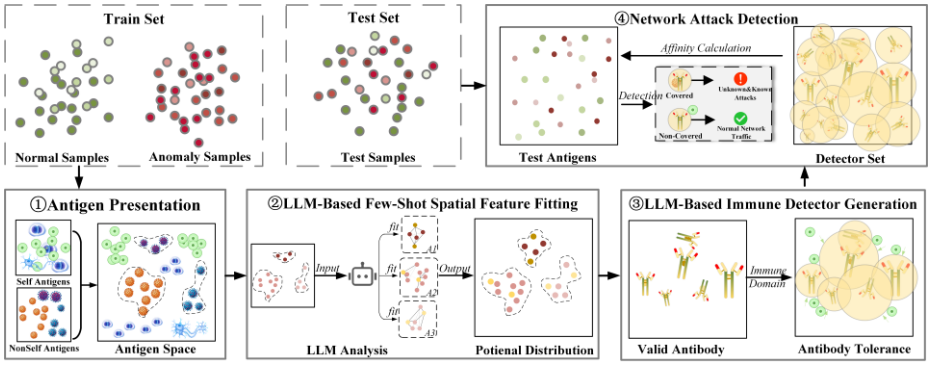


Fig. 1. The framework of proposed method

3.1 Antigen Presentation

Industrial control network data, represented as a point in geometric space, is denoted as an antigen in artificial immune system. It can be divided into self train set, nonself train set, self test set, and nonself test set. For the training data:

$$\begin{cases} Self_{train} : ag_{self}^{(1)}, ag_{self}^{(2)}, ..., ag_{self}^{(|Self_{train}|)} \\ Nonself_{train} : ag_{nonself}^{(1)}, ag_{nonself}^{(2)}, ..., ag_{nonself}^{(|N_{train}|)} \end{cases} \tag{1}$$

Each self antigen $ag_{self}^{(i)}$ has k_d dimensions, expressed as:

$$ag_{self}^{(i)} = [ag_{self}^{(i,1)}, ag_{self}^{(i,2)}, ..., ag_{self}^{(i,k_d)}]^T, \quad i = 1,2,...,|Self_{train}| \tag{2}$$

Similarly, each nonself antigen $ag_{nonself}^{(j)}$ is represented as:

$$ag_{nonself}^{(j)} = [ag_{nonself}^{(j,1)}, ag_{nonself}^{(j,2)}, ..., ag_{nonself}^{(j,k_d)}]^T, \quad j = 1,2,...,|Nonself_{train}| \tag{3}$$

Where $|Self_{train}|$ and $|Nonself_{train}|$ represent the cardinality of nonself train set and self train set respectively. Additionally, the self and nonself sets must be disjoint.

Similarly, the test set can be defined as:

$$\begin{cases} Self_{test} : ag_{self}^{(1)}, ag_{self}^{(2)}, ..., ag_{self}^{(|Self_{test}|)} \\ Nonself_{test} : ag_{nonself}^{(1)}, ag_{nonself}^{(2)}, ..., ag_{nonself}^{(|Nonself_{test}|)} \end{cases} \tag{4}$$

To address dimensional imbalances, we normalize all antigen features using min-max scaling. For both self antigens $ag_{self}^{(i)}$ and nonself antigens $ag_{nonself}^{(i)}$ in dimension d, namely $ag^{(i,d)}$, we apply:

$$\begin{cases} \tilde{ag}^{(i,d)} = \dfrac{ag^{(i,d)} - ag_{min}^{(d)}}{ag_{max}^{(d)} - ag_{min}^{(d)}}, \text{ if } ag_{max}^{(d)} \neq ag_{min}^{(d)} \\ \tilde{ag}^{(i,d)} = 0, \text{ if } ag_{max}^{(d)} = ag_{min}^{(d)} \end{cases} \tag{5}$$

Where $ag_{min}^{(d)} = \min\{ag_{self,min}^{(d)}, ag_{nonself,min}^{(d)}\}$, $ag_{max}^{(d)} = \max\{ag_{self,max}^{(d)}, ag_{nonself,max}^{(d)}\}$ represent the global minimum and maximum values across all training samples.

3.2 LLM-Based Few-Shot Spatial Feature Fitting

After the antigen presentation, each nonself and self antigen has been mapped as a point in (0,1) space. Since feature values strongly influence the results of classification, different categories of nonself and self antigens exhibit distinct clustering patterns in high-dimensional space. This naturally formed distribution characteristic provides an important foundation for subsequent detector generation based on spatial distribution.

We entrust this complex task to large language models that excel at spatial reasoning and distribution analysis. Through carefully designed prompts, we guide LLM to leverage the massive spatial distribution modeling formulas learned from pretraining to deeply understand the spatial distribution characteristics of nonself antigens. The prompt design template is shown in **Fig. 2**.

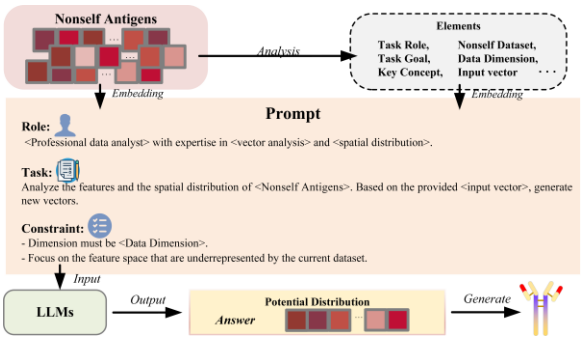


Fig. 2. Prompt design template.

In this template, by clarifying the task objective, we help the model understand this is a space analysis and data generation task, effectively reducing the possibility of model misinterpretation and hallucination. We emphasize the concepts of "spatial distribution" and "feature space" multiple times in the prompt, and explicitly require that newly generated vectors cover unknown spaces as much as possible - areas not covered by the original nonself antigens. This strategy guides the model to perform more complex reasoning and deeply understand the potential geometric properties of the dataset. Simultaneously, we assign a professional role to the model, helping to activate its domain-specific knowledge - utilizing high-quality spatial distribution modeling formulas, thereby significantly improving the professionalism and applicability of the model output. We also ensure output consistency while optimizing model response efficiency by strictly defining input and output data formats and clear constraints.

By inputting carefully designed prompts, nonself antigen dataset, and specific nonself antigen vectors from the dataset to the large model, we guide the LLM to fit the few-shot feature space based on the distribution characteristics of known nonself antigens. In this process, the large language model first regresses the target nonself antigen vector into the distribution space of the entire dataset, precisely analyzing the distribution characteristics of points around it. Subsequently, the model fits antigen features based on relevant evolutionary algorithms learned during pretraining and fine-tuning phases. After fitting antigen features, we obtain feedback from the LLM. These feedback data neither disrupt the original distribution patterns nor highly conform to the feature expression of nonself antigens themselves, which we call large language model antigens ag_{llm}. They satisfy:

$$ag_{llm} = G(\theta, t; Input_{llm}) \tag{6}$$

In other words, we view the LLM as a generator G, where θ represents the parameters of the LLM itself, optimized through pretraining and prompt guidance. The temperature parameter t is typically used to adjust the stability of generated data. A larger t results in more diverse generated vectors that tend to explore unknown spaces, while a smaller t causes generated vectors to be closer to the input distribution, tending to utilize existing patterns. The input to the LLM $Input_{llm}$ satisfies:

$$Input_{llm} = Prompt(E(ag_{input}), R, T, C) \tag{7}$$

We extract a portion from the preprocessed nonself antigens as input antigens. These vector-form input antigens are serialized and encoded into text form, then combined with contextual role R, task description T, and constraint conditions C. Through template embedding, all information is concatenated into a complete prompt and converted into an input format processable by the LLM. Through an iterative process of prompting and feedback, we can guide the LLM to deeply fit the few-shot space feature distribution.

3.3 LLM-Based Immune Detector Generation

Through the prompt engineering in 3.2, the LLM can now fit the few-shot feature space distribution. At this point, by initializing candidate detector seeds and inputting them into the LLM, effective detectors can be generated in the immune space under the guidance of the LLM. The LLM-based detector generation method is as follows:

$$\vec{a} = LLM([x_i = random(0,1) \mid i \in \{1,...,k_d\}]) \tag{8}$$

Where $[x_i = random(0,1) \mid i \in \{1,...,n_d\}]$ represents the initialized detector seed features, and LLM represents the large language model.

In artificial immunity, the key to detector activation lies in tolerance with self data, which is the calculation of affinity. When candidate detectors cover known self data, they will be eliminated or discarded. When the affinity between candidate detectors and self data meets the threshold, the detector will be activated as a mature detector. The affinity calculation method is as follows:

$$affinity(\vec{d}, Self_{train}) = \sqrt{\sum_{i=1}^{n_d} (x_i - s_i)^2} \tag{9}$$

Affinity represents the degree of spatial distribution difference between candidate detectors and self data. When $affinity(\vec{d}, Self_{train}) \le r_s$, it indicates that the current candidate detector $\vec{d}$ falls in the self region, and it will be eliminated, requiring regeneration of a candidate detector. When $affinity(\vec{d}, Self_{train}) > r_s$, it indicates that the candidate detector $\vec{d}$ meets the minimum affinity requirement. The immune capability of a detector is determined by its immune domain, which is the detector radius r_d. Each different detector, under different affinities, possesses different immune domains r_d.

The determination of the immune domain is jointly decided by self data. To avoid the influence of affinity calculation with a single self data on the detector's immune domain, we evaluate self density through affinity threshold and further calculate the immune domain r_d of candidate detectors. First, the neighboring self density of candidate detector $\vec{d}$ is calculated as follows:

$$\rho_{th} = \frac{k}{\sum_{i=1}^{k} affinity(\vec{a}, s^{nearest})} \tag{10}$$

Where k represents the number of selected self data neighboring the candidate detector, and $Self_i^{(nearest)}$ represents the self data closest to the current candidate detector, satisfying:

$$Self^{nearest} = \{s_j \mid affinity(\vec{d}, s_j) \in \{affinity_1,...,affinity_k\}, s_j \in Self_{train}, j \neq i\} \tag{11}$$

Distance is a measure of spatial difference between data points. Smaller distances indicate closer points and higher concentration. Using the reciprocal of distance converts the relationship between distance and density, so that small distances correspond to high density, and large distances correspond to low density. For example, if neighboring points of a candidate detector $\vec{d}$ are all close to it, then the sum of distances between these neighboring self points S would be small, and the reciprocal of this sum would be large, indicating high density around this point. This conforms to the intuitive understanding of density: areas with more concentrated points have higher density.

When density is higher, there are more self data near the candidate detector. To prevent false positives of the candidate detector against self data, the radius of the candidate detector should be smaller; conversely, it should be larger. The calculation method for the immune domain r_d of candidate detector $\vec{d}$ is as follows:

$$r_i = \frac{\alpha}{\rho_{th}}[(1-\beta)\times\overline{affinity} + \beta \times affinity_{min}] \tag{12}$$

Where α is a constant used to adjust the size of the radius. $\overline{affinity}$ is the average affinity of the current k points, and $affinity_{min}$ is the minimum affinity. β is a proportion coefficient in the range [0,1], used to balance the contribution of average affinity and nearest affinity in radius calculation.

Through this formula design, we can comprehensively consider multiple aspects of information such as neighboring density of data points, average distance, and nearest self distance. This calculates a radius value that can reflect both the overall density characteristics around the data point and accommodate the local relationship with similar points. It achieves activation of immune detectors within a reasonable range, forcing them to cover unknown immune space.

3.4 Network Attack Detection

In the detection process, each test network sample is represented as an k_d -dimensional antigen vector $ag_{test} = (ag_{test}^{(1)}, ag_{test}^{(2)}, ..., ag_{test}^{(n_d)})$, where $ag_{test}^{(d)}$ denotes the feature value in dimension d. To maintain consistent scaling with training data, we normalize test samples using:

In the actual detection process, we also represent each test network sample as a d dimension antigen vector, where represents the feature value of the sample in the dimension ($d = 1,2,...,k_d$). To ensure that test samples are compared with train samples on the same scale, we also need to standardize the test samples. The standardized test sample $\tilde{ag}_{test}$ is calculated according to the following formula:

$$\begin{cases} \tilde{ag}_{test}^{(d)} = \dfrac{ag_{test}^{(d)} - ag_{min}^{(d)}}{ag_{max}^{(d)} - ag_{min}^{(d)}}, \text{ if } ag_{max}^{(d)} \neq ag_{min}^{(d)} \\ \tilde{ag}_{test}^{(d)} = 0, \text{ if } ag_{max}^{(d)} = ag_{min}^{(d)} \end{cases} \tag{13}$$

Where $ag_{min}^{(d)} = \min\{ag_{1,test}^{(d)}, ag_{n_d,test}^{(d)}\}$ and $ag_{max}^{(d)} = \max\{ag_{1,test}^{(d)}, ag_{n_d,test}^{(d)}\}$ represent the minimum and maximum values across all test samples in dimension d.

For each test antigen ag_{test}, the system will check whether it falls within the coverage area of any detector. As long as there exists a detector $\mathcal{D}_{llm}^{(t)}(ag_{llm}^{(t)}, r_{llm}^{(t)}) \in \mathcal{D}_{llm}(t = 1,2,...,|\mathcal{D}_{llm}|)$ such that

$$\| \tilde{ag}_{test} - ag_{llm}^{(t)} \|_2 \leq r_{llm}^{(t)} \tag{14}$$

then ag_{test} is identified as a nonself antigen (i.e., an attack sample). This inequality indicates the Euclidean distance between ag_{test} and detector center ag_{llm} is less than or equal to the detector's radius r_{llm}, meaning ag_{test} falls in the detector's coverage. Conversely, when ag_{llm} is not covered by any detector, it is classified as a normal sample.

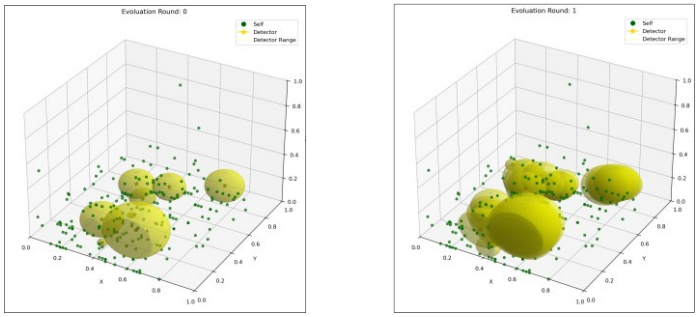
4 EXPERIMENTAL RESULTS

In this section, we conduct extensive experiments to evaluate the performance of our method. We first perform validation experiments for scheme effectiveness, followed by performance comparison experiments between different intrusion detection systems. We detail the experimental setup, including dataset description, dataset partitioning, baseline studies, and evaluation metrics. Subsequently, we comprehensively compare its performance with various application studies, covering intrusion detection systems based on machine learning, deep learning, and artificial immune systems.

4.1 Validation Experiments

We first use Haberman dataset to validate that vectors generated by the LLM are meaningful and valuable. The Haberman dataset is a standard dataset widely used for artificial immune system performance verification. It contains 306 data instances, 3 features, and 1 target variable, facilitating visual analysis. Using the proposed method in 3, we first mapped the dataset to three-dimensional space and divided 225 self and 81 nonself antigens accordingly. To simulate an environment with scarce attack features, we randomly extracted only 8 vectors from the nonself antigen set as a few-shot train set, with the remaining self samples used for antibody tolerance train.

Using locally deployed large language models, we performed transformation operations on these eight nonself antigen vectors. Through prompt guidance, the model generated 8 new nonself antigen vectors for each original nonself antigen vector. We conducted three rounds of generation, creating antibody detectors using the corresponding generated antigens in each round. This progressive generation process not only expanded the spatial information content of the original few-shot samples but also enabled the large language model to continue exploring unknown areas in the feature space based on previously generated vectors. The experimental results are shown in **Fig. 3**.



(a): Round 0 (b): Round 1

(c): Round 2 (d): Round 3

Fig. 3. Experimental Results on the Haberman Dataset

As we can see, the newly generated detectors fully meet the three requirements for antibodies in artificial immune systems: (1) Cover the antigen space as widely as possible, achieving effective monitoring of both known and unknown regions; (2) Cover all nonself antigens as comprehensively as possible, ensuring high detection rates; (3) Avoid covering self antigens as much as possible, maintaining low false positive rates. Based on this method, new nonself antigens can be generated without quantity limitations, effectively addressing key technical bottlenecks in traditional artificial immune algorithm-based evolutionary algorithms, such as antigen gaps and premature antibody convergence.

4.2 Comparative Experiments Settings

Dataset Description: Comparative experiments will be conducted on two widely used network intrusion detection datasets: UNSW_NB15 [22] and CICIDS-2018 [23]. These datasets are selected as benchmarks to comprehensively evaluate the performance of our method against other IDS in different application scenarios.

UNSW_NB15: It is a network security dataset designed to reflect the complexity of contemporary network traffic and attack scenarios. It effectively addresses inherent problems in NSL-KDD and is specifically designed to evaluate the performance of new

network intrusion detection systems when facing modern network threats. Each record in this dataset contains 47 features and is labeled as normal traffic or attack traffic. Attacks cover nine types: Analysis(A), Backdoors(B), DoS(D), Exploits(E), Fuzzers(F), Generic(G), Reconnaissance(R), Shellcode(S), and Worms(W).

CICIDS-2018: It is a modern large-scale dataset with a massive scale containing approximately 10 million records. Each record in the dataset includes 82 features, containing complete network traffic information and detailed traffic statistical features, labeled as normal data or abnormal data. The abnormal data include various actual network attack scenarios, which we categorize into six major attack types: Bot(B_1), Brute Force(B_2), DoS(D_1), DDoS(D_2), Infiltration(I), and SQL injection(S).

Dataset Configuration: We constructed 9 and 6 different experimental setups for UNSW_NB15 and CICIDS-2018 datasets respectively to evaluate the performance of our method. These experimental setups are specifically designed to simulate two key features in industrial control systems: limited attack samples and massive unknown network attacks. In each setup, we adopted the following strategies: (1) Few-shot simulation: Only extracting a small number of attack samples in the train set; (2) Unknown attack simulation: Purposely removing a specific type of attack samples from the train set; (3) Balanced sample design: Extracting normal samples in equal quantity to attack samples. We denote the corresponding type of dataset as the Missing train set, abbreviated as Miss_type. The specific setup schemes are shown in Table 4 and Table 5.

Table 1. Experimental Setup for UNSW_NB15 dataset.

Setting	Self	Nonself								
	Normal	A	B	D	E	F	G	R	S	W
Miss_A	1040	0	130	130	130	130	130	130	130	130
Miss_B	1040	130	0	130	130	130	130	130	130	130
Miss_D	1040	130	130	0	130	130	130	130	130	130
Miss_E	1040	130	130	130	0	130	130	130	130	130
Miss_F	1040	130	130	130	130	0	130	130	130	130
Miss_G	1040	130	130	130	130	130	0	130	130	130
Miss_R	1040	130	130	130	130	130	130	0	130	130
Miss_S	1040	130	130	130	130	130	130	130	0	130
Miss_W	1040	130	130	130	130	130	130	130	130	0

Baseline Studies: We compared the performance of ours with various widely applied intrusion detection methods. RF-IDS [24] and SVM-IDS [25] are intrusion detection systems based on classical machine learning, representing the most used technical approaches in current industrial practice [29]. Random Forest enhances classification through ensemble learning of multiple decision trees, while Support Vector Machine employs kernel functions to establish optimal hyperplanes in high-dimensional feature

spaces. These comparisons assess our method's practical advantages and deployment potential. CNN_LSTM-IDS [26] and GRU_LSTM-IDS [27] represent advanced intrus-

Table 2. Experimental Setup for CICIDS_2018 dataset.

Setting	Self	Nonself					
	Normal	B_1	B_2	D_1	D_2	I	S
Miss_B_1	1420	0	100	500	500	300	20
Miss_B_2	1620	300	0	500	500	300	20
Miss_D_1	1220	300	100	0	500	300	20
Miss_D_2	1220	300	100	500	0	300	20
Miss_I	1420	300	100	500	500	0	20
Miss_S	1700	300	100	500	500	300	0

ion detection technologies based on deep learning. CNN_LSTM combines convolutional and recurrent architectures to process both spatial and temporal traffic characteristics, while GRU_LSTM utilizes simplified recurrent structures that maintain temporal feature learning capabilities. These networks dynamically perceive potential unknown attacks through temporal relationship analysis. DGA-PSO-IDS [30] and V-Detector-IDS [29] are intrusion detection methods based on artificial immune system. DGA-PSO employs Particle Swarm Optimization to generate detectors that fill nonself antigen space gaps, while V-Detector implements variable radius mechanisms to improve nonself space coverage with demonstrated operational stability. These comparisons specifically validate our antigen generation effectiveness against optimization-based approaches and established artificial immune benchmarks. This comprehensive evaluation framework spans the full spectrum of detection technologies, providing robust performance benchmarks across multiple technical dimensions.

Evaluation Metrics: We apply a series of widely used evaluation indicators to measure system performance, including Unknown Detection Rate (UDR), Accuracy, Precision, Recall, F1 (weighted average of precision and recall rate) and False positive rate (FPR).

4.3 Comparative Experiments Results

Performance Comparison on UNSW-NB15 Dataset: Table 3 presents the performance comparison between ours and other baseline models on UNSW-NB15 dataset. This dataset features higher real-world complexity, effectively addressing issues of attack class imbalance and synthetic data lacking realistic network complexity. On this dataset, traditional methods like SVM and CNN_LSTM, show obvious limitations, such as high false positive rates and low detection rates. Our method, leveraging the powerful high-dimensional feature analysis capabilities and optimized detector generation strategies, generally outperforms other models across key metrics. In Miss_B and Miss_W, ours achieves higher accuracy and recall rates with only a slight precision loss

(approximately 2.3%), while maintaining superior F1 scores compared to other models, demonstrating ours' adaptability in more complex network environments. The model can achieve satisfactory detection accuracy while maintaining low false positive rates, effectively capturing both known and unknown network attacks.

Performance Comparison on CICIDS_2018 Dataset: Table 4 presents the performance comparison on CICIDS_2018 dataset. This dataset, with its massive data volume and high-dimensional features, creates a challenging environment for model computational and generalization ability. In this complex environment, traditional models show very limited learning effectiveness under few-shot conditions, almost losing their detection capabilities for both known and unknown network attacks. Models like VD and GRU_LSTM perform even worse than random guessing in detecting attack categories such as SQL injection, clearly highlighting these methods' sensitivity and limitations to few-shot conditions and high-dimensional features. In contrast, our method, leveraging the large language model's ability to explore potential network attack features in high-dimensional space almost without restriction, demonstrates low sensitivity to data dimensionality. Additionally, the model can generate high-quality detectors that meet the requirements of intrusion detection tasks, maintaining high accuracy and detection rates in this more rigorous network environment. Although ours shows a false positive rate (FPR) generally between 22-25%, higher than its performance on the UNSW_NB15 dataset, it still achieves significant improvements compared to the generally higher false positive rates of baseline models. It can be considered to achieve a satisfactory balance between high detection rates and low false positive rates, maintaining consistently excellent performance across all unknown attack categories, demonstrating unique advantages in handling high-dimensional complex cyberattacks.

Table 3. Performance Comparison between Ours and Baseline on UNSW-NB15 Dataset

Type	Metric	RF	SVM	CNN_LSTM	GRU_LSTM	VD	DGAPSO	Ours
Miss_A	UDR	91.43	90.99	88.18	91.43	90.40	91.43	**99.91**
	Acc	89.64	73.02	80.76	81.31	78.88	77.83	**90.48**
	Pre	90.06	81.06	76.23	77.73	87.18	87.48	**90.51**
	Recl	89.12	60.08	89.40	87.76	67.72	64.96	**90.44**
	F1	89.59	69.01	82.29	82.44	76.23	74.56	**90.48**
	FPR	9.84	14.04	27.88	25.14	9.96	**9.30**	9.48
Miss_B	UDR	94.51	85.94	97.60	98.80	90.05	90.74	**99.83**
	Acc	88.60	73.99	81.16	82.18	78.88	77.71	**89.87**
	Pre	**93.45**	76.08	80.38	79.02	87.18	86.20	90.93
	Rec	83.02	69.98	82.44	87.62	67.72	65.98	**88.58**
	F1	87.93	72.90	81.40	83.10	76.23	74.75	**89.74**
	FPR	**5.82**	22.00	20.12	23.26	9.96	10.56	8.84
Miss_D	UDR	93.84	96.57	83.27	93.62	75.35	71.56	**98.19**
	Acc	88.19	73.34	75.69	81.72	78.88	77.35	**90.24**

Type	Metric	RF	SVM	CNN_LSTM	GRU_LSTM	VD	DGAPSO	Ours
	Pre	88.46	77.92	74.75	78.62	87.18	85.83	**91.18**
	Rec	87.84	65.14	77.58	87.14	67.72	65.52	**89.10**
	F1	88.15	70.96	76.14	82.66	76.23	74.31	**90.13**
	FPR	11.46	18.46	26.20	23.70	9.96	10.82	**8.62**
Miss_E	UDR	64.58	47.14	84.11	66.91	26.82	29.44	**89.68**
	Acc	85.51	77.37	81.95	80.85	76.93	77.31	**90.00**
	Pre	87.37	77.66	80.64	80.13	86.03	85.57	**90.73**
	Rec	83.02	76.84	84.08	82.04	64.30	65.70	**89.10**
	F1	85.14	77.25	82.33	81.08	73.60	74.33	**89.91**
	FPR	12.00	22.10	20.18	20.34	10.44	11.08	**9.10**
Miss_F	UDR	32.51	28.19	40.02	55.43	42.43	38.73	**74.94**
	Acc	88.94	57.46	82.61	81.83	78.88	76.89	**89.87**
	Pre	91.01	64.94	81.05	78.30	87.18	84.47	**91.51**
	Rec	86.42	32.42	85.12	88.06	67.72	65.90	**87.90**
	F1	88.65	43.25	83.04	82.90	76.23	74.04	**89.67**
	FPR	8.54	17.50	19.90	24.40	9.96	12.12	**8.16**
Miss_G	UDR	46.74	42.02	41.20	72.27	91.57	97.15	**99.69**
	Acc	75.66	67.10	68.11	75.54	69.97	77.51	**91.27**
	Pre	77.53	69.05	68.16	74.03	70.66	86.07	**91.44**
	Rec	72.26	61.98	67.98	78.68	68.30	65.64	**91.06**
	F1	74.80	65.32	68.07	76.28	69.46	74.48	**91.25**
	FPR	20.94	27.78	31.76	27.60	28.36	10.62	**8.52**
Miss_R	UDR	90.73	90.33	54.29	83.38	24.97	46.02	**99.30**
	Acc	88.66	80.81	75.13	82.04	78.88	77.43	**91.25**
	Pre	88.97	83.01	79.12	80.50	87.18	86.24	**91.41**
	Rec	88.26	77.48	68.28	84.56	67.72	65.28	**91.06**
	F1	88.61	80.15	73.30	82.48	76.23	74.31	**91.23**
	FPR	10.94	15.86	18.02	20.48	9.96	10.42	**8.56**
Miss_S	UDR	92.59	78.31	80.16	90.48	48.68	44.71	**93.18**
	Acc	88.08	80.67	82.81	82.21	71.22	77.84	**92.01**
	Pre	85.90	82.37	80.10	78.68	74.61	87.60	**91.44**
	Rec	91.12	78.04	87.32	88.36	64.34	64.86	**92.70**
	F1	88.43	80.15	83.55	83.24	69.09	74.53	**92.06**
	FPR	14.96	16.70	21.70	23.94	21.90	9.18	**8.68**
Miss_W	UDR	86.36	40.91	86.36	95.45	22.73	13.64	**97.75**
	Acc	88.86	78.43	85.67	81.41	76.71	78.36	**90.66**
	Pre	**91.38**	78.76	85.24	77.86	85.95	87.79	91.29

Type	Metric	RF	SVM	CNN_LSTM	GRU_LSTM	VD	DGAPSO	Ours
	Rec	85.82	77.86	86.28	87.78	63.86	65.88	**89.90**
	F1	88.51	78.31	85.76	82.52	73.28	75.27	**90.59**
	FPR	**8.10**	21.00	14.94	24.96	10.44	9.16	8.58

Table 4. Performance Comparison between Ours and Baseline on CICIDS_2018 Datase

Type	Metric	RF	SVM	CNN_LSTM	GRU_LSTM	VD	DGAPSO	Ours
$Miss\_B_1$	UDR	47.68	48.30	50.12	0.02	0.03	49.85	**99.15**
	Acc	70.02	74.70	75.95	53.62	48.38	48.99	**77.25**
	Pre	68.31	78.77	**80.35**	66.54	42.29	48.24	77.77
	Rec	74.68	67.62	68.70	14.56	8.88	27.66	76.32
	F1	71.35	72.77	74.07	23.89	14.68	35.16	**77.04**
	FPR	34.64	18.22	16.80	**7.32**	12.12	29.68	21.82
$Miss\_B_2$	UDR	46.20	37.20	33.60	19.60	25.60	58.80	**97.69**
	Acc	73.50	62.94	50.71	34.04	46.43	49.22	**75.52**
	Pre	**76.83**	66.78	51.25	23.84	21.35	48.98	75.13
	Rec	67.30	51.50	29.08	14.54	2.66	37.62	**76.30**
	F1	71.75	58.15	37.11	18.06	4.73	42.56	**75.51**
	FPR	20.30	25.62	27.66	46.46	**9.80**	39.18	25.26
$Miss\_D_1$	UDR	56.25	24.17	0.03	0.00	23.89	99.76	**99.92**
	Acc	69.43	59.95	64.12	40.82	46.37	52.81	**77.07**
	Pre	69.52	62.01	64.53	38.99	38.62	54.53	**77.47**
	Rec	69.20	51.38	62.72	32.52	12.32	33.82	**76.34**
	F1	69.36	56.20	63.61	35.46	18.68	41.75	**76.90**
	FPR	30.34	31.48	34.48	50.88	**19.58**	28.20	22.20
$Miss\_D_2$	UDR	21.35	15.35	9.27	1.90	0.00	27.65	**41.48**
	Acc	54.65	54.81	49.20	38.84	48.38	54.62	**77.10**
	Pre	58.61	70.06	48.88	20.85	42.29	58.24	**77.60**
	Rec	31.66	16.80	34.78	7.98	8.88	32.66	**76.20**
	F1	41.11	27.10	40.64	11.54	14.68	41.85	**76.89**
	FPR	22.36	**7.18**	36.38	30.30	12.12	23.42	22.00
Miss_I	UDR	14.94	35.68	21.83	0.04	19.94	26.27	**79.70**
	Acc	75.52	67.88	67.64	50.18	37.31	54.55	**77.63**
	Pre	81.12	67.44	76.23	**87.50**	17.44	58.87	80.49
	Rec	66.52	69.14	51.26	0.42	6.80	30.20	**72.94**
	F1	73.10	68.28	61.30	0.84	9.79	39.92	**76.53**
	FPR	15.48	33.38	15.98	**0.06**	32.18	21.10	17.69
Miss_S	UDR	49.25	43.28	64.18	73.13	0.00	56.72	**98.83**

Type	Metric	RF	SVM	CNN_LSTM	GRU_LSTM	VD	DGAPSO	Ours
	Acc	59.76	61.81	62.97	36.32	45.04	48.78	**75.06**
	Pre	60.69	70.60	63.63	37.58	9.48	47.92	**74.41**
	Rec	55.42	50.76	60.56	41.40	1.16	28.14	**76.40**
	F1	57.93	59.06	62.06	39.40	2.07	35.46	**75.39**
	FPR	35.90	21.14	34.62	68.76	**11.08**	30.58	26.28

5 CONCLUSION

In this paper, we propose an LLM-based immune detection method for unknown network attacks in industrial control systems under few-shot conditions. First, we map cyberattack data to the antigen space of an artificial immune system, then design a specialized prompt template to guide the large language model in learning and analyzing the spatial distribution characteristics of non self antigen space. Finally, we generate immune detectors guided by LLM and activate the detectors through tolerance. Through extensive experiments, we demonstrate that our proposed method outperforms current mainstream methods on key metrics, including intrusion detection systems based on machine learning, deep learning, and artificial immune systems.

Future work will focus on processing the generated detectors, including strategies for detector coordinate mutation, promoting tolerance between detectors, and reducing the number of excessively redundant detectors, thereby reducing false positives caused by too many detectors. Additionally, future work will also attempt to optimize the large language model's strategy for generating detector vectors through fine-tuning, reinforced retrieval, and other methods to enhance understanding of the feature space.

Acknowledgments. This work was supported in part by the National Natural Science Foundation of China (No. 62402300); in part by the Sichuan Provincial Science and Technology Department regional innovation cooperation key project (Grant No.2025YFHZ0265); in part by the Youth Science Foundation of Sichuan,(No.2025ZNSFSC1474); in part by the China Postdoctoral Science Foundation (No.2024M752211); in part by the key laboratory of data protection and intelligent management ministry of education (SCUSACXYD202301).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. AV-TEST: Malware Homepage. https://www.av-test.org/en/statistics/malware/, last accessed 2025/3/7
2. Kaspersky: Kaspersky Security Bulletin 2024. Statistics. https://securelist.com/ksb-2024-statistics/114795/, last accessed 2025/3/7
3. Umer, M.A., Junejo, K.N., Jilani, M.T., Mathur, A.P.: Machine Learning for intrusion detection in industrial control systems: Applications, challenges, and recommendations. International Journal of Critical Infrastructure Protection 38, 100561 (2022)

4. Raman, M.R.G., Ahmed, C.M., Mathur, A.: Machine learning for intrusion detection in industrial control systems: challenges and lessons from experimental evaluation. Cybersecurity 4, 27 (2021)
5. Kaspersky: Threat landscape for industrial automation systems. Q2 2024. https://ics-cert.kaspersky.com/publications/reports/2024/09/26/threat-landscape-for-industrial-automation-systems-q2-2024, last accessed 2025/2/21
6. Dilara, G., Tulay, Y., Angelo, G., Fabio, S.: A comprehensive survey of databases and deep learning methods for cybersecurity and intrusion detection systems. IEEE Systems Journal 15(2), 1717–1731 (2021)
7. Rudd, E.M., Rozsa, A., Günther, M., Boult, T.E.: A survey of stealth malware: attacks, mitigation measures, and steps toward autonomous open world solutions. IEEE Communications Surveys & Tutorials 19(2), 1145–1172 (2016)
8. Vu, L., Cao, V.L., Nguyen, Q.U., Nguyen, D.N., Hoang, D.T., Dutkiewicz, E.: Learning latent representation for IoT anomaly detection. IEEE Transactions on Cybernetics 52(5), 3769–3782 (2022)
9. Cao, V.L., Nicolau, M., McDermott, J.: Learning neural representations for network anomaly detection. IEEE Transactions on Cybernetics 49(8), 3074–3087 (2019)
10. Saurabh, P., Verma, B.: Negative selection in anomaly detection—A survey. Computer Science Review 48, 100557 (2023)
11. Huang, J., Chang, K.C.-C.: Towards Reasoning in Large Language Models: A Survey. In: Findings of the Association for Computational Linguistics: ACL 2023, pp. 1049–1065. Association for Computational Linguistics, Toronto (2023)
12. Huang, H., Li, T., Li, B., Wang, W., Sun, Y.: A Bidirectional Differential Evolution Based Unknown Cyberattack Detection System. IEEE Transactions on Evolutionary Computation, 1–1 (2024)
13. Bai, Y., Sun, M., Zhang, L., Wang, Y., Liu, S., Liu, Y., Tan, J., Yang, Y., Lv, C.: Enhancing Network Attack Detection Accuracy through the Integration of Large Language Models and Synchronized Attention Mechanism. Applied Sciences 14, 3829 (2024)
14. Liu, C., Zhang, Y.: An Intrusion Detection Model Combining Signature-Based Recognition and Two-Round Immune-Based Recognition. In: 17th International Conference on Computational Intelligence and Security (CIS), pp. 497–501. IEEE, Chengdu (2021)
15. Aoudni, Y., et al.: Cloud security based attack detection using transductive learning integrated with Hidden Markov Model. Pattern Recognition Letters 157, 16–26 (2022)
16. Nguyen, X.-H., Le, K.-H.: Robust detection of unknown DoS/DDoS attacks in IoT networks using a hybrid learning model. Internet of Things 23, 100851 (2023)
17. Yang, G., Wang, L., Yu, R., He, J., Zeng, B., Wu, T.: A Modified Gray Wolf Optimizer-Based Negative Selection Algorithm for Network Anomaly Detection. International Journal of Intelligent Systems 2023(1), 8980876 (2023)
18. Li, S., Cao, Y., Liu, S., Lai, Y., Zhu, Y., Ahmad, N.: HDA-IDS: A Hybrid DoS Attacks Intrusion Detection System for IoT by using semi-supervised CL-GAN. Expert Systems with Applications 238, 122198 (2024)
19. Kholgh, D.K., Kostakos, P.: PAC-GPT: A Novel Approach to Generating Synthetic Network Traffic With GPT-3. IEEE Access 11, 114936–114951 (2023)
20. Zhou, X., Jia, Q., Hu, Y., Xie, R., Huang, T., Yu, F.R.: GenG: An LLM-Based Generic Time Series Data Generation Approach for Edge Intelligence via Cross-Domain Collaboration. In: IEEE INFOCOM 2024 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 1–6. IEEE, Vancouver (2024)
21. Banday, B., Thopalli, K., Islam, T.Z., Thiagarajan, J.J.: On The Role of Prompt Construction In Enhancing Efficacy and Efficiency of LLM-Based Tabular Data Generation. In: IEEE

International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1–5. IEEE, Hyderabad (2025)
22. Moustafa, N., Slay, J.: UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In: Military Communications and Information Systems Conference (MilCIS). IEEE (2015)
23. Sharafaldin, I., Lashkari, A.H., Ghorbani, A.A.: Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. In: 4th International Conference on Information Systems Security and Privacy (ICISSP). Portugal (2018)
24. Farnaaz, N., Jabbar, M.A.: Random Forest Modeling for Network Intrusion Detection System. Procedia Computer Science 89, 213-217 (2016)
25. Kim, D.S., Park, J.S.: Network-Based Intrusion Detection with Support Vector Machines. In: Kahng, H.K. (ed.) Information Networking. ICOIN 2003. Lecture Notes in Computer Science, vol. 2662. Springer, Berlin, Heidelberg (2003)
26. Altunay, H.C., Albayrak, Z.: A hybrid CNN+LSTM-based intrusion detection system for industrial IoT networks. Engineering Science and Technology, an International Journal 38, 101322 (2023)
27. Al-kahtani, M.S., Mehmood, Z., Sadad, T., Zada, I., Ali, G., ElAffendi, M.: Intrusion Detection in the Internet of Things Using Fusion of GRU-LSTM Deep Learning Model. Intelligent Automation & Soft Computing 37(2), 2279–2290 (2023)
28. Issa, M.M., Aljanabi, M., Muhialdeen, H.M.: Systematic literature review on intrusion detection systems: Research trends, algorithms, methods, datasets, and limitations. Journal of Intelligent Systems 33(1), 20230248 (2024)
29. Ji, Z., Dasgupta, D.: Real-valued negative selection algorithm with variable-sized detectors. In: Genetic and Evolutionary Computation Conference, pp. 287–298. Springer (2004)
30. Zhang, G., He, J., Li, W., Li, T., Lan, X., Wang, Y.: DGA-PSO: An improved detector generation algorithm based on particle swarm optimization in negative selection. Knowledge-Based Systems 278, 110892 (2023)