International Journal for Housing Science and Its Applications
Publish August 10, 2025. Volume 46, Issue 4 Pages 8149-8160

https://doi.org/10.70517/ijhsa464698

Deep Learning-Empowered Cybersecurity Threat Prediction and Defense Mechanism Collaboration Architecture

Shan Lu[1,*]
[1] Public Security Information Technology and Intelligence College, Criminal Investigation Police University of China, Shenyang, Liaoning, 110000, China
Corresponding authors: (e-mail: Scs3766@163.com).

Abstract This paper proposes a method for designing a network information security threat prediction and defense mechanism based on deep learning. In terms of threat prediction, through data preprocessing, a deep learning feature extraction model, and the network threat intelligence identification model TriDeepE, efficient classification of network traffic and identification of threat entities are achieved. In terms of defense mechanisms, a multi-layered, adaptive protection system is designed. By leveraging input preprocessing, model enhancement, and continuous security monitoring strategies, the success rate of adversarial sample attacks is effectively reduced. In simulation experiments, the threat prediction model achieved a data anomaly prediction accuracy rate of 95.08%, with MAE and RMSE metrics of 0.0042 and 0.0198, respectively, significantly outperforming other comparison models. Three types of attacks were conducted using H4. After attack cleaning and filtering operations, the Packet-In rate successfully returned to normal levels, validating the effectiveness of the threat defense system.

Index Terms deep learning, network information security, threat prediction, TriDeepE model, defense mechanism

I. Introduction

In today's rapidly evolving information technology landscape, the internet has become an indispensable information tool in people's daily lives and work, as well as the foundational infrastructure and critical pillar of an information-driven society. With the continuous expansion of network infrastructure resources, the steady growth of internet users, and the ongoing innovation and advancement of internet technology, the internet has emerged as an irreplaceable and pivotal influence across all aspects of people's lives, social activities, and economic development [1]-[3]. However, the continuous development of the internet also faces increasing cybersecurity threats. Current mainstream cyber threats include botnet attacks, distributed denial-of-service (DDoS) attacks, spam, worm attacks, and phishing attacks [4]-[7]. These malicious cyber threat activities cause significant economic losses to societal progress and pose increasingly severe challenges and tests for the cybersecurity field.

In response to the increasingly severe situation of rising cybersecurity issues, countries around the world have launched a competition to enhance cybersecurity performance. To more effectively protect the security of computer network systems, researchers have proposed various cybersecurity defense technologies [8]. Early static defense measures primarily include identity authentication, access control, data encryption, firewalls, and hardening operating systems, which serve as the first line of defense for protecting computers and network systems [9], [10]. However, due to their limited functionality, these static technologies cannot form a complete information-sharing network architecture and thus cannot fully prevent network intrusions.

Cybersecurity situational awareness (CSA) has emerged as a prominent cybersecurity management approach in recent years. In simple terms, cybersecurity situational awareness refers to the ability to real-time monitor changes in cybersecurity status and predict future cybersecurity trends [11]-[13]. Almoaigel and Abuabid developed a cybersecurity situational awareness model through empirical analysis, aiming to provide guidance for small and medium-sized enterprises in Saudi Arabia to implement effective cybersecurity measures, specifically to predict subsequent attack behaviors after a network attack [14]. Xu et al. proposed a novel cybersecurity situational awareness model (NSSA) based on semantic ontologies and user-defined rules, aiming to enhance security monitoring, emergency response, and trend prediction capabilities, addressing the limitations of traditional methods in reasoning ability [15]. Liu utilized DS evidence theory to enhance the predictive capabilities of cybersecurity situational awareness and designed a new model that demonstrated high accuracy and robust interference resistance in simulated experiments, providing a theoretical basis for future applications [16]. The aforementioned studies have proposed the concept of cybersecurity situational awareness prediction with proactive predictive capabilities, demonstrating significant innovative ideas and theoretical value.

To ensure the security of increasingly complex network structures, some experts have conducted related research

on network information security situation prediction using machine learning algorithms. For example, Feng et al. proposed a network security situation prediction method combining convolutional neural networks (CNN), gated recurrent units (GRU), and attention mechanisms, which improves prediction accuracy by effectively processing spatial and temporal features [17]. Li et al. proposed a cybersecurity threat prediction method based on feature separation and dual attention mechanisms. Compared to traditional threat prediction models, this method improves threat prediction accuracy to some extent and reduces overfitting [18]. Luo proposed a cybersecurity threat prediction technique based on knowledge graphs, integrating self-attention mechanisms and gate recurrent units. Through empirical testing, this technique was found to improve data reliability and efficiency, achieving high detection accuracy and recall rates [19]. Yao et al. proposed a cybersecurity threat prediction model using an improved attention mechanism combined with a bidirectional long short-term memory-based temporal convolutional network. When tested on real network traffic data, the model performed exceptionally well [20]. Chang proposed a big data and machine learning-based intelligent network security situation prediction method, which achieved a prediction accuracy rate of 96.7% in network security situation prediction, with a prediction time range of 53 to 63 ms, featuring high precision and fast prediction [21]. Chen integrated causal convolutional and temporal convolutional structures to construct a novel cybersecurity threat prediction model, which achieves high-precision predictions and fast prediction speeds, thereby enhancing the network's proactive security defense capabilities [22]. Zhao et al. proposed an attention-based long-short-term cybersecurity threat prediction scheme (ALSNAP), which improves prediction accuracy by combining advanced deep learning algorithms and has the potential for application in real-world networks and intelligent security systems [23].

To date, researchers both domestically and internationally have widely applied artificial intelligence technologies, particularly deep learning, to network information security threat prediction and defense mechanism design, including typical deep learning algorithms such as deep feedforward neural networks (FNN), convolutional neural networks (CNN), recurrent neural networks (RNN), and deep belief networks (DBN) [24]-[26]. Numerous research findings indicate that deep learning-based methods outperform traditional rule-based and machine learning-based methods in terms of attack detection accuracy, stability, efficiency, and defense capabilities [27]. Yan et al. developed a network intrusion detection system (NIDS) based on CNN and utilized generative adversarial networks to synthesize attack records. Experimental results on the KDDCUP99 dataset validated the effectiveness of this method [28]. Lin et al. designed a dynamic network anomaly detection system that uses LSTM for anomaly detection and incorporates an attention mechanism, with its dynamic characteristics reflected in the use of anomaly detection-based methods [29]. To address unknown network attacks, recent research combines deep learning algorithms with statistical extremal theory to achieve open-set classification of network intrusions. Henrydoss et al. applied the Extremal Value Machine (EVM) to identify unknown network attacks, and when classifying known and unknown intrusion attacks on the KDD' CUP 99 dataset, the EVM achieved higher accuracy than traditional network intrusion detection methods [30]. In terms of defense mechanism design, Liu et al. proposed a network information security defense mechanism based on deep learning data interaction. This mechanism optimizes data processing and strategy formulation to reduce the probability of successful attacks, thereby enhancing security [31]. Chen et al. addressed the issue of defense vulnerabilities under intentional attacks targeting IoT infrastructure by proposing a novel defense mechanism. This mechanism was evaluated using a zero-sum game framework and demonstrated the ability to significantly enhance IoT stability [32]. However, the aforementioned deep learning-based network intrusion detection research fundamentally follows a closed-set classification protocol, lacking adaptability to unknown attacks.

This paper first provides a detailed explanation of the preprocessing workflow for network traffic data, including session-based traffic segmentation methods and data cleaning strategies. A deep learning feature extraction model incorporating LSTM was designed to effectively capture the temporal characteristics of network traffic. The TriDeepE model was proposed to address the issue of data scarcity in threat intelligence identification through data augmentation and ensemble learning. Mainstream models were introduced for comparative experiments to explore the performance level of the proposed model. A multi-layered defense system comprising adversarial training, robust optimization, and real-time monitoring was constructed, and its superiority in threat defense efficiency was verified through a series of experiments.

II. Design and Experimentation of Network Information Security Threat Prediction Based on Deep Learning

II. A. Data preprocessing

Network traffic communication is essentially the transmission of “01” bit streams, so the task of classifying network traffic can be viewed as a natural language processing task. The model proposed in this paper is specifically applied to the scenario of traffic classification, so the input data must be preprocessed and formatted in PNG format.

This section primarily introduces the process of converting data from the PCAP format in the dataset into the PNG format required as model input. The entire process is divided into four main parts: traffic segmentation, data cleaning, length standardization, and visualization. The overall workflow is illustrated in Figure 1.

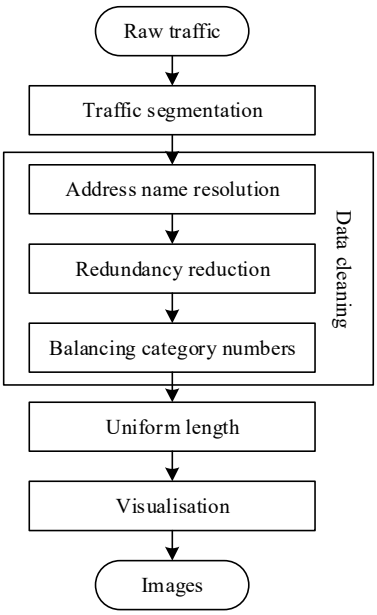

Figure 1: Overall process of data preprocessing

II. A. 1) Traffic Segmentation

All data prior to preprocessing is raw traffic data, which needs to be sliced according to the required traffic granularity. There are two common methods for slicing raw traffic: flow-based slicing and session-based slicing.

(1) Raw traffic data P: a set composed of collected packets p, i.e., $P=\{p^1,...,p^n\}$ where each packet is represented by a quintuple (source IP, source port, destination IP, destination port, transport layer protocol), byte count, and start time, i.e., $p^i=\{x^i,b^i,t^i\}$

(2) Flow-based slicing f: Group packets with the same quintuple in the original traffic data, then arrange the packets in this group in chronological order by start time, ultimately forming a flow f, i.e., $f=\{p^1=(x^1,b^1,t^1),...,p^n=(x^n,b^n,t^n)\}=(x,b,T,t)$, where $x=x^1=...=x^n$, $t^1<t^2...<t^n$, $T=t^n-t^1$. By slicing the original traffic P into streams, it can ultimately be transformed into a set F composed of several streams f, i.e., $F=\{f^1,...,f^n\}$.

(3) Session-based slicing s: Except that the source and destination IP addresses and port numbers can be swapped, packets in the original traffic with the same five-tuple are grouped together, sorted by start time, and ultimately form a session s. By slicing the traffic into sessions, the original traffic P can ultimately be converted into a set S composed of several sessions s, i.e., $S=\{s^1,...,s^n\}$.

This paper uses a session-based traffic segmentation method, treating each session as a single traffic stream to determine whether the traffic is malicious. In addition, based on session-based segmentation, this paper retains all layers of packets as model input. Although the primary characteristics of network session flows are manifested at the application layer—for example, FTP represents file transfer traffic and POP represents email retrieval traffic—the features required for traffic classification tasks, particularly anomaly detection, often manifest in other layers. For instance, transport layer flags can retain some characteristics of network attacks, and transport layer port information can retain some characteristics related to network applications. Wang et al.'s research demonstrates that the session + all-layer traffic representation indeed has the most positive impact on classification performance and is the most appropriate way to segment traffic.

II. A. 2) Data Cleaning

Data cleaning consists of three steps: address anonymization, removal of redundant samples, and balancing the

number of categories.

(1) Address anonymization: Randomize the MAC addresses and IP addresses at the data link layer and network layer, as location information is irrelevant to traffic classification tasks and may interfere with model training.

(2) Removal of redundant samples: Sessions with identical content added as duplicate samples to the training set may affect the weight of the sample in the loss function, leading to model training bias. Therefore, only one instance of each identical session is retained.

(3) Balancing the number of categories: Significant disparities in the number of samples across different categories can cause label imbalance, affecting training performance. Therefore, random downsampling is used to select data from the majority category, ensuring that the number of sessions per category in the training set is approximately equal.

The dataset used in the simulation experiment is the KDD-Cup99 network intrusion detection dataset, which includes 38 dimensional features and a total of 450,000 data points, covering DoS, Probing, R2L, and U2R attacks, all of which are currently mainstream network intrusion types.

II. B.Deep learning feature extraction model

For sequential data such as network traffic, two algorithms are commonly used: recurrent neural networks (RNNs) and convolutional neural networks (CNNs). Long Short-Term Memory (LSTM) networks belong to the category of recurrent neural networks and are effective at extracting temporal features from network traffic time series data. The following sections provide a detailed description of the computational process in conjunction with the network architecture. First, the input layer receives the vector representation of network traffic data x^t. The LSTM layer processes the input data to identify the temporal relationships between data packets. Assuming there are h hidden units, the output dimension of the LSTM layer is also h. The specific computation process of LSTM is as follows:

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{1}$$

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right) \tag{2}$$

$$o_t = \sigma\left(W_o \cdot [h_{t-1}, x_t] + b_o\right) \tag{3}$$

In the equation: f_t, i_t, and o_t are the activation values of the forget gate, input gate, and output gate, respectively; x_t is the current input value; W and b are the weights and biases, respectively. The following calculations are then performed:

$$\tilde{C}_t = \tanh\left(W_C \cdot [h_{t-1}, x_t] + b_C\right) \tag{4}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{5}$$

$$h_t = o_t * \tanh\left(C_t\right) \tag{6}$$

In the formula: $\tilde{C}_t$ is the candidate value of the current unit, C_t is the state value of the current unit, and h_t is the current hidden state value.

After obtaining the features, a fully connected layer is used to map the features to a vector of fixed size, where the size of the vector is the number of network traffic types to be identified. In the fully connected layer, the following calculations are performed:

$$z = W_{fc} \cdot h_{out} + b_{fc} \tag{7}$$

In the equation: h_{out} is the last hidden state of the LSTM layer, and W_{fc} is the weight and bias of the fully connected layer. Subsequently, the Softmax layer is used to convert the output of the fully connected layer into a probability distribution for classification, yielding $p = \{p_i\}$, where p_i is the predicted probability for each network traffic type.

II. C.Network Threat Intelligence Identification Model

Due to the difficulty in obtaining cyber threat intelligence and the complex and time-consuming process of labeling threat intelligence identification datasets, there are currently limited publicly available datasets with few threat entities. This has resulted in existing research models not being adequately trained, leading to poor generalization performance in threat entity identification tasks. To address these issues, this paper proposes a cyber threat intelligence identification model called TriDeepE, which is based on data augmentation and ensemble deep learning. First, the model employs data augmentation strategies to effectively expand the training dataset, ensuring that the deep learning model is adequately trained. Second, the model incorporates ensemble learning principles into the encoding layer. BiLSTM, BiGRU, and CNN models are parallelly integrated using the Bagging algorithm. This integration strategy fully leverages the strengths of each model, thereby enhancing the model's performance in

identifying threat entities despite data scarcity. Finally, a multi-task learning framework is adopted in the decoding layer, enabling the designed integrated deep learning model to simultaneously handle two tasks: cybersecurity text classification and threat entity identification. It fuses the prediction results of each base classifier using the majority voting method to ensure the accuracy of cybersecurity text classification results, and then employs a CRF model to achieve more precise threat entity identification. The model architecture is shown in Figure 2, with its main components including the input layer, embedding layer, and decoding layer. The following is a detailed description of each module.

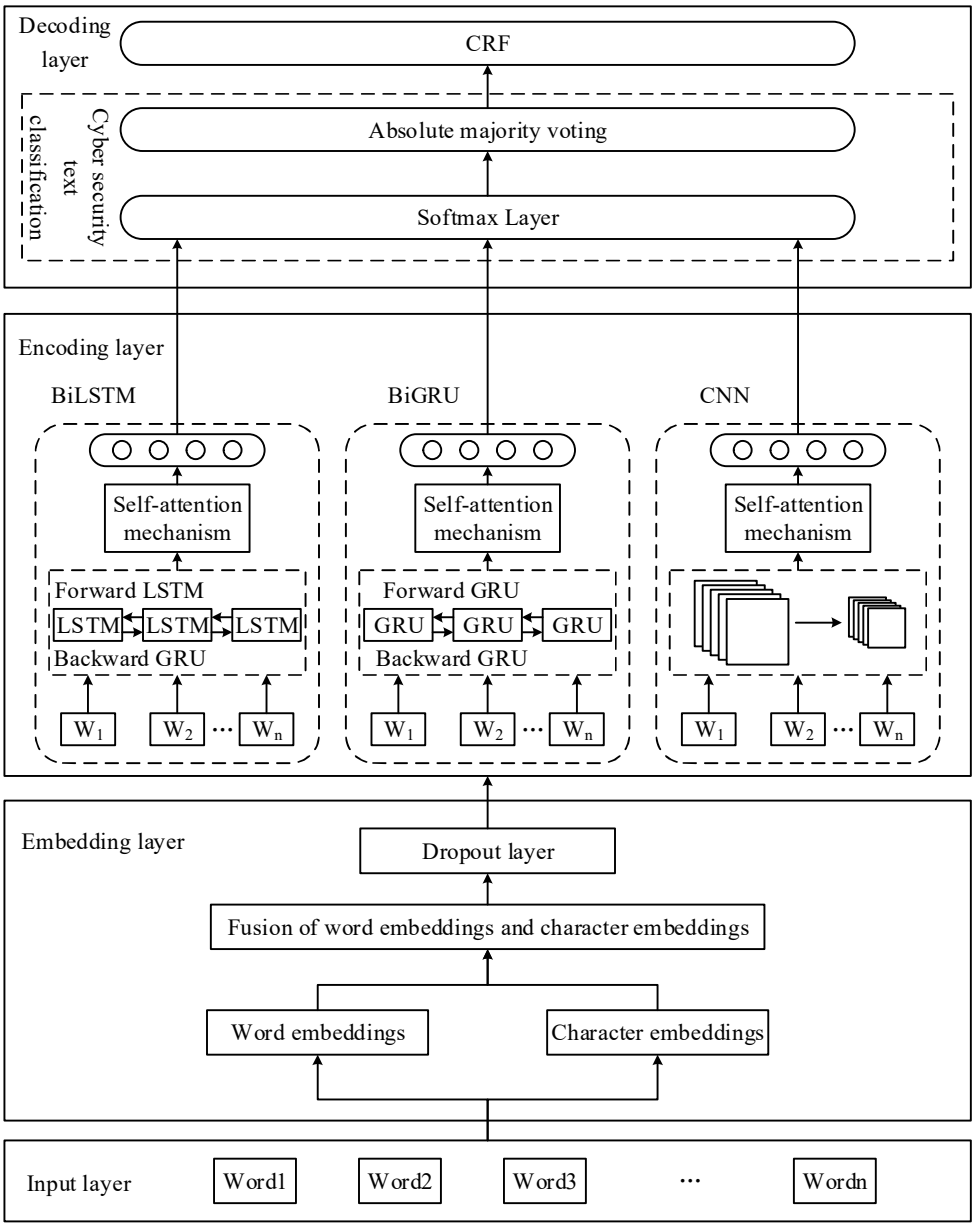

Figure 2: TriDeepE model Architecture

II. C. 1) Input Layer

The input is a sequence of words in an article $S=\{x_1,x_2,...,x_i,...,x_m\}$ is the input, where S represents the threat intelligence text after tokenization, m represents the number of words in the sentence, i.e., the sentence length, and x_i represents the i th word in the sentence. Each word can be represented as $x=\{C_1,C_2,...,C_t,...,C_p\}$, where C_p denotes the i th character of the word, and p denotes the number of characters in the word, i.e., the word length.

II. C. 2) Embedded Layer

In the embedding layer, text semantic representations are enriched by concatenating word embeddings and character embeddings. Word embeddings utilize the skip-gram model from Word2vec, and the word vectors learned through this model can capture semantic relationships between words, providing the model with rich contextual information. Character embeddings first convert characters into fixed-length vectors using one-hot encoding, then utilize the DPCNN model to extract features from these vectors, thereby capturing character-level pattern information.

To prevent the training model from over-relying on specific words or character patterns, leading to insufficient generalization ability, a regularization (Dropout) mechanism is introduced on top of the embedding layer, as shown in Figure 3. The core idea of this mechanism is to randomly "drop out" a portion of neurons during the model training phase, reducing the network's dependence on specific neurons, thereby enabling the model to learn more generalized features and enhancing the robustness of the neural network. In addition, the Dropout mechanism reduces the mutual influence between feature detectors (i.e., hidden layer nodes), allowing each feature detector to operate more independently. Thus, even if some feature detectors are temporarily disabled due to Dropout, other detectors can still maintain normal operation, ensuring the robustness and reliability of the model. Therefore, this paper adopts the Dropout mechanism in the embedding layer to improve the overall performance of the model and, to a certain extent, enhance its noise resistance, ensuring that it can demonstrate more outstanding performance when dealing with complex real-world scenarios.

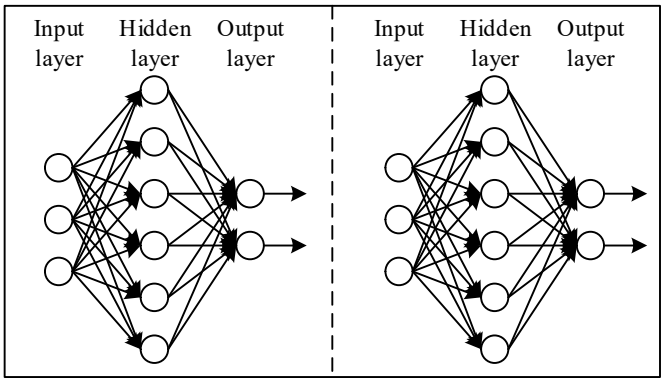

Figure 3: Dropout

II. C. 3) Decoding Layer

In the decoding layer, a multi-task learning framework is still adopted, with different decoding methods designed for cybersecurity text classification and network threat intelligence entity recognition.

(1) Cybersecurity text classification. For this classification task, the TriDeepE model effectively integrates the base learners into a single learner with stronger generalization capabilities through majority voting. First, the decoding layer of each base learner uses the Softmax function to output the final prediction results. Then, the results from the base learners are efficiently fused using a majority voting strategy, ensuring that the ensemble model fully leverages the strengths of each base learner to achieve optimal prediction performance.

During the voting process, all base classifiers are given equal weights, ensuring that each base classifier has an equal vote. This mechanism ensures the fairness and consistency of the model. The model then follows the principle of majority rule, determining the final prediction result based on the number of votes. The category with the highest number of votes is selected as the final prediction result. The voting formula is as follows:

$$V(x) = Max_{j=1}^{n} \sum_{i=1}^{T} c_{i,j} \tag{8}$$

In this context, n represents the total number of categories in entity classification, while T denotes the number of base classifiers. For base classifier i, the predicted category j on test set x is denoted by $c_{i,j}$, and $\sum_{i=1}^{T} c_{i,j}$ calculates the total number of votes for category j across all base classifiers on test set x. Finally, the category with the highest number of votes is selected as the final classification result for the sample x. This strategy ensures classification accuracy and fully utilizes the prediction information from each base learner.

(2) Threat intelligence entity recognition. In this task, the CRF model is further used to perform entity recognition on texts predicted by each base learner to contain threat intelligence entities.

II. D.Experimental Analysis

II. D. 1) Model Testing

This section uses the KDD-Cup99 dataset as a network intrusion dataset and validates it using multiple comparison models, specifically: Bi-LSTM, Att-LSTM, XGBoost, BP neural network, and random forest algorithms. The evaluation metrics include accuracy, model training time, and testing time. The experimental results are shown in Table 1 and Table 2. Due to the simplification of some structures at the expense of accuracy, the BP neural network model has the lowest training and testing times among comparable models. The proposed model achieves an accuracy of 95.08% for predicting data anomalies, significantly outperforming other comparison models.

Table 1: Experimental Results

Algorithm	Accuracy rate/%	Training time/s	Test time/s
Bi-LSTM	89.58	10.43	1.29
Att-LSTM	91.62	9.11	0.96
XGBoost	92.77	13.28	1.02
BP neural network	80.16	6.06	0.76
Random Forest	92.65	9.37	1.35
The proposed	95.08	12.53	1.31

Table 2: Experimental Test Results

Algorithm	MAE	RMSE
Bi-LSTM	0.0058	0.0314
Att-LSTM	0.0041	0.0229
XGBoost	0.0101	0.0223
BP neural network	0.0166	0.0511
Random Forest	0.0104	0.0402
The proposed	0.0042	0.0198

II. D. 2) Situation Assessment Experiment

Four days were randomly selected from the KDD-Cup99 dataset for experimentation, and the security status error results obtained from the experiment are shown in Figure 4. By comparing with the actual status values at the current time, it was found that the method proposed in this paper has a certain degree of accuracy, with the model's quantified status values and actual status values having an error of no more than 0.02.

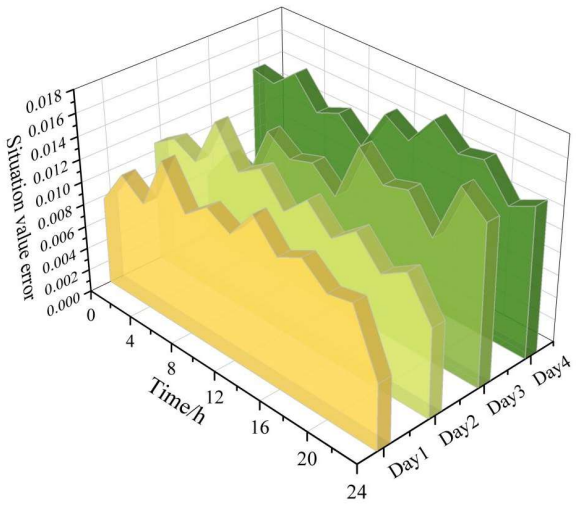

Figure 4: Security situation error

Using the data from Friday throughout the day in the KDD-Cup99 dataset as the sample, the sample is divided into 60-minute time intervals to quantify the situation on Friday. The experimental results comparing the proposed model with the control model are shown in Figure 5. The Bi-LSTM model yielded the largest error between the quantified situation values and the actual situation values, with a maximum difference of over 0.15. In contrast, the proposed model produced the closest match between the quantified situation values and the actual situation values, validating the advantages of the proposed model in practical cybersecurity situation assessment.

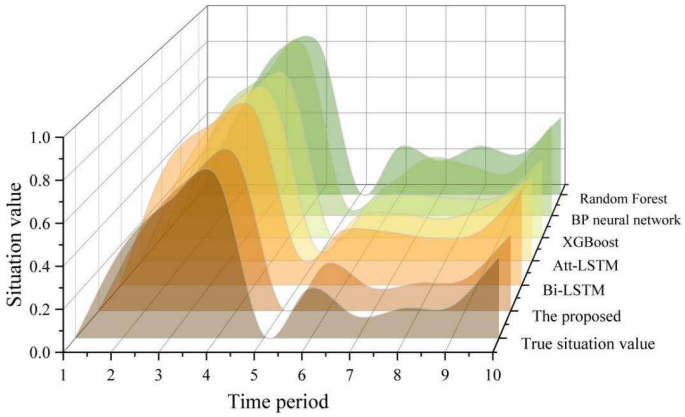

Figure 5: Comparison test results

Further, 200 data samples were randomly selected from the 60-minute time segment as the research subjects. The comparison results of the situation predictions for the 200 samples are shown in Figure 6. It can be seen that the situation value quantified by the model in this paper is still the best, with an average error of less than 0.01 compared to the actual situation value, further verifying the effectiveness of the model in this paper.

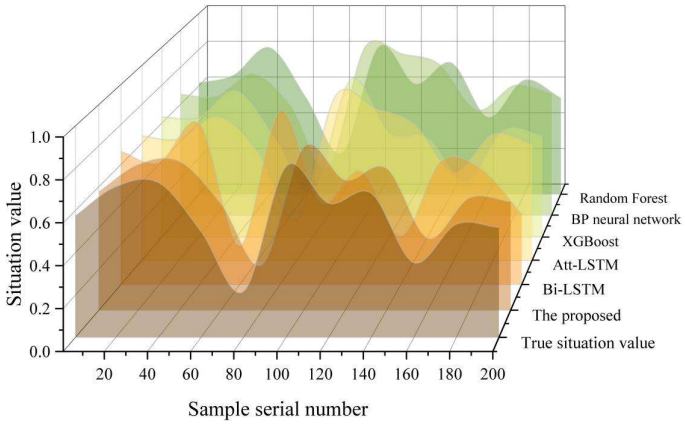

Figure 6: Comparison results of situation prediction for 500 samples

III. Research on network security defense mechanisms based on deep learning

III. A. Defense Strategy Design

In deep learning-driven network threat detection systems, adversarial attacks have become a key challenge affecting model reliability. To build robust cybersecurity defense mechanisms, this study proposes a multi-layered, adaptive protection system covering multiple dimensions, including input preprocessing, model enhancement, system architecture optimization, and continuous security monitoring.

In terms of input preprocessing, the defense mechanism focuses on disrupting the perturbation patterns of adversarial samples. Randomized encoding techniques are employed to dynamically transform network traffic, including random reordering of packet arrival sequences, nonlinear mapping of feature dimensions, and discretization of continuous features. For example, in a network traffic detection system, packet arrival sequences can be randomly scrambled or continuous features can be discretized. Experimental results demonstrate that the success rate of adversarial sample attacks is reduced by 80%. In terms of model enhancement, a strategy combining adversarial training and robust optimization is adopted. During training, the system alternates between using original samples and adversarial samples for model updates, where adversarial samples are generated using

a multi-step projection gradient descent method to cover a wide range of attack scenarios. To improve training efficiency, an adaptive adversarial sample generation algorithm is designed to dynamically adjust the perturbation magnitude and attack direction based on the model's current vulnerability.

More advanced defense strategies include: detection mechanisms, where a dedicated adversarial sample identification module is developed to detect potential attacks by analyzing signals such as input feature anomalies and inconsistent prediction results. System architecture, where a deep integration model is constructed to combine the prediction results of multiple heterogeneous submodels. This architecture significantly improves the overall robustness of the system by increasing the deception complexity for attackers. Security monitoring, where a model behavior audit mechanism is established to regularly check whether the prediction logic deviates from expectations. At the formal verification level, a robustness proof method based on abstract interpretation is proposed. By constructing the linear relaxation boundaries of each layer of the neural network, the worst-case output deviation of the model within a given perturbation range is calculated. This method provides verifiable security guarantees for specific types of adversarial perturbations, particularly suitable for high-reliability scenarios in critical infrastructure. Additionally, for threats during the training phase, various protective measures have been developed, including data source verification and distributed consensus verification.

This defense mechanism not only provides effective protection against currently known adversarial attack methods but also lays the foundation for addressing future novel threats through its scalable framework design. Its core innovation lies in combining traditional security engineering's layered defense philosophy with deep learning characteristics to build collaborative defense capabilities at the algorithmic, system, and architectural layers.

III. B. Threat Defense Experiment

III. B. 1) Threat Detection

The primary function of the threat detection module is to compare the transmission rate of Packet-In with a predefined threshold to enable attack alerts. When the transmission rate of Packet-In exceeds the predefined threshold, the module triggers an alert and automatically initiates the threat detection algorithm and threat defense system. This module primarily targets UDP Flood attacks originating from attack hosts H1, H2, and H3. The attack simulation process involves first injecting 60 seconds of normal traffic. During the attack, different attack frequencies can be set. The module records the transmission rate of Packet-In during the attack and displays how the transmission rate changes at different attack rates. If the transmission rate exceeds the predefined threshold, an alert is triggered, allowing system administrators to promptly take measures to prevent the attack from escalating further. The changes in the transmission rate of Packet-In at different attack rates are shown in Figure 7. The generation rate of Packet-In varies at different attack rates. For low-rate attacks, the changes in Packet-In messages are not significant, making them difficult to monitor and identify.

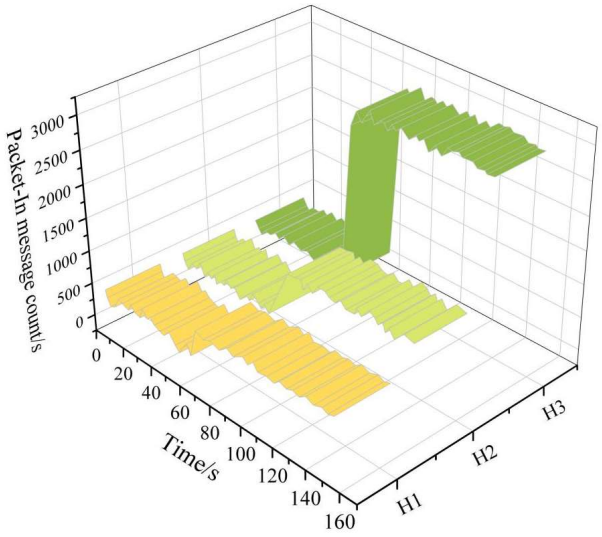

Figure 7: Packet-In sending rate under different attack rates

Three types of attacks were carried out using H4, and the rate changes of Packet-In messages are shown in Figure 8. Observing Figure 8, it can be seen that the rate of Packet-In messages changed beyond the preset threshold in all three types of attacks. Once the threat assessment system detects a change in rate, it can predict

an attack, issue an alert, and take appropriate measures.

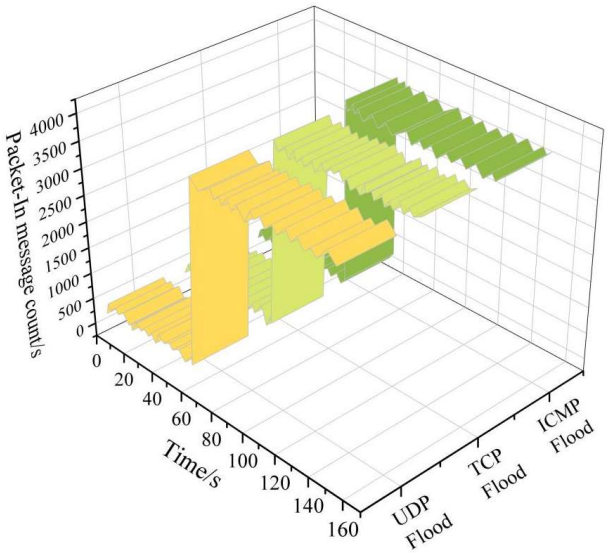

Figure 8: Packet-In sending rate under different attack methods

III. B. 2) Attack Mitigation

The attack mitigation module experiment in the threat defense system is divided into two parts. The first part involves the system immediately issuing an alert and initiating service redirection upon detecting a threat, with the aim of mitigating the impact of the attack on the host. In this part, the attacking host or suspicious host is redirected to a new server, thereby reducing the load on the affected server. The second part involves cleaning and filtering operations to remove attack packets and protect the server from unauthorized access. First, the Packet-In transmission rate of the experimental host is monitored. Once an abnormal change in the transmission rate is detected, the Packet-In messages are immediately used to locate the victim host, and service redirection is initiated to redirect the attack packets.

To implement attack data redirection and cleaning and filtering measures, the controller must compare the collected attack traffic information with the feature information from the information collection module to identify and clean abnormal traffic. After the attack cleaning and filtering operation, the Packet-In rate after attack mitigation is shown in Figure 9. The results indicate that the Packet-In rate has returned to normal levels, confirming that the cleaning and filtering operation successfully mitigated the threat attack.

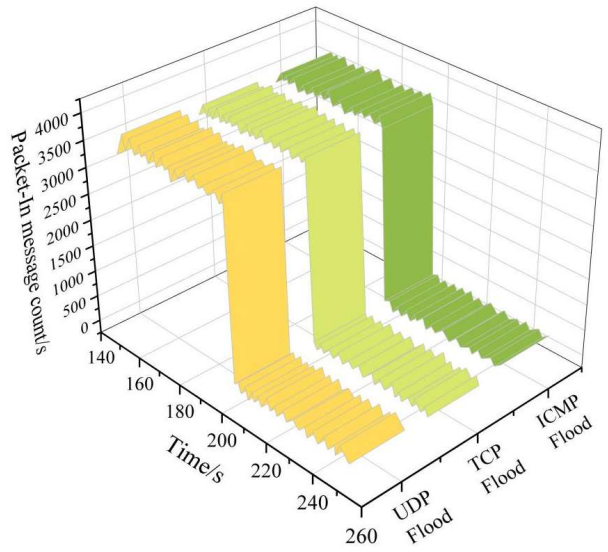

Figure 9: Packet-In rate after the attack is mitigated

In summary, the experiments demonstrate that the threat assessment defense system proposed in this paper can effectively identify threat attacks and take corresponding measures to successfully mitigate threat attack behavior, ensuring the normal operation of the network.

IV. Conclusion

This paper systematically investigates a deep learning-based network information security threat prediction and defense mechanism, and designs corresponding experiments to evaluate its performance.

The threat prediction model achieves an accuracy rate of 95.08% for data anomaly prediction, with MAE and RMSE metrics of 0.0042 and 0.0198, respectively, significantly outperforming other comparison models. In the experimental samples divided into 60-minute time intervals, the quantitative threat values of this model are closest to the actual threat values. In the threat prediction of 200 samples, the quantitative threat values of this model remain optimal, with an average error of less than 0.01 compared to the actual threat values.

When implementing three types of attacks using H4, the rate of Packet-In messages exceeded the predefined threshold in all three attack types. After attack cleaning and filtering operations, the Packet-In rate successfully returned to normal levels post-attack mitigation, validating the effectiveness of the threat defense system.

References

[1] Thakur, K., Qiu, M., Gai, K., & Ali, M. L. (2015, November). An investigation on cyber security threats and security models. In 2015 IEEE 2nd international conference on cyber security and cloud computing (pp. 307-311). IEEE.

[2] Dastres, R., & Soori, M. (2021). A review in recent development of network threats and security measures. International Journal of Information Sciences and Computer Engineering.

[3] Wu, X., Du, Y., Fan, T., Guo, J., Ren, J., Wu, R., & Zheng, T. (2023). Threat analysis for space information network based on network security attributes: a review. Complex & Intelligent Systems, 9(3), 3429-3468.

[4] Hoque, N., Bhattacharyya, D. K., & Kalita, J. K. (2015). Botnet in DDoS attacks: trends and challenges. IEEE Communications Surveys & Tutorials, 17(4), 2242-2270.

[5] Dalmazo, B. L., Marques, J. A., Costa, L. R., Bonfim, M. S., Carvalho, R. N., da Silva, A. S., ... & Cordeiro, W. (2021). A systematic review on distributed denial of service attack defense mechanisms in programmable networks. International Journal of Network Management, 31(6), e2163.

[6] Bhatia, S., Behal, S., & Ahmed, I. (2018). Distributed denial of service attacks and defense mechanisms: current landscape and future directions. Versatile Cybersecurity, 55-97.

[7] Jain, A. K., & Gupta, B. B. (2022). A survey of phishing attack techniques, defence mechanisms and open research challenges. Enterprise Information Systems, 16(4), 527-565.

[8] Zhao, G., & Song, J. (2020). Network security model based on active defense and passive defense hybrid strategy. Journal of Intelligent & Fuzzy Systems, 39(6), 8897-8905.

[9] Cárdenas, A. A., Roosta, T., Taban, G., & Sastry, S. (2008). Cyber security basic defenses and attack trends. Homeland Security Technology Challenges, 73-101.

[10] Zheng, Y., Li, Z., Xu, X., & Zhao, Q. (2022). Dynamic defenses in cyber security: Techniques, methods and challenges. Digital Communications and Networks, 8(4), 422-435.

[11] Kim, K., Youn, J., Kim, H., Shin, D., & Shin, D. (2024). State-of-the-Art in Cyber Situational Awareness: A Comprehensive Review and Analysis. KSII Transactions on Internet and Information Systems (TIIS), 18(5), 1273-1300.

[12] Gutzwiller, R., Dykstra, J., & Payne, B. (2020). Gaps and opportunities in situational awareness for cybersecurity. Digital Threats: Research and Practice, 1(3), 1-6.

[13] Alavizadeh, H., Jang-Jaccard, J., Enoch, S. Y., Al-Sahaf, H., Welch, I., Camtepe, S. A., & Kim, D. D. (2022). A survey on cyber situation-awareness systems: Framework, techniques, and insights. ACM Computing Surveys, 55(5), 1-37.

[14] Almoaigel, M. F., & Abuabid, A. (2023). Implementation of Cybersecurity Situation Awareness Model in Saudi SMES. International Journal of Advanced Computer Science & Applications, 14(11).

[15] Xu, G., Cao, Y., Ren, Y., Li, X., & Feng, Z. (2017). Network security situation awareness based on semantic ontology and user-defined rules for Internet of Things. IEEE Access, 5, 21046-21056.

[16] Liu, D. (2020). Prediction of network security based on DS evidence theory. ETRI Journal, 42(5), 799-804.

[17] Feng, Y., Zhao, H., Zhang, J., Cai, Z., Zhu, L., & Zhang, R. (2024). Prediction of Network Security Situation Based on Attention Mechanism and Convolutional Neural Network–Gated Recurrent Unit. Applied Sciences, 14(15), 6652.

[18] Li, Z., Zhao, D., Li, X., & Zhang, H. (2021). Network security situation prediction based on feature separation and dual attention mechanism. EURASIP Journal on Wireless Communications and Networking, 2021, 1-19.

[19] Luo, W. (2024). Network Security Situation Prediction Technology Based on Fusion of Knowledge Graph. International Journal of Advanced Computer Science & Applications, 15(4).

[20] Yao, C., Yang, Y., Yang, J., & Yin, K. (2022). A network security situation prediction method through the use of improved TCN and BiDLSTM. Mathematical Problems in Engineering, 2022(1), 7513717.

[21] Chang, Z. (2025). Intelligent Prediction Method for Network Security Situation based on Big Data and Machine Learning. International Journal of High Speed Electronics and Systems, 2540461.

[22] Chen, H. (2025). Active Defense Mechanism for Network Security Situation Prediction Based on Transformer and TCN. Engineering Research Express.

[23] Zhao, D., Shen, P., & Zeng, S. (2023). ALSNAP: Attention-based long and short-period network security situation prediction. Ad Hoc Networks, 150, 103279.

[24] Okoli, U. I., Obi, O. C., Adewusi, A. O., & Abrahams, T. O. (2024). Machine learning in cybersecurity: A review of threat detection and defense mechanisms. World Journal of Advanced Research and Reviews, 21(1), 2286-2295.
[25] Zhang, C., Costa-Perez, X., & Patras, P. (2022). Adversarial attacks against deep learning-based network intrusion detection systems and defense mechanisms. IEEE/ACM Transactions on Networking, 30(3), 1294-1311.
[26] Sulaiman, M., Waseem, M., Ali, A. N., Laouini, G., & Alshammari, F. S. (2024). Defense strategies for epidemic cyber security threats: modeling and analysis by using a machine learning approach. IEEE Access, 12, 4958-4984.
[27] Khaleel, Y. L., Habeeb, M. A., Albahri, A. S., Al-Quraishi, T., Albahri, O. S., & Alamoodi, A. H. (2024). Network and cybersecurity applications of defense in adversarial attacks: A state-of-the-art using machine learning and deep learning methods. Journal of Intelligent Systems, 33(1), 20240153.
[28] Yan, Q., Wang, M., Huang, W., Luo, X., & Yu, F. R. (2019). Automatically synthesizing DoS attack traces using generative adversarial networks. International journal of machine learning and cybernetics, 10(12), 3387-3396.
[29] Lin, P., Ye, K., & Xu, C. Z. (2019). Dynamic network anomaly detection system by using deep learning techniques. In Cloud Computing–CLOUD 2019: 12th International Conference, Held as Part of the Services Conference Federation, SCF 2019, San Diego, CA, USA, June 25–30, 2019, Proceedings 12 (pp. 161-176). Springer International Publishing.
[30] Henrydoss, J., Cruz, S., Rudd, E. M., Gunther, M., & Boult, T. E. (2017, December). Incremental open set intrusion recognition using extreme value machine. In 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA) (pp. 1089-1093). IEEE.
[31] Liu, Y., Wang, Q., Zheng, Z., & Cui, L. (2024). Control modeling and optimization of network information security system based on deep learning data interaction. Measurement: Sensors, 33, 101221.
[32] Chen, P. Y., Cheng, S. M., & Chen, K. C. (2014). Information fusion to defend intentional attack in internet of things. IEEE Internet of Things journal, 1(4), 337-348.