A Network Security Situation Prediction Model Enhanced by Multi Head Attention Mechanism

Jian Chen[1*], Huanqiang Bian[2], Hao Liang[3]
[1]Laboratory and Information Technology Center, Ningbo University of Finance & Economics, Ningbo 315175, China
[2]Library (Information Technology Center), Ningbo City College of Vocational Technology; Ningbo 315175, China
[3]Big Data Division, Ningbo University of Technology, Ningbo 315175, China
E-mail: chenjian@nbufe.edu.cn
*Corresponding author

Keywords: multi-head self-attention mechanism, network security, situation prediction, gated recurrent neural network, encoder

Received: November 26, 2024

The Internet has grown as a result of information technology advancements, and cybercrime is becoming more and more common. To improve the network defense against all kinds of network attacks and reduce the success rate of cyber crimes, the research innovatively proposes to use the multi-tease attention mechanism to improve the gating cycle unit, and use the multi-head attention mechanism to obtain network security feature information at different locations, so as to improve the learning characteristics of network situation prediction and realize network security situation prediction. Three layers comprised the model: the prediction layer, the transform layer, and the circular network layer. The circular network layer was responsible for dimensionality reduction of network information data. Information features were extracted via the transform layer. The outcomes of the predictions were output by the prediction layer. The study's model performed better when taught in both directions, according to the data, and its accuracy could reach roughly 93.5%. The highest level of model accuracy could be reached when other parameters were fixed and the neurons in the feed-forward layer was 28. Compared with other network security situation prediction models, the proposed model could improve the prediction accuracy to around 93.5% and the precision to around 91% on the UNSW-NB15 dataset, while maintaining the F1 value of the model at around 92%. The research-designed model can accurately predict the network security situation changes, which improves the Internet's defense against attacks and maintains the normal operation of the Internet community.

Povzetek: Raziskava predstavlja model napovedovanja omrežne varnostne situacije, ki z večglavim mehanizmom pozornosti izboljša GRU in dosega visoko kvaliteto pri zaznavanju kompleksnih napadov.

1 Introduction

Networks have become the foundation of contemporary civilization due to the quick advancement of information technology. However, all kinds of network attacks are emerging, and network security (NS) problems are becoming increasingly serious. As an important means to ensure the stable operation of network environment, network security situation prediction (NSSP) is directly related to the security and reliability of network system [1-2]. NSSP technology is a key research direction in the field of information security. It predicts the future development trend by analyzing the type and frequency of attack events and the situation value after correlation and fusion, as well as by using the information of past attack events and the situation value [3-4]. The current state of research shows that network security situation (NSS) sensing system currently exists shortcomings such as sensing without action, unable to recognize new attack methods. Therefore, there is an urgent need to propose a responsive security situation awareness platform architecture to solve the problem of integrated perception and action construction [5]. The traditional approach assumes that network attack events are independent, meaning that the occurrence of one attack event does not affect the occurrence of another attack event. However, in the real world, attack events are often interrelated, and one attack event can trigger or mask other attack events. This assumption results in significant prediction errors for traditional NSSP methods when dealing with irregularly changing and fluctuating data, limiting their effectiveness in practical applications. With the continuous increase of network data and the diversification of network attack methods, the NSSP becomes more and more difficult. The traditional NSSP method has a large error in the prediction results when facing irregularly changing and fluctuating data. Its high requirement for data independence leads to a large limitation in the scope of use [6-7]. Therefore, in order to improve the network defense capability and network situational prediction accuracy, the study proposes to improve the gated recurrent unit (GRU) by using multi-head attention mechanism (MHAM). Furthermore, the NSS is predicted and analyzed using the enhanced GRU.

The main objective of this study is to improve the

accuracy and robustness of NSSP by proposing a new model that combines MHAMGRU. It is assumed that the integration of MHAM into the GRU framework will significantly enhance the model's ability to capture complex patterns and long-term dependencies in NS data, thereby improving the accuracy and precision of predictions. The expected results of this study include the development of a model that can accurately predict changes in the NSS, thereby improving the accuracy and robustness of the NSSP. The main measurement indicators are accuracy, precision and F1 value.

The research innovatively proposes to use MHAM to improve the GRU, which utilizes the MHAM to obtain the NS feature information of different locations, so as to increase the learning characteristics of network situation prediction to realize NSSP. Network attacks typically have diversity and complexity, and different types and methods of attacks can produce different characteristics in different network locations. By analyzing the characteristics of different locations, the model can more comprehensively capture the diversity and complexity of attacks, thereby improving the accuracy of predictions. The main contribution of this research is to design a deep learning (DL) neural network (NN) that considers NS features of different locations and use the network to conduct prediction studies on complex NSSs. The NS defense is improved and the NSSP is improved.

2 Related works

NSSP can significantly improve NS defense capability and guarantee network operation security. Chen developed a radial basis function NN prediction model based on enhanced genetic algorithm optimization to increase the precision of NSS perception prediction and adapt to evolving network assault technologies. According to the findings, the optimized model prediction value was reasonably close to the real value, which helped with NS maintenance [8]. To solve the security difficulties of mobile IoT healthcare networks, Xu et al. suggested an enhanced convolutional NN model in conjunction with an intelligent prediction method for security performance. The outcomes demonstrated that the algorithm successfully safeguarded the medical data while increasing prediction accuracy by 20% [9]. Zhang M et al. proposed a NS encryption method based on chaotic iterative system to meet the encryption protection of indoor surveillance video data. The results showed that this scheme effectively improved the encryption level of surveillance data [10]. Kure et al. suggested an integrated NS risk management method based on fuzzy sets, machine learning and comprehensive assessment model in order to deal with cyber threats to cyber-physical systems. The results indicated that the method effectively assessed asset criticality, accurately predicted risk types, and helped to proactively manage risks [11].

One popular DL NN is called GRU. To increase the precision of rotating machinery condition maintenance, Ni et al. proved a rolling bearing prediction method based on health indicators. The scheme analyzed the bearing state by methods such as spectral correlation and combined with an intelligent model to predict RUL. The outcomes revealed that the new indicator was monotonic and had high prediction accuracy [12]. Zhang et al. suggested a DL method based on GRU in order to accurately predict landslide displacement. The results revealed that the method outperformed other DLNNs. It could better capture historical information and cycle displacement changes to improve the prediction accuracy [13]. For intelligent transportation systems, Shu et al. suggested a prediction model based on an enhanced GRU NN to increase the precision of short-term traffic flow prediction. The outcomes revealed that the model combined bidirectional positive feedback with RAdam optimizer to significantly improve the prediction accuracy [14]. Lin H et al. illustrated a GRU DL model based on dual GRU with pattern decomposition based in order to predict the groundwater level in Qo Say Plain, Iran. The outcomes revealed that the dual GRU model performed better with high prediction accuracy and computational efficiency [15]. The summary of research and investigation literature is shown in Table 1.

In summary, accurate prediction of NSS can effectively improve NS, but existing DL models are prone to over-fitting when processing high-dimensional data, resulting in good performance on the training set but poor generalization ability on the TeS. The traditional NSSP method has a large prediction error when dealing with irregular and fluctuating data, and its

Table 1: Summary of related works survey results

Reference	Method	Data set	Accuracy (%)	Precision (%)	F1 value
Chen [8]	GA-RBF	UNSW-NB15	89.0	88.5	88.8
Xu et al. [9]	Enhanced CNN	IoT Medical Network Dataset	92.0	91.5	91.8
Zhang et al. [10]	Docker-deep learning	SiteWhere Platform Dataset	90.5	90.0	90.3
Kure et al. [11]	Improving machine learning with fuzzy sets	Industrial Control System Dataset	91.0	90.5	90.8
Ni et al. [12]	Health indicator prediction	Mechanical Fault Dataset	92.5	92.0	92.3
Zhang et al. [13]	GRU	Geological hazard dataset	93.0	92.5	92.8
Shu et al. [14]	EnhancedGRU	Traffic flow dataset	93.5	93.0	93.3
Lin H et al. [15]	Pattern decomposition GRU	Groundwater level dataset	93.2	92.7	93.0

high data independence requirement limits its application scope. In addition, existing research does not focus on the location characteristics of network attacks, resulting in a decrease in the sensitivity of network attack detection models to network attack behavior. Therefore, a NSSP model based on MHAM-enhanced GRU is proposed. The acquisition of NS feature information from different locations by MHAM improves the learning characteristics of network situation prediction. The Transformer encoder architecture is utilized to improve the feature extraction capability and training effectiveness of the model. Residual links and layer normalization in the Transformer architecture can effectively prevent gradient vanishing problems and improve model training performance.

3 Methods and materials

3.1 Improved GRU Based on MHAM

Attention mechanisms (AMs) are common techniques in the field of DL. The mechanism enables the model to focus on the most important parts of the input data by mimicking human visual attention [16-17]. The introduction of an AM can improve the feature sensitivity of the network to different attack methods and increase the sensitivity of the model to network attack behavior. Common AMs include scaled dot product attention mechanism (SDPAM) and multi-head self-attention mechanism (MHSAM). SDPAM determines the weights by calculating the dot product of Query and Key, which is simple to operate and easy to understand, as shown in Equation (1) [18-19].

$$A(Q,K,V) = soft\max\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (1)$$

In Equation (1), d_k and d_v denote the target feature dimensions. Q and K denote the input feature dimensions of the SDPAM when the feature dimension is d_k. The MHSAM can be obtained by splicing multiple SDPAMs and performing a linear transformation. The computation of the MHSAM is shown in Equation (2).

$$MH(Q,K,V) = C(h_1,h_2,\cdots,h_n)W^o \quad (2)$$

In Equation (2), n is the total heads counts. h_i denotes the i th head feature. C denotes vector splicing operation. W^o denotes the WM. The calculation of h_i is shown in Equation (3).

$$h_i = A\left(QW_i^Q, KW_i^K, VW_i^V\right) \quad (3)$$

The length of time series data is often not fixed, and traditional NN models are difficult to handle variable-length sequence data. GRU can handle variable-length sequence data, and the input to the model can be a sequence of any length, and the output can be any position in the sequence. This allows GRU to be very flexible in processing time series data and to adapt to variable length sequence data. Sequential data processing commonly uses the DL model GRU [20]. By implementing a gating mechanism, GRU aims to regulate information flow. The introduction of gates allows GRU to better capture long-term dependencies when processing long sequence data. Figure 1 depicts the internal organization of GRU.

In Figure 1, □ denotes element-by-element multiplication. The basic structure of GRU includes two gates and two hidden states (HS). The two gates are update gate (UG) and reset gate (RG). The UG of the GRU determines how much information is retained from the input data of the current time step (TS), namely the characteristics of the network attack behavior, as well as how much information is retained from the HS of the previous TS. The formula for the UG is given in Equation (4).

$$z_t = \sigma\left(W_z\left[h_{t-1}, x_t\right]\right) \quad (4)$$

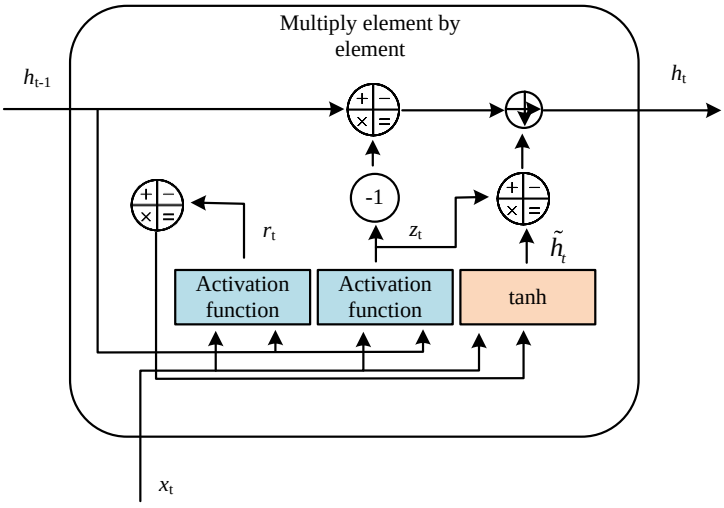


Figure 1: Gated recurrent unit major structure (Author's self drawn).

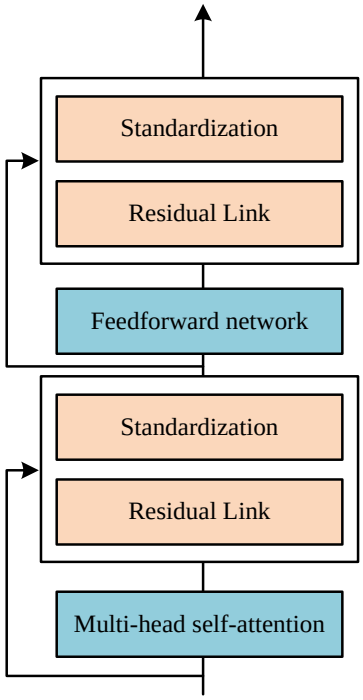


Figure 2: Internal structure of the transformer encoder architecture.

In Equation (4), σ denotes the activation function. W_z is the update weight matrix (WM). h_{t-1} denotes the HS of the previous TS. x_t denotes the input of the current TS. How much data is kept from the previous TS's HS is decided by the RG. Equation (5) displays the formula for the RG.

$$r_t = \sigma\left(W_r\left[h_{t-1}, x_t\right]\right) \tag{5}$$

In Equation (5), W_r is the reset WM. The two HSs are the candidate HS and the final HS, respectively. Equation (6) shows the formula for calculating the candidate HS, which is the sum of the current TS and RG inputs.

$$\tilde{h}_t = \tanh\left(W\left[r_t \square\ h_{t-1}, x_t\right]\right) \tag{6}$$

In Equation (6), W denotes the candidate HS WM. The final HS is the combination of the UG and the candidate HS. The MHSAM is usually found in the Transformer architecture. Transformer architecture usually consists of encoder and decoder. The study uses only its encoder architecture in the improvement of GRU using MHSAM. The Transformer encoder architecture adds a residual connection structure and layer normalization between the multi-head self-attention (MHSA) layer and the feed forward network layer. The internal structure of the encoder is shown in Figure 2 [21].

The positional connection properties of the analyzed data cannot be extracted by the GRU model when it is optimized solely based on the encoder structure. The cosine function and sine function can be used to encode the expression of the data position through angular analysis. The study adds a sine function with cosine function before inputting the original features to express the data position as shown in Equation (7).

$$\begin{cases} PE_{(pos,2i)} = \sin\left(\dfrac{pos}{1000^{2ild_{\text{mod}}}}\right) \\ PE_{(pos,2i+1)} = \cos\left(\dfrac{pos}{1000^{2ild_{\text{mod}}}}\right) \end{cases} \tag{7}$$

In Equation (7), pos denotes the position. $PE_{(pos,2i)}$ and $PE_{(pos,2i+1)}$ display the position information of the original data when the dimension is even or odd. d_{mod} denotes the encoder input features. For sequential data, positional information can help models understand the temporal order and contextual relationships of the data. The sine and cosine functions have periodicity that allows the model to handle sequences longer than the length of the training sequence. The values of the sine and cosine functions are unique for different locations. This ensures that each location has a unique encoding, preventing confusion of location information. The values of sine and cosine functions change smoothly, which enables the model to better capture subtle changes in positional information. Residual connection adds the output of the feed-forward network layer to the output of the MHSAM, and layer normalization normalizes the added result. This structure ensures effective transmission of information in multi-layer networks and prevents gradients from disappearing during back-propagation, thereby improving the training effectiveness and convergence speed of the model. The encoder's feed forward network can successfully prevent the issue of model training degradation brought on by gradient vanishing during training. Figure 3 displays the network structure of the MHSAM-based research-designed network situational prediction.

The overall structure of the model constructed in the study is divided into GRU layer, MHSA layer and prediction layer. The MHSA layer adopts the Transform architecture, which consists of position embedding and encoder. The prediction layer and MHSA layer are connected by dropout mechanism and full connectivity.

3.2 Improved GRU-based NSSP

The study utilizes MHSAM after improving the GRU network. That is, the model's primary component for predicting and analyzing NSS is the enhanced network. The data related to NSS is characterized by large data size and high data dimension. High dimensional data increases the computational complexity of the model, resulting in slower training and inference processes [22-23]. Due to the high dimensionality of NSS data, the raw data cannot be directly input into the model. It is necessary to first use an improved GRU to pre-process the raw data, and then use the GRU to map the high-dimensional raw data into a low-dimensional space to

reduce the dimensionality of the data. As a result, Figure 4 depicts the structure of the NSSP model that the study created.

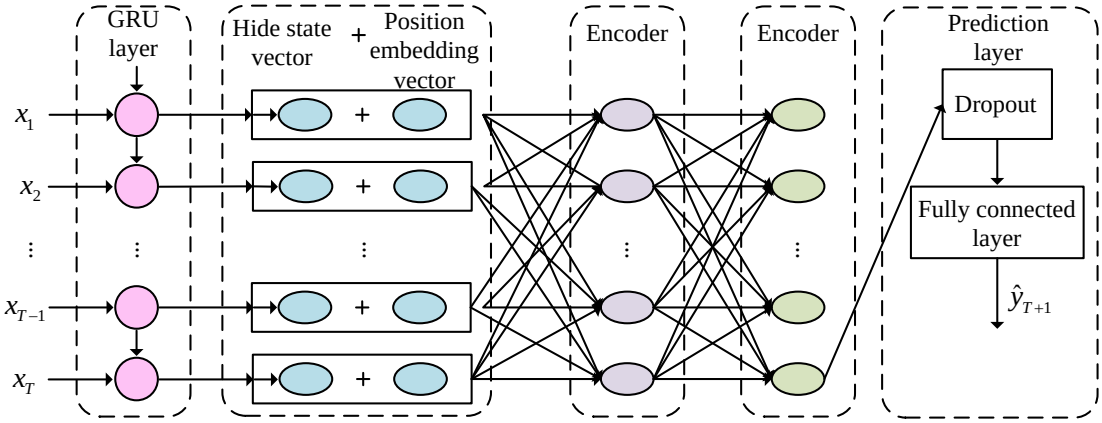


Figure 3: Advanced representation of GRU network based on MHSA improvement.

The study's NSSP model is built using a three-layer design overall. The data preparation layer is the first layer. The GRU network layer, which is enhanced based on MHSA, is the second layer. The model output layer is the third layer. For data correction, the original data must pass through the pre-processing layer before reaching the enhanced GRU network layer. The main operations of the pre-processing layer include data filtering, numerization, normalization and time serialization. Data filtering is to remove invalid data from the input information. Numericalization is to convert the encoding type of data information to facilitate model training and learning. Standardization can effectively reduce the differences between different data. Time serialization is to make the current cyber-attack data coincide with the historical cyber-attack data. Network attacks often do not occur in isolation, but have continuity and evolutionary processes. Historical data can provide background information about attacks and help models understand the origin, development, and evolution of attacks. Therefore, it is necessary to maintain data consistency. The original data set can be input into the improved GRU network layer after pre-processing. If the input data is assumed to be sequence $X=(x_1,x_2,\cdots,x_T)$, the sequence enters into the improved GRU network and needs to undergo data dimensionality reduction first, as shown in Equation (8).

$$H=GRU\left(\left[x_t\right]t=1,2,\cdots,T\right) \tag{8}$$

In Equation (8), H denotes the data features after dimensionality reduction. T denotes the TS. The data dimensionality reduction method used in the study is feature compression, where the GRU network maps high-dimensional input data to a low-dimensional HS space. Through this mapping, the model can compress the information in the input data into a low-dimensional representation, reducing the dimensionality of the data and improving computational efficiency. After the input data is processed by dimensionality reduction, the position information can be input through the position embedding vector in the improved GRU network. The embedding of location information relies on location encoding. The encoder is used to extract the information features of the current sequence after embedding the location data. After the encoder completes the feature extraction, NSSP can be achieved by Dropout with full connectivity. The Dropout mechanism effectively reduces the risk of model over-fitting by randomly dropping certain neurons, and this operation is relatively simple and does not affect the model structure. Therefore, this regularization technique is used in this study. A fully connected layer refers to a type of layer in a NN where each neuron is connected to all neurons in the previous

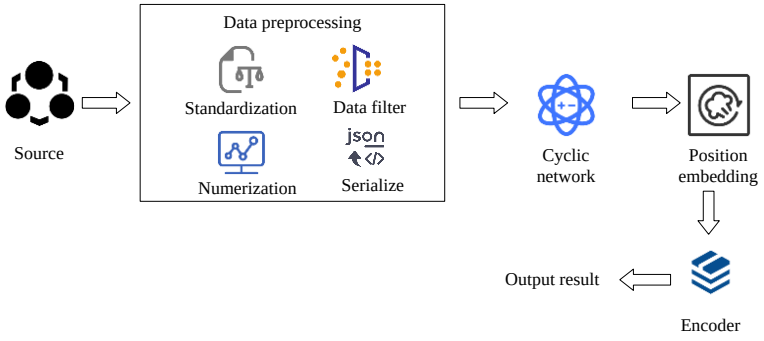


Figure 4: Prenetwork model of prediction model based on improved GRU.

layer. Dropout is a regularization technique widely used in DL. This keeps the NN from over-fitting and enhances the model's capacity for generalization. Its operation process can be defined as Equation (9) [24].

$$\begin{cases} z_i^{(l+1)} = b_i^{(l+1)} + w_i^{(l+1)} y^l \\ y_i^{(l+1)} = f\left(z_i^{(l+1)}\right) \end{cases} \tag{9}$$

In Equation (9), $w_i^{(l+1)}$ is the weight corresponding to the i th hidden unit in the $l+1$ th layer. f is the activation function. $z_i^{(l+1)}$ denotes the input of the i th hidden unit in the $l+1$ layer. $b_i^{(l+1)}$ is the bias corresponding to the i th hidden unit in the $l+1$ layer. The activation function used in improving GRU networks is the softmax function, which has high applicability in classification tasks. However, NSSP is an attack behavior classification problem. Therefore, this function is used as the activation function in the study. The expression of the function is shown in Equation (10).

$$p_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_i}} \tag{10}$$

In Equation (10), K is the values in the output vector. z_j is the j th value in the network output result. p_i is the probability of the current category. i denotes the current computational category. Since the core structure of the prediction model is the GRU, its parameters need to be optimized when training the model. The optimizer chosen for the study is small batch gradient descent. Small batch gradient descent can take advantage of matrix operations to perform parallel processing on data from small batches of samples. Moreover, small batch gradient descent can quickly reduce losses in the early stages of training, and by continuously adjusting parameters in the later stages, the model can better fit the data and finally achieve a good convergence state. Therefore, the study chose small batch gradient descent as the optimization algorithm [25]. By using a small batch of training samples from the training set (TrS), this optimizer first optimizes the NN's parameters. Moreover, the mean value of the gradient of that batch of samples is calculated to optimize the parameters [26]. Figure 5 depicts the overall NSSP flow based on MHSAM.

The NSSP using the model designed by the research needs to be started with data collection and pre-processing of the data set. The collected data needs to be divided into TrS and test set (TeS). To train the model, some of the samples are first forward propagated on the improved GRU network. The data samples start back-propagation training immediately after the end of forward propagation. Furthermore, the network parameters have been optimized based on the training outcomes. After the parameters are updated and optimized, the network performance is compared before and after the update. Networks with better generalization performance are retained until the maximum iterations is reached. The model must be tested once it has been trained. Input the TeS into the network retained after reaching the maximum number of iterations, perform NSSP and output the results. The pseudo-code for the research and design model is shown in Figure 6.

4 Results

4.1 Experimental environment and parameter settings

The dataset used in the study for training and testing the models constructed for the study is the UNSW-NB15 dataset. The UNSW-NB15 dataset covers various types of network attacks, including but not limited to DoS attacks, DDoS attacks, port scans, backdoor attacks, fuzz attacks, malware attacks, and more. This diversity allows researchers to test multiple attack scenarios in a single dataset. The traffic in the dataset is partly derived from the real network environment and partly generated by simulation. This combination method aims to provide data that is closer to the actual network environment, while maintaining the controllability and diversity of the

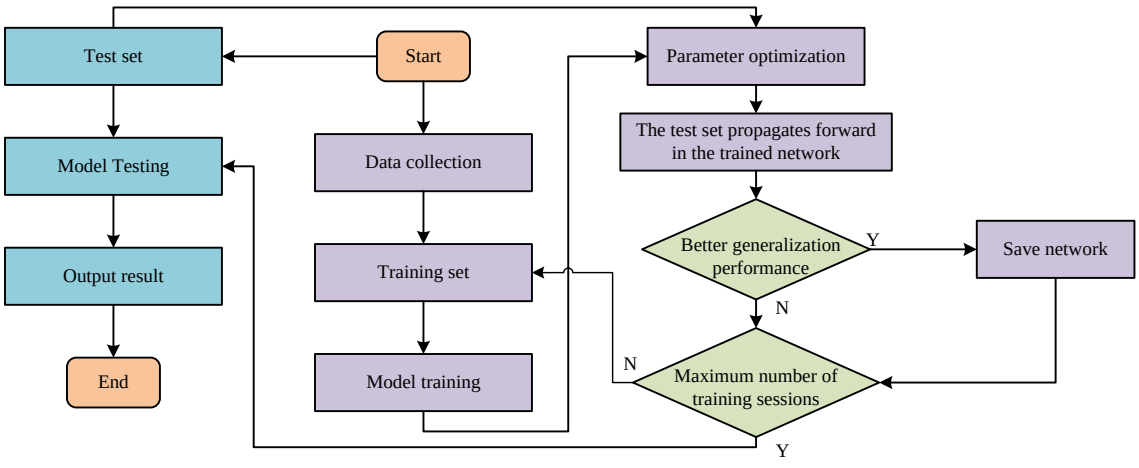


Figure 5: Network security state assessment means with MHSA framework and GRU.

```
# Define model parameters
num_neurons_circ_layer = 16  # Number of neurons in the recurrent layer
input_dim = 16  # Input dimension
num_heads = 2  # Number of attention heads
subvector_dim = 8  # Subvector dimension
num_neurons_encoder = 12  # Number of neurons in the encoder
num_neurons_feedforward = 28  # Number of neurons in the feedforward layer

# Data preprocessing
def preprocess_data(data):
    # Filter invalid data
    filtered_data = filter_invalid_data(data)
    # Convert to numerical format
    numerical_data = convert_to_numerical(filtered_data)
    # Normalize data
    normalized_data = normalize(numerical_data)
    # Transform to time series
    time_series_data = time_series(normalized_data)
    return time_series_data

# Model training
def train_model(train_data, train_labels, epochs, batch_size):
    for epoch in range(epochs):
        for batch in range(0, len(train_data), batch_size):
            batch_data = train_data[batch:batch + batch_size]
            batch_labels = train_labels[batch:batch + batch_size]
            batch_data = preprocess_data(batch_data)
            gru_output = gru_layer(batch_data, num_neurons_circ_layer)
            encoder_output = encoder_layer(gru_output, num_heads, num_neurons_encoder)
            predictions = prediction_layer(encoder_output, num_neurons_feedforward)
            loss = compute_loss(predictions, batch_labels)
            gradients = compute_gradients(loss)
            update_parameters(gradients)
        accuracy = evaluate_model(test_data, test_labels)
        print(f'Epoch {epoch + 1}, Accuracy: {accuracy:.4f}')
```

Figure 6: Pseudo-code for the study design model.

data. Each data set contains 42 characteristics covering various aspects of network traffic, such as duration, source and destination IP addresses, port numbers, protocol types, packet sizes, etc. The dataset has a total of 2, 540, 044 records stored in four CSV files. The dataset is separated into TeS and TrS subsets. 175,341 records are in the TrS, while 82,332 records are in the TeS. This dataset contains several common types of network attacks, which can comprehensively cover security threats in today's network environments. This allows the model to learn the characteristics of different types of attacks during the training process, improving the model's generalization ability. The attack behavior in the dataset is generated from actual network traffic and has high authenticity and integrity. This allows the model to be exposed to real network attack scenarios during the training process, improving the practicality and reliability of the model. The dataset is large in size and contains a large number of training and test samples, which can provide sufficient data to support the training and validation of the model. Meanwhile, the quality of the dataset is high, and the data cleaning and pre-processing work is relatively complete, reducing the impact of data noise on model training. There may be an imbalance in the sample size of different attack types in the dataset, with some attack types having a larger sample size while others have a smaller sample size. This can lead to over-fitting of the model to attack types with large sample sizes during training, and insufficient generalization ability to attack types with small sample sizes. As the size of the dataset increases, the computational complexity of the model increases linearly. Specifically, the computational complexity of data pre-processing, position coding, MHAM, GRU layer, encoder layer, and prediction layer is linearly related to the size of the dataset. Therefore, the demand for computational resources will increase significantly when the model processes large datasets. As the complexity of the network increases, the computational complexity of the model increases significantly. In particular, increasing the number of heads in the MHAM, the HS dimension of the GRU layer, and the HS dimension of the encoder layer will significantly increase the computational complexity. Therefore, as the model deals with more complex network structures, the demand for computational resources will increase significantly. Table 2 displays

Table 2: Experimental environment configuration

Item	Type	Item	Type
Operating system	Windows 10	Programming Language	Python
Memoryspace	16GB	Interpreter version	Python3.6
GPU	RTX 2060	Main libraries used	Tensorflow-gpu-1.15.0、pandas
CPU	i7-10750H CPU @ 2.60GHz	/	/

how the experimental environment is set up for the model's testing and training. In model training, the data standardization method used in the study is Z-Score normalization, and the sampling method is mixed sampling. The dropout rate is 0.5.

The study's enhanced GRU network is a deep neuron network where the algorithm's performance is directly impacted by the selection of hyper parameters. The study configures the algorithm hyper-parameters according to Hannan et al. and Khodabandelou et al. A total of 4 sets of hyper-parameter configurations are designed in the study, as shown in Table 3. The hyper-parameter configuration information of the research design can reflect the impact of different parameters on the model performance, so only 4 sets of hyper-parameter configurations need to be set. The hyper-parameters selected for the study include the number of recurrent layer neurons, the input dimension, the total number of heads, the sub-vector dimension, the number of encoder neurons, and the number of model parameters. The algorithm used to evaluate whether the hyper-parameter configuration is reasonable is the grid search algorithm.

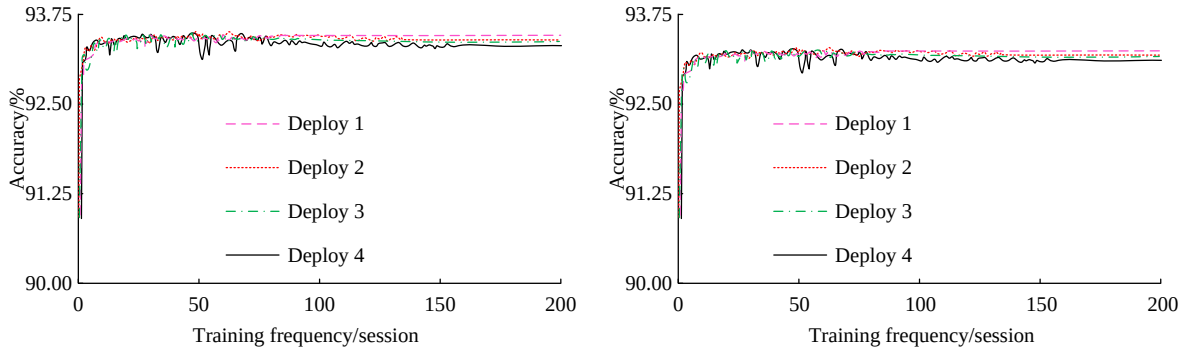
To prevent over-fitting and ensure the generalization ability of the model, this study adopted a cross-validation method. Cross-validation evaluates the performance of a model by dividing the training dataset into multiple small data sets and performing multiple training and validation on these small datasets. The training set is divided into k equally sized subsets. For each subset, it is imperative to utilize it as the validation set, while the remaining k-1 subsets are to be employed as the training set. The model is then trained on the training set, and its performance is evaluated on the validation set. The performance metrics are then calculated for each subset, and the average of these metrics is taken as the final performance evaluation result. Through cross-validation, it is possible to effectively prevent over-fitting of the model during training and ensure its generalization ability on new data. In this study, the number of folds k for cross-validation is set to 5, i.e., 5-fold cross-validation.

4.2 Algorithm performance analysis

Long short-term memory network (LSTM) is a classical prediction-like neuronal network with similar functions as GRU. The use of LSTM can verify the feasibility of improving the GRU network through research, and also verify whether the selection of the network is correct. One-way and two-way training are employed in the study to confirm the efficacy of the network architecture. Figure 7 compares the two enhanced networks' performances and displays the findings.

Table 3: Improved hyper-parameter configuration of the GRU network

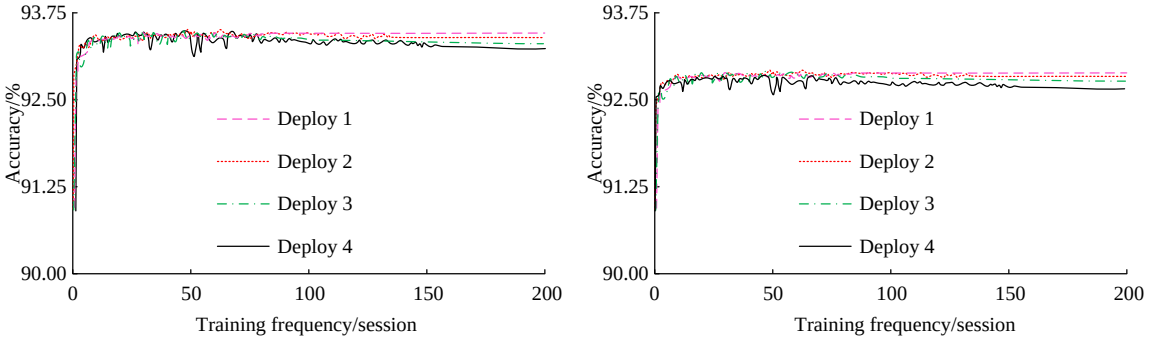
Deploy	Number of neurons in the circulating layer	Enter the dimension	Totalheads counts	Subvector dimensions	Number of encoder neurons	Parameter quantity
1	16	16	2	8	12	14970
2	16	16	8	16	12	14970
3	32	32	2	8	12	35498
4	48	48	8	6	12	61658



(a) Experimental results of improving GRU network

(b) Experimental results of improving LSTM network

Figure 7: Comparison of accuracy of proposed GRU and improved LSTM during one-way training.



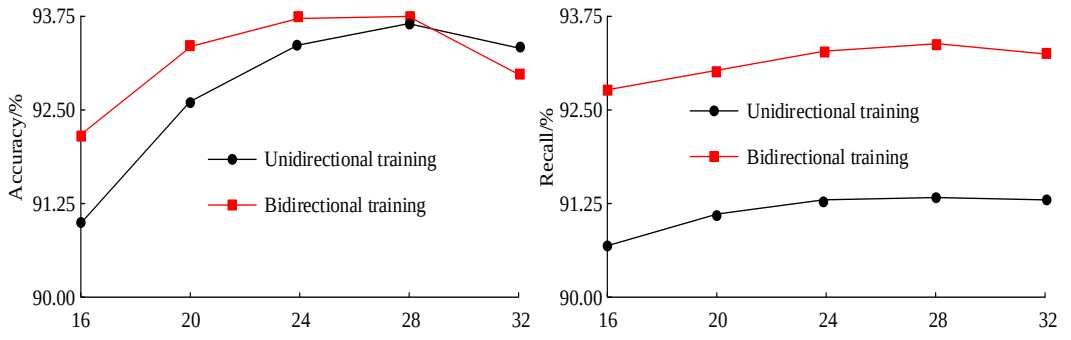
(a) Experimental results of improving GRU network

(b) Experimental results of improving LSTM network

Figure 8: Comparison of accuracy of proposed GRU network with improved LSTM network during bidirectional training.

In Figure 7, training frequency/session refers to the number of model training times. The experimental outcomes of the enhanced GRU network under various hyper parameter configurations are displayed in Figure 7(a). The accuracy remains around 93.5% under all four hyper-parameter configuration schemes. The fourth set of hyper-parameter configuration schemes shows significant over-fitting during the training process, and the model accuracy decreased after reaching its maximum value. This indicates that increasing the number of hyper-parameters may lead to an increase in model complexity, thereby increasing the risk of over-fitting. Figure 7(b) shows the outcomes of the improved LSTM network under different hyper parameter configurations. The training results of the enhanced GRU network show a change trend that is largely compatible with the experimental findings of this network under various hyper parameter combinations. However, compared to the enhanced GRU network, the model's accuracy is marginally less. This finding suggests that GRU and LSTM demonstrate comparable performance in network situation prediction tasks. However, GRU may exhibit superior efficiency in processing long sequence data. Figure 8 illustrates how the upgraded LSTM and GRU networks' accuracy changed after they are trained in both directions.

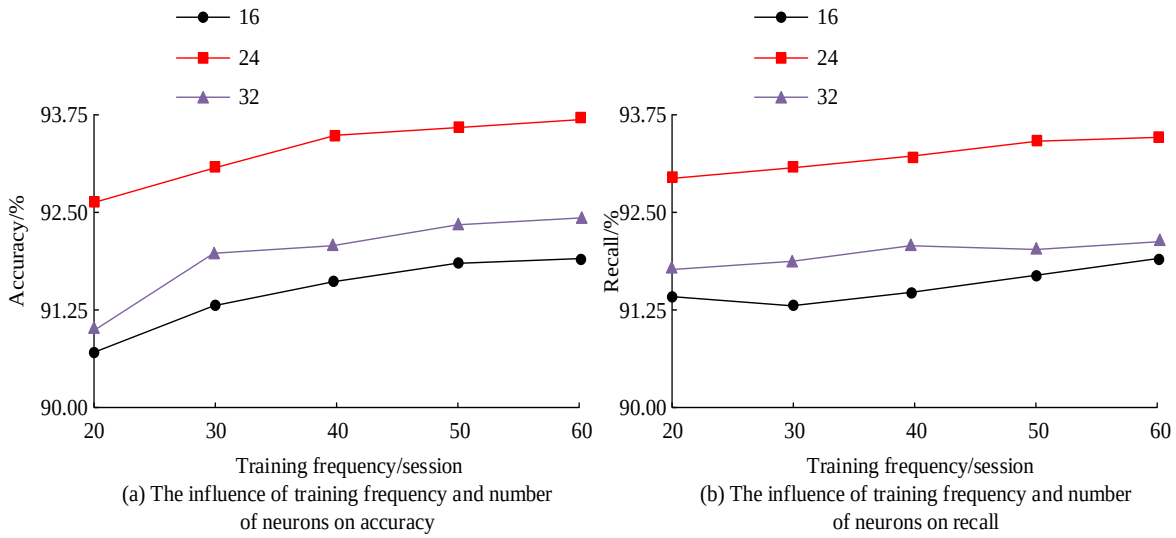
Figure 8(a) shows the change in accuracy when the proposed GRU network is trained using two-way training. The over fitting of the model is more obvious when the 4th hyper parameter configuration scheme is used. At this time, the performance degradation after model over fitting is also more obvious. In the 3rd hyper-parameter configuration scheme, the model also shows a similar over-fitting phenomenon as in the 4th hyper-parameter configuration scheme. Further increasing the number of feed-forward neurons will result in a decrease in model performance. This indicates that increasing the number of neurons in the feed-forward layer can lead to an increase in model complexity, thereby increasing the risk of over-fitting. Figure 8(b) shows the change in accuracy of the improved LSTM network when two-way training is used. The decrease in model accuracy is more serious after using bidirectional training, and the over fitting phenomenon is also more obvious. The network reaches its highest accuracy at the 50th training, which is about 93.00%. Conversely, as training times grow, the accuracy of the model progressively declines to approximately 92.5%. The performance of the algorithm is also directly impacted by the feed forward layer's neurons. This indicates that bidirectional training may increase the complexity of the model, leading to more severe over-fitting. The study examines how the feed forward layer neurons affect the upgraded GRU network's performance in both unidirectional and bidirectional training. The results are shown in Figure 9.



(a) The variation of model accuracy with the number of feedforward neurons under different training methods

(b) The variation of model recall with the number of feedforward neurons under different training methods

Figure 9: Effect of the number of FLNs on the Proposed GRU.



(a) The influence of training frequency and number of neurons on accuracy

(b) The influence of training frequency and number of neurons on recall

Figure 10: Effect of training methods and the number of feed-forward neurons on proposed GRU.

The impact of the feed forward layer's neuron count on the accuracy of the network under various training strategies is depicted in Figure 9(a). Using unidirectional training, the model accuracy is highest at about 93.60% when the feed forward layer neurons (FLNs) is 28. With further increase in the FLNs, there is a decrease in the model performance. The model accuracy drops to about 92.7% when the FLNs is 32 numbers. When bi-directional training is applied, the model's accuracy changes in a way that is consistent with what happens when uni-directional training is applied. On the other hand, the bidirectional training model has a better accuracy of up to 93.7%. Further increasing the number of feed-forward neurons will result in a decrease in model performance. This indicates that increasing the number of neurons in the feed-forward layer can lead to an increase in model complexity, thereby increasing the risk of over-fitting. The effect of the feed forward layer neurons on the network recall under various training strategies is depicted in Figure 9(b). The neurons have little effect on the model recall. This suggests that an increase in the number of neurons in the feed-forward layer exerts minimal influence on the recall rate of the model, yet it may potentially result in a reduction in the model's accuracy. Figure 10 displays the effects of the feed forward layer neurons on the model's performance at various training intervals.

In Figure 10, Training frequency/session refers to the number of model training times. Figure 10(a) displays the effect of the model training times and the FLNs on the model accuracy. When the FLNs is equal, the model test accuracy increases with the increasing training times. When the FLNs is 16, the highest model test accuracy is about 91.8%. When the neurons in the feed forward layer is 24, the highest model test accuracy is about 92.2%. The maximum accuracy of the model test is around 93.6% when there are 32 neurons in the feed forward layer. The maximum accuracy of the model test is around 93.6% when there are 32 neurons in the feed forward layer. This finding suggests that an optimal number of training iterations can enhance the model's performance. However, it is important to note that excessive training iterations may lead to over-fitting, which can compromise the model's generalization ability. Figure 10(b) shows the effect of the model training times and the neurons in the feed forward layer on the model recall. The model training times has basically no effect on the model recall. When the training times is the same, the more the number of FLNs, the higher the model recall. When the FLNs is 16, the highest model recall is about 91.5%. When the FLNs is 32, the highest model recall is around 93.2%. This indicates that increasing the number of feed-forward neurons has a positive impact on the recall rate of the model, but may lead to a decrease in the accuracy of the model. The number of neurons in the feed-forward layer can improve the feature extraction ability of the model, allowing it to learn more complex feature representations. The number of neurons in the feed-forward layer increases the complexity of the model, making it easier to fit to the training data. Increasing the complexity of the model can improve its performance on the training set, but can also lead to over-fitting. Regularization techniques such as dropout can effectively prevent over-fitting and improve the generalization ability of the model. In the last section of the study, performance of several NSS algorithms at various TSs are compared. Table 4 presents the findings. In selecting algorithms for comparison, this article referred to the research results of various scientists. Decision trees, deep NNs, convolutional NNs, transformers, and GRUs are currently the most widely used and effective models in this field. Therefore, these models are chosen as the control models for the algorithm in this study.

In Table 4, IGRU represents the network proposed by the research institute, the improved GRU algorithm designed by the study is always higher than the other algorithms in terms of accuracy, precision and F1 index, regardless of the TS. Moreover. When the TS is 5, the accuracy of the improved GRU algorithm can reach 93.458%, the precision can reach 91.892% and the F1 value can reach 91.892%. At a TS of 15, the accuracy of the improved GRU algorithm can reach 93.512, the

Table 4: Comparison of performance of NSSP algorithm under different TSs

Index		Decision tree	Deep neural network	Convolutionalneural network	Transformer	IGRU	GRU
5	**Accuracy (%)**	88.574***	93.248*	93.216*	93.268*	93.458	92.658**
	Precision (%)	89.195***	90.748*	90.769*	90.586*	91.892	90.867**
	F1 (%)	88.748***	91.747*	91.734*	91.766*	91.892	90.658**
15	**Accuracy (%)**	88.168***	93.189*	93.265*	93.364*	93.512	92.684**
	Precision (%)	88.857***	90.066*	90.636*	90.748*	91.016	90.052**
	F1 (%)	88.467***	91.621*	91.693*	91.923*	92.068	90.596**

Note: Compared to IGRU: *P<0.05. **P<0.01. ***P<0.001

precision can reach 91.016% and the F1 value can reach 92.068%. The performance of decision tree models is always lower than that of research design models at different TSs, which is significant at the 0.1% level. Deep NNs, convolutional NNs, and transformers are also always lower than research design models, and are significant at the 5% level. The performance of the GRU models is also significantly lower than that of the research design models, and is significant at the 1% level. The research design method outperforms other models in terms of accuracy, precision, and F1 value, and is significantly superior to other methods. The NSSP method designed by the study can accurately predict the NSS changes.

5 Discussion

The MHAM-GRU NSSP model proposed in the study significantly outperformed the existing best methods in terms of performance. The MHAM-GRU model achieved an accuracy of about 93.5% on the UNSW-NB15 dataset, which was significantly higher than other methods such as traditional GRU and LSTM networks. For example, the accuracy of the standard GRU model was about 92.658%, while the accuracy of the LSTM network was about 93.00%. The reason why the MHAM-GRU model could achieve better performance was attributed to the integration of MHAMs, which allowed the model to simultaneously focus on different parts of the input data and weigh their importance in different ways.

The MHAM-GRU model increased the depth and complexity of the model by introducing the encoder part of the Transformer architecture. However, this increased complexity brought better feature extraction capabilities and training effectiveness. Residual connections and layer normalization in the Transformer architecture could effectively prevent gradient vanishing problems and improve the training performance of the model. In addition, the introduction of positional encoding allowed the model to better handle the positional information of sequence data, further improving the performance of the model.

6 Conclusion

In order to improve network information security and enhance the network's defense against attack behavior, it is proposed to use MHSAM to improve the GRU network. The model is based on GRU network, with Transformer as the framework, and combined with MHSAM. Furthermore, the model adopts Dropout and full connectivity to predict the NSS. The results indicated that when a bidirectional training method was used to train the model constructed for research, the model accuracy could be maintained at 93.7%. The highest accuracy of the model remained unchanged when a bidirectional training approach was used. When the training times was fixed and the neurons in the feed forward layer was less than 28, the model test accuracy increased with the increase in the neurons in the feed forward layer, up to about 93%. When the TS is 15, the proposed method achieved an accuracy of 93.512%, a precision of 91.016%, and an F1 value of 92.068% in NSSP. The study's NSSP approach could successfully increase the accuracy of predicting the NSS, thereby improving the network's resistance to various types of attacks. Although this study has achieved significant results in predicting the NSS, there are still some limitations. The model exhibits over-fitting during the training process, especially when the number of training iterations or feed-forward neurons is increased, which may lead to a decrease in model performance. In the future, research will use a regularization term added to the loss function to penalize the weights of the model and prevent excessive weight values, thereby reducing the complexity of the model. This will alleviate the problem of over-fitting.

Author contributions

Jian Chen provided the concept, designed the experiment and wrote the manuscript; Huanqiang Bian validated the experiment, analyzed the data; Hao Liang revised the manuscript critically. All authors reviewed and approved this submission.

Data availability statement

All data generated or analyzed during this study are included in this article. Further enquiries can be directed to the corresponding author.

Ethics approval

An ethics statement was not required for this study type, no human or animal subjects or materials were used.

References

[1] J. Zhao, R. Masood, and S. Seneviratne, “A review of computer vision methods in network security,” IEEE Communications Surveys & Tutorials, vol. 23, no. 3, pp. 1838-1878, 2021. https://doi.org/10.1109/COMST.2021.3086475

[2] H. I. Kure, S. Islam, and H. Mouratidis, “An integrated cyber security risk management framework and risk predication for the critical infrastructure protection,” Neural Computing and Applications, vol. 34, no. 18, pp. 15241-15271, 2022. https://doi.org/10.1007/s00521-022-06959-2

[3] M. Li, "Application of GAN-Based Data Encryption Technology in Computer Communication System," Informatica, vol. 48, no. 15, 2024. https://doi.org/10.31449/inf.v48i15.6390

[4] M. A. Al-Shareeda, S. Manickam, M. A. Saare, and N. B. Omar, "Sadetection: Security mechanisms to detect SLAAC attack in IPv6 link-local network," Informatica, vol. 46, no. 9, 2023. https://doi.org/10.31449/inf.v46i9.4441

[5] J. Luo and R. Zhu, "Network Embedding Technology Based on Breadth Learning for Information Extraction and Review in Social Media," Informatica, vol. 48, no. 19, 2024. https://doi.org/10.31449/inf.v48i19.6544

[6] A. Islam, F. Othman, N. Sakib, and H. M. H. Babu, “Prevention of shoulder-surfing attack using shifting condition with the digraph substitution rules,” Artificial Intelligence and Applications, vol. 1, no. 1, pp. 58-68, 2023.https://doi.org/10.47852/bonviewAIA2202289

[7] Z. Zhang, "SD-WSN Network Security Detection Methods for Online Network Education," Informatica, vol. 48, no. 21, 2024. https://doi.org/10.31449/inf.v48i21.6257

[8] Z. Chen, “Research on internet security situation awareness prediction technology based on improved RBF neural network algorithm,” Journal of Computational and Cognitive Engineering, vol. 1, no. 3, pp. 103-108, 2022. https://doi.org/10.47852/bonviewJCCE149145205514

[9] L. Xu, X. Zhou, Y. Tao, L. Liu, X. Yu, and N. Kumar, “Intelligent security performance prediction for IoT-enabled healthcare networks using an improved CNN,” IEEE Transactions on Industrial Informatics, vol. 18, no. 3, pp. 2063-2074, 2021. https://doi.org/10.1109/TII.2021.3082907

[10] M. Zhang and N. Chen, "Chaotic Encryption-Based Network Robot for Indoor Security and Remote Video Monitoring," Informatica, vol. 48, no. 14, 2024. https://doi.org/10.31449/inf.v48i14.6073

[11] H. I. Kure, S. Islam, M. Ghazanfar, A. Raza, and M. Pasha, “Asset criticality and risk prediction for an effective cybersecurity risk management of cyber-physical system,” Neural Computing and Applications, vol. 34, no. 1, pp. 493-514, 2022. https://doi.org/10.1007/s00521-021-06400-0

[12] Q. Ni, J. C. Ji, and K. Feng, “Data-driven prognostic scheme for bearings based on a novel health indicator and gated recurrent unit network,” IEEE Transactions on Industrial Informatics, vol. 19, no. 2, pp. 1301-1311, 2022. https://doi.org/10.1109/TII.2022.3169465

[13] W. Zhang, H. Li, L. Tang, X. Gu, L. Wang, and L. Wang, “Displacement prediction of Jiuxianping landslide using gated recurrent unit (GRU) networks,” Acta Geotechnica, vol. 17, no. 4, pp. 1367-1382, 2022. https://doi.org/10.1007/s11440-022-01495-8

[14] W. Shu, K. Cai, N. N. Xiong, “A short-term traffic flow prediction model based on an improved gate recurrent unit neural network,” IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 9, pp. 16654-16665, 2021. https://doi.org/10.1109/TITS.2021.3094659

[15] H. Lin, A. Gharehbaghi, Q. Zhang, S. S. Band, H. T. Pai, K. W. Chau, et al. Time series-based groundwater level forecasting using gated recurrent unit deep neural networks. Engineering Applications of Computational Fluid Mechanics, vol. 16, no. 1, pp. 1655-1672, 2022. https://doi.org/10.1080/19942060.2022.2104928

[16] J. He and J. Yang, "Network security situational level prediction based on a double-feedback Elman model," Informatica, vol. 46, no. 1, 2022. https://doi.org/10.31449/inf.v46i1.3775

[17] D. A. Neu, J. Lahann, and P. Fettke, “A systematic literature review on state-of-the-art deep learning methods for process prediction,” Artificial Intelligence Review, vol. 55, no. 2, pp. 801-827, 2022.https://doi.org/10.1007/s10462-021-09960-8

[18] K. Yu, L. Tan, S. Mumtaz, S. Al-Rubaye, A. Al-Dulaimi, A. K. Bashir, and F. A. Khan, “Securing critical infrastructures: deep-learning-based threat detection in IIoT,” IEEE Communications Magazine, vol. 59, no. 10, pp. 76-82, 2021.https://doi.org/10.1109/MCOM.101.2001126

[19] S. Capobianco, L. M. Millefiori, N. Forti, P. Braca, and P. Willett, “Deep learning methods for vessel trajectory prediction based on recurrent neural networks,” IEEE Transactions on Aerospace and Electronic Systems, vol. 57, no. 6, pp. 4329-4346, 2021.https://doi.org/10.48550/arXiv.2101.02486

[20] V. Veeramsetty, K. R. Reddy, M. Santhosh, A. Mohnot, and G. Singal, “Short-term electric power load forecasting using random forest and gated recurrent unit,” Electrical Engineering, vol. 104, no.

1, pp. 307-329, 2022.https://doi.org/10.1007/s00202-021-01376-5

[21] R. Rouhi Ardeshiri, and C. Ma, “Multivariate gated recurrent unit for battery remaining useful life prediction: A deep learning approach,” International Journal of Energy Research, vol. 45, no. 11, pp. 16633-16648, 2021.https://doi.org/10.1002/er.6910

[22] J. Zhou, Y. Qin, J. Luo, S. Wang, and T. Zhu, “Dual-thread gated recurrent unit for gear remaining useful life prediction,” IEEE Transactions on Industrial Informatics, vol. 19, no. 7, pp. 8307-8318, 2022. https://doi.org/10.1109/TII.2022.3217758

[23] W. Chen, D. Sharifrazi, G. Liang, S. S. Band, K. W. Chau, and A. Mosavi, “Accurate discharge coefficient prediction of streamlined weirs by coupling linear regression and deep convolutional gated recurrent unit,” Engineering Applications of Computational Fluid Mechanics, vol. 16, no. 1, pp. 965-976, 2022. https://doi.org/10.1080/19942060.2022.2053786

[24] M. A. Hannan, D. N. T. How, M. B. Mansor, M. S. H. Lipu, P. J. Ker, and K. M. Muttaqi, “State-of-charge estimation of li-ion battery using gated recurrent unit with one-cycle learning rate policy,” IEEE Transactions on Industry Applications, vol. 57, no. 3, pp. 2964-2971, 2021. https://doi.org/10.1109/TIA.2021.3065194

[25] F. Zeng, R. Tang, and Y. Wang, "User Personalized Recommendation Algorithm Based on GRU Network Model in Social Networks," Mobile Information Systems, vol. 2022, no. 1, pp. 1487586, 2022. https://doi.org/10.1155/2022/1487586

[26] G. Khodabandelou, W. Kheriji, F. H. Selem, “Link traffic speed forecasting using convolutional attention-based gated recurrent unit,” Applied Intelligence, vol. 51, no. 4, pp. 2331-2352, 2021. 10.1007/s10489-020-02020-8