EAI Endorsed Transactions
on Collaborative Computing

Research Article EAI.EU

# A Framework for Predicting Network Security Situation Based on the Improved LSTM

Shixuan Li[1,2], Dongmei Zhao[1,2,]* and Qingru Li[1,2]

[1]College of Computer and Cyber Security, Hebei Normal University, Shijiazhuang 050024, China
[2]Hebei Key Laboratory of Network & Information Security, Shijiazhuang 050024, China

## Abstract

In recent years, raw security situation data cannot be utilized well by fully connected neural networks. Generally, a cyber infiltration is a gradual process and there are logical associations between future situation and historical information. Taking the factors into account, this paper proposes a framework to predict network security situation. According the needs of this framework, we improve Long Short-Term Memory (LSTM) with Cross-Entropy function, Rectified Linear Unit and appropriate layer stacking. Modules are designed in the framework to transform raw data into quantitative results. Finally, the performance is evaluated on KDD CUP 99 dataset and UNSW-NB15 dataset. Experiments prove that the framework built with the improved LSTM has better performance to predict network security situation in the near future. The framework achieves a relatively practical prediction of network security situation, helping provide advanced measures to improve network security.

**Keywords:** network security situation, deep learning, situation prediction, neural network, LSTM

Received on 09 April 2020, accepted on 10 June 2020, published on 12 June 2020

doi: 10.4108/eai.12-6-2020.165278

## 1. Introduction

Various networks play an indispensable role in modern society. In Cyberspace, situation awareness is an important approach to ensure network security. Endsley et al. first introduced the concept of situation awareness explicitly in 1988 [1]. It is defined as the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning and the projection of their status in the near future. She constructed a three-layer model to describe and guide situation awareness. Since then, traditional situation awareness has been broadly used in consideration of human factors in aviation and military confrontation. In 1999, Bass et al. proposed the next-generation network intrusion detection system that required a large number of heterogeneous distributed sensors as data sources for integration to realize network situation awareness [2]. And they built a network situation awareness model based on integration of data from sensors.

On the basis of Endsley's conceptual model and Bass's functional model, researchers created many network security situation awareness models later [3]. The structure and name of models are different, but their functions are divided into three levels.

The first layer of situation awareness focuses on extracting indicators that can be prepared to reflect network security situation from massive data of network parameters and related data generated by operation and management process. Related necessary data is gathered through the first layer. The function of the second layer is comprehension and evaluation, which aims to interpret the extracted data as information related to current network security status. We take the data from the first layer as input and generate information to describe current security situation. It concentrates on the changing trend of the network security situation rather than a single intrusion alarm. The second layer will present the current

*Corresponding author. Email:zhaodongmei666@126.com security situation wholly and macroscopically. Projection, the third layer of situation awareness, is to predict the evolution of network security situation in the near future, based on current situation information.

In the past, administrators would patch the information systems after hackers had hacked the targets. Systems could continue to provide services after being patched. However, attackers had achieved their goals. They might have got the data and interrupted the service during a specific period of time. Facing more and more cyber-attacks, administrators need to consider how to be aware of security situation in the near feature effectively. In other words, it is necessary to learn how the network state will evolve and what cyber-attacks may occur. This is the aim of security situation prediction. With prediction, administrators can get early warning and take precautions to respond to upcoming threats effectively. Data leakages and business interruptions caused by attackers may be avoided. Correct prediction of the security situation can guide us to protect networks. The work of this paper mainly focuses on projection.

The contributions of this paper are as follows: (1) We propose a framework to predict network security situation, which realizes the transformation from raw data to quantitative results. (2) According to the needs of the framework, we improve Long Short-Term Memory (LSTM) with Cross-Entropy (CE) function, Rectified Linear Unit (ReLU) and appropriate stacking of layers.

The remainder of this paper is organized as follows: Section 2 reviews related work on situation awareness and some applications of LSTM. Section 3 introduces the theoretical basis of LSTM applications. Section 4 details the improvement scheme and the framework for situation prediction. Section 5 presents experimental procedures and simulation results. Section 6 concludes the paper.

## 2. Related Work

With massive data, prediction of network security situation aims to obtain network security situation in the near future through data analysis and integration. Situation awareness not only generates a single alarm, but also presents overall security status from a macro perspective. Therefore, we can achieve a comprehensive awareness of network security situation [4].

Methods to be aware of network security situation can be roughly divided into the following four categories [3]:

### 2.1. Awareness Methods Based on Mathematical Models

Mathematical model-based awareness methods are to comprehensively analyse various factors impacting on the security situation with mathematic tools. Generally speaking, the impact weight is determined by experts' experience and Analytic Hierarchy Process (AHP) [5]. The evaluation function *f* is described as formula (1).

$$S = f(f_1, f_2, f_3, \dots, f_n), f_i \in F(n) \tag{1}$$

In formula (1), $f_i$ is the *i*-th feature in the feature set *F*. The function of *f* is to map feature set *F* to security situation state space *S*.

These methods represented by weighted average method can easily map data indicators to security situation through mathematical tools. Xiao et al. presented a security situation awareness method to comprehend and predict the security situation by calculating security distance [6]. Chen et al. established a quantitative hierarchical model to estimate network threats [7]. However, weight is determined by particular methods and experience rather than universal criteria, which lacks an objective basis. Xu et al. combined mathematical models with information theory to judge if the network is secure [8], but the proposed scheme couldn't determine the types of attacks.

### 2.2. Awareness Methods Based on Rule Inference

Awareness methods based on rule inference are represented by Dempster-Shafer (D-S) evidence theory. To achieve the purpose, the theory states the influence of each factor on the overall security situation, and then uses evidence synthesis rules to generate supportiveness of the whole feature set to different security situations. Evidence synthesis rules are the core of the evaluation process. Qiu et al. used the exponential weight D-S to complete information fusion [9]. Wei et al. developed a situation awareness model on the basis of information fusion, using the modified D-S evidence theory [10]. They alleviated the problem of evidence conflict to some extent, but real-time situation awareness wasn't achieved.

### 2.3. Awareness Methods Based on Probability Statistics

Bayesian Network (BN) and Hidden Markov Model (HMM) are widely used as network security situation awareness methods based on probability statistics. Starting from statistical characteristics of prior knowledge and considering uncertainty of security situation information, BN and HMM constructed awareness models to be aware of security situation in networks. Hu et al. studied how to quantify network security situation and discussed the prediction of attacks with the dynamic Bayesian attack graph [11]. Wang et al. established a solution with BN to assess information security risk [12]. HMM was also utilized by Zhang et al. to guide network security situation awareness and proved to have a good performance [13].

Awareness methods based on probability statistics can learn new prior knowledge and evidence. The process of state transition is obvious. However, models for long-term awareness require a large amount of data storage and computation. And it is easy to cause dimensional explosions.

## 2.4. Awareness Methods Based on Neural Networks

Neural networks have advantages of high adaptability and nonlinear processing capacity. Meanwhile, a large number of connections and weight matrices between neurons in neural networks form structural redundancy, which makes neural networks have strong fault tolerance and self-organization ability. Back Propagation (BP) neural network has been broadly used in network security situation awareness. Chen et al. improved BP neural network by simulated annealing algorithm to assess cyberspace situation [14]. Zhu et al. tried CE loss function in the training of BP neural layers [15]. Jin et al. combined particle swarm optimization with BP neural network to predict network security situation in electric power telecommunication [16]. The aforementioned methods are intended to alleviate problems existing in BP neural network, such as falling into local minimum values and slow training efficiency.

Zhang et al. created an improved application on the basis of Convolutional Neural Network (CNN) [17]. Compared with the fully connected network structure of BP neural network, sparse connections and weight sharing of CNN improve the generalization ability of models while reducing training time. Similar to BP neural network, a CNN also has its structural problems when used for network security situation awareness. Data indicators in the evaluation of network security situation come from the extraction of situation factors. At present, situation factors are time-series data from multi-factor integration. The structure of CNN makes it better to cope with partly associated data rather than time-series data.

To summarize, related work either only considered logical associations, or achieved real-time awareness without time associations. In brief, there isn't real-time situation awareness with the association between past and future to predict situation trends in the near future. Without time associations, there will be less information to make an accurate prediction. Without real-time awareness, administrators can't take precautions further. Accordingly, further research needs to be done when it comes to real-time situation awareness with associations.

At present, deep neural networks and new network structures have been used in applications to process time-series data. Li et al. extracted features of air combat situation by deep Q network model algorithm [18]. Dou et al. gave a method to detect anomaly with LSTM [19]. Wielgosz et al. used LSTM to monitor the large hadron collider superconducting magnets [20]. Galih et al. researched different structures of LSTM for weather forecasting [21]. Guedes et al. presented the results for a classification of chronic laryngitis with LSTM [22]. From the examples, we can find that deep neural networks and LSTM are appropriate for processing time-series data and make better results. Security situation is related to time-series data, so LSTM is theoretically feasible for situation prediction.

# 3. LSTM Network

LSTM network is designed to achieve long-term dependence. It has been used in natural language processing, event analysis, and prediction.

## 3.1. Structure of LSTM Network

LSTM network is a special form of Recurrent Neural Network (RNN). Both LSTM and RNN have a chain form of repeated artificial neural network modules. The structure of a LSTM cell is shown in Figure 1. Arrow lines indicate the flow of data in the cell. $f_t$ is the forget gate. $i_t$ and $d_t$ compose the input gate. $o_t$ is the output gate. At time $t$, the input is $X_t$, and the output is $H_t$. $C_t$ is the cell state. $H_{t-1}$ is the output at time $t\text{-}1$, and $C_{t-1}$ is the cell state at time $t\text{-}1$.

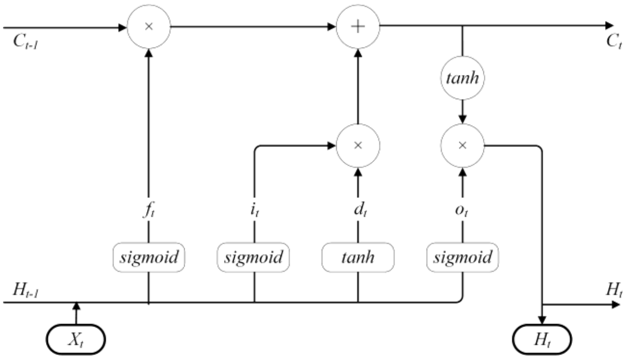


**Figure 1.** The structure of a LSTM cell

The cell state, data flow and the changing process of cell state are critical. Compared with RNN, LSTM forms a chain structure through $C_t$ and its changes, while maintaining fewer linear interactions so that long-term information can be transmitted steadily.

## 3.2. Forward Propagation of LSTM

Three structures called gates play different roles in forward propagation. In the cell state, LSTM manages long-term memory through forget gate shown as formula (2).

$$f_t = sigmoid\left(W_f \cdot [H_{t-1}, X_t] + b_f\right) \quad (2)$$

In formula (2), $W_f$ and $b_f$ are the weight matrices and the biases of the forget gate.

Input gate decides the information added to long-term memory. The sigmoid layer is as formula (3).

$$i_t = sigmoid(W_i \cdot [H_{t-1}, X_t] + b_i) \tag{3}$$

In formula (3), $W_i$ and $b_i$ are the weight matrices and the biases of the input gate. It determines which parts of the information need to be updated. The tanh layer is described as formula (4).

$$d_t = tanh(W_d \cdot [H_{t-1}, X_t] + b_d) \tag{4}$$

In formula (4), $W_d$ and $b_d$ are the weight matrices and the biases for generating new memory. $d_t$ is the new information generated.

With the aforementioned parameters, input gate can update long-term memory in the cell state, as shown in formula (5).

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

Output gate selectively outputs long-term memory in the cell state according to formula (6).

$$o_t = sigmoid(W_o \cdot [H_{t-1}, X_t] + b_o) \tag{6}$$

In formula (6), $W_o$ and $b_o$ are the weight matrices and the biases of the output gate.

Finally, cell state is processed by the tanh function and outputs information according to the strategy chosen by output gate. The mathematical description of this process is as formula (7).

$$H_t = o_t * tanh(C_t) \tag{7}$$

The tanh function is shown in formula (8).

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{8}$$

Complex functions make LSTM spend much time to compute. Compared with the sigmoid function that has one power operation, there are four power operations in the tanh function. Therefore, we can try to use sigmoid functions instead of the two tanh operations in LSTM. In this way, we can reduce six power operations in each LSTM cell. Compared with the sigmoid function, a better approach will be discussed specifically in the next section.

### 3.3. Back Propagation of LSTM

LSTM gets 8 parameters through training. They are weight matrices and biases of forget gate, input gate, generating new memory and output gate, described as $W_f$, $b_f$, $W_i$, $b_i$, $W_d$, $b_d$, $W_o$, $b_o$. Weight matrices are divided into $W_{fh}$, $W_{fx}$, $W_{ih}$, $W_{ix}$, $W_{dh}$, $W_{dx}$, $W_{oh}$, $W_{ox}$ in back propagation.

Back propagation of LSTM proceeds in two directions. One propagation is along time series and the other is towards the upper layers.

Through the back propagation, LSTM is trained with labeled data. There are many parameters and a large amount of calculation, so time cost will inevitably increase. We need to discuss how to build real-time situation awareness with it. In other words, we need to reduce the time cost to a lower level. Secondly, a simple LSTM cell cannot learn complex nonlinear relationship well. And LSTM can only finish abstract mapping of data features. It needs to be combined with other structures to predict situation. Finally, the data organizing in the field of cyber security is different from that in the field of natural language processing. LSTM can't output what we want to get directly. What the trained LSTM neural network is used for depends on what we use to train it and what data we enter. How to get the information we want through LSTM needs to be considered. Thus, we need to do some optimization and design to make it more suitable for predicting situation.

## 4. Network Security Situation Prediction Based on the Improved LSTM

It is suitable for LSTM to provide deep learning analysis solutions for serialized data due to its structural characteristics. With the gates and the cell state of LSTM, it establishes a link between past data and future situation so that it can get more information from past attacks to predict future security situation more accurately. Therefore, we can apply the improved LSTM to network security situation awareness. This section mainly discusses how to use the improved LSTM to predict security situation of networks in the near future.

### 4.1. The Improved LSTM Network Structure

Network statistics related to network security situation can be organized and processed in time series. Besides, network attacks can't be done in a short time. Behaviours at different stages are reflected in the relevant data, and serialized internal logical associations are generated. Although the structure of LSTM forms its inherent advantages in processing serialized data, a single LSTM layer is not effective enough for network security situation awareness. A successful network attack consists of several stages of the operation. An attack step may generate multiple network flow records. A flow record may contain numerous descriptive attributes and statistical characteristics. It is very difficult for the single-layer LSTM to learn features in complex scenarios.

Therefore, we build an improved LSTM network structure for security situation awareness. First, we increase the depth of the LSTM network by using a three-layer LSTM stack. The last layer of the LSTM network is then directly connected to a fully connected neural network layer whose activation function is softmax. The improved LSTM network structure is shown in Figure 2.

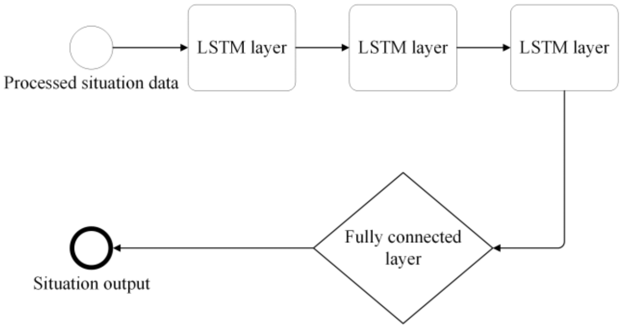


**Figure 2.** The core structure of the improved LSTM network

Every LSTM layer in the stack is the same. But through the stack of LSTM layers, the trained model can extract data features of different network security situations more accurately so that it can improve abstract mapping of situation data. Increasing the network's depth can optimize the neural network structure to a certain extent. The number of neurons and training time are reduced within one layer. It has a certain effect on improving the performance and efficiency of the neural network. As the number of LSTM layers increases, the efficiency and time cost of the model both increase. In practice, we find that the three-layer structure can balance cost and performance. It achieved better results. With more layers, time cost will be unacceptable. With less layers, it can't achieve the required ability to process features. The fully connected layer can map abstract features learned by pre-existing LSTM stack to situation space. It works as a classifier, which allows the model to output predicted cyber-attacks and situation values quantified according to certain standards. Appropriate redundancy of parameters in the fully connected layer also guarantees generalization and migration of the model. However, this structure further increases the time cost. To set up real-time situation awareness, there should be some measures to improve operation efficiency.

## 4.2. The Improved LSTM Cell Based on ReLU

The functions in a LSTM cell mainly include tanh and sigmoid. Sigmoid function is as formula (9).

$$sigmoid(x) = \frac{1}{1+e^{-x}} \tag{9}$$

Compared with sigmoid, tanh performs four times more exponentiation. During an execution of a LSTM cell, sigmoid and tanh are executed twice and three times respectively. It means that tanh plays a dominant role to affect the efficiency of a LSTM cell. As a result, tanh will reduce the execution efficiency of a LSTM cell. By analysing tanh function and its derivative, we can conclude that tanh is only sensitive to gradient changes near zero. Therefore, when multiple data are located far from zero, tanh cannot process them effectively. The gradients cannot be propagated in the deep network.

To solve problems caused by tanh, we introduce ReLU to replace tanh. ReLU is as formula (10).

$$ReLU(x) = max(0, x) \tag{10}$$

ReLU performs thresholding rather than exponentiation, which reduces computational complexity. Such an improvement can reduce the amount of exponentiation from 11 to 3 in a LSTM cell. It improves overall operating speed so that real-time requirements are met as much as possible in situation prediction. The derivatives are shown in Figure 3.

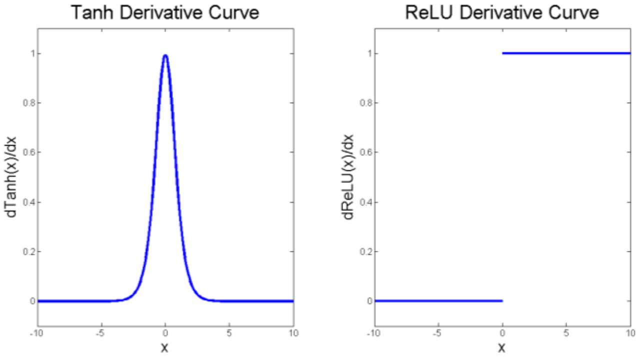


**Figure 3.** Comparisons between derivatives of tanh function and ReLU

The derivative of tanh decreases drastically as the independent variable moves away from zero. But the gradient remains stable and effective when it comes to the positive interval of ReLU. On the other hand, the gradient is constantly zero in the negative interval, which makes the neural network sparse and alleviates overfitting. Hence a LSTM cell can be improved as Figure 4.

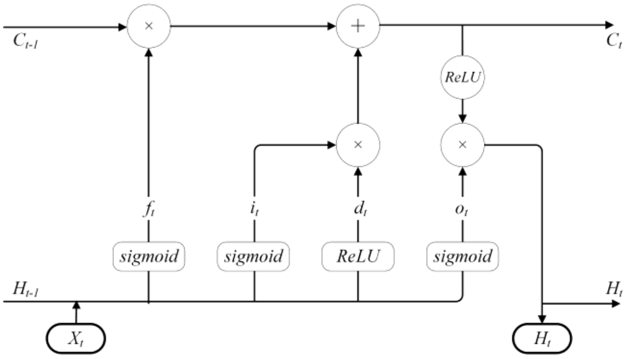


**Figure 4.** The structure of an improved LSTM cell

## 4.3. The Loss Evaluation Based on CE Function

Traditional artificial neural networks mainly choose Mean Square Error (MSE) to calculate losses and activate neurons. MSE function is as shown in formula (11).

$$MSE(u,v) = \frac{\sum_{i=1}^{n}(u_i - v_i)^2}{2n} \quad (11)$$

For a sample, the loss calculated by MSE is as formula (12).

$$MSE = \frac{(p-y)^2}{2} \quad (12)$$

In formula (12), *y* is the actual value, and *p* is the prediction.

The gradient of *w* is as formula (13).

$$\frac{\partial MSE}{\partial w} = (p-y)\cdot f'(wx+b) = (p-y)\cdot f'(z)\cdot x \quad (13)$$

The gradient of *b* is as formula (14).

$$\frac{\partial MSE}{\partial b} = (p-y)\cdot f'(wx+b) = (p-y)\cdot f'(z) \quad (14)$$

In formula (12), *f* is the activation function.

The gradients of parameters are proportional to the gradient of the activation function in back propagation of the network. There may be local minima when MSE is used for loss evaluation. At the same time, it is usually for regression. However, situation awareness involves more classification.

CE function is as formula (15).

$$CE(u,v) = -\frac{\sum_{i=1}^{n} u_i \log v + (1-u_i)\log(1-v)}{n} \quad (15)$$

For a sample, the loss calculated by CE is as formula (16).

$$CE = -y\log(p) - (1-y)\log(1-p) \quad (16)$$

The gradient of *w* is as formula (17).

$$\frac{\partial CE}{\partial w} = -\left(\frac{y}{p} - \frac{1-y}{1-p}\right)\frac{\partial f(z)}{\partial w} = (f(z) - y)\cdot x \quad (17)$$

The gradient of *b* is as formula (18).

$$\frac{\partial CE}{\partial b} = f(z) - y \quad (18)$$

Compared with MSE, the gradients of parameters in CE are proportional to the error between predicted and actual values.

To solve the problem caused by MSE, we try CE function. The second derivative of CE is always greater than or equal to zero. It means CE loss function is convex, while MSE loss function is non-convex. Therefore, CE function resolves the problem of massive local minima caused by MSE function. Compared with MSE employed to solve regression problems, it is more suitable to do classification in situation awareness. CE provides scientific and objective standards for loss evaluation.

## 4.4. A Framework for Network Security Situation Prediction

Prediction of network security situation is the third level of security situation awareness, which, on the basis of the extraction of situation factors, aims to present the security situation in the near future. It takes collected network security situation data as input and predicts quantifiable situation value of the next stage. To predict the security situation in the near future, what we need is not just LSTM. We mainly use LSTM to associate data. The framework for predicting network security situation involves processing data, training models, mapping to situation values and so on. The workflow of situation prediction is shown in Figure 5.

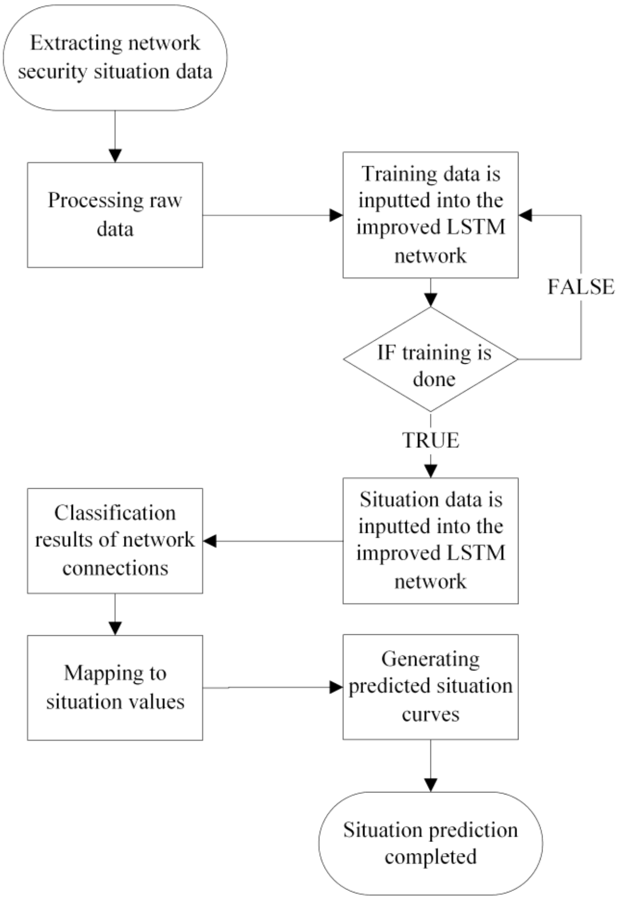

Extracting network security situation data
Processing raw data
Training data is inputted into the improved LSTM network
IF training is done
FALSE
TRUE
Situation data is inputted into the improved LSTM network
Classification results of network connections
Mapping to situation values
Generating predicted situation curves
Situation prediction completed

**Figure 5.** The flow chart of network security situation prediction

Acquired raw situation data requires pre-processed as the data source of the improved LSTM network. Labels and other discrete data can be dealt with only if they are presented in the form of continuous variables. In the initial stage, we need a certain amount of labeled situation data to train the neural network. Situation data generated by networks can be preprocessed and entered into the improved LSTM network. The trained model will analyse input data and output classification of every connection. We need quantitative situation values rather than classification results to describe the changing trend of the security situation to achieve prediction of network security situation, so we map classification results of network connections to situation values. Information of a single connection is not enough to describe the recent situation. There are logical associations among network connections, so we use the time step of LSTM to analyse relations and a sliding time window to display future security situation in networks. When we train the models, we need to match past and present indicators with state labels of the next phase because what we want to achieve is to predict security situation in the near future. To describe the changing trend of the situation vividly, we use a sliding time window to generate security situation curves.

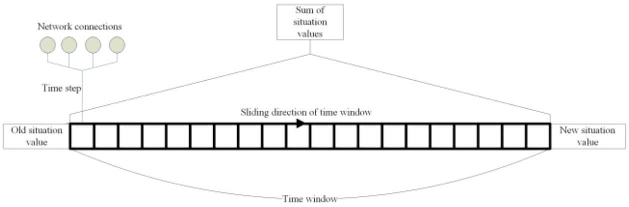


**Figure 6.** A sliding time window for generating security situation curves

Figure 6 shows that the time window slides from the old values to the new values. Sum of situation values in the time window is calculated to present future security situation. Time step decides how many connections to analyse at one time. The size of the sliding time window decides how long the presentation of security situation will cover. The time step of LSTM is used in both training and predicting. But the sliding time window is only used in predicting to make a large-scale presentation of evolutionary trends in the near future. The method can make a presentation more wholly and avoid the instability of changing trends. It enhances tolerance to a single judgment error, but the error cannot be avoided. It may also increase the amount of calculation to some extent. Finally, the prediction of the network security situation in the near future is achieved.

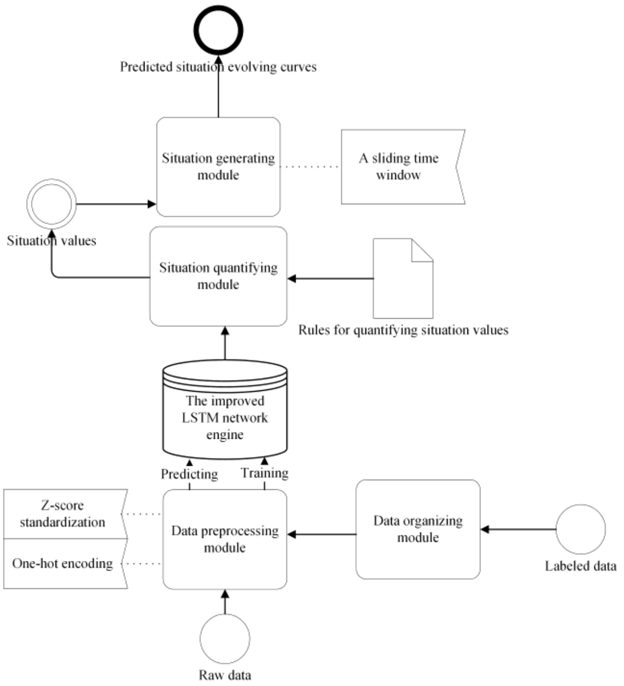

Predicted situation evolving curves
Situation generating module
A sliding time window
Situation values
Situation quantifying module
Rules for quantifying situation values
The improved LSTM network engine
Predicting
Training
Z-score standardization
One-hot encoding
Data preprocessing module
Data organizing module
Labeled data
Raw data

**Figure 7.** The architecture of the framework for predicting security situation

The architecture of the framework for predicting network security situation based on the improved LSTM is shown in Figure 7. The data preprocessing module and the data organizing module will be introduced in the next section. The improved LSTM network is the decision engine in the framework. The situation quantifying module needs rules that refer to actual conditions, which will be introduced in the next section as well. The situation generating module is based on the sliding time window.

Compared with the work of Xu et al. [8], the proposed method can predict different types of situations. Compared with the work based on rule inference and probability statistics, the framework has higher efficiency and achieves real-time situation prediction better. Compared with the work based on BP neural network and the work of Zhang et al. [17], the method effectively utilizes the time association between past situations and future situations. Introducing CE and ReLU, the framework reduces the time cost.

## 5. Experiments and Simulations

To test the performance of the framework we propose, we implemented the aforementioned neural network, using TensorFlow-GPU 2.0.0-beta0. The published KDD CUP 99 dataset and UNSW-NB15 dataset are used in experiments and simulations.

### 5.1. KDD CUP 99 Dataset

During experiments, processed description attributes of every connection are the input of the neural network. According to the labels in the dataset, connections can be classified into five categories. They are NORMAL, PROBE, DOS, U2R and R2L. Thus, the neural network finally outputs classification results.

Table 1. Corresponding situation values of different types of network connections

| Type of network connections | Situation value |
|---|---|
| NORMAL | 0 |
| PROBE | 1 |
| DOS | 2 |
| U2R | 3 |
| R2L | 4 |

As shown in Table 1, we defined corresponding situation values of different connections. According to our definition, the higher the situation values are, the more dangerous the network connections are. And a relatively big time window was employed to present security situation more macroscopically. It was easy to calculate the sum of situation values that are corresponding to the latest 25 connections in the time window. The sum is the index to describe the future trend of the security situation.

## Data Processing

Every record in the KDD CUP 99 dataset consists of 42 attributes. Among them, the first 41 dimensions are descriptions and the last one is a label. There are 7 discrete attributes and 34 continuous attributes in the description attributes. Discrete ones were processed by one-hot encoding. One-hot code introduces a k-bit status register to encode k status. One state corresponds to one binary, and only one bit is valid at any time. There will be 131 dimensions rather than 42 dimensions in the dataset. Examples of one-hot encoding are shown in Table 2.

Table 2. Examples of one-hot encoding

| No. | Before | Is_tcp | Is_udp | Is_icmp |
|---|---|---|---|---|
| 1 | tcp | 1 | 0 | 0 |
| 2 | tcp | 1 | 0 | 0 |
| 3 | udp | 0 | 1 | 0 |
| 4 | icmp | 0 | 0 | 1 |
| 5 | udp | 0 | 1 | 0 |

Standardization were performed in each dimension. Z-score standardization is as shown in formula (19).

$$x'_{ij} = \frac{x_{ij} - Mean_j}{MAD_j} \tag{19}$$

In formula (19), $x_{ij}$ is the *j*-th dimension attribute value of the *i*-th record. *Meanj* is the mean value of the *j*-th dimension attribute and *MADj* is the average absolute deviation of the *j*-th dimension attribute.

## Training of the Neural Network Model

There are many samples in the dataset so that we can use 10% of the dataset to train each time. 10% of the training set consists of 494021 samples. To train the models to predict the next one connection according to current and past data, we chose 494020 pairs to train. In the improved LSTM network, we took the data of recent records as a group to input into the model and the loss between the output result and the label of a future record was calculated. In BP neural networks, we just took a single record as input and calculated the loss between the output and a future record. In this way, we can analyse the changing trend of the security situation in the near future with the past data. Training performance can be measured by the means of multiple training results. It is different from the traditional method when records are sampled from the whole dataset. To persist logical associations between network flows in different stages, we need to keep sampled records in time order to compose the training set. It's different from simple random sampling. In this way, the improved LSTM model can learn correct associations among past, current and future situations. The generation of the training data in LSTM is as shown in Figure 8.

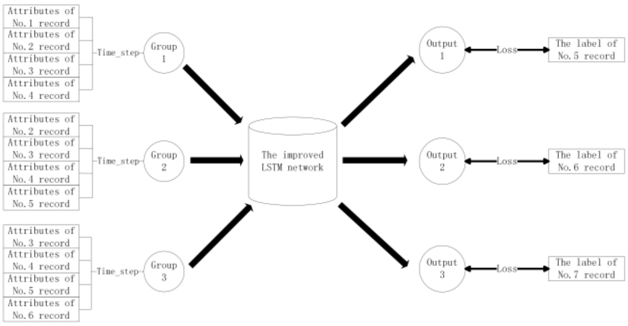

Attributes of No.1 record
Attributes of No.2 record
Attributes of No.3 record
Attributes of No.4 record
Time_step
Group 1
Attributes of No.2 record
Attributes of No.3 record
Attributes of No.4 record
Attributes of No.5 record
Time_step
Group 2
Attributes of No.3 record
Attributes of No.4 record
Attributes of No.5 record
Attributes of No.6 record
Time_step
Group 3
The improved LSTM network
Output 1
Loss
The label of No.5 record
Output 2
Loss
The label of No.6 record
Output 3
Loss
The label of No.7 record

**Figure 8.** Generation of the training data in LSTM

We separately implemented a BP neural network with MSE function and sigmoid function (MSE-BP), a BP neural network with CE function and ReLU (CE-BP), an improved LSTM neural network with CE function and ReLU (CE-LSTM). The number of neurons in the neural network is 18-18-18-5. Loss curves are shown in Figure 9.

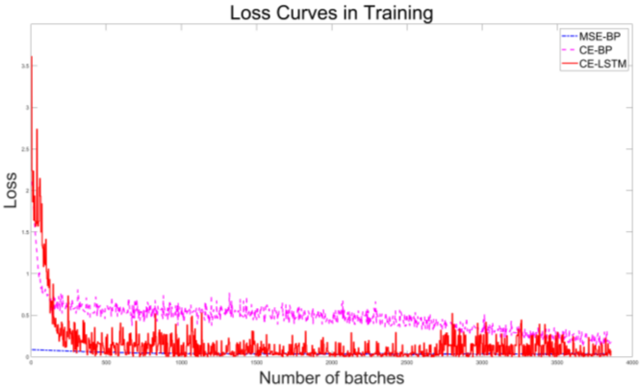


**Figure 9.** Loss comparison of different network models during the training process

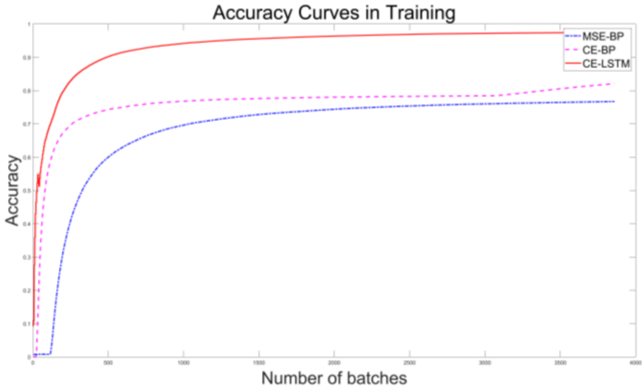


**Figure 10.** Accuracy comparison of different network models during the training process

As shown in Figure 9, the improved LSTM network proposed in the paper is with better convergence. Its loss decreases rapidly from the highest level. We find that the improved LSTM has learnt features of the training data. The loss of CE-BP is converged to a higher level. It means CE-BP doesn't fit the training set as well as CE-LSTM does. The loss of MSE-BP keeps relatively stable and is very low. We don't think it can predict very well. The reason leading to the result may be that MSE acts as the loss function.

Table 3. Training effectiveness of different neural network models

| Network models | Converged loss | Accuracy(%) |
|---|---|---|
| MSE-BP | 0.0424 | 74.58 |
| CE-BP | 0.1805 | 82.11 |
| CE-LSTM | 0.0676 | 97.54 |

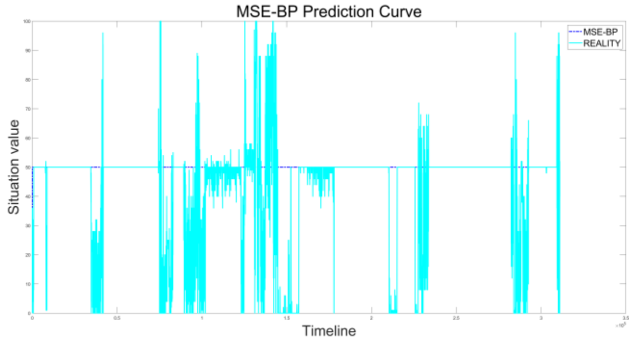
As shown in Table 3, the converged loss of CE-LSTM is close to that of MSE-BP. The loss of CE-BP is relatively higher than the other two models'. But when it comes to the accuracy, the result is different. The accuracy of MSE-BP is the lowest. The accuracy of CE-BP is nearly 10% higher than MSE-BP's. The accuracy of CE-LSTM can be up to 97.54%.

We exported accuracy data during training from tensorboard. Tensorboard is a component of tensorflow. Accuracy of different models changed in the training process. The accuracy comparison of different models in the training process is as shown in Figure 10.

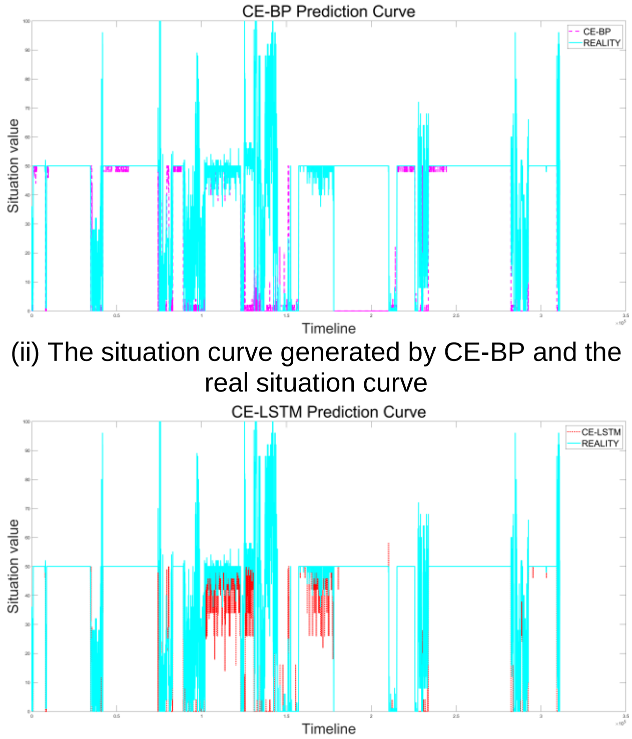
We can find that MSE-BP fits training data most slowly. And the level of accuracy it reaches finally is the lowest among the three models. The accuracy of CE-BP increases fast. And it is then maintained at a relatively stable level. Its final result is at the middle level. The accuracy of CE-LSTM increases rapidly from a very low level. It is trained nearly as fast as CE-BP. And it achieves the highest accuracy finally. In a word, CE-LSTM has the best performance in the training process.

## Testing of the Neural Network Model

The testing set was inputted into the trained neural network models to generate comparisons between situation predicted by models and real situation. There are 311029 samples in the testing set. We used the whole testing set to test prediction performance of different models. Situation curves are as shown in Figure 11.



(i) The situation curve generated by MSE-BP and the real situation curve



(ii) The situation curve generated by CE-BP and the real situation curve

(iii) The situation curve generated by CE-LSTM and the real situation curve

**Figure 11.** Comparison of situation curves predicted by different models

The prediction made by MSE-BP tells that the security situation is kept at a stable middle risk level. It's obviously different from the real situation. Compared with CE-LSTM, CE-BP mistakes more moderate risk situations for low risk situations. Comparisons show that the prediction made by the improved LSTM network is closer to real situation. Compared with other two methods, the improved LSTM network has the best performance.

Compared with the method in [17], the framework predicts the network security situation in real time with better performance. In the testing set, 21 points are depicted by [17] to predict the situation. However, our framework predicts the situation in a connection-level scale. The result means the framework makes real-time prediction better. Furthermore, it may guide automated network defence in the future.

Table 4. Prediction performance of different neural network models

| Network models | Prediction accuracy(%) |
|---|---|
| MSE-BP | 73.90 |
| CE-BP | 80.51 |
| CE-LSTM | 90.56 |

From the results shown in Table 4, we can see that the prediction accuracy of MSE-BP is the lowest. The accuracy of CE-LSTM is more than 10% higher than that of CE-BP. The improved LSTM network judges security situation more accurately. It can predict the intentions of 90% of network flows correctly. And the traditional neural network has poorer application capability.

To evaluate differences between predicted situation curves and real situation curves, we calculated the correlation coefficients between the predicted curves made by three models and the real curves. Pearson correlation coefficients, Spearman correlation coefficients and Kendall correlation coefficients are given in Table 5.

Table 5. Analysis of correlation coefficients

| Network models | Pearson | Spearman | Kendall |
|---|---|---|---|
| MSE-BP | 0.0124 | 0.0133 | 0.0124 |
| CE-BP | 0.5072 | 0.4958 | 0.4630 |
| CE-LSTM | 0.7052 | 0.7034 | 0.6629 |

The correlation coefficients represent the correlation degree between the predicted curves and the real curves. In other words, correlation coefficients show how well the predicted situation curves fit the actual security situation curves. The higher correlation coefficients are, the more accurate security situation prediction is. The correlation coefficients of the prediction made by MSE-BP are around 0.01 ($p<0.005$). It means that there is a very weak correlation between prediction and real situation. The prediction made by MSE-BP hardly reflects real changes. The correlation coefficients of the prediction made by MSE-BP are around 0.5 ($p<0.005$). Consequently, there is a moderate correlation between predicted situation trends and real situation trends. The correlation coefficients of the prediction made by MSE-BP are around 0.7 ($p<0.005$). We can take it as a strong correlation and conclude that the predicted situation curve fits the changing trend of real situation curve in the near future well.

## 5.2. UNSW-NB15 Dataset

The raw network packets of the UNSW-NB 15 dataset was created by the IXIA PerfectStorm tool in the Cyber Range Lab of the Australian Centre for Cyber Security for generating a hybrid of real modern normal activities and synthetic contemporary attack behaviors, and a partition from this dataset is configured as a training set and a testing set [23].

In the experiments, processed description attributes are inputted into the neural network. Network connections can be classified into ten categories here. They are Normal, Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms.

It is difficult to find out which cyber-attacks are more threatening intuitively because there are more types of attacks in the UNSW-NB15 dataset. To define corresponding situation values, we referred to the concept of cyber kill chain [24]. The cyber kill chain is a collection of processes related to cyber-attacks. It can describe how a complex attack is done step by step. The cyber kill chain is as shown in Figure 12.

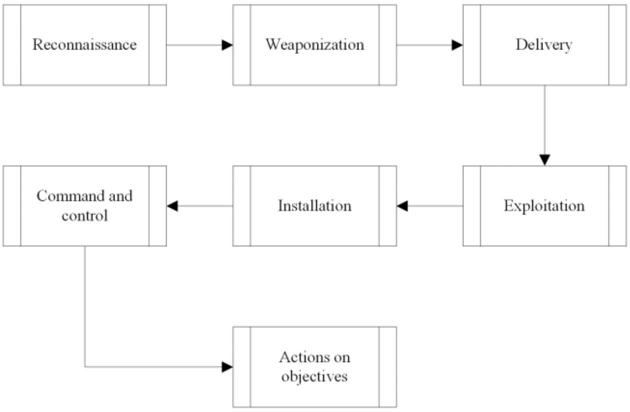


**Figure 12.** Cyber kill chain

There are some descriptions of different connection types in the dataset. Hence, we defined situation values based on both the cyber kill chain and descriptions in the UNSW-NB15 dataset. We supposed that attackers would do more damage if their attacks got closer to the end of the cyber kill chain. Different types of connections could find their places in the cyber kill chain so that we could tell which one was more dangerous. Corresponding situation values are as shown in Table 6. A higher situation value means that the network is in greater danger.

Table 6. Corresponding situation values of different types of network connections

| Type of network connections | Situation value |
|---|---|
| Normal | 0 |
| Analysis | 1 |
| Reconnaissance | 2 |
| DoS | 3 |
| Fuzzers | 4 |
| Generic | 5 |
| Shellcode | 6 |
| Worms | 7 |
| Exploits | 8 |
| Backdoors | 9 |

## Data Processing

There are 43 description attributes and 2 label attributes. Among the description attributes, there are 5 discrete attributes and 38 continuous attributes. The ID attribute is only the sequence number. We need multi-category labels rather than two-category labels, so we dropped the useless attributes. Then discrete ones were processed by one-hot encoding. There would be 200 dimensions in descriptions and 10 dimensions in labels.

Z-score standardization was performed in each dimension.

## Training of the Neural Network Model

There are 175341 records in the training set. We inputted the training set into three neural networks. To train the models to predict the security situation in the near future, we chose 175340 pairs to finish the training process. The training set of UNSW-NB15 is not as large as that of KDD CUP 99, so we used the training set to do 3 epochs of training. The number of neurons in the neural network is 35-35-35-10. Loss curves are as shown in Figure 12.

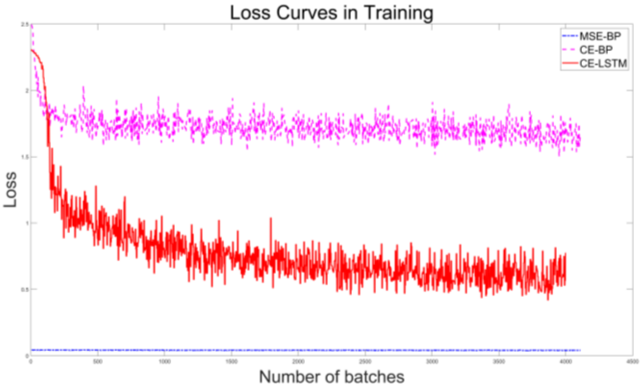


**Figure 13.** Loss comparison of different network models during the training process

From Figure 13, we can find that the improved LSTM can converge the loss as fast as CE-BP. The final loss of CE-LSTM is at a lower level than CE-BP's. The loss of MSE-BP is at a quite low level and doesn't increase or decrease. It may indicate that MSE-BP doesn't learn features in the dataset well. MSE-BP isn't good at dealing with classification in the dataset.

The performance is given in Table 7. The converged loss of each model is as displayed in Figure 13. The accuracy of MSE-BP during training is only 31.94%. The accuracy of CE-BP is higher than that of MSE-BP, but it is not up to 50%. CE-LSTM fits the training set much more accurately. Its accuracy is nearly 80%. In the scenario with more classifications, CE-LSTM performs much better.

Table 7. Training effectiveness of different neural network models

| Network models | Converged loss | Accuracy(%) |
|---|---|---|
| MSE-BP | 0.0417 | 31.94 |
| CE-BP | 1.5801 | 40.29 |
| CE-LSTM | 0.5673 | 77.51 |

To evaluate the performance of neural network models, we exported accuracy data from tensorboard as well. The accuracy curves are as shown in Figure 14.

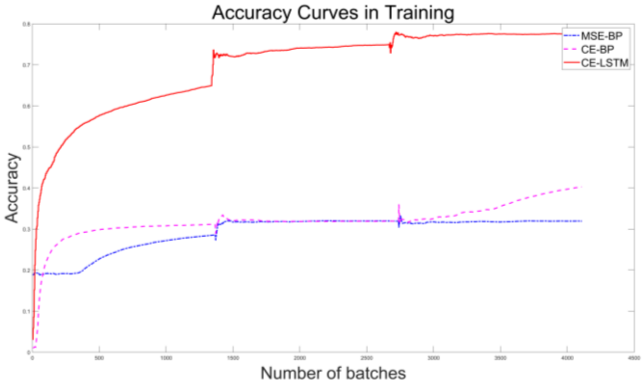
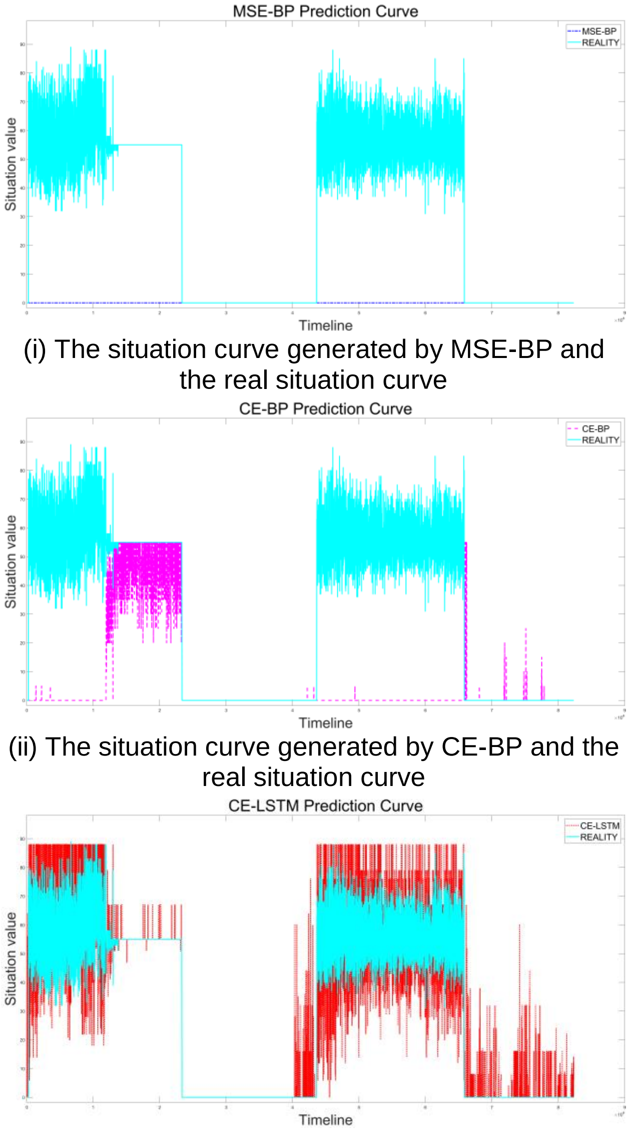


**Figure 14.** Accuracy comparison of different network models during the training process

We can easily find that the performance of the improved LSTM is better than the others' as shown in Figure 14. The accuracy of CE-LSTM increases faster and stays at a higher level. Just as we find from Figure 13, the improved LSTM learns features in the dataset better. The accuracy of CE-BP increases more slowly than CE-LSTM's at the beginning. It stays at a fixed level in most of the time and just increases a little finally. But it's much lower than the accuracy of CE-LSTM. MSE-BP learns much slower than other models. It just gets to the level of CE-BP finally. We may conclude that CE-LSTM gets a better performance in applications of network security situation awareness. The BP neural network isn't good at classifying for situation awareness. CE improves performance of BP neural network a little, but it can't change the essence of a BP neural network. LSTM, ReLU and CE contributed to the result together. LSTM established the time association between past data and future trends. ReLU reduced computation and avoided vanishing gradient problem. CE provided an appropriate standard for error calculation and speeded up the training process.

## Testing of the Neural Network Model

There are 82,332 records from different types in the testing set. We inputted the whole testing set into the three trained neural network models to generate situation prediction curves. Different situation curves are as shown in Figure 15.



(i) The situation curve generated by MSE-BP and the real situation curve

(ii) The situation curve generated by CE-BP and the real situation curve

(iii) The situation curve generated by CE-LSTM and the real situation curve

**Figure 15.** Comparison of situation curves predicted by different models

From Figure 15, we can find that network security situation curves generated by the models are different from each other. The cyan curves reflect the real situation in the network. MSE-BP can't predict how the security situation will evolve in the near future. It predicts that the network will be secure all the time. CE-BP can't predict most of middle-risk situations and high-risk situations. It predicts some steady middle-risk situations correctly and mistakes secure situations for low-risk situations finally. The improved LSTM predicts the future situation better. Compared with the other two models, CE-LSTM responds to moderate risk situations and high-risk situations more strongly. It predicts almost all medium risk situations and high-risk situations. Although it misjudges secure situations as risk situations at last, the misjudgement of secure situation is at a relatively low-risk level. We can distinguish levels of situation risks in different periods from the figure. Reviewing all the results, we can conclude that the improved LSTM has better performance than the other two.

Table 8. Prediction performance of different neural network models

| Network models | Prediction accuracy(%) |
|---|---|
| MSE-BP | 44.94 |
| CE-BP | 56.20 |
| CE-LSTM | 79.02 |

Data in Table 8 shows that different neural network models have different effectiveness when predicting the security situation on the UNSW-NB15 dataset. The accuracy of MSE-BP is not up to 50%. And CE-BP's accuracy is more than 10% higher than MSE-BP's. CE-LSTM predicts security situation much more accurately. Its accuracy is close to 80%. The performance differences between different models are greater here. It shows that the improved LSTM has much better performance when applied to predicting the security situation in complex environments.

To evaluate how well the models predict the changing trend of the security situation in the near future, correlation coefficients are shown in Table 9. We compare Pearson correlation coefficients, Spearman correlation coefficients and Kendall correlation coefficients here.

Table 9. Analysis of correlation coefficients

| Network models | Pearson | Spearman | Kendall |
|---|---|---|---|
| MSE-BP | NaN | NaN | NaN |
| CE-BP | 0.3131 | 0.2319 | 0.2007 |
| CE-LSTM | 0.9288 | 0.8452 | 0.6830 |

From Table 9, we can find that the prediction made by MSE-BP can hardly fit the change of the real situation at all. It means MSE-BP doesn't have the ability to predict risks. The correlation coefficients of CE-BP range from 0.2 to 0.4 ($p<0.005$). It means that there is a weak correlation between the prediction made by CE-BP and the real situation. Obviously, it won't work well enough in practice. Kendall correlation coefficient of CE-LSTM is close to 0.7 ($p<0.005$). Spearman correlation coefficient and Pearson correlation coefficient are about 0.9 ($p<0.005$), so there is a strong correlation between real trend and predicted trend. The framework built with LSTM can predict the security situation in the near future more accurately. With the existing dataset, CE-LSTM has the best performance among the three models. From the final results, CE is more suitable for predicting situation, and ReLU alleviates overfitting. LSTM effectively establishes the logical association between cyber-attacks.

## 6. Conclusions

After analysing related work on network security situation awareness, we find that existing neural networks applied to situation awareness are limited by their own structures. The framework built with LSTM proposed in the paper can be applied to network security situation awareness with serialized data. We have tested the performance of the framework thoroughly and have compared the differences among the three models in detail. Compared with the other two models, the framework based on the improved LSTM makes better results. The performance of the application is checked by experiments, and the framework has the basis for prediction of the security situation in real scenarios. Further research can further optimize the framework with algorithms and other structures to improve performance in prediction of the network security situation.

### Acknowledgements.

This research was funded by National Natural Science Foundation of China (No.61672206) and funded by Hebei Science Supported Planning Projects(No.20310701D). The authors would like to thank the reviewers for their detailed reviews and constructive comments, which have helped improve the quality of our study.

## References

[1] Endsley M. Design and Evaluation for Situation Awareness Enhancement. In Proceedings of the Human Factors&Ergonomics Society Annual Meeting; 1988. p. 97-101.

[2] Bass T. Multisensor Data Fusion for Next Generation Distributed Intrusion Detection Systems. In Proceedings of the Iris National Symposium on Sensor&Data Fusion; 1999. p. 24-27.

[3] Xi R, Yun X, Jin S, Zhang Y. Research survey of network security situation awareness. Journal of Computer Applications. 2012; 32: 1-4+59.

[4] Gong J, Zang X, Su Q, Hu X, Xu J. Survey of Network Security Situation Awareness. Journal of Software. 2017; 28: 1010-1026.

[5] Liu X, Wang H, Lyu H, Yu J, Zhang S. Fusion-based cognitive awareness-control model for network security situation. Journal of Software. 2016; 27: 2099-2114.

[6] Xiao J, Zhang, B, Luo, F. Distribution Network Security Situation Awareness Method Based on Security Distance. IEEE Access. 2019; 7: 37855-37864.

[7] Chen X, Zheng Q, Guan X Lin, C. Quantitative hierarchical threat evaluation model for network security. Journal of Software. 2006; 17(4): 885-897.

[8] Xu M, Li X, Liu H, Zhong C, Ma J. An intrusion detection scheme based on semi-supervised learning and information gain ratio. Journal of Computer Research and Development. 2017; 54: 2255-2267.

[9] Qiu H, Wang K, Yang H. Network alerts depth information fusion method based on time confrontation. Computer Engineering and Applications. 2016;36:499-504.

[10] Wei Y, Lian Y, Feng D. A network security situational awareness model based on information fusion. Journal of Computer Research and Development. 2009; 46: 353-362.

[11] Hu H, Ye R, Zhang H, Yang Y, Liu Y. Quantitative method for network security situation based on attack prediction. Journal on Communications. 2017; 38: 122-134.

[12] Wang L, Wang B, Peng Y. Research the information security risk assessment technique based on Bayesian network. In International Conference on Advanced Computer Theory and Engineering; 2010. p. 600-604.

[13] Zhang Y, Tan X, Cui X, Xi H. Network security situation awareness approach based on Markov game model. Journal of Software. 2011; 22: 495-508.

[14] Chen W, Ao Z, Guo J, Yu Q, Tong J. Research on cyberspace situation awareness security assessment based on improved BP neural network. Computer Science. 2018; 45: 335-337+341.

[15] Zhu J, Ming Y, Wang S. Mechanism of security situation element acquisition based on deep auto-encoder network. Journal of Computer Applications. 2017; 37: 771-776.

[16] Jin X, Li L, Su G, Liu X, Ji J. Prediction about network security situation of electric power telecommunication based on spark framework and PSO algorithm. Computer Science. 2017; 44: 366-371.

[17] Zhang R, Zhang Y, Liu J, Fan Y. Network security situation prediction method using improved convolution neural network. Computer Engineering and Applications. 2019; 55: 86-93.

[18] Li G, Ma Y. Feature extraction algorithm of air combat situation based on deep neural networks. Journal of System Simulation. 2017; 29: 98-105+112.

[19] Dou S, Zhang G, Xiong Z. Anomaly detection of process unit based on LSTM time series reconstruction. CIESC Journal. 2019; 70: 481-486.

[20] Wielgosz M, Skoczeń A, Mertik M. Using LSTM recurrent neural networks for monitoring the LHC superconducting magnets. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment. 2017; 867: 40-50.

[21] Salman A.G, Heryadi Y, Abdurahman E, Suparta W. Single Layer & Multi-layer Long Short-Term Memory (LSTM) Model with Intermediate Variables for Weather Forecasting. Procedia Computer Science. 2018; 135: 89-98.

[22] Guedes V, Junior A, Fernandes J, Teixeira F, Teixeira J.P. Long Short Term Memory on Chronic Laryngitis Classification. Procedia Computer Science. 2018; 138: 250-257.

[23] Moustafa N, Slay J. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In Proceedings of 2015 Military Communications and Information Systems Conference (MilCIS); 10-12 Nov. 2015. p. 1-6.

[24] Yadav T, Rao A.M. Technical Aspects of Cyber Kill Chain. In International Symposium on Security in Computing and Communications. SSCC 2015. Communications in Computer and Information Science; 2015. p. 438-452.